

IBM TXSeries AIX 版



CICS 管理指南

版本 4.2

IBM TXSeries AIX 版



CICS 管理指南

版本 4.2

注意事项

在使用该信息和它支持的产品之前，请阅读 第291页的『公告』中的一般信息。

第一版（1997 年 12 月）

本版本适用于：

IBM TXSeries AIX 版，版本 4.2，程序号 5697-D17

IBM TXSeries Solaris 版，版本 4.2，程序号 5697-D18

IBM TXSeries Windows NT 版(Gradient DCE)，版本 4.2，程序号 5697-D20

IBM TXSeries Windows NT 版(IBM DCE)，版本 4.2，程序号 5697-D21

IBM TXSeries HP-UX 版，版本 4.2，程序号 5697-D22

以及所有的后续版本、发行版和修订本，除非在新版本中另有特别说明。请参考可用的系统书目的最新版本，获得这些产品的当前信息。

可以通过 IBM 或 Transarc 代理或通过当地的 IBM 分部来订购出版物。

在本出版物的背后有一张“读者意见表”。如果您要提出意见，而这里描述的方式又不适用于您的话，请将它们寄往以下地址：

Transarc Corporation, The Gulf Tower,
707 Grant Street, Pittsburgh, PA 15219, USA.

当您向 IBM 或 Transarc 发送了信息后，就授予了它们非专有权，它们可以以任何认为适当的方法使用或散发这些信息，而不必对您负任何责任。

© Copyright International Business Machines Corporation 1993, 1997; Transarc Corporation, 1997. All rights reserved.

目录

图	xi
表	xiii
欢迎使用 CICS 管理指南.	xv
第1部分 配置与定制	1
第1章 管理 CICS 区域和资源	3
建立 CICS 系统管理员.	3
CICS 资源定义	4
定义资源.	4
资源定义类的列表	6
在资源定义中如何使用简写	7
CICS 系列成员之间的资源定义比较	7
使用 cicscp 建立 CICS 系统	8
cicscp 命令范例	9
cicscp 系统设定值	9
使用 SMIT 建立 CICS 系统.	10
元字符限制	12
使用 SMIT 用新资源定义刷新区域	13
使用 SMIT 添加一个区域.	13
使用 SMIT 删除区域	14
使用 SMIT 添加资源定义.	14
使用 SMIT 更改资源定义.	15
使用 SMIT 删除资源定义.	17
使用 SMIT 查看资源定义.	18
使用 CICS 资源定义命令.	19
用于建立 CICS 系统的命令	19
用于为产品区域作准备的命令	20
第2章 创建和配置 CICS 区域	21
区域的安全性考虑	21
配置共享的资源定义.	22
共享的区域定义 (RD)	22
共享的通信定义 (CD)	22
可功能附带的共享定义.	22
共享 SFS 区域定义 (RD)	23
定义可选择安装的资源.	23
在 SFS 上建立队列和文件管理.	24
用于建立 SFS 的方法	24
SFS 使用的资源定义	24
SFS 的安全性考虑	26
CICS 使用的 SFS 必须为何样	26
设置 FLT 环境变量.	27
在 DB2 上设置队列和文件管理	28
用于建立队列和文件的 DB2 的方法	28
DB2 文件管理的安全性考虑.	28
用于 DB2 队列和文件管理的资源定义	28

迁移队列和文件管理.	28
从 SFS 将队列和文件管理移动到 DB2	29
将队列和文件管理从 DB2 移动到 SFS	31
关于使用 cicscp 建立区域	32
cicscp 系统设定值	32
DCE 先决条件.	32
cicscp 使用的基本命令.	33
建立 CICS 日志的单独文件系统	33
区域和文件管理设置过程	33
更改新区域的区域定义 (RD) 属性	33
设置区域的语言	34
使用 cicscp start sfs_server 建立 SFS	34
使用 cicscp 删除区域	35
使用 cicscp 删除 SFS	36
第3章 配置 CICS 资源	37
建立事务处理定义 (TD)	37
配置事务处理定义 (TD)	37
配置不确定的条件	38
建立程序定义 (PD)	39
配置程序定义 (PD)	39
建立不同语言的基本映象支持 (BMS)	40
建立文件定义 (FD)	41
与数据文件工作时使用的过程	41
关于文件定义 (FD)	42
配置文件定义 (FD)	43
配置文件的模式文件定义 (SCD)	44
使用 cicsdb2import 建立 DB2 上的数据文件.	45
使用 cicssfimport 建立 SFS 上的数据文件	46
使用 cicsddt 建立 DB2 上的数据文件	47
使用 cicssdt 建立用户文件	48
建立临时存储器队列 (TSQ)	50
关于临时存储器队列 (TSQ)	50
配置临时存储器定义 (TSD)	51
建立瞬时数据队列 (TDQ)	52
关于瞬时数据队列 (TDQ)	52
配置瞬时数据定义 (TDD)	53
建立日志定义 (JD)	54
关于日志定义 (JD)	54
配置日志定义 (JD)	55
建立终端定义 (WD)	56
CICS 提供的自动安装模型	56
配置终端定义 (WD)	58
建立监控定义 (MD)	60
建立 Micro Focus COBOL 运行时间环境	61
何时使用 cicsmkcobol	61
建立应用程序服务器.	61
Micro Focus COBOL 程序和应用程序服务器.	62
建立产品定义 (XAD)	62
第4章 配置客户程序	65
客户的安全性考虑	65

使用开放系统上的 CICS (cicsterm) 客户	66
cicsterm 客户如何工作	66
为与 CICS 客户机使用建立 DCE	72
设置客户的语言	73
用 CICS 使用 telnet 客户	73
Telnet 客户和 cicsteld	74
使用 inetd 精灵程序启动 cicsteld	75
从操作系统命令行启动 cicsteld	78
使用 cicscp 启动 cicsteld	79
使用 telnet 客户连接到区域	79
破坏 telnet 服务器配置	79
IBM cics 客户机产品	79
侦听程序进程如何工作	80
必须与服务器定义匹配的客户定义	80
配置限制	81
ECI 和 EPI 客户应用程序	81
配置 IBM cics 客户机产品的区域	82
将新用户添加到区域	83
关于添加用户	83
更改 CICS 口令	83
如何为从 CICS 应用程序打印而建立 cicstermp	84
cicstermp 如何工作	84
若对 cicstermp 产生问题	85
启动 3270 打印机仿真器 (cicstermp)	85
 第5章 为使用关系数据库配置 CICS	 87
CICS 提供的数据库样本程序	87
用 CICS 使用关系数据库	87
单阶段落实优化	88
迁移考虑	89
应用程序开发	89
备份和恢复	89
诊断问题	89
SQL 数据库设置过程	90
启用到 DB2 的连接	90
启用到 Informix 的连接	93
启用到 Oracle 的连接	96
启用到 Sybase 的连接	100
 第6章 使用用户出口和用户程序定制	 105
CICS 用户出口	105
如何使用用户出口	105
使用动态事务处理路由选择用户出口	108
使用动态分布程序链接用户出口	111
用户程序	114
使用外部安全性管理器用户程序	114
使用终端自动安装用户程序	115
使用 IBM cics 客户机产品的自动安装程序	115
使用性能监控用户程序	116
使用后初始用户程序	117
使用关机用户程序	118

第2部分 操作 121

第7章 启动 CICS 系统	123
启动顺序	123
冷启动和热启动	124
有关冷启动 SFS 的警告	124
对于启动使用 cicscp.	125
启动过程	126
使用 cicscp start dce 启动 DCE	126
使用 cicscp start sfs_server 启动 SFS	126
使用 cicscp start region 启动区域	126
使用 cicscp start all 启动 CICS	127
CICS 启动的恢复和重新启动	127
在启动时资源定义的恢复	127
CICS 重新启动	128
恢复控制处理	128
逆序（现场）恢复处理	128
在启动时和运行事务时文件管理器不可用	128
在 CICS 启动时不可用的文件管理器的影响	128
在各阶段不可用的文件管理器的影响	130
不可用的文件管理器的其它影响	131
更改系统时钟的可能影响	131
第8章 关闭 CICS 系统	133
关闭顺序	133
CICS 关闭进程	134
正常关闭	134
立即关闭	135
取消（强制异常）关闭	135
软件启动关闭	135
未控制关闭	136
在关闭时产生的统计资料	136
打印软件启动的关闭转储	136
关闭过程	137
使用 cicscp stop region 关闭区域	137
使用 cicscp stop sfs_server 关闭 SFS	138
使用 cicscp stop dce 关闭 DCE	138
使用 cicscp stop all 关闭 CICS	138
第9章 使用 CICS RDO 命令更新区域运行时资源	141
如何在运行区域更新资源定义	141
通过使用 CICS RDO 命令的运行时资源数据库过程	142
把新的资源定义添加至运行区域	142
更改运行时中的资源定义	143
从永久数据库安装一个资源定义	143
仅从运行时删除资源定义	144
安装一组定义至运行时中	145

第3部分 备份和恢复 147

第10章 CICS 备份和恢复介绍	149
保护 CICS 系统和数据	149

协助备份、恢复和重新启动的机构.	149
同步点 (sync points)	150
两阶段落实处理	150
CICS 可恢复资源.	150
自动启动.	151
第11章 为 CICS 系统的恢复做准备	153
启用 DCE 服务器的恢复	153
初始配置后的备份	153
阶段备份.	153
测试 DCE 恢复过程.	154
备份 DCE	154
启用 SFS 的恢复.	155
备份 SFS 逻辑卷.	156
启用和禁用媒体归档.	157
启用 DB2 的恢复	158
启用版本恢复	158
启用失败恢复点	159
启用 CICS 的恢复	159
备份 CICS 日志	160
使用 cicsexport 和 cicsimport 用于备份和恢复	160
启用 CICS 子系统项的恢复	160
启用 CICS 用户标识符的恢复	161
启用应用程序文件的恢复	161
将应用程序文件恢复到故障点	161
将应用程序文件恢复到以前的备份级别	161
备份系统的建议	162
非产品的 CICS 系统	162
产品 CICS 系统	162
改进性能以恢复服务.	163
使用镜像卷	163
使用第二网络接口卡和额外的网络布线	163
使用第二处理器作为保留 (备份)	163
使用一个高可用性簇多重处理 (HACMP)	164
第12章 备份和恢复的实现	167
正在恢复 DCE.	167
SFS 队列和文件管理恢复过程	168
DB2 队列和文件管理恢复过程	169
完全故障后的完全恢复过程	170
步骤 1.安装 DCE、Encina 和 CICS	170
步骤 2.恢复 DCE 服务.	170
步骤 3.恢复队列和文件管理	170
步骤 4.恢复 CICS 服务	171

第4部分 性能 173

第13章 如何改进系统的性能	175
测量性能.	175
由操作系统提供的性能测量工具	175
CICS 提供的性能测量工具	175
由关系数据库提供的性能测量工具.	176

改进 CICS 系统的性能.	176
标识 CICS 性能约束	177
弱响应时间解决方案的摘要	179
可能影响性能的 CICS 设施	179
改进启动和关机的性能	180
查看物理内存来改进性能	181
通过交换空间来改进性能	182
对于性能的程序设计考虑	182
改进访问数据库的性能	185
用快速本地传送 (FLT) 来改进 SFS 的性能	186
最大吞吐量或最小响应时间的决定	186
I/O 调整	187
改进文件控制的性能	188
控制至 RPC 侦听程序进程请求的速率	189
控制区域中运行事务的数量数	190
分配工作负荷区域	190
用 SP2 改进性能	190
第14章 使用 CICS 工具测量性能.	193
收集统计	193
CICS 统计的信息收集	193
如何收集统计和格式化报告	194
监控性能	197
标准 CICS 信息	197
来自 CICS 应用程序的信息	197
在监控定义 (MD) 中必须定义的	198
如何控制 CICS 监控	201
CICS 监控设施输出的处理	202
跟踪 CICS 执行性能	202
使用辅助跟踪的一般注解	202
附录A. CICS 环境中的安全性	205
认证用户访问 CICS	205
CICS 用户认证概述	205
DCE 用户认证概述	206
用 DCE 委托人定义 CICS 用户	207
多注册用户的安全性	207
Telnet 客户的安全性考虑	208
cicsterm 客户的安全性考虑	209
XA 生效环境的安全性考虑	209
DCE RPC 和 ACL 如何维护系统的安全	210
用于设置已认证 RPC 保护级别的属性	210
了解 DCE 组和缺省 ACL	212
认证策略	213
赋予访问权给另一个单元的区域	213
使用带外部安全性管理器的 DCE ACL 外部安全性管理器 (ESM)	214
授权访问 CICS 事务	214
访问用户定义事务	214
运行安全事务的实例	215
授权访问 CICS 提供的事务	216
授权访问 CEDF 调试事务	216
事务安全性检查表	217

授权用户访问 CICS 资源	218
运行时间以外的访问	218
授权访问启动和关闭程序	218
授权用 CICS 命令访问资源定义	219
授权访问运行时间区域以外的资源	219
从应用程序授权访问 CICS 资源	220
使用 TSLKeyList 和 RSLKeyList 属性	221
资源安全性的更多信息	221
资源安全性检查表	224
使用外部安全性管理器	225
如何与 CICS 一起使用 ESM	225
定义 CICS 和 ESM 间的接口	226
启动时定义 ESM	229
定义 CICS 至 ESM	230
用 CICS 系统事务来使用 ESM	230
附录B. 使用 CICS 命令的 CICS 系统管理过程	231
有关使用基本命令来设置 CICS 的信息	232
使用基本命令的用来设置 CICS 的过程	232
用 cicsetupclients 设置 DCE 和 cicsterm 客户	232
创建区域的缺省用户标识符	233
使用 cicdefault 定义区域	233
使用 cicsexport 和 cicsimport 定义区域	234
为区域设置 CICS_HOSTS	235
创建 SFS 的用户标识符	236
准备 SFS 的物理存储设备	236
使用 cicssfscreeate 定义 SFS	237
设置 SFS 的联接文件	238
使用 cicssfss 启动 SFS	239
使用 cicssfssconf 设置队列	239
使用本地命令设置队列	240
为队列和文件管理设置 DB2	243
用 cicbdb2conf 配置资源和 DB2	246
从命令行启动区域	247
从命令行关闭区域	247
使用 cicssfsshut 关闭 SFS	248
使用 cicssdestroy 删除区域	249
使用 cicssfssdestroy 删除 SFS	249
附录C. 使用 DCE	251
DCE 先决条件	251
DCE 设置过程	252
将主机添加到现存的 DCE 单元中	252
将自身认证为 DCE 单元管理员	253
设置 CICS 客户的 DCE 请求	254
添加与 CICS 有关的组和对象	254
为仅 RPC 设置配置 DCE 客户	255
消除使用 cicsep 的 DCE 配置	255
从 DCE 客户主机中删除 DCE 客户	255
删除单元的 DCE 配置	256
将 CICS 从 DCE 单元移植到仅 RPC 设置	257
消除单元	260

从单元中删除所有的 CICS 主机	261
从单元中删除一些 CICS 主机	262
配置仅 RPC 设置	262
将 CICS 从仅 RPC 设置移植到 DCE 单元	265
移植具有 DCE 服务器的 CICS 主机.	265
移植只具有 DCE 服务器的 CICS 主机.	266
移植没有 CICS 的 PPC 网关服务器.	267
启动 CICS 系统	267
附录D. 配置 DBCS/MBCS 环境的 CICS.	269
使用 nlsString 数据类型	269
设置 LANG 变量.	269
建立和使用终端	270
建立 cicsteld	270
设置打印机	271
附录E. 使用 HACMP 的示例脚本	273
HACMP 示例脚本 1 (APPLID1)	273
HACMP 示例脚本 2 (APPLID2)	278
HACMP 示例脚本 3 (APPLID3)	281
HACMP 示例脚本 4 (APPLID4)	284
与以上示例相关联的脚本	286
公告	291
商标和服务标记	292
索引	293
读者意见表	311

1. 用户信息控制系统 (CICS) 版本 4.2 面板	11
2. AP、RM、TM 模型的 CICS 实现	62
3. 使用 CESN 更改 CICS 口令	84
4. 结合 SFS 备份镜像的媒体恢复归档 (MRA)	157
5. 方案 1	258
6. 方案 2	259
7. 方案 3	260
8. APPLID1 - 准备	273
9. APPLID1 启动	274
10. APPLID1 恢复	276
11. APPLID2 准备	278
12. APPLID2 启动	279
13. APPLID2 恢复	280
14. APPLID3 准备	281
15. APPLID3 启动	282
16. APPLID3 恢复	283
17. APPLID4 准备	284
18. APPLID4 启动	285
19. APPLID4 恢复	286
20. cterminate	287
21. locate_terminate	287
22. hacmpsetup	288
23. hacmpappl.	289

表

1. 管理指南导向图	xv
2. 本书中使用的约定.	xv
3. 配置与定制的导向图.	1
4. 有关管理 CICS 区域和资源的信息导向图	3
5. CICS 管理所使用的环境变量.	3
6. 资源定义类型	7
7. 建立区域和文件管理器的导向图.	21
8. 队列名称值 (其中 %R 为区域名)	30
9. 为区域准备运行 CICS 应用程序的导向图	37
10. TSQ 文件和索引的名称.	51
11. TDQ 文件名.	53
12. 定制用户和客户程序的导向图	65
13. 终端功能比较	69
14. 关系数据库配置导向图	87
15. SQL 数据库设置过程的导向图	90
16. 每个事务环境必需的过程步骤	90
17. 每个事务环境必需的过程步骤	94
18. 每个事务环境必需的过程步骤	97
19. 每个事务环境必需的过程步骤	101
20. 使用用户出口和用户程序定制的导向图	105
21. 用户出口号	108
22. 操作的导向图	121
23. 启动 CICS 系统的导向图	123
24. 在 CICS 启动时不可用的文件管理器的影响	129
25. 在 CICS 处理的各阶段一个不可用的文件管理器的影响	130
26. 其它文件管理器的不可用性的影响.	131
27. 关闭 CICS 系统的导向图	133
28. 使用 CICS 命令更新区域运行时资源的导向图	141
29. 备份与恢复的导向图.	147
30. 准备备份和恢复的导向图	153
31. 备份和恢复实现的导向图	167
32. 测量和改进性能的导向图	173
33. 改进系统性能的导向图	175
34. 改进 CICS 系统性能的导向图.	176
35. CICS 响应时间检查列表	179
36. RPC 保护级别和性能	180
37. 启动 CICS 的性能开销.	180
38. 使用 CICS 工具测量性能的导向图.	193
39. 监控字段表	198
40. 安全性信息的导向图.	205
41. 已认证 RPC 的保护级别	211
42. 可用于为区域定义 (RD) 设置已认证 RPC 保护的属性	211
43. CICS 产品文件必需的许可权.	219
44. 缺省 ESM 参数字段长度	228
45. 使用 CICS 命令的 CICS 系统管理过程的道路映像.	231
46. 重新配置 DCE 的导向图	251
47. 配置 DBCS/MBCS 环境 CICS 的导向图	269

欢迎使用 CICS 管理指南

本书描述了可操作的 CICS 系统的定制和操作及其所有特性，并假设已按照 *快速入门* 中所描述的过程，安装并测试了 CICS 系统。本书未描述如何安装 CICS。

读者 本书专为负责定制并操作 CICS 的系统管理员而撰写。

本书对于熟悉 AIX 系统管理的人士大有裨益，因为您将能够使用到许多系统管理过程。

您可以无需任何 DCE 或 Encina 知识来建立起一套基本的 CICS 配置。但是，更加复杂的配置则需要一个或两个上述产品的知识。

若将使用 DB2 版本 2 数据库以进行文件管理，则您需要了解 DB2 管理。

导向图

本书包含指导信息。参考第xv页的表 1:

表 1. 管理指南导向图

若需...	参考...
建立并配置 CICS 系统。	第1页的『第1部分 配置与定制』。
启动或关闭 CICS 系统，或管理联机资源定义数据库。	第121页的『第2部分 操作』。
准备 CICS，为备份并恢复，或在系统故障后恢复 CICS。	第147页的『第3部分 备份和恢复』。
调整系统性能。	第173页的『第4部分 性能』。
保护系统免受未授权的访问。	第205页的『附录A. CICS 环境中的安全性』。
使用基本命令建立区域。	第231页的『附录B. 使用 CICS 命令的 CICS 系统管理过程』。
阅读在本书中使用的约定。	第xv页的表 2。
阅读有关开放系统上的 CICS 库和相关书籍。	在 <i>概念与功能</i> 中的“IBM TXSeries 库”。

约定

TXSeries 文档使用下列印刷和按键约定:

表 2. 本书中使用的约定

约定	含义
黑体	表明您必须如实使用的值，例如命令、函数、资源定义属性以及它们的值。指图形用户界面(GUI)时，黑体还表明菜单、菜单项、标签、按钮、图标和文件夹。
等宽字体	表明必须在命令提示上输入的文本。等宽字体还表明屏幕文本和代码示例。
斜体	表明需由您提供的变量值（例如，为 <i>fileName</i> 提供的文件名）。斜体还可以表示强调和书籍的标题。
< >	括起键盘上键的名称。
<Ctrl-x>	其中的 <i>x</i> 是键的名称，表明是控制字符组合。例如，<Ctrl-c> 代表在按 c 键的同时按住 Ctrl 键。
<Return>	指标签为 Return、Enter 或向左箭头的键。
%	表示不需要 root 特权命令的 UNIX 命令外壳提示。
#	表示需要 root 特权命令的 UNIX 命令外壳提示。
C:\	表示 Windows NT® 命令提示。

表 2. 本书中使用的约定 (续)

约定	含义
输入命令	当指示您“输入”或者“发出”一个命令时，输入命令，然后按 <Return>。例如，“输入 ls 命令”的指令是指在命令提示上输入 ls ，然后按 <Return>。
[]	在语法说明中，括起可选项。
{ }	在语法说明中，括起必须从中选择一项的列表。
	在语法说明中，分隔括在 { } (大括号)中选项列表的项。
...	语法说明中的省略号表明可以重复一次或多次前面的项。例子中的省略号表明为简洁起见，省略了信息。
IN	在函数说明中，表明其值用于将数据传递到函数的参数。这些参数不用于将已修改的数据返至调用例程。（不要在代码中包含 IN 说明。）
OUT	在函数说明中，表明其值用于将已修改的数据至调用例程的参数。这些参数不用于将数据传递到函数。（不要在代码中包含 OUT 说明。）
INOUT	在函数说明中，表明一些参数，它们的值传递到函数、由函数修改，并返至调用例程。这些参数同时起 IN 和 OUT 参数的作用。（不要在代码中包含 INOUT 说明。）
\$CICS 和 <i>prodDir</i>	表明安装 CICS 产品的全路径名（AIX 上是 /usr/lpp/cics ，其它 UNIX 平台上是 /opt/cics ，Windows NT 上是 C:\opt\cics ）。如果名为 CICS 的环境变量设置为该产品路径名，可以使用如上所示的例子；否则，必须用 CICS 产品路径名替换所有出现 \$CICS 的实例。
开放系统上的 CICS	请集中参考 CICS AIX 版、CICS Solaris 版和 CICS HP-UX 版产品。
CICS	请分别参考开放系统上的 CICS 和 CICS Windows NT 版产品。对开放系统上的 CICS 产品特定版本的引用用于突出开放系统上的 CICS 产品之间的差别。CICS 系列中的其它 CICS 产品由它们的操作系统（例如，CICS OS/2 版或 IBM 基于大型机的 CICS ESA、MVS 和 VSE 平台版）来区分。

第1部分 配置与定制

表 3. 配置与定制的导向图

若需...	参考...
有关管理 CICS 资源的背景信息。	第3页的『第1章 管理 CICS 区域和资源』。
建立区域，其中包括文件管理器。	第21页的『第2章 创建和配置 CICS 区域』。
准备用于运行应用程序的区域。	第37页的『第3章 配置 CICS 资源』。
准备最终用户能够连接到的区域。	第65页的『第4章 配置客户程序』。
建立 CICS 和 SQL 数据库之间的 XA 接口。	第87页的『第5章 为使用关系数据库配置 CICS』。
使用 CICS 用户出口或用户程序。	第105页的『第6章 使用用户出口和用户程序定制』。

第1章 管理 CICS 区域和资源

本章描述了 CICS 资源及用来配置那些资源的设施。

表 4. 有关管理 CICS 区域和资源的信息导向图

若需...	参考...
建立一个 CICS 系统管理员。	第3页的『建立 CICS 系统管理员』
查找到何为 CICS 资源定义，且如何使用它们。	第4页的『CICS 资源定义』。
查找到如何使用 cicscp 定义资源。	第8页的『使用 cicscp 建立 CICS 系统』。
阅读有关如何使用 SMIT 定义资源。	第10页的『使用 SMIT 建立 CICS 系统』。
阅读有关基本 CICS 命令。	第19页的『使用 CICS 资源定义命令』。

建立 CICS 系统管理员

许多过程要求您使用超级用户（root）特权注册入一个用户标识符。一些过程将无法运行除非您真正地作为超级用户注册。需要使管理更为简便，我们建议您建立起 CICS 管理的超级用户标识符，并在所有 CICS 管理的任务中使用该用户标识符。

许多过程可能还要求使用 DCE 委托人，它拥有执行特定任务的足够特权。不同于为这些步骤创建一个新委托人，使用 **cell_admin** 是因为它已拥有适当的特权而存在。若您倾向于创建另一个 DCE 用户，参考 DCE 文档，以获得更多信息。

需要建立 CICS 系统管理员：

1. 验证超级用户是 **cics** 和 **cicsterm** 组中的成员。
2. 若在 DCE 单元中工作，则必须作为单元管理员来进行 DCE 注册。将环境变量 **cell_admin_pw** 设置为 DCE 单元管理员的口令。这样可避免提示口令。

注：若未配置 DCE，则省略本步骤。

3. 将第3页的表 5 中列出的环境变量设置为包括指定的值。

表 5. CICS 管理所使用的环境变量

设置或扩充:	需要包括:
LANG	en_US
PATH	/usr/lpp/cics/bin /usr/lpp/dcelocal/bin /usr/lpp/dcelocal/etc /usr/lpp/encina/bin （若使用了 Encina SFS）
LIBPATH	/usr/lpp/cics/lib /usr/lpp/dcelocal/lib /usr/lpp/encina/lib （若使用了 Encina SFS）
NLSPATH	/usr/lib/nls/msg/%L/%N /usr/lpp/cics/msg/%L/%N /usr/lpp/nls/msg/en_US/%N /usr/lpp/dcelocal/lib/nls/msg/%L/%N /usr/lpp/encina/msg/%L/%N

表 5. CICS 管理所使用的环境变量 (续)

设置或扩充:	需要包括:
CICSREGION	区域的名称

在何处设置环境变量取决于系统的需求。因为所有 CICS 用户都需要这些值，可以将其设置在 `/etc/environment` 中。若并不需要让每个用户都拥有这些环境变量设置，则可以将其设置在超级用户的 `.profile` 和单独用户的概要中。

若将环境变量设置在 `/etc/environment` 或超级用户的 `.profile` 中，则需要返回注册到超级用户，以使这些值生效。否则，将其导出至外壳。

4. 若将为 CICS 队列和文件管理配置一个使用 DB2 数据库的区域，则必须建立管理员（如区域即将使用的管理员）。参考 DB2 文档，以获得更多信息。

相关信息

在 *CICS 管理参考大全* 中的“CICS 使用的环境变量”。

CICS 资源定义

本章描述了 CICS 资源定义。

定义资源

CICS 所需用于执行一项任务的任何事物都是 CICS 资源。CICS 资源可为下列任何一项:

- 连接和 Encina PPC 网关服务器
- 数据文件
- 用户日志
- 数据库管理器
- 用户程序
- Encina 结构化文件服务器
- 临时存储器队列
- 瞬时数据队列
- 终端
- 用户

需要被 CICS 使用，这些资源中的每项都需要资源定义。

如何使用区域资源定义数据库

用于存储资源定义的数据库有三种：*永久*、*自动启动*和*运行时间*。

永久数据库，包含在冷启动时，用于定义区域状态及其资源的定义。当建立区域时，初始创建该数据库。

当一个区域冷启动时，通过使用永久数据库中的资源定义子集，并行地安装新自动启动数据库和运行时间数据库。该子集包括：

- 那些 **ActivateOnStartup** 设置为 **yes** 的定义。
- 那些 **Group** 属性设置为在区域定义（RD）**Group** 属性中所列出的组名的定义。例如，一个事务的 **Group** 属性设置为 **Sales**，并且定义的组列表带有包含 **Sales** 的区域 **Group** 属性。无论事务的 **ActivateOnStartup** 属性是否设置为 **yes**，冷启动区域时，该事务的定义将安装在自动启动和运行时间的数据库中。

热启动时，运行时间数据库从资源定义的自动启动数据库中安装。区域从运行时间数据库运行。

永久、自动启动数据库在 `/var/cics_regions/regionName/database` 中。运行时间数据库仅在区域正在运行时存在。

更改数据库中的定义

永久数据库

永久数据库中的资源定义用 **cicsadd**、**cicsupdate** 和 **cicsdelete** 命令更新。

注：使用 **--P**、**--R** 和 **--B** 选项能够对 CICS 定义起到显著的作用。

自动启动和运行时间数据库

并行地更改这些数据库中的资源定义。

- 使用带 **-B** 选项的 **cicsadd** 命令，以添加一个新的资源定义。
- 使用 **cicsinstall** 命令，以安装一个资源定义组。
- 使用带 **-B** 选项的 **cicsupdate** 命令，以更新一个资源定义，该资源定义存在于永久数据库中，但是未在自动启动和运行时间数据库中安装。这种情况下，对永久数据库进行更新，资源定义并行地安装在自动启动和运行时间数据库中。

如果因为一些原因未能更新运行时间数据库（例如，若区域未在运行之中），则不更新自动启动数据库。

需要更改安装在自动启动和运行时间数据库中的资源定义，则必须首先使用带 **-R** 选项的 **cicsdelete** 命令，从运行时间删除该资源定义，然后，在永久数据库中作更改，并用 **cicsupdate** 命令将其安装回运行时间。

运行时间数据库

仅能够用 CEMT 设施和 EXEC CICS SET 命令对运行时间数据库作一些更改。但是，当区域关闭时，对运行时间数据库所作的任何更改都将丢失。

注：CICS 保留了一些运行时间数据库中的附加属性，以供内部使用。您对这些内部属性没有存取权。

何时能够更改数据库中的定义有特例：

- 除了下列属性之外，仅能够在永久数据库中更改区域定义（RD）：
 - 仅能够使用 CEMT 设施在运行时间数据库中更改 **MaxServer**、**MinServer**、**SysDump**、**PCDump** 和 **ABDump**。
 - 仅能够通过使用 EXEC CICS SET STATISTICS RECORDING 命令，在运行时间数据库中更改 **StatsRecord**。
 - 除了 **MonitorStatus** 属性之外，监控定义（MD）只能够在永久数据库中更改。可以通过使用 CEMT 设施更改运行时间数据库中的 **MonitorStatus** 属性。

- 与所有其它资源类不同的是：产品定义（XAD）在热启动时从永久数据库装入。仅当执行紧急重启动时，XAD 才从自动启动数据库中装入。

Encina 服务器的资源定义

在 /var/cics_servers 文件系统中有一组用于 Encina 服务器的资源定义。这些将用于当服务器启动时将参数传递到 Encina，并且帮助将数据文件添加到文件管理器。不存在与这些资源定义相关联的运行时间数据库。

资源定义类的列表

下列列表包含可定义资源的类：

通信定义（CD）

CICS 使用通信定义（CD）来描述能够与 CICS 区域通信的其它系统配置，其中包括其它 CICS 系列系统。每个 CD 项描述了一个远程系统。

文件定义（FD）

CICS 使用文件定义（FD）定义区域中可用的数据文件。可以在 FD 中定义远程和本地的文件。本地 FD 提供了对在任何 SFS（定义在本区域的结构化文件服务器定义（SSD）中）中文件的存取方法。

网关定义（GD）

CICS 使用网关定义（GD）定义 PPC 网关服务器。

网关服务器定义（GSD）

CICS 使用网关服务器定义（GSD）启动并停止 PPC 网关服务器。

日志定义（JD）

CICS 使用日志定义（JD）描述用户日志。

侦听程序定义（LD）

CICS 使用侦听程序定义（LD）运行进程，这种进程始终“侦听”来自 IBM cics 客户机产品跨越 TCP/IP 或本地 SNA 的连接请求。

监控定义（MD）

CICS 使用监控定义（MD）描述 CICS 执行的监控操作（数据集合）。这是一类包括了单个记录的特殊资源类。

产品定义（XAD）

CICS 使用产品定义（XAD）保留与其它事务型产品接口所需的信息，例如，使用 X/Open XA 协议的 XA 遵从的数据库。

程序定义（PD）

CICS 使用程序定义（PD）定义程序、映象集和表。可以在 PD 中定义远程和本地的程序。

区域定义（RD）

CICS 使用区域定义（RD）定义 CICS 初始化进程所使用的全部参数。RD 包括用于初始化一个区域的参数和用于配置 SFS 的参数。

模式文件定义（SCD）

CICS 使用模式文件定义（SCD）定义模式文件所使用的的所有参数，模式文件由 SFS 服务器依次使用。这是一类包括了单个记录的特殊资源类。

结构化文件服务器定义（SSD）

CICS 使用结构化文件服务器定义（SSD）定义区域可用的所有 SFS 服务器。

临时存储器定义（TSD）

CICS 使用临时存储器定义（TSD）定义临时存储器队列的模板数据标识。CICS 需要 TSD 项以求安全性、可靠性和远程目的。因为指定了类属数据标识，故任何以与类属标识（在 TSD 中）相同字符开头的唯一临时存储器（在程序中动态地生成）都能够自动地获取与 TSD 项相同的属性。可以在 TSD 中定义远程和本地的队列。

终端定义（WD）

CICS 使用终端定义（WD）定义区域中可用终端的配置，包括可自动安装的终端模型。可以在 WD 中定义远程和本地终端。

事务处理定义（TD）

CICS 使用事务处理定义（TD）标识并初始化事务。CICS 需要该定义来验证进入请求，以启动事务并提供有关事务的信息。可以在 TD 中定义远程和本地的事务。

瞬时数据定义（TDD）

CICS 使用瞬时数据定义（TDD）定义本区域中可用的瞬时数据队列。可以在 TDD 中定义远程和本地的队列。

用户定义（UD）

CICS 使用用户定义（UD）定义终端用户信息。CICS 使用该定义来执行事务处理和资源安全性检查。

在资源定义中如何使用简写

CICS 提供下列简写方法，用来指定一些 CICS 属性的值：

%R 或 %r 扩展为当前区域名。

%H 或 %h 扩展为当前主机名。

%S 或 %s 扩展为当前 SFS 或 PPC 网关服务器的**短名**。

无论简写何时发生，CICS 都将扩展简写符号。若需要将百分符号（%）作为文字表达，则使用两个连续的百分符号。

CICS 系列成员之间的资源定义比较

其它 CICS 系列成员为资源存储器使用表格。而开放系统上的 CICS 使用区域相关的存储器定义。第7页的表 6显示两者之间的相应关系：

表 6. 资源定义类型

其它 CICS	开放系统上的 CICS 名称
目的控制表（DCT）	瞬时数据定义（TDD）
文件控制表（FCT）	文件定义（FD）
日志控制表（JCT）	日志定义（JD）
程序控制表（PCT）	事务处理定义（TD）
程序列表（PLT）	这些包含在区域定义（RD）

表 6. 资源定义类型 (续)

其它 CICS	开放系统上的 CICS 名称
处理程序表 (PPT)	程序定义 (PD)
系统初始化表 (SIT)	区域定义 (RD)
注册表 (SNT)	用户定义 (UD)
终端控制表 (TCT)	终端定义 (WD) 通信定义 (CD)
临时存储器表 (TST)	临时存储器定义 (TSD)
事务处理列表 (XLT)	这包含在 事务处理定义 (TD)
监控表 (MCT)	监控定义 (MD)

相关信息

第21页的『第2章 创建和配置 CICS 区域』。

第37页的『第3章 配置 CICS 资源』。

快速入门 书籍。

每个资源都在 *CICS 管理参考大全* 中的“资源定义”有详细描述。

也可参阅“cicsupdate - 修改 CICS 资源定义”、“cicsadd - 添加 CICS 资源定义”、“cicsdelete - 删除 CICS 资源定义”、“cicsget - 获得 CICS 资源定义信息”和 *CICS 管理参考大全* 中的“cicsinstall - 安装 CICS 资源定义”。

使用 cicscp 建立 CICS 系统

CICS 控制程序 (**cicscp**) 通过在必要时使用系统设定值和一些强制加入的命名约定的方法，以使用来运行 CICS 区域必需的所有主机配置成为自动化。它也提供了在本框架内定制的灵活性。例如，**cicscp** 能够在主机上配置 DCE 服务器，或若服务器已在单元中配置，则 **cicscp** 能够仅配置 DCE 客户程序组件。另一个选择是只配置仅 RPC 设置的 RPC 端点映象器 (精灵程序)。

尽管 **cicscp** 能够完全自动地完成一项简单的配置，但是不建议使它成为配置所有必需组件的单一方法。应该认识到这点，并且使用其它 CICS 命令，以创建更复杂的配置。

cicscp 没有更新的命令，所以必须通过其它命令来经常改变一套现存的配置。建议在开始时通过 **cicscp** 进行所有必需的配置，除非需要一套更复杂的配置。

存在几种 **cicscp** 命令。用于配置的命令是 **cicscp create** 和 **cicscp destroy**。可在其上使用这些选项的一些对象是 **dce**、**ppcgwy_server**、**region** 和 **sfs_server**。

例如，需要通过使用系统设定定义来创建一个命名为 **regionA** 的区域，请使用下列命令：

```
cicscp create region regionA
```

该命令配置了创建区域的所有必需组件，尤其是 DCE、SFS 和区域本身。

该命令也创建了必需的操作系统用户帐户和区域使用的内部文件。**cicscp** 的处理取决于资源定义中包含的信息。但是，它并不建立 DCE 或 SFS（若已经配置）。

若 **cicscp** 能够检测出现存 DCE 或 SFS 的配置是无效的，则它发出一条信息，建议消除并重新配置该配置。

cicscp destroy 命令用于消除特定的对象。例如，若发出下列命令：

```
cicscp destroy sfs_server hostABCD
```

则消除了作为 SFS 的 hostABCD。但是，该命令不会消除任何使用 SFS 的区域，也不会消除为 SFS 创建的操作系统用户帐户或物理设备。

注：若执行的任务要求您作为 DCE 单元管理员注册，则会产生下列事宜：

1. **cicscp** 试图将使用 cell_admin_pw 环境变量定义的值作为口令注册。若定义了一个不正确的值，则 **cicscp** 失败。
2. 若不是用 cell_admin_pw 环境变量定义的值，则 **cicscp** 将试图使用 DCE 管理员帐户的系统设定口令。若正确，则 **cicscp** 作为 DCE 单元管理员注册。若不正确，则会产生下列事宜：
 - a. 若正在交互式地使用 **cicscp**，则 **cicscp** 提示口令。
 - b. 若正在将 **cicscp** 作为批处理使用，则 **cicscp** 失败。

cicscp 命令范例

例如，若使用下列命令：

```
cicscp -v -l /tmp/cicscp.log start region regionA
```

cicscp:

1. 在本地主机上配置并启动 DCE 客户程序精灵程序。（精灵程序是运行来执行标准服务的无人照管程序。DCE 客户程序精灵程序也指职员。）
2. 若必需的话，配置并启动 DCE 单元目录服务器和 DCE 安全性服务器。
3. 创建 SFS 的操作系统用户标识符。
4. 建立 SFS 逻辑卷。
5. 使用系统设定命名约定和选项，在本地主机上配置并启动 SFS。
6. 使用系统设定区域定义，配置并启动命名为 **regionA** 的区域。
7. 使用 **-l** 选项，产生一个活动的日志。范例使用 /tmp/cicscp.log。

cicscp 系统设定值

cicscp 使用下列系统设定值：

- 为区域创建的用户标识符是 **cics**。
- 区域定义（RD）**DefaultUserId** 属性的系统设定值是 CICSUSER。
- 在一个 DCE 单元中，为区域创建的 DCE 委托人是 **cics/regionName**。
- SFS 在主机名后命名。例如，若主机名为 hostABCD，则命名 SFS：

```
././cics/sfs/hostABCD
```
- 创建 SFS 的短名，短名由字母『S』和紧跟其后的主机名的前七个字符构成。例如，若主机名是 hostABCD，则短名是 **ShostABC**。

- 系统设定 SFS 用户标识符和 DCE 委托人使用短名命名。
- SFS 逻辑卷名称取决于短名。例如，若短名是 ShostABC，则命名队列和文件的逻辑卷：

sfs_ShostABC

并且命名用于队列和文件日志的逻辑卷：

log_ShostABC

注：资源定义命令和 SMIT 使用 **log_SFS_SERV** 和 **sfs_SFS_SERV** 作为 SFS 逻辑卷的系统设定名称。

- 每个逻辑卷的系统设置大小是 64 兆字节（MB）。若不需要 **cicscp** 创建这种大小的逻辑卷，则使用第236页的『准备 SFS 的物理存储设备』中描述的过程，在用 **cicscp** 命令之前创建逻辑卷。

参阅 *CICS 管理参考大全* 中的“cicscp - CICS 控制程序”，以获得有关 cicscp 更详细的内容。

使用 SMIT 建立 CICS 系统

系统管理界面工具（SMIT）也能够用于配置 CICS 区域和资源。它可用下列命令访问：

smit cics

该命令在调用 SMIT 之前检查 DISPLAY 环境变量。若该环境变量已设置，则调用 SMIT 接口的 Motif 版本。否则，调用字符基准的接口。

smitty cics

该命令调用字符基准的接口。本书的范例使用字符基准的接口显示。

注：需要使用这些命令，您需要以超级用户注册，并作为 DCE 单元管理员认证。

警告：请勿直接编辑资源定义文件。

初始的面板（字符基准的接口）如第11页的图 1中所示。

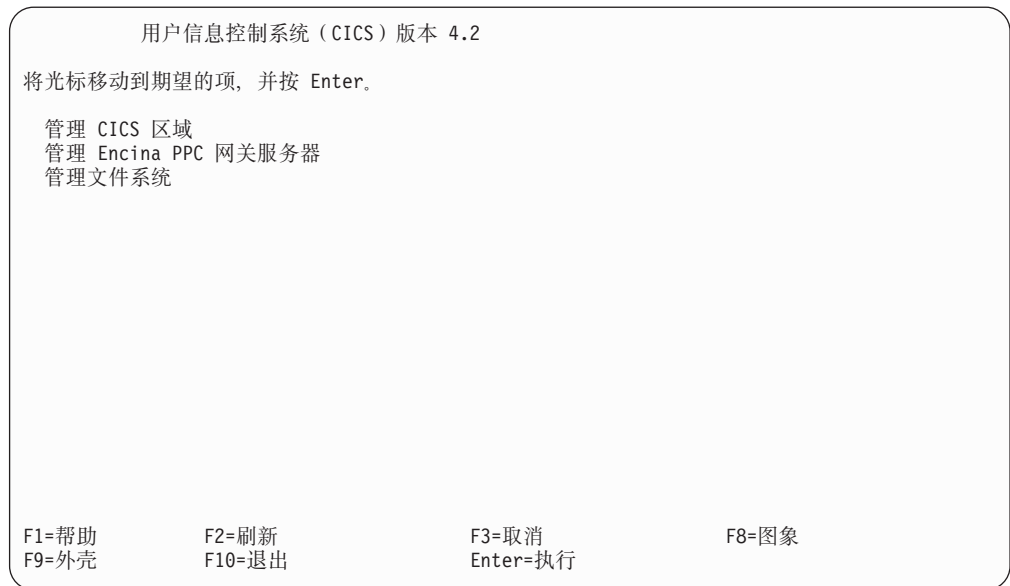


图 1. 用户信息控制系统 (CICS) 版本 4.2 面板

使用该面板, 选择:

管理 CICS 区域

创建、更改或删除与区域和区域资源相关联的资源定义。

管理 PPC 网关服务器

创建、更改或删除与 PPC 网关服务器相关联的资源定义。

管理文件系统

创建、更改或删除与文件管理器相关联的资源定义。

F1 (帮助)

显示本面板的帮助信息。

F2 (刷新)

刷新屏幕。

F3 (取消)

退出 SMIT。若这是一个随后的面板, F3 将返回到前一个面板。

F8 (图象)

将屏幕图象保存到一个日志文件 (在自用目录中)。

F9 (外壳)

创建外壳。

F10 (退出)

退出 SMIT。

Enter 执行面板当前显示的操作。在这种情况下, 该项目突出显示。

SMIT 有一条快速路径, 允许您直接转至特定的面板, 而不是通过屏面分层结构。SMIT 快速路径设施是直接从操作系统命令行显示 SMIT 对话面板或菜单的快捷方式。

元字符限制

在对永久数据库进行插入、更新、读出和删除操作时，CICS 检查组成数据库关键字的字节，以保证关键字不包含任何操作系统软件（如 SMIT）无法处理的字节。当遇到任何这样的字节时，CICS 报出一条错误信息，并拒绝对永久数据库的操作。这些字节是：

- 十进制数 0 至 32（自 NULL 到间隔字符）
- 十进制数 35（井号 [#]）
- 十进制数 39（单引号 [']）
- 十进制数 58（冒号 [:]）
- 十进制数 61（等号 [=]）
- 十进制数 92（反斜杠 [\]）
- 十进制数 127 至 255（自 DEL 到所有 8 位字符）

另外，对于所有插入、更新、读出和删除永久数据库的操作，以及在运行时间数据库的插入操作中，CICS 对字节执行特定类的验证，这些字节组成事务程序、程序、终端、通讯、用户、文件、临时存储器和瞬时数据的数据库关键字。

若正在使用 CICS 资源管理命令（如 **cicsadd**、**cicsupdate** 或 **cicsdelete**），并且希望输入包括元字符的调用（如 "（引号）、\（反斜杠）、（dollar）或 *（星号）），必须使用外壳跳出机制。例如，要输入值：

aa

可输入：

a\\a

或

a''a

SMIT 不支持在 SMIT 字段中的元字符用户。若需要指定包括元字符的值，则从 AIX 命令提示使用 CICS 资源管理命令。

需要有关在关键字中允许的字节值的信息，请阅读 *CICS 管理参考大全* 中下列标题中的关键字信息：

- “事务定义 (TD)”
- “程序定义 (PD)”
- “终端定义 (WD)”
- “通信定义 (CD)”
- “用户定义 (UD)”
- “文件定义 (FD)”
- “临时存储器定义 (TSD)”
- “瞬时数据定义 (TDD)”

SMIT 接口和 CICS 资源管理命令（如 **cicsupdate**、**cicsadd**、**cicsdelete** 和 **cicsget**）在 CICS 运行时间环境之外运行，并且执行该服务不需要与 CICS 区域接口。**cicsinstall** 命令需要运行 CICS，因为 **cicsinstall** 需要访问 CICS 运行时间数据库。

使用 SMIT 用新资源定义刷新区域

能够用下列刷新运行时间数据库:

- 来自特定组的资源定义。
- 来自永久数据库的资源定义, 其中将 **ActivateOnStartup** 属性设置为 **yes**。

注: 当更改或添加资源定义, 并且希望将更改应用到运行区域时, 需要使用该过程。

使用 SMIT 刷新运行时间资源数据库:

1. 输入:

smit

2. 选择应用程序。

3. 选择用户控制信息系统 (CICS) 版本 4.2。

4. 选择管理 CICS 区域。

如果需要的话, 选择更改工作 CICS 区域, 并且从列表选中适当的区域。

5. 选择 CICS 区域的定义资源。

6. 选择安装组。SMIT 显示可用区域的列表。当选中了一项时, SMIT 显示一个标题为“选择要安装的组”的面板, 列出所有可用的区域。

7. 使用 F7 选择要安装的组, 然后按 Enter 键。SMIT 显示确认所作选项的面板, 并且当按下 Enter 键时开始安装。

还有一条安装一组资源定义的快速路径方法。在操作系统命令行上输入 **smit cicsinstallgr**。

若试图从 SMIT 安装永久资源定义, 则 SMIT 总是产生一些信息, 指示任何已经安装的 **cicsinstall** 和已安装的资源总数。

使用 SMIT 添加一个区域

需要使用 SMIT 添加一个区域:

1. 输入:

smit cics

2. 当用户信息控制系统 (CICS) 版本 4.2 面板显示时, 选择管理 CICS 区域。

3. 在管理 CICS 区域面板上, 选择创建 (导入) 一个 CICS 区域。当下一个面板显示时:

a. 在创建的区域名字段中输入新的区域名。

b. 若正在从系统设定的资源定义创建区域, 则按 Enter。

c. 若正在从现存的资源定义组中创建区域, 则在输入文件字段输入区域的路径, 然后按 Enter。

区域定义位于下列目录中:

/var/cics_regions/regionName

其中 *regionName* 为选定的区域名。

4. 然后, 返回管理 CICS 区域面板, 并选择更改工作 CICS 区域选项。显示选择区域窗口, 其中列出新区域。

5. 选择新区域。显示命令状态面板。当 **Command: OK** 显示在左上角时，按 F3 返回到**管理 CICS 区域**面板。
6. 选择定义 **CICS 区域**的资源。它显示了定义 **CICS 区域**的资源面板。
7. 选择**管理资源**。
8. 当下一个面板显示时，选择要工作的资源类。

能够工作的资源定义类列在第6页的『资源定义类的列表』中。

能够在一个区域的区域定义和另一个区域永久数据库中的资源定义之间建立符号链接。参阅第22页的『配置共享的资源定义』，以获得如何指定共享资源的信息。

使用 SMIT 删除区域

使用 SMIT 添加一个区域:

1. 输入:
smit cics
2. 当用户信息控制系统 (**CICS**) 版本 **4.2** 面板显示时，选择**管理 CICS 区域**。
3. 选择**消除 CICS 区域**。当消除 **CICS 区域**面板显示时，在**区域名**输入字段中输入要删除的区域名称，并按 Enter。
4. 当**确认吗?** 面板显示时，按 Enter 继续。

除了那些为 SFS 定义的资源定义以外，所有资源定义都被删除。

使用 SMIT 添加资源定义

需要使用 SMIT 将资源定义添加到区域，请执行这些步骤:

1. 输入:
smit cics
2. 选择**应用程序**。
3. 选择**用户控制信息系统 (CICS) 版本 4.2**。
4. 选择**管理 CICS 区域**。
如果需要的话，选择**更改工作 CICS 区域**，并且从列表选中适当的区域。
5. 选择 **CICS 区域**的定义资源。
6. 选择**管理资源**。
7. 选择要添加的资源类。注意，对于区域和监控定义，仅允许更新现存的资源定义。
8. 选择**添加新的**。
选择模型...对话框允许将新定义基于现存的模型，或从系统设定值创建一个新定义。可以通过使用 **cicsupdate** 命令修改这些系统设定值。
9. SMIT 显示新资源的定义。输入属性值并按 ENTER。现在 SMIT 自动地更新适当的定义。

当正在添加用户时，将注意到 SMIT 面板上的一些附加字段。这些字段允许自动地将用户信息添加到 DCE。（不能够使用 **cicsupdate** 或 **cicsadd** 命令更改这些字段。）

字段为:

添加或更改 DCE 安全性注册表中的用户吗？

将此设置为 **yes**，若希望在该面板中输入的信息自动地创建用户的 DCE 委托人和帐户。

DCE 安全性组

输入 DCE 帐户的 DCE 组名。

DCE 安全性组织

输入 DCE 帐户的组织名称。

自用目录

输入 DCE 帐户的自用目录。

初始程序

这是 DCE 帐户的初始外壳。

DCE 委托人名取自**用户标识符**字段，全名取自**资源说明**字段。

忽略错误吗？

将该字段设置为 **yes**，以允许部分成功的操作能够再确认。正常情况下，若添加一个项目的操作发现该项目已经存在，操作将其作为错误处理。

还有一条添加资源定义的快速路径方法。需要使用快速路径，必须将环境变量 **CICSREGION** 作为区域名设置并导出。然后在操作系统命令行输入命令 **smit cicsaddxx**，其中 **xx** 是资源类的两或三个字符的助记符，如以下所示：

cd	通信
fd	文件
jd	日志
pd	程序
td	事务
tdd	瞬时数据队列
tsd	临时存储器队列
ud	用户
wd	终端
xad	产品
gd	网关
ld	侦听程序

例如，输入 **smit cicsaddfd** 添加文件定义（FD）。

显示适当的 SMIT 对话面板，按照先前步骤的第 8 步继续。

使用 SMIT 更改资源定义

作为替换，使用为简化系统管理任务而设计的交互式接口应用程序 **SMIT**。**SMIT** 显示一个分层菜单，带入交互式对话。**SMIT** 快速路径设施是直接从操作系统命令行显示 **SMIT** 对话面板或菜单的快捷方式。

执行这些步骤，以便修改 **CICS** 区域中的资源定义：

1. 输入:

smit cics

2. 选择应用程序。

3. 选择用户控制信息系统 (CICS) 版本 2。

4. 选择管理 CICS 区域。

如果需要的话, 选择更改工作 CICS 区域, 并且从列表选择适当区域。

5. 选择 CICS 区域的定义资源。

6. 选择管理资源。

7. 选择需要修改的资源类。

8. 选择显示/更改, 然后选择特定的资源显示或更新。

9. 更改所有必需的组件, 并按 Enter。SMIT 自动地更新适当的定义。

修改用户定义 (UD)

当正在修改用户时, 将注意到 SMIT 面板上的一些附加字段。这些字段允许自动地更新 DCE 的用户信息。(不能够使用 **cicsupdate** 或 **cicsadd** 命令更改这些字段。) 附加的字段为:

用户的 DCE 委托人。用户的 DCE 委托人名。仅当正在更改委托人时输入本字段。

添加或更改 DCE 安全性注册表中的用户吗? 将此设置为 **yes**, 若希望输入在该面板中的信息自动地更新用户的 DCE 委托人和帐户。

若更改了, 从 DCE 安全性注册表中删除用户吗? 若希望删除用户的旧委托人, 将此设置为 **yes**。请勿删除其除非委托人仅用于访问 CICS。

DCE 安全性组。DCE 帐户的 DCE 组名。

DCE 安全性组织。DCE 帐户的组织名称。

自用目录。DCE 帐户的自用目录。

初始程序。DCE 帐户的初始外壳。

DCE 委托人名取自委托人字段, 全名取自资源说明字段。

忽略错误吗? 将该字段设置为 **yes**, 以允许部分成功的操作能够再确认。正常情况下, 若添加一个项目的操作发现该项目已经存在, 操作将其作为错误处理。

还有一条更改资源定义的快速路径方法。需要使用快速路径, 必须将环境变量 CICSREGION 作为区域名设置并导出。然后在操作系统命令行输入命令 **smit cicsupdatexx**, 其中 **xx** 是资源类的两或三个字符的助记符, 如以下所示:

cd 通信

fd 文件

jd 日志

md 监控程序

pd 程序

rd	区域
td	事务
tdd	瞬时数据队列
tsd	临时存储器队列
ud	用户
wd	终端
xad	产品
gd	网关
ld	侦听程序

例如，输入 **smit cicsupdatetd** 更新事务定义。

显示适当的 SMIT 对话面板，按照先前步骤的第 8 步继续。

使用 SMIT 删除资源定义

作为替换，可使用 SMIT，它是为简化系统管理任务而设计的交互式接口应用程序。SMIT 显示一个分层菜单，带交互式对话。SMIT 快速路径设施是直接从操作系统命令行显示 SMIT 对话面板或菜单的快捷方式。

执行这些步骤，从 CICS 区域中删除资源定义：

1. 输入：

```
smit cics
```

2. 选择应用程序。

3. 选择用户信息控制系统（CICS）版本 2。

4. 选择管理 CICS 区域。

如果需要的话，选择更改工作 CICS 区域，并且从列表选择适当区域。

5. 选择 CICS 区域的定义资源。

6. 选择管理资源。

7. 选择需要删除的资源类。

8. 选择删除，然后使用 F7 从列表选择一个或一个以上定义。当完成选择后，按 Enter。

9. SMIT 显示要删除的资源定义。现在可以选择：

- 删除，从永久数据库中删除定义。
- 卸装，从运行时间数据库中删除定义。
- 两者，从永久数据库和运行时间数据库中删除定义。

若正在删除的资源是用户，则在 SMIT 面板上有一个附加的字段标题为从 DCE 安全性注册表删除用户吗？。若将该字段设置为 **yes**，则删除有关用户的 DCE 信息和来自 CICS 数据库的信息。系统设置值为 **no**，因为典型地一个用户属于许多区域，您可能希望仅从一个区域的数据库中删除用户信息。

10. 按回车键。SMIT 更新适当的数据库。

还有一条删除资源定义的快速路径方法。需要使用快速路径，必须将环境变量 **CICSREGION** 作为区域名设置并导出。然后在操作系统命令行输入命令 **smit cicsdelete_{xx}**，其中 **xx** 是资源类的两或三个字符的助记符，如下所示：

cd	通信
fd	文件
jd	日志
pd	程序
td	事务
tdd	瞬时数据队列
tsd	临时存储器队列
ud	用户
wd	终端
xad	产品
gd	网关
ld	侦听程序

例如，输入 **smit cicsdeleteud** 添加用户定义。

显示适当的 **SMIT** 对话面板，按照先前步骤的第 8 步继续。

使用 **SMIT** 查看资源定义

作为替换，使用为简化系统管理任务而设计的交互式接口应用程序 **SMIT**。**SMIT** 显示一个分层菜单，带交互式对话。**SMIT** 快速路径设施是直接从操作系统命令行显示 **SMIT** 对话面板或菜单的快捷方式。

执行这些步骤，以便在 **CICS** 区域中查询资源定义：

1. 输入：
smit cics
2. 选择应用程序。
3. 选择用户信息控制系统 (**CICS**) 版本 2。
4. 选择管理 **CICS** 区域。
如果需要的话，选择更改工作 **CICS** 区域，并且从列表选中适当的区域。
5. 选择 **CICS** 区域的定义资源。
6. 选择管理资源。
7. 选择需要显示或更改的资源类。
8. 选择显示/更改。
9. **SMIT** 显示所选中资源定义的所有属性值。若需更详细的信息，请阅读第143页的『更改运行时中的资源定义』。

还有一条查询资源定义的快速路径方法。需要使用快速路径，必须设置并导出环境变量 **CICSREGION** 作为区域名。然后在操作系统命令行输入命令 **smit cicsupdate_{xx}**，其中 **xx** 是资源类的两或三个字符的助记符，如以下所示：

cd	通信
fd	文件
jd	日志
md	监控
pd	程序
rd	区域
td	事务
tdd	瞬时数据队列
tsd	临时存储器队列
ud	用户
wd	终端
xad	产品
gd	网关
ld	侦听程序

例如，输入 **smit cicsupdatepd**，以从程序定义获得信息。

显示适当的 SMIT 对话面板，按照先前步骤的第 8 步继续。

使用 CICS 资源定义命令

可以使用基本 CICS 资源定义命令建立 CICS 系统。下列列表描述了一些最常用的命令：

用于建立 CICS 系统的命令

“cicssetupclients - 为 CICS 主机配置 DCE”

使用本命令建立 DCE 和 CICS 客户程序。它通常是第一个用于为 CICS 主机作准备的命令。

“cicssetupdce - 在一个 DCE 单元创建目录和组”

使用本命令添加 CICS 相关的组和对象。

“cicsdefault - 创建系统默认区域的副本”

本命令允许您从系统设定 CICS 区域的归档副本创建一个新的区域。

“cicsexport - 导出 CICS 资源定义”

本命令允许您获取一个现存 CICS 区域中所有资源定义的归档副本。与 **cicsimport** 命令一起使用 **cicsexport** 命令。

“cicsimport - 导入 CICS 资源定义”

使用本命令从一个现存 CICS 区域还原资源定义的归档副本。与 **cicsimport** 命令一起使用 **cicsexport** 命令。

“cicsdefaultservers - 为服务器创建缺省资源文件”

使用本命令创建 Encina 服务器的文件系统和系统设定资源定义文件。

“cicssfscscreate - 创建 CICS SFS”

使用本命令创建 CICS 使用的 SFS 的资源定义。

“cicssfscs - 启动 CICS SFS”

用 CICS 所必需的参数使用本命令，以便启动 SFS。

“cicssfscsconf - 配置 CICS SFS”

通过将 CICS 队列添加到 SFS，以使用本命令配置区域的该 SFS。

“cicssdb2conf - 配置 DB2”

通过将 CICS 队列添加到 DB2 数据库，以使用本命令配置区域的该 DB2 数据库。

“cicssstart - 启动 CICS 区域”

使用本命令启动区域。

用于为产品区域作准备的命令

当区域建立时，可以使用下列命令定义区域的资源：

“cicssadd - 添加 CICS 资源定义”

使用本命令添加资源的定义。

“cicssupdate - 修改 CICS 资源定义”

使用本命令更改资源的定义。

“cicssdelete - 删除 CICS 资源定义”

使用本命令删除资源的定义。

“cicssget - 获得 CICS 资源定义信息”

使用本命令查看资源的定义。

“cicssinstall - 安装 CICS 资源定义”

使用本命令将资源定义组从永久数据库安装到自动启动数据库和运行时间数据库中。

“cicssdt - SFS 诊断工具”

使用本命令将数据文件添加到 SFS。

“cicssddt - DB2 诊断工具”

使用本命令将数据文件添加到 DB2。

相关信息

第22页的『配置共享的资源定义』。

第21页的『第2章 创建和配置 CICS 区域』。

第37页的『第3章 配置 CICS 资源』。

第231页的『附录B. 使用 CICS 命令的 CICS 系统管理过程』。

若需所有 CICS 命令的完整描述，参阅 *CICS 管理参考大全* 中的“资源定义命令和脱机实用程序”。

第2章 创建和配置 CICS 区域

表 7. 建立区域和文件管理器的导向图

若需...	参考...
使区域不受未经授权访问的威胁。	第21页的『区域的安全性考虑』。
与另一个区域共享资源。	第22页的『配置共享的资源定义』。
将资源定义为可选择性地安装到运行时间。	第23页的『定义可选择安装的资源』。
在 SFS 上建立队列和文件管理。	第24页的『在 SFS 上建立队列和文件管理』。
启用或禁用 SFS 的“快速本地传送”。	第27页的『设置 FLT 环境变量』。
阅读在 DB2 上建立队列和文件管理。	第28页的『在 DB2 上设置队列和文件管理』。
将队列和文件管理从 SFS 移动到 DB2。	第29页的『从 SFS 将队列和文件管理移动到 DB2』。
将队列和文件管理从 DB2 移动到 SFS。	第31页的『将队列和文件管理从 DB2 移动到 SFS』。
建立区域。	第32页的『关于使用 cicscp 建立区域』。
更改刚建立区域的区域属性。 注意： 应该在开始使用产品环境中的区域之前执行该过程。需要冷启动一次，以便新属性值生效。	第33页的『更改新区域的区域定义（RD）属性』。
建立 CICS 日志的单独文件系统。	第33页的『建立 CICS 日志的单独文件系统』。
设置区域的语言。	第34页的『设置区域的语言』。
不建立区域地建立 SFS。	第34页的『使用 cicscp start sfs_server 建立 SFS』。
删除区域。	第35页的『使用 cicscp 删除区域』。
删除 SFS。	第36页的『使用 cicscp 删除 SFS』。

区域的安全性考虑

本部分中描述的过程告诉您如何使用或不使用 DCE 安全性服务配置区域。这确定了是否创建 DCE 委托人，以及是将区域定义（RD）中的 **AuthenticationService** 设置设置在 **DCE** 中还是 **CICS** 中。

若正在使用 DCE 安全性服务，则将保护级别设置为区域、SFS 和 PPC 网关服务器将收发“远程过程调用”（RPC）的级别上。这用区域定义（RD）**RuntimeProtection** 属性、网关服务器定义（GSD）**ProtectionLevel** 属性和结构化文件服务器定义（SSD）**ProtectionLevel** 属性来完成。需要有关保护级别的更多信息，请参阅 DCE 文档。切记，保护级别愈高，所需的认证时间愈长，生效的也愈慢。

发送 RPC 的一般策略是发送人的 **ProtectionLevel** 必须大于等于 RPC 的接收人。例如，因为区域将 RPC 发送到 SFS，则区域的 **RuntimeProtection** 必须大于等于 SFS 的 **ProtectionLevel**。类似地，因为 RPC 在区域和 PPC 网关服务器之间发送并接收，则区域的 **RuntimeProtection** 和 PPC 网关服务器的 **ProtectionLevel** 必须相同。参阅 *CICS 内部通信指南* 中的“认证跨越 SNA 连接的系统”，以获得更多信息。

相关信息

第65页的『客户的安全性考虑』。

DCE Administration Guide Module 5. DCE Security Service.

配置共享的资源定义

区域能够共享整个资源定义数据库或是一个或多个资源类的定义。共享整个数据库是建立测试区域的简便方法，但是，若一些资源类差别很大，则可能在产品区域之间不实用。建立并共享仅包含最公用的资源的组将会更简便。

建立共享的资源数据库有两种方法：

1. 删除区域的整个数据库目录或单独资源类，然后使用 **ln** 命令创建一个链路连到另一个区域的数据库目录或文件。
2. 当正在创建区域时，使用 **cicsexport** 和 **cicsimport** 命令自动地创建指定数据库类的符号链接。当正在创建归档副本（**cicsexport**）时需要执行它，则指定希望创建的链路类。然后若使用 **cicsimport** 为新区域使用在那个归档中的定义，则区域将已经具有从其到指定数据库类子目录的链路。需要更多有关这两个命令的信息，请参阅第234页的『使用 **cicsexport** 和 **cicsimport** 定义区域』以及 *CICS 管理参考大全*。

共享的区域定义（RD）

每个区域必须有一个唯一的 **LocalSysId**，它是区域定义（RD）中的一项属性。但是，若区域正在共享 RD，则每个区域自动地给出相同的 **LocalSysId**。需要避免这种情况，请确保当区域启动时 **LocalSysId** 属性已覆盖。参阅第247页的『从命令行启动区域』，以获得更多信息。

所有共享一组“区域定义”的区域必须用相同的 **-g groupId** 参数创建，（**cicsimport** 和 **cicsdefault**）或都没有。带有不同 **groupids** 的区域是无法共享资源的；它们对数据库没有存取权，因此不会启动。

共享的通信定义（CD）

当启动区域时，可能获得一条警告，告之本地系统存在一个定义。可以忽略该警告。

需要为每个区域创建一组共享相同数据库的 CD。

可功能附带的共享定义

当指定资源定义的 **RemoteName** 属性（可将其作为功能（TD、TDQ、TSQ、FD、PD）附带）时，使用空格或在其存在的机器上的资源名。

在本地系统上忽略 **RemoteName**。完成远程和本地信息的属性列表，因为它们的用法各不相同。

共享 SFS 区域定义 (RD)

当指定明确说明时，使用简写符号 **%R**。**% R** 扩展为本地区域名。若不使用 **%R**，所有 SFS 文件将被共享。参阅第7页的『在资源定义中如何使用简写』，以获得使用简写符号的更多信息。

定义可选择安装的资源

可以将资源定义为可选择性地从永久数据库安装到运行时间：

通过定义资源组

需要定义资源组，将 **Group** 属性设置为组名。可为下列资源类执行该步骤：

- 事务处理定义 (TD)
- 程序定义 (PD)
- 终端定义 (WD)
- 通信定义 (CD)
- 文件定义 (FD)
- 临时存储器定义 (TSD)
- 瞬时数据定义 (TDD)
- 产品定义 (XAD)
- 日志定义 (JD)
- 网关定义 (GD)
- 侦听程序定义 (LD)

组可以当区域冷启动时安装，也可以在运行时间安装。

通过定义冷启动时要安装的组

需要当区域冷启动时将组安装入运行时间，用区域定义 (RD) **Groups** 属性将组名定义为区域。该属性可包含一列表，例如，**销售单、工资单**等等。

通过定义要在运行时间内安装的组

若不希望组在冷启动时安装，但是，希望能够将其安装入运行区域，则不要将组名包括在 RD **Groups** 属性中。然后，可以使用 **cicsinstall** 命令，如第145页的『安装一组定义至运行时中』中所描述。

注：**ActivateOnStartup** 属性设置为 **yes**，本命令也可用于重新安装资源组或重新安装所有资源定义。

通过使用 **ActivateOnStartup** 属性

将 **ActivateOnStartup** 属性设置为 **no**，**Group** 属性为空白或组，该组未定义为冷启动时防止将资源定义安装到运行时间。需要将资源定义安装入运行区域，参阅第143页的『从永久数据库安装一个资源定义』。

相关信息

第4页的『定义资源』。

第141页的『第9章 使用 CICS RDO 命令更新区域运行时资源』。

在 SFS 上建立队列和文件管理

用于建立 SFS 的方法

注：本章不讨论将数据文件添加到 SFS。参阅第41页的『建立文件定义（FD）』。

所用的方法取决于：

- 是否已经建立 SFS。
- 是否 SFS 在 DCE 单元中。
- SFS 和区域是如何分布的。例如：
 - 若建立的区域将使用在本地主机上的 SFS，则使用 CICS 设施建立 SFS 并且添加上 CICS 队列。
 - 若建立的区域将共享一个在远程 CICS 主机上的现存 SFS，则使用 CICS 设施建立 SFS，但是使用原始 Encina 命令将 CICS 队列添加到 SFS。
 - 若建立的区域将使用在未安装 CICS 的主机上的 SFS，则使用原始 Encina 命令建立 SFS 并且添加上 CICS 队列。

使用第21页的表 7 中的指示建立 SFS。

SFS 使用的资源定义

本部分提供 SFS 上的队列和文件管理所使用的资源定义概述：

区域定义（RD）

通过用 **DefaultFileServer** 属性命名 SFS，配置要使用特定 SFS 的区域。该属性指向 SFS 的结构化文件服务器定义（SSD），并用于 SFS 配置过程。

对于 SFS 上存储的每种类型的信息，都创建一组文件、索引和归档，并指定唯一的名称。若更改区域定义（RD）中的名称，则必须保证新名称不与相同 SFS 中的任何其它文件名冲突。

CICS 提供的系统设定值使用区域名作为前缀。例如，用于本地队列非保护启动的文件所用的系统设定值为 **%Rcicsnlqfile**。

使用下列属性：

- 对于本地队列非保护性启动：
 - **LocalQFile** - 文件名
 - **LocalQIndex** - 索引
 - **LocalQMaxRecs** - 最大记录数
 - **LocalQPrePages** - 预分配的页面数

- **LocalQVol** - 卷名
- 对于本地队列保护性启动:
 - **LocalQProtectFile** - 文件名
 - **LocalQProtectIndex** - 索引
 - **LocalQProtectMaxRecs** - 最大记录数
 - **LocalQProtectPrePages** - 预分配的页面数
 - **LocalQProtectVol** - 卷名
- 对于逻辑可恢复瞬时数据队列:
 - **LogicalTDQFile** - 文件名
 - **LogicalTDQIndex** - 索引
 - **LogicalTDQMaxRecs** - 最大记录数
 - **LogicalTDQPrePages** - 预分配的页面数
 - **LogicalTDQVol** - 卷名
- 对于不可恢复瞬时数据队列:
 - **NonRecTDQFile** - 文件名
 - **NonRecTDQIndex** - 索引
 - **NonRecTDQMaxRecs** - 最大记录数
 - **NonRecTDQPrePages** - 预分配的页面数
 - **NonRecTDQVol** - 卷名
- 对于不可恢复临时存储器队列:
 - **NonRecTSQFile** - 文件名
 - **NonRecTSQIndex** - 索引
 - **NonRecTSQMaxRecs** - 最大记录数
 - **NonRecTSQPrePages** - 预分配的页面数
 - **NonRecTSQVol** - 卷名
- 对于物理可恢复瞬时数据队列:
 - **PhysicalTDQFile** - 文件名
 - **PhysicalTDQIndex** - 索引
 - **PhysicalTDQMaxRecs** - 最大记录数
 - **PhysicalTDQPrePages** - 预分配的页面数
 - **PhysicalTDQVol** - 卷名
- 对于可恢复辅助临时存储器队列:
 - **RecTSQFile** - 文件名
 - **RecTSQIndex** - 索引
 - **RecTSQMaxRecs** - 最大记录数
 - **RecTSQPrePages** - 预分配的页面数
 - **RecTSQVol** - 卷名

完整的一组 SFS 的区域定义 (RD) 在 *CICS 管理参考大全* 中的“文件服务器的区域定义 (RD)”描述。应该在开始使用 SFS 定义过程之前先查看该部分。

结构化文件服务器定义 (SSD)

SSD 用 **cicssfscreate** 命令、**cicscp** 命令或 SMIT 定义。结构化文件服务器定义 (SSD) 的作用是在启动时存储 SFS 必需的静态和动态信息, (如 *CICS 管理参考大全* 描述)。这包括控制和初始化信息。SSD 的名称 (关键字) 表示 SFS 名称。

当 **cicssfscreate** 命令定义 SSD 时, 应该考虑到的项目是:

1. SFS 使用的保护级别。SSD **ProtectionLevel** 属性用于设置该 SFS 收发“远程过程调用 (RPC)”的最小认证级别。可能的值为: **none**、**connect**、**call**、**pkt**、**pkt_integ**、**pkt_privacy**¹ 和 **default**。它们表示 DCE 递增的认证级别。
2. 逻辑卷和 SFS 用户标识符在使用 **cicssfscreate** 之前创建。这些名称在 SSD 中定义, 并且应该用 **cicssfscreate** 命令提供。若未提供名称, 则假设系统设定值。
3. 希望将数据写入到日志的频率在 SSD 中指定。这将影响可恢复性和性能。

完整的一组结构化文件服务器定义 (SSD) 在 *CICS 管理参考大全* 中的“结构化文件服务器定义(SSD)”描述。应该在开始使用 SFS 定义过程之前先查看该部分。

临时存储器定义 (TSD) 和瞬时数据定义 (TDD)

cicssfsconf 命令为队列管理建立 SFS。若应用程序使用辅助临时存储器队列或内分区瞬时数据队列, 则如第51页的『配置临时存储器定义 (TSD)』和第53页的『配置瞬时数据定义 (TDD)』中描述的定义它们。

文件定义 (FD) 和模式文件定义 (SCD)

FD (每个数据文件一个) 描述应用程序如何使用文件。尽管在配置过程中未使用 FD, 但是, 为了 CICS 能够使用文件, 它必须存在。

可以用 **cicssfsimport** 命令使用模式文件定义 (SCD), 以帮助您在 SFS 上建立文件。一个 SCD 包含有关文件类型、字段类型和长度、索引的信息。可以使用相同的 SCD 建立几个文件。

完整的一组模式文件定义 (SCD) 属性在 *CICS 管理参考大全* 中的“模式文件定义 (SCD)”描述。应该在开始使用 SFS 定义过程之前先查看该部分。另见第43页的『配置文件定义 (FD)』。

注: **cicssdt** 命令也可以用于在 SFS 上建立文件, 它不要求模式文件定义 (SCD)。参阅第48页的『使用 **cicssdt** 建立用户文件』。

SFS 的安全性考虑

在 **cics** 组中运行 SFS 潜在地允许用户访问 SFS 备份文件。需要避免这种情况, SFS 可以在受限程度更高的组内运行。

CICS 使用的 SFS 必须为何样

对于 CICS 使用, SFS 需求:

- 操作系统用户标识符。

1. 尽管 CICS 支持 **pkt_privacy** 的使用, 该安全级别的 DCE 软件可能在系统上没有。参考 DCE 产品信息。

- 若 SFS 在 DCE 单元中：
 - DCE 委托人、帐户和密钥表文件。
 - 所有使用 SFS 的区域在相同 DCE 单元中。
 - 存取控制列表（ACL）指定谁对 SFS 有存取权。
- 两个逻辑卷，一个针对 CICS 队列和文件，一个针对 CICS 队列和文件的日志信息。
- 目录结构由下列构成：


```
/var/cics_servers/SSD
/var/cics_servers/SCD
```

SSD 目录包含：

- 当 SFS 正在运行或正在更改其配置时，使用一个锁定文件锁定对 SSD 目录中 SFS 配置的访问。锁定文件的完整路径名是：


```
/var/cics_servers/SSD/cics/sfs/serverId/lock
```

CICS 在需要时创建本文件。

- 若 SFS 在 DCE 单元中，包含密钥表文件的目录包含了 SFS 的 DCE 委托人的加密口令。密钥表文件的完整路径名是：


```
/var/cics_servers/SSD/cics/sfs/serverId/key
```
- 一个信息文件包含服务器的注册信息。CICS 在需要时创建本文件。信息文件的完整路径名是：


```
/var/cics_servers/SSD/cics/sfs/serverId/msg
```

SFS 在需要时创建本文件。

- 两个重新启动文件包含 SFS 需要的热启动信息。重新启动文件的完整路径名是：


```
/var/cics_servers/SSD/cics/sfs/serverId/restart
/var/cics_servers/SSD/cics/sfs/serverId/restart.bak
```

SFS 在需要时创建这些文件。

当 CICS 设施用来配置 SFS 时，其中大部分已经为您建立好。

设置 FLT 环境变量

快速本地传送（FLT）提高了 SFS 文件操作的速度，如第186页的『用快速本地传送（FLT）来改进 SFS 的性能』中描述。SFS 使用两种 FLT：

- 基于管道传送是系统设定启用的值（当 SFS 和区域位于相同主机上时）。不需要操作就能启用基于管道 FLT。
- 共享内存传送，比基于管道传送快。该 FLT 不加密信息，也不执行认证。它要求 Encina 环境变量 ENCINA_FLT_SMS_SERVER_ENABLE 设置为一个值（0 除外）。其它环境变量用来定制共享内存传送是如何使用的。

需要禁用 FLT，使用下列 Encina 环境变量：

- 当 ENCINA_FLT_CLIENT_MAX_FDS 设置为 0 时，禁用区域的 FLT。
- 当 ENCINA_FLT_SERVER_MAX_FDS 设置为 0 时，禁用 SFS 的 FLT。

需要仅对于区域禁用 FLT，则在区域的环境文件中将 ENCINA_FLT_CLIENT_MAX_FDS 设置为 0。

区域和 SFS 必须在相同的主机上。

在 DB2 上设置队列和文件管理

本部分描述了如何在 DB2 上建立队列和文件管理。

用于建立队列和文件的 DB2 的方法

注：本章不讨论将数据文件添加到 DB2。参阅第41页的『建立文件定义（FD）』。

下列过程用于配置 CICS 队列和文件管理的 DB2 版本 2 数据库。

1. 第243页的『为队列和文件管理设置 DB2』用于验证数据库和实例是否与 CICS 需求相符。
 2. 第246页的『用 cicsdb2conf 配置资源和 DB2』用于设置资源定义中的必需属性，并且创建 DB2 中用于队列管理的表。若 **cicscp** 配置区域，则这些将自动地执行。
- 使用第21页的表 7 中的指示，建立队列和文件管理的 DB2。

DB2 文件管理的安全性考虑

在 CICS 中可定义资源，以便仅使特定的用户标识符对其有存取权。这在第218页的『授权用户访问 CICS 资源』中有所描述。

可以通过使用 CICS 管理参考大全 中“cicsdb2conf - 配置 DB2”描述的 **cicsdb2conf -g** 选项进一步限制对 DB2 数据库的存取权。

用于 DB2 队列和文件管理的资源定义

本部分描述的 DB2 配置过程设置下列资源定义属性：

- 区域定义（RD）
 - **FileSystemType** 设置为 **DB2**。
 - **DefaultFileServer** 设置为用于区域的队列和文件管理的数据库名。
 - **RDBMSInstance** 设置为用于区域的队列和文件管理的实例名。
- 产品定义（XAD）
 - **XAOpen** 设置为用于连接到数据库的字符串。
 - **SwitchLoadFile** 设置为指向包含 `xa_switch_t` 结构定义和 DB2 的 XA 支持子程序的目标文件。

注：当按照描述的使用时，过程使用一个系统设定的交换装入文件。若必需的话，可以使用自用交换装入文件，如第243页的『为队列和文件管理设置 DB2』中描述。

迁移队列和文件管理

第31页的『将队列和文件管理从 DB2 移动到 SFS』和第29页的『从 SFS 将队列和文件管理移动到 DB2』描述如何在一个现存的区域内更改 CICS 队列和文件管理。在使用这些过程前，请先查看下列信息：

1. 若在更改文件管理器后冷启动区域或区域为空，则不需要传送 CICS 队列文件。
cicssdt 和 **cicsddt** 可用于列示每个文件中的记录数。
2. 对于许多文件的大量迁移，请参阅 /usr/lpp/cics/utlis/cicsdb2m 中的 README 文件。
3. 尽管当 DB2 使用作文件管理器时所有 CICS 队列都完全可恢复，但正如在队列临时存储器定义（TSD）和瞬时数据定义（TDD）的输入中所定义的那样，可恢复属性不改变。这就确定了什么文件可用于存储此队列的数据。在 SFS 和 DB2 之间迁移时，重要的是定义为不可恢复的在从 DB2 向 SFS 移动时仍为不可恢复。
4. 将 DB2 作为文件管理器使用的 CICS 将所有超过 4005 字节的记录都作为可变记录处理。
5. 用于访问长度超过 4005 字节固定记录而写的移植 CICS 应用程序将查找到 CICS 行为内的下列更改：
 - 在 READ 时，返回长度字段的值是数据的真实长度，不是期待的固定长度。
 - 在 READ 时，必须指定 LENGTH 选项，否则返回 LENGERR。
6. 当 DB2 被用作文件管理器时，超过 4005 字节的文件必须由变量类型的最后一列（VARCHAR、LONGVAR、BLOB）创建。预先设定固定列的总长度不能超过 4005 字节。
7. 将 CICS 文件从 SFS 移植到 DB2 的过程受限于下列条件：
 - 必须重定义固定长度超过 4005 字节的 SFS 文件。
 - 固定长度超过 4005 个字节且有超过 4004 字节的关键字段的 SFS 不能直接移植。必须重新设计文件，以保证所有关键字段在记录的前 4005 字节内。
8. DB2 上的文件锁定和并行与 SFS 上的文件锁定和并行不同。请参阅 *CICS 应用程序设计指南* 中的“应用程序中的 CICS 锁定功能”。

从 SFS 将队列和文件管理移动到 DB2

本过程的目的在于从区域的 SFS 将 CICS 队列和文件管理移动到 DB2，该区域已被配置为使用 SFS。

注:

1. 本过程创建用来定义区域的 DB2 数据库的产品定义（XAD）。若已经有为该数据库定义的 XAD，则使用 **cicsdelete** 命令或当使用 **cicsdb2conf** 命令时使用 **-s** 选项。否则，该数据库可能有两个 XAD 项，产生不可预测的结果。
2. DB2 不允许在字段名中有特殊字符。在将 SFS 文件移植到 DB2 之前，参考 DB2 SQL 参考大全中描述的字段名限制信息。

本过程使用 **cicsdb2conf** 配置区域和文件管理的 DB2 数据库。但是，**cicscp** 有其它可能重要的选项。例如，若需要使用 DB2 显示认证，则使用 **-u** 选项指定用户。若需要指定单阶段落实产品定义（XAD）项的创建，则使用 **-n** 选项，若不需要 **cicscp** 创建该数据库的产品定义（XAD）项（当已经创建了一个时），则需要使用 **-s** 选项。

请阅读 *CICS 管理参考大全* 的“cicscp - 区域命令”，以获得更多信息。

需要使用本过程:

- 必须作为 CICS 服务器主机上的超级用户注册，并且超级用户必须如第3页的『建立 CICS 系统管理员』中所描述的那样建立起来。
- 验证数据库已经启动。

- 若正在使用显示 DB2 认证将 CICS 连接到 DB2:
 - 必须建立用户，并授予最小许可权。（参阅快速入门中的“设置 CICS 所使用的 DB2”。）
 - 若不希望提示口令，则将 CICSDB2PASSWD 环境变量设置为数据库用户标识符口令。
- 1. 使用下列命令关闭区域:
`cicscp -v stop region regionName`
- 2. 使用下列命令重新配置 DB2 的区域:
`cicsdb2conf -r regionName -C -i instanceName -a databaseName`
- 3. 使用下列步骤将文件从 SFS 传送到 DB2:
 - a. 标识区域必需的文件。就是：
 - 所有在区域的文件定义（FD）中定义的文件。用 **FD BaseName** 属性指定文件名。
 - 在区域定义（RD）中定义的 CICS 队列。参阅第30页的表 8。

表 8. 队列名称值（其中 %R 为区域名）

RD 属性	SFS 名称（系统设定值）	DB2 名称（必要的）
LogicalTDQFile	%Rcicstdqlgfile	%Rlogtdq#
PhysicalTDQFile	%Rcicstdqphfile	%Rphtdq#
NonRecTDQFile	%Rcicstdqnofile	%Rnrectdq#
RecTSQFile	%Rcicsrectsqfile	%Rrectsq#
NonRecTSQFile	%Rcicsnrectsqfile	%Rnrectsq#
LocalQProtectFile	%Rcicsplqfile	%Rlqprot#
LocalQFile	%Rcicsnlqfile	%Rlque#

- b. 使用带 **stof** 选项的 **cicssdt**，将文件从 SFS 传送到平面文件格式。例如：
`cicssdt -s serverName -c stof BaseName`

Specify **N** in the Dump file Asis? field.
`[Flat Filename: BaseName.sdt`
`[Dump file Asis ? Y/[N]: N`
`[Dump Record(s): ALL`
- c. 在 DB2 中创建用户定义的文件。
- d. 通过使用带 **ftod** 选项的 **cicsddt**，将文件从平面文件格式传入 DB2。例如：
`cicsddt -s databaseName -c ftod tableName`

使用在 **cicssdt** 步骤中创建的相同平面文件名。需要在整个字段中叠打。例如：
`[Flat Filename: BaseName.sdt`

需要更多信息，请参阅CICS 管理参考大全中的“cicssdt - SFS 诊断工具”和“cicsddt - DB2 诊断工具”。

注意需要导入的队列文件在第30页的表 8 中显示的 DB2 名称下。例如，需要导入 regionA 的可恢复瞬时数据队列文件，使用下列命令（假设 SFS 文件正在使用系统设定名称）：

```
cicssdt -s ./:/cics/sfs/hostABCD -c stof regionAcicsrectsqfile.sdt
```


它产生了一个名为 `regionAcicsrectsqfile.sdt` 的平面文件，当使用下列命令时：

```
cicsddt -s cicstest -c ftod regionArectsq#
```

提示您平面文件名。输入 `regionAcicsrectsqfile.sdt`

4. 自动启动区域。

将队列和文件管理从 DB2 移动到 SFS

本过程的目的在于从区域的 DB2 将 CICS 队列和文件管理移动到 SFS，该区域配置已被为使用 DB2。

在使用过程之前，请阅读第28页的『迁移队列和文件管理』，以便完全了解该更改的蕴含式。也查看下列：

- 当 DB2 用于队列和文件管理时，恢复属性将被忽略。验证是否正确设置了下列属性：
 - 瞬时数据定义（TDD） **RecoveryType** 属性
 - 临时存储器定义（TSD） **RecoverFlag** 属性
 - 文件定义（FD） **RecoverStatus** 属性

不然，无法删除并重定义队列；可以更改文件定义（FD）。

1. 使用正常关机关闭区域。例如：

```
cicscp -v -l /tmp/cicscp.log stop region regionA
```

2. 若 SFS 不存在，则区域无法使用，按照第34页的『使用 `cicscp start sfs_server` 建立 SFS』建立起一个。

3. 使用下列步骤将文件从 DB2 传送到 SFS：

a. 标识区域必需的文件。就是：

- 所有在区域的文件定义（FD）中定义的文件。用 FD **BaseName** 属性指定文件名。
- 在区域定义（RD）中定义的 CICS 队列。参阅第30页的表 8。

b. 使用带 **dtod** 选项的 **cicsddt**，将文件从 DB2 传送到平面文件格式。参阅 *CICS 管理参考大全* 中的“`cicsddt - DB2 诊断工具`”。

c. 在 SFS 中创建用户定义的文件。

d. 使用 **cicssfscnf** 命令配置 SFS。例如，若区域名是 `regionA`，SFS 是 `././cics/sfs/hostABCD`，则使用下列命令：

```
cicssfscnf -R wc regionA DefaultFileServer=././cics/sfs/hostABCD
```

-R 选项将文件系统重复位为区域定义（RD）和文件定义（FD）中的 SFS，并重新命名队列。若队列文件在以前步骤中传送过，则使用 **-I** 选项。

注意：

这将删除任何系统定义的该区域的产品定义（**XAD**）。

e. 通过使用带 **ftos** 选项的 **cicsddt**，将文件从平面文件格式传送入 SFS。参阅 *CICS 管理参考大全* 中的“`cicsddt - SFS 诊断工具`”。

注意，需要导入的队列文件在第30页的表 8 中显示的 SFS 名称下。例如，需要导入 `regionA` 的可恢复 TDQ 文件，使用下列命令（假设 SFS 文件正在使用系统设定名称）：

```
cicsddt -s cicstest -c dtof regionArechtsq#.ddt
```

它产生了一个名为 regionArechtsq#.ddt 的平面文件，当使用下列命令时：

```
cicssdt -s ../cics/sfs/hostABCD -c ftos regionAcicsrechtsqfile
```

提示您平面文件名。输入 regionArechtsq#。

4. 自动启动区域。

关于使用 **cicscp** 建立区域

cicscp 拥有的选项比本书中显示的更多，例如导入一组自用区域定义（RD），而不是使用系统设定定义，并指定为文件定义（FD）使用的组 ID。若正在使用文件管理的 DB2，可以指定 DB2 显示认证或单阶段落实。

需要更多有关 **cicscp** 其它可用选项的信息，请参阅 *CICS 管理参考大全* 中的“cicscp - 区域命令”。

cicscp 系统设定值

cicscp 使用下列系统设定值：

- 为区域创建的用户标识符是 **cics**。
- 区域定义（RD）**DefaultUserId** 属性的系统设定值是 CICSUSER。
- 在一个 DCE 单元中，为区域创建的 DCE 委托人是 **cics/regionName**。
- SFS 在主机名后命名。例如，若主机名为 hostABCD，则命名 SFS：
../cics/sfs/hostABCD
- 创建 SFS 的短名，短名由字母『S』和紧跟其后的主机名的前七个字符构成。例如，若主机名是 hostABCD，则短名是 **ShostABC**。
- 系统设定 SFS 用户标识符和 DCE 委托人使用短名命名。
- SFS 逻辑卷名称取决于短名。例如，若短名是 ShostABC，则将队列和文件创建的逻辑卷命名为：

```
sfs_ShostABC
```

并且将用于队列和文件日志的逻辑卷命名为：

```
log_ShostABC
```

注：资源定义命令和 SMIT 使用 **log_SFS_SERV** 和 **sfs_SFS_SERV** 作为 SFS 逻辑卷的系统设定名称。

- 每个逻辑卷的系统设置大小是 64 兆字节（MB）。若不需要 **cicscp** 创建这种大小的逻辑卷，则在用 **cicscp** 命令之前简单地创建逻辑卷。这在第236页的『准备 SFS 的物理存储设备』中描述。

DCE 先决条件

cicscp 配置 CICS 取决于 DCE 的配置。例如，若配置 DCE 单元，则将区域、文件管理器和客户程序配置为在 DCE 单元中工作。

cicscp 使用的基本命令

cicscp 调用基本命令创建并启动区域和文件管理器。

1. **cicsdefault** 或 **cicsimport**，取决于是否导入区域定义。
2. 创建、配置并启动 SFS，使用：
 - a. **cicssfscree**
 - b. **cicssfs**
 - c. **cicssfsconf**
3. 配置文件管理的 DB2 数据库，使用 **cicsdb2conf**。（在使用 **cicsdb2conf** 或 **cicscp** 之前，一个 DB2 数据库必须创建并启动。）
4. 启动区域，使用 **startsrc**。

相关信息

第61页的『建立 Micro Focus COBOL 运行时间环境』。

第90页的『SQL 数据库设置过程』。

建立 CICS 日志的单独文件系统

CICS 日志（/var/cics_regions/*regionName*/log）保留 CICS 区域的动态运行记录数据。可使用 CICS 日志的单独文件系统，但是配置了区域之后才能建立它。

注：如同 CICS 日志中包含的信息需要热启动区域一样，建议镜像日志，以便在文件系统数据丢失发生时将系统恢复到『故障点』。需要有关 CICS 日志和恢复的更多信息，请参阅第149页的『第10章 CICS 备份和恢复介绍』。

区域和文件管理设置过程

本部分包含建立区域和文件管理器的过程。按照第21页的表 7 的指导使用这些过程。

更改新区域的区域定义（RD）属性

本过程的目的在于对区域的区域定义（RD）进行永久性的更改，该区域必须刚建立但尚未投入产品。本过程需要区域冷启动。

在使用本过程之前，请查看 *CICS 管理参考大全* 中的“区域定义 (RD)”和“文件服务器的区域定义 (RD)”的信息。请勿更改任何 RD 文件服务器属性，除非需要重新配置文件管理器。

1. 注册到一个具有超级用户（root）特权的用户标识符，该用户标识符如第3页的『建立 CICS 系统管理员』中描述的那样建立。
2. 若正在 DCE 单元中工作，则作为管理员（dce_login cell_admin）注册到DCE。
3. 使用下列命令关闭区域：

```
cicscp -v stop region regionName
```

4. 使用 **cicsupdate** 命令更改 RD 属性。例如:

```
cicsupdate -c rd -r regionA ClassMaxTasks=2,2,2,2,2,2,2,2 \
ClassMaxTaskLim=1,2,0,3,5,2,2,2,2
```

5. 冷启动区域。使用下列命令:

```
cicscp -v start region regionName StartType=cold
```

相关信息

CICS 管理参考大全 中的“区域定义 (RD)”。

设置区域的语言

使用本过程指定希望区域用于发出信息的语言。

使用该过程:

- 必须配置区域。
- 验证安装了适当的信息目录。例如, 若一个客户程序要求中文, 另一个客户程序要求日语, 而区域要求英语, 则验证安装了中文、日语、英语。检查 `/usr/lpp/cics/msg` 目录。

注: 设置 **cicsterm** 的语言, 如第73页的『设置客户的语言』中所描述的。用 **cicsteld -l** 选项设置 Telnet 客户程序的语言, 如 *CICS 管理参考大全* 中的“**cicsteld -** 将 telnet 客户程序连接至区域”描述。

需要指定区域的语言:

1. 将下列添加到 `/etc/environment` 中的 NLSPATH:
`/usr/lpp/cics/msg/%L/%N:/usr/lpp/cics/msg/C/%N`
2. 设置 `/var/cics_regions/regionName/environment` 中的 LANG 环境变量为区域需要的语言。

使用 **cicscp start sfs_server** 建立 SFS

本过程的目的在于:

- 在 DCE 单元中, 创建 SFS 的 DCE 委托人和帐户、ACL 和密钥表文件。
- 在仅 RPC 设置中, 创建 SFS 的联接字符串。

注: 若不希望 **cicscp** 创建联接字符串, 则在使用本过程之前, 先按照第238页的『设置 SFS 的联接文件』执行。这将给出端口和协议的选择。若服务器的联接字符串已经存在, 则 **cicscp** 跳过该步骤。

- 创建 SFS 用户标识符。
 - 创建 SFS 逻辑卷。
 - 创建结构化文件服务器定义 (SSD)。
 - 启动 SFS。
1. 在 SFS 名中选择一个使用的基名。例如, 若基名为 **hostABCD**, 则 SFS 名为 **./cics/sfs/hostABCD**。系统设定基名为主机名。

2. 注册一个具有超级用户（root）特权的用户标识符，该用户标识符如第3页的『建立 CICS 系统管理员』中描述的那样建立。
3. 若正在 DCE 单元中工作，则作为管理员（dce_login cell_admin）注册到 DCE。
4. 若正在仅 RPC 设置中工作，则将 ENCINA_BINDING_FILE 环境变量设置为 `/var/cics_servers/server_bindings`。
5. 若不希望用 **cicscp** 系统设定值建立 SFS 逻辑卷，则设置下列环境变量：

CICS_SFS_SIZE

为 SFS 指定数据和日志逻辑卷的大小（兆字节（MB））。**cicscp** 系统设定值为每个 **64**。

CICS_SFS_DATA_SIZE

为 SFS 指定数据逻辑卷的大小（兆字节（MB））。这将覆盖环境变量 CICS_SFS_SIZE。**cicscp** 系统设定值为 **64**。

CICS_SFS_LOG_SIZE

为 SFS 指定日志逻辑卷的大小（兆字节（MB））。这将覆盖环境变量 CICS_SFS_SIZE。**cicscp** 系统设定值为 **64**。

CICS_SFS_VG

为 SFS 指定数据和日志逻辑卷的逻辑卷组。**cicscp** 的系统设定值为 **rootvg**。

CICS_SFS_DATA_VG

为 SFS 指定数据逻辑卷的逻辑卷组。这将覆盖环境变量 CICS_SFS_VG。**cicscp** 的系统设定值为 **rootvg**。

CICS_SFS_LOG_VG

为 SFS 指定日志逻辑卷的逻辑卷组。这将覆盖环境变量 CICS_SFS_VG。**cicscp** 的系统设定值为 **rootvg**。

6. 使用下列命令：

```
cicscp -v -l /tmp/cicscp.log start sfs_server /./cics/sfs/serverId
```

使用 **cicscp** 删除区域

通过使用 **cicscp** 按照本过程删除区域。本过程不删除区域的操作系统用户标识符。

1. 注册一个具有超级用户（root）特权的用户标识符，该用户标识符如第3页的『建立 CICS 系统管理员』中描述的那样建立。
2. 若正在 DCE 单元中工作，则作为管理员（dce_login cell_admin）注册到 DCE。
3. 使用下列命令：

```
cicscp destroy region regionName
```

本命令：

- 停止区域（若正在运行）。
- 从 CICS 服务器数据库中删除区域定义。
- 删除操作系统机制中用于启动和停止长型运行处理的项。
- 删除区域的 DCE 委托人、其 DCE 帐户和其 DCE 密钥表项（若存在）。
- 删除区域目录：

```
/var/cics_regions/regionName
```

4. 若正在 DCE 单元中工作，则通过再次注册入 DCE 或使用 DCE 命令 **kinit** 重新认证任何访问区域的 **dce_login** 上下文的 DCE 存取权限。这将删除旧区域任何高速缓存的存取权限，并避免产生问题（若使用与刚消除的那个区域的相同名称创建新区域）。

参阅 *CICS 管理参考大全* 中的“cicscp - 区域命令”，以获得更多有关使用该命令的信息。

使用 cicscp 删除 SFS

通过使用 **cicscp**，使用本过程删除 SFS。本过程不删除 SFS 的操作系统用户标识符。

1. 注册一个有超级用户（root）特权的用户标识符，该用户标识符如第3页的『建立 CICS 系统管理员』中描述的那样建立。
2. 若正在 DCE 单元中工作，则作为管理员（**dce_login cell_admin**）注册到 DCE。
3. 使用下列命令：

```
cicscp destroy sfs_server ./cics/sfs/serverId
```

本命令：

- 停止 SFS（若正在运行）
- 从 CICS 服务器数据库中删除 SFS 定义
- 删除操作系统机制中用于启动和停止长型运行处理的项
- 删除 SFS 的 DCE 委托人、其 DCE 帐户和其 DCE 密钥表项（若存在）
- 删除 SFS 目录：

```
/var/cics_servers/SSD/cics/sfs/serverId
```

4. 若正在 DCE 单元中工作，则通过再次注册入 DCE 或使用 DCE 命令 **kinit** 重新认证任何访问 SFS 的 **dce_login** 上下文的 DCE 存取权限。这将删除旧服务器任何高速缓存的存取权限，并避免产生问题（若使用与刚消除的那个 SFS 的相同名称创建新 SFS）。

第3章 配置 CICS 资源

表 9. 为区域准备运行 CICS 应用程序的导向图

若需...	参考...
为区域运行事务作准备。	第37页的『建立事务处理定义（TD）』。
为区域运行程序并使用映像集作准备。	第39页的『建立程序定义（PD）』。
为区域使用数据文件作准备。	第41页的『建立文件定义（FD）』。
为区域使用临时存储器队列作准备。	第50页的『建立临时存储器队列（TSQ）』。
为区域使用瞬时数据队列作准备。	第52页的『建立瞬时数据队列（TDQ）』。
为区域使用日志作准备。	第54页的『建立日志定义（JD）』。
为区域使用终端和打印机作准备。	第56页的『建立终端定义（WD）』。
为区域执行监控程序作准备。	第60页的『建立监控定义（MD）』。
建立 \Micro Focus COBOL 运行时间。	第61页的『建立 Micro Focus COBOL 运行时间环境』。
查找区域是如何使用应用程序服务器的。	第61页的『建立应用程序服务器』。
为客户程序和用户准备区域。	第65页的『第4章 配置客户程序』。
为区域使用关系数据库作准备。	第62页的『建立产品定义（XAD）』和第87页的『第5章 为使用关系数据库配置 CICS』。

建立事务处理定义（TD）

CICS 中的事务是一个调用 CICS 应用程序的部件或处理。事务由单个请求启动，这一请求通常来自终端，并使用一个 4 字符的事务标识符。

事务处理定义（TD）定义所有 CICS 能够在区域内处理的事务。TD 中每个项目包含 CICS 用来标识并运行事务的控制信息。

配置事务处理定义（TD）

需要使用永久数据库中的事务处理定义（TD）：

1. 若正在 DCE 单元中工作，则作为管理员（dce_login cell_admin）注册到 DCE。当且仅当需要使用 --B 选项并且区域正在运行时才需要它。
2. 使用下列命令例子：

添加新 TD

本例添加一个在『Sales』组中的 TD 项。为其它所有 TD 属性假设系统设定值：

```
cicsadd -c td -r regionA -P TRN1 \
      Group='Sales'
```

当这样定义时，若区域的 **Groups** 属性列表中包括『Sales』（参阅第23页的『定义可选择安装的资源』），则安装该事务的项。若不包括『Sales』，则使用 **cicsinstall** 命令，第145页的『安装一组定义至运行时中』中有所描述。

若区域正在运行，并且希望将定义添加到运行时间数据库和永久数据库，则使用 **-B** 选项，如下显示：

```
cicsadd -c td -r regionA -B TRN1 \
      Group='Sales'
```

需要更多有关添加定义的信息，请参阅 *CICS 管理参考大全* 中的“cicsadd - 添加 CICS 资源定义”。

更改现存的 TD

本例显示如何更改 TD 项，以致区域下一次冷启动时不将其装入运行时间。

```
cicsupdate -c td -r regionA -P TRN1 \
      ActivateOnStartup=no \
      Group=''
```

需要将定义安装入运行时间，参阅第143页的『从永久数据库安装一个资源定义』。

注：若 **Group** 属性包含一个不包括在区域 **Groups** 列表中列出的组名，则无论 **ActivateOnStartup** 值是什么，都将安装 TD 项。

需要更多有关更改定义的信息，参阅 *CICS 管理参考大全* 中的“cicsupdate - 修改 CICS 资源定义”。

删除 TD 项

本例显示如何从永久数据库中删除 TD 项目。

```
cicsdelete -c td -r regionA -P TRN1
```

需要从运行时间删除项，参阅第144页的『仅从运行时删除资源定义』。

需要更多有关删除定义的信息，参阅 *CICS 管理参考大全* 中的“cicsdelete - 删除 CICS 资源定义”。

查看 TD 项

本例显示如何查看 TD 项：

```
cicsget -c td -r regionA TRN1
```

需要更多有关查看定义的信息，参阅 *CICS 管理参考大全* 中的“cicsget - 获得 CICS 资源定义信息”。

需要使用 **SMIT** 为区域定义资源，参阅第10页的『使用 **SMIT** 建立 CICS 系统』。

需要一套完整的用于事务处理的所有属性的说明，请参阅 *CICS 管理参考大全* 中的“事务定义 (TD)”。

配置不确定的条件

内部通信经常使用两阶段落实。两阶段落实是一个协议，当一个事务使用多个资源管理器时，用于协调对可恢复资源的更改。在第一阶段中，要求资源管理器准备工作；在第二阶段中，它们都或被要求落实，或被要求逆序恢复（重算）。

在处理两阶段落实事务时，可能会发生连接或系统错误。某类错误将使事务一直等待，直到由您或 CICS 本身改正了错误为止。这些等待事务被称作处于不确定状态。

由于您可能不希望事务在改正错误之前一直等待，请按照下列执行：

1. 使用事务处理定义 (TD) **InDoubt** 属性指定如何处理条件。当事务处于两阶段落实之间阶段的等待之中，并且对事务发出 CEMT SET TASK FORCEPURGE 时，CICS 使用该属性确定是确认还是逆序恢复任何在等待发生以前事务所作的更改。 **InDoubt** 属性可以设置为：
 - **wait_commit** (确认更改)
 - **wait_backout** (逆序恢复更改)
2. 使用 CEMT INQUIRE TASK INDOUBT 确定事务是否处于不确定状态。
3. 使用 CEMT SET TASK FORCEPURGE 并根据 TD **InDoubt** 属性设置，分辨不确定情况。

相关信息

转至 *CICS 管理参考大全* 中的 CEMT 运行时间资源管理，以使用 CEMT:

- 查看运行时间事务处理定义 (TD)。
- 设置运行时间事务处理的优先级。
- 启用或禁用运行时间事务处理。

需要查找如何在运行时间中写 CICS 应用程序管理状态或事务操作，请转至 *CICS 应用程序设计参考大全* 中的“INQUIRE 和 SET 命令”。

CICS 管理参考大全 中的“CEMT INQ/SET TRANSACTION”和“事务定义 (TD)”。

建立程序定义 (PD)

有几类程序在 CICS 中使用。*CICS 应用程序设计指南* 中的“CICS 应用程序设计接口 (API)”描述了使用 CICS API 的应用程序。第105页的『第6章 使用用户出口和用户程序定制』描述了从 CICS 中调用的用户写入程序。*CICS 应用程序设计指南* 中的“基本映射支持 (BMS)服务”也描述了用于呈现服务的程序。所有这些程序都需要程序定义 (PD)。

程序定义 (PD) 定义所有 CICS 能够在区域内处理的程序。PD 中的每个项目包含 CICS 用来标识并运行程序的控制信息。

配置程序定义 (PD)

需要使用永久数据库中的程序定义 (PD)：

1. 若正在 DCE 单元中工作，则作为管理员 (dce_login cell_admin) 注册到 DCE。当且仅当正在使用 **-B** 选项，并且 CICS 区域正在用 DCE 保护级别运行时才需要它。
2. 使用下列命令例子：

需要添加新 PD

本例添加一个命名为『sales1』的程序 PD 项。该程序在 /var/sales/bin 内。为其它所有 PD 属性假设系统设定值：

```
cicsadd -c pd -r regionA -P sales1 \  
      PathName='/var/sales/bin/sales1'
```

若区域正在运行，并且希望将定义添加到运行时间数据库和永久数据库，则使用 **-B** 选项，如下显示：

```
cicsadd -c pd -r regionA -B sales1 \  
      PathName='/var/sales/bin/sales1'
```

需要更多有关添加定义的信息，请参阅 *CICS 管理参考大全* 中的“cicsadd - 添加 CICS 资源定义”。

更改现存的 PD

本例显示如何更改 PD 项状态：

```
cicsupdate -c pd -r regionA -P sales1 \  
      EnableStatus=disabled
```

需要更多有关更改定义的信息，参阅 *CICS 管理参考大全* 中的“cicsupdate - 修改 CICS 资源定义”。

删除 PD 项

本例显示如何从永久数据库中删除 PD 项目。

```
cicsdelete -c pd -r regionA -P sales1
```

需要从运行时间删除项，参阅第144页的『仅从运行时删除资源定义』。

需要更多有关删除定义的信息，参阅 *CICS 管理参考大全* 中的“cicsdelete - 删除 CICS 资源定义”。

查看 PD 项

本例显示如何查看 PD 项：

```
cicsget -c pd -r regionA sales1
```

需要更多有关查看定义的信息，参阅 *CICS 管理参考大全* 中的“cicsget - 获得 CICS 资源定义信息”。

需要使用 SMIT 为区域定义资源，参阅第10页的『使用 SMIT 建立 CICS 系统』。

需要一套完整的用于程序的所有属性的说明，请参阅 *CICS 管理参考大全* 中的“程序定义 (PD)”。

建立不同语言的基本映像支持 (BMS)

若希望不同语言显示出可用的相同映射和映像集，可通过取出映射源并将其翻译成新语言来完成。然后，用 **cicsmap** 命令处理映射表，保证物理映射表置于下列目录下的适当目录中：

```
/var/cics_regions/regionName/maps
```

例如，若原始映像集在 en_US 中，使用 **cicsmap** 将物理映像集置于 en_US 目录中（参考 *CICS 应用程序设计指南* 的“cicsmap - 处理源文件”中 **-f locale** 选项的描述。需要映射表的法语版，再次使用 **cicsmap**，验证在处理中使用正确场所。确保物理映射表置于该场所的目录中。若不存在一个目录，则可能需要创建新目录。

在映射表的程序定义 (PD) 中，需要仅将映像集的 **PathName** 设置为映像集名，以便 CICS 搜索适当的目录，而不是指定一个显示目录路径。

一旦完成，根据 **cicsterm** 或 Telnet 客户程序的 LANG 环境变量，将显示适当的映射表。

注意虽然物理映射表可能不同，但是，逻辑映射表应该仍是相同的。

相关信息

转至 *CICS 管理参考大全* 中的 CEMT 运行时间资源管理，以使用 CEMT:

- 查询运行时间程序定义 (PD)。
- 用特定源语言查询运行时间程序。
- 启用或禁用运行时间程序或映像集。
- 强制 CICS 使用程序或映像集的最新版本。

CICS 管理参考大全 中的 “CEMT INQ/SET PROGRAM” 和 “程序定义 (PD)”。

CICS 内部通信指南 中的 “配置 DPL 的程序定义(PD)” 和 “分布式程序链路 (DPL)”。

CICS 应用程序设计指南 中的 “基本映射支持 (BMS)服务”。

建立文件定义 (FD)

CICS 为类属型面向记录的文件管理器提供了一个接口。如果正在从 IBM 基于大型机的 CICS 迁移到 IBM CICS AIX 版，那么您可能会熟悉虚拟存储存取法 (VSAM) 数据集。VSAM 是一种访问方法，对直接存取设备上的定长记录或变长记录进行直接处理或顺序处理。VSAM 数据集或文件可以通过关键字段 (键序列) 以逻辑顺序、以他们被写在数据集或文件 (输入顺序) 的物理顺序或相关记录数组织起来。

您的操作系统不支持 VSAM。IBM CICS AIX 版通过使用 SFS 或 DB2 文件，以仿真 VSAM。

与数据文件工作时使用的过程

第42页的『关于文件定义 (FD)』和第43页的『配置文件定义 (FD)』，描述如何更改、删除或查看文件的 FD。

注意:

在更改现存 **FD** 前，确认文件已关闭。

还需要将文件添加到文件管理器。应该选做下列之一:

- 需要建立能帮助您将文件导入文件管理器的文件概要，按照第44页的『配置文件的模式文件定义 (SCD)』。
- 若已使用文件概要，需要将文件添加到文件管理器，按照第45页的『使用 cicsdb2import 建立 DB2 上的数据文件』或第46页的『使用 cicssfimport 建立 SFS 上的数据文件』。
- 若已使用文件概要，需要将文件添加到文件管理器，按照第47页的『使用 cicsddt 建立 DB2 上的数据文件』或第48页的『使用 cicssdt 建立用户文件』。

若是远程文件，还可参阅 *CICS 内部通信指南* 中的 “定义远程文件”。

关于文件定义 (FD)

用于文件的各种属性描述在 *CICS 管理参考大全* 中的“文件定义 (FD)”。将 CICS FD 与文件管理联结在一起的特定属性是:

FileServer

文件定位的 SFS 名称。若使用 DB2 作为文件管理器（因为区域仅能为文件管理使用一个 DB2 数据库），则本属性将被忽略。

BaseName

文件名。

IndexName

索引名。

需要为用于访问文件的每个索引创建单独 FD。

关键字顺序数据集 (KSDS) 文件范例

假设已创建了 KSDS 文件名 **personnel**，其主索引是 **number**，其辅助索引是 **name**。您将创建两个 FD 项；其中一个允许用主索引对文件的访问，另一个允许对辅助索引的访问。定义将包含:

```
FD key      : PERSNUMB
BaseName    : personnel
IndexName   : number
```

```
FD key      : PERSNAME
BaseName    : personnel
IndexName   : name
```

输入顺序数据集 (ESDS) 文件范例

可使用与 KSDS 文件所使用的相似方法创建 ESDS 文件的定义。假设已经创建了 ESDS 文件 **history**，主索引为 **sequence**，辅助索引为 **owner**。可以创建文件定义允许对这两个索引进行访问:

```
FD key      : HISTSEQ
BaseName    : history
IndexName   : sequence
```

```
FD key      : HISTOWN
BaseName    : history
IndexName   : owner
```

注: 当在 SFS 上创建 ESDS 文件时，应该使用 SFS 设施对文件中的记录数加以限制，因为 CICS 无法访问项目序列号码不能压缩成 32 位记录字节地址的记录。

相对记录数据集 (RRDS) 文件范例

CICS 仅允许用固定长度记录对相对文件访问。当定义这种文件时，应该切记记录包括一个相对记录号的字段，但是该字段对于使用 CICS 文件控制接口的应用程序不是直接可见的。

文件管理器（SFS 或 DB2）允许相对记录文件有辅助索引。虽然这些辅助索引的存在并不防止文件的使用，但是 CICS 却无法使用它们。

假设您已创建了一个名为 **level** 的 RRDS 文件，其主索引为 **level_number**，辅助索引为 **description**。可以创建一个文件定义，允许对主、相对记录号和索引进行访问，但是，CICS 不允许使用辅助索引。主文件定义可为:

```
FD key      : LEVEL
BaseName    : level
IndexName   : level_number
```

另外，可以创建引用任何这些文件的其它文件定义（FD），这些文件可能对一些其它属性有不同的值。例如，可有不同 RSL 关键字的定义，它对删除访问状态有不同值。换言之，您可能还有两个文件定义引用相同的文件名和索引名，但是却对于可恢复性状态有不同的值。

您可使用第43页的『配置文件定义（FD）』中所描述的过程，以在区域正在运行时执行所有上述区域配置步骤。

注：SFS 和 DB2 允许一个以上的客户程序同步执行相同文件上的操作。因此，若区域使用该设施开发的适合应用程序，则可以将相同文件的 FD 项添加到一个以上区域。

配置文件定义（FD）

需要使用永久数据库中的文件定义（FD）：

1. 若正在 DCE 单元中工作，则作为管理员（dce_login cell_admin）注册到 DCE。当且仅当正在使用 **-B** 选项，并且 CICS 区域正在用 DCE 保护级别运行时才需要它。
2. 使用下列命令例子：

需要添加新 FD

本例添加一个 FD 项，CICS 已知该文件为 file1，RSL 关键字为 1，文件名为『personnel』，索引名为『number』。为其它所有 FD 属性假设系统设定值：

```
cicsadd -c fd -r regionA -P file1 RSLKey=1 \
      BaseName=personnel \
      IndexName=number
```

若区域正在运行，并且希望将定义添加到运行时间数据库和永久数据库，则使用 **-B** 选项，如下显示：

```
cicsadd -c fd -r regionA -B file1 RSLKey=1 \
      BaseName=personnel \
      IndexName=number
```

需要更多有关添加定义的信息，请参阅 *CICS 管理参考大全* 中的“cicsadd - 添加 CICS 资源定义”。

更改现存的 FD

本例显示如何更改 FD 项：

```
cicsupdate -c fd -r regionA -P file1 \
      OpenStatus=open
```

需要更多有关更改定义的信息，参阅 *CICS 管理参考大全* 中的“cicsupdate - 修改 CICS 资源定义”。

删除 FD 项

本例显示如何从永久数据库中删除 FD 项目。

```
cicsdelete -c fd -r regionA -P file1
```

需要从运行时间删除项，参阅第144页的『仅从运行时删除资源定义』。

需要更多有关删除定义的信息，参阅 *CICS 管理参考大全* 中的“cicsdelete - 删除 CICS 资源定义”。

查看 FD 项

本例显示如何查看 FD 项：

```
cicsget -c fd -r regionA file1
```

需要更多有关查看定义的信息，参阅 *CICS 管理参考大全* 中的“cicsget - 获得 CICS 资源定义信息”。

需要使用 SMIT 为区域定义资源，参阅第10页的『使用 SMIT 建立 CICS 系统』。

需要一套完整的用于文件所有属性的说明，请参阅 *CICS 管理参考大全* 中的“文件定义 (FD)”。

配置文件的模式文件定义 (SCD)

定义文件的模式文件定义 (SCD)：

1. 若正在 DCE 单元中工作，则作为管理员 (dce_login cell_admin) 注册到 DCE。
2. 使用下列命令例子：

需要添加新 SCD

本例添加一个 SCD 项。数据库选项 (-P) 和不这样使用的区域名与 SCD 项无关。

```
cicsadd -c scd -f file.schema \  
  FileType=sequential \  
  FieldName1=happy \  
  FieldType1=unsignedInt16 \  
  FieldName2=sneezy \  
  FieldType2=unsignedInt16 \  
  FieldName3=grumpy \  
  FieldType3=unsignedInt16 \  
  FieldName4=dopey \  
  FieldType4=unsignedInt16
```

需要更多有关添加定义的信息，请参阅 *CICS 管理参考大全* 中的“cicsadd - 添加 CICS 资源定义”。

更改现存的 SCD

本例显示如何更改 SCD 项内的一个属性：

```
cicsadd -c scd -f file.schema \  
  FileType=clustered
```

需要更多有关更改定义的信息，参阅 *CICS 管理参考大全* 中的“cicsupdate - 修改 CICS 资源定义”。

删除 SCD 项

本例显示如何从永久数据中删除 SCD 项。

```
cicsdelete -c scd -f file.schema
```

需要更多有关删除定义的信息，参阅 *CICS 管理参考大全* 中的“cicsdelete - 删除 CICS 资源定义”。

查看 SCD 项

本例显示如何查看 SCD 项：

```
cicsget -c scd -f file.schema
```

需要更多有关查看定义的信息，参阅 *CICS 管理参考大全* 中的“cicsget - 获得 CICS 资源定义信息”。

需要使用 SMIT 为区域定义资源，参阅第10页的『使用 SMIT 建立 CICS 系统』。

需要一套完整的用于文件概要所有属性的说明，请参阅 *CICS 管理参考大全* 中的“模式文件定义 (SCD)”。

使用 cicsdb2import 建立 DB2 上的数据文件

本过程的目的在于使用模式文件定义 (SCD) 中指定的数据，将表格添加到 CICS 文件管理的 DB2 数据库。这将使用 **cicsdb2import** 命令。

SCD 是一组表格和索引规格说明，它能够应用于任何 DB2 数据库上的任何表格。在 SCD 中，可以指定：

- 表格类型
- 字段名、类型和长度
- 表的最大记录数
- 主要与辅助索引名
- 主要与辅助索引降序字段
- 文件与辅助索引预分配页面
- 文件与辅助索引卷名
- 可变长度字段

需要更多信息，请参阅 *CICS 管理参考大全* 中的“模式文件定义 (SCD)”。

使用该过程：

- 在能够将数据文件添加到 DB2 数据库之前，必须完成第243页的『为队列和文件管理设置 DB2』。
- 必须用 DB2 管理特权注册到用户标识符。

需要使用 **cicsdb2import** 将 CICS 文件管理的表格添加到 DB2 数据库中：

1. 注册一个具有超级用户 (root) 特权的用户标识符，该用户标识符如第3页的『建立 CICS 系统管理员』中描述的那样建立。
2. 若正在 DCE 单元中工作，则作为管理员 (dce_login cell_admin) 注册到 DCE。
3. 使用 **cicsdb2import** 命令，如 *CICS 管理参考大全* 中的“cicsdb2import - 建立 DB2 的元数据”描述。例如，需要使用来自名为 SCHEMA1 模式文件的项在数据库 ALIAS2 上创建 FILE1 和 FILE2 的表，则可使用下列命令：

```
cicsdb2import -i SCHEMA1 -d ALIAS2 -u DB2USER -p xy93k1d \  
-f FILE1 -x INDEX1 -x INDEX2 -x INDEX3 \  
-f FILE2 -x INDEX4 -x INDEX5
```

在本命令中，FILE1 有索引 INDEX1、INDEX2 和 INDEX3，FILE2 有索引 INDEX4 和 INDEX5。用该命令将 DB2 用户标识符和口令传递到 DB2。

使用 **cicssfsimport** 建立 SFS 上的数据文件

本过程的目的在于使用模式文件定义（SCD）中指定的数据，将文件添加到 CICS 文件管理的 SFS 逻辑卷。这将使用 **cicssfsimport** 命令。

SCD 是一组文件和索引规格说明，它能够应用于任何 DB2 上的任何文件。在 SCD 中，可以指定：

- 文件类型
- 字段名、类型和长度
- 文件记录的最大数
- 主要与辅助索引名
- 主要与辅助索引降序字段
- 文件与辅助索引预分配页面
- 文件与辅助索引卷名
- 可变长度字段

需要更多信息，请参阅 *CICS 管理参考大全* 中的“模式文件定义（SCD）”。

在执行该过程前，下列过程必须完成：

- 必须配置区域。参阅第21页的『第2章 创建和配置 CICS 区域』。
- 第236页的『准备 SFS 的物理存储设备』（若是新 SFS，或若区域无法使用现存的逻辑卷）。
- 第237页的『使用 **cicssfscreate** 定义 SFS』（若是新 SFS）。
- 第239页的『使用 **cicssfs** 启动 SFS』。

确保区域定义（RD）**DefaultFileServer** 属性或文件定义（FD）**FileServer** 属性设置为用于这些文件的 SFS 名称。例如，若已经使用过程为该配置准备 **hostB**，则检查 **DefaultFileServer** 或 **FileServer** 是否设置为 **././cics/sfs/hostB**。

需要使用 **cicssfsimport**，将用于 CICS 文件管理的文件添加到 SFS 逻辑卷：

1. 注册一个具有超级用户（root）特权的用户标识符，该用户标识符如第3页的『建立 CICS 系统管理员』中描述的那样建立。
2. 若正在 DCE 单元中工作，则作为管理员（**dce_login cell_admin**）注册到 DCE。
3. 若 SFS 定义为使用 DCE 作为安全性，（若 **ProtectionLevel** 比 **none** 高），则必须作为 DCE 单元管理员注册。
4. 若 SFS 建立为仅接受认证过的 RPC，则需要将模式文件定义（SCD）应用于 SFS，您还需要成为 **cics admin** 组中的一员。
5. 使用 **cicssfsimport** 命令，如 *CICS 管理参考大全* 中的“**cicssfsimport** - 建立 SFS’元数据”所描述。例如，需要使用来自名为 **SCHEMA1** 的 SCD 的项在服务器 **hostB** 上创建两个文件 **FILE1** 和 **FILE2**，使用下列命令：

```
cicssfsimport -i SCHEMA1 -s ././cics/sfs/hostB \  
-f FILE1 -x INDEX1 -x INDEX2 -x INDEX3 \  
-f FILE2 -x INDEX4 -x INDEX5
```

用该命令，**FILE1** 有索引 **INDEX1**、**INDEX2** 和 **INDEX3**，**FILE2** 有索引 **INDEX4** 和 **INDEX5**。

注意:

当 **SFS** 冷启动时, 将删除所有 **SFS** 逻辑卷上的文件。每次 **SFS** 冷启动都必须再次创建这些文件。

安全性注意事项: CICS 拥有保护文件不受未经授权的用户访问的设施 (第218页的『授权用户访问 CICS 资源』中描述)。DCE 可用于增强这种安全性。参阅第210页的『DCE RPC 和 ACL 如何维护系统的安全』。

使用 **cicsddt** 建立 DB2 上的数据文件

本过程的目的在于使用 CICS DB2 诊断工具 (**cicsddt**), 将文件添加到 CICS 文件管理的 DB2 数据库。本工具提供了到 DB2 数据库功效强大的交互式接口。虽然 **cicsddt** 有几个选项, 但是, **create** 是该过程最常用的一个, 因为它用来创建新 DB2 文件。也可以使用 **cicsddt**, 以:

- 列出 DB2 数据库上的所有文件。
- 从 DB2 数据库上读记录。
- 将记录写入 DB2 数据库上的文件。
- 清空所有确认记录的文件。
- 从 DB2 数据库删除文件。
- 将辅助索引添加到 DB2 数据库文件。
- 从 DB2 数据库文件删除辅助索引。
- 显示有关 DB2 数据库文件的详细信息。
- 将 VSAM 文件传送并转换为 DB2 文件。
- 将 DB2 文件转换为平面文件。
- 将平面文件转换为 DB2 文件。

也可与 **cicssdt** 结合使用 **cicsddt**, 以在 SFS 和 DB2 之间迁移文件。参阅第29页的『从 SFS 将队列和文件管理移动到 DB2』。

使用该过程:

- 在能够将数据文件添加到 DB2 数据库之前, 必须完成第243页的『为队列和文件管理设置 DB2』。
- 必须用 DB2 管理特权注册到用户标识符。

需要使用 **cicsddt**, 将用于 CICS 文件管理的文件添加到 DB2 数据库:

1. 注册到一个具有超级用户 (root) 特权的用户标识符, 该用户标识符如第3页的『建立 CICS 系统管理员』中描述的那样建立。
2. 若正在 DCE 单元中工作, 则作为管理员 (dce_login cell_admin) 注册到 DCE。
3. 对于每个数据文件, 使用 **cicsddt** 命令, 如 *CICS 管理参考大全* 中的“cicsddt - DB2 诊断工具”所描述的。例如, 若需要在 DB2 数据库 ALIAS 上创建一个名为 FILE1 的文件, 则使用下列命令:

```
cicsddt -s ALIAS -c create FILE1
```

当使用本命令时, 可能会提示出:

- 文件使用的数据库
- 文件组织

- 字段名
- 字段类型
- 字段大小
- 最大记录数
- 主索引名
- 索引字段名
- 索引字段定序，例如升序
- 分配的页面数

使用 **cicssdt** 建立用户文件

本过程的目的在于使用 CICS SFS Diagnostic Tool，将用户文件添加到 CICS 文件管理的 SFS 逻辑卷（**cicssdt**）。本工具提供了到 SFS 数据库的功效强大的交互式接口。虽然 **cicssdt** 有几个选项，但是，**create** 是该过程最常用的一个，因为它用来创建新 SFS 文件的。也可以使用 **cicssdt**，以：

- 列出 SFS 上的所有文件。
- 从 SFS 上的数据库读记录。
- 将记录写入 SFS 上的数据库。
- 清空所有确认记录的文件。
- 从 SFS 删除文件。
- 将辅助索引添加到 SFS 文件。
- 从 SFS 文件删除辅助索引。
- 显示有关 SFS 文件的详细信息。
- 将 VSAM 文件传送并转换为 SFS 文件。
- 将 SFS 文件转换为平面文件。
- 将平面文件转换为 SFS 文件。

注意：

当 **SFS** 冷启动时，将删除所有 **SFS** 逻辑卷上的文件。每次 **SFS** 冷启动，必须再次创建这些文件。

在能够将用户文件添加到 SFS 逻辑卷之前，必须完成下列过程。

- 第236页的『准备 SFS 的物理存储设备』（若是新的 SFS，或若区域无法使用现存的逻辑卷）。
- 必须配置区域。参阅第21页的『第2章 创建和配置 CICS 区域』。
- 第237页的『使用 **cicssfscree** 定义 SFS』（若是新的 SFS）。
- 第239页的『使用 **cicssf** 启动 SFS』。

确保区域定义（RD）**DefaultFileServer** 属性或文件定义（FD）**FileServer** 属性设置为用于这些文件的 SFS 名称。例如，若已经使用过程为该配置准备 **hostB**，则检查 **DefaultFileServer** 或 **FileServer** 是否设置为 **/./cics/sfs/hostB**。

需要使用 **cicssdt**，将用于 CICS 文件管理的文件添加到 SFS 逻辑卷：

1. 注册到一个具有超级用户（root）特权的用户标识符，同时它也是 **cics** 组中的一员。

2. 若 SFS 定义为使用 DCE 安全性，（若 **ProtectionLevel** 比 **none** 高），则必须作为 DCE 单元管理员注册。
3. 对于每个数据文件，使用 **cicssdt** 命令，如 *CICS 管理参考大全* 中的“cicssdt - SFS 诊断工具”所描述。例如，若需要在 SFS hostB 上创建一个名为 FILE1 的文件，则使用下列命令：

```
cicssdt -s hostB -c create FILE1
```

当使用它时，可能会提示：

- 用于文件的逻辑卷，如 sfs_ShostB
- 文件组织，如 btreeclustered
- 字段名
- 字段类型，如 bytearray 和 varlenbytearray
- 字段大小
- 最大记录数
- 主索引名
- 索引字段名
- 索引字段定序，例如升序
- 分配的页面数

安全性注意事项：CICS 拥有保护文件不受未经授权的用户访问的设施（第218页的『授权用户访问 CICS 资源』中描述）。DCE 可用于增强这种安全性。参阅第210页的『DCE RPC 和 ACL 如何维护系统的安全』。

— 相关信息 —

转至 *CICS 管理参考大全* 中的 CEMT 运行时间资源管理，以使用 CEMT:

- 查看运行时间文件定义 (FD)。
- 查看运行时间文件的位置。
- 查看运行时间文件的状态。
- 更改运行时间文件状态。
- 查看或更改运行时间文件处理选项。
- 查看或更改运行时间文件的基名。
- 查看或更改运行时间文件的索引名。
- 查看或更改运行时间文件的文件管理器。

需要查找如何在运行时间中写 CICS 应用程序管理状态或文件操作，请转至 *CICS 应用程序设计参考大全* 中的“INQUIRE 和 SET 命令”。

CICS 应用程序设计指南 中的“文件服务”和“SFS 的 VSAM 仿真”。

CICS 内部通信指南 中的“定义远程文件”。

建立临时存储器队列 (TSQ)

当一个应用程序创建临时存储器队列时，CICS 将其在临时存储器队列 (TSD) 模板中定义的属性赋给它。

当区域和文件管理器建立时，区域需求的临时存储器将会建立。参阅第21页的『第2章 创建和配置 CICS 区域』。

CICS 应用程序设计指南 中的“临时存储器队列服务”描述了如何使用 CICS API 将数据存储于 TSQ，或在主存储器中，或在直接存取存储设备的辅助存储器中。

TSQ 模板在区域数据库和临时存储器定义 (TSD) 中定义。TSD 的目的是为区域确定 TSD 的给定属性。

关于临时存储器队列 (TSQ)

主 TSQ 的数据保存在内存中。辅助 TSQ 的数据保存在文件中。当文件管理器建立时，这些文件位于用区域定义 (RD) **DefaultFileServer** 属性指定的文件管理器上。若使用第21页的『第2章 创建和配置 CICS 区域』中描述的过程时，文件给定必需的文件名、索引和结构。

第51页的表 10 列出 TSQ 文件的文件名和索引名。

表 10. TSQ 文件和索引的名称

对于这类队列:	命名的 SFS 文件和索引（系统设定值）:	命名的 DB2 文件和索引（强制性）:
可恢复	<i>regionName</i> cicsrectsqfile <i>cicsrectsqidx</i>	<i>regionName</i> rectsq# <i>dxregionName</i> rectsq#
不可恢复	<i>regionName</i> cicsnrectsqfil <i>cicsnrectsqidx</i>	<i>regionName</i> nrectsq# <i>dxregionName</i> nrectsq#

配置临时存储器定义（TSD）

需要使用临时存储器定义（TSD）：

1. 若正在 DCE 单元中工作，则作为管理员（dce_login cell_admin）注册到 DCE。当且仅当正在使用 **-B** 选项，并且 CICS 区域正在用 DCE 保护级别运行时才需要它。
2. 使用下列命令例子：

添加新 TSD

本例用关键字 『QUE1』 添加 TSD 项。对于所有其它 TSD 属性假设系统设定值：

```
cicsadd -c tsd -r regionA -P QUE1
```

需要更多有关添加定义的信息，请参阅 *CICS 管理参考大全* 中的“cicsadd - 添加 CICS 资源定义”。

更改现存的 TSD

本例显示如何更改 TSD 项，以致区域下一次冷启动时不将其装入运行时间。

```
cicsupdate -c tsd -r regionA -P QUE1 \  
            ActivateOnStartup=no \  
            Group=''
```

需要将定义安装入运行时间，参阅第143页的『从永久数据库安装一个资源定义』。

注：若 **Group** 属性包含一个包括在区域 **Groups** 列表中列出的组名，则无论 **ActivateOnStartup** 值是什么，都将安装 TSD 项。

需要更多有关更改定义的信息，参阅 *CICS 管理参考大全* 中的“cicsupdate - 修改 CICS 资源定义”。

删除 TSD 项

本例显示如何从永久数据库中删除 TSD 项目。

```
cicsdelete -c tsd -r regionA -P QUE1
```

需要从运行时间删除项，参阅第144页的『仅从运行时删除资源定义』。

需要更多有关删除定义的信息，参阅 *CICS 管理参考大全* 中的“cicsdelete - 删除 CICS 资源定义”。

查看 TSD 项

本例显示如何查看 TSD 项：

```
cicsget -c tsd -r regionA QUE1
```

需要更多有关查看定义的信息，参阅 *CICS 管理参考大全* 中的“cicsget - 获得 CICS 资源定义信息”。

需要使用 SMIT 为区域定义资源，参阅第10页的『使用 SMIT 建立 CICS 系统』。
需要一套完整的用于 TSD 所有属性的说明，请参阅 *CICS 管理参考大全* 中的“临时存储器定义 (TSD)”。

— 相关信息 —

CICS 应用程序设计指南 中的“临时存储器队列服务”。

CICS 内部通信指南 中的“定义远程临时存储器队列”。

CICS 管理参考大全 中的“临时存储器定义 (TSD)”和“文件服务器的区域定义 (RD)”。

建立瞬时数据队列 (TDQ)

瞬时数据队列 (TDQ) 包含对 CICS 定义符号目的地的属性，在 CICS 区域中，瞬时数据路由到达或来自该符号目的地。

在应用程序使用队列前，每个 TDQ 的名称必须在 TDD 中定义给 CICS。与临时存储器队列不同的是 TDQ 无法动态由 EXEC CICS WRITEQ 命令从应用程序创建。

关于瞬时数据队列 (TDQ)

当区域和文件管理器建立时，区域需求的瞬时数据队列将会建立。参阅第21页的『第2章 创建和配置 CICS 区域』。

CICS 应用程序设计指南 中的“临时数据队列服务”描述了 CICS API 如何为随后的内部或外部处理排队数据。在应用程序中所指定的选定数据可路由到或来自预先定义的符号目的地，无论是内分区还是附加分区。

若目的地与分配到区域的设施相关联，则该目的地是内分区，若数据导向一个区域外部的目的地，则该目的地是附加分区。目的地必须在瞬时数据定义 (TDD) 中定义。

每个 CICS 区域需求 TDQ 的三个文件。一个保存逻辑可恢复 TDQ 数据，一个保存物理可恢复 TDQ 数据，一个保存不可恢复 TDQ 数据。当文件管理器建立时，这些文件位于用区域定义 (RD) **DefaultFileServer** 属性指定的文件管理器上。若使用在第21页的『第2章 创建和配置 CICS 区域』描述的过程时，文件给定必需的文件名、索引和结构。

第53页的表 11 列出 TDQ 文件的文件名和索引名。

表 11. TDQ 文件名

对于这类队列	命名 DB2 文件和索引（强制性）：	命名 SFS 文件和索引（系统设定值）：
逻辑可恢复	<i>regionName</i> logtdq# <i>dxregionName</i> logtdq#	<i>regionName</i> cicstdqlgfile cicstdqlgidx
物理可恢复	<i>regionName</i> epgtdq# <i>dxregionName</i> ephtdq#	<i>regionName</i> cicstdqphfile cicstdqphidx
不可恢复	<i>regionName</i> enreqtdq# <i>dxregionName</i> enrectdq#	<i>regionName</i> cicstdqnofile cicstdqnoidx

配置瞬时数据定义（TDD）

需要使用瞬时数据定义（TDD）：

1. 若正在 DCE 单元中工作，则作为管理员（dce_login cell_admin）注册到 DCE。当且仅当正在使用 **-B** 选项，并且 CICS 区域正在用 DCE 保护级别运行时才需要它。
2. 使用下列命令例子：

需要添加新 TDD

本例用关键字 『QUE1』 添加 TDD 项。对于所有其它 TDD 属性假设系统设定值：

```
cicsadd -c tdd -r regionA -P QUE1
```

需要更多有关添加定义的信息，请参阅 *CICS 管理参考大全* 中的“cicsadd - 添加 CICS 资源定义”。

更改现存的 TDD

本例显示如何更改 TDD 项，以致区域下一次冷启动时不将其装入运行时间。

```
cicsupdate -c tdd -r regionA -P QUE1 \
    ActivateOnStartup=no \
    Group=''
```

需要将定义安装入运行时间，参阅第143页的『从永久数据库安装一个资源定义』。

注：若 **Group** 属性包含一个包括在区域 **Groups** 列表中列出的组名，则无论 **ActivateOnStartup** 值是什么，都将安装 TDD 项。

需要更多有关更改定义的信息，参阅 *CICS 管理参考大全* 中的“cicsupdate - 修改 CICS 资源定义”。

删除 TDD 项

本例显示如何从永久数据库中删除 TDD 项目。

```
cicsdelete -c tdd -r regionA -P QUE1
```

需要从运行时间删除项，参阅第144页的『仅从运行时删除资源定义』。

需要更多有关删除定义的信息，参阅 *CICS 管理参考大全* 中的“cicsdelete - 删除 CICS 资源定义”。

查看 TDD 项

本例显示如何查看 TDD 项：

```
cicsget -c tdd -r regionA QUE1
```

需要更多有关查看定义的信息，参阅 *CICS 管理参考大全* 中的“cicsget - 获得 CICS 资源定义信息”。

需要使用 SMIT 为区域定义资源，参阅第10页的『使用 SMIT 建立 CICS 系统』。
需要一套完整的用于 TDQ 所有属性的说明，请参阅 *CICS 管理参考大全* 中的“瞬时数据定义 (TDD)”。

— 相关信息 —

第21页的『第2章 创建和配置 CICS 区域』。

转至 *CICS 管理参考大全* 中的 CEMT 运行时间资源管理，以使用 CEMT:

- 查看运行时间瞬时数据定义 (TDD)。
- 查看运行时间 TDQ 的队列类型。
- 查看或更改运行时间 TDQ 的触发器级别。
- 查看间接运行时间 TDQ 指向的队列。
- 查看需要触发的事务名。
- 查看与运行时间 TDQ 相关联的终端名。
- 查看运行时间 TDQ 的状态。
- 更改运行时间 TDQ 状态。
- 清空附加分区运行时间 TDQ。

需要查找如何在运行时间中写 CICS 应用程序管理状态或瞬时数据队列操作，请转至 *CICS 应用程序设计参考大全* 中的“INQUIRE 和 SET 命令”。

CICS 应用程序设计指南 中的“临时数据队列服务”。

CICS 内部通信指南 中的“定义远程瞬时数据队列”和“为功能装运定义远程资源”。

CICS 管理参考大全 中的“临时存储器定义 (TSD)”和“结构化文件服务器定义 (SSD)”。

建立日志定义 (JD)

日志是一组专用的顺序文件。日志可能包含任何应用程序需要用以方便后续事件重构或数据修改的数据。例如,一个日志能用作审查记录、记录数据库更新和添加的更改文件或事务传递给系统的记录（通常称为日志）。每个日志可从任何任务写。

关于日志定义 (JD)

用于日志的各种属性在 *CICS 管理参考大全* 中的“日志定义 (JD)”有所描述。特别重要的是属性 **DiskA** 和 **DiskB**。用于指定文件名（在 *regionName/data* 中）的这些属性保存日志。虽然一次仅使用一个文件，但是，两个文件必须都为可写入文件。使用中

的文件设置为 **current**（参阅 **DiskAStatus** 和 **DiskBStatus** 属性）。可以使用 EXEC CICS SET JOURNAL 或 CEMT SET JOURNALNUM，请求在 **DiskA** 和在 **DiskB** 中的替代文件之间切换。

CICS 从不清空用户日志，所以当不再需要时必须清除它们。若 CICS 是脱机的，则将定义为 **DiskA** 和 **DiskB** 的两个操作系统文件删除。

配置日志定义（JD）

需要使用永久数据库中的日志定义（JD）：

1. 若正在 DCE 单元中工作，则作为管理员（dce_login cell_admin）注册到 DCE。当且仅当正在使用 **-B** 选项，并且 CICS 区域正在用 DCE 保护级别运行时才需要它。
2. 使用下列命令例子：

需要添加新 JD

本例添加 CICS 识别为『01』的 JD 项，并为日志记录使用下列文件：

```
/var/cics_regions/regionA/data/J1PRIME  
/var/cics_regions/regionA/data/J1ALT
```

对于所有其它 JD 属性假设系统设定值：

```
cicsadd -c jd -r regionA -P 01 \  
DiskA=J1PRIME \  
DiskB=J1ALT
```

如同 **DiskAStatus** 的系统设定设置是 **current**，且 **DiskBStatus** 的系统设定设置是 **ready** 一样，/var/cics_regions/regionA/data/J1PRIME 用于日志记录除非更改设置。

若区域正在运行，并且希望将定义添加到运行时间数据库和永久数据库，则使用 **-B** 选项，如下显示：

```
cicsadd -c jd -r regionA -B 01 \  
DiskA=J1PRIME \  
DiskB=J1ALT
```

需要更多有关添加定义的信息，请参阅 *CICS 管理参考大全* 中的“cicsadd - 添加 CICS 资源定义”。

更改现存的 JD

本例显示如何更改 JD 项，以致区域下一次冷启动时不安装其入运行时间中。

```
cicsupdate -c jd -r regionA -P 01 \  
ActivateOnStartup=no \  
Group=''
```

需要将定义安装入运行时间，参阅第143页的『从永久数据库安装一个资源定义』。

注：若 **Group** 属性包含一个包括在区域 **Groups** 列表中列出的组名，则无论 **ActivateOnStartup** 值是什么，都将安装 JD 项。

需要更多有关更改定义的信息，参阅 *CICS 管理参考大全* 中的“cicsupdate - 修改 CICS 资源定义”。

删除 JD 项

本例显示如何从永久数据库中删除 JD 项目。

```
cicsdelete -c jd -r regionA -P 01
```

需要从运行时间删除项，参阅第144页的『仅从运行时删除资源定义』。

需要更多有关删除定义的信息，参阅 *CICS 管理参考大全* 中的“cicsdelete - 删除 CICS 资源定义”。

查看 JD 项

本例显示如何查看 JD 项：

```
cicsget -c jd -r regionA 01
```

需要更多有关查看定义的信息，参阅 *CICS 管理参考大全* 中的“cicsget - 获得 CICS 资源定义信息”。

需要使用 SMIT 为区域定义资源，参阅第10页的『使用 SMIT 建立 CICS 系统』。

需要一套完整的用于日志所有属性的说明，请参阅 *CICS 管理参考大全* 中的“日志定义 (JD)”。

相关信息

转至 *CICS 管理参考大全* 中的 CEMT 运行时间资源管理，以使用 CEMT：

- 查看运行时间日志定义 (JD)。
- 打开或关闭运行时间日志。
- 切换磁盘文件或运行时间日志。
- 查看运行时间日志磁盘文件的状态。
- 清空日志。

需要查找如何在运行时间中写 CICS 应用程序管理状态或日志操作，请转至 *CICS 应用程序设计参考大全* 中的“INQUIRE 和 SET 命令”。

CICS 管理参考大全 中的“CEMT INQ/SET JOURNAL”和“日志定义 (JD)”。

建立终端定义 (WD)

终端定义 (WD) 定义区域内可用的终端。虽然提供了几类终端的预定义终端定义，但是，在必需时仍能够对其进行添加、修改或删除。

CICS 应用程序设计指南 中的“终端服务”描述 CICS 是如何处理数据翻译、事务标识、输入输出操作的同步化以及读出、写入终端的会话控制。从而使应用程序不必再负责控制终端。

CICS 提供的自动安装模型

自动安装模型是终端定义 (WD) 项，当调用会话和将自动安装终端定义到 CICS 系统时，它能够作为模型使用。CICS 附带的 WD 项包含几个可作为模型使用的定义。以下列出每个模型。

关键字 说明

- MDCL** 对于 IBM CICS 客户程序产品模型（设备类型 `ibm-cics-client`）。
- MDCI** 对于 IBM CICS ECI 模型（设备类型 `ibm-cics-eci`）。
- MD01** 对于 `vt100` 模型（设备类型 `vt100`）。
- MD02** 对于 `aixterm` 模型（设备类型 `aixterm`）。
- MD03** 对于 `aixterm-m` 模型（设备类型 `aixterm-m`）。
- MD04** 对于 `ibm3151` 模型（设备类型 `ibm3151`）。
- MD05** 对于 `ibm3161` 模型（设备类型 `ibm3161`）。
- MD06** 对于 `ibm3164` 模型（设备类型 `ibm3164`）。
- MD07** 对于高性能多字节颜色（设备类型 `hft-mb`）。
- MD08** 对于高性能单字节颜色（设备类型 `hft`）。
- MD09** 对于高性能多字节单色（设备类型 `mft-mb`）。
- MD10** 对于高性能单字节单色（设备类型 `mft`）。
- MD11** 对于 ASCII 多字节单色（设备类型 `lft-mb`）。
- MD12** 对于 ASCII 单字节单色（设备类型 `lft`）。
- MD13** 对于 `hp` 终端定义（设备类型 `hp`）。
- MD14** 对于 `hpterm` 模型定义（设备类型 `hpterm`）。
- MD15** 对于 `xterm` 模型（设备类型 `xterm`）。
- MD16** 对于 `cicsteld` 的 `327x-2-E`（设备类型 `3270C`）。
- MD17** 对于 `cicsteld` 的 `327x-2`（设备类型 `3270M`）。
- MD18** 对于 `vt220` 模型（设备类型 `vt220`）。
- MD19** 对于 `dxterm`（`vt300`）模型（设备类型 `vt300`）。
- MD20:** 对于 `dtterm` 模型（设备类型 `dtterm`）。
- MD21** 对于 `cicsteld` 的 `327x-2-E`（设备类型 `3270C-3`）。
- MD22** 对于 `cicsteld` 的 `327x-2`（设备类型 `3270M-3`）。
- MD23** 对于 `cicsteld` 的 `327x-4-E`（设备类型 `3270C-4`）。
- MD24** 对于 `cicsteld` 的 `327x-4`（设备类型 `3270M-5`）。
- MD25** 对于 `cicsteld` 的 `327x-5-E`（设备类型 `3270C-5`）。
- MD26** 对于 `cicsteld` 的 `327x-5`（设备类型 `3270M-5`）。
- MD27** 对于 `97801` 模型（设备类型 `97801`）。
- MD28** 对于 Sun 命令工具和外壳工具（设备类型 `sun-cmd`）。

注:

1. 自动安装模型通过其 **DevType** 属性识别。
2. 有几个属性未在本表中显示。需要参阅自动安装模型的完整属性列表，请使用 **cicsget** 命令，如第144页的『仅从运行时删除资源定义』中描述。

选择模型类型

用 **cicsterm** 和 **cicsteld** 过程选定模型类型，如下列所示：

1. 使用带有 **WD DevType** 的模型，它与 **cicsterm -m** 选项中指定的模型参数匹配。
2. 若未指定 **-m** 选项，则选定的模型与环境变量 **CICSTERM** 匹配并带有 **DevType**。若未设置 **CICSTERM**，则选定的模型与环境变量 **TERM** 相匹配并带有 **DevType**。
3. 系统设定自动安装程序选中最先通过其的那个模型。
4. 当区域为 Telnet 客户程序而连接时，**cicsteld** 检查 Telnet 客户程序是否正在使用一个有效的终端。若终端是有效的，**cicsteld** 为该终端选择一个模型类型。
5. **cicsteld -e** 选项能用于覆盖 **cicsteld** 选定的模型类型。本选项允许指定一个用户定义的模型类型或 CICS 提供的模型类型。可用于 **cicsteld** 更好的 CICS 模型类型是带有扩展数据流的终端的 **hft** 和 **mft**。以及不带有扩展数据流的终端的 **lft**。

更改提供的模型类型定义

使用 **cicsupdate** 命令更改特定模型类型的属性。例如：

```
cicsupdate -c wd -r regionName key \
          attribute=newValue \
          .
          .
          .
          attribute=newValue
```

其中，*regionName* 是区域名，*attribute=newValue* 是希望更改的属性以及更改为的值。

配置终端定义 (WD)

需要使用永久数据库中的终端定义 (WD)：

1. 若正在 DCE 单元中工作，则作为管理员 (**dce_login cell_admin**) 注册到 DCE。当且仅当正在使用 **-B** 选项，并且 CICS 区域正在用 DCE 保护级别运行时才需要它。
2. 使用下列命令例子：

添加新 WD

本例添加 CICS 识别为 **TERM** 的 **WD** 项，这是自动安装用户出口程序的候选程序（参阅第115页的『使用终端自动安装用户程序』），并且针对的是支持“多字节字符集”输入的终端。为其它所有 **WD** 属性假设系统设定值：

```
cicsadd -c wd -r regionA -P TERM \
        ModelId=MyTerm \
        SOSI=yes
```

若区域正在运行，并且希望将定义添加到运行时间数据库和永久数据库，则使用 **-B** 选项，如下显示：

```
cicsadd -c wd -r regionA -B TERM \
        ModelId=MyTerm \
        SOSI=yes
```

需要更多有关添加定义的信息，请参阅 *CICS 管理参考大全* 中的“**cicsadd** - 添加 CICS 资源定义”。

更改现存的 WD

本例显示如何更改 WD 项，以致区域下一次冷启动时不将其装入运行时间。

```
cicsupdate -c wd -r regionA -P TERM \  
          ActivateOnStartup=no \  
          Group=''
```

需要将定义安装入运行时间，参阅第143页的『从永久数据库安装一个资源定义』。

注：若 **Group** 属性包含一个包括在区域 **Groups** 列表中列出的组名，则无论 **ActivateOnStartup** 值是什么，都将安装 WD 项。

需要更多有关更改定义的信息，参阅 *CICS 管理参考大全* 中的“cicsupdate - 修改 CICS 资源定义”。

删除 WD 项

本例显示如何从永久数据库中删除 WD 项目。

```
cicsdelete -c wd -r regionA -P TERM
```

需要从运行时间删除项，参阅第144页的『仅从运行时删除资源定义』。

需要更多有关删除定义的信息，参阅 *CICS 管理参考大全* 中的“cicsdelete - 删除 CICS 资源定义”。

查看 WD 项

本例显示如何查看 WD 项：

```
cicsget -c wd -r regionA TERM
```

需要更多有关查看定义的信息，参阅 *CICS 管理参考大全* 中的“cicsget - 获得 CICS 资源定义信息”。

需要使用 SMIT 为区域定义资源，参阅第10页的『使用 SMIT 建立 CICS 系统』。

需要一套完整的用于终端所有属性的说明，请参阅 *CICS 管理参考大全* 中的“终端定义 (WD)”。

— 相关信息 —

转至 *CICS 管理参考大全* 中的 CEMT 运行时间资源管理，以使用 CEMT:

- 查看运行时间终端定义 (WD)。
- 查看或更改运行时间终端的优先级。
- 在 CICS 区域中将运行时间终端置入置出服务。
- 使运行时间终端可用于 CICS 事务。
- 使运行时间终端不可用于其启动的事务。
- 查询 CICS 运行时间终端的网络名。
- 清除 CICS 运行时间终端上的任务。
- 查询自动安装程序名。
- 设置新自动安装程序的名称。

需要查找如何在运行时间中写 CICS 应用程序管理状态或终端操作，请转至 *CICS 应用程序设计参考大全* 中的“INQUIRE 和 SET 命令”。

CICS 应用程序设计指南 中的“终端服务”、“终端服务的设计思想”和“3270 终端仿真”。

需要为 DBCS/MBCS 环境选择模型的信息，请参阅第269页的『附录D. 配置 DBCS/MBCS 环境的 CICS』。

在 *CICS 管理参考大全* 中，参阅:

- “cicsteld - 将 telnet 客户程序连接至区域”
- “cicsterm - 调用 CICS 3270 终端仿真器”
- “已提供的终端自动安装程序”
- “当存取终端自动安装程序时”
- “终端自动安装程序的参数”

建立监控定义 (MD)

若应用程序使用 EXEC CICS ENTER 语句写跟踪入口，则需要监控定义 (MD)。 *CICS 管理参考大全* 中的“监控定义 (MD)”描述了 MD 属性。

— 相关信息 —

转至 *CICS 管理参考大全* 中的 *CEMT 运行时间资源管理*，以使用 *CEMT*：

- 查找监控是否在运行时间内切换。
- 在运行时间内切换监控设施开关。

第116页的『使用性能监控用户程序』。

第197页的『监控性能』。

CICS 管理参考大全 中的“性能监视器用户程序”、“监控定义 (MD)”和“*CEMT INQ/SET MONITOR*”。

建立 Micro Focus COBOL 运行时间环境

CICS 应用程序设计指南 中的“翻译、编译和链接编辑 *CICS 应用程序*”描述了如何使用支持的编译器。Micro Focus COBOL 需要一些配置。

何时使用 **cicsmkcobol**

若安装或升级 Micro Focus COBOL 产品，则需要使用 **cicsmkcobol** 命令。本命令创建 Micro Focus COBOL 语言方式文件。需要用于 *CICS*，必须用名称 **cicsprCOBOL** 将 COBOL 语言方式文件置于 `/usr/lpp/cics/bin` 或 `/var/cics_regions/regionName/bin` 中。在 `/var/cics_regions/regionName/bin` 中的文件版本优先于在 `/usr/lpp/cics/bin` 中的文件。这允许对于每个区域有不同的 COBOL 语言方式文件。

参阅 *CICS 管理参考大全* 中的“**cicsmkcobol** - 创建 **cicsprCOBOL** 文件”，以获得更多信息。

还参阅 *CICS 应用程序设计指南* 中的“更新非 *CICS 应用程序*的文件”。

— 相关信息 —

CICS 应用程序设计指南 中的“翻译、编译和链接编辑 *CICS 应用程序*”。

CICS 管理参考大全 中的“**cicsmkcobol** - 创建 **cicsprCOBOL** 文件”。

建立应用程序服务器

应用程序服务器是一个多线程的 *CICS* 进程，它提供运行 *CICS* 事务的完整环境。若您的事务是会话式的，则它将会仅由一个应用程序服务器处理，但是若它是拟会话式的，那么该序列中的每个事务将按顺序被不同的服务器执行。

配置 *CICS* 时，您需要为区域指定服务器的最小数目和最大数目。（参阅 *CICS 管理参考大全* 中“区域定义 (RD)”中的 **MaxServer** 和 **MinServer** 属性。）*CICS* 应用

程序管理器将确保始终存在最小数目的服务器，并且根据整个工作负荷情况将服务器创建到所定义的最大数目。

池中所有的应用程序服务器必须定位在相同的主机上，但是不需要与处理的 CICS 客户机位于同一个主机上。

Micro Focus COBOL 程序 and 应用程序服务器

每个应用程序服务器都加载了每个 Micro Focus COBOL 应用程序的一个副本，这些 Micro Focus COBOL 应用程序在应用程序服务器中使用。因为这样，系统要求对于即将装入每个应用程序服务器的每个 Micro Focus COBOL 应用程序都有足够的可用存储器。

若存储器容量不足且该标准无法满足，则通过将 Micro Focus COBOL COBSW 环境变量设置为限制 Micro Focus COBOL 在 Micro Focus COBOL 程序的应用程序服务器中所使用的存储器数量，以限制一个应用程序服务器为 Micro Focus COBOL 应用程序所使用的空间数量。

相关信息

需要使用 CEMT 在运行时间内更改 **MaxServer** 或 **MinServer** 值，请转至 *CICS 管理参考大全* 中的 CEMT 运行时间资源管理。

建立产品定义 (XAD)

产品定义 (XAD) 用于提供给 CICS 有关 CICS 应用程序访问的 XA 遵从产品的信息。

CICS 事务可以通过在 CICS 应用程序体中包含嵌入式 SQL 调用，以访问关系数据库管理器 (RDBM)。对于协调事务落实和恢复，RDBM 必须与 X/Open 分布式事务处理 (DTP) 标准相一致。

X/Open DTP 标准使用应用程序 (AP)、资源管理器 (RM) 和事务管理器 (TM) 模型。CICS 的实现过程中，CICS 应用程序是 AP，RDBM 是 RM，CICS 是 TM。参阅下列图表：

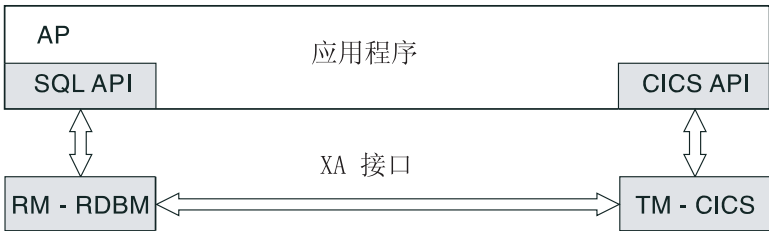


图 2. AP、RM、TM 模型的 CICS 实现

注：

1. 用于 CICS 队列和文件管理的 DB2 数据库需要 XAD。但是，在第28页的『在 DB2 上设置队列和文件管理』中所描述的过程为您创建了 XAD。

2. 与所有其它资源类不同的是: XAD 在热启动时从永久数据库装入。 当且仅当执行紧急重启动时, XAD 才从运行时间数据库中装入。
3. 可能要求您将环境变量添加到:

```
/var/cics_regions/regionName/environment
```

第90页的『SQL 数据库设置过程』描述了何时执行该步骤。

XAD 属性在 *CICS 管理参考大全* 中的“产品定义 (XAD)”有所描述。

— 相关信息 —

第90页的『SQL 数据库设置过程』。 第145页的『安装一组定义至运行时中』。

CICS 应用程序设计指南 中的“SQL 限制和关系数据库服务”和“一个 XA 启用关系数据库的事务范例”。

第4章 配置客户程序

CICS 最终用户作为客户程序连接到区域，以使用 CICS 应用程序、服务和资源。一个 CICS 最终用户客户程序可为：

- 开放系统上的 CICS 客户程序
- 有 3270 仿真能力的 Telnet 客户程序
- IBM cics 客户机产品
- 打印机

表 12. 定制用户和客户程序的导向图

若需...	参考...
使区域不受未经授权的用户访问的威胁。	第65页的『客户的安全性考虑』。
从开放系统上的 CICS（ cicsterm ）客户程序建立访问的区域。	第66页的『使用开放系统上的 CICS（ cicsterm ）客户』。
从 telnet 客户程序建立访问的区域。	第73页的『用 CICS 使用 telnet 客户』。
从 IBM CICS 客户机产品建立访问的区域。	第79页的『IBM cics 客户机产品』。
从 ECI 和 EPI 客户应用程序建立访问的区域。	第81页的『ECI 和 EPI 客户应用程序』。
用 ATI 事务查找 cicsterm 和 cicsteld 行为的差异。	第75页的『使用 ATI 事务的 cicsterm 和 cicsteld 行为差异』。
将新用户添加到区域。	第83页的『将新用户添加到区域』。
准备好区域从应用程序打印。	第84页的『如何为从 CICS 应用程序打印而建立 cicstermp 』。
与使用 telnet 的区域连接。	第79页的『使用 telnet 客户连接到区域』。

客户的安全性考虑

当区域配置为使用 DCE 安全性服务， **AuthenticationService** 属性设置为 **DCE** 并且为在创建区域时所定义的每个用户创建 DCE 委托人。接着，当使用 **SMIT** 或 **cicsimport** 命令将用户定义（UD）添加到区域时，也创建 DCE 委托人，正如 **AuthenticationService** 属性仍然设置为 **DCE**。

参阅第205页的『认证用户访问 CICS』，以获得更多信息。

若区域和资源配置为这样，则 TSL 和 RSL 密钥用于限制对区域内资源的访问。用户必须拥有与用户定义（UD） **TSLKeyList** 和 **RSLKeyList** 属性一起列出的 TSL 和 RSL 密钥，它们与资源 TSL 和 RSL 密钥相匹配，用于允许访问。参阅第214页的『授权访问 CICS 事务』和第218页的『授权用户访问 CICS 资源』，以获得更多信息。

相关信息

在 *CICS 管理参考大全* 中:

参阅“区域定义 (RD)”中的下列属性说明

- **FileRSLCheck**
- **JournalRSLCheck**
- **ProgramRSLCheck**
- **TemporaryStorageRSLCheck**
- **TransactionRSLCheck**
- **TransientDataRSLCheck**

参阅下列 **RSLKey** 属性的说明

- “日志定义 (JD)”
- “文件定义 (FD)”
- “瞬时数据定义 (TDD)”
- “临时存储器定义 (TSD)”
- “事务定义 (TD)”

还可参考“事务定义 (TD)”，以获得下列属性的说明:

- **RSLCheck**
- **TSLCheck**
- **TSLKey**

使用开放系统上的 **CICS (cicsterm)** 客户

cicsterm 客户在安装 CICS 客户机/服务器或仅 CICS 客户机时初始建立。若原始配置丢失或毁坏，或若安装新级别的操作系统或 DCE，则您可能需要再使用一次设置过程。

cicsterm 客户如何工作

CICS 通过 3270 信息显示系统数据流将信息呈示给用户。需要连接到 CICS 区域，最终用户需要其工作站使用 3270 数据流通信。**cicsterm** 命令仿真 3270 数据流并且能够从不具有这些功能的工作站使用。

注: 拥有具有 3270 仿真能力或安装 Telnet 客户的工作站的用户能够访问 CICS AIX 版区域，如第74页的『Telnet 客户和 cicsteld』中所描述的。

下列是为 **cicsterm** 配置区域和客户的必需条件:

1. 若与从客户请求的连接一起传递过来的委托人与 CICS 用户不关联，或若没有传递任何委托人，则用户使用区域的系统设定用户标识符注册。若该用户标识符没有足够的访问特权，则 **cicsadd -c ud** 能够用于建立所传递用户标识符的用户定义 (UD)。本命令在正运行区域的机器上使用。参阅 *快速入门* 中的“为用户添加用户定义 (UD)”。

注：若 DCE 用于认证，则所有使用 **cicsterm** 的用户应该为 **cicsterm** 组的成员。这用来保证它们对 `/var/cics_clients` 中的密钥表文件有存取权。

2. **cicscp** 和 **cicssetupclients** 在 开放系统上的 CICS 客户上建立 DCE。

使用 **cicsterm** 命令

当 **cicsterm** 接收到一个用户中断信号 SIGINT 时（用户按 Ctrl-C 键时生成），终端仿真器立即终止。若事务正在运行，则它会被清除。清除仅在 EXEC CICS 调用启动或终止时发生，所以若事务正在等待资源或在不调用 EXEC CICS 的循环中时，该事务不会终止。若需要取消处于这种情况中的事务，则系统管理员必须通过从 CICS 内使用 CEMT SET TASK (*termid*) FORCEPURGE 命令来清除它。

3270 密钥映象

终端键盘上的物理键到 3270 设备上的密钥映象由 3270 密钥映象文件管理。该文件命名为 `.3270keys`，并且位于用户的自用目录。若 **cicsterm** 在自用目录未找到该文件，它则使用 `/etc/3270.keys` 中的全系统文件。流行的终端类型 3270 密钥映象文件的范例能够在 `/usr/lpp/cics/etc` 目录中找到。需要在自用目录或 `/etc` 中包含的有用 3270 密钥映象文件为 `/usr/lpp/cics/etc/3270keys`，此文件类型着依据使用中的终端类型的适当范例。

3270 密钥映象文件还由 Telnet 使用，并且在操作系统文档中有完整的描述。请再参阅第74页的『Telnet 客户和 **cicsteld**』。

当用 **cicsterm** 来使用 3270 密钥映象文件时，请注意下列事宜：

- **cicsterm** 从新行 (`\n`) 分辨回车 (`\r`)，然而，Telnet 不能分辨。若终端上的键发出回车字符，则确认将其指定为 `\r`，而非 `\n`。由特定键发出的字符能够在 **terminfo** 数据库中校验。例如，IBM 3151 的功能在文件 `/usr/share/lib/terminfo/ibm.ti` 中描述。在该文件中，虽然 3151 有唯一的功能，但是，它将它们中的大多数与 IBM 3161 设备共享。在 `ibm3161` 下，F1 键定义为生成『`\ea\r`』，它是一个换码，紧接其后的是小写『`a`』和回车字符。
- **cicsterm** 试图尽可能地仿真一个实时 3270 设备，但是，特定键的行为与 Telnet 不同。例如，“下一个字”键将光标定位在下一个字的开始，而许多 Telnet 仿真器却将光标定位在下一个字的结尾。

使用高性能 **cicsterm** 程序

高性能 **cicsterm** 程序使用流接口，将正在通过终端和 **cicsterm** 客户之间的字符分成块。

注：高性能 **cicsterm** 程序仅用于 SBCS 场所。当使用 Micro Focus COBOL Animator 时，也不能使用高性能 **cicsterm** 程序。

支持的终端： 高性能 **cicsterm** 程序（以后指 **cicsterm**）支持多种终端设备，如 IBM 3151 和 vt100 ASCII 异步终端。在图形环境中，**cicsterm** 支持窗口类型，如 `aixterm` 和 `xterm`。

AIX 存储关于 **terminfo** 数据库中终端设备特性的信息。若终端在该数据库中定义，则 **cicsterm** 能够使用所提供的终端，并且该终端支持一组最小功能。（参阅操作系统文档，以获得更多关于 **terminfo** 数据库的信息。）必需的一组最小的终端功能是：

- clear (clear_screen)
- el (clr_eol)
- ed (clr_eod)
- cup (cursor_address3)

仅具有这些功能的终端将不能显示 BMS 支持的不同属性。当需要在映射包含扩展属性或颜色时需要正确地显示它时，需要一个带有适当功能的终端。

若映射显示不正确（例如，若为 BMS 指定了下划线，但未显示出来），则应该首先检查 terminfo 数据库中的终端功能。数据库位于 /usr/share/lib/terminfo 目录中。

cicsterm 从 TERM 环境变量中取得终端类型。这通常是自动设置的。TERM 的不正确设置是引起 **cicsterm** 不正确行为的原因。

颜色支持：**cicsterm** 上的颜色支持值得附加说明。并非仅当终端支持颜色时才显示颜色。但是，通过在一些单色设备上使用不同的灰色阴影来仿真颜色是可能的。例如，在附带单色显示器的 X-station 上，窗口类型 aixterm（非 aixterm-m）使用不同层次的灰色描述颜色。

cicsterm 在不同的颜色模式中操作：

- 显示颜色方式

在该模式中，为 BMS 映射中所有字段和字符显示地指定颜色。没有显示颜色设置的字段获取系统设定颜色。没有显示颜色设置的字符获取在字段中出现的颜色。3270 系统设定颜色是绿色。

- 四色模式

在该模式中，映射中不指定颜色。**cicsterm** 使用在密钥映象文件中的颜色属性显示所有字段。

多数映射在四色模式中工作良好。但是对于输出的更好控制（例如，需要在财务应用程序中用红色显示负值），则必需明确地指定颜色。当使用字符属性时，显示颜色模式也是必需的。

cicsterm 动态地从四色模式切换到显示颜色模式。这发生在当一个四色模式中的屏幕用显示颜色信息更新时。例如，若终端当前显示四色模式中的屏幕，应用程序不用 ERASE 就在底部写出红色的一行，则全屏从四色切换到显示颜色模式，并且没有颜色属性的字段获取系统设定颜色。这可能不是所必需的。最佳的策略是不在同一屏幕将颜色模式混合。

支持的数据流：**cicsterm** 支持所有能在单字节场所中的 BMS 映射中指定的命令、次序和属性。就 3270 数据流而言，它们如下所示：

命令

- WRITE
- ERASE WRITE
- WRITE ALTERNATE
- ERASE WRITE ALTERNATE
- ERASE ALL UNPROTECTED
- READ BUFFER
- READ MODIFIED

当前，没有方法能够指定可替代的屏幕大小，所以假设其与主屏幕相同。

次序

- START FIELD
- START FIELD EXTENDED
- SET BUFFER ADDRESS
- SET ATTRIBUTE
- MODIFY FIELD
- INSERT CURSOR
- PROGRAM TAB
- REPEAT TO ADDRESS
- ERASE UNPROTECTED TO ADDRESS

也支持格式控制次序。

字段属性:

支持所有字段属性字节中的位。数字字段限制输入为 0-9、加号、减号、逗号和十进制小数点。

扩展字段属性

支持下列扩展字段属性:

- 3270 字段属性
- 扩展突出显示
- 前景色
- 背景色

字段提纲可略微忽略。

字符属性

支持下列字符属性:

- 扩展突出显示
- 扩展彩色 - 仅前台

cicsterm 也支持未格式化的屏幕，如不包含字段的屏幕。

如何显示扩展属性: 扩展属性的显示要求具有高于等于基本设置功能的终端设备。**cicsterm** 使用下列映象显示 3270 属性的不同类型:

表 13. 终端功能比较

3270 功能	终端功能 (来自 terminfo)
下划线	smul (enter_underline_mode)
闪烁	blink (enter_blink_mode)
反向显示	rev (enter_reverse_mode)
前景色	colf0..7 (color_fg_0..7)
背景色	colb0..7 (color_bg_0..7)
在显示颜色模式增亮	bold (enter_bold_mode)

希望开发左边的 3270 属性，则需要支持表右边的功能的 ASCII 异步终端设备。

注意，在一些通讯包内的异步终端并不完全仿真其意旨仿真的硬件。terminfo 数据库项通常反映硬件的功能，而非其软件仿真器。

限制: **cicsterm** 当前有下列局限和限制:

字段属性

字段属性中『黑色』的前景色/背景色显示为『系统设定』。

字符属性

忽略字符属性中的背景色。

结构化字段

不支持结构化字段。

寻址能力

仅 12 位 ASCII 地址。

终端定义: **cicsterm** 支持所有 BMS 可定义属性, 并取决于上述详细描述的限制。但是, BMS 生成的数据流由在区域的终端定义 (WD) 中为终端设备类型而指定的区域终端定义修改。

一般而言, 指定任何单字节属性都是安全的, 如突出显示、前台和 ExtDS。 **cicsterm** 尽可能显示在数据流中显示存在的属性, 并且忽略那些无法在使用之中的终端设备中显示的属性。例如, 若一个终端不支持下划线字符, 则带有下划线属性的字段和字符都以不带下划线的方式显示。但是, 当 **cicsterm** 正在单字节场所内运行时, 不要启用多字节属性是重要的。

如上所述, 若四色模式中的屏幕用颜色信息更新, 则其可能切换到显示颜色模式。设置 **ERRColor** 可能产生这种情况。

解决牵涉 **cicsterm** 的问题

本部分告诉您如何解决一些常见的 **cicsterm** 问题, 它们可能是配置错误的结果。

若 **cicsterm** 不工作: 检查:

1. **cicssetupclients** 已经运行。当不用单元目录服务 (CDS) 和 DCE 安全性服务操作时, 仅需要 **-m** 选项。参阅第232页的『用 **cicssetupclients** 设置 DCE 和 **cicsterm** 客户』。

注: 当 **cicscp** 用于配置客户时, **cicssetupclients** 在运行。

2. DCE 的所有必需组件正在运行。
3. 属于或不属于 **cicsterm** 组的用户对 **/var/cics_clients** 目录中的密钥表文件有存取权。参阅 快速入门 中的“为用户添加用户定义 (UD)”。
- 当不用单元目录服务 (CDS) 和 DCE 安全性服务操作时, 将不需要它。
4. 3270 密钥文件包含将适当密钥联接到正在使用的终端类型键。参阅 快速入门。
5. 终端类型正确指定在 **TERM** 环境变量中。该环境变量由操作系统设置。若因为一些原因未设置正确值, 则使用覆盖任何 **TERM** 设置的 **CICSTERM** 环境变量。
6. **NLSPATH** 环境变量正确地建立为取得 CICS 信息编目。还检查 **LANG** 环境变量是否设置正确。
7. **/var** 文件系统已满。

若 **cicsterm** 无法连接到区域: 确保:

- 区域正在运行。
- 若区域的 **ProtectionLevel** 不是 **none**, 用户已经注册入 DCE。

- 若客户在 DCE 单元中，检查选择的区域在相同的单元内。
- 若未使用单元目录服务（CDS）和 DCE 安全性服务，则检查区域主机是否列在 CICS_HOSTS 环境变量中，或使用 **cicsterm -h** 选项指定主机。

若功能密钥不工作： 该问题经常由运行通过 ASCII 异步终端仿真器的 **cicsterm** 引起。例如，用户可能从 PC 拨号进入使用 vt220 终端仿真器的 UNIX 系统。TERM 环境变量可能正确设置为 vt220。然后，用户可启动 **cicsterm**，并找到尽管联接在密钥文件中而不工作的特定 3270 密钥。

这原因通常在于 ASCII 异步终端，并且一般表示有问题的密钥为：

- 未建立。
- 为本地密钥--它们对终端仿真器和调用本地功能（如文件传送）有特殊的意义。
- 生成不正确的密钥码--终端仿真器是生成非标准代码的原因。

向终端仿真器文档询问，以找到如何重联接这些密钥的方法。或者，能够添加需求多密钥笔划的联接，就如 ASCII 终端的 /etc/3270.keys 联接密钥。例如，可将 Esc 1 映射到 PF1，Esc 2 映射到 PF2。

发生密钥不工作的另一个可能的原因是使用了全系统 3270.keys 文件（就是，/etc/3270.keys）。为终端而不是该文件中的窗口化终端联接的密钥经常需求按多键，如 PF1 的 Esc 1。

若当连接到不同的区域时 cicsterm 的行为发生变化： BMS 生成的数据流由定义到区域的终端定义（WD）修改。确保终端在区域的 WD 启用了突出显示、前台、ExtDS 等等。

若 cicsterm 不正确地显示字段属性： 这可能由于上述 WD 的不正确设置、使用 **cicsteld** 时 ASCII 终端设备或终端仿真器不具有足够的功能。终端类型的终端功能在 /usr/share/lib/terminfo 目录中的 terminfo 数据库存储。

可简单地通过使用操作系统 **tput** 命令来检查功能。例如，需要确定终端是否有下划线功能，您可使用下列命令将终端置入下划线模式：

```
tput smul
```

若终端支持下划线，则所有随后的输入将被加上下划线。测试命令后复位下划线模式：

```
tput rmul
```

参考操作系统文档，以获得更多信息。

若 cicsterm 不能与给定的终端类型工作： 该症状能够由下列原因引起：

- 终端类型没有足够功能运行 **cicsterm**。一个有帮助的功能测试就是运行 vi 编辑器。若 vi 在通常全屏幕方式中的启动失败，则终端设备不能支持 **cicsterm**。
- **terminfo** 数据库不包含该终端类型的定义。

更改数字字段验证

该验证不能关闭，但是，算法在允许的字符集中不包括加号和逗号。这使欧洲用户能启用逗号表示数字字段中的十进制小数点。

注意，最终的输入数据验证是 CICS 事务本身的责任，如同数字字段的验证通常不在 3270 Telnet 客户中实现一样。

更改字段颜色

用于显示不同属性字段的颜色在 3270.keys 文件中指定。可能更改下列字段类型的前景色：

- 受保护的低亮度
- 不受保护的低亮度
- 受保护的高亮度
- 不受保护的高亮度

另外，可能指定背景色（仅在单字节场所）。样本密钥文件在 /usr/lpp/cics/etc 目录中。

这些颜色联接仅当屏幕处于四色模式中应用，就是，没有为屏幕上的任何字段和字符指定显示颜色。

相关信息

第209页的『cicsterm 客户的安全性考虑』。

第65页的『客户的安全性考虑』。

为与 CICS 客户机使用建立 DCE

若正在配置需要运行未安装 CICS 服务器的 CICS 客户机的机器，请使用本过程。本过程假设 DCE 单元中的 DCE 服务器已经用 CICS 对象配置。

在使用本过程前，验证本地主机的日期和时间与DCE 安全性服务器主机的日期和时间相同。您可使用几种方式中的一种：

- 若分布式时间服务（DTS）在单元内配置，包括 DCE 客户配置中的 DTS 客户。则如果在单元中有三个或三个以上机器，应该仅使用 DTS。
- 使用一些其它时间同步化服务。
- 若未使用时间同步化服务，手工将本地主机的日期和时间设置为 DCE 安全性服务器主机的日期和时间。

注意：

若在主机和单元之间发生时间偏差，DCE 安全性不允许 CICS 正确行使其功能。因此，不建议依赖手工调整使日期和时间同步。

1. 您需要了解单元名、DCE CDS 在的主机和 DCE 安全性服务器在的主机。从已在单元中的主机使用 **getcellname** 命令，以查找单元的名称。向 DCE 管理员询问 DCE 服务器主机的名称。
2. 注册到一个拥有超级用户特权的用户标识符，该用户标识符如第3页的『建立 CICS 系统管理员』中描述的那样建立。
3. 使用 **rmldce all** 命令删除任何现存的 DCE 配置。**注意：**
若在本地主机上配置 DCE 服务器，请勿使用 **rmldce all**。若在本地主机上配置 DCE 服务器，请按照 快速入门 中的步骤。
4. 选做下列之一：

在仅 RPC 设置中:

使用本命令:

```
cicscp -v -l logFile create dce -R
```

在 DCE 单元中:

选做下列之一:

- 若 DCE CDS 和 DCE 安全性服务器在相同的主机上, 使用本命令:

```
cicscp -v -l logFile create dce -n cellName [-s serverHost]
```

其中, *cellName* 是单元名称, *serverHost* 是运行 DCE CDS 和 DCE 安全性服务器机器的主机名。

- 若 DCE CDS 和 DCE 安全性服务器不在相同的主机上:

- a. 若已配置分布式时间服务 (DTS), 使用下列命令:

```
mkdce -n cellName -s secSrvHost -c CDSHost sec_cl cds_cl dts_cl
```

若未配置分布式时间服务 (DTS), 则使用下列命令,

```
mkdce -n cellName -s secSrvHost -c CDSHost sec_cl cds_cl
```

- b. 运行 **cicssetupclients** 设置 DCE 环境和密钥表文件。

设置客户的语言

使用本过程指定希望客户用于发出信息的语言。

从区域和客户发出客户信息, 因此:

- 验证在每个有客户的主机上安装了适当的信息目录。例如, 若客户的一个用户需求英语, 客户的另一个用户需求日语, 则验证英语和日语是否都安装了。检查 `/usr/lpp/cics/msg` 目录。
- 一些客户信息从区域发出。验证正确建立了区域, 所以它能够用客户需求的语言将信息发送到客户。参阅第34页的『设置区域的语言』。

需要指定客户语言:

1. 将下列添加到 `/etc/environment` 中的 `NLSPATH`:

```
/usr/lib/nls/msg/%L/%N:/usr/lib/nls/msg/en_US/%N
```

2. 将最终用户环境中的 `LANG` 环境变量设置为该用户所需求的语言。

用 CICS 使用 telnet 客户

CICS telnet 服务器 (**cicsteld**) 允许具有仿真 IBM 3270 信息显示系统的 telnet 客户连接到 CICS 区域。这些 telnet 客户类型在一些操作系统上可用, 其中包括;

- AIX
- Solaris
- HP-UX
- Digital UNIX
- MVS
- OS/2

- PC-DOS
- SINIX
- VM
- Windows NT

需要使用 Telnet 连接到 CICS 区域，则 Telnet 客户必须提供 **tn3270** 协议。参考 Telnet 客户文档，以获得更多信息。

Telnet 客户和 **cicsteld**

telnet 客户贮留在客户的系统上，接收输入，并且与目的地机器上的 **cicsteld** 通信。然后，telnet 服务器与 CICS 区域通信。

必须为每个 telnet 客户和 CICS 区域之间的连接启动 **cicsteld**（telnet 服务器）实例。可通过三种方法中的任何一种来完成：

- 隐式地，作为 inetd 精灵程序的子女（建议于产品系统）。参阅第75页的『使用 inetd 精灵程序启动 **cicsteld**』。
- 显式地，作为操作系统进程，如从操作系统命令行使用 **cicsteld** 命令。该方法对于测试有利。参阅第78页的『从操作系统命令行启动 **cicsteld**』。
- 使用配置 inetd 精灵程序的 CICS 控制程序。参阅第79页的『使用 **cicscp** 启动 **cicsteld**』。

一旦 CICS telnet 服务器（**cicsteld**）可用，则用户需要运行 telnet 客户访问 CICS。当选定了一个区域时，**cicsteld** 连接到该区域，并且请求终端自动安装。然后，用户可运行 CICS 事务。若其无法连接到区域，则 **cicsteld** 将一条信息返回到 telnet 客户。

使用 3270 telnet 程序的 CICS 连接将非 CICS 产品作为终端程序使用。重要的是，该产品提供了 CICS 事务所必需的数据流支持层。许多 3270 telnet 客户不支持所有能在 BMS 中指定的属性。3270 密钥的密钥映象是单独客户特别关心的。telnet 客户的用户手册应该能够说明如何定制密钥。

注：若 ATI 事务将要启动，则键盘继续锁定。当且仅当 ATI 事务启动并发出终端请求，（如 SEND），键盘才解锁。

cicsteld 与 **cicsterm** 的行为有何种差异

cicsterm 和 **cicsteld** 为两个截然不同的目的服务。**cicsterm** 是 3270 终端仿真器，它特别为能与 CICS 更好地工作而设计。**cicsteld** 是 3270 telnet 服务器，它允许使用一个类属 3270 telnet 客户连接到 CICS 区域。下列是主要差别：

- **cicsterm** 执行终端仿真，并且负责在屏幕上显示 3270 数据。**cicsteld** 不执行仿真--它将数据转换为 3270 telnet 协议，并将其发送到 telnet 客户显示。
- **cicsterm** 使用 3270 ASCII 数据工作。例如，**cicsterm** 过程用于 DBCS 和 MBCS 场所能够显示大小写单字节和多字节字符。**cicsteld** 通过 EBCDIC 3270 数据的方式与 3270 telnet 客户通信，因为这是 3270 telnet 协议必需的格式。这种情况下，由于缺少 EBCDIC 代码页，则不可能将小写单字节字符显示在多字节场所中。
- **cicsterm** 内的 3270 仿真器能够识别并显示所有 BMS 设施支持的属性。虽然是相同的数据流，但是在 EBCDIC 中，用 **cicsteld** 将它发送至 telnet 客户，而最终显示则受限于正在使用的 telnet 客户功能。

使用 ATI 事务的 **cicsterm** 和 **cicsteld** 行为差异

在事务结束时，**cicsteld** 和 **cicsterm** 都隐含地包括键盘的隐式释放。若该事务在将要结束之前启动了另一个事务，则在 ATI 事务启动前会有一个短暂的延迟。对于 **cicsterm**，将锁定键盘，然后可安排事务。对于 **cicsteld** 而言，能够预定事务，但是，没有一种 3270 数据流的形式能够发送用来锁定 3270 仿真器末端上的 **cicsteld** 键盘。键盘仍然未锁定，直至启动了新事务。

参阅 *CICS 应用程序设计指南* 中的“任务初始化”，以获得更多信息。

当从 **inetd** 精灵程序启动时，**cicsteld** 不工作

若 **cicsteld** 未从 **inetd** 精灵程序启动，则检查是否选定了一个不使用的端口侦听 **cicsteld** 连接。该端口指定在 `/etc/services` 中。然后，检查 `/etc/inetd.conf` 是否包含适当行，当在本端口接收到连接请求时，能够启动 **cicsteld** 精灵程序。确保 **cicsteld** 过程启动所使用的 UID 拥有阅读 `/var/cics_clients` 中的客户密钥表文件的许可（若正在使用单元目录服务（CDS）和 DCE 安全性服务）。

更改 **inetd.conf** 文件后，需要刷新 **inetd** 精灵程序。参考标题为 **inetd.conf** 中的指示，以获得如何刷新 **inetd** 精灵程序的信息。刷新过程取决于精灵程序初始启动的方式。

cicsteld.sh 外壳原本作为定制 **cicsteld** 调用的基础提供。注意，最多可在 **inetd.conf** 中指定五个参数，多余的参数将被删除。若需要更多 **cicsteld** 的参数（例如，服务器和客户代码页信息），则必须使用外壳原本。诊断跟踪也可写入来自 **cicsteld.sh** 脚本内的文件，以确认 **cicsteld** 传递正确的参数，并正在适当的环境中运行。

— 相关信息 —

需要有关建立 **cicsteld** 的安全性蕴含式的信息，请参阅第208页的『Telnet 客户的安全性考虑』。

第189页的『控制至 RPC 侦听程序进程请求的速率』。

CICS 管理参考大全 中的“**cicsteld** - 将 telnet 客户程序连接至区域”和“**cicscp** - telnet 服务器命令”。

使用 **inetd** 精灵程序启动 **cicsteld**

可在特定端口将 **inetd** 精灵程序连续配置为“侦听”，以便使 telnet 客户能够尝试连接到 CICS 区域。**inetd** 无论何时检测出一个连接尝试，它都将创建一例 **cicsteld** 实例。

若区域定义在 DCE 单元内，则在使用本过程前必须运行 **cicssetupclients**。若 DCE 和 CICS 客户机如第251页的『附录C. 使用 DCE』中那样建立，则本命令将自动运行。

1. 当从 **inetd** 运行 **cicsteld** 需要建立正确的环境时，需要一个外壳原本。脚本样本如下提供：

```
/usr/lpp/cics/etc/cicsteld.sh
```

将样本从 /usr/lpp/cics/etc 复制到 /usr/lpp/cics/bin。检查 NLSPATH 和 LANG 的值对于系统是否是正确的。

2. 若环境为安全性使用 DCE，则当 **cicsteld** 服务器启动时，需要 DCE 委托人。（参阅 第208页的『Telnet 客户的安全性考虑』。）若 DCE 委托人未已经存在，则使用 dcecp 创建一个。下列范例使用 rgy_edit。（需要更多有关 dcecp 的信息，请参阅 AIX 文档。）

```
rgy_edit
rgy_edit=> ktadd -p principal -pw password
rgy_edit=> quit
```

注：DCE 委托人的口令应该与其帐户的口令相同。

若没有超级用户权限，或若未使用系统设定密钥表文件，则可在本地主机 dcecp 上创建自用密钥表文件以及自己的委托人标识符。下列范例使用 rgy_edit:

```
rgy_edit
rgy_edit=> ktadd -p principal -pw password -f keytab
rgy_edit=> quit
```

若区域不为安全性使用 DCE，则不需要密钥表文件。

若环境未使用 DCE 安全性，则可使用 **cicsteld -t initialTransaction** 选项指定一个在自动安装后初始连接时运行的事务，如强制用户注册的 CESN。参阅第208页的『Telnet 客户的安全性考虑』，以获得更多信息。

3. 需要配置 inetd，则必须作为超级用户注册。inetd 需要知道要侦听的端口号、当连接请求到达时需要运行的程序、以及任何传递到程序的任选参数；您通过编辑 /etc/services 文件和 /etc/inetd.conf 文件来提供该信息。

/etc/services 文件是一张表，其中包含服务名项、TCP/IP 端口号和使用的 TCP/IP 连接类型。/etc/services 中的项目格式是：

serviceName portNumber/protocol

其中：

serviceName

必须在 /etc/services 文件内是唯一的；不允许重复的服务名。必须与 /etc/inetd.conf 文件中指定的服务名匹配。

portNumber

端口号，inetd 在该端口号侦听，并通过其连接。端口号的范围可从 1 到 65535，并且必须是唯一的。

protocol

可为 **tcp**（对于面向连接的协议），或 **udp**（对于面向数据报的协议）。需要连接到 CICS，必须指定 **tcp**。

/etc/inetd.conf 文件是 inetd 精灵程序的配置文件。它包含下列信息：

.serviceName. .dataStream. .protocol. .wait. .userid. .program. .parameters.

其中：

serviceName

必须在 /etc/inetd.conf 文件中唯一。必须与 /etc/services 文件中的服务名匹配。

dataStream

可为面向流或面向数据报。对于 **cicsteld**，这必须为 **stream**。

protocol

可为 **tcp**（对于面向连接的协议），或 **udp**（对于面向数据报的协议）。需要连接到 CICS，必须指定 **tcp**。

wait 控制允许到同一端口号的串行化连接。指定 **wait**，以将请求串行化，或 **nowait**，以允许多个并行的请求。

userid **cicsteld** 作为运行的操作系统用户标识符。若区域使用 DCE 安全性，则本用户标识符应该为对键表文件有访问权限的一个，该密钥表文件用 **cicsteld -k** 参数指定，或若未指定 **-k** 则用 **/krb5/v5srvtab** 指定。

program

当连接请求到达时要运行的程序。必须指定程序的完整路径名，就是，**/usr/lpp/cics/etc/cicsteld.sh**

parameters

可使用其来指定 **cicsteld** 的命令参数，该命令参数用来传递到上述 *.program* 中指定的程序。所指定的第一个参数应该为可执行名，就是，**cicsteld.sh**。参阅 *CICS 管理参考大全* 中的“**cicsteld** - 将 telnet 客户程序连接至区域”，以获得更多有关 **cicsteld** 命令参数的信息。

当将新项目添加到该文件时，inetd 必须读新版本。若“网络信息服务器”（NIS）正在使用之中，则 **/etc/services** 文件必须在 NIS 主机上编辑，并且执行 **ypmake** 命令，以使修订副本在客户机上可用。

使用 **ps** 命令查找 inetd 的进程标识，然后输入：

```
kill -HUP .pid.
```

或

```
refresh -s inetd
```

在 **/etc/services** 文件中的例项是：

```
cicsteld 6192/tcp  
payroll 6193/tcp
```

例如，需要在区域 **regionA** 启动一个运行工资单事务的终端，可将下列项置于 **/etc/inetd.conf** 文件内：

```
cicsteld stream tcp nowait cics /usr/lpp/cics/etc/cicsteld.sh
```

```
cicsteld.sh -p principal payroll stream tcp nowait \  
cics /usr/lpp/cics/etc/cicsteld.sh
```

```
cicsteld.sh -t pay1 -p principal -r regionA
```

在本例中，**payroll** 是服务名，**pay1** 是要运行的 CICS 事务。若未使用 DCE 安全性，**-p .principal.** 将被忽略。但是，**-t .initialTransaction.** 可用于指定在自动安装后的初始连接时间运行的事务，如强制用户注册的 **CESN**。

参阅第208页的『Telnet 客户的安全性考虑』，以获得更多信息。

从操作系统命令行启动 **cicsteld**

当 **cicsteld** 启动时，以下列两种方法中的一种指定分配给客户连接附件的端口：

- 在 **cicsteld** 命令上使用 **-P** 选项。
- 在 **cicsteld** 的 `/etc/services` 数据库中。然后，操作系统能够查询该文件，并且尝试分配端口。

然后，**cicsteld** 进程在那个端口上侦听，侦听单个进入的 telnet 客户连接附件请求。

使用下列步骤，以从操作系统命令行启动 **cicsteld**。在开始前，确认包含 **cicsteld** 程序 (`/usr/lpp/cics/bin`) 的目录在路径中。

步骤 1. 必需的话，建立 DCE 安全性

若环境使用 DCE 安全性，则 **cicsteld** 服务器启动需要 DCE 委托人。（参阅第208页的『Telnet 客户的安全性考虑』。）若不存在现存的一个，则使用 **dcecp** 创建一个。下列范例使用 **rgy_edit**。（需要有关 **dcecp** 的信息，请参阅 AIX DCE 文档。）

```
rgy_edit
rgy_edit=> ktadd -p principal -pw password
rgy_edit=> quit
```

若没有超级用户权限，或若未使用系统设定密钥表文件，则可在本地主机 **dcecp** 上创建自用密钥表文件，以及自己的委托人标识符。下列范例使用 **rgy_edit**：

```
rgy_edit
rgy_edit=> ktadd -p principal -pw password -f keytab
rgy_edit=> quit
```

若区域不使用 DCE 安全性，则不需要密钥表文件。

若环境未使用 DCE 安全性，则可使用 **cicsteld -t cicsteld -t initialTransaction** 选项指定一个在自动安装后初始连接时运行的事务，如强制用户注册的 CESN。参阅第208页的『Telnet 客户的安全性考虑』，以获得更多信息。

步骤 2. 选择有效的端口号

请勿使用在 `/etc/services` 文件中指定的端口号。

步骤 3. 启动 **cicsteld**

使用 **cicsteld** 命令，如下所示：

```
cicsteld -P portNumber -p principalName -r regionName &
```

portNumber

为数字端口号，**cicsteld** 在其上侦听进入 telnet 客户请求。若端口号是无效的，则 **cicsteld** 发出一条说明信息。若未指定端口号，则 **cicsteld** 从 `/etc/services` 文件中 **cicsteld** 的服务名项取得端口号。

principalName

为 DCE 委托人，**cicsteld** 使用它来认证。若环境未使用 DCE 安全性，则它不是必需的。

regionName

cicsteld 连接到的区域。若未指定区域，则 telnet 客户有一条菜单，包含用户可从其连接的可用区域。

若接收到信息 ERZ018001E/0003，告之所选定的端口已经被另一个应用程序使用，则应该用不同的端口号再试一次。

使用 **cicscp** 启动 **cicsteld**

可使用 **cicscp create telnet_server** 命令创建可用 **inetd** 使用的 **telnet** 服务器配置。对于每个由包含 **cicsteld** 命令变量的 **cicscp** 所创建的 **telnet** 服务器，在 **/var/cics_clients** 中自动创建外壳原本。

对于使用 **DCE** 为安全性的环境，则需要委托人和密钥表文件。这将用 **-p principalName** 指定。若未指定委托人，则 **cicscp** 用与服务器相同的名称创建一个。使用的委托人自动地添加到 **cics_users** **DCE** 组。若未指定密钥表文件，则 **cicscp** 在 **/var/cics_clients** 中创建一个。

若区域不使用 **DCE** 安全性，则密钥表文件或委托人都将不会创建。但是，可使用 **-t transId** 选项指定要在自动安装后的初始连接时间运行的事务，如强制用户注册的 **CESN**。参阅第208页的『**Telnet** 客户的安全性考虑』，以获得更多信息。

-m 标志允许从用 **-m** 指定的 **telnet** 服务器复制所有参数值（除了必须唯一的网络名）。若 **cicscp** 未创建服务器，或在创建服务器后手工编辑外壳原本，则 **cicscp** 不可能从模型服务器检索参数值。

使用 **telnet** 客户连接到区域

从安装了 **telnet** 客户的系统内调用指定了主机名和端口号的 **telnet** 客户。

一个面板提示选择可能显示的区域，或可能已经连接到在 **CICS telnet** 服务器建立时已经预确定的一个区域。

若显示了一个面板，则使用 **F7** 和 **F8** 键滚动，然后将光标移动到适当的屏幕行并按 **Enter** 键，选择要连接到的区域。选择了区域后，**cicsteld** 连接到区域，并请求终端的自动安装。可用当使用 **cicsterm** 时的相同方法与 **CICS** 交互。

需要取消会话而不需要连接到区域时，按 **F3** 键。

可能需要注册到区域。若是这样，使用 **CESN** 事务。需要从 **CICS** 区域注销，使用 **CSSF** 事务。

破坏 **telnet** 服务器配置

cicscp destroy telnet_server 命令破坏 **telnet** 服务器配置，但是，不删除 **DCE** 委托人。

参阅 *CICS 管理参考大全* 中的“**cicscp - telnet** 服务器命令”，以获得更多信息。

IBM cics 客户机产品

本部分描述如何使用带有 开放系统上的 **CICS** 区域的 **IBM cics** 客户机产品。

侦听程序进程如何工作

所有开放系统上的 CICS 区域通过侦听程序进程支持来自 IBM cics 客户机产品的请求。若存在 **Protocol** 属性设置为 **TCP** 的侦听程序定义 (LD) 项, 则该多线程进程在区域启动时启动。需要配置侦听程序进程, 转至第82页的『配置 IBM cics 客户机产品的区域』。

监听程序进程与客户机上的相应进程进行通信。这些进程间的通信模拟 CICS 域之间的流。例如, 由 EPI 调用生成的流程与参与事务途程作业的 CICS 区域之间所传递的那些流程相同。由 ECI 请求生成的流程与 CICS DPL 请求所生成的那些 CICS 流程相同。因此, 客户进程以受限的终端域出现在服务器中。

由于数据流与 CICS 系统之间的那些相同, 所以服务器需要客户系统的通信定义 (CD) 项。

当客户进程在客户上启动时, 将启动初始化事务 (CCIN) 的流程发送到服务器。本事务负责 CD 中的客户系统自动安装。(参阅第115页的『使用 IBM cics 客户机产品的自动安装程序』。)

硬编码 IBM cics 客户机产品的客户所使用的 CD 项是不可能的。然而, 若 CICS 客户机, 或从正在运行 CICS 客户程序的代码页支持的数据转换于 ISO8859-1 代码页不同, 则需要更改系统设定的 CD 项中的 RemoteCodePage TR 值。这适用于所有 DBCS/MBCS 环境。

当 IBM cics 客户机产品 EPI 应用程序连接到区域, 则 CICS 执行自动安装, 其结果为定义格式 =xxx 的终端, 其中 xxx 是 CICS 选择的字符串。本术语与开放系统上的 CICS 客户的自动安装所使用的不同, 后者根据客户系统的前两个字符生成终端 ID。对于那些写其自身的自动安装用户出口程序和那些不希望 = 字符的客户, 这将引起一些问题。

必须与服务器定义匹配的客户定义

客户配置通过修改 CICSCLI.INI 文件执行。下列项很重要:

Protocol

设置为 TCPIP。

NetName

为网络上的服务器设备名。通常, 其形式为 *machine.company.com*, 但是也可能是 Internet 地址。

Port 服务器在上侦听的端口标识符。正在使用系统设定端口 (1435), 则该项可设置为 **0**, 这是系统设定值。若服务器正在另一个端口侦听, 则将替代此值。

UpperCaseSecurity

本项应该从系统设定值 **Y** 更改为 **N**。许多系统对用户标识符和口令使用混合大小写的字符, 在此输入 **Y** 将引起认证失败。

InitialTransid

由客户 cicsterm 运行的第一个事务。可将其设置为用户事务, 或类似 CESN 的事务。

注:

1. 当写 ECI 程序时, 指定在初始化 (eci_transid) 后由服务器运行的第一个事务名是可能的。该事务的系统设定是 CICS 镜像事务 (CPMI)。若应用程序指定了不同的事务, 则由指定的事务所指向的后端程序必须为 DFHMIRS 程序。
2. 为了能够正确地处理 DBCS/MBCS 环境, 需要将 **ModelTerm** 参数指定为 **hft-mb**、**mft-mb** 或 **lft-mb** 中的一项, 否则将要安装的系统设定终端类型将不支持 DBCS/MBCS。

相关信息

CICS 内部通信指南 中的“配置 CICS 系列 TCP/IP 的 LD 输入项”。

CICS 管理参考大全 中的“侦听程序定义 (LD)”。

配置限制

一个操作系统进程能够一次最多有 2000 个打开文件描述符。这在正常情况下不会成为问题, 但是对于侦听程序进程则可能会有麻烦。每次一个客户程序系统连接到服务器系统, 启动线程处理分配的系统端口。该端口映射使用文件描述符。因此, 单个侦听程序进程不能处理大约 1950 以上的客户程序 (允许为其它打开文件浪费 50 个)。您能够:

- 将一块新适配卡添加到机器。

这种方式具有一大优点, 端口地址对于所有客户程序系统都是相同的, 但是意味着服务器 Internet 地址是不同的。

若有 2000 个客户连接到服务器, 则可能发生通过添加新适配卡递增适配器带宽的例子。

这可通过配置两个 LD 执行, 一个在网络适配器上专用侦听, 第二个在其它侦听。在 LD 中的 **TCPAddress** 属性用于该过程。

- 使用第二个端口。

本方法具有一条优点, 服务器设备的 Internet 地址仍是相同的, 但是端口地址是不同的。

若附加适配器的代价不是有保证的, 或适配器性能不保护系统性能, 则可使用该选项。LD **TCPAddress** 属性留为空白, 并且显示定义端口地址。

ECI 和 EPI 客户应用程序

开放系统上的 CICS 客户和 IBM cics 客户机产品使用非 CICS 应用程序设计接口, 它提供对 CICS 设施的外部访问。那些接口为:

- 外部调用接口 (ECI)
- 外部呈现接口 (EPI)

下列是 ECI 和 EPI 应用程序需要用来访问区域的配置:

1. 若与来自客户的连接请求一起传递的用户标识符未定义到区域, 则用户使用区域的系统设定用户标识符注册。若该用户标识符没有足够的访问特权, 则可使用 **cicsadd**

-c ud 为传递的用户标识符建立用户定义 (UD)。本命令用于正在运行区域的机器。参阅 *快速入门* 中的“为用户添加用户定义 (UD)”。

2. 未使用 DCE 安全性服务的开放系统上的 CICS 客户和单元目录服务 (CDS) 必须使主机名对于 CICS 客户程序软件可用。参阅 *快速入门*。
3. 开放系统上的 CICS 客户使用 DCE RPC 与区域通信。但是, IBM cics 客户机不是这样。正因为如此, 区域必须将来自 IBM cics 客户机产品的请求配置为『侦听』。cics 客户机产品是:
 - IBM CICS 客户 OS/2 版
 - IBM CICS 客户 DOS 版
 - IBM CICS 客户 Windows 3.1 版
 - IBM CICS 客户 Windows NT 版
 - IBM CICS 客户 Windows 95 版
 - IBM CICS 客户 Macintosh 版

当 IBM cics 客户机产品第一次请求与 CICS 区域通信时, 自动安装通信定义 (CD) 项。这在第115页的『使用 IBM cics 客户机产品的自动安装程序』中描述。

参阅 *CICS 内部通信指南*, 以获得更多信息。

配置 IBM cics 客户机产品的区域

若 IBM cics 客户机产品从区域请求服务, 则需求 **Protocol** 属性设置为 **TCP** 的侦听程序定义 (LD) 项:

需要配置 IBM cics 客户机产品的 侦听程序定义 (LD):

1. 使用 **cicsadd** 命令添加一个 LD 项。在下列范例中, 一个 LD 项通过协议 TCP, 添加到识别为 CLIENTS 的 regionA 的永久数据库中。将所有其它 LD 属性的系统设定值假设为:

```
cicsadd -c ld -r regionA -P CLIENTS Protocol=TCP
```

2. 若正在使用一个非标准的端口, 则将一项添加到 `/etc/services` 文件中。
3. 添加这项后, 刷新 `inetd` 精灵程序。
4. 重新启动 CICS 系统。这时将启动侦听程序进程。

需要使用 SMIT 为区域定义资源, 参阅 第10页的『使用 SMIT 建立 CICS 系统』。

用于 IBM cics 客户机产品的 LD 属性为:

<Key> 标识, 当系统引用为支持该定义而运行的侦听程序时而使用。

Protocol

对于 IBM cics 客户机产品支持, 这项设置为 **TCP**。

TCPAddress

机器上适配器的 Internet 地址 (运行侦听程序进程的那一个)。它告知侦听程序进程在特定的适配器上侦听。客户系统在所有网络适配器上的系统设定侦听。

TCPService

若留为空白, 则所使用的端口是 CICS 客户的系统设定端口。IBM 已经为

此目的用 **California** 的 **Internet** 指定数字权限 (IANA) 注册端口 **1435**。因此, 若该选项留为空白, 则 CICS 在区域启动时启动端口 1435 上的侦听程序进程。

若在这里输入名称, 则其应该与 `/etc/services` 文件中的项匹配。例如, 若希望侦听程序进程在端口 6789 上进行侦听, 则在该字段输入名称 **cicsUser**。在 `/etc/services` 文件中, 必须包含下列行:

```
cicsUser      6789/tcp      #CICS Client Listener
```

需要一套完整的有关所有可用于侦听程序进程的属性说明, 请参阅 *CICS 管理参考大全* 中的“侦听程序定义 (LD)”。

将新用户添加到区域

CICS 用户是使用本章的客户描述连接到 CICS 区域的用户。若用户没有用户定义 (UD) 项, 则用户用区域的系统设定用户标识符连接到区域 (参阅第233页的『创建区域的缺省用户标识符』)。

关于添加用户

用户的各种属性在 *CICS 管理参考大全* 中的“用户定义 (UD)”描述。特别重要的属性如下所示:

- **Principal**, 包含用户的 DCE 委托人。若区域配置为 **AuthenticationService=DCE**, 则将使用该委托人。
- **RSLKeyList**, 包含了该用户资源安全性密钥的列表。
- **TSLKeyList**, 包含了该用户事务安全性密钥的列表。
- **CICSPassword**, 包含了当区域用 **AuthenticationService=CICS** 配置时, 用于口令检查的口令。该口令由 CICS 设施加密, 如 **cicsadd**。

更改 CICS 口令

使用本过程更改 CICS 口令。

需要使用本过程, 必须首先执行下列:

- 区域必须定义为使用 CICS 授权 (**AuthenticationService=CICS**)。
 - 必须已经定义用户定义 (UD)。
 - 若不拥有 CICS 管理权限, 则需要知道当前的口令。
1. 连接到使用 **cicsterm** 或 **cicsteld** 的区域。
 2. 启动 CESN 事务。
 3. 当 CESN - CICS SIGNON 面板显示时, 键入 CICS 用户标识符、当前口令和新口令。参阅第84页的图 3:

CESN - CICS SIGNON APPLID REGIONA

输入用户标识符和口令，然后按 ENTER:

用户标识符 SALLY

口令 tempPass

新口令 ORZ87TU

请输入用户标识符。

F3=退出

图 3. 使用 CESN 更改 CICS 口令

这将 UD 中的 SALLY 的 **CICSPassword** 属性更改为 **ORZ87TU** 的加密版本，给出当前值是 **tempPass** 的加密版本。参阅 *CICS 管理参考大全* 中的“CESN (注册)”，以获得更多关于使用 CESN 的信息。

若拥有 CICS 管理权限，则能够使用 **cicsupdate** 命令更改 **CICSPassword** 属性。在这种情况下，不需要旧口令。例如：

```
cicsupdate -c ud -r regionA -B SALLY \
          CICSPassword=ORZ87TU
```

设置 UD SALLY 的 **CICSPassword** 属性为 **ORZ87TU** 的加密版本（参阅 *CICS 管理参考大全* 中的“用户定义 (UD)”）。

如何为从 CICS 应用程序打印而建立 **cicstermp**

本部分描述了如何使用 **cicstermp** 进程，以允许从 CICS 应用程序打印。

cicstermp 如何工作

cicstermp 用于启动 CICS 打印机仿真器，如 *CICS 管理参考大全* 中的“**cicstermp** - 打印机仿真”所描述。与打印机设备关联的 CICS 3270 终端仿真器等同于一般的用户 CICS 3270 终端仿真器，（除了打印机 CICS 3270 终端仿真器不连接到交互式设备），所以它们不能接收键盘输入。

当在发送到终端的数据流中检测出打印控制时，打印命令（如，由 **cicstermp -P** 选项或其系统设定值指定）用于处理数据。数据置于临时文件，打印命令与作为参数附加的临时文件一起发出。然后，擦除临时文件。

注意，不要求存在确切的打印命令打印数据。例如，只是能够复制数据，以便使之合并入文档。还请注意，因为临时文件是立即擦除的，所以打印命令必须在返回前完成提供的文件副本。这预先避免了一些命令的使用，例如 **lpr**（不带 **-c** 选项），这些命令仅简单地记录以后处理的文件名。

那些仅与 PostScript 打印机类型（**-f**、**-M**、**-s** 和 **-R**）相关联的参数必须是有效值。否则，可能发生不正确的打印机输出。若当未使用 PostScript 打印机类型时，使用了这些参数中的任何一个，则发出一条错误信息，且 **cicstermp** 终止。

若对 **cicstermp** 产生问题

当正在使用 CICS 打印机仿真器时，请使用下列信息帮助您解决问题：

cicstermp 不工作

确保：

- **cicstermp** 用 **-n netName** 项调用。
- 定义了一个终端定义（WD）项，其 **IsPrinter** 属性设置为 **yes**，**NetName** 属性设置为用 **cicstermp -n** 选项指定的网络名。
- **PRINT** 选项在 CICS 程序中的 **EXEC CICS SEND MAP** 语句中指定。
- 若 **CECI** 用于调用事务，则打印机的 **TERMINAL** 在 **CECI** 命令行上指定。

打印到本地打印队列时出错

当将一个打印作业排队到本地队列时，操作系统假设正在被打印的文件仍然在文件系统上直至打印完成。因为 **cicstermp** 在打印命令返回时删除临时文件，所以当打印到本地打印队列时，打印作业可能失败。需要解决这个问题，当在打印命令中使用“**lp**”或“**lpr**”时，请包括“-c”复制标志。

打印机能够跳出传递到 **cicstermp** 的代码吗？

cicstermp 仅接受 3270 数据，并且相应地将所有数据作为 3270 数据流的一部分解释。若打印机需求原始 ASCII 代码（例如初始化字符串），则该代码必须格式化为合法 3270 数据。然后能够用筛选程序对 **cicstermp** 进行后处理。

例如，需要将 X'1B431B45120E484F' 发送到打印机，则应该实际发送类似下列的内容：

```
C'PRINTERCONTROL1B431B45120E484FPRINTERCONTROLEND'
```

应该选择字符串 **PRINTERCONTROL**，以使其唯一。然后，筛选程序名可在 **cicstermp -P** 选项中指定。

遗憾的是，这些字符占用 3270 设备缓冲区中的空间，所以这仅在页面未完全满时才有可能。

启动 3270 打印机仿真器（**cicstermp**）

CICS 应用程序设计指南 中的“用 SDA 打印 BMS 映射信息”描述如何从 CICS 应用程序请求打印输出。本过程描述如何启动 CICS 3270 打印机仿真器（**cicstermp**），它与应用程序一起使用。必须在应用程序能够请求一个打印输出之前启动本过程。

CICS 从应用程序将打印机数据流路由到临时文件，然后，使用调用 **cicstermp** 时指定的操作系统命令来处理该临时文件。系统设定命令是 **lp -c**。

cicstermp 命令使用 3270 打印机数据流中的 **WCC** 字节来确定打印行的宽度。所有宽度（40、64、80 和 132）都支持。属性（如突出显示、下划线、高亮度和颜色）都被忽略。

需要启动 **cicstermp** 进程：

步骤 1.验证打印机的终端定义 (WD) 存在

打印机的 WD 项必须具有下列属性设置:

- **IsPrinter** 必须设置为 **yes**。
- **NetName** 必须设置为打印机的 ASCII 八字符标识。
- **NumColumns** 必须设置为 **132**。
- **NumLines** 必须设置为 **64**。

参阅 *CICS 管理参考大全* 中的“终端定义 (WD)”，以获得更多信息。

步骤 2.启动 **cicstermp**

区域启动后，应用程序需求打印机前，请使用下列命令:

```
cicstermp -n  
netName -P printCommand
```

其中，*netName* 是打印机标识符，*printCommand* 是希望 **cicstermp** 使用的打印命令。

参阅 *CICS 管理参考大全* 中的“cicstermp - 打印机仿真”，以获得更多信息。

第5章 为使用关系数据库配置 CICS

CICS 事务能够通过将 SQL 调用嵌入到 CICS 应用程序的主体内来访问关系数据库。包含 CICS 和 SQL 调用的协调落实和事务恢复仅在支持 X/Open XA 接口的关系数据库中有可能会。

注:

- 1. 该 CICS 发行本支持的数据库产品发行本描述在发行说明和 README 文件中。
- 2. 若要使用 COBOL, 验证 COBOL 产品以发行说明和 README 文件中指定的级别安装。

CICS 完成 X/Open DTP 模型中“事务处理管理器”(TM)的角色(参阅 X/Open 分布式事务处理参考模型 ISBN 1 872630 16 2)。它能够将分布式事务协调为支持 XA 接口的关系数据库。

XA 接口不是普通的应用程序设计接口(API), 而是在 X/Open DTP 模型中的事务处理管理器(TM)与资源管理器(RM)之间的系统层接口。然而, 应用程序员需要意识到在支持 XA 的环境中编码。需要更多有关在这种环境中的应用程序设计的信 息, 请参阅 CICS 应用程序设计指南 中的“一个 XA 启用关系数据库的事务范例”。

表 14. 关系数据库配置导向图

若需...	参考...
查找关于 CICS 提供的数据库样本程序。	第87页的『CICS 提供的数据库样本程序』。
查找关于用 CICS 使用关系数据库。	第87页的『用 CICS 使用关系数据库』。
查找关于 DB2 单阶段落实优化。	第88页的『单阶段落实优化』。
启用到 DATABASE2 的连接。	第90页的『启用到 DB2 的连接』。
启用到 Informix 的连接。	第93页的『启用到 Informix 的连接』。
启用到 Oracle 的连接。	第96页的『启用到 Oracle 的连接』。
启用到 Sybase 的连接。	第100页的『启用到 Sybase 的连接』。

CICS 提供的数据库样本程序

CICS 在 \$CICS/src/examples/xa (支持 XA) 中, 以及在 \$CICS/src/examples/nonxa (支持非 XA) 中提供样本程序。需要验证正确地建立了数据库, 极力推荐在产品环境中试图使用数据库之前使用这些样本程序。有关指示在 uxa1.README (对于支持 XA) 和 nxa1.README (对于支持非 XA) 中。

用 CICS 使用关系数据库

CICS 拥有高级基于装入体系结构, 在这种体系结构下, 事务码动态地装入长期运行应用程序服务器中。这允许将新事务添加到正在运行 CICS 的系统中, 简化了应用程序的维护, 因为在内部库的更改的情况下应用程序不需要重连接。

希望支持本体系结构, 则支持 XA 的关系数据库的供应商需要提供为与 CICS 使用的其库的共享可重入版本。这些库在供应商的特定文档中描述。另外, CICS 配置的数据库文档在第90页的『SQL 数据库设置过程』中提供。

对于在支持 XA 的环境中的备份与恢复，您必须估计支持 XA 的关系数据库的失败对 CICS 安装所造成的影响；

- 静态注册支持 XA 的关系数据库已经紧密地集成入 CICS 系统。区域内的所有 CICS 事务都调用 XA。若关系数据库遭到致命的故障（如磁盘外磁头碰撞或远程链路故障），CICS 则捕捉到该故障并立即异常终止该区域。在恢复数据库或区域将异常终止之前不能热启动区域。
- 动态地注册支持 XA 的关系数据库也已经紧密的与 CICS 系统集成。但是，只有在支持 XA 的数据库上调用 SQL 命令的事务才能调用 XA。若关系数据库遭到致命的故障，则 CICS 事务以针对支持 XA 的数据库故障的方式运行。CICS 捕捉数据库故障仅当新 CICS 应用程序服务器启动时。必须在数据库恢复后返回联机后关闭 CICS。若试图在恢复数据库前自动启动 CICS 区域，则异常终止区域。

因为这些原因，可考虑运行另一个独立于数据库交互的区域。然后，若 CICS 受到数据库关机的影响，则其它区域能够继续行使附带、事务路由选择和写入 SFS 以维护有限制 CICS 服务的职责。

若关系数据库可能在长时间内不可用，并且系统包括不调用关系数据库的事务，则能够执行 CICS 冷启动，但是在关系数据库失败时的任何进行中的事务都将丢失。若执行冷启动，则必须从 CICS 配置中暂时删除不可用的关系数据库的产品定义（XAD）。

若关系数据库完全恢复前一直在等待，则 CICS 热启动恢复自从关系数据库发生故障以来所有丢失的进行中的事务。

紧密集成的支持 XA 的关系数据库进一步暗示若区域无法打开一个到数据库的连接，则 CICS 将立即异常终止该区域。在这种情况下，将适当信息写入 CICS console.nnnnnn。

致命的诊断信息转发到 console.nnnnnn 文件和 symrecs 文件中的 CICS 系统管理员。因此，初始的数据库连接失败或随后的致命故障用该方式报告。有关失败性质的进一步信息在：

DB2 操作系统 syslog。

Sybase

XA 错误记录文件。

Informix

无更多信息。

Oracle7

XA 错误记录文件。

应用程序员负责捕获 SQL 错误。

单阶段落实优化

一些配置（如那些仅有一个数据库的配置）不需要由 XA 接口的两阶段落实协议提供的完全数据完整性。单阶段落实优化为数据完整性不成问题的 CICS 配置提供了更高性能的方案。

在看似并非数据库失败的事件发生时，CICS 从失败事务中将不一致数据的可能性汇报出来。然而，若在失败事务中牵涉了多个资源管理器，则管理员还是有必要检查数据的一致性。若数据库是唯一参与事务中的资源管理器，则数据将是一致的。若事务对

其它资源进行更改，如 CICS 队列、文件或其它 RDBMS 表，则数据将会产生不一致。在这种情况下，则需要干预，以便重同步化各种资源。

单阶段落实优化通过 CICS 区域适当产品定义 (XAD) 的配置生效。参阅适当的数据库部分，以获得更详尽的信息。

尤其需要引起注意的是 CICS 区域在单阶段落实优化下可最多支持一例任何特定的 RDBMS。而且，用其本身版本的单阶段落实来并存地支持一个支持 XA 的 RDBMS 是不可能的。

并存地支持其它支持 XA 的关系数据库是有可能的，但是不能保证支持 XA 的关系数据库和单阶段落实优化的 RDBMS 之间的完全数据完整性。

迁移考虑

对于带有支持 XA 的 RDBMS 的 CICS 当前用户而言，可通过更改其区域数据库中的产品定义 (XAD)，并执行一次冷启动，以便从 XA 环境移动到单阶段落实环境。不需要应用程序代码，也不需要重建该环境支持 XA 的应用程序代码。

希望从支持非 XA 的环境移动到单阶段落实环境的用户应该配置一个新产品定义 (XAD) 项，但是，需要重做应用程序代码，以使事务控制由 CICS 执行，而不是由 EXEC SQL COMMIT 或 ROLLBACK 语句。

应用程序开发

若希望使用单阶段落实优化开发 RDBMS 的新应用程序，则应按照 *CICS 应用程序设计指南* 中的开发支持 XA 的事务指南。

备份和恢复

使用单阶段落实优化的 RDBMS 将不参与 CICS 恢复工作。如上所述，在一些失败方案中，可能必需手工地用其它资源管理器使数据库重新同步化。

因为单阶段落实优化仍然是有效的 XA 实现，所以 RDBMS 单阶段落实数据库紧密地与 CICS 区域集成。所有区域内的 CICS 事务都将执行 SYNCPOINT 处理调用数据库。

若关系数据库遭到致命的故障（如磁盘外磁头碰撞），所有系统内的事务将失败，并且应关闭区域。在恢复数据库或区域将异常终止前不能热启动 CICS 区域。因为该原因，可考虑运行另一个带有支持非 XA 的数据库的区域。然后，若 CICS 必须关闭，则其它区域能够使用支持非 XA 的数据库继续执行附带、事务路由选择和写入 SFS。

若数据库可能在长时间内不可用，并且系统包括不调用数据库的事务，则可执行 CICS 冷启动，但是在关系数据库失败时的任何进行中的事务（会影响到其它资源）都将丢失。若执行这种冷启动，则必须从 CICS 配置中暂时删除 RDBMS 单阶段落实数据库的产品定义 (XAD)。

诊断问题

错误处理着重于基本 SQL CONNECT、COMMIT 和 ROLLBACK 错误代码的传播，以及通过系统控制台和 symrecs（症状记录文件）到管理员的错误信息。

问题诊断需要 RDBMS 问题确定文档和技术。有关失败性质的进一步信息可能在 RDBMS - 特定错误记录中。参阅相关的 RDBMS 实现过程，以获得更详尽的细节。

SQL 数据库设置过程

本部分包含建立 CICS 和关系数据库管理器之间接口的过程。

表 15. SQL 数据库设置过程的导向图

若需...	参考...
启用到 DB2 的连接。	第90页的『启用到 DB2 的连接』。
启用到 Informix 的连接。	第93页的『启用到 Informix 的连接』。
启用到 Oracle 的连接。	第96页的『启用到 Oracle 的连接』。
生效到 Sybase 11 的连接。	第100页的『启用到 Sybase 的连接』。

启用到 DB2 的连接

本过程描述了如何启用区域连接到 DB2 数据库。描述了 XA、非 XA、两阶段落实和单阶段落实连接，以及 Micro Focus COBOL 运行时间需求。

前提任务和条件:

- 应运行 **db2ln** 命令连接 DB2 库，并将目录包括到 /usr/lib and /usr/include。验证已运行该命令。参考 **DB2: 命令说明**。
- 必须定义 DB2 实例和数据库。
- 若使用 DB2 认证服务，并且不希望提示口令，则将 CICSDB2PASSWORD 环境变量设置为数据库用户标识符口令。
- shr.o 共享对象从 CICS 应用程序的 libdb2.a 档案库中抽取出来。需要抽取本共享对象:

```
cd $DB2DIR/lib
ar -vx libdb2.a
mv shr.o db2.o
```

然后，将共享对象连接到 /usr/lib，如下所示:

```
ln -s $DB2DIR/lib/db2.o /usr/lib/db2.o
```

- 用户名超级用户和 cics 必须在 DB2 admin 组中。
- 用户超级用户需要 DB2 环境执行下列过程。

过程:

本部分描述如何建立环境，以从支持 XA 和支持非 XA 的 CICS 中启用 'C' 和 COBOL 事务。 第90页的表 16 描述了应该为每个事务环境使用何种过程步骤。

表 16. 每个事务环境必需的过程步骤

非 XA 'C'	XA 'C'	非 XA COBOL	XA COBOL
步骤 1	步骤 1	步骤 1	步骤 1
		步骤 2	步骤 2
	步骤 3		步骤 3
	步骤 4		步骤 4

表 16. 每个事务环境必需的过程步骤 (续)

	步骤 5		步骤 5
	步骤 6		步骤 6
步骤 7	步骤 7	步骤 7	

1. 步骤 1. 建立 DB2 的区域环境

区域的环境是当区域启动时从超级用户的外壳继承而来的。可将数据库环境信息添加到区域的环境文件（/var/cics_regions/*regionName*/environment），以避免若区域从外壳启动时会发生的连接错误，这些外壳没有必需的数据库环境变量设置，如：

至少，区域的环境文件必须包括：

```
INSTHOME=DB2instanceHomeDirectory
DB2INSTANCE=DB2instanceName
```

当将库路径添加到区域的环境文件中时，保证包括了所有在本区域运行的应用程序必需的库。原因是当区域启动时存在的库路径值将被区域环境文件中的库路径值替代。这对于所有值都是正确的，如 PATH、NLSPATH、LANG 等。

并且，请勿在区域的环境文件中使用 \$ 值，因为这些值是不能扩充的。参考区域的环境文件前言中的信息。

2. 步骤 2. 创建 Micro Focus COBOL 运行时间

COBOL 运行时间必须包括所有在区域上运行的 CICS 应用程序所使用的产品的特定产品库和目标文件。cicsmkcobol 命令包括 Encina、DCE 和 CICS 库，以及 CICS 必需的目标文件。需要添加附加的产品库，则将其附加在 cicsmkcobol 命令中。例如，需要在 COBOL 运行时间中包括 DB2 库，则下列附加到 cicsmkcobol 命令中：

```
-L$DB2DIR/lib -ldb2 $DB2DIR/lib/sqlgmf.o
```

其中，\$DB2DIR 是 DB2 产品目录。

注：sqlgmf.o 对象必须首先从 libdb2gmf.a 档案库中抽取出来。需要执行这些：

```
cd $DB2DIR/lib
ar -vx libdb2gmf.a
```

然后，通过使用下列命令，将目标文件连接到 /usr/lib:

```
ln -s $DB2DIR/lib/sqlgmf.o /usr/lib/sqlgmf.o
```

需要建立用于所有区域的全系统 COBOL 运行时间，从 \$CICS/bin 目录发出命令。需要建立仅用于区域的 COBOL 运行时间，从 /var/cics_regions/*regionName*/bin 目录发出命令。

3. 步骤 3. 建立交换装入文件（可选）

XA 接口定义一个结构，该结构列出资源管理器（RM）中实现的 xa_ 例行程序。在 CICS 中，本结构称为交换装入文件。

交换装入文件与 CICS 一起提供，并建议使用提供的交换装入文件，如步骤 4 中所描述。但是，若由于 DB2 库中的更改（如应用 DB2 维护升级）引起库或符号访问出错，则请使用下列指示，重建交换装入文件：

- 转至 CICS 目录，该目录包含希望建立的交换装入文件的源和实用程序生成文件。例如：

```
$CICS/src/examples/xa - 对于两阶段落实
$CICS/src/examples/onephase - 对于单阶段落实
```

- b. 将 DB2 交换装入文件源和实用程序生成文件复制到区域的 bin 目录中。例如:

```
cp db2xa.* /var/cics_regions/regionName/bin
```

- c. 转至区域的 bin 目录, 并编辑实用程序生成文件, 以验证 DB2DIR 的值设置为 DB2 产品目录。

- d. 使用 make 编译交换装入文件, 如下所示:

```
make -f db2xa
-或-
make -f db21pc
```

以便了解使用单阶段落实连接的完全含义。

4. 步骤 4. 添加产品定义 (XAD)

CICS XAD 是为 CICS 定义资源管理器 (RM) 的工具。为了 CICS 能够与数据库 RM 集成, 必须提供开放字符串和交换装入文件名。下列 XAD 属性用于该目的:

XAOpen

定义作为变量传递到 xa_open() 接口例行程序的文本。至少, 必须提供数据库名, 否则使用系统设定用户名 cics。

DB2 中的 xa_open 字符串使用下列格式:

```
databaseName[,userName,password]
```

- *databaseName* 是数据库名。若创建数据库后已经明确编目了别名, 则使用别名。
- *userName* 是操作系统用户标识符。这是任选项。若建立的数据库带有 authentication=server, 则该任选项用于给数据库提供认证信息。
- *password* 是上述用户标识符的口令。

SwitchLoadFile

本属性用于指定列出 xa_ 例行程序名称的交换装入文件名称。下列交换装入文件附带在 CICS 中:

```
$CICS/bin/cicsxadb2 (对于两阶段落实)
$CICS/bin/cics1pcdb2 (对于单阶段落实)
```

cicsadd 命令用于将 XAD 添加到区域。例如:

```
cicsadd -c xad -r regionName DB2XAD \
ResourceDescription="XA Connection for DB2" \
XAOpen="cicsDB2database" \
SwitchLoadFile="/usr/lpp/cics/bin/cicsxadb2"
```

注: 其中, 'cicsxadb2' 是两阶段落实的 'cicsxadb2xa', 或单阶段落实的 'cicsdb21pc'。参阅步骤 3。

5. 步骤 5. 执行数据库的授权

当区域启动时, CICS 应用程序服务器使用 XA 开放字符串, 执行到数据库的初始连接。所有 CICS 事务或用已明确在 XAOpen 字符串中指定的用户运行, 或用系统设定用户 cics 运行。

至少, 本用户需要下列特权 (以 SQL 语句表示):

```
grant bindad on database to user cics
grant select on table sysibm.sysindexes to user cics
```

6. 步骤 6. 验证连接

CICS 在区域的控制台文件中记录 XA 信息。需要确定区域是否能够连接到 LRM，则启动区域并检查控制台信息。

- 启动区域。若这是新创建的区域，则按照第21页的『第2章 创建和配置 CICS 区域』中的步骤执行，否则使用下列命令：

```
startsrc -s cics.regionName -a StartType=cold
```

- 在紧随控制台信息的第二个窗口中，使用下列命令：

```
cicstail -r regionName
```

若发生 XA 问题，则检查 syslog 中的附加信息，需要更多信息，请参阅 *DB2: 问题确定指南*。

7. 步骤 7. 测试连接

CICS 提供的范例给出测试 XA 和非 XA 连接的绝佳方法。这些范例在：

```
$CICS/src/examples/xa  
$CICS/src/examples/nonxa
```

本例可用于了解如何编码并编译 CICS/SQL 应用程序。参考单独的 README 文件、*\$CICS/src/examples/xa* 中的 *uxa1.README* 和 *\$CICS/src/examples/nonxa* 中的 *nax1.README*。

启用到 Informix 的连接

本过程描述了如何启用区域连接到 Informix 数据库。描述了非 XA、XA 两阶段落实和单阶段落实连接，以及 Micro Focus COBOL 运行时间需求。

先决任务和条件:

Informix 环境: 若 Informix 服务器和客户机环境驻留在相同机器上，则重要的是按照在 Informix-CLI 安装指南 Microsoft Windows 环境和 UNIX 版（版本 2.5），以及 Informix UNIX 产品安装指南（版本 7.21）中指定的次序，安装 Informix 产品。

Informix 服务器环境 - 必须存在可存取的 Informix 服务器环境。*Informix UNIX 产品安装指南* 版本 7.21 解释如何安装 Informix 联机工作组服务器。

Informix 客户机环境 - 必须存在 Informix 客户环境。Informix 客户环境由 Informix ESQL（C 和/或 COBOL）构成。

若数据库服务器和客户机驻留在相同的机器上，则这些 Informix 应在 Informix 服务器之前安装。*Informix-CLI 安装指南* Microsoft Windows 环境和 UNIX 版（版本 2.5）和 *Informix UNIX 产品安装指南*（版本 7.21）解释如何安装 Informix ESQL 产品。（参阅 *Informix 联机动态服务器管理指南*（版本 7.2），以获得有关如何配置 Informix 数据库服务器的完整说明）。

Informix 服务器配置 -- 在 XA 环境中，不需要特定配置用于操作 Informix 联机工作组服务器，尽管应注意应该为数据库定义未缓冲的事务日志。

Informix 客户机配置 -- 若需要使用 Micro Focus COBOL 事务，则确保正确建立 COBOL 环境（参阅 *Informix-TP/XA 用户手册* 版本 7.2）。

过程:

本部分描述如何建立环境，以从支持 XA 和支持非 XA 的 CICS 中启用 C 和 COBOL 事务。 第94页的表 17 描述了应该为每个事务环境使用何种过程步骤。

表 17. 每个事务环境必需的过程步骤

非 XA 'C'	XA 'C'	非 XA COBOL	XA COBOL
步骤 1	步骤 1	步骤 1	步骤 1
		步骤 2	步骤 2
	步骤 3		步骤 3
	步骤 4		步骤 4
	步骤 5		步骤 5
	步骤 6		步骤 6
步骤 7	步骤 7	步骤 7	步骤 7

1. 步骤 1. 建立 Informix 的区域环境

区域的环境是当区域启动时，从超级用户的外壳继承而来的。可将数据库环境信息添加到区域的环境文件（/var/cics_regions/*regionName*/environment），以避免若区域从外壳启动时会发生的连接错误，这些外壳没有必需的数据库环境变量设置，如：

至少，区域的环境文件必须包括：

```
INFORMIXDIR>
INFORMIXSERVER
```

当将库路径添加到区域的环境文件中时，确保包括了所有在本区域运行的应用程序必需的库。原因是当区域启动时存在的库路径值将被区域环境文件中的库路径值替代。这对于所有值都是真实的，如 PATH、NLSPATH、LANG 等。

并且，请勿在区域的环境文件中使用 \$ 值，因为这些值是不能扩充的。参考区域的环境文件前言中的信息。

2. 步骤 2. 创建 Micro Focus COBOL 运行时间环境

当装入 COBOL 事务时，CICS 应用程序服务器需求 CICS COBOL 运行时间环境。通过调用 cicsmkcobol 实用程序创建 CICS COBOL 运行时间环境（如 CICS 管理参考大全 中描述）。若 COBOL 运行时间环境为一个区域特有，则它从区域的 bin 目录调用 cicsmkcobol。若 COBOL 运行时间环境是所有当前机器上的区域共享的，则它从 \$CICS/bin 目录调用 cicsmkcobol。

调用 **cicsmkcobol** 前，应先执行下列活动：

- 确定 COBOL 运行时间环境必需的 Informix 库和对象。

对于支持非 XA 的 COBOL 事务，下列 Informix 库和对象是必需的：

```
cicsmkcobol -L$INFORMIXDIR/lib/esql $INFORMIXDIR/lib/esql/libecob_mf2shr.o
```

注：INFORMIXDIR 环境变量应设置为安装 Informix 产品的目录。

3. 步骤 3. 建立交换装入文件

支持 XA 的数据库提供称为 XA 交换的结构，XA 事务管理器通过该结构访问各种被该数据库支持的 XA 功能。CICS 在运行时间装入 XA 交换结构，用于执行此项任务的代码为交换装入文件。

可建立两类交换装入文件，名为单阶段落实和两阶段落实。适当的两阶段落实切换装入文件在一些应用程序中，这些应用程序使用分布式资源管理器，然而，适当的

单阶段落实版本存在于仅使用资源管理器的应用程序中。在事务同步点期间，单阶段落实选项比两阶段落实协议略显简易。交换装入文件位于 CICS 根目录下的 \$CICS/src/examples 目录。

一阶段确认交换装入文件通过编译源文件建立。

```
$CICS/src/examples/onephase/inform1pc.ec
```

或使用实用程序生成文件。

```
$CICS/src/examples/onephase/inform1pc.mk:  
make -f inform1pc.mk
```

两阶段落实交换装入文件通过编译源文件建立。

```
$CICS/src/examples/xa/informxa7.c
```

或使用实用程序生成文件。

```
$CICS/src/examples/xa/informxa7.mk:  
make -f informxa7.mk
```

行交换装入实用程序生成文件前，确保正确设置了实用程序生成文件中的环境变量。运行 make 命令后将新建立的交换装入文件复制到区域的 bin 目录或将产品定义中的显示路径赋予可执行的 'informxa'。

4. 步骤 4. 添加产品定义 (XAD)

XA 规格允许将每个资源管理器定义为开放字符串，字符串指定用于连接到 RM 的 xa_open() 接口使用的参数。

XAD 描述 CICS 的 XA 环境，并包含其它内容，RM 特定的 XA 开放字符串 (XAOpen 属性) 和交换装入的位置 (SwitchLoadFile 属性)。

对于 Informix 7.2, XAOpen 属性的语法如下:

dataBaseName

- *dataBaseName* 是 Informix 数据库名。

SwitchLoadFile 属性用于标识在步骤 3 中建立的交换装入文件的位置。下面给出用于将产品定义 (XAD) 添加到区域的 **cicsadd** 命令范例:

```
cicsadd -c xad -r InfXAD \  
ResourceDescription="XA Connection for Informix" \  
XAOpen="cicsINF database" \  
SwitchLoadFile="informxa"
```

5. 步骤 5. 授予对 CICS 的数据库访问

当支持 XA 的区域启动时，CICS 应用程序服务器执行到由 XAOpen 字符串标识的数据库的连接，并使用由变量 INFORMIXSERVER 标识的服务器。'cics' 拥有应用程序服务器，必须给予用户 'cics' 对将要发生处理的数据库的访问权限。

可将 dbaccess Informix ISQL 设施作为 Informix 数据库管理员调用并选择用户数据库，以便运行下列 SQL 命令，用于执行给予对用户 'cics' 的访问权限:

```
grant resource to cics;
```

6. 步骤 6. 验证连接

CICS 在区域的控制台文件中记录 XA 信息。需要确定区域是否能够连接到资源管理器，则启动区域并检查控制台信息。

- a. 启动区域。若这是新创建的区域，则按照第21页的『第2章 创建和配置 CICS 区域』中的步骤执行，否则使用下列命令:

```
startsrc -s cics.regionName -a StartType=cold
```

- b. 在紧随控制台信息的第二个窗口中，使用下列命令：

```
cicstail -r regionName
```

检查注册到控制台的信息。当 CICS 应用程序服务器启动时，生成第一条 XA 信息。若发生的 XA 问题获得带有 Informix 实用程序 'onstat' 和 'onlog' 的附加信息，参阅 *Informix-TP/XA 用户手册* 版本 7.2 和 *Informix 联机动态服务器管理指南* 版本 7.2。

7. 步骤 7. 运行测试事务

CICS 提供的范例提供了测试 XA 和非 XA 连接的方法。 范例在：

```
$CICS/src/examples/xa  
$CICS/src/examples/nonxa
```

参考单独的 README 文件、\$CICS/src/examples/xa 中的 uxa1.README 和 \$CICS/src/examples/nonxa 中的 nax1.README。

启用到 Oracle 的连接

本过程描述了如何启用区域连接到 Oracle 数据库。描述非 XA、XA 一阶段和两阶段落实连接，以及有关使用支持 XA 的区域的 Micro Focus COBOL 的用法需求。

先决任务和条件：

- 验证安装了 Oracle 软件，如相关的 Oracle 安装手册中描述。
- 需要安装 Oracle XA 的 Oracle 共享程序库。正如其名称暗示，共享代码一次性装入进内存，并由所有引用其的进程共享。在运行时间，Micro Focus COBOL 运行时间环境、CICS 事务和交换装入文件需要引用 Oracle XA 共享程序库。

•

安装过程在 \$ORACLE_HOME/lib 目录中创建共享程序库。通过作为超级用户操作系统用户标识符运行下列命令，将该库复制到 /usr/lib 目录：

```
cp $ORACLE_HOME/lib/libcintsh.a /usr/lib
```

初始化 Oracle 视图

对于 **Oracle 7.3.2**：若 v\$xatrans\$ 视图不存在，则创建一个。可通过执行如下所示的 start \$ORACLE_HOME/rdbms/admin/xaview.sql 完成此操作：

```
cd $ORACLE_HOME/rdbms/admin  
run 'sqlplus' logging in as the user 'SYS'  
xaview
```

不需要 sql 文件扩展名。

- 将 v\$xatrans\$ 视图上选中的特权授予所有 xa_open 字符串中指定的用户。例如，若仅需要作为用户 scott 连接到数据库，则运行下列 sqlplus 命令：

```
grant select on v$xatrans$ to scott;
```

对于 **Oracle 8.0.3**：将 DBA_PENDING_TRANSACTIONS 视图上选中的特权授予所有 xa_open 字符串中指定的用户。例如，若仅需要作为用户 scott 连接到数据库，则运行下列 sqlplus 命令：

```
grant select on dba_pending_transactions to scott;
```

过程：

本部分描述如何建立环境，以从支持 XA 和支持非 XA 的 CICS 中启用 C 和 COBOL 事务。 第97页的表 18 描述每个事务环境应遵循的过程步骤。

表 18. 每个事务环境必需的过程步骤

非 XA 'C'	XA 'C'	非 XA COBOL	XA COBOL
步骤 1	步骤 1	步骤 1	步骤 1
		步骤 2	步骤 2
	步骤 3		步骤 3
	步骤 4		步骤 4
	步骤 5		步骤 5
步骤 6	步骤 6	步骤 6	步骤 6

1. 步骤 1. 建立 Oracle 的区域环境

区域的环境是当区域启动时从超级用户的外壳继承而来的。可将数据库环境信息添加到区域的环境文件（/var/cics_regions/*regionName*/environment），以避免若区域从外壳启动时会发生的连接错误，这些外壳没有必需的数据库环境变量设置，如：

ORACLE_SID
LIBPATH

当将库路径添加到区域的环境文件中时，确保包括了所有在本区域运行的应用程序必需的库。原因是当区域启动时存在的库路径值将被区域环境文件中的库路径值替代。这对于所有值都是真实的，如 PATH、NLSPATH、LANG 等。

并且，请勿在区域的环境文件中使用 \$ 值，因为这些值是不能扩充的。参考区域的环境文件前言中的信息。

2. 步骤 2. 创建 Micro Focus COBOL 运行时间支持文件

Micro Focus COBOL 应用程序需求称为 cicsprCOBOL 的运行时间支持文件。基于 COBOL 的事务使用该文件定位在运行时间必需的目标文件。例如，CICS 提供的功能或数据库库中的功能通过使用 cicsprCOBOL 定位。

需要被 CICS 使用， Micro Focus COBOL 运行时间支持文件必须置于\$CICS/bin 或 /vary/cics_regions/*regionName*/bin 中，这样将给予位于 \$CICS/bin 中的文件以优先权。允许您对于每个区域都有不同的 COBOL 运行时间支持文件。

提供一个称为 cicsmkcobol 的工具，它将建立 cicsprCOBOL 文件。

需要创建一个适用于支持 XA 的区域的 cicsprCOBOL 文件：

a. 确保工作目录是区域 bin 目录或 \$CICS/bin

b. 确保 PATH 允许 cicsmkcobol 查找：

- + Micro Focus COBOL executable files
- + COBOL executable files
- + Commands such as echo, sed and grep
- + CICS executable files

c. 确保 COBDIR 环境变量设置为 Micro Focus COBOL 安装目录。例如：

确保 LIBPATH=\$COBDIR/coblib:/usr/lib:/lib

d. 输入下列命令：

\$ make -f \$CICS/src/examples/xa/cicsprCOBOL_ora.mk

这将在当前目录中创建一个 cicsprCOBOL 文件。

3. 步骤 3. 建立交换装入文件

支持 XA 的数据库提供称为 XA 交换的结构，XA 事务管理器通过该结构访问各种被该数据库支持的 XA 功能。CICS 在运行时间装入 XA 交换结构，用于执行此项任务的代码为交换装入文件。

可建立两类交换装入文件，名为单阶段落实和两阶段落实。适当的两阶段落实切换装入文件在一些应用程序中，这些应用程序使用多支持 XA 的资源管理器，然而，适当的单阶段落实版本存在于仅使用一个支持 XA 的资源管理器中。在这种情况下，单阶段落实选项在事务同步点期间花费的系统开销比两阶段落实协议的较少。参考第88页的『单阶段落实优化』，以获得相关主题的更多信息。

实用程序生成的文件建立两类交换装入文件与 CICS 一起提供。

需要建立交换装入文件：

- a. 确保工作目录是区域的 bin 目录。
- b. 确保 ORACLE_HOME 环境变量根据本地 Oracle 安装而设置
- c. 需要建立单阶段落实交换装入文件，输入下列命令：

```
$ make -f $CICS/src/examples/onephase/oracle1pc.mk
```

这将在 bin 目录中建立一个名为 oracle1pc 的目标文件。需要建立两阶段落实交换装入文件，输入下列命令：

```
$make -f $CICS/src/examples/xa/oraclexa.mk
```

这将在 bin 目录中建立一个名为 oraclexa 的目标文件。

4. 步骤 4. 添加产品定义 (XAD)

CICS XAD 文件是将 XA 遵从资源管理器 (RM) 定义为 CICS 的工具。XAD 文件能在目录: /var/cics_regions/regionName/database/XAD 中找到在每节中的下列属性尤其重要。

XAOpen

定义 XA 开放字符串。

对于 Oracle 7.3.2, 字符串包含:

```
Oracle_XA +Acc=P/username/password
           +SesTm=sessionTimeLimit
           [+DB=databaseName]
           [+GPwd=P/groupPassword]
           [+LogDir=logDir]
           [+MaxCur=maximumOpenCursors]
           [+SqlNet=connectString]
           [+DbgFl=DebugBitMask]
```

对于 Oracle 8.0.3, 字符串包含:

```
Oracle_XA +Acc=P/username/password
           +SesTm=sessionTimeLimit
           [+DB=databaseName]
           [+LogDir=logDir]
           [+MaxCur=maximumOpenCursors]
           [+SqlNet=connectString]
           [+Loose_Coupling=true|false]
           [+SesWt=sessionWaitLimit]
           [+Threads=true|false]
           [+DbgFl=DebugBitMask]
```

方括号 ([...]) 中的字段是任选的。用户名和口令可以为空, "P/" 是有效的用户名/口令组合。参考 Oracle 文档, 以获得有关 XA 开放字符串中的每个字段的详细的说明。

Oracle 的 XA 开放字符串范例是:

```
Oracle_XA+Acc=P/scott/tiger+SesTm=35+LogDir=/tmp+DbgFl=1
```

本字符串引起 cicsas 连接到数据库

- 作为用户 scott 使用口令 tiger (Acc=P/scott/tiger)
- 35 秒后非活动事务异常终止 (SesTm=35)
- 在 /tmp 目录创建日志文件 (LogDir=/tmp)
- 调用 XA 过程并返回登录入日志文件 (DbgFl=1) "

SwitchLoadFile

本属性是一字符串, 定义交换装入文件对象的名称。在步骤 3 中给出指令, 说明如何建立一个适当的交换装入文件。

若交换装入文件称为 oraclexa, 并在 region/ bin 目录中, 则应将 SwitchLoadFile 属性设置为 "oraclexa"。

XAClose

对于 Oracle 7.3.2 和 Oracle 8.0.3, 这应设置为空串 ("")。

cicsadd 命令用于将 XAD 节添加到 XAD 节文件。例如, 可使用下列命令添加带有上述属性的节。

```
$cicsadd -c xad -r region name Oracle_XA \  
ResourceDescription="Oracle XA product definition" \  
XAOpen="Oracle_XA+Acc=P/scott/tiger+SesTm=35+LogDir=/tmp+DbgFl=1" \  
XAClose="" \  
SwitchLoadFile="oraclexa" $cicsadd -c xad -r
```

5. 步骤 5. 验证连接

CICS 在区域的控制台文件中记录 XA 信息。需要确定区域是否能够连接到资源管理器, 则启动区域并检查控制台信息。

- a. 启动区域。若这是新创建的区域, 则按照第 2 章 '创建并配置 CICS 区域' 中的步骤执行, 否则使用下列命令:

```
$startsrc -s cics.
```

- b. 在紧随控制台信息的第二个窗口中, 使用下列命令:

```
$cicstail -r
```

检查注册到控制台的信息。当 CICS 应用程序服务器启动时, 生成第一条 XA 信息。若发生 XA 问题, 则检查 Oracle 跟踪文件中的附加信息。(参阅步骤 4)。

6. 步骤 6. 运行测试事务

CICS 提供的例子提供了练习 XA 和非 XA 连接的简便方法。范例在:

```
$CICS/src/examples/xa  
$CICS/src/examples/nonxa
```

参考单独的 README 文件、\$CICS/src/examples/xa 中的 uxa1.README 和 \$CICS/src/examples/nonxa 中的 nax1.README。

启用到 Sybase 的连接

本过程描述了如何启用区域连接到 Sybase 逻辑资源管理器 (LRM)。描述 XA 和非 XA 连接, 以及 Micro Focus COBOL 运行时间需求。

注: *Configuring Sybase(R) Adaptive Server Enterprise UNIX Platforms* 描述如何启用 CICS 的 XA 连接。

先决任务和条件:

Sybase 服务器环境

- 必须存在可存取的 Sybase 服务器环境。 安装 *Sybase 自适应服务器* 说明如何安装 Sybase 自适应服务器

Sybase 开放式客户环境

- 对于运行支持 XA 的事务, 必须存在 Sybase XA 开放式客户环境。 Sybase XA 开放式客户环境包含:
 1. 嵌入式 SQL (C 和/或 COBOL) 的 Sybase
 2. Sybase 开放式客户/C 开发者组件
 3. CICS 的 Sybase XA 库

这些 Sybase 产品应按该次序安装。 *Sybase Installing Open Client/Server Products on UNIX platforms* 和 *Installing Sybase Products on IBM RISC System/6000* 说明如何安装 Sybase 产品。

- 对于运行支持非 XA 的事务, 必须存在 Sybase 开放式客户环境。 Sybase 开放式客户环境包含:
 - 嵌入式 SQL (C 和/或 COBOL) 的 Sybase
 - Sybase 开放式客户/C 开发者组件

这些 Sybase 产品应按该次序安装。 *Sybase Installing Open Client/Server Products on UNIX platforms* 和 *Installing Sybase Products on IBM RISC System/6000* 说明如何在 Solaris 上安装 Sybase 产品。

注: 若 Sybase 服务器与 Sybase 开放式客户在相同机器上, 则请参考 *Sybase Installing Open Client/Server Products on UNIX platforms*, 以获得关于在相同目录安装 SYBASE 产品的警告。

Sybase 服务器配置

- 按照 *Sybase 自适应服务器* 中定义的指令启动 *Sybase SQL 服务器*。
- 对于运行支持 XA 的事务, Sybase SQL 服务器需要定义 XA 存储的过程 (参阅 *Sybase XA 库集成指南*)。 可将脚本 `$SYBASE/scripts/xa_load.sh` 添加到 Sybase XA 存储的过程。 查看该脚本的内容, 然后将其作为命令运行。

Sybase 开放式客户配置

- 对于运行支持 XA 的事务, 更新 Sybase XA 配置文件 `$SYBASE/xa_config`。 Sybase XA 配置文件用逻辑资源管理器 (LRM) 名称标识 Sybase 服务器。 LRM 名将用于 CICS XAD 产品定义的 XAOpen 字符串。 LRM 的功能和属性也可在 Sybase XA 配置文件中指定。
- 在 XA 配置文件中标识的 Sybase 服务器必须在 `$SYBASE/interfaces` 文件中标识。

- 对于运行非 XA 事务，\$SYBASE/interfaces 文件必须标识由事务连接到的 Sybase 服务器。
- 需要确保装入器与 Sybase 开放式客户机库文件之间不存在问题，应该运行下列脚本：
\$SYBASE/sample/xalibrary/CICS/mkshlibs.sh

需要更多信息，请读者参考 *Configuring Sybase(R) Adaptive Server Enterprise for UNIX Platforms*。

过程:

本部分描述如何建立环境，以从支持 XA 和支持非 XA 的 CICS 中启用 'C' 和 COBOL 事务。 第101页的表 19 描述每个事务环境应遵循的过程步骤。

表 19. 每个事务环境必需的过程步骤

非 XA 'C'	XA 'C'	非 XA COBOL	XA COBOL
步骤 1	步骤 1	步骤 1	步骤 1
		步骤 2	步骤 2
	步骤 3		步骤 3
	步骤 4		步骤 4
	步骤 5		步骤 5
	步骤 6		步骤 6
步骤 7	步骤 7	步骤 7	步骤 7

1. 步骤 1. 建立 Sybase 的 CICS 区域环境

区域的环境是当区域启动时从超级用户的外壳继承而来的。 可将数据库环境信息添加到区域的环境文件（/var/cics_regions/regionName/environment），以避免若区域从外壳启动时会发生的连接错误，这些外壳没有必需的数据库环境变量设置，如:

至少，区域的环境文件必须包括:

```
SYBASE=sybaseInstanceHomeDirectory
DSQUERY=sybaseInstanceName
```

当将库路径添加到区域的环境文件中时，确保包括了所有在本区域运行的应用程序必需的库。原因是当区域启动时存在的库路径值将被区域环境文件中的库路径值替代。这对于所有值都是真实的，如 PATH、NLSPATH、LANG 等。 并且，请勿在区域的环境文件中使用 \$ 值，因为这些值是不能扩充的。 参考区域的环境文件前言中的信息。

2. 步骤 2. 创建 Micro Focus COBOL 运行时间环境

CICS 应用程序服务器需求 CICS Micro Focus COBOL COBOL 环境，以便运行 COBOL 事务。通过调用 cicsmkcobol 实用程序创建 Micro Focus COBOL 运行时间环境（如 *CICS 管理参考大全* 中描述）。若 COBOL 运行时间环境为一个区域特有，则它从那个区域的 bin 目录调用 cicsmkcobol。若 COBOL 运行时间环境为所有当前机器上的区域共享，则从 \$CICS/bin 目录调用 cicsmkcobol。

调用 **cicsmkcobol** 前，应先执行下列活动:

- 从 Sybase COBOL 库中抽取对象:

```
ar -x $SYBASE/lib/libcobct_r.a
```
- 设置一个环境变量可能会有帮助，它指定从 Sybase COBOL 库中抽取的对象。下列范例显示如何从命令行执行:

```
COBOJS='ar -t $SYBASE/lib/libcobct_r.a'
```

- c. 确定 COBOL 运行时间环境必需的 Sybase 库。对于支持非 XA 的 COBOL 事务，下列 Sybase 开放式客户库是必需的：

```
libct_r.so libcs_r.so libintl_r.so libtcl_r.so libcomn_r.so
```

确定 COBOL 运行时间环境必需的 Sybase 库。对于支持 XA 的 COBOL 事务，下列 Sybase 开放式客户库是必需的：

```
libct_r.so libcs_r.so libintl_r.so libtcl_r.so libcomn_r.so libintr_r.so
```

设置一个环境变量可能会有用，它指定包括在 CICS COBOL 运行时间环境中的 Sybase 库。下列范例显示如何从命令行执行：

```
SYBLIBS=ListOfLibraries
```

其中，非 XA 环境中的 *ListOfLibraries* 是：

```
-lct_r.so -lcs_r.so -lintl_r.so -ltcl_r.so -lcomn_r.so
```

其中，XA 环境中的 *ListOfLibraries* 为：

```
-lct_r.so -lcs_r.so -lintl_r.so -ltcl_dce.so \  
-lcomn_dce.so -linsck_r.so
```

在区域的 bin 目录或 \$CICS/bin 中建立 COBOL 运行时间环境。若已经如上所述设置了环境变量，应输入下列命令：

```
cicsmkcobol -L$SYBASE/lib $COBOJS $SYBLIBS
```

注：SYBASE 应如步骤 1 定义，SybaseOpenClientHomeDirectory COBDIR 应该作为 Cobol 自用目录定义。

3. 步骤 3. 建立交换装入文件

交换装入文件仅在支持 XA 的环境中必需。

交换装入文件的源位于 CICS 根目录下的 \$CICS/src/examples/xa 目录中。需要建立交换装入文件：

- a. 将交换装入文件源和实用程序生成文件复制到区域的 bin 目录中：

```
cd /var/cics_regions/regionName/bin  
cp /opt/cics/src/examples/xa/sybasexa.*
```

- b. 编辑实用程序生成文件并更改必需的环境值。

- c. 使用下列命令建立交换装入文件：

```
make-f sybasexaimk
```

4. 步骤 4. 添加产品定义 (XAD)

XA 规格允许将每个 RM 供应商定义开放字符串，字符串指定参数用于连接到 RM 供应商的 xa_open() 接口。

XAD 描述 CICS 的 XA 环境，并包含其它内容，RM 特定的 XA 开放字符串 (XAOpen 属性) 和交换装入的位置 (SwitchLoadFile 属性)。

XAOpen 属性的语法如下：

```
-U userName -P passWord -N LRMname -L logFile [-T traceLevel]
```

- *userName* 和 *passWord* 用于标识到 Sybase 数据库的连接。两者都必须引用有效的 Sybase 用户帐户。Sybase 用户帐户由 Sybase 系统管理员通过使用存储过程的 sp_addlogin 和 sp_adduser 创建 (参阅 Sybase XA 库集成指南)。

- *LRMname* 是逻辑资源管理器的名称，如 Sybase XA 配置文件 `$SYBASE/xa_config` 中所定义。LRMname 在 XA 事务中用命令 `'EXEC SQL SET CONNECTION LRMname'` 标识。
- *logFile* 是将写入跟踪信息的文件的完整路径名。本 LogFile 对于错误信息是必需的，Sybase 必须对该文件有写访问的权限。
- *traceLevel* 是写入日志文件的跟踪级别。若跟踪希望将此设置为 'xa'，则给出 xa 过程的项和出口点跟踪。
- 参阅 *Configuring Sybase(R) Adaptive Server Enterprise for UNIX Platforms*，以获得其它有效的设置。SwitchLoadFile 属性用于标识在步骤 3 中建立的交换装入文件的位置。范例将 'sybasexa' 标识为交换装入文件，不给出路径，所以 'sybasexa' 预料在区域的 bin 目录中。例如，如下所示，**cicsadd** 命令将产品定义 (XAD) 添加到区域：

```
cicsadd -c xad -r regionName SybaseXA \
ResourceDescription="XA Connection for Sybase" \
XAOpen="-Ucicsid -Pcicspw -NlrName -L/tmp/lrmXA.log" \
SwitchLoadFile="sybasexa"
```

5. 步骤 5. 授权对 CICS 的数据库访问

当支持 XA 的区域启动时，CICS 应用程序服务器通过使用 XA 开放字符串执行到 LRM 的初始连接。对于 Sybase，XA 开放字符串标识将执行所有 CICS XA 事务的用户名和口令。确保 XA CICS 用户对 Sybase XA 存储的过程有存取权是非常重要的。

步骤 7 中所指出的例子使用名为 'cicstest' 的数据库。下列 isql 命令建立 XA CICS 用户 'cicsid'，起始数据库为 'cicstest'，口令为 'cicspw'。期望从 Sybase 系统管理员的会话内运行 isql 命令：

```
sp_addlogin cicsid, cicspw, cicstest
go
use cicstest
go
quit
```

6. 步骤 6. 验证连接

CICS 在区域的控制台文件中记录 XA 信息。需要确定区域是否能够连接到 LRM，则启动区域并检查控制台信息。

- 启动区域。若这是新创建的区域，则按照第21页的『第2章 创建和配置 CICS 区域』中的步骤执行，否则使用下列命令：

```
startsrc -s cicsregionName -a StartType=cold
```

- 在紧随控制台信息的第二个窗口中，使用下列命令：

```
cicstail -r regionName
```

本例也可用于了解如何编码并编译 CICS/SQL 应用程序。参考 `$CICS/src/examples/xa` 和 `$CICS/src/examples/nonxa` 目录中的自述文件。

7. 步骤 7. 运行测试事务

CICS 提供的范例提供完全测试 XA 和非 XA 连接的绝佳方法。范例在：

```
$CICS/src/examples/xa
$CICS/src/examples/nonxa
```

参考单独的 README 文件、`$CICS/src/examples/xa` 中的 `uxa1.README` 和 `$CICS/src/examples/nonxa` 中的 `nax1.README`。

第6章 使用用户出口和用户程序定制

本章描述如何使用用户出口和用户程序来配置 CICS。在本章中描述的大多数用户程序是 CICS 提供的，在这种情况下，所提供的程序能为您的需要而修改。后初始和关机程序不是 CICS 提供的。

CICS 管理参考大全 包含用户出口和 CICS 提供的用户程序的参考信息。

表 20. 使用用户出口和用户程序定制的导向图

若需...	参考...
使用下列: <ul style="list-style-type: none">• 任务结束用户出口。• 转储请求用户出口。• 动态事务路由选择用户出口。• 动态分布程序链接用户出口。• 同步点用户出口。• ECI 和 EPI 用户出口。	第105页的『CICS 用户出口』。
使用下列: <ul style="list-style-type: none">• 外部安全性管理器用户程序。• 终端自动安装用户程序。• cics 客户机自动安装用户程序。• 性能监控程序用户程序。• 后初始用户程序。• 关机用户程序。	第114页的『用户程序』。

CICS 用户出口

用户出口点（也指用户出口）是位于 CICS 能够传送控制到程序的 CICS 模块的位置，已经写该程序（一个用户出口程序），并且当继续结束其工作时，在该点上 CICS 能够继续控制。不必须使用任何用户出口，但是，能够根据您的需求，使用其来扩展并定制 CICS 系统的功能。

如何使用用户出口

CICS 提供的用户出口是：

任务结束

可在正常和异常中断时调用任务结束用户（在取出任何同步点后）。参阅 CICS 管理参考大全 中的“任务终止用户退出 (UE014015)”，以获得更多信息。

转储请求

可调用转储请求用户出口：

- 当发出 EXEC CICS DUMP 命令时

- 当发生事务异常终止时
- 当请求 CEMT PERF SNAP 时
- 当区域关闭时
- 当发生系统异常终止时

参阅 *CICS 管理参考大全* 中的“转储请求用户退出 (UE052017)”，以获得更多信息。

动态事务路由选择

当请求路由一个事务时，可调用动态事务路由选择用户出口。本用户出口根据用户出口条件中指定的条件，动态地路由事务。参阅第108页的『使用动态事务处理路由选择用户出口』。

动态分布的程序连接

当请求连接到另一个程序时，可调用动态分布程序连接用户出口。本用户出口根据用户出口条件中指定的条件，动态地连接程序。参阅第111页的『使用动态分布程序链接用户出口』。

同步点 当事务发出 EXEC CICS SYNCPOINT 时，可调用同步点用户出口。参阅 *CICS 管理参考大全* 中的“同步点用户退出 (UE016051)”。

ECI 和 EPI

ECI 和 EPI 用户出口描述在 *CICS 系列：客户机/服务器程序设计* 中。

将出口程序与其用户出口关联

将出口程序与用户出口关联有一个半静态方法，取决于程序定义 (PD) 中的 **UserExitNumber** 属性。它提供：

- 将单个程序与用户出口关联的类属方法
- 调用用户出口的类属方法
- 用户出口的标准命名约定
- 当调用用户出口时，传递的标准参数的结构

注：虽然等价的 CICS OS/2 版出口由命名约定标识，但是，由于基本 CICS 环境存在差异，所以用户出口的行为和传递的参数表应视为是相似而非等同的。

PD **UserExitNumber** 属性为将单个程序与用户出口关联提供了类属方法。与程序关联的用户出口被认为是活动的。这种关系将在新应用程序服务器创建时传递给它们，但不是传递给现存的应用程序服务器。何处程序已经与用户出口关联，则该请求替代任何以前新应用程序服务器的请求。

当 PD 卸载时，删除新应用程序服务器的用户出口程序关系。当没有用户出口程序的 PD 时，用户出口被视为非活动的；这适用于**所有**应用程序服务器。若不同 PD 的用户出口稍后重激活，则本新用户出口程序关联仅适用于新应用程序服务器。不可能为现存的应用程序服务器复制新用户出口程序。

当应用程序服务器启动时，根据 PD 定义，在该应用程序服务器的生存期设置程序与用户出口的关联。当应用程序服务器开始保存每次执行调用用户出口的性能系统开销时，装入用户出口。

需要确认相同的程序用户出口关系在所有应用程序服务器中是活动的，无论何时现存的用户出口将要被另一个程序或相同程序的新版本所替换，都建议关闭并重新启动 CICS。

写用户出口程序的规则和指导

样本用户出口程序在下列目录中提供：

`/usr/lpp/cics/src/samples/userexit`

当写用户出口程序时，请考虑下列各项事宜：

1. 用户出口程序必须使用 C 语言。
2. 用户出口程序必须为可重入的。
3. 请勿切换用户出口程序中的线程。
4. 因为共享资源可能导致封锁，使用 `mutexes`²。
5. 用户的责任是不损坏 CICS 的系统完整性。
6. 用户出口程序必须不使用 EXEC CICS 命令（应用程序设计接口）。
7. 提供的特定用户出口的功能原型必须包括在用户出口程序内。
8. 提供的特定用户出口的出口专用结构必须包括在用户出口程序内。
9. 必须在所有用户出口程序中使用提供的 `cics_UE_Header_t` 结构。（参阅注意 21。）
10. 所有写 CICS 应用程序的文档化限制（参阅 *CICS 应用程序设计指南*）也适用于写用户出口程序。另外，请勿使用系统调用。
11. 请勿将用户出口程序作为 CICS 应用程序使用，因为如果这样，将导致出不可预测的结果。
12. 无专用设施针对处理用户出口程序中的出错。
13. 用户出口程序必须设置一个返回码，定义其用来支持用户出口；否则，用户出口程序将执行系统设定操作，它在许多情况下将对应用程序服务器禁用用户出口。
14. 在用户出口程序内，没有提供访问 CICS 资源的设施。
15. 若将重试机制提供给用户出口，该机制就明显地成为用户出口定义中的一部分，并且提供一个参数指示正在重驱动用户出口。
16. 在所有提供给用户出口程序内使用的结构上，都供有版本号。必须保证用户出口程序能检查版本号，并且使用结构的正确版本。
17. 给出供在应用程序服务器内的用户出口之间共享的任务工作区，这些应用程序服务器使用相同的 TCA。必须保证在该任务工作区内的字段用法中不存在冲突。指向任务工作区的指针传递给所有用户出口。在任务之间初始化工作区。
18. 任务工作区的长度固定为 128 字节，并不能更改。
19. 信号处理器必须不包括在用户出口程序中。
20. 用户出口程序与其它软件包接合的用法可能是不可预测并不可支持的。在 CICS 产品环境中实现用户出口之前，能为任何这样的需求寻求指南将是明智之举。
21. 用户出口的标准首部结构（`cics_UE_Header_t`）、常数、数据定义、出口专用结构和函数原型都在头文件 `cicsue.h` 中。`cicsue.h` 文件应包括在所有用户出口程序中。

2. 在应用程序内，为短期与内部数据结构（相对于可恢复的数据）而使用的互斥机制。如需更多信息，请参考 Encina 文档。

- 22. 所有用户出口都用连接的共享内存调用。必须保证用户出口码不泄露 CICS 的系统完整性。
- 23. 所有用户出口程序的入口点是 `cics_UE_entry`。
- 24. 所有用户出口都用两个参数调用:
 - 指向 `cics_UE_Header` 参数表的指针
 - 指向所调用用户出口的出口专用参数表的指针
- 25. 若用户出口参数表中的 `enum` 参数在 `cicsue.h` 中定义为 `typedef`, 则当在引用参数时需要避免不可预测的结果时, 用户出口程序应使用提供的数据名。

迁移和共存

需要辅助从其它 CICS 系列成员中迁移, 用户出口命名约定为存在的 OS/2 出口号合并等价的 CICS。

表 21. 用户出口号

用户出口号	说明	用户出口名
15	任务结束	UE014015
17	转储请求	UE052017
25	动态事务处理路由选择	UE014025
50	动态分布程序连接	UE015050
51	同步点	UE015051

本表显示:

用户出口号

在 PD **UserExitNumber** 属性中使用的号码。

说明 用户出口的说明。

用户出口名

每个用户出口有唯一名称, 并且位于模块中有利于进行附加处理的一点。用户出口名的格式是:

`UEcccnnn`

其中, `ccc` 是用于调用用户出口程序 CICS 组件号码, `nnn` 是用户出口号, 为零开头的三位填充字符。

使用动态事务处理路由选择用户出口

动态事务路由选择使您能够定义如何根据以下因素路由一个事务:

- 输入到事务
- 可用的 CICS 系统
- 可用系统的相对装入

并且, 路由选择程序能够执行除了重定向事务请求以外的功能, 如:

- 平衡工作负荷。例如, 在多 CICS 环境内, 程序能在并行系统上的等价事务之间作出智能型的选则。
- 处理不可路由的事务, 如无可用远程 CICS 区域时。
- 在路由到的事务内, 处理异常终止。

- 监控路由到特定系统的请求数。

动态事务路由选择不能用于重路由远程 ATI 请求。

CICS 通过使用 CICS 提供的动态事务路由选择用户出口来管理动态事务路由选择。调用该用户出口：

- 在路由选择定义为 **Dynamic=yes** 的事务之前。
- 若路由选择中发生错误。
- 在一个路由事务的最后（若初始调用请求在终止时重调用）。
- 若路由的事务异常终止，且初始调用请求在终止时重调用。

参数在 CICS 和动态事务路由选择用户出口程序之间的结构中传递。程序可能更改这些参数中的一些，以影响随后的 CICS 操作。参数包括：

- 当前调用的原因。
- 出错信息。
- 目标系统名。最先，这是用 **TD RemoteSysId** 属性指定的系统。若未指定系统，则传递的名称是本地系统的。
- 目标事务名。最先，这是用 **RemoteName** 属性指定的名称。若未指定目标事务，则传递的名称为本地事务的名称。
- 指向 CWA 的指针。
- 指向 TCTUA 的指针。
- 一个用户区。

CICS 提供一个可由用户出口调用的样本程序，但是可用自用程序替代。需要执行，则定义区域的 PD 中的程序，并将 **UserExitNumber** 属性设置为 25。

虽然 **RemoteSysId** 和 **RemoteName** 属性用于动态事务路由选择用户出口，以路由选择事务，但是，当事务本地运行时，它们是被忽略的。若建立动态事务路由选择用户出口为允许本地运行事务，则需要定义本地程序。

不调用该用户出口：

- 当定义为动态的事务在中间应用程序自用区域中启动时。CICS 试图异常终止这种事务，但是无法检测所有到“菊链”动态事务的尝试。**必须确保动态事务路由选择程序不“菊链”动态事务。**
- 当由于缺少存储器而发生问题，而正在建立用户出口的参数表或标准首部。这导致事务异常终止。
- 当动态事务由一个在终端自用区域发出的下列类型命令在终端自用区域内启动时：

```
EXEC CICS START TERM(yyyy) TRAN(xxxx)
```

在这种情况下，不调用用户出口，且事务路由回发出请求的系统。

写动态事务路由选择用户出口

因为动态事务路由选择的内部通信特征和为用户出口建立的参数表，在使用动态事务路由选择用户出口的过程中牵涉性能的系统开销。但是，如果是一个写得很好的动态事务路由选择用户出口，针对几个连接到的区域的整个性能，则其益处远远超出了其负面影响。

下面的信息概括了一个可能希望动态事务路由选择用户出口执行的任务，并描述如何以最有效的方法在程序中将其实现。

如何在 CICS 和用户出口之间传递信息： CICS 以参数表方式将信息传递到动态事务路由选择出口。参数表包含标准头结构（cics_UE_Header_t），该结构将被传递到所有用户出口和特定为动态事务路由选择（cics_UE014025_t）的结构。这些结构都包括在头文件（cicsue.h）中。传递到参数表中动态事务路由选择程序的数据排序是：

- 事务处理定义（TD）中指定的远程 CICS 区域的系统 ID
- 远程事务名
- 本地任务用户数据区域

可写一个动态事务路由程序，它或接受这些值、更改其，或指示 CICS 不继续路由选择事务。使用的值取决于要执行的功能；也就是，可忽略一些值。

通过本部分，给出将一参数设置为空串的参考。空串是以空字符 '\0' 开头的字符串。例如，可将空串置于用户出口 UE014025 的参数表的系统 ID 参数中，如下所示：

```
strcpy (UE_specificptr->UE_Dyrsysid, "");
```

所有其它由用户出口返回的字符串必须以空字符结束。

需要一套完整的在 CICS 和动态事务路由选择程序之间传递的参数说明，请参阅 *CICS 管理参考大全* 中的“动态事务路由选择用户退出 (UE014025)”。

更改目标 CICS 系统： 传递到动态事务路由选择用户出口的参数表初始包含系统设定 CICS 区域的系统 ID，事务将路由至该系统设定 CICS。这些值从安装事务的 **RemoteSysId** 属性值处得到。若事务定义未指定 **RemoteSysId** 值，则传递的系统 ID 为本地 CICS 区域的系统 ID。

传递到用户出口特定结构中的动态事务路由程序的信息可更改为重路由事务。

更改程序名： 当调用动态事务路由选择用户出口时，**UE_dyrprog** 参数包含取自事务 TD 的 **ProgName** 属性的程序名。若未在事务 TD 的 **ProgName** 属性中定义程序名，则在用户出口的程序名参数中传递一空串。若决定本地路由事务，则可使用本字段指定要运行的替代程序。例如，如果所有远程 CICS 系统不可用，并且不能路由事务，则可能希望在本本地 CICS 系统运行一个程序，将适当信息发送给用户。

注：若动态事务路由选择用户出口在 **UE_Dyrprog** 参数返回一空串，则 CICS 发出异常终止信息，尽管动态事务路由选择用户出口已选择路由到一个远程系统。

通知 CICS 是路由还是终止事务： 若希望路由一个事务，则无论您是否已经更改了值，都以返回码返回 **UE_Normal**。若希望以信息和异常终止来终止事务，则提供 **UE_Term_Abend** 返回码。进一步选项（返回码 **UE_Terminate**）通知 CICS 以信息而不是异常终止来终止事务。

当以返回码 **UE_Normal** 返回控制，CICS 首先将返回的系统 ID 与本地系统 ID 作比较：

- 若系统 ID 都是相同的（或返回的系统 ID 是空串），则 CICS 在本地执行事务，使用在参数 **UE_Dyrprog** 中指定的程序名。
- 若两个系统 ID 不是相同的，则 CICS 将事务路由至远程 CICS 系统，使用由用户出口指定的远程事务名和远程系统名。

若系统不可用或不知道： 若在路由选择上指定的远程系统名不知道或不可用，并且通过将 **UE_Dyrretry** 设置为 **UE_Yes** 指定希望重试路由请求，则将再次调用动态事务路由选择程序。当发生这种情况时，可选择操作：

- 可通过发出 **UE_Term_Abend** 返回码，指示 CICS 不再继续尝试路由事务。若出错的原因在于系统不可用，则 CICS 发出信息 'ERZ1433E' 和异常终止 'A149'。
- 可通过仅发出一条 **UE_Terminate** 返回码，通知 CICS 终止事务。
- 可更改系统 ID，并发出 **UE_Normal** 返回码，以再次路由事务。若更改了系统 ID，则可能还需要提供一个不同的远程事务 ID。例如，若事务在每个系统都有不同的远程名，则需要执行该项操作。
- 可通过提供本地系统 ID（或将系统 ID 设置为空串）并通过提供程序名，以选择在本地运行事务。注意，程序名可为空缺，系统设定值为在事务 TD 中指定的程序名。
- 可尝试再次路由至相同系统。

为该事务的路由选择目的而调用路由选择程序，调用次数的计数在字段 **UE_Dyrcount** 中传递。在您的程序中，可为该参数设置一个限制，以启用其决定何时停止尝试路由一个特定的事务实例。

在路由的事务最后调用用户出口： 若希望当路由的事务已完成时再次调用动态事务路由选择程序，则必须在返回控制到 CICS 之前的用户出口初始调用时，将参数表中的 **UE_Dyropter** 字段设置为 **UE_Yes**。例如，若正保存着在特定 CICS 系统上当前执行事务的计数值，则可能希望执行该项操作。但是，在该重调用期间，动态事务路由选择程序应仅更新其自用资源。

异常终止时调用用户出口： 若路由的事务异常终止，则启动事务的 CICS 系统重调用动态路由选择程序（若通过将 **UE_Dyropter** 设置为 **UE_Yes** 来请求）。若动态事务异常终止，则无论它是路由到远程还是本地区域，都将调用本出口点。

本出口点无法重试路由请求，并应仅更新其自用的资源。本出口点可用于保存关于异常终止事务的信息，程序可使用该信息影响今后要路由的事务。

相关信息

CICS 内部通信指南 中的“事务处理路由选择的应用程序设计”。

CICS 管理参考大全 中的“动态事务路由选择用户退出 (UE014025)”。

在 *CICS 系列：产品间通信* 中的『启动事务路由选择』。

使用动态分布程序链接用户出口

当从应用程序请求程序的链接时，CICS 根据程序是如何在 PD 上定义，或者根据应用程序内传递的参数，将其链接到本地或远程系统上的程序。动态分布程序链接用户出口允许您动态地选择链接请求路由到的系统。

CICS 调用动态分布程序链接用户出口：

- 当用户出口说明为用户出口程序使用 PD 中的 **UserExitNumber** 属性时。用户出口号为 50。
- 当将要链接到在 EXEC CICS LINK 命令的 PROGRAM 字段中所指定的程序时。
- 当外部 CICS 接口 (ECI) 程序调用一个到 CICS 应用程序的链接请求时。
- 当由 CICS 应用程序或 ECI 程序发出的链接请求指定了一个没有 PD 的程序时。
- 在动态链接到的程序遇到问题后，且初始调用请求在终止时重调用。

- 在动态链接到的程序成功完成后，且初始调用请求在终止时重调用。

不能使用该用户出口，以：

- 动态链接未使用链接请求启动的程序
- 当调用用户出口时，发生存储器不够问题
- 动态链接 CICS 提供的事务。

传递到用户出口的信息

CICS 通过参数表的方式，将信息传递到动态分布程序链接用户出口。参数表包含标准头结构（`cics_UE_Header_t`），用来传递到所有用户出口和动态 DPL 特定结构（`cics_UE015050_t`）。这些结构都包括在头文件（`cicsue.h`）中。

给出将一参数设置为空串的参考。所有其它由用户出口返回的字符串必须以空字符结束。

用户出口的初始调用

若发出了一个程序的链接请求，则 CICS 调用动态分布程序链接出口。本链接请求能以应用程序指定的 EXEC CICS LINK 命令或以 ECI 程序的链接请求为格式。

若在链接请求上指定的程序拥有有效的 PD 项，则用户出口因 **UE_LINKSEL** 调用。但是，链接请求上指定的程序不需求 PD 项。若接收到不含有数据库项的程序的链接请求，则因 **UE_LINKUNKNOWN** 调用用户出口。用户的责任在于决定是尝试重路由链接请求，还是判断程序是否是无效的（通过返回 **UE_ProgramNotKnown** 返回码）。

当 CICS 调用动态分布程序链接用户出口时，它传递入一些信息关于程序将在何处运行：该信息包括：

- 需要运行程序的远程系统名。若需要使用本地系统，该项可留为空白。
- 需要使用的镜像事务名。若需要使用 CICS 提供的镜像事务 CPMI，该项可留为空白。
- 在本地或远程系统上运行的程序名。
- 在本地或远程系统上，用来运行程序的 CICS 用户标识符。这是初始执行 EXEC CICS LINK 命令的用户标识符。若请求是到本地系统的，则可更新它。

上述字段是由 CICS 建立的，以便指示用于程序链接请求的系统设定操作。当调用用户出口的原因是 **UE_LINKSEL** 时，这些字段可根据用户出口决定的操作，相应地留为系统设定值或进行更改。

若调用是 **UE_LINKUNKNOWN**，则必须更改程序 SYSID 或程序名。若需要，这些字段（和镜像事务名称）都可更改。

上述字段的系统设定值通过使用下列来决定，并取决于是否进行了隐式还是显示请求、程序是否有 PD 项、是本地还是远程链接请求：

- 若这是显示链接请求（无论程序是否有 PD 项）：
 1. 远程程序名取自 EXEC CICS LINK 命令的 PROGRAM 字段。
 2. 远程 SYSID 取自 EXEC CICS LINK 命令的 SYSID 字段。
 3. 镜像事务 ID 或取自 EXEC CICS LINK 的 TRANSID 选项，或若未指定 TRANSID 选项，该项设置为空串。
- 若这是隐式请求不存在的程序：

1. 远程程序名取自 EXEC CICS LINK 命令的 PROGRAM 字段。
 2. 远程 SYSID 设置为空串。
 3. 镜像事务 ID 或取自 EXEC CICS LINK 的 TRANSID 选项，或若未指定 TRANSID 选项，该项设置为空串。
- 若这是隐式本地请求存在的程序：
 1. 远程程序名取自 EXEC CICS LINK 命令的 PROGRAM 字段。
 2. 远程 SYSID 设置为空串。
 3. 镜像事务 ID 或取自 EXEC CICS LINK 的 TRANSID 选项，或若未指定 TRANSID 选项，该项取自 PD 的 TransId 字段。
 - 若这是隐式远程请求存在的程序：
 1. 远程程序名取自 PD 的 RemoteName 属性，或若这是空串，则使用 EXEC CICS LINK 命令的 PROGRAM 字段。
 2. 远程 SYSID 取自程序 PD 项的 RemoteSysId 选项。
 3. 镜像事务 ID 或取自 EXEC CICS LINK 的 TRANSID 选项，或若未指定 TRANSID 选项，该项取自 PD 的 TransId 字段。

更改目标 CICS 系统

用户出口传递系统设定远程 SYSID。若系统设定的 SYSID 为本地 SYSID，则将空串传递到用户出口。

可更改传递到的动态分布程序链接出口的信息，该链接出口在用户出口特定结构中，以便能够重路由分布程序链接请求。

若希望将程序链接重路由到本地系统，则 SYSID 应留为空白，或设置为本地 SYSID（使用参数 **UE_Dplclsys** 传入）。

更改远程程序名

用户出口传递系统设定程序名。若请求是本地的，则这就是要在本地系统上运行的程序名。若请求是远程的，则这就是远程系统上的程序名。

注：若用户出口在 **UE_Dplprog** 参数中返回空串，则发出异常终止和信息。

更改镜像事务名

用户出口传递系统设定镜像事务 ID。空白的镜像事务 ID 指示将使用 CICS 提供的镜像事务 CPMI 作为系统设定值。

更改用户标识符

当用户标识符执行 EXEC CICS LINK 命令时，传递用户出口。若请求是到远程系统的，并且请求还不是系统内现存的逻辑工作单元的一部分，则可在远程系统修改过的用户标识符下执行 DPL 请求。修改过的用户标识符不需要在执行 DPL 用户出口的 CICS 系统上配置。验证请求后发送用户标识符，也就是忽略任何远程系统**通信定义**中的 **OutboundUserids** 属性的设置，并不带口令地发送用户标识符。若出口将字段用户标识符 **UE_Dpluserid** 更改为空串，则请求将不带安全性属性发送。若未更改用户标识符，则安全性属性如远程系统的**通信定义**中配置的那样发送。

若在相同工作单元内发出到相同远程系统的进一步请求，则对于特定逻辑工作单元，为所有到该系统的请求使用相同的用户标识符。

在路由的程序结尾调用用户出口

若希望当路由的程序已完成时再次调用动态分布程序链接用户出口程序，则必须在返回控制到 CICS 之前初始调用用户出口时，将参数表中的 **UE_Dploppter** 字段设置为 **UE_Yes**。

或因 **UE_LINKTERM** 调用最终出口（到成功完成应用程序的链接），或 **UE_LINKABEND**（到未成功完成应用程序的链接）。对于以上两个原因，参数表都包含用于路由链接请求的值。因此，最终出口点可用于保存统计数据，如成功的链路请求，或表现为不可用的系统。

相关信息

CICS 内部通信指南 中的“分布式程序链路(DPL)”。

CICS 管理参考大全 中的“用户退出”。

在 *CICS 系列: 产品间通信* 中『启动事务路由选择』。

用户程序

用户程序使您可能修改系统而无需更改其任何标准用户接口。可定制或替换这些程序，以适应安装需求。

使用外部安全性管理器用户程序

系统设定的外部安全性管理器（ESM）程序 **cics_esm** 作为 CICS 的一部分提供。可以其未更改的形式实现它，或可进行定制，以满足您特殊位置的特殊需求。注意，在其未更改的形式内，ESM 简单地将 **CICS_ESM_RETURN_OK** 返回给每个安全性检查，指示已将特许访问授权给事务或资源。

CICS 区域的初始配置将外部安全性设置为 **off**，使用区域定义（RD）中的 **ESMLoad** 属性。当 **ESMLoad** 为 **off** 时，CICS 执行内部安全性检查。若希望覆盖内部安全性检查，则必须修改 ESM 程序，以便程序执行所需要的安全性检查。

修改之后，必须将生成的可执行装入模块置入需要的装入库内，以便为 CICS 系统使用。然后，必须设置 RD 中的 ESM 属性，以使程序可用于区域。

相关信息

第225页的『使用外部安全性管理器』。

CICS 管理参考大全 中的“区域定义 (RD)”。

CICS 应用程序设计指南 中的“翻译、编译和链接编辑 CICS 应用程序”。

使用终端自动安装用户程序

对于使用终端定义 (WD) 的 CICS 定义终端。仅有少数终端项长期存储在 WD 内。当这些终端请求一个 CICS 会话时，CICS 必须自动地创建一个 WD 项，以便 CICS 能够识别终端。CICS 通过从模型列表中选出一个模型名来执行。每个模型在 WD 中都有一项。就是这个项用于创建终端的运行时间 WD 项。

这些自动安装的终端仅当连接到 CICS 时才在运行时间中定义。连接终止时，CICS 自动删除 WD 项。自动安装的特殊优点是，不必要创建并维护一个大型的永久 WD。可自动安装任何不在区域的 WD 中含有项的终端或伪终端。

CICS 能够使用终端自动安装用户程序，以控制将项添加到运行时间 WD 的进程，无论仍未定义到 CICS 的终端请求何时连接到 CICS。

提供的用户程序在合适模型的列表中选择第一个模型，以导出终端标识。可将程序定制为执行任何所需要的处理。例如，可能需要计数并限制在终端上注册的总数，或保存关于终端的使用率信息。

当从 WD 中卸载并删除终端时，也调用用户程序。例如，这可用于更新任何在自动安装时建立的统计数据。

参阅 *CICS 管理参考大全* 中的“终端自动安装用户程序”，以获得更多信息。

使用 IBM cics 客户机产品的自动安装程序

注：本讨论仅适用于来自 IBM cics 客户机产品的自动安装请求，不适用于开放系统上的 CICS 客户程序。IBM cics 客户机产品为：

- IBM CICS 客户机 OS/2 版
- IBM CICS 客户机 DOS 版
- IBM CICS 客户机 Windows 3.1 版
- IBM CICS 客户机 Windows NT 版
- IBM CICS 客户机 Windows 95 版
- IBM CICS 客户机 Macintosh 版

IBM cics 客户机产品允许工作站应用程序与 CICS EPI 和 ECI 通信，以启用到 CICS 服务器的事务路由选择和分布程序链接 (DPL) 调用。对于任何开放系统上的 CICS 区域需要与 IBM cics 客户机产品通信，则通信定义 (CD) 项是必需的。当客户第一次请求与服务器通信时，该 CD 自动安装。

客户可请求一个 CD 项的 **RemoteLUName** 以知的特定名称，（CICS 试图使用该特定名称），或客户可让服务器构造一个名称。无论在何种情况下，在客户安装在区域

的 CD 之前，服务器调用自动安装程序。这将允许取消客户的安装，或更改用来安装的名称，或发生任何其它处理，如客户用法的记录。

当客户与服务器的通信结束时，它请求从 CD 中删除自己。在该阶段中再次调用自动安装，以便允许更新统计数据之类的处理。

本程序的程序定义（PD）是 DFHCCINX。CICS 提供系统设定的 C 程序、cics_client 作为源代码，它们在目录：

```
/usr/lpp/cics/src/samples/ccinx
```

也将它作为可执行文件，提供在下列目录中：

```
/usr/lpp/cics/bin
```

在样本目录中还提供了实用程序生成文件，头文件（cics_client.h）和其它头文件。

系统设定程序简单地允许继续进行客户安装，而不进行任何更改。若希望改变程序，可将样本代码作为自用出口的根据。应将 DFHCCINX PD **PathName** 属性更改为使用新程序，按照程序必须在何处定位的一般规则。

相关信息

第81页的『ECI 和 EPI 客户应用程序』。

参考 *CICS 内部通信指南* 中的下列事宜：

- “写自己的 DFHCCINX 版本”
- “配置 CD 输入项的自动安装”
- “CICS 用户安全性”
- “设置 CICS 区域来使口令流动”

使用性能监控用户程序

通过使用 CICS 管理命令，设置在永久 监控定义（MD）中的 **MonitorStatus** 属性，以指定是否需求性能监控。也可使用 CEMT SET MONITOR 事务设置 **MonitorStatus** 属性。可使用 **cicsupdate** 或 SMIT，以指定 MD 内的 **UserMonitorModule** 属性中的性能监控用户程序名。

CICS 调用程序：

- 在任务启动期间
- 在任务出口期间
- 处理 EXEC CICS ENTER 命令
- 停止用户时钟并将要写的用户数据返回到监控瞬时数据队列中
- 重新启动用户时钟
- 任务最终清除用户数据

注：若 CICS 应用程序服务器在 XA 环境内以异常终止 A147 终止，则将不写监控程序记录。若事务以异常终止 A147 清除，则将写出监控程序记录。

可定制系统设定程序，以执行任何需要的处理。

参阅 *CICS 管理参考大全* 中的“性能监视器用户程序”，以获得更多信息。

— 相关信息 —

第197页的『监控性能』。

CICS 管理参考大全 中的 CEMT 运行时间资源管理。

CICS 管理参考大全 中的“cicsmfmt - 监控数据格式化器”和“监控定义 (MD)”。

使用后初始用户程序

后初始程序是在区域定义 (RD) 中指定的用户编写程序。CICS 自动地依次运行这些程序，这个过程在完成初始化阶段之后，但在允许用户注册之前。还必须为每个后初始程序在程序定义 (PD) 中指定项。

用户编写后初始程序允许您实现系统专用的功能，反映您的专有需求。例如，可能希望从后初始程序内发出 EXEC CICS START 命令，以执行长期运行的后台任务。但是写这些程序时，必须牢记它们在 CICS 普遍可用于用户之前运行。这将其功能施加一些限制。

启动时调用用户编写程序

需要为 CICS 标识后初始程序，它们必须有一个为其定义的 PD 项。使用 **RD StartupProgList** 属性定义它们运行的次序。这些定义描述在 *CICS 管理参考大全* 的“区域定义 (RD)”中。

CICS 在其安装完成之后和允许任何用户注册之前，依次地运行每个后初始程序。

可以该方式最多运行 10 个程序。若该限制对于处理需求太小，则可通过在每个程序内使用 EXEC CICS LINK 和 EXEC CICS XCTL 命令，以便运行更多程序。

写后初始程序指南

当为 CICS 在后初始阶段期间运行写程序时，应该考虑下列因素：

- 所有 CICS 设施（除了直接与终端通信）可用于后初始程序。因为系统间通信含有与其功能相关联的伪终端项，所以不能在区域初始化时间内的后初始程序处理期间运行任何 ISC 功能。
- 必须在 RD 和 PD 内定义所有用户编写程序，以便 CICS 在后初始期间运行。CICS 从 RD 列表中启动的程序对事务处理工作区 (TWA) 没有访问权限。
- 后初始程序不能请求任何在运行中可能逻辑挂起或异常终止任务的服务。
- 后初始程序不能依赖于由间隔控制启动的事务，因为在 CICS 完成后初始阶段前，不可能发生间隔控制的启动。
- 后初始程序不能显示地发出异常终止。并且，必须注意隐式的异常终止，若未编码 EXEC CICS HANDLE CONDITION，则很有可能以您的名义发出异常终止。
- 后初始程序不能发出转储请求。
- 虽然可能在后初始处理期间发出 EXEC CICS START 命令，但是 CICS 将任务排队，且在 CICS 完成后初始处理前不会启动运行这些任务。

相关信息

CICS 管理参考大全 中的“区域定义 (RD)”和“程序定义 (PD)”。

使用关机用户程序

在区域定义 (RD) 中指定 CICS 将要在关机期间运行哪个用户编写程序。CICS 在正常关机处理的第一、二停顿阶段期间，依次地运行这些程序。CICS 在关机过程期间自动地启动这些程序。还必须为每个关机程序在程序定义 (PD) 中创建一项。

关机时调用用户编写程序

CICS 允许两个用户编写列表在正常关机处理的不同阶段自动运行。必须在区域定义 (RD) 中定义这两个列表，并在程序定义 (PD) 中定义每个关机程序。

CICS 处于正常关机处理的第一次停顿阶段，以便运行用户编写程序的第一个列表，第二个列表处于第二次停顿阶段。

在正常关机处理的第一次停顿阶段期间，CICS 自动按顺序运行用户编写关机程序的列表，如 RD 中的 **ShutdownProgList1** 属性所指定。在第二次停顿阶段期间，CICS 自动运行第二个用户编写程序，如 RD 中的 **ShutdownProgList2** 属性所指定。

在每个停顿阶段，以这种方式可最多运行 10 个程序。若该限制对于处理需求太小，则可通过在每个程序内使用 EXEC CICS LINK 和 EXEC CICS XCTL 命令来运行更多程序。

写关机程序指南

当为 CICS 能在正常关机期间运行写程序时，应切记这些程序仅在正常关机的第一次停顿阶段或第二次停顿阶段运行。

第一次停顿阶段

在第一次停顿阶段期间，终端仍是可用的，但是 CICS 拒绝由终端输入启动的事务，除非事务标志在 事务处理定义 (TD) 中。CICS 允许事务由其它方式启动，如自动事务启动 (ATI)。

当 CICS 已经运行列在区域定义 (RD) 中的第一阶段关机程序，并且没有用户任务在运行之中时，则第一次停顿阶段完成。

第二次停顿阶段

在第一次停顿阶段区域活动完成之前，第二次停顿阶段不会开始。关机，然后继续，且不允许启动新事务。当所有定义为将要在第二次停顿阶段运行的程序都完成后，CICS 停止。

注:

1. 指定为第二次停顿阶段的程序不能使用任何终端控制服务，也不能启动新事务。CICS 不允许这种请求，并将一条信息写入系统日志。CICS 不允许任何自动启动的事务在第二次停顿阶段期间启动。
2. 若用户关机程序异常终止，则 CICS 执行同步点重算。
3. 应确保用户关机程序处理所有异常终止条件。

需要关于如何在异常关机后恢复 CICS 的信息，参阅第147页的『第3部分 备份和恢复』。

关机时如何启动新事务

可能需要在 CICS 关机期间启动新事务。则只能在正常关机过程的第一次停顿阶段期间执行。在关机时启动新事务的目的取决于用户的类型和特定系统，但是，范例包括：为收集统计资料启动的事务、从 CICS 区域注销、以及使用 CEMT 清除长期运行任务，这些长期运行任务将引起关机第一次停顿阶段延迟。

关机时启动事务：

- CICS 必须处于正常关机的第一阶段。
- 必须将事务处理定义（TD）项中的 **InvocationMode** 属性设置为 **at_shutdown_start**，以将事务定义为在关机期间可启动，或事务必须为自动启动。

按照一般情况，从终端启动事务。

相关信息

CICS 管理参考大全 中的“程序定义 (PD)”、“事务定义 (TD)”和“区域定义 (RD)”。

第2部分 操作

表 22. 操作的导向图

若需...	参考...
使用 CICS 启动过程。	第123页的『第7章 启动 CICS 系统』。
使用 CICS 关闭过程。	第133页的『第8章 关闭 CICS 系统』。
使用 CICS 命令来更改运行时中的资源定义。	第141页的『第9章 使用 CICS RDO 命令更新区域运行时资源』。

第7章 启动 CICS 系统

本章描述了 CICS 系统是如何启动的。

表 23. 启动 CICS 系统的导向图

若需...	参考...
阅读有关 DCE、文件管理器和区域应该启动的次序的信息。	第123页的『启动顺序』。
阅读有关启动期间发生了什么的信息。	第124页的『冷启动和热启动』。
使用 cicscp 启动 CICS 系统。	第125页的『对于启动使用 cicscp』。
启动 DCE、SFS 和区域。	第126页的『启动过程』。
阅读有关 CICS 如何在各种类型的关闭后自动恢复并重新启动的信息。	第127页的『CICS 启动的恢复和重新启动』。
阅读有关当在启动时服务器不可用会发生什么的信息。	第128页的『在启动时和运行事务时文件管理器不可用』。
阅读有关当更改系统时钟的时候如何做的信息。	第131页的『更改系统时钟的可能影响』。
启动 cicsteld 服务器。	第73页的『用 CICS 使用 telnet 客户』。
启动 cicstermp 服务器。	第84页的『如何为从 CICS 应用程序打印而建立 cicstermp』。

启动顺序

应该以这个次序启动 CICS 系统:

DCE 在最小化时, RPC 精灵程序应该正在运行 (仅 RPC 设置)。如果在 DCE 单元中配置 CICS, 那么 DCE 服务器和 CDS 以及安全性职员也必须正在运行。

文件管理器

文件管理器应该在任何使用它的区域启动以前启动。

本地 SNA

如果一区域使用本地 SNA 支持, 那么 SNA 应该在启动区域以前启动。对于有关使用通信服务器 AIX 版的信息, 转至在 *Using IBM Communications Server for AIX with CICS* 中的“启动 AIX SNA”。也可参阅 *AIX SNA Server/6000: 用户指南*。

PPC 网关服务器

如果任何区域使用 PPC 网关服务器, 那么 PPC 网关服务器必须在使用它的区域启动以前启动。这假设为也已启动了 SNA。如果 PPC 网关服务器正在同时安装有 CICS 的机器上运行, 参阅在 *CICS 内部通信指南*中的“在 CICS 环境中创建和使用 PPC 网关服务器”。如果 PPC 网关服务器在没有安装 CICS 的机器上运行, 参阅 *Encina 管理指南 卷 2: 服务器管理*。

区域 当启动了所有所需的 DCE、文件管理器、PPC 网关服务器和 SNA 组件时, 可启动区域。

本章描述了如何使用 CICS 控制程序 (**cicscp**) 启动 CICS。

另外, 您可使用 SMIT 来启动系统 (参阅第10页的『使用 SMIT 建立 CICS 系统』), 或您可使用 SMIT 和 **cicscp** 下面的命令 (参阅第231页的『附录B. 使用 CICS 命令的 CICS 系统管理过程』)。

相关信息

第21页的『第2章 创建和配置 CICS 区域』。

在 *CICS 管理参考大全* 中的“关于 CICS 命令”。

冷启动和热启动

当冷启动 CICS 时，不进行尝试恢复任何以前正在运行的事务并且丢失存储在瞬时数据队列和临时存储器队列中的任何数据。换句话说，区域启动干净得就象区域是第一次启动一样。

当自动启动 CICS 时，CICS 按它以前在该区域终止的方式以下列方法之一启动：

热 如果 CICS 以前的运行正常终止。CICS 保留存储在瞬时数据队列和临时存储器队列中的任何数据。

应急 如果 CICS 以前的运行异常终止，例如一个立即关闭或一个硬件错误。CICS 执行在该方式下任何必要的恢复处理，并保留在可恢复瞬时队列和临时存储器队列中存储的任何数据，但丢弃存储在不可恢复队列中的数据。

一个区域在它创建后第一次启动时，它必须被冷启动以保证正确的区域初始化。

可在缺省区域或一特定区域（通过指定一区域名称作为启动参数）启动 CICS。

一次仅可启动 CICS 区域的一个实例。可启动如您喜欢一样多的 CICS 区域，不过，CICS 区域的性能取决于整个系统的负载。

如果要运行一拥有特定场所的区域，可在区域的环境文件中设置场所。

```
/var/cics_regions/regionName/environment
```

参阅第34页的『设置区域的语言』。

CICS 初始化完成时，在 CICS 允许用户登记至区域以前，CICS 自动运行任何用户写的站初试化程序。

有关冷启动 SFS 的警告

冷启动 SFS 丢弃了在 SFS 逻辑卷上的所有文件。不仅文件中的数据丢失，而且文件本身也丢失。每次冷启动 SFS 时，这些文件必须重新创建。即，在可使用 SFS 以前包括瞬时数据队列、临时存储器队列、未到期的 START 请求的所有文件以及应用程序数据文件（由 文件定义（FD）定义的文件）。因此，不建议为 SFS 使用冷启动。

如果您必须冷启动 SFS，对于有关重新创建这些文件的信息，参阅第24页的『在 SFS 上建立队列和文件管理』。

相关信息

第117页的『使用后初始用户程序』。

通信定义 (CD) 和 区域定义 (RD)。

第218页的『授权访问启动和关闭程序』提供了有关防止未授权的用户启动或关闭 CICS 区域的信息。

对于启动使用 **cicscp**

cicscp 执行下列许多必需的操作。例如，它：

1. 设置并启动 DCE。
2. 设置开放系统上的 CICS 客户。
3. 为 SFS 创建用户 ID。
4. 为 SFS 创建逻辑卷。
5. 如果文件管理器是 SFS，启动它。
6. 为区域配置文件管理器。
7. 配置 CICS 区域。
8. 启动 CICS 区域。

cicscp 命令的新用户应该总使用 **-v** 和 **-l** 选项，以用于理解和跟踪正发生什么。**-v** 显示了使用的命令的输出并且 **-l logfile** 在 *logfile* 中产生活动的日志。例如，使用 **-v** 和 带有命名为 *regionA* 的日志文件的 **-l** 选项来启动一命名 *regionA* 的区域。

```
cicscp -v -l regionA.log start region regionA
```

如果您需要非缺省配置，那么您可通过使用带有必要覆盖的 **cicscp** 显式创建它。

对于 **cicscp** 命令的全部功能的描述，参阅 *CICS 管理参考大全* 中的“**cicscp** - CICS 控制程序”。

注：如果您正执行的任务需要您作为 DCE 单元管理员注册，那么发生下列事件：

1. **cicscp** 试图使用 *cell_admin_pw* 环境变量定义的值作为口令注册。如果定义了一个值，但它是不正确的，那么 **cicscp** 失败。
2. 如果值没有以 *cell_admin_pw* 环境变量定义，那么 **cicscp** 尝试使用 DCE 单元管理员帐户的缺省口令。如果这是正确的，那么 **cicscp** 作为 DCE 单元管理员注册。如果这是不正确的，那么发生下列事件之一：
 - a. 如果您正交互使用 **cicscp**，那么 **cicscp** 提示要口令。
 - b. 如果您正把 **cicscp** 作为批处理过程使用，那么 **cicscp** 失败。

相关信息

第8页的『使用 `cicscp` 建立 CICS 系统』。

在 *CICS 管理参考大全* 中的“`cicscp` - CICS 控制程序”。

启动过程

该章节描述了如何使用 **`cicscp`** 来启动 CICS 系统。对于替代过程，参阅第10页的『使用 `SMIT` 建立 CICS 系统』或第231页的『附录B. 使用 CICS 命令的 CICS 系统管理过程』。

使用 `cicscp start dce` 启动 DCE

该过程描述了如何使用 **`cicscp start dce`** 命令启动 DCE。启动在本地主机上配置的 DCE 的所有组件。

1. 注册一个有超级用户（`root`）特权的用户标识符，该用户标识符如第3页的『建立 CICS 系统管理员』中描述的那样建立。
2. 若正在 DCE 单元中工作，则作为管理员（`dce_login cell_admin`）注册到 DCE。
3. 使用下列命令：

```
cicscp start dce
```

对于有关使用该命令的更多信息，参阅在 *CICS 管理参考大全* 中的“`cicscp` - DCE 命令”。

使用 `cicscp start sfs_server` 启动 SFS

使用该过程以通过使用 **`cicscp start sfs_server`** 来启动 SFS。

注意：

直到已启动 **DCE** 才启动 **SFS** 。

1. 注册一个有超级用户（`root`）特权的用户标识符，该用户标识符如第3页的『建立 CICS 系统管理员』中描述的那样建立。
2. 若正在 DCE 单元中工作，则作为管理员（`dce_login cell_admin`）注册到 DCE。
3. 使用下列命令：输入：

```
cicscp start sfs_server /./cics/sfs/serverId
```

其中 *serverId* 是该服务器的基本名称。

使用 `cicscp start region` 启动区域

区域依赖于操作的基本组件。因此，应该在启动区域以前启动 DCE 和文件管理器。

可通过打开另一个窗口并发出下列命令来跟踪区域启动序列：

```
tail -f /var/cics_regions/regionName/console.nnnnnn
```


1. 注册一个有超级用户（root）特权的用户标识符，该用户标识符如第3页的『建立 CICS 系统管理员』中描述的那样建立。
2. 若正在 DCE 单元中工作，则作为管理员（dce_login cell_admin）注册到 DCE。
3. 使用下列命令启动区域：

```
cicscp -v start region regionName
```

其中 *regionName* 是区域的名称。

对于有关使用该命令的更多信息，参阅在 *CICS 管理参考大全* 中的“cicscp - 区域命令”。

使用 cicscp start all 启动 CICS

使用该过程来通过使用 **cicscp start all** 启动在本地主机上配置的所有 DCE、Encina 和 CICS 对象。

1. 注册一个有超级用户（root）特权的用户标识符，该用户标识符如第3页的『建立 CICS 系统管理员』中描述的那样建立。
2. 若正在 DCE 单元中工作，则作为管理员（dce_login cell_admin）注册到 DCE。
3. 使用下列命令：输入：

```
cicscp start all
```

对于有关使用该命令的更多信息，参阅在 *CICS 管理参考大全* 中的“cicscp - 所有选项”。

相关信息

第21页的『第2章 创建和配置 CICS 区域』。

第133页的『第8章 关闭 CICS 系统』。

CICS 启动的恢复和重新启动

在冷启动时，区域的启动脱离以前的区域活动而发生。对于自动启动，CICS 按上次关闭的状态启动区域。这将得到下面的结果：

- 冷启动，如果您以前还没有启动区域。
- 热启动，如果区域以正常关闭终止。
- 紧急启动，如果区域以立即关闭终止，或由于异常终止。

在启动时资源定义的恢复

如果您使用自动启动来重新启动 CICS，那么 CICS 恢复最近的资源定义。这意味着 CICS 精确地带有相同的资源而重新启动，而不论终止以前的 CICS 运行所用的方式。如果您希望以您的资源定义的永久设置重新启动 CICS，那么您必须冷启动 CICS。如果您希望以您的资源定义的永久设置重新启动 CICS，那么您不能回复至任何以前的定义。

注：CICS 不恢复通过 CEMT 事务或 EXEC CICS SET 命令进行的任何更改。

CICS 重新启动

如果区域在所有事务已正常完成以前终止，那么当您下一次启动区域时，CICS 必须在正常处理可继续以前，恢复区域的可恢复资源至一个已知的一致状态。

- 如果事务没有结束对于同步点的准备，CICS 丢弃进行的更改（倒退它们）。
- 如果事务准备了同步点，但没有在区域终止时提交同步点，那么 CICS 准备变更并等待逻辑工作单元（LUW）结果。
- 如果事务完成了同步点，那么更改的资源处在一个一致的状态，并且不需要进一步的操作。

如果事务准备成功但没有提交，那么 CICS 在系统日志文件中保存那事务所做的更改，以允许 CICS 在区域异常终止时重新应用这些更改。在恢复期间，CICS 阅读系统日志并重新准备事务。如果所有事件执行正常，CICS 提交更改并使数据处在一个一致的状态。

恢复控制处理

CICS 搜集了下列信息：

- 相关于进行中的逻辑工作单元和任务的信息
- 相关于完成的 LUW 和任务的信息

逆序（现场）恢复处理

CICS 和文件管理器合作倒退进行中事务对下列资源所造成的影响。

- 可恢复瞬时数据队列
- 可恢复临时存储器队列
- 可恢复的数据文件
- EXEC CICS START PROTECT 请求

对于更多信息，参阅第147页的『第3部分 备份和恢复』。

在启动时和运行事务时文件管理器不可用

该章节包含下列表格：

- 第128页的『在 CICS 启动时不可用的文件管理器的影响』
- 第130页的『在各阶段不可用的文件管理器的影响』
- 第131页的『不可用的文件管理器的其它影响』

在 CICS 启动时不可用的文件管理器的影响

下列表格显示了在启动时一个不可用的文件管理器的影响：

表 24. 在 CICS 启动时不可用的文件管理器的影响

文件管理器保存.... 为 不可用	CICS 执行冷启动	CICS 执行自动的启动
一个或一个以上数据文件。	CICS 应已打开的文件被取而代之标记为 CLOSED, UNENABLED。对于每个这样的文件记录消息 ERZ021004E。每个文件可在文件管理器成为可用时, 通过显式使用 SET FILE 或 CEMT 来打开。CICS 继续启动。	同冷启动。
受保护的本地排队的事务 [START PROTECT NOCHECK 保留在文件中]。	CICS 需要清除该文件。当它不能这样做时, 它通过一个自动预定 CICS 专用事务对于这将要做的事设置一个指示器。记录消息 ERZ061013E。CICS 继续启动。	CICS 调度两个 CICS 专用事务: 一个为发送本地排队请求到它们的远程系统, 并且另一个用于清除旧的请求。记录消息 ERZ061013E。这些事务也记录该消息直到文件成为可用。 CICS 继续启动。
非受保护的本地排队事务 [START NOCHECK 保留在文件中]。	同受保护本地排队事务。	同受保护本地排队事务。
逻辑可恢复瞬时数据队列。	CICS 需要清空文件。记录消息 ERZ022132W。当删除了所有以前的项目时, 文件在文件管理器可用后的第一次存取时打开。 CICS 继续启动。	如果自动启动跟在正常关闭后, 那么 CICS 从 LOG 记录恢复它的队列指针 (不需要文件)。不记录消息。文件在以后文件管理器成为可用后的第一次存取时打开。CICS 继续启动。 否则, CICS 需要从文件恢复它的队列指针。记录消息 ERZ022124E 并终止启动。
物理可恢复瞬时数据队列。	同逻辑可恢复瞬时数据队列。	同逻辑可恢复瞬时数据队列。
不可恢复瞬时数据队列。	同逻辑可恢复瞬时数据队列 (除了记录消息 ERZ022123E)。	如果自动启动跟在正常关闭后, 那么同逻辑可恢复瞬时数据队列, 否则同冷启动。
可恢复临时存储器队列 (由用户拥有)。	CICS 需要清空文件。记录消息 ERZ023195W。当删除了所有以前的项目时, 文件在文件管理器可用后的第一次存取时打开。 CICS 继续启动。	如果自动启动跟在正常关闭后, 那么 CICS 从运行记录恢复它的队列指针 (不需要文件)。不记录消息。文件在以后文件管理器成为可用以后的第一次存取时打开。CICS 继续启动。 否则, 同逻辑可恢复瞬时数据队列, 但记录消息 ERZ022372E。
不可恢复临时存储器队列 (由用户拥有)。	同可恢复临时存储器队列 (由用户拥有)。	如果自动启动跟在正常关闭后, 那么同可恢复临时存储器队列 (由用户拥有), 否则同冷启动。
可恢复临时存储器队列 (由计时器服务拥有)。	同用户拥有的可恢复临时存储器队列。	CICS 需要从文件中保存的记录重建它的 ICQ。如果自动启动跟在正常关闭后, 那么文件不能打开并且 CICS 延迟该工作直至以后。如果文件可打开, 则一个或一个以上记录可能不能阅读 (因为它们等待事务解决)。 CICS 记下这样的记录并在以后从它们重建 ICQ 项。 CICS 继续启动。 如果自动启动跟在正常关闭后, 那么同可恢复临时存储器队列 (由用户拥有)。

表 24. 在 CICS 启动时不可用的文件管理器的影响 (续)

文件管理器保存.... 为 不可用	CICS 执行冷启动	CICS 执行自动的启动
不可恢复临时存储器队列（由计时器服务拥有）。	同用户拥有的可恢复临时存储器队列。	如果自动启动跟在正常关闭后，那么 CICS 需要从该文件恢复非受保护的 START 请求的 ICQ，否则同冷启动。 如果文件管理器不可用，则记录消息 ERZ023103E 并终止启动。如果文件可用但不能阅读，记录消息 ERZ023104E 并且启动继续。
可恢复临时存储器队列（由终端服务拥有）。	同可恢复临时存储器队列（由用户拥有）。	如果自动启动跟在正常关闭后，那么 CICS 可能需要恢复一些自动安装的终端。这些终端的定义驻留在保存于文件中的可恢复临时存储器队列中，所以行为同可恢复 TS 队列（由用户拥有）。如果自动启动跟在正常关闭后，那么不恢复自动安装的终端，并且不需要该信息。

在各阶段不可用的文件管理器的影响

下列表格显示了在 CICS 处理的各阶段一个不可用的文件管理器的影响：

表 25. 在 CICS 处理的各阶段一个不可用的文件管理器的影响

文件管理器保存.... 不可用	事务先于同步点需要资源	事务在同步点期间需要资源
一个或一个以上数据文件。	取决于以前在 LUW 中的处理，或者提高条件码并设置 EIBRCODE 为文件管理器错误，或者异常终止发生。记录消息 ERZ022012E。CICS 继续，但在文件管理器上的文件变成不可用直至它返回。	这里所有两阶段提交活动由文件管理器负责。事务可提交或异常中止，这取决于在两阶段提交期间不可用性的时间选择。CICS 不标记文件为不可用的，或记录任何消息。CICS 继续，但事务可能阻塞。（不可恢复的文件完全不包含在两阶段提交中。）
受保护的本地排队的事务 [START PROTECT NOCHECK 保留在文件中]。	这意味着正在执行一个启动远程区域上事务的 START PROTECT NOCHECK，但远程区域是不可用的，并且事务的 TD 项显示了请求将于本地排队。NOCHECK 选项的语义表示文件管理器的不可用性不返回给 START 命令，但记录消息 ERZ061005E，并且 LUW 在同步点滚回。CICS 继续，但受保护的本地排队暂时成为不可用。	所有两阶段提交活动由文件管理器负责。事务可提交或异常中止，这取决于在两阶段提交期间不可用性的时间选择。CICS 不记录任何消息。CICS 继续，但事务可能阻塞。
非受保护的本地排队事务 [START NOCHECK 保留在文件中]。	同受保护本地排队事务（除了 PROTECT 选项不存在，并且 LUW 因此不滚回）。忽略 START 请求。	不可适用的 - 该 START 不可恢复，并且因此没有加入到两阶段提交中。
逻辑可恢复瞬时数据队列。	取决于以前在 LUW 中的处理，或者提高条件码并设置 EIBRCODE 为文件管理器错误，或者异常终止发生。记录消息 ERZ022012E。CICS 继续，但逻辑可恢复瞬时数据队列变成不可用直到文件管理器返回。	事务可提交或异常中止，这取决于在两阶段提交期间不可用性的时间选择。CICS 继续，但事务可能阻塞。
物理可恢复瞬时数据队列。	同逻辑可恢复瞬时数据队列，（除了记录消息 ERZ022011E）。	不可适用的 - 文件管理器处理是非事务的。

表 25. 在 CICS 处理的各阶段一个不可用的文件管理器的影响 (续)

文件管理器保存.... 不可用	事务先于同步点需要资源	事务在同步点期间需要资源
不可恢复瞬时数据队列。	同逻辑可恢复瞬时数据队列。	不可适用的 - 瞬时数据队列是不可恢复并且因此不加入到两阶段提交中。
可恢复临时存储器队列（由用户拥有）。	取决于以前在 LUW 中的处理，或者提高条件码并设置 EIBRCODE 为文件管理器错误，或者异常终止发生。记录消息 ERZ023097E。CICS 继续，但可恢复临时存储器队列变成不可用直到文件管理器返回。	事务可提交或异常中止，这取决于在两阶段提交期间不可用性的时间选择。CICS 继续，但事务可能阻塞。
不可恢复临时存储器队列（由用户拥有）。	同可恢复临时存储器队列（由用户拥有）。	不可适用的 - 临时存储器队列是不可恢复并且因此不加入到两阶段提交中。
可恢复临时存储器队列（由计时器服务拥有）。	取决于以前在 LUW 中的处理，或者提高条件码并设置 EIBRCODE 为文件管理器错误，或者异常终止发生。记录消息 ERZ007033E。CICS 继续，但 START PROTECT 变成不可用直至文件管理器返回。	事务可提交或异常中止，这取决于在两阶段提交期间不可用性的时间选择。CICS 继续，但事务可能阻塞。
不可恢复临时存储器队列（由计时器服务拥有）。	同可恢复临时存储器队列（由计时器服务拥有）。	不可适用的 - START 为非受保护的并且因此没有加入到两阶段提交中。
可恢复临时存储器队列（由终端服务拥有）。	不可适用的 - CICS 仅当终端开始工作和停止，以及在启动和关闭时访问临时存储器队列。	不可适用的。

不可用的文件管理器的其它影响

下列表格显示了在 CICS 处理的各阶段其它不可用文件管理器的影响:

表 26. 其它文件管理器的不可用性的影响

不可用的文件管理器	在启动时（冷的和自动的）	在正常运行期间不存在的影响
PPC 网关服务器。	CICS 启动，但使用那网关的 SNA 对话是不可能的。管理员可随后启动网关，从 CICS 运行时删除它的定义，并且然后再次安装它，从而重新配置网关。对于更多信息，参阅在 CICS 内部通信指南 中的“内部通信的配置”。	当前运行的 SNA 对话异常终止。您可简单地再次引入网关备份并且 CICS 继续使用它而无须管理员的介入。对于更多信息，参阅在 CICS 内部通信指南 中的“内部通信的配置”。
DCE 安全性服务器	CICS 无法认证它的用户并因此无法启动。	CICS 鉴定远程过程调用（RPCs）不成功。区域可能异常终止。

如果有未解决的非受保护的本地启动请求，那么在正常区域关闭期间对于文件管理器的可用性有新的请求。这些记录在区域正在运行时保存在主存储器中，但在关闭时传送给文件管理器，这样可在随后的自动启动时检索它们。如果文件管理器在这时不可用，则记录消息 ERZ023102E 并且区域异常终止。

更改系统时钟的可能影响

一些 CICS 产品存储它们自己的系统时钟（在 CICS 启动时设置）。如果有这些 CICS 产品，您必须重新启动 CICS 以更改系统时钟。不过，这不适用于开放系统上的 CICS。

CICS 总查看操作系统时钟以获取时间。如果您更新操作系统的操作系统时钟，倘若适当的线程为活动的，则 CICS 立即拾取更新的时间。如果适当的线程不为活动的，则 CICS 不知道操作系统时钟已更新。要保证线程为活动的，您可运行 EXEC CICS START 命令。

您可写一事务来发出 EXEC CICS START NOW 命令并做一些无害的事。每次更改系统时钟，您可运行该事务来保证 CICS 可立即识别新的时间。

如果正使用 DCE，更改系统时钟可能中断安全性服务器的操作。

操作系统上的时钟可能因为偏差而需要调整，并且它们可由操作员人工或由网络时间服务例如 DCE 分布式时间服务来进行调整。如果正使用 DTS，不建议人工更改系统时钟。

本地的时间区，时间从冬天变更至夏天，并且更改的日期是由操作系统时间区环境变量控制的。这是在启动区域以前设置的，并且使区域能正确执行跨越冬天和夏天的时间更改而不需要重新启动。

您需要知道可能因为时间变更而发生的问题，例如：

从冬天时间（标准时间）更改至夏天时间（夏令时间）

在 18:00，事务以一段时间间隔发出一条 EXEC CICS START 命令，这样被启动的事务将在 04:00 运行。在 02:00，操作系统更改系统时钟至 03:00。启动的事务将在 05:00 而不是 04:00 运行。

从夏天时间更改至冬天时间

在 18:00，事务以一段时间间隔发出一条 EXEC CICS START 命令，这样被启动的事务将在 04:00 运行。在 02:00，操作系统更改系统时钟至 01:00。启动的事务将在 03:00 而不是 04:00 运行。

第8章 关闭 CICS 系统

本章描述了 CICS 关闭进程和过程。

表 27. 关闭 CICS 系统的导向图

若需...	参考...
阅读有关 DCE、文件管理器和区域所应该关闭的次序的信息。	第133页的『关闭顺序』。
阅读有关不同类型的 CICS 关闭进程的信息。	第134页的『CICS 关闭进程』。
关闭 DCE、SFS 或区域。	第137页的『关闭过程』。

关闭顺序

区域依赖于操作的基本组件。因此，您不应该关闭那些基本组件直到关闭了区域。关闭顺序应该为：

1. 区域
2. PPC 网关服务器（如果一个正在运行）
3. SFS（但不在关闭区域或 PPC 网关服务器前）
4. DCE（但不在关闭 SFS 以前）

CICS 控制程序的 **cicscp stop all** 命令以适当的顺序关闭组件。参阅在 *CICS 管理参考大全* 中的“cicscp - CICS 控制程序”。

注：如果您正在执行的任务需要您作为 DCE 单元管理员注册，那么发生下列事件：

1. **cicscp** 试图使用以 cell_admin_pw 环境变量定义的值作为口令注册。如果定义了一个值，但它是不正确的，那么 **cicscp** 失败。
2. 如果值不是以 cell_admin_pw 环境变量定义的，那么 **cicscp** 尝试使用 DCE 单元管理员帐户的缺省口令。如果这是正确的，那么 **cicscp** 作为 DCE 单元管理员注册。如果这是不正确的，那么发生下列事件之一：
 - a. 如果您正交互使用 **cicscp**，那么 **cicscp** 提示要口令。
 - b. 如果您正把 **cicscp** 作为批处理过程使用，那么 **cicscp** 失败。

相关信息

第248页的『使用 `cicssfsshut` 关闭 SFS』。

对于有关关闭 PPC 网关服务器的信息，请参阅在 *CICS 内部通信指南* 中的“在 CICS 环境中创建和使用 PPC 网关服务器”。

CICS 关闭进程

有三种类型的用户请求关闭：

- 正常关闭（缺省关闭类型）
- 立即关闭
- 取消（强制异常）关闭

CICS 也可关闭它自己（软件启动）或由于掉电或其它系统故障不可控制地关闭。

注：如果 CICS 在关闭期间挂起并且没有用户进程未完成，如在 *CICS 管理参考大全* 中的“`cicsnotify` - 释放资源”中所述使用 **`cicsnotify`** 命令。如果这无帮助，那么联系 你的支持组织。

正常关闭

正常关闭发生在如下面所述的 **停顿阶段**中：

在终止的第一停顿阶段期间

所有终端为活动的，所有 CICS 设施为可用的并且 CICS 如下所述运行任务和程序：

- 允许下列任务完成：
 - 已正在运行的任务
 - 将自动启动的任务，假设它们可先于第二停顿阶段启动
- CICS 运行任何在 区域定义（RD）中第一关闭程序列表中列出的用户写的程序。
- 当且仅当事务已标志在事务处理定义（TD）中以允许在关闭时启动时，CICS 允许由终端输入所请求的新事务启动。

注：当安装区域时，CICS 最初标志 CICS 提供的事务 CEMT 和 CSSF 为在关闭期间可启动。

当在第一关闭程序列表中的程序中的最后一个已运行并且完成了所有用户任务时完成第一停顿阶段。

注：长运行任务（例如会话式任务）必须在 CICS 终止可进行以前终止。

在终止的第二停顿阶段期间

在该阶段期间：

- 终端为不活动的。
- CICS 不允许任何新的事务启动。

- CICS 运行任何在区域定义（RD）中的第二关闭程序列表中列出的用户写的程序（如果有的话）。这些程序不能与终端通信或进行任何将引起启动新任务的请求。

第二停顿阶段当在第二关闭程序列表中列出的最后一个程序已结束运行时结束。

- CICS 把统计资料写至在 RD 中定义的统计资料文件中。
- CICS 关闭所有文件。
- CICS 把用于随后的自动启动的信息写至重新启动文件中。
- CICS 停止运行。

如果已在 RD 中设置 **SysDump** 属性为 **yes**，则 CICS 生成一个系统转储。CICS 不总是完成转储进程。如果 **SysDump** 永久地设置为 **yes**，那么 CICS 总在关闭时产生一个转储。这些转储是非常大的，并且您正常地将不在关闭时需要一个转储。

立即关闭

当请求 CICS 的立即关闭时，处理在以下重要方式上与正常关闭不同：

- CICS 不等待用户任务完成。
- CICS 不运行任何在区域定义（RD）中定义的关闭程序列表中的程序。
- CICS 不把信息写至重新启动文件。

注：CICS 不等待用户任务完成，但强迫终止运行的任务。CICS 对当任务终止时处在中间事务的任务运行动态事务复原，并然后关闭资源。

如果已在 RD 中设置 **SysDump** 属性为 **yes**，则 CICS 尝试生成系统转储。CICS 不总完成转储进程。

对于有关在异常关闭后如何恢复 CICS，参阅第147页的『第3部分 备份和恢复』。也可参阅第137页的『关闭过程』和第247页的『从命令行关闭区域』。

取消（强制异常）关闭

如第137页的『关闭过程』和第247页的『从命令行关闭区域』中所述，当请求 CICS 的一个取消（强制异常）关闭时，区域不带有处理地强迫关闭。当且仅当系统在关闭期间逆转或不能清除任务并正常关闭时使用这种关闭。

如果使用一个取消（强制异常）关闭，您也必须使用 **cicsnotify** 命令来释放任何还没有解除分配的资源，并删除 *region_lock* 文件。

对于有关在异常关闭后如何恢复 CICS，参阅第147页的『第3部分 备份和恢复』。

软件启动关闭

当 CICS 检测到它的内部数据结构已被无效数据覆盖，或当一些外部的事件，例如发送 SIGKILL 信号，引起关键的 CICS 进程被破坏时，该类型的关闭发生。

由软件引起的 CICS 终止：

- 不总完成用户任务
- 不运行在区域定义（RD）中列出的关闭程序
- 不把信息写至重新启动文件中
- 允许进行转储，如果在 RD 中的 **SysDump** 属性设置为是
- 不总关闭打开文件

对于有关在异常关闭后如何恢复 CICS，参阅第147页的『第3部分 备份和恢复』。

未控制关闭

CICS 的未控制终止可由以下原因引起：

- 掉电
- 处理器检查
- 操作系统错误

在以上每种情况下，CICS 不能执行任何关闭处理。特别地，CICS 不把任何信息写至重新启动文件中。

对于有关在异常关闭后如何恢复 CICS，参阅第147页的『第3部分 备份和恢复』。

在关闭时产生的统计资料

CICS 管理模块控制如何在 CICS 中管理事件。当事件发生时，CICS 以系统统计资料的格式产生对您有用的信息。统计资料显示 CICS 如何管理在它的控制下的资源。这些资源包括文件、日志、事务、程序和任务。

CICS 在每次正常关闭和立即关闭时产生关闭统计资料。关闭统计资料报告先前的 CICS 启动和它关闭之间的系统活动。可使用关闭统计资料以用于记录走向和故障点。可使用 **cicssfnt** 程序打印该报告，但注意 **cicssfnt** 是一个应该按您自己需要修改的样本程序。

对于 CICS 要记录关闭统计资料，您必须在区域定义（RD）中指定一个 CICS 定义文件或一个用户定义文件。这确保了 CICS 有一个发送记录数据的地方。CICS 发送关闭统计资料至该文件中。

打印软件启动的关闭转储

大多数软件启动的终止产生有关可帮助您处理问题的终止信息。信息源如下：

- console.nnnnnn
- CSMT.out 文件
- symrecs 文件
- 转储文件

可通过使用 **cicsdfnt** 脱机实用程序来格式化并打印转储文件。该程序读取转储文件并通过使用保存在转储文件控制块中的格式描述以及提供给程序的变量来格式化它们。

如果不能从转储信息确定软件启动的终止的原因，您应该联系您的支持组织。

相关信息

第21页的『第2章 创建和配置 CICS 区域』。

第118页的『使用关机用户程序』。

第193页的『CICS 统计的信息收集』描述了如何获取并确定关闭统计路径。

第124页的『冷启动和热启动』描述了您可用来启动 CICS 的不同的方法和选项。

第137页的『关闭过程』描述了您可用来关闭 CICS 区域的方法。

在 *CICS 应用程序设计参考大全* 中的“INQUIRE 和 SET 命令”。

在 *CICS 应用程序设计指南* 中的“调试服务”提供了有关生成、命名、格式化并打印关闭转储的信息。在 *CICS 管理参考大全* 中：

“事务定义 (TD)” 和 “程序定义 (PD)”

“终端定义 (WD)” 描述了如何能标志一事务为可在关闭期间启动。

“区域定义 (RD)” 描述了如何能指定在关闭时 CICS 运行的程序列表。

“cicssfmt - 统计格式化器” 描述了 CICS 提供的统计资料形成工具的功能和用途。

关闭过程

该章节描述了如何使用 **cicscp** 来关闭 CICS 系统。对于替代过程，参阅第10页的『使用 SMIT 建立 CICS 系统』或 第231页的『附录B. 使用 CICS 命令的 CICS 系统管理过程』。

使用 **cicscp stop region** 关闭区域

区域依赖于操作的基本组件。因此，您应该直到区域关闭才关闭 DCE 或文件管理器。

正常地，您不将关闭文件管理器或 DCE，因为可以启动或停止区域而不停止这些组件。当 SFS 和 DCE 由其它区域或非 CICS 应用程序共享时这是特别重要的。

可通过打开另一个窗口并发出下列命令来跟踪区域关闭序列：

```
cicstail -r /var/cics_regions/regionName/console.nnnnnn
```

1. 注册一个有 root 特权并如第3页的『建立 CICS 系统管理员』中所述而设置的用户标识符。
2. 若正在 DCE 单元中工作，则作为管理员（dce_login cell_admin）注册到 DCE。
3. 使用下列命令关闭区域：

```
cicscp -v stop region regionName
```

其中 *regionName* 为区域名称。

对于有关使用该命令的更多信息，参阅在 *CICS 管理参考大全* 中的“cicscp - 区域命令”。

使用 **cicscp stop sfs_server** 关闭 SFS

使用该过程以通过使用 **cicscp stop sfs_server** 来关闭 SFS。

注意:

不要关闭 **SFS** 直到使用 **SFS** 的所有区域已首先被关闭。

1. 注册一个有超级用户（root）特权的用户标识符，该用户标识符如第3页的『建立 CICS 系统管理员』中描述的那样建立。
2. 若正在 DCE 单元中工作，则作为管理员（dce_login cell_admin）注册到 DCE。
3. 使用下列命令:

```
cicscp stop sfs_server ./cics/sfs/serverId
```

其中 *serverId* 是该服务器的基本名称。

使用 **cicscp stop dce** 关闭 DCE

该过程描述了如何使用 **cicscp stop dce** 命令关闭 DCE。

注意:

该过程关闭在本地主机上的 **DCE** 配置，包括 **DCE** 服务器（如果它们是本地的话）。不要关闭 **DCE** 直到您已首先关闭所有的区域、**Encina** 服务器或任何其它使用您将关闭的组件的应用程序。

1. 注册一个有超级用户（root）特权的用户标识符，该用户标识符如第3页的『建立 CICS 系统管理员』中描述的那样建立。
2. 若正在 DCE 单元内工作，则作为单元管理员（dce_login cell_admin）注册到 DCE。
3. 使用下列命令:

```
cicscp stop dce
```

该命令停止了在本地主机上配置的 **DCE** 的所有组件。如果它们在远程主机上，则它不停止 **DCE** 服务器。

对于有关使用该命令的更多信息，参阅在 *CICS 管理参考大全* 中的“cicscp - DCE 命令”。

使用 **cicscp stop all** 关闭 CICS

使用该过程来通过使用 **cicscp stop all** 关闭在本地主机上的所有 **DCE**、**Encina** 和 **CICS** 对象。

注意:

如果 **DCE** 服务器在本地主机上，确保在使用该过程以前所有的区域、**Encina** 服务器以及在单元中的其它应用程序被首先关闭。如果任何 **Encina** 服务器在本地主机上，确保在使用该过程以前使用服务器的所有区域被首先关闭。

1. 注册一个有超级用户（root）特权的用户标识符，该用户标识符如第3页的『建立 CICS 系统管理员』中描述的那样建立。

2. 若正在 DCE 单元中工作，则作为管理员（dce_login cell_admin）注册到 DCE。
3. 使用下列命令：

cicscp stop all

对于有关使用该命令的更多信息，参阅在 *CICS 管理参考大全* 中的“cicscp - 所有选项”。

相关信息

第21页的『第2章 创建和配置 CICS 区域』。

第123页的『第7章 启动 CICS 系统』。

第9章 使用 CICS RDO 命令更新区域运行时资源

本章描述了如何使用 CICS RDO 命令更新运行时中的资源定义。

表 28. 使用 CICS 命令更新区域运行时资源的导向图

如果要...	参阅...
查找使用哪一个过程。	第141页的『如何在运行区域更新资源定义』。
把一个新的定义添加至运行区域。	第142页的『把新的资源定义添加至运行区域』。
在运行时中更改定义。	第143页的『更改运行时中的资源定义』。
从永久数据库安装一个定义。	第143页的『从永久数据库安装一个资源定义』。
仅从运行时删除一个定义。	第144页的『仅从运行时删除资源定义』。
安装一组定义。	第145页的『安装一组定义至运行时中』。

如何在运行区域更新资源定义

当冷启动一个区域时，新的自动启动和运行时数据库并发地从永久数据库中的资源定义的子集安装。该子集包括：

- 那些 **ActivateOnStartup** 设置为 **yes** 的定义。
- 那些 **Group** 属性设置为在区域定义（RD）中的 **Groups** 属性中所列出的组名的定义。例如，如果事务的 **Group** 属性设置为 **Sales**，并且用区域的 **Groups** 属性定义的组的列表包含 **Sales**，则当冷启动区域时在运行时中安装那事务的定义，而不管事务的 **ActivateOnStartup** 属性是否设置为 **yes**。

有几个使用 CICS RDO 命令在运行区域更改资源定义的方法：

- 如果资源定义不存在于永久数据库，那么您可使用带 **-B** 选项的 **cicsadd** 命令来把它添加至永久数据库以及运行时中。按照第142页的『把新的资源定义添加至运行区域』。
- 要更改在运行时中安装的资源定义，您必须首先把它从运行时删除，然后在永久数据库中进行更改并把它安装回运行时中。按照第143页的『更改运行时中的资源定义』。
- 如果资源用 **ActivateOnStartup=no** 定义并且它不属于定义至区域的组，则当冷启动区域时，不安装它至运行时中。要安装它至运行时中，按照第143页的『从永久数据库安装一个资源定义』。
- 要仅从运行时删除资源定义，按照第144页的『仅从运行时删除资源定义』。
- 要安装或重新安装一组把 **Group** 属性值设置为组名的定义，或重新安装把 **ActivateOnStartup** 属性设置为 **yes** 的所有定义，按照第145页的『安装一组定义至运行时中』。

相关信息

第4页的『定义资源』。

第23页的『定义可选择安装的资源』。

在 *CICS 管理参考大全* 中的 CEMT 运行时资源管理。

在 *CICS 应用程序设计参考大全* 中的 “INQUIRE 和 SET 命令”。

通过使用 CICS RDO 命令的运行时资源数据库过程

该主题包含了更新在区域运行时中的资源定义的过程。按照在 第141页的表 28 中的指导使用这些过程。

把新的资源定义添加至运行区域

使用该过程来把新的资源定义添加至永久、自动启动和运行时数据库。不能在区域定义（RD）或监控定义（MD）上使用该过程。

1. 如果正在一个 DCE 单元上以一些 DCE 保护级别进行工作，那么作为单元管理员（dce_login cell_admin）注册至 DCE。
2. 使用下列命令：

```
cicsadd -c class -r regionName -B key \  
        attribute=value \  
.  
.  
.  
        attribute=value
```

其中 *key* 是资源定义项的主要名称并且 *class* 为下列之一：

cd	通信定义（CD）
fd	文件定义（FD）
gd	网关定义（GD）
jd	日志定义（JD）
ld	侦听程序定义（LD）
pd	程序定义（PD）
td	事务处理定义（TD）
tdd	瞬时数据定义（TDD）
tsd	临时存储器定义（TSD）
ud	用户定义（UD）
wd	终端定义（WD）
xad	产品定义（XAD）

如果因为一些缘故，未能更新运行时数据库（例如，若区域未在运行之中），则也不更新自动启动数据库。不过，将更新永久数据库。

如果这发生，修正问题（例如，启动区域），然后按照第143页的『从永久数据库安装一个资源定义』继续操作。

更改运行时中的资源定义

使用该过程来更改永久、自动启动和运行时数据库中的资源定义。不能在区域定义（RD）或监控定义（MD）上使用该过程。

1. 如果正在一个 DCE 单元上以一些 DCE 保护级别进行工作，那么作为单元管理员（dce_login cell_admin）注册至 DCE。
2. 使用下列命令来从运行时中删除资源定义项：

```
cicsadd -c class -r regionName -R key
```

其中 *key* 是资源定义项的主要名称并且 *class* 为下列之一：

cd	通信定义（CD）
fd	文件定义（FD）
gd	网关定义（GD）
jd	日志定义（JD）
ld	侦听程序定义（LD）
pd	程序定义（PD）
td	事务处理定义（TD）
tdd	瞬时数据定义（TDD）
tsd	临时存储器定义（TSD）
ud	用户定义（UD）
wd	终端定义（WD）
xad	产品定义（XAD）

3. 使用下列命令来更改永久数据库中的资源定义项并在运行时中安装它：

```
cicsupdate -c class -r regionName  
-B key \  
    attribute=value \  
    .  
    .  
    .  
    attribute=value
```

从永久数据库安装一个资源定义

使用该过程来在冷启动区域时从永久数据库安装一个资源定义至没有安装的运行时中。该过程允许您更改永久数据库中的资源定义（在安装它至运行时中以前）。

不能在区域定义（RD）或监控定义（MD）上使用该过程。

1. 如果正在一个 DCE 单元上以一些 DCE 保护级别进行工作，那么作为单元管理员（dce_login cell_admin）注册至 DCE。

2. 要添加项目而不更改它，使用下列命令：

```
cicsupdate -c class -r regionName -B key
```

其中 *key* 是资源定义项的主要名称并且 *class* 为下列之一：

cd	通信定义 (CD)
fd	文件定义 (FD)
gd	网关定义 (GD)
gd	网关定义 (GD)
jd	日志定义 (JD)
ld	侦听程序定义 (LD)
pd	程序定义 (PD)
td	事务处理定义 (TD)
tdd	瞬时数据定义 (TDD)
tsd	临时存储器定义 (TSD)
ud	用户定义 (UD)
wd	终端定义 (WD)
xad	产品定义 (XAD)

3. 如果要当把资源定义添加到自动启动和运行时数据库时更改它，使用下列命令：

```
cicsadd -c class -r regionName  
-B key \  
    attribute=value \  
    .  
    .  
    .  
    attribute=value
```

如果因为一些缘故，未能更新运行时数据库（例如，若区域未在运行之中），则也不更新自动启动数据库。

如果这发生，修正问题（例如，启动区域），然后再次做该过程。

仅从运行时删除资源定义

使用该过程来从运行时删除资源定义。不能在区域定义 (RD) 上使用该过程。

1. 如果正在一个 DCE 单元上以一些 DCE 保护级别进行工作，那么作为单元管理员 (dce_login cell_admin) 注册至 DCE。
2. 使用下列命令：

```
cicsdelete -c class -r regionName  
-R key
```

其中 *key* 是资源定义项的主要名称并且 *class* 为下列之一：

cd	通信定义 (CD)
fd	文件定义 (FD)
gd	网关定义 (GD)
jd	日志定义 (JD)

ld	侦听程序定义 (LD)
pd	程序定义 (PD)
td	事务处理定义 (TD)
tdd	瞬时数据定义 (TDD)
tsd	临时存储器定义 (TSD)
ud	用户定义 (UD)
wd	终端定义 (WD)
xad	产品定义 (XAD)

安装一组定义至运行时中

使用该过程来从永久数据库安装一组资源定义至运行时中。要设置将安装的组，参阅第23页的『定义可选择安装的资源』。

1. 如果正在一个 DCE 单元上以一些 DCE 保护级别进行工作，那么作为单元管理员 (dce_login cell_admin) 注册至 DCE。
2. 要安装已带有 **Group** 属性设置为组名的资源定义，使用下列命令：

```
cicsinstall -r regionName -g groupName
[groupName...]
```

3. 要重新安装有 **ActivateOnStartup** 属性设置为 **yes** 的资源定义，使用下列命令：

```
cicsinstall -r regionName -a
```

注：该过程覆盖用 CEMT 或 EXEC CICS SET 对这些资源定义进行的任何变更。

第3部分 备份和恢复

表 29. 备份与恢复的导向图

若需...	参考...
在 CICS 环境中找出有关的备份和恢复。	第149页的『第10章 CICS 备份和恢复介绍』。
准备系统恢复。	第153页的『第11章 为 CICS 系统的恢复做准备』。
备份或恢复 CICS 系统。	第167页的『第12章 备份和恢复的实现』。

第10章 CICS 备份和恢复介绍

本章论述了一些方法，可以利用它们防止 CICS 区域配置和商业数据因无法控制的系统关机而造成的丢失或损坏。它同样也论述了一些由 CICS 提供的，用来帮助完成这些操作的机构。

— 相关信息 —

第127页的『CICS 启动的恢复和重新启动』。

在 *CICS 应用程序设计指南* 中：

- “动态撤消及恢复”
- “恢复”
- “恢复的编码思想”

在 *CICS 内部通信指南* 中：

- “用同步化支持保证数据完整性”。
- “写内部通信的应用程序”。

保护 CICS 系统和数据

要防止系统和数据因无法控制的关机而造成丢失或损坏，并且允许迅速恢复服务，则须实现备份和恢复策略。

典型地，备份和恢复策略需要：

- 一旦已为 CICS 配置了 DCE，则需备份 DCE。
- 定义为包含重要数据（可由 CICS 事务更改）的可恢复资源。
- 备份永久资源定义数据库中的资源定义。
- 确保已为恢复而编写了可在区域中更新数据的应用程序（这在 *CICS 应用程序设计指南* 中已有说明）
- 备份应用程序数据
- 备份事务更新的日志

协助备份、恢复和重新启动的机构

CICS 和文件管理器提供下列机构，它们可用于备份和恢复策略中：

- 逻辑工作单元（LUW）
- 可恢复资源
- 动态事务复原逆序（现场）恢复
- CICS 日志
- 用户日志
- 自动启动
- 事务日志

- 转发恢复

同步点 (sync points)

逻辑工作单元的末端由同步点 (通常缩写为 *sync point*) 向 CICS 指出。

同步点可由下列方法得到:

- 隐含地, 在事务的末端, 由最高逻辑级别上的 EXEC CICS RETURN 命令标记。这表明逻辑工作单元不能跨任务。
- 明显地, 则由应用程序 (位于事务中逻辑上恰当的点处) 发出的 EXEC CICS SYNCPOINT 命令标记。

它由 LUW 启动的地方延续:

- 在任务开始处
- 无论何时事务发出一显式 EXEC CICS SYNCPOINT 命令

没有更改可恢复资源的 LUW 则没有有意义的结果, 无论是对您或 CICS 恢复机构来说都是如此。CICS 没有落实任务的资源更改, 直到 CICS 到达同步点。如果因为任何类型的错误导致 CICS 中断任务处理, CICS 逆序 (现场) 恢复在异常终止 LUW 期间所做的更改。

两阶段落实处理

要提供 ACID 特性, CICS 实现了一个称为 **两阶段落实处理** 的协议。当落实一个 LUW 时, CICS 需要保证所有在 LUW 中的可恢复服务器都落实它们的更新; 如果一个或一个以上 (服务器) 不能做到这点 (例如, 到服务器的通信线路中断), 则所有服务器都必须逆序 (现场) 恢复它们的更新。为了完成上述操作, 提交过程有两个阶段。在第一个阶段, CICS 要求每个参与者轮流记录足够的关于 LUW 的信息, 以便 CICS 在第二阶段落实或逆序恢复更新。这称为准备 LUW。每个参与者可以用三种方法之一来响应 CICS:

yes 参与者要落实或逆序恢复, 如上所示。

no 参与者要逆序恢复 LUW。

read only 参与者不关心这个 LUW。

如果参与者以 **yes** 响应,但落实或逆序恢复还没有发生, 则参与者具有状态 **indoubt**。

如果没有响应为 **no**, 则 CICS 指示所有参与者去落实 LUW。如果任何响应都为 **no**, 则 CICS 指示所有参与者逆序恢复。这是两阶段过程的第二阶段。

CICS 可恢复资源

要保证商业数据根据 ACID 委托人而更新, 则必须定义资源, (在当中的数据存储为 **可恢复资源**)。

可定义为可恢复的资源是:

- 数据文件
- 内分区瞬时数据 (逻辑可恢复)
- 辅助临时存储器存储器队列

- 自动安装终端定义
- EXEC CICS 启动保护请求

在 *CICS 管理参考大全* 中，参阅

- “文件定义 (FD)” 中的 **RecoveryStatus** 属性说明
- “瞬时数据定义 (TDD)” 中的 **RecoveryType** 属性说明
- “临时存储器定义 (TSD)” 中的 **RecoveryFlag** 属性说明

在正常的操作和关机期间，CICS 以及用来和记录信息通信的可恢复服务器都请求重新启动。

如果有一完全或局部的错误、或者是关机，则该信息允许 CICS 恢复并重新启动区域。在自动重新启动期间，CICS 将可恢复资源返回它们的落实状态。

如果事务异常终止，并且终止任务影响的资源可恢复，则 CICS 自动执行动态事务复原。然而，在动态事务复原后，CICS 并不自动重新启动事务。除非文件在 DB2 数据库（这里所有文件都可恢复）中，CICS 不会逆序恢复对没有定义为可恢复的资源所做的更改。

自动启动

可以使用 **StartType=auto** 属性在故障后启动 SFS。要成功地重新启动 SFS，必须要有 CICS 日志。如果没有 CICS 日志，则需要 SFS 的冷启动。

相关信息

第127页的『CICS 启动的恢复和重新启动』。

第128页的『在启动时和运行事务时文件管理器不可用』。

第134页的『CICS 关闭进程』。

第167页的『正在恢复 DCE』。

DB2: 管理指南。

DB2: 程序设计指南。

第11章 为 CICS 系统的恢复做准备

如果已将资源定义为可恢复的话，则在个别事务故障及在一些无法预料的关机发生后恢复数据是很简单的。 CICS 和 SFS 在重新启动系统时自动恢复在这些资源中的数据。

本节的信息说明了如何使 CICS 系统在数据丢失故障的情况下进行恢复。

表 30. 准备备份和恢复的导向图

若需...	参考...
启用 DCE 服务器的恢复。	第153页的『启用 DCE 服务器的恢复』。
启用 SFS 恢复。	第155页的『启用 SFS 的恢复』。
启用 DB2 的恢复。	第158页的『启用 DB2 的恢复』。
启用 CICS 区域的恢复。	第159页的『启用 CICS 的恢复』。
启用子系统项的恢复。	第160页的『启用 CICS 子系统项的恢复』。
启用 CICS 用户标识符的恢复。	第161页的『启用 CICS 用户标识符的恢复』。
启用应用程序数据的恢复。	第161页的『启用应用程序文件的恢复』。
备份系统。	第162页的『备份系统的建议』。
恢复 CICS 服务。	第163页的『改进性能以恢复服务』。

启用 DCE 服务器的恢复

DCE 服务器和它们的数据库在故障以后必定是可恢复的，因为 DCE 名称和安全服务器在 SFS 和 CICS 启动的网络中必定是可操作的。备份和恢复的步骤必须由管理员完成。

最小需求是:

- 在初始配置后的备份
- 附加阶段的备份

初始配置后的备份

在已为 CICS 配置 DCE 单元后，备份 DCE。这保证了基本名称服务目录结构，以及初始的委托人（如 cell_admin），可在故障后恢复。

这点对于恢复该信息，（而不是在冷启动后重新配置 DCE 单元）十分重要。如果 DCE 单元由冷启动配置，与单元更改关联的用户标识符和该项更改将防止存取到有 SFS 管理的数据。

阶段备份

阶段备份减少了恢复服务的尝试，例如在重新定义委托人，复位口令以及相似任务中。

DCE 备份只是运行 DCE 安全服务和 DCE 名称服务的机器所需要的。不必做客户机上 DCE 的备份。如果 DCE 从客户机上丢失，则需要重新安装 DCE 并将它们重新配置为客户。

测试 DCE 恢复过程

很难测试 DCE 服务器的恢复，因为恢复过程扰乱了网络的操作，并且一个失败的恢复试图停止服务。要测试恢复，则要将单机与网络相连并恢复到它上面。

备份 DCE

需要恢复 DCE 的信息自动更改，并且在信息更改时很难预测。然而，当备份 DCE 时，还是有必要知道：

- 包含 SFS 服务器 DCE 网络口令的 SFS 关键文件包含在以下对象中：

`/var/cics_servers/SSD/cics/sfs/baseName/key`

- 包含区域 DCE 网络口令的区域密钥表文件包含在以下对象中：

`/var/cics_regions/regionName/keytab`

要备份 DCE：

1. 认证为 DCE 委托人 `cell_admin`。
2. 将安全服务放入 *维护* 模式中，以防止在备份时更新到安全数据库，使用以下命令：

```
sec_admin
Default replica: /.../DCECellName/subsys/dce/sec/master
Default cell: /.../DCECellName
sec_admin> state -maintenance
sec_admin> exit
```

在维护模式中，安全服务是只读。委托人可以注册并执行认证，但不能更改口令并且不能定义新的委托人。

不必暂停 CDS 服务器，因为可以动态备份名称服务。不必用管理操作来停止它。

3. 复制（如复制到磁带上）下列文件：

```
/krb5/*
/etc/rc.dce
/etc/dce_cf.db
/etc/mkdce.data
/etc/dce/*
/var/dce/*
```

注：星号（*）表示整个子树。

`/var/dce` 和 `/etc/dce` 目录结构包含

要备份的文件比需要备份的更多，但如果复制整个目录子树，该过程则更直接。备份需要的存储器相对于 SFS 备份仍然较小。

当复制这些文件时，必须保留他们的所有权和许可权，例如通过使用操作系统命令 **cpio**。

在目录结构中有一些隐藏文件，例如：

`/var/dce/security/.mkey`

保证拷贝同样包括文件的这种类型。

4. 将安全服务返回到整个操作，通过将状态设置回服务方式：

```
sec_admin
Default replica: /.../DCECellName/subsys/dce/sec/master
Default cell: /.../DCECellName
sec_admin> state -service
sec_admin> exit
```

相关信息

第167页的『正在恢复 DCE』。

第168页的『SFS 队列和文件管理恢复过程』。

第170页的『完全故障后的完全恢复过程』。

同样参阅 DCE 上的 AIX 文档。

启用 SFS 的恢复

机器上的所有 SFS 配置和运行信息都位于:

`/var/cics_servers`

该目录包含下列子目录:

- SSD, 它包含 结构化文件服务器定义 (SSD) 自动启动和冷启动数据库 (所有本机上的 SFS 的) 以及一一对应于 SFS 的信息
- SCD, 它包含 模式文件定义 (SCD) 自动启动和冷启动数据库 (所有本机上的 SFS 的)
- GSD, 它包含 网关服务器定义 (GSD) 自动启动和冷启动数据库 (是任何可能存在于本机上的 PPC 网关服务器 的) 以及一一对应于网关的信息。

对于每个 SFS, SSD 目录包含:

- 一个锁定文件, 用来在更新时锁定文件。锁定文件的全路径名是:
`/var/cics_servers/SSD/cics/sfs/baseName/lock`
- 一个密钥表文件, 它包含 SFS DCE 委托人的加密口令。密钥表文件的全路径名是:
`/var/cics_servers/SSD/cics/sfs/baseName/key`
- 一个信息文件, 它包含注册到服务器上的信息。信息文件的全路径名是:
`/var/cics_servers/SSD/cics/sfs/baseName/msg`
- 两个重新启动文件, 它们包含热启动是服务器需要的信息。重新启动文件的全路径名是:
`/var/cics_servers/SSD/cics/sfs/baseName/restart`
`/var/cics_servers/SSD/cics/sfs/baseName/restart.bak`

总之, SFS 逻辑卷的恢复对于启用应用程序数据的恢复十分重要。SFS 逻辑卷和 CICS 日志的连续镜像是应用程序可以恢复到故障点的唯一途径。复制逻辑卷并保存 CICS 日志可以使数据恢复到拷贝所到的级别。

媒体恢复归档 (MRA) 是 Encina 提供的一个功能, 它支持 Encina 服务器的备份和恢复。它允许创建一个 SFS 的脱机备份而同时不打扰服务。

MRA 文件的备份可以用来恢复应用程序数据, 但仅可以到且包括上一个可用 MRA 文件上的数据。这可能是还没有备份的系统可用的 MRA 文件。

注: MRA 没有备份 CICS 日志, (它在 `/var/cics_regions/regionName` 目录中), 如果要恢复区域相应级别中的应用程序数据, 则 MRA 是没有用的。原因是 CICS 日志必须在与 SFS 逻辑卷相同的处理点上, 使数据变得有意义。

备份 SFS 逻辑卷

SFS 管理了与两个逻辑卷上的 CICS 文件控制、辅助临时存储器队列、内分区瞬时数据队列和本地队列的 ATI 请求相关联的数据。当冷启动 SFS 时, 这些逻辑卷上的数据就被丢弃。

选择 SFS 脱机备份方式

因为 SFS 是一个 Encina 部件, 所以有两个 SFS 数据脱机备份选项:

- 由 Encina 提供的媒体恢复归档 (MRA)
- 使用操作系统设施复制数据

MRA 的使用或备份 SFS 的操作系统各有优点和缺点。当决定做什么时, 应该考虑以下几点:

1. MRA 允许在 SFS 运行时备份。因为日志归档文件持续写入, 所以不必经常备份 SFS 数据。数据可以经常恢复到与最近归档的文件一致的级别中。
2. 复制逻辑卷有一些缺点, 即 SFS 必须在备份进行时停止, 并且数据只能恢复到备份时的级别。这意味着要经常备份。
3. MRA 归档文件的持续写入也可认为是一个缺点, 因为它引起 SFS 性能的下降并且文件需要磁盘空间。如果没有适当维护, MRA 写入的文件会引起 SFS 的终止。要避免 SFS 的问题, 需要经常复制归档文件以确保它们驻留的文件系统不会饱和。
4. 复制逻辑卷有一个优点, 即数据是恢复到一已知的级别 (此级别是拷贝产生时的级别)。结果, 如果 CICS 日志同时保存, 区域可以重新启动到已知的级别。
5. 有了 MRA, 如果记录卷丢失, 数据卷可恢复到的级别不能完全定义, 尽管它可能比数据卷备份更早。这说明数据不能恢复到与已保存的 CICS 日志一致的级别。
6. MRA 有一优点, 如果数据卷丢失, 它可从备份恢复到它丢失时的级别。仅就这方面来说, 它可视为镜像 SFS 数据卷的另一方法。

使用媒体恢复归档 (MRA) 备份 SFS

备份 SFS 过程定义如下:

1. 服务器在启用 MRA 的状态下运行。
2. 创建了一个 SFS 数据文件逻辑卷的全备份并保存到磁带或存储在其他系统中。

注: 虽然恢复并不可能, 但可以以增量形式 (模糊备份) 做到这点, 直到完成了全备份。

3. 重新启动文件 (**restart** 和 **restart.bak**) 是从服务器自用目录里保存的。

注: 实际上, 一旦保存了这些以后, 它们只需要在更改服务器的逻辑卷时重新保存即可。

4. 继续使用 SFS 。
5. 每次在给定的级别中填入 SFS 日志文件逻辑卷, SFS 将日志文件逻辑卷的状态转储到归档文件中。

6. 日志文件被分阶段保存。

该过程应该以有规律的间隔完成。

如果发生系统故障，并且要恢复重新启动文件，则恢复过程取决于是否丢失了日志文件的逻辑卷。

如果丢失了日志文件逻辑卷，不考虑是否丢失了数据卷，则必须首先从归档文件中恢复日志文件逻辑卷然后是数据卷。

如果使用 MRA，则要保证分阶段减少系统上日志归档文件的数目。这些文件可以很大，并且如果留下来填充文件系统驻留的卷，则 SFS 不能归档任何数据并且异常终止。应该以适当的间隔从目录中保存和删除这些文件。对于称为 `./cics/sfs/hostB` 的 SFS 来说，文件写入到此目录：

`/var/cics_servers/archives/hostB`

或者，可以使用 SFS 诊断工具（SDT）将 SFS 文件转换成平面文件（可将它存储在另一个系统上）。要得到使用 SDT 的信息，参阅 *CICS 管理参考大全* 书中的“cicssdt - SFS 诊断工具”。

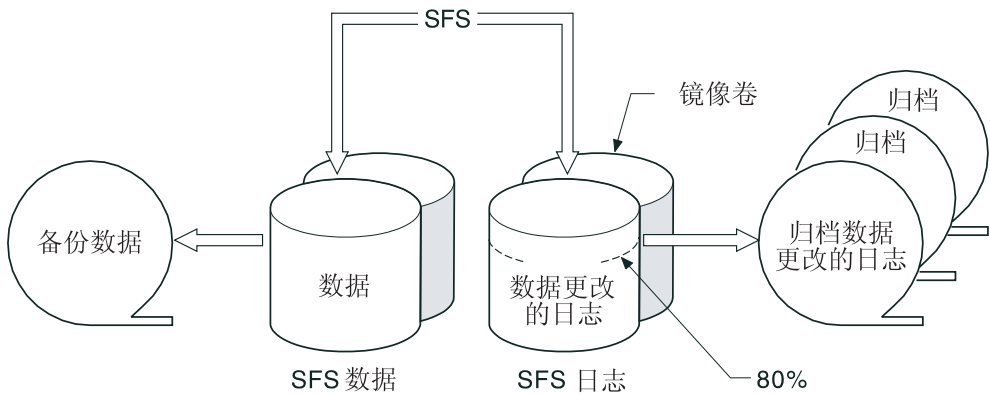


图 4. 结合 SFS 备份镜像的媒体恢复归档（MRA）

复制 SFS 逻辑卷

可以使用操作系统来复制逻辑卷的整个内容。对于 SFS 来说，这需要将重新启动文件、数据、和记录卷复制到一致的状态。完成它的唯一方法是当 SFS 不在运行时，将它们同步复制。重新启动文件（`restart` 和 `restart.bak`）在服务器的自用目录中。

如果发生系统故障，则可从可用的最近一级中恢复 SFS 数据和记录卷，并自动启动 SFS。

启用和禁用媒体归档

媒体归档是可启用与禁用的，如 *CICS 管理参考大全* 中的“结构化文件服务器定义 (SSD)”说明的那样。

1. 关闭 SFS。
2. 使用 `cicsupdate` 命令来更改 `MRAArchivingEnabled` 属性，改为 `yes` 或 `no`。参阅 第143页的『更改运行时中的资源定义』。

```
cicsupdateclass -w -c ssd MRAArchivingEnabled -n yes
cicsupdateclass -w -c ssd MRAArchivingEnabled -n no
```


3. 用 startype=auto 热启动 SFS

或者，媒体归档可不用停止 SFS 而启用或禁用，只需使用 **tkadmin** 命令来覆盖 **SSD MRAArchivingEnabled** 属性。**tkadmin** 命令只影响当前 SFS 运行的媒体归档设置，并不能永久更改 SSD 属性。

```
tkadmin enable mediaarchiving -server serverId
tkadmin disable mediaarchiving -server serverId
```

注：tkadmin 命令可用来查询是否启用或禁用媒体归档。

```
tkadmin query mediaarchiving -server serverId
```

为了使用媒体归档来建立恢复的基准线，应该使用 Encina 恢复和记录服务来备份该 SFS 的逻辑卷或该点的逻辑卷。因为备份文件由 SFS 创建，它必须有目录（要在其中创建备份段）的操作系统读写的许可权。作为示例的命令提纲是：

```
mkdir backup-directory
cd backup-directory
chown SFS2 backup-directory
chmod 777 backup-directory
tkadmin backup 1vol -server ./:/cics/sfs/hostABCD sfs_ShostABC
```

媒体归档备份的建议

要备份 SFS 数据，需要备份路径和 SFS 备份文件名的记录信息。该信息在产生数据活动特殊阈值时，记录到媒体归档文件中。如果丢失了日志数据，SFS 备份信息（使用媒体归档文件的日志数据恢复所必须的），将在媒体归档文件中捕捉到。

当日志数据被消除并且在 SFS 数据恢复后只有很少的媒体归档备份文件可用，则会有一间接的可能性：必需的 SFS 备份文件信息将不记录到媒体归档文件中，在这种情况下，SFS 数据不能恢复。

该情况可以避免：

- 如果 SFS 模糊备份在局部阶段（SFS 备份是主要为之设计的）完成。
- 如果只需要 SFS 数据（而不是日志数据）。

启用 DB2 的恢复

有两种类型的恢复：

- 恢复到以前的备份（版本恢复）
- 恢复到故障点

启用版本恢复

要能将数据恢复到以前的备份，需要一个与 CICS 日志备份同时所做的数据库备份。这样做的原因是数据和 CICS 日志需要恢复到相同的处理点。

使用 DB2 备份数据库实用程序得到 DB2 数据的脱机备份。使用操作系统设施得到 CICS 日志的脱机备份。这两个备份必须在区域关闭以及数据库脱机的状态下得到。否则，CICS 日志和数据库可能不同步，且恢复数据的能力也将是不可靠的。

启用失败恢复点

要能够将数据和应用程序恢复到故障点，需要有数据库和故障发生时的 CICS 日志的副本。要启用它：

- 需要为 CICS 日志做镜像。操作系统文档说明了如何使用镜像来创建活动文件系统的脱机映像备份。
- 需要镜像包含 DB2 数据的逻辑卷或启用前卷恢复。操作系统文档说明了如何镜像逻辑卷。要使用前卷恢复，需要：
 - 数据库的脱机备份拷贝
 - 启用前卷恢复

如 *DB2: 管理指南* 所述，恢复允许将一已恢复的备份建立到一指定的及时点上。数据库必须配置为允许前卷恢复。可通过设置在 DB2 数据库配置文件中的指定参数来做到这点。例如，**logprimary** 参数用来设置创建主日志文件的数量，并且 **logfilsiz** 参数用来设置每个日志文件的页数。

我们建议镜像 DB2 数据库，但在 DB2 日志和 CICS 日志间缺乏协调。

相关信息

第169页的『DB2 队列和文件管理恢复过程』。

第170页的『完全故障后的完全恢复过程』。

启用 CICS 的恢复

每个区域的 CICS 目录结构都位于：

`/var/cics_regions/regionName/`

这里的 *regionName* 是区域的名称。目录结构包含：

- 冷启动数据库（同样称为永久数据库），它包含所有在区域冷启动时使用的资源定义
- 自动启动数据库（同样称为运行时数据库），它包含所有在区域自动启动时使用的资源定义。

注：这些数据库不包含结构化文件服务器定义（SSD）、模式文件定义（SCD）和网关服务器定义（GSD）。它们都在 `/var/cics_servers/` 上，如 第155页的『启用 SFS 的恢复』中说明的那样。

- CICS 日志，它包含队列和 START 请求的内部信息
- 区域环境变量
- 密钥表文件，它包含为区域创建的 DCE 委托人的加密口令
- 用于热启动的区域重新启动信息
- 组成 CICS 应用程序的可执行文件（缺省的）
- CSMT 瞬时数据队列（TDQ），和任何到统计文件的输出
- 事务或系统转储文件
- CICS 应用程序使用的 BMS 映象定义（缺省的）

包含在区域目录中的信息可从备份中恢复。然而，如果要将应用程序恢复到故障点处，则需要故障点处的 CICS 日志（SFS 逻辑卷）的拷贝。

备份 CICS 日志

无论何时在 CICS 区域中启动事务，CICS 都将有关信息存储到 CICS 日志中。要确保日志数据可恢复的最好办法是使用为包含日志数据的文件系统镜像的数据。实际上，这说明控制文件系统的逻辑卷必须为镜像而配置。最好有一个日志数据的专用文件系统。它给予您对文件系统存储器的较大控制并确保存储器只能用于 CICS 日志文件的存储。

注：不要使用 CICS 日志的网络文件系统（NFS）。

使用 `cicsexport` 和 `cicsimport` 用于备份和恢复

可以使用 `cicsexport` 和 `cicsimport` 命令来备份和恢复在区域目录上的一些数据，但它只适用于在自动启动（运行时）还未用这些命令备份和恢复时的冷启动。

`cicsexport` 复制了下列数据：

- 冷启动（永久）数据库，它包含区域的资源定义。如果需要任何区域定义都在复制区域和恢复拷贝的区域间共享，则会发现维护单个定义的物理拷贝，并且使用符号链接提供第二区域中的更新拷贝是较为简单的。
- 转换模板。
- 所有程序
- 所有映象
- 区域数据目录的内容。

例如，要将 REGION1 的资源定义归档到名为 REG1ARCH 的输出文件中，使用下列命令：

```
cicsexport -r REGION1 -o REG1ARCH
```

然后要恢复 REGION1 的资源定义，资源定义下列命令：

```
cicsimport -r REGION1 -i REG1ARCH
```

在 *CICS 管理参考大全* 中的“`cicsexport` - 导出 CICS 资源定义”和“`cicsimport` - 导入 CICS 资源定义”应该在尝试使用这些命令前复查一下。

如果 CICS 资源定义不能从备份或归档中重新创建，则它们可使用在 第1页的『第1部分 配置与定制』中说明的步骤重新配置。

启用 CICS 子系统项的恢复

在 第1页的『第1部分 配置与定制』中说明的大多数配置任务创建了操作系统子系统项。

CICS 部件的子系统项作为管理任务的结果而更新，并只需要偶尔的备份。参阅操作系统文档，以得到关于子系统项定位地点和如何备份的信息。

如果 CICS 子系统项不能从备份中恢复，则它们可以通过重新配置 CICS 部件来重新创建，如 第1页的『第1部分 配置与定制』所示。

启用 CICS 用户标识符的恢复

CICS 部件的操作系统用户标识符信息作为管理任务的结果而更新，并只需要偶尔的备份。参阅操作系统文档，以得到关于用户标识符信息定位地点和如何备份的信息。

如果 CICS 用户标识符不能从备份中恢复，则它们可以通过重新配置 CICS 部件来重新创建，如 第1页的『第1部分 配置与定制』所示。

启用应用程序文件的恢复

如果有足够的数据，则可以将应用程序文件恢复到故障点处。否则，可以将应用程序文件恢复到上一次备份时的状态。

将应用程序文件恢复到故障点

要将系统数据恢复到故障点处的状态，以下是必需的：

- CICS 日志内容的当前拷贝
 - 下列之一：
 - 故障点处数据内容的副本。它可用逻辑卷的 *后续映像* 来维护。参阅操作系统文档，以得到如何使用镜像和使用它的蕴含式的信息。同样请参阅 第163页的『使用镜像卷』。
 - 最近的备份和 DB2 日志或 SFS 日志
- 要使用 DB2 日志恢复数据库，可以使用 DB2 前卷方式。前卷允许使用复原数据库和前卷数据库实用程序重新建立数据库，这里的数据库活动和归档日志用来将更新重新应用于数据库（自上一备份开始）。使用这种方式，可以选择一正在处理的特殊点来将数据库恢复到，或者可以将数据库恢复到数据库日志的末尾。参考 *DB2: 管理指南* 以得到关于备份数据库和启用前卷恢复的信息。

在 第24页的『在 SFS 上建立队列和文件管理』中的过程说明了如何建立 SFS 逻辑卷。这些逻辑卷中的一个用来将更新记录到数据中。参阅 第170页的『完全故障后的完全恢复过程』以得到关于从 SFS 日志中恢复 SFS 的信息。

将应用程序文件恢复到以前的备份级别

要将数据恢复到以前的级别，可以选择使用脱机备份（在区域和文件管理器关闭时所做的备份）。将数据库从备份中恢复返回了数据库到备份时的状态。

在 DB2 中，这称为 *版本恢复*。必须明确地使用备份数据库实用程序做数据库的副本。

如果要使用媒体恢复归档（MRA）来备份数据，须复查关于 *CICS 管理参考大全* 中“结构化文件服务器定义(SSD)”的 **MRAArchivingEnabled** 属性的信息。

注：在区域和文件管理器关闭时做备份是很重要的，因为这保证了 CICS 日志和数据的副本是在处理中的相同点上。

相关信息

参考 DB2: 管理指南 中的前卷恢复特性的有关信息。

备份系统的建议

如何计划恢复取决于个人的需求、CICS 系统的配置和可用的资源。

复查下列的说明以帮助决定如何设计系统用于需要的恢复级别。

非产品的 CICS 系统

如果使用一非产品的 CICS 系统，如用来开发和测试的，则运行应用程序的动态恢复可能不如产品系统重要。因此，不必进行镜像。

如果需要将非产品系统恢复到原始配置，则非产品系统的脱机备份是必需的。然而，要使脱机备份高效地将系统热启动到故障以前的状态，在区域和文件管理器关闭时做备份是十分重要的。否则，数据，如存储在文件管理器上的瞬时数据队列，将与存储在区域文件系统上的 CICS 日志数据不同步，这样将需要一个冷启动来启动区域。

CICS 冷启动放弃了在 CICS 日志中的数据、任何在运行时资源定义数据库中的数据以及在任何 CICS 队列中的数据。SFS 冷启动放弃了所有由 SFS 维护的文件和它们的内容。没有 DB2 冷启动的概念；重新启动 DB2 维护了所有在数据库中的数据。然而，它对由 DB2 管理的队列和文件没有影响。

在运行时资源定义数据库中的数据可能是重要的，特别是在资源定义落实到永久资源定义数据库前，资源定义测试的环境中。

冷启动 SFS 需要重新创建 SFS 逻辑卷上的文件。如果必须冷启动 SFS，参阅 第24页的『在 SFS 上建立队列和文件管理』以得到重新创建文件的信息。

产品 CICS 系统

用来为产品 CICS 系统恢复作准备的方式取决于系统活跃的程度、系统上的应用程序、以及如何迫切需要系统处在恢复的时刻。

一个区域，作为终端集线器，装运所有的事务到一个或一个以上的操作远程区域，或者运行事务，需要大量运算，但只有有限的数据（由远程区域管理的），可能只需要备份而无须镜像。然而，除非使用镜像，否则不能将系统恢复到故障时的状态 --它只能恢复到系统上一次备份时的状态。仅当上一次备份的进行中的事务状态可用时，它才是足够的。

— 相关信息 —

第167页的『正在恢复 DCE』.

第168页的『SFS 队列和文件管理恢复过程』.

第169页的『DB2 队列和文件管理恢复过程』.

改进性能以恢复服务

通过测试备份和恢复过程，应该有一个对恢复服务的估计，不论是在完全故障（磁盘或处理器完全消除），或在各种较小的故障后。

如果在较小的故障后，要恢复服务的时间长度不可接受，则可以重新计划备份策略以提供一个较好的恢复。

在事务处理网络（这里有多管理域）中，例如一个网络联接了两个或更多的区域，每个区域可能有它们自己的文件管理器。在这种情况下，事务将更新在公共同步点下文件管理器中的文件。在这种情况下，能恢复 CICS 日志是十分重要的。这样做的一个原因是：如果故障发生时，在同步点处理的准备和落实阶段捕捉到事务，则未受影响的文件管理器继续保持与受影响事务相关联的锁定关联，直到它们被手工释放或 CICS 日志（在受影响系统中的）恢复到故障是的状态。要手工释放在 DB2 中的锁定，则使用 LIST INDOUBT TRANSACTION 命令，如 DB2: 命令说明 中说明的那样。

使用镜像卷

要帮助您快速恢复 CICS 服务，可以依次将镜像卷用于：

- CICS 日志，文件管理器日志和数据。典型地，需要分配相当大的（相对于其他磁盘空间的使用）磁盘空间给文件管理器，所以镜像磁盘的消耗是很明显的。
- 配置信息

使用第二网络接口卡和额外的网络布线

这避免了服务（因为有一个有缺陷的网络接口卡，或其他在网络中防止网络通信的有缺陷的设备）的打扰。

使用第二处理器作为保留（备份）

这避免了在服务中因为机器硬件故障的干扰。取决于应用程序的设计，通常可以在处理器上运行一个区域和一个文件管理器。

也可以使用第二处理器和两套双线外部磁盘。有了这样一个系统，所有的配置信息和应用程序数据都在外部磁盘上并已镜像，而且所有的处理器都直接连接到两个磁盘上。

使用一个高可用性簇多重处理（HACMP）

高可用性簇多重处理（HACMP）是一个松联结、群集的 RISC System/6000 机器（执行了 AIX 上的 HACMP 软件）环境，它支持冗余和共享资源访问间的高可用性。除了基本的 RISC System/6000 机器，HACMP 不需要任何特殊的硬件和工作。

HACMP 提供了一套设备，保证了系统故障的快速恢复。它不保证连续的操作但它提供了系统故障后的最小恢复时间。HACMP 为关键任务数据库和事务处理应用程序提供了高可用性。

资源,如磁盘驱动器、适配器和网络链接是由 HACMP 软件监视的。如果检测到故障，HACMP 软件可以自动启动由系统管理员定义的恢复操作。

使用 CICS 的 HACMP 脚本示例

第273页的『附录E. 使用 HACMP 的示例脚本』包含四个用于 CICS HACMP 的脚本示例。示例脚本说明如下：

脚本：两台机器和一个塔形：该示例说明了一个环境，这里两个 RISC System/6000 模型 53H 由一个 9333-500 塔相连。机器共享了在塔上的卷组，尽管每个机器通常有这些组中的一个。如果有机器故障，则 HACMP 能够将 CICS、Encina 和 DCE 移动到第二机器上。示例脚本复制通过了必需的 DCE 和 CICS 文件，自动启动了 DCE、CICS Encina 服务器，最后是 CICS。参阅第273页的『HACMP 示例脚本 1（APPLID1）』。

脚本 2：使用一测试系统作为备份：另一示例是针对这些位置的：即它们无法负担备份机器，但可能有机用于测试系统（它可在紧急情况下作为产品系统）。在这种情况下，在接管时 HACMP 将 CICS 和 Encina 服务器移动到（已有一工作 DCE 客户和一可操作 CICS 堆栈的）第二服务器上。在接管时，逻辑卷和文件系统由 HACMP 簇管理器传送。有多个 CICS 区域运行的位置同样希望使用该示例。在传送后有一个性能上的影响。参阅 第278页的『HACMP 示例脚本 2（APPLID2）』。

脚本 3：在 Encina 和 CICS 在单独的机器上时：示例 3 表示了具有不同机器上 Encina 和 CICS 的用户。在这种情况下，如果机器有故障，则故障部件将移动到可用机器上，而不是释放整个系统。在发生接管时，逻辑卷和文件系统由簇管理器传送，并且启动了示例脚本。这样合并了必需的 CICS 文件，并且在剩余的系統上启动了区域。DCE 和 Encina 自始至终保留了可操作性。参阅 第281页的『HACMP 示例脚本 3（APPLID3）』。

脚本 4：传送逻辑卷和文件系统：在第四个示例中，逻辑卷和文件系统由簇管理器传送并且启动了示例脚本。这样合并了必需的 CICS 文件，终止了 CICS 区域并自动启动 CICS Encina 服务器，然后启动了 CICS。参阅 第284页的『HACMP 示例脚本 4（APPLID4）』。

使用 HACMP 时的考虑

当在 HACMP 环境中设置 CICS 时，特别要注意以下几点：

- 在簇中文件管理器的用户标识符必须是唯一的，并且对于整个组织都适用。

应该创建用户，如 快速入门 所述，但须完成 'User ID' 字段。这必须要在任何将运行文件管理器上完成。

mkuser 的命令行版本应该加上 **id=number** 参数，如果要使用 SMIT 的话。

- 为了 CICS 使用的文件管理器，创建了 SRC 子系统。它通常在文件管理器创建时由 CICS 创建。同样必须在任何要运行文件管理器的系统上手工创建信息。SRC 子系统信息可由下列命令添加：

```
cicssrccreate -s filemanagerUser -n ././cics/sfs/name -u SFSuser
```


第12章 备份和恢复的实现

本章说明了如何恢复 DCE、文件管理器和区域。

表 31. 备份和恢复实现的导向图

若需...	参考...
恢复 DCE。	第167页的『正在恢复 DCE』。
恢复 SFS。	第168页的『SFS 队列和文件管理恢复过程』。
恢复 DB2。	第169页的『DB2 队列和文件管理恢复过程』。
执行完整的恢复。	第170页的『完全故障后的完全恢复过程』。

正在恢复 DCE

要从备份中恢复 DCE （参阅 第154页的『备份 DCE』）：

1. 安装 DCE。
2. 从备份中恢复下列文件：
 /krb5/*
 /etc/rc.dce
 /etc/dce_cf.db
 /etc/mkdce.data
 /etc/dce/*
 /var/dce/*

 注：星号（*）表示整个子树。
3. 使用 **auto** 选项启动 DCE。不能重新配置 DCE，因为需要使用已从备份中恢复的配置信息：
 /etc/rc.dce
4. 认证为 cell_admin。
5. 将安全服务设置为服务方式：
 sec_admin
 Default replica: /.../DCECellName/subsys/dce/sec/master
 Default cell: /.../DCECellName
 sec_admin> state -service
 sec_admin> exit
6. 可以通过列出所有的 CDS 项来测试已恢复的 DCE，它们应该是象备份以前的那样。
 例如，可以使用下列命令：
 cdslr -Rodw /.:

可以显示定义到安全性服务器的 DCE 组，它们应该象备份以前的那样，只须使用 **rgy_edit** 或 **dcecp** 即可显示。下列示例使用了 **rgy_edit**：
rgy_edit -g -v

当恢复 DCE 时，在其上恢复的机器必须有和故障机相同的 IP 地址和网络名称。这保证了客户机信任安全服务。

相关信息

第153页的『启用 DCE 服务器的恢复』。

同样参阅 DCE 上的 AIX 文档。

SFS 队列和文件管理恢复过程

本信息说明了如何通过将 SFS 移动到新的机器从而恢复它。假设具有 SFS 数据文件逻辑卷的备份、SFS 日志文件的逻辑卷（参阅 第161页的『启用应用程序文件的恢复』）和 DCE 的备份（参阅 第153页的『启用 DCE 服务器的恢复』）。

使用下列步骤恢复 SFS：

1. 将 DCE、Encina 和 CICS 安装到新机器中。
2. 恢复 DCE 备份并启动 DCE（参阅 第167页的『正在恢复 DCE』）。确保机器有与备份机相同的 TCP/IP 名称和地址。
3. 创建 SFS 以匹配原始的 SFS。参阅 第24页的『在 SFS 上建立队列和文件管理』。SFS 的结构化文件服务器定义（SSD）必须精确地与正恢复的原始 SFS 的 SSD 相匹配。
4. 冷启动 SFS，参阅 第237页的『使用 `cicssfscscreate` 定义 SFS』。
5. 关闭 SFS。
6. 恢复 SFS 日志文件逻辑卷。使用的方法取决于可用的备份。

如果 SFS 在允许媒体恢复归档（MRA）状态下运行，则记录卷可以使用记录归档文件来恢复。要做到这点，必须在管理模式下冷启动 SFS（使用 `Encina sfs` 命令而不是 `cicssfscs`）并使用 `tkadmin recover logvol` 命令。使用该方式，则记录卷在上一个日志归档文件写入时恢复到它本来的状态。

如果在 SFS 停止时，使用操作系统命令来取得日志文件逻辑卷的备份拷贝，则该拷贝可用来将逻辑卷恢复到本来的状态。对于这种恢复类型，SFS 必须是空闲的。

7. 恢复 SFS 数据文件逻辑卷。

SFS 数据文件逻辑卷应该只在日志文件逻辑卷恢复后再恢复。用来完成的方式取决于可用的备份和日志文件逻辑卷如何恢复。

如果使用 `tkadmin backup lvol` 命令来取得 SFS 数据文件逻辑卷的备份，则可使用 `tkadmin recover lvol` 命令来恢复数据。它将数据恢复到与日志文件逻辑卷已恢复状态一致的状态。要做到这点，SFS 必须保留在管理模式中（使用 `Encina sfs` 命令启动的），如使用媒体恢复归档的 SFS 日志恢复需要的那样。如果 SFS 数据文件仅要求恢复（就是说，SFS 日志文件不需要恢复），则 SFS 必须以管理方式热启动（使用 `Encina sfs` 命令而不是 `cicssfscs`）。数据备份必须是后备级别（比较日志文件逻辑卷的恢复级别来说）。

在恢复后，使用 `Encina sfs` 命令、`tkadmin stop server` 命令来关闭 SFS。

注：如果日志文件没有丢失且无须恢复，则在数据卷丢失时，该方式可用来将数据恢复到它本来的状态。

如果使用操作系统命令来取得备份，则在制作拷贝时，备份的拷贝可用来恢复数据文件到它本来的状态。然而，对于一个成功的恢复来说，日志文件逻辑卷必须已精确恢复到数据文件逻辑卷的相同级别。实际上，这意味着这两个卷的拷贝必须在 SFS 空闲时同时取得。

8. 在恢复逻辑卷时，以正常方式自动启动 SFS，而不是以管理方式。

SFS 现在应该恢复。

可通过在 SFS 上显示文件列表来测试已恢复的数据。例如，如果 SFS id 是 hostB，使用下列命令：

```
sfsadmin list files -server ./:/cics/sfs/hostB
```

可以定义并启动一个 CICS 区域并且用它来发出事务以测试 SFS 数据的有效性。

参阅示例，它们在：

```
/usr/lpp/cics/src/examples/backup/backup_sfs.sh
```

相关信息

第128页的『在启动时和运行事务时文件管理器不可用』。

第24页的『在 SFS 上建立队列和文件管理』。

第167页的『正在恢复 DCE』。

第153页的『启用 DCE 服务器的恢复』。

第237页的『使用 cicssfscree 定义 SFS』。

DB2 队列和文件管理恢复过程

本信息描述了如何恢复 DB2 队列和文件管理数据库。假设已有了 DB2 数据的备份。

在恢复 DB2 数据库时，使用了两种方式：

1. 复原恢复，它在制作备份时将数据库重新建立为本来的状态。它同样在 DB2 中引用为 版本恢复。
2. 前卷恢复（在数据库已恢复的拷贝上建立的），使用日志（它包含上一次备份以来的修改）来更改该拷贝。

要恢复用于 CICS 队列和文件管理的 DB2 数据库：

1. 确定是否要将数据恢复到以前的备份（复原恢复）或到故障发生点（前卷恢复）。
2. 如果要重新创建数据库，则须验证数据库是否已如第28页的『在 DB2 上设置队列和文件管理』说明的那样为 CICS 队列和文件管理配置了。

注：DB2 恢复数据库实用程序可用来创建新的数据库并恢复数据映像的备份。

3. 启动数据库。
4. 到 *DB2: 管理指南* 中以得到关于使用恢复数据库实用程序来恢复数据库的指导。

相关信息

第128页的『在启动时和运行事务时文件管理器不可用』。

第28页的『在 DB2 上设置队列和文件管理』。

完全故障后的完全恢复过程

在故障之后，第一件要做的事是标识故障处，然后决定适合当时条件的操作。

在 CICS 系统完全失效（如替换硬件）后需要一个完整的恢复。该过程假设已经有 DCE、Encina、DB2（如果正使用队列和文件管理的 DB2）和 CICS 的安装媒体，并且有原先所做的备份磁带。

使用下列步骤：

步骤 1.安装 DCE、Encina 和 CICS

如果要替换硬件，或者如果丢失了包含 DCE、Encina 或 CICS 代码的磁盘，则用原来安装的方法重新安装安装一遍。

步骤 2.恢复 DCE 服务

如果丢失了 DCE 安全服务器或名称服务器，则必须重新配置他们，并按 DCE 文档恢复安全数据库和名称服务数据库。

必须恢复安全性数据库，而不是清除并重新配置它。DCE 安全性按用户标识符工作，所以如果重新配置，每个委托人有一个新的用户标识符，并且不能自动启动 SFS 以重新得到对事务处理系统中的数据的存取。

如果丢失了 DCE 客户配置，可能不得不将机器重新配置为 DCE 客户。

步骤 3.恢复队列和文件管理

转至：

1. 第170页的『步骤 3a. SFS 恢复』 如果正使用队列和文件管理的 SFS。
2. 第171页的『步骤 3b.DB2 恢复』 如果正使用队列和文件管理的 DB2。

步骤 3a. SFS 恢复

如果没有丢失 SFS 的操作系统逻辑卷，则可自动启动 SFS。在自动启动 SFS 后，查看信息文件：

```
/var/cics_servers/SSD/cics/sfs/baseName/msg
```

为正常自动启动创建的信息文件包含近似于如下的一个信息：

```
1 A sfs: Initialized ... Thu Aug 19 07:55:54 1994
```

使用信息文件的内容来诊断任何未解决的问题。可能的问题和解决办法是:

- SFS 的 DCE 网络口令丢失。

SFS 的 DCE 委托人是:

`cics/sfs/委托人`

并且 DCE 网络口令包含在下列文件中:

`/var/cics_servers/SSD/cics/sfs/baseName/key`

该文件必须是可读和可写的(通过运行 SFS 的操作系统用户标识符)。可以使用 **DCErgy_edit** 工具或 **dcecp** 来重新创建该口令文件或更改口令。

- SFS 的逻辑卷不再存在。

如果信息文件表明是这样,参阅第168页的『SFS 队列和文件管理恢复过程』,它说明了如何恢复 SFS 逻辑卷。

如果不能解决信息文件中的错误,可尝试 SFS 的冷启动。然而,仅将冷启动作为最后的办法放弃了在所有在 SFS 逻辑卷上的文件。

步骤 3b.DB2 恢复

如果需要重新配置数据库,参考第28页的『在 DB2 上设置队列和文件管理』。

参考 *DB2: 管理指南* 以得到关于恢复数据的信息。

步骤 4.恢复 CICS 服务

恢复 CICS 永久和运行时的资源定义文件,它们驻留在:

`/var/cics_regions/regionName/database`

冷启动不需要冷启动,恢复 CICS 的日志文件,它们驻留在:

`/var/cics_regions/regionName/log`

如果能自动启动 SFS,则尝试热启动区域。如果必须冷启动 SFS,则必须冷启动该区域。然而,如果冷启动该区域,则运行时资源定义文件和 CICS 日志文件将废弃。

CICS 区域将任何问题都报告到它的区域控制台文件中,它在:

`/var/cics_regions/regionName/console.nnnnnn`

复查信息文件,找出错误。`console.nnnnnn` 文件按顺序生成。`console.nnnnnn` 在 `/var/cics_regions/regionName/data` 中。每一次新的 CICS 区域启动, `console.nnnnnn` 文件的数目都增加一次。该文件的大小通过区域定义的 **MaxConsoleSize** 属性控制。设置 **MaxConsoleSize** 为零意味着 `console.nnnnnn` 将增大,直到达到系统的极限。将它设置为特定的大小表示第一个文件将关闭而新文件将打开,同时具有下一个可用的号码。发出两个信息,告诉何时关闭文件和新文件的名称。**cicstail** 自动地继续截除新文件。

注: 如果 CICS 区域正在运行,并且 SFS 因任何原因而停止,文件将标记为 **DISABLED**。如果用管理操作来恢复 SFS 并且区域仍在运行,则必须设置这些文件为可用,例如,通过使用 **CEMT SET ENABLED**。如果有许多文件复位,自动重新启动 CICS 可能比较简单。

第4部分 性能

表 32. 测量和改进性能的导向图

若需...	参考...
改进系统的性能	第175页的『第13章 如何改进系统的性能』。
测量系统的性能	第193页的『第14章 使用 CICS 工具测量性能』。

第13章 如何改进系统的性能

完成可接受级别的性能是一项困难的任务，因为 CICS 应用程序用户预计的响应速度与其它业务的需要冲突。

- 验证至 CICS 区域和资源的访问
- 完成 CICS 任务的大吞吐量
- 把优先级给最重要的任务、终端或用户
- 备份 CICS 任务影响的数据
- 保持企业的硬件预算

表 33. 改进系统性能的导向图

若需...	参考...
发现有关性能测量的工具： <ul style="list-style-type: none">• 由操作系统提供• 由 CICS 提供	第175页的『测量性能』。
改进 CICS 系统的性能。	第176页的『改进 CICS 系统的性能』。

测量性能

操作系统和 CICS 提供几个工具或设施。可以使用这些设施的组合来测量系统的性能。

由操作系统提供的性能测量工具

vmstat

报告有关虚拟内存、磁盘、陷阱和 CPU 活动的统计。用来发现系统装入活动。

iotat 报告 tty 设备、磁盘和 CD-ROM 的 CPU 和输入/输出统计。

监控程序

启动和停止概要文件的子程序。参考操作系统文档，以获得更多信息。

跟踪

监视器选择诸如进入和退出选定的子程序、内核例行程序和中断句柄的系统事件。事件按时间顺序，在其它事件的上下文中显示。可用来检查是谁，在何时，如何和为什么发生事件。请参阅操作系统文档，以获取更多的信息。

sar 记录并报告多种系统统计。

netstat

通过显示各种与网络有关的数据结构的内容来记录并报告网络活动。

请参阅操作系统文档，以获得更多信息。

CICS 提供的性能测量工具

监控 在 CICS 中一些预先决定的点上收集与性能有关的信息。也可把监控点嵌入用户定义的应用程序。例如，可使用监控设施来确定特殊的任务：

- 作了一些同步点请求

- 等待终端输入的总时间
- 等待 TCP/IP 链接的经过时间

可使用 **cicsmfmt** 命令来格式化和打印性能记录。请参阅第197页的『监控性能』，以获取更多的信息。

统计 显示 CICS 如何管理它控制下的资源。资源分为几个类别（如文件、LUW、终端），统计分为几个类型（如间隔、天结束、请求的）。

用统计设施可回答诸如：文件控制多久更新一次特殊文件？在特殊时间周期有多少事务在区域中运行？从远程系统接收到多少分布程序设计链接（DPL）功能装运的请求？

可使用 **cicssfmt** 命令来格式化间隔报告和摘要报告。请参阅第194页的『如何收集统计和格式化报告』，以获取更多的信息。

跟踪 通过 CICS 或应用程序来显示控制流。CICS 提供至三个类型跟踪的访问：主存、辅存和外存（操作系统）。可使用 **cicstfmt** 命令来有选择地格式化和打印辅助跟踪文件的内容。请参阅第202页的『跟踪 CICS 执行性能』，以获取更多的信息。

由关系数据库提供的性能测量工具

请参考 DB2 文档，以获得更多有关性能工具的信息。如果使用 DB2 来进行队列和文件的管理，应该读一读 *DB2: 管理指南* 中有关调整和配置系统的信息。

改进 CICS 系统的性能

该节描述一些 CICS 设施的性能消耗并描述了如何调整系统。

表 34. 改进 CICS 系统性能的导向图。

若需...	参考...
标识性能的 CICS 约束。	第177页的『标识 CICS 性能约束』。
发现有关弱响应时间的解决方案。	第179页的『弱响应时间解决方案的摘要』。
查找有关性能的消耗。	第179页的『可能影响性能的 CICS 设施』。
发现在启动和关机时改进性能的方法。	第180页的『改进启动和关机的性能』。
查找物理内存。	第181页的『查看物理内存来改进性能』。
查找交换空间。	第182页的『通过交换空间来改进性能』。
通过程序设计改进性能。	第182页的『对于性能的程序设计考虑』。
改进数据库性能。	第185页的『改进访问数据库的性能』。
开发 Encina 快速本地传送（FLT）来改进本地 SFS 的性能。	第186页的『用快速本地传送（FLT）来改进 SFS 的性能』。
查找有关平衡最大吞吐量和最小响应的内容。	第186页的『最大吞吐量或最小响应时间的决定』。
查找有关 I/O 调整。	第187页的『I/O 调整』。
查找有关文件控制性能。	第188页的『改进文件控制的性能』。
控制至 RPC 侦听程序进程的请求速率。	第189页的『控制至 RPC 侦听程序进程请求的速率』。
限制区域的任务数量。	第190页的『控制区域中运行事务的数量数』。

表 34. 改进 CICS 系统性能的导向图。 (续)

若需...	参考...
阅读有关平衡区域间事务工作负荷的内容。	第190页的『分配工作负荷区域』。
阅读有关用 SP2 改进性能的内容。	第190页的『用 SP2 改进性能』。

标识 CICS 性能约束

CICS 系统的主要约束表现为外部症状。主要症状为存储器压力条件弱的响应时间。本节详细描述这些症状，使您在系统遇到性能问题时可以识别它们并知道 CICS 是如何尝试解决各种情况的。

标识存储器压力条件

存储器压力是 CICS 用来指示 CICS 存储器中将有或现有的空闲内存空间的术语。

- 区域存储器池
- 共享任务存储器池
- 专用任务存储器池

CICS 通过在 **console.nnnnnn** 文件中显示警告信息指出池中有压力条件。CICS 在池分配的存储器量超过池的压力阈值（对于区域和任务共享池），或无存储器剩余（对于任何池）时提高存储器条件。

可使用区域定义（RD）中的各种属性来配置存储器阈值（定义为存储器池的百分率）。阈值可响应的缺省值为池大小的 90 %。

把压力信息看作是一个早期警告，CICS 可能开始显示低劣的性能特性。例如，事务可能因为区域池中沒有足够的空闲存储器而异常终止，事务因为等待任务共享池中的存储器而挂起来满足图或数据表的请求，或者事务因为任务共享池中无可用的存储器而从 GETMAIN 提为 NOSTG 条件。如果无足够的调页空间，开始不会启动。

可使用 CICS 统计和监控设施来分析不同池中的存储器用法。从这些数据，可检测使用和操作的模式。

CICS 区域中存储器压力条件的主要原因是：

- 区域池比区域中活动和定义的资源数要小。

CICS 使用控制块的区域池。区域中活动的资源数和为控制块分配的区域存储器量之间有关系。

- 任务共享池比区域中使用的图和数据表的大小要小。

CICS 跨越应用程序服务器共享加载的图和表，但大量不同的图和表可能增大较小任务共享池的压力。CICS 使用散列法机制来定位任务共享的存储器区域和加载图和表。可使用区域定义（RD）配置散列表的大小。两个散列表是 **TaskSHNumBlocks** 和 **LoadDataNumBlocks**。**LoadDataNumBlocks** 包含可真正加载的分散值，如程序和表。除了程序和表，**TaskSHNumBlocks** 包含终端输入输出区（TIOA），和其它任务使用的存储区。接着，**LoadDataNumBlocks** 是 **TaskSHNumBlocks** 的子集。因为该原因，最好保持这两个值相同或 **TashSHNumBuckets** 在大小上为一种次序。然后可使用存储器统计来调整两者。另见 CICS 管理参考大全 中“区域定义 (RD)”的 **LoadDataNumBlocks** 和 **TaskSHNumBlocks** 属性的描述。

如果区域使用了大量的图和表，则表的大小大于 512 比较适当。

- 应用程序代码包括许多使用了大量未释放的 GETMAIN 存储器的 LINK 或 XCTL 程序。应该审查这样的应用程序，并重新编写代码，在不需要存储器时使用 FREEMAIN。可能需要重新设计这样的应用程序并把它们分为更小的单元。请参阅第185页的『改进访问数据库的性能』，以获取更多消息。
- 操作系统在处理大量进程（CICS 应用程序服务器或其它程序）时可显示存储器压力的症状。

要了解关于调整各种存储器条件的信息，请参阅第181页的『查看物理内存来改进性能』和 第182页的『通过交换空间来改进性能』。

标识存储器违反

CICS 通过使用 特征 字符串来检测存储器违反。CICS 在存储器分配给调用程序后立即将特征字符串放入存储器。CICS 在释放关联的存储区时检查特征字符串。CICS 自我一致检查事务检查区域中的所有特征字符串和共享任务存储器池。

CICS 在存储器违反发生时进行不同的操作，这取决于影响了哪个存储器池：

专用任务池

CICS 在程序明确调用 FREEMAIN 和任务结束回收存储器期间检查存储器违反。

如果 CICS 检测存储器的违反情况，它把信息记录至 CSMT 或 console.nnnnnn 来指示发生存储器违反处的地址并恢复存储区。CICS 在 FREEMAIN 中检测到存储器违反时，将提高任务的 A47C ab 正常终止代码。如果已经为事务在 事务处理定义（TD）项中设置了 **TransDump** 标志，则 CICS 将初始化事务转储。

共享任务池

CICS 使用其自我一致事务来检查该池中的存储器违反。如果 CICS 检测到存储器的违反情况，它把信息记录至 console.nnnnnn 来还原特征字符串并恢复存储区。运行事务且区域继续处理。

区域池 CICS 使用其自我一致事务来检查该池中的存储器违反。如果 CICS 检测到存储器的违反情况，它把信息记录至 console.nnnnnn 并异常终止区域。如果在区域定义（RD）中已经设置了 **SysDump** 标志，则 CICS 将初始化系统转储。

可使用下列所述来帮助标识问题区域：

- CSMT.out 中的信息
- console.nnnnnn 中的信息
- 存储器统计
- 事务转储
- 系统转储

要详细了解存储器违反，请参阅 *CICS 问题确定指南* 中的“CICS 检测出一个存储器违例”。另见 第182页的『对于性能的程序设计考虑』以获取选择正确的存储器类型的提示。

弱响应时间解决方案的摘要

下列图表显示响应时间的级别，按严重性递减次序排列。对于每个级别都显示主要原因和建议的解决方案的范围。

表 35. CICS 响应时间检查列表

级别	症状	主要原因	各种解决方案
1	对于所有事务的所有加载的弱响应。	<ul style="list-style-type: none">调页的高级别。主要资源的高级用法。	<ul style="list-style-type: none">分配更多物理内存。重新估计系统资源需求和重新设计系统。应用程序错误和资源争用的检查。
2	中、高负载的弱响应。	<ul style="list-style-type: none">调页的高级别。高级处理器用法。高级 DB 或文件用法。高级通信网络用法。TP 或 I/O 存取法约束。超过 CICS 限制值。	<ul style="list-style-type: none">分配更多物理内存。减少路径长度，增加处理器功能或跨越处理器分配系统的部件（SFS，DCE 名称或安全性服务器）。标识文件，减少数据传送或增加容量。减少数据传送或增加容量。增加缓冲区可用性。更改操作数，提供更多的资源或在应用程序中出错时检查。
3	仅某一事务的弱响应。	<ul style="list-style-type: none">标识公共特性。行和（或）终端用法。文件用法。高级存储器用法。与影响的事务相同的子程序。与事务所使用的相同的存取法或 CICS 功能。限制条件。	<ul style="list-style-type: none">作为级别 2，增加容量，减少数据传送或更改事务逻辑或文件设计。更改文件位置缓冲区分配或更改排队逻辑或文件设计。重新设计和（或）调整应用程序。重新设计和（或）调整应用子程序。重新分配资源或更改应用程序。重新估计有问题的功能使用。
4	某一终端的弱响应。	<ul style="list-style-type: none">检查作为适当的网络加载。检查操作员技术。检查 WD 输入项。	<ul style="list-style-type: none">增加网络该部分的容量。修订终端过程。重新定义 WD 输入项。

可能影响性能的 CICS 设施

下列 CICS 设施可能影响性能。

- CICS 监控和跟踪。**可以在不需要时关闭跟踪和监控。或剪裁跟踪和监控以仅产生最小的统计。
- 使用 CEDF。**要避免此类型调试的性能成本，不要在编译 **cicstran** 时使用 **-e** 选项。
- 使用主要临时存储器而不是辅助临时存储器**如果应用逻辑允许，应该使用主要临时存储器而不是辅助临时存储器来获取最大性能。

- **使用多个区域。**
 - **可恢复资源。** 如果使用只读的文件、瞬时数据队列和临时存储器队列，则它们可定义为不可恢复的，并可在系统或磁盘损坏时在其他地方备份。
 - **可恢复文件。** 可能有不需该保证的文件；例如审查记录文件可以容忍写入其它记录，甚至装入事务按次序重算。把这样的文件定义为不可恢复的就允许同步点在一些情况下更轻松地运行。定义只读文件为不可恢复，比定义它们为可恢复能够提供更好的性能。
- 请注意如果使用 DB2 进行队列和文件的管理，所有不可恢复的文件都作为可恢复的处理。
- **CICS 安全性。** CICS 依据 DCE 的远程过程调用（RPC）来进行区域、服务器和内部 CICS 进程间的通信。要获取更高的安全性，可使用认证的 RPC 来防止某些人获得或篡改敏感的进程和信息。认证的 RPC 提供从仅在 RPC 会话开始时校验认证到数据加密的保护级别的范围。然而，保护级别更高则需要更多的加密和更大的性能成本。

表 36. RPC 保护级别和性能

保护级别	性能
none	无性能影响。
pkt	影响与调用 相同。
pkt_integ	可能比影响调用和 pkt 大两倍。
pkt_privacy	可能比影响 pkt_integ 大五倍。请参阅注意事项。
注: 尽管 CICS 支持 pkt_privacy 的使用，但该安全性级别的 DCE 软件可能在系统上不可用。参考 DCE 产品信息。	

- 看到改进性能时，可能想重新检查 CICS 每个部件的安全性需求。
- **同步点。** 要获取最好的同步点性能，把事务中的参与者数量缩至最小，特别是更新的参与者。
 - **区域散列表大小。** 通常，对于 区域定义（RD）中每个资源应该把散列表设置为输入项数量的两倍。
 - **CICS 应用程序服务器。** 确认已设置 RD **MaxServer**，**MinServer** 和 **ServerIdleLimit** 属性为适当值。
 - **CICS 内部检查。**

改进启动和关机的性能

下表描述了 CICS 的启动顺序（包括 DCE 和 SFS）并简单说明了启动和关机将如何影响性能。这是完整的顺序；如果有组件已经运行，则某些部分不适用。

表 37. 启动 CICS 的性能开销

启动任务	性能开销	可能改进性能的操作
DCE 客户进程与安全性服务器联系。	两端都执行加密和解密来认证彼此。	None。
SFS 在 DCE 注册。	如果 SFS 在异常终止的 30 分钟内重新启动，则 DCE 确认旧的 SFS 不再存在需要暂停 40 秒。	通过在自动启动 SFS 之前输入下列命令来跳过暂停： cdscp delete object /./cics/sfs/sfsname

表 37. 启动 CICS 的性能开销 (续)

启动任务	性能开销	可能改进性能的操作
文件系统运行恢复。	如果文件系统异常终止，它将必须重做从最近的检查点已来的所有工作。文件系统提交那些在崩溃之前提交的事务，重算未到达同步点的事务并为到达了同步点的准备阶段但还未到达提交阶段的事务而再次锁定。	可以缩小检查点间隔，例如为一分钟，则文件系统将重做少于一分钟的工作。然而，当文件系统到达其检查点时，正常操作期间的每一分钟都将有系统开销。
CICS RPC 侦听程序在 DCE 上注册。	如果 CICS 在异常终止的 30 分钟内重新启动，则 DCE 确认旧的 RPC 不再存在需要暂停 40 秒。	通过在自动启动区域之前输入下列命令来跳过暂停： <code>cdscp delete object ./cics/regionName/ts</code>
CICS 区域运行恢复。	如果区域异常终止或用立即选项关闭，它从文件系统为 TS 和 TD 队列恢复指针，并与适当的文件系统和其它 CICS 区域联系来解决崩溃时处于同步点准备和提交阶段间的事务。如果无法与这些其它应用程序联系也可开启 CICS；在此情况下，由影响的事务加锁的记录和队列保持加锁直到事务被解决。CICS 区域然后尝试恢复崩溃时连接的每个终端。这些是按顺序来恢复的；如果终端可用，就恢复得快一些；但如果终端不可用它将在 8 秒后超时。如果有许多终端在重新启动之前不可用，崩溃的区域在该阶段将使用很长时间。	如果可能，总是正常关闭区域。

如果注意到在您的系统中有很多冗余 CICS 区域，它们可能对系统性能产生不利的影响。可以通过重新启动系统来删除这些冗余区域。如果决定做的话，按下列次序来关闭 CICS 系统的组件非常重要：

1. CICS 区域
2. SFS
3. DCE

如果 CICS 在 5 分钟后不终止，可以使用操作系统的 **kill** 命令来强迫终止。

查看物理内存来改进性能

如果运行 CICS 和 SFS 的系统有足够的内存，则事务使用物理磁盘的唯一时间是在同步点时。在同步点时，强制信息至磁盘使事务可在系统失败时重做。所有其它数据，如事务程序、文件中的记录和工作存储器在首次需要时被读入物理内存，并保存在那里。

如果有足够的物理内存，磁盘操作时间不会限制您的系统。然而，实际系统一般没有这么多物理内存而且管理的文件也通常比可用的物理内存大得多。当可用的物理内存进一步减少时，系统进入 系统颠簸。这在事实上系统中所有的操作系统进程由于等待缺页错误的解决而不能进行时发生。一旦缺页错误解决了，等待它的进程只能有一点点进展，但马上又进入另一个缺页错误而再次延迟。

当系统发生系统颠簸，短项补救将减少应用程序服务器的数量。用减少了的加载系统可更快地执行任务。然而，会降低事务的吞吐量。

长项上问题的解决方案是需要安装更多实存储器或重新设计事务程序以使用更少的存储器。也可通过增加用来调页的磁盘速度或数量。

可知道系统何时调页，何时通过观察终端响应时间进行系统颠簸；通常在调页期间放慢速度。如果增加应用程序服务器的数量且系统吞吐量降低了，系统将发生颠簸。

使用内存调整帮助的提示

使用一些 SFS 资源属性和操作系统命令，可调整内存存储器或内存存储器的使用：

SFS 缓冲区

SFS 直接管理大量存储器。SFS 存储器满时，一页一页地把数据传递到操作系统。如果 SFS 缓冲太小，它将耗费所有的时间来收取和传递数据至操作系统。如果 SFS 缓冲太大，操作系统在 SFS 下执行过度调页。可以通过改变结构化文件服务器定义（SSD）中的 **BufferPoolSize** 属性来调整 SFS 存储量。

SFS 处理线程

结构化文件服务器定义（SSD）中的 **OpThreadPoolSize** 属性设置至 SFS 的可在任何时间进展的文件控制请求（代表所有区域中的所有应用程序服务器）。过量的请求在 CICS 和 SFS 间的接口中排队，并在以前的请求完成后进行处理。较好的初始数是 10；如果系统由于等待不必要的 SFS 线程而空闲，可以增加该数，而在系统发生颠簸时也可以减少该数。

schedtune

操作系统进程分配器有检测系统颠簸的开始并采取措施停止整个系统，使它不再继续做无用功而停机的功能。它进行的操作是挂起那些出现大量缺页错误的操作系统进程，通常是几十秒。挂起 CICS 或 SFS 进程可能引起事务的重新算或事务程序的异常终止，而且它将引起几十秒的终端用户的延迟。如果选择了用户的 **cicsterm**，它将引起进程挂起时击键响应的丢失，或由于 EXEC CICS SEND、EXEC CICS CONVERSE 调用中失败而产生的事务程序的异常终止。

可以使用操作系统 **schedtune** 命令来覆盖该检测的灵敏度和操作的严重性，但不管该设施是否出错，用户还是将在系统颠簸时看见不可接受的长的和可变的响应时间。正常情况下，执行步骤来防止系统颠簸比尝试使用 **schedtune** 改变系统颠簸的影响要好。

通过交换空间来改进性能

经常推荐交换空间为内存的三倍；这通常被认为是系统必须开始之前可利用的最大量。系统颠簸。定义比此更多的交换空间也没害处；唯一的缺点是这种分配的磁盘空间不能被 SFS 或操作系统文件所使用。如果需要大量的交换空间，则把它分配在多个磁盘上更有效。

为仅运行 CICS 的系统定义比此更小的虚拟存储也许更适当。如果使用了大量的虚拟存储，则可能影响用户的响应时间。

对于性能的程序设计考虑

CICS 应用程序的编写者应该考虑下列问题：

选择适当的存储器级别

- EXEC CICS GETMAIN

GETMAIN 存储器用与操作系统分配和释放内存相似的机制来分配，并来自进程数据段。CICS 强制限制提供给事务程序的该类型存储器的数量，这在 区域定义 (RD) 中可配置。如果事务程序异常终止或不释放存储器而终止，该级别的存储器由 CICS 收回。

- EXEC CICS GETMAIN SHARED

GETMAIN SHARED 从所有 CICS 任务共享的存储器分配。可用的总量在区域数据库中配置；在 AIX 中，这可转换为至多 256 Mb。

GETMAIN SHARED 保持分配直到由事务程序明确地释放。

- 堆栈

利用使用 C 语言的函数 `alloca`，来为自动变量和动态大小的自动变量分配堆栈存储器。该存储器分配和释放最快，并在函数退出或事务程序异常终止时自动清除。

总计可用的堆栈为 64 Kbytes；然而其中一些在启动事务程序之前由 CICS 使用了，另一些将在服务 CICS API 调用时使用。

操作系统强制限制提供给每个 CICS 任务的该类型存储器的数量。这是为运行 CICS 区域的用户（普通 cics）配置的量。

```
smitty chuser
    ► cics
        ► Max DATA Segment
```

指示当前值。在 AIX 中，可至多增加到 256 MB。

该 256 MB 的一小部分由 CICS 内部使用。

由应用程序员来保证在适当的时间释放该内存。

因为不容易消除事务程序在保持 `malloc` 存储器时异常终止的可能性，所以对于事务程序一般使用 EXEC CICS GETMAIN 来代替。

BMS 映射后缀

特定设备的映射比独立设备的映射快得多。两种类型映射的创建在 *CICS 应用程序设计指南* 中的“使用屏幕设计辅助 (SDA) 进行 BMS 映射集编码 (仅适用于开放系统中的 CICS)”中描述。

驻留选项和共享库

C, IBM COBOL 或 PL/I 事务程序的大小可使用 **size -f** 命令来确定。字段文本（程序代码），数据（初始化的数据）和 bss（为 0 的数据）都是适当的。

操作系统程序大量使用共享程序库来减少代码大小。当前不支持程序使用共享程序库，包含如 EXEC CICS 调用的 CICS 依赖性。然而，仍然可从为公共函数使用共享程序库中得到好处，这些公共函数虽然是从 CICS 程序调用的，但它们自己不包含任何 CICS 依赖性。

\Micro Focus COBOL 程序的大小可由 **ls -l** 命令来确定。\\Micro Focus COBOL 程序由 CICS \\Micro Focus COBOL run-time 来管理。\\Micro Focus COBOL 程序有可能使用 EXEC CICS LINK 来调用其它 \\Micro Focus COBOL 程序，并通过重建 CICS \\Micro Focus COBOL run-time 或使用 EXEC CICS LINK 来访问 C 函数。

为该目的使用 EXEC CICS LINK 可能引起耗费进程时间时存储器使用的减少。

避免加锁

事务程序只要是不依靠或更新相同的数据，就可以并行地运行。如果它们需要相同的数据，第二个事务可能不得不等待第一个事务到达同步点。

可以设计应用程序来把加锁的争用量缩至最小。例如，假定必须把系统中执行的事务数量的计数值保存在文件中。每个事务需要包含如下语句：

```
EXEC CICS READ FILE ("count_file") UPDATE ... ;  
count = count + 1 ;  
EXEC CICS REWRITE FILE ("count_file") ;
```

因此每个事务将需要锁定该记录。

如果在同步点之前的最后一步执行该操作，可以得到更好的一致性；该方法中保持锁定为最短的可能时间。如果保持几个需要总计时就可进行摘要的计数器，您可能得到更好的一致性，计数器可以把事务类型作为关键字或把一个随机数作为关键字。

读程序

用 C, IBM COBOL 或 PL/I 编写的程序映射到文件系统的存储器中，所以需要很小的调页空间。采用每个应用程序服务器在首次运行事务程序时复制自己的方法来从文件系统中明确读取用 \Micro Focus COBOL 编写的程序。对于带有大量应用程序服务器和大量 \Micro Focus COBOL 事务程序的系统，这可以显著增加虚拟存储器的需求。

使用程序高速缓存

如果应用程序代码可以高速缓存，它将拥有更好的性能。这是因为程序使用时节约了重新加载的消耗。下列选项影响到高速缓存：

- 区域定义 (RD) **ProgramCacheSize** 属性。

C, IBM COBOL 或 PL/I 程序仅在加载程序，程序定义 (PD) **Resident** 属性设置为 **yes** 和使用中的高速缓存程序的数量未达到最大值时进行高速缓存。

定义 C, IBM COBOL 或 PL/I 程序可以改进性能，这是因为 C, IBM COBOL 或 PL/I 程序在下列事件之一发生前不能再次加载：

- 发出 SET PROGRAM NEWCOPY 命令。
- **Resident** 属性更改为 **no**。
- 高速缓存满了且为了给新入口提供空间而删除了程序。在这种情况下，删除最近最少使用的程序，而把较常用的程序留在高速缓存里。

因为正在使用的程序未从高速缓存里删除，使用 EXEC CICS LINK 时推荐高速缓存的大小应该允许各逻辑层次的程序都可以高速缓存。

- COBSW=*lmm*

这里的 *mm* 是以字节为单位的高速缓存的大小。这是一个 \Micro Focus COBOL 环境变量，它设置 \Micro Focus COBOL 程序高速缓存的长度。这需要定义在区域的环境文件中。

设置程序高速缓存可能影响应用程序使用 EXEC CICS SET PROGRAM NEWCOPY 选项的能力。同样按访问的次序把程序加载到高速缓存中。

注:

1. 程序高速缓存在预应用程序服务器基础中, 所以程序的新拷贝对还未运行程序的应用程序服务器无影响。
2. \Micro Focus COBOL 程序的 NEWCOPY 删除应用程序服务器预先加载的每个程序, 所以在 NEWCOPY 后使用每个该程序的新拷贝, 而不是 NEWCOPY 发出的程序。

改进访问数据库的性能

下列列表提供一些提示来帮助您设计应用程序以更有效地访问数据文件:

- 如果使用的是非 XA 数据库, 您的应用程序需要发出 EXEC CICS CONNECT 命令为每个事务启动数据库服务器。如此频繁的连接使用了大量的系统资源。应用程序服务器运行时, 应用程序可以多次重用 XA 管理的数据库连接, 而不需要重复调用 EXEC CICS CONNECT。
- 把 CICS 事务程序和如 SFS 或数据库产品的文件系统间的交互数量缩至最小。

CICS 事务程序和文件系统间的调用明显增加了完成程序的总时间。为了减少时间, 可能想使用动态 SQL 调用或调用 SQL 存储的允许在一个激活中处理几行或几个文件的过程。要获取如何使用这些功能的指导, 请参阅与您选择的数据库产品一起提供的 SQL 文档。

- 如果大多数事务不需要访问数据库, 则使用支持 XA 动态注册的数据库。

如果数据库支持 XA 动态注册, 则仅于数据库被提供的事务真正更新时, CICS 才可以安排在数据库中驱动同步点。当您在多个 XA 数据库和 CICS 文件控制文件环境中运行, 而其中的事务通常仅更新一些应用程序管理的数据时, 这能减少化在同步点的时间。

- 避免过多使用说明光标或准备动态 SQL 的 SQL 操作。

典型的是运算光标的 SQL 操作 (EXEC SQL DECLARE *CursorName*) 或准备动态 SQL 命令 (EXEC SQL PREPARE), 与 SQL 操作 (OPEN, FETCH, EXECUTE, CLOSE) 相比开销相当大。可能想在每个事务程序中仅使用一次这些开销甚大的调用。如果您的数据库产品支持的话, 显示在下面的程序设计风格可能更适当。

```
EXEC SQL OPEN ... ;
if (SQLCODE == {cursor-not-declared} )
{
    EXEC SQL DECLARE ... ;
    EXEC SQL OPEN ... ;
}
EXEC SQL FETCH ... ;
EXEC SQL CLOSE ... ;
```

- DB2 提供两个类型表空间的选择:

- 系统管理空间 (SMS) 表空间, 在这里使用操作系统文件管理器来控制存储空间。
- 数据库管理空间 (DMS) 表空间, 在这里数据库管理器将控制存储空间。

DB2 缺省使用 SMS 表空间。然而由于 CICS 提供性能改进, 所以推荐通常在 CICS 中使用 DMS 表空间。

请参阅 *DB2: 管理指南*, 获取如何创建 DMS 表空间的详细内容。另见第185页的『改进访问数据库的性能』以获取有关 CICS 和数据库性能的更多信息。

用快速本地传送（FLT）来改进 SFS 的性能

Encina 提供快速本地传送（FLT），这是增加如读取和写入的 SFS 文件操作速度的功能。CICS 和 SFS 间的初始集结使用远程过程调用（RPC）来交换它们各自的套接字名称。包括文件操作，随后的调用都带有套接字传送机制，这是比 RPC 更为快速的机制。

考虑和先决条件

要开发 FLT，必须遵循下列条件：

- 区域和区域的 SFS 必须在相同的机器上。
- 对于顺序的入口或关键字顺序文件，记录长度必须是 1024 个字节或更少，而对于相对记录文件，记录长度必须是 1020 个字节或更少。
- 下列属性必须设置为 **connect** 或更小。这些是用来确定对于特殊级别的队列或文件是否使用 FLT。

在 区域定义（RD）中：

- **RuntimeProtection**
- **LogProtection:**
- **LogicalTDQProtection**
- **PhysicalTDQProtection**
- **LogicalTDQProtection**
- **NonRecTDQProtection**
- **RecTSQProtection**
- **NonRecTSQProtection**
- **LocalQProtectProtection**
- **LocalQProtection**

在 文件定义（FD）中：**FileProtection**。

要获取这些属性，请参阅 *CICS 管理参考大全* 中的“区域定义 (RD)”和“文件定义 (FD)”。

安全性考虑

SFS 设置的操作系统域套接字允许任何注册到机器上的用户都可以读取或写入。这意味着注册至该机器的用户都可以建立会话来从 SFS 读取数据或向 SFS 写入数据，而不用引用 DCE 口令。如果允许用户注册至机器但又不想让他们访问 SFS，您应该使 FLT 禁用并使用 RPC 来保证服务器的安全。请参阅 第27页的『设置 FLT 环境变量』。

最大吞吐量或最小响应时间的决定

可以配置 CICS 用最小响应时间或最大吞吐量来运行。

下列参数将影响该决定：

MAXSERVERS

系统同时进行任务的最大数量。

如果该值太小，可能会浪费进程时间，因为系统中没有足够的磁盘读/写操作的工作来与其它事务重叠。

如果该值太大，系统可能耗费比运行事务处理更多的时间来管理服务器。

这被设置在 CICS 区域定义 (RD) 中。

SFS 线程

这确定了在 SFS 中同时进展的同时文件控制操作的最大数量。其它请求在至 SFS 的入口排队。通常 10 比较适当。这被设置在 SFS 的结构化文件服务器定义 (SSD) 中。

I/O 调整

如果忙磁盘约束了系统的吞吐量，可使用操作系统设施，如 **vmstat** 和 **sar** 命令来确定。如果看见在系统中有重要的为其它用户工作的 I/O 等待，则用此方法来约束系统。

可能的操作是：

- 安装更多物理内存
- 调整磁盘上数据的位置
- 运行更多应用程序服务器
- 安装快速磁盘
- 配置映像卷
- 配置磁条卷

下列列表可用来审查：

- **vmstat** 显示大量为进页面而锁住的处理（在 b 列）。

这提示高调页或交换活动。减少应用程序服务器的数量或安装更多的物理内存。在进行之前，不要做更多的调整操作。

- CICS 不启动足够的并行的事务。

每个会话式事务在事务进行期间阻塞一个应用程序服务器。剩余的应用程序服务器代表大量可启动的并行事务。如果该数量很小，系统不启动足够工作来重叠磁盘的等待。特别的，对于 CICS 区域网络中的拟会话式工作可用的应用程序服务器的总数不能小于在该网络上管理文件的 SFS 服务器的数量。

- **filemon** 显示逻辑和物理卷的用法速率。

来自操作系统的统计 **filemon** 为每个物理卷指示平均数据传送速率是多少，查找模式是什么；为每个逻辑卷指示用法模式是什么。

用作日志的卷在每个同步点写入。可能希望把每个都放到自己的物理卷上。否则，在数据库和日志区域间就可能有大量的查找。

添加物理内存降低了磁盘用法目的的需要，而不是同步点的需要。

镜像卷通常调度与镜像复制一样多的并行独立读取（2 或 3）；由于子系统必须保证所有拷贝的更新，这就有增加写入时间的风险。

分配逻辑卷时可以选择把它分在最大数量或最小数量的物理卷上。为磁条卷指定最大值，就把数据分在几个物理卷上，这改进了性能但增加了丢失一些数据的风险。

如果允许操作系统把逻辑卷跨越多个磁盘，它将自动划上条纹。

改进文件控制的性能

可以用下列方法调整文件控制的性能:

- **物理文件位置**应该把事务程序和文件间的跳跃数量缩至最小。特别是如果在网络中任何地方的文件都由 SFS 管理, 应该把 CICS 配置为直接访问它而不是由通过其他 CICS 区域的功能装运。
- **修订文件可恢复性的需要**可能会有不需要可恢复性保证的文件; 例如审查记录甚至在包括的事务最后回卷时都能够承受写入的附加记录。把这样的文件定义为不可恢复的就允许同步点更轻松地运行。同样, 只读文件可定义为不可恢复的来辅助性能, 并可在系统或磁盘损坏时在其他地方备份。
- **SFS 缓冲区大小**选择对于您的文件系统和硬件适合的值。缺省值为 1 兆字节 (MB) 且在结构化文件服务器定义 (SSD) 中。它可增加至 200 兆字节 (MB)。
- **错误隔离 文件定义 (FD) ErrorIsolation** 属性的缺省值为 **on**, 在此情况下, 所有 SFS 错误都传递至应用程序。因为消去了传递和处理错误的时间, 所以设置属性为 **off** 性能改进不大。
- **使用浏览高速缓存** CICS_BROWSE_CACHE 环境变量用来指定在 SFS 上浏览文件时使用的高速缓存大小。高速缓存建立在多个 4K 字节的记录中。确定高速缓存应该为多大且是否要使用高速缓存时应仔细考虑。记住所有文件浏览操作都将使用高速缓存。在某些情况下, 它会改进性能, 但在多个短浏览时可能降低性能。

DB2 文件管理性能的考虑

不需要由 XA 接口两阶段提交协议提供的附加数据完整性的地方, 使用 DB2 单阶段提交优化可获得性能上的优势。请参阅 第88页的『单阶段落实优化』。

通常, DB2 单阶段提交优化应该在 SFS 文件管理上提供性能的改进。

DB2 提供给用户一整套调整参数。这些参数标志为高 (中、低) 性能影响来指示每个的相对重要性。参数的详细描述在 *DB2: 管理指南* 中。

可配置的参数以 2 个类别定义:

- 数据库管理器类别影响所有数据库。
- 数据库参数影响个别数据库。

DB2 也提供收集数据库管理器操作上的统计信息和数据库处理上的活动信息的数据库系统监视器。该信息可用来帮助您决定调整哪个配置参数。DB2 监控在 *DATABASE 2 System Monitor Guide and Reference* 中描述。

对于 CICS 有特殊重要性的参数描述如下。参考 *DB2: 管理指南* 以获取细节。

tp_mon-name

该参数标识处理使用的监视器产品的事务名称且在使用 DB2 两阶段提交时有效。该参数应该设置为 CICS。设置后, 该参数启用 XA_END 链接。

mincommit

直到执行了提交的最小量或 1 秒超时到期, 该参数才允许把运行日志延迟写入磁盘。

对于用户应用程序适当时且在一些应用程序服务器连接至数据库处, 提交组可能减少 DB2 运行记录的写开销并提供性能的改进。连接至数据库的应用程序的数量应该大于或等于该值。

num_ioservers

该参数定义数据库代理在执行预取 I/O 和异步 I/O 的行为中使用的 I/O 服务器的数量。

maxagents

该参数定义在任何给定时间接受应用程序请求的可用的数据库管理器代理的最大数量。

maxcagents

该参数定义当前执行数据库管理器事务的数据库管理器代理的最大数量。

通常，在应用程序大量使用写入 TD 和 TS 队列处，如果使用 DMS（数据库管理空间）表空间而不是作为缺省值的 SMS（系统管理空间）表空间，则有望提高性能。

数据库管理器，而不是操作系统的文件管理器，使用 DMS 表空间来控制存储空间，就如同使用 SMS 表空间一样。

参考 *DB2: 管理指南* 以获取如何创建表空间的细节。

控制至 RPC 侦听程序进程请求的速率

外部对安装终端和执行事务的请求可从几个来源到达区域。CICS RPC 侦听程序进程（cicsrl）请求 **cicsterm**、**cicsternp**、**cicsteld**、开放系统上的 CICS 客户 EPI 和 ECI 应用程序以及 PPC 网关服务器。验证这些请求然后在应用程序服务器进程中执行。

RPC 侦听程序进程处理这些请求的速率由进程在 CICS 初始化期间指定的 RPC 侦听程序线程的数量来控制。可以通过设置 区域定义（RD）属性 **RPCListenerThreads** 为期望的值或把 **RPCListenerThreads** 属性设置为缺省值 0 来明确指定该值，让 CICS 计算分配的 RPC 侦听程序线程的数量。

缺省设置 RD 属性 **RPCListenerThreads** 为 0 且 RPC 侦听程序进程把 RPC 侦听程序线程的数量计算为 RD **MaxServer** 属性值的十倍。

每个正在验证请求的 RPC 侦听程序线程都有相关联的可以至多挂起 8 个进一步请求的 RPC 队列。如果有空闲的 RPC 侦听程序线程，则立即验证提交至 RPC 侦听程序进程的请求，否则请求等待 RPC 队列直到 RPC 侦听程序线程变为可用。所有 RPC 侦听程序线程为忙且所有关联的 RPC 队列为满时，任何进一步的请求都被 RPC 侦听程序进程拒绝。

CICS 通过管理提交、执行和完成请求的方法来改进 EPI、ECI 和 **cicsterm** 客户应用程序的性能。验证请求成功时，挂起 RPC 侦听程序线程直到事务通过应用程序服务器完成处理。

会话式，长运行拟会话式事务和自动安装请求可能在无 RPC 侦听程序线程可用时引起延迟。大约 90% 的 RPC 侦听程序线程为忙或等待请求完成时，CICS 通过处理管理进一步的请求。每个进一步的自动安装请求将仍然占用这些 RPC 侦听程序线程中的一个直到自动安装完成。

如果您有 CICS 的前一发行版且是从终端来提交请求，或用 EPI 和 ECI 来从那个 CICS 发行版提交请求到 CICS 当前发行版的区域上，则一旦验证请求（除了自动安装请求），RPC 侦听程序进程就释放 RPC 侦听程序线程。与之相似的是，从当前 CICS 发行版上使用终端或 EPI 和 ECI 来提交事务请求到旧级别的 CICS 上，也是同样管理。

控制区域中运行事务的数量数

通过使用 **事务类** 可控制在区域中同时运行事务的数量。可通过指定事务处理定义 (TD) 中的 **TClass** 属性把事务赋值给类。当使用 **区域定义 (RD)** 中的 **ClassMaxTasks** 属性时, 在给定类中同步运行的数量是有限制的。

CICS 允许 11 事务类: 类 1 到 10 以及 **none**。只有类 1 到 10 可使用 **ClassMaxTasks** 控制。定义为 **none** 类的事务不受任何类的限制。所有 CICS 提供的事务缺省为类 **none**。

要使用事务类保证 **CEMT** 在任何时间都运行, 则要:

- 把所有用户事务都指定为除了 **none** 外的类 (包括 CICS 提供的用户事务)。
- 把 **CEMT** 定义为类 **none**。
- 把 **MaxServer** 设置为至少比 **ClassMaxTasks** 中指定值的和大 1 。

任何时间中在区域里运行的事务数量由区域定义 (RD) 中的 **ClassMaxTasks** 和 **MaxServer** 控制。

分配工作负荷区域

通过把事务路由至其它 CICS 区域或使用动态事务用户出口, 可以在多区域环境中分配工作负荷来按照 CICS 事务用户的预计改进系统的响应时间、容量和可用性。

至其它 CICS 区域的路由选择事务

当定义至 CICS 的事务时, 可以确定把该区域作为“本地”或“远程”事务。事务缺省定义为本地, 这意味着它们总是在启动的 CICS 区域上运行。然而可以把事务定义为远程, 在这种情况下它路由至另一个 CICS 区域来执行。要定义事务为远程, 请包括事务处理定义 (TD) 中下列属性说明之一或两者都包括:

RemoteSysId

远程 CICS 系统或事务所在区域的名称。

RemoteName

如果与本地区域中定义的名称不同, 就根据该名称在远程系统上识别事务。

动态事务路由选择用户出口

把事务路由至多个区域也许会更好。这对于许多情况都有帮助, 如把事务路由至未能启动的区域时。把事务处理定义 (TD) 中的**动态**属性设置为 **yes**, 可使本地事务的执行或路由选择由动态事务路由选择用户出口来处理。请参阅第108页的『使用动态事务处理路由选择用户出口』, 以获取更多消息。

用 SP2 改进性能

可以用 IBM RS/6000 可伸缩 POWERparallel 系统 (SP2) 显著改进 CICS 系统的性能。SP2 系统包含几个处理器节点并可为节点间通信设置成使用高性能交换 (HPS)。SP2 管理指南描述如何配置和启动 HPS。然而, DCE 需要同样设置环境变量, 而且在一些配置中对于 SP2 外的客户, 寻址信息必须可用。

相关信息

要获取有关 RPC 保护级别的详细信息，请参阅 第210页的『DCE RPC 和 ACL 如何维护系统的安全』。

CICS 应用程序设计指南 中的“文件服务”和“关系数据库服务”。

CICS 管理参考大全 中的“文件定义 (FD)”和“区域定义 (RD)”。

CICS 应用程序设计参考大全 中的“SYNCPOINT”。

第14章 使用 CICS 工具测量性能

CICS 提供可获得有关 CICS 系统运行信息的工具。这些工具是 statistics, monitoring 和 trace。这些工具提供的信息可使用下列脱机实用程序来格式化: **cicssfmt**, **cicsmfmt** 和 **cicstfmt**。然后可以分析格式化数据来提供有关任何资源争用或其它可能影响 CICS 系统的问题的信息。

注: **cicsmfmt** 是样本程序, 它仅格式化监控数据的子集。 **cicssfmt** 是样本程序, 它提供基本的统计列表。

可以剪裁这些工具和脱机程序在 CICS 系统运行的某一事件点或某一时间间隔提供信息。也可剪裁它们来指定 CICS 提供信息的数据的类型。

表 38. 使用 CICS 工具测量性能的导向图

若需...	参考...
收集统计。	第193页的『收集统计』。
监视器性能。	第197页的『监控性能』。
跟踪 CICS 执行性能。	第202页的『跟踪 CICS 执行性能』。

收集统计

本节描述如何使用 CICS 统计设施收集来自 CICS 系统的统计数据并格式化基于那些统计数据的报告。

CICS 统计的信息收集

CICS 统计设施允许跟踪资源的管理, 如文件、队列、日志和事务; 文件读取的次数、对特定日志的未授权的访问尝试或在时间上特定点的事务启动的总次数。

CICS 统计信息的类别

CICS 统计信息有几种类别。分别为:

DUMP - 转储统计

转储写的记录号, 转储目录写错误的号码。

FILE - 文件统计

应用程序向文件请求的记录输入和输出活动。

INTERSYSTEM - 系统间通信统计

记录系统间通信统计。

JOURNAL - 日志统计

提供定义至区域的每个日志的统计信息。

PROGRAM - 程序统计

提供程序定义 (PD) 中定义的所有程序的统计信息。

RUNTIMEDB - 运行时间数据库统计

有关运行时间数据库的用法和支持散列表冲突特性的记录统计。

STORAGE - 存储器统计

提供有关区域池、共享任务池和专用任务池的信息。

TASK - 任务统计

提供与任务有关的信息。

TDQUEUE - 瞬时数据统计

提供有关瞬时数据定义（TDD）中定义的每个瞬时数据队列的统计信息。同样也为所有 TD 请求从区域启动开始累加统计。

TERMINAL - 终端统计

提供有关终端定义（WD）中定义的终端的信息。在显示用户如何加载系统时十分重要。

TRANSACTION - 事务统计

提供有关事务处理定义（TD）中定义的所有事务的信息。

TSQUEUE - 临时存储器统计

提供有关临时存储器定义（TSD）中定义的临时存储器队列的信息。

UNITOFWORK - 逻辑工作单元（LUW）统计

提供与 LUW 有关的累加统计。

统计的类型

CICS 产生四种类型的统计，每种类型都描述如下。要获取如何产生每种类型的统计，请参阅第194页的『如何收集统计和格式化报告』。

间隔统计

间隔统计指示指定时间间隔的系统活动。在每个间隔结束时，CICS 把收集到的统计写入磁盘并把统计值复位为 0。间隔统计对于特定时间周期或最高活动周期的整个分析非常有用。通过指定特定周期中的小间隔，可以更详细地观察特殊的活动。小时间片可以发现隐藏在大量统计数据后的潜在问题。

天结束的统计

天结束的统计报告上一次间隔结束和

- 天结束的到期时间
- 当 CICS 终止（正常关机）
- 当 CICS 终止（立即关机）

可以在有注意趋势和疑问点处使用天结束统计。因为该报告在每个正常的 CICS 终止后生成，应该按规律检查它。

请求统计

请求统计立即记录资源的状态，而不等待当前间隔到期后。请求统计对于分析临时或短的术语问题非常有用。

非请求统计

CICS 为动态分配和释放资源自动集中非请求统计。这些统计描述有关将被删除和丢失其统计的资源。

如何收集统计和格式化报告

两个区域定义（RD）属性和几个 CICS API 命令用来记录统计。它们是：

RD StatFile 属性

统计数据发送至该属性指定的路径名和文件。除了来自 EXEC CICS COLLECT STATISTICS 的，所有统计都写至该文件，用来为应用程序提供统计信息。

RD StatsRecord 属性

来收集间隔或非请求统计，该属性必须设置为 **yes**。CICS 在产生天结束统计和请求统计时忽略该属性。

EXEC CICS SET STATISTICS

这条命令在应用程序中使用来设置 CICS 收集天结束统计和间隔统计的时间。这对于请求或非请求统计不可用。

EXEC CICS PERFORM STATISTICS RECORD

该命令用来生成详细的请求统计并把统计值复位为 0。该命令发送输出至 RD **StatFile** 属性指定的文件。这对于请求天结束统计、间隔统计或非请求统计不可用。

EXEC CICS COLLECT STATISTICS

该命令用来提供应用程序至统计的访问，并返回给调用的应用程序单个资源的当前统计。一旦 CICS 把数据写入 RD **StatFile** 属性指定的文件，就不能用 EXEC CICS COLLECT STATISTICS 命令来访问。

API 命令可使用 CECI 事务或从应用程序中发出。请参阅 *CICS 应用程序设计参考大全* 中的“INQUIRE 和 SET 命令”，以获取更多的信息。

统计输出未格式化。可使用提供的脱机实用程序 **cicssfmt** 来格式化并打印该输出，或编写自己的程序来报告和分析统计。

如何生成天结束统计

1. 确认 RD 中的 **StatFile** 属性已设置。
这些统计的产生与 RD 中设置的 **StatsRecord** 属性无关。
2. 在应用程序中使用 EXEC CICS SET STATISTICS 命令来设置 CICS 收集天结束统计的时间。

如何生成请求统计

要生成请求统计：

1. 确认 RD 中的 **StatFile** 属性已设置。
2. 使用 EXEC CICS PERFORM STATISTICS RECORD 命令来请求当前挂起的所有或特定的资源统计。例如仅请求终端统计，可使用 EXEC CICS PERFORM STATISTICS RECORD TERMINAL 发送输出至 RD 中 **StatFile** 属性指定的文件。
也可使用 EXEC CICS PERFORM STATISTICS RECORD ALL RESET NOW 复位统计值为 0。

EXEC CICS COLLECT STATISTICS 把单个命名资源的当前统计或命名资源类型的全局摘要统计返回给调用的应用程序。例如，您可以得到 CICS 系统中所有程序或单个程序的统计。

如何生成间隔统计

间隔统计设施允许自动获取指定时间间隔的统计。该设施累加时间间隔的信息，且 CICS 于每次间隔结束时清除这些信息。可以指定范围为一分钟至 24 小时的间隔。

CICS 在启动时自动启动记录间隔统计，缺省值为三个小时，一天结束的时间为 00:00。

要生成间隔统计：

1. 确认 RD 中的 **StatFile** 属性已设置。
CICS 把统计输出写入 RD 中 **StatFile** 属性指定的文件。
2. 确认 RD 中的 **StatsRecord** 属性已设置为 **yes**。
3. 使用 EXEC CICS SET STATISTICS 命令来控制何时记录和记录什么。例如，如果 CICS 失败，它尝试为运行写天结束统计，但可能不成功。如果 CICS 不能写天结束统计，您可能想周期性地请求所有统计，例如每小时或半小时地使用间隔统计，并使用该信息来分析失败的原因。

或者，可以在一天的间隔时手工运行 EXEC CICS PERFORM STATISTICS RECORD 命令来请求间隔的完全统计。正常关机文件提供一统计的最后设置；其它情况中，得到最近可用的设置。

如何生成非请求统计

要生成非请求统计：

1. 确认 RD 中的 **StatFile** 属性已设置。
2. 确认 RD 中的 **StatsRecord** 属性已设置为 **yes**。
CICS 把生成的统计数据发送至指定的路径名。统计输出未格式化。

禁止把间隔和非请求统计集合起来

可以通过设置 RD 中的 **StatsRecord** 属性为 **no** 或设置 SET STATISTICS 命令中的 RECORDING 参数为 OFF 来禁止间隔和非请求统计的产品。天结束统计和请求统计仍然生成，与此属性的设置无关。

CICS 继续记录数据直到使用 EXEC CICS SET STATISTICS 命令来取消操作。

生成统计报告

可以使用 **cicssfnt** 来生成两个不同类型的统计报告。一个是间隔报告，这是详细的，另一个是摘要报告。请参阅 *CICS 管理参考大全* 中的“**cicssfnt** - 统计格式化器”。

两个报告使用文件包含的所有统计集合。所以，根据生成报告和最近清除文件的时间，您可以产生覆盖一小时、一星期或任何您想要的时间周期的报告。推荐您把摘要周期限制在一年或一年以内，因为摘要数据可能变得很大。

间隔报告： 该报告包含每个单独资源的详细统计和每个资源类别的摘要统计。例如，它包含瞬时数据统计中 CSMT 的信息，事务统计中 CECI 的信息以及如瞬时数据和系统间通信的资源类别的信息。间隔报告显示统计文件中包含的所有统计集合。这些统计集合是间隔、天结束、请求和非请求的集合。

摘要报告： 摘要报告只包含每个报告类别的摘要统计，例如来自统计文件包含的所有集合的瞬时数据和系统间通信。这些统计集合是间隔、天结束、请求和非请求的集合。

摘要报告设施也产生平均资源类别摘要统计并提供总计、最高和平均值的报告。这些报告只由天结束和间隔集合生成。

cicssfnt 产生的标题

下列标题由 **cicssfnt** 产生:

REQUESTED_RESET 统计集合

所有资源类型的统计记录, 用下列命令来请求:

```
EXEC CICS PERFORM STATS REC ALL RESETNOW
```

使用 RESETNOW 时, 统计计数器复位所有资源。

REQUESTED 统计集合

所有资源类型的统计记录, 用如下列命令来请求:

```
EXEC CICS PERFORM STATS REC ALL
```

INTERVAL 统计集合

间隔时间到期时取得统计。这节包含详细和摘要的统计。

ENDOFDAY 统计集合

到天结束时间时取得统计。这节包含摘要统计。

ENDOFREGION 统计集合

区域关闭时取得统计。这节包含摘要统计。

UNSOLICITED 统计集合

为动态分配和释放资源取得统计。 这节包含摘要统计。

监控性能

可以收集两个类型的信息:

- 标准 CICS 信息
- 来自 CICS 应用程序的信息

标准 CICS 信息

CICS 集中的信息主要与性能相关, 但与异常条件有关的信息也可用。

事务级数据记录(事务性能)写至由 MD 中 TDQ 属性指定的瞬时数据队列中。这些记录包含与个别任务资源用法有关的信息。任务终止时为每个任务记录。

可通过调用 CEMT SET MONITOR ON 事务或设置 MD 中的 **MonitorStatus** 属性为 **yes**来启用系统性能级监控。或者, 可使用 SMIT。请参阅 第143页的『更改运行时的资源定义』。

来自 CICS 应用程序的信息

除了收集标准 CICS 信息, CICS 监控设施允许应用程序把用户字段中的数据传递到监控设施。该数据可用来记录和进行事务级性能评估。事务级性能记录中允许可变数量的用户字段(计数, 时间周期和字符串)。

如果除了 CICS 监控设施提供的信息, 还想收集其它信息, 可以在应用程序中选择点来获取收集监控数据的特殊跟踪命令。这些跟踪命令称为事件监控点。可启动和停止用户定义的时钟并更新这些点的用户计数值。

要支持用户监控信息的收集，必须提供用户定义的应用程序来执行每个事件监控点的必需操作。该应用程序获得必需的信息并返回用户监控信息使 CICS 在事务结束时把它附加到写入瞬时数据队列中的性能监控程序数据之后。

CICS 提供附加用户定义的计数器和时钟字段，使您能计算应用程序或整个 CICS 系统中某一事件发生的次数并使您能在应用程序或整个 CICS 系统中定时两个事件间的间隔。

请参阅CICS 应用程序设计参考大全 中的“ENTER”，以获取更多的信息。

在监控定义（MD）中必须定义的

MD 中，定义：

1. **MonitorStatus** 属性，决定监控是否活动。
2. **TDQ** 属性，指定 CICS 用来收集需要的监控数据的瞬时数据队列。TDQ 包含空队列名，则监控发生时无信息写入。
3. **UseMonitorModule** 属性，在应用程序员指定用户定义的事件监控点时，指定收集监控数据的用户定义程序。
4. 哪个级别的监控数据要收集，哪个不需要。包括和排除属性为此目的而提供。

注：未定义可保持使用事件监控点字段的最大值，所以可以保持无限量的数据。然而，如果用户（事件监控点）区域的整个大小大于瞬时数据队列记录大小的最大值，则禁用该任务的监控。性能区域的大小取决于包括或排除哪个监控。

监控字段列表和与之关联的组显示在第198页的表 39 中。

远程过程调用到达应用程序管理器进程时连接事务。远程过程调用到达调度程序和远程过程调用到达应用程序管理器之间的时间就是为连接事务所耗费的时间。

表 39. 监控字段表

字段标识符	组	说明
001	CICSTASK	系统通过该名称来识别事务。
002	CICSTERM	系统通过该名称来识别终端。
003	CICSCICS	系统通过该名称来识别操作员。
004	CICSTASK	事务类型可能是下列之一： <ul style="list-style-type: none">• A（通过自动事务启动来连接（ATI））• C（会话式任务的第二个或随后部分）• D（通过瞬时数据触发器级别来连接）• T（从终端连接）• Z（拟会话式任务的第二个或随后部分）
005	CICSCICS	记录启动时间。这是任务连接或数据记录在写会话式或 ULM 记录后复位的时间。
006	CICSCICS	记录停止时间。这是任务连接或数据记录在支持写会话式或 ULM 记录中完成的时间。
007	CICSTASK	任务在系统中执行的经过时间。在监控点间测量，并包含应用程序服务器在任务中开始和停止工作间的时间。不包括从 CICS 3270 终端仿真器到调度程序，在调度程序中排队或从调度程序到应用程序进程的时间。该字段也包含为其它字段中所计算的时间，如字段 027，063 和 101。

表 39. 监控字段表 (续)

字段标识符	组	说明
008	CICSTASK	任务处于正在执行的用户空间中的 CPU 时间。用户空间意味着执行的应用程序代码在操作系统之外。它包括执行用户代码和 EXEC CICS 语句的时间。例如，它包括执行 SQL 语句的时间。
009	CICSTERM	任务等待终端 I/O 的总计时间。
010	CICSJOUR	等待日志 I/O 的经过时间。
011	CICSTevent 监控点	任务等待 TS I/O 的经过时间。
015	CICSTERM	任务等待终端管理器的总计时间。这是应用程序进程中终端 I/O 处理所用的经过时间。它包括在终端上等待用户响应的时间。这与字段 009 不同，因为区域不能区分处理消耗时间和等待 cicsterm 进程消耗的时间。
016	CICSTERM	任务等待终端管理器的系统时间。系统时间是用来处理 CICS 区域中的终端请求的时间。它不包括 cicsterm 进程中处理的时间。
024	CICSTASK	任务在事务调度程序中的经过时间。该字段总是为 0。
027	CICSTASK	任务自愿挂起的经过时间。
029	CICSSTOR	在不需要 I/O 活动的任务的生存周期发生的缺页错误数量。
030	CICSSTOR	在不需要 I/O 活动的任务的生存周期发生的缺页错误数量。
031	CICSTASK	任务的 TCA 顺序号码。
033	CICSSTOR	任务使用的数据内存的最大量。该字段作为 EXEC CICS GETMAIN 请求的结果来更新。
034	CICSTERM	由任务从主终端设施接收的信息数量。
035	CICSTERM	由任务发送至主终端设施的信息数量。
036	CICSFILE	由任务发出的文件 GET 的数量。
037	CICSFILE	由任务发出的文件 PUT 的数量。
038	CICSFILE	由任务发出的文件 BROWSE 的数量。
039	CICSFILE	由任务发出的文件 ADD 的数量。
040	CICSFILE	由任务发出的文件 DELETES 的数量。
041	CICSDEST	由任务发出的瞬时数据 GET 的数量。
042	CICSDEST	由任务发出的瞬时数据 PUT 的数量。
043	CICSDEST	由任务发出的瞬时数据 PURGE 的数量。
044	CICSTevent 监控点	由任务发出的临时存储器 GET 的数量。
046	CICSTevent 监控点	由任务发出的辅助 TS PUT 的数量。
047	CICSTevent 监控点	由任务发出的主 TS PUT 的数量。
050	CICSMAPP	由任务发出的 BMS 映射请求的数量。
051	CICSMAPP	由任务发出的 BMS IN 请求的数量。
052	CICSMAPP	由任务发出的 BMS OUT 请求的数量。
054	CICSSTOR	由任务发出的 GETMAIN 请求的总数。
055	CICSPROG	由任务发出的 LINK 请求的数量。
056	CICSPROG	由任务发出的 XCTL 请求的数量。
057	CICSPROG	由任务发出的 LOAD 请求的数量。
058	CICSJOUR	由任务发出的日志输出请求的总数。

表 39. 监控字段表 (续)

字段标识符	组	说明
059	CICSTASK	由任务发出的 START 或 INITIATE 请求的数量。
060	CICSSYNC	由任务发出的 SYNCPOINT 请求的数量。
061	CICSSTOR	任务从主存交换出来的次数。
063	CICSFILE	等待文件 I/O 的经过时间。
064	CICSTASK	保持在任务的执行期间检测信号异常条件信息的字段。这些位标志定义为：位 1 检测阶段时钟启动/停止是否正常。位 5 检测是否破坏数据存储器。位 22 检测最大任务条件。位 23 检测是否有存储器短缺情况。
067	CICSTERM	由任务从可选终端设施接收的信息数量。
068	CICSTERM	由任务发送至可选终端设施的信息数量。
071	CICSprog	连接时调用的第一个程序的名称。
083	CICSTERM	由任务从主终端设施接收的字符数量。
084	CICSTERM	由任务发送至主终端设施的字符数量。
085	CICSTERM	由任务从可选终端设施接收的字符数量。
086	CICSTERM	由任务发送至可选终端设施的字符数量。
087	CICSSTOR	使用的文本内存的最大量。该字段作为 LINK 或 XCTL APL 命令的结果来更新。
088	CICSSTOR	从 EXEC CICS FREEMAIN 请求获得的内存的总量。
089	CICSCICS	系统通过该名称来识别用户。
090	CICSMAPP	由任务发出的 BMS 请求的总数。
091	CICSDEST	由任务发出的 TD 请求的总数。
092	CICSTevent 监控点	由任务发出的 TS 请求的总数。
093	CICSFILE	由任务发出的文件请求的总数。
094	CICSSTOR	任务在程序压缩上耗费的总时间。
095	CICSSTOR	用户任务的数据段占有。该字段在 GETMAIN、FREEMAIN 请求和任务退出之前更新。
096	CICSTASK	任务耗费在 CICS 空间的经过时间。CICS 空间是指 EXEC CICS 语句的处理。与把应用程序代码上耗费的经过时间排除在外的字段 007 不同。
097	CICScoms	通过它在连接时间识别原系统或本地终端的全限定名。这是为连接至 CICS 区域的终端由自动安装用户出口生成的名称。对于使用系统网络体系结构 (SNA) 链接连接的终端，这是远程系统的 LUNAME。所以，对于 CICS 是应用程序拥有的区域的事务路由终端，这是终端拥有区域的 LUNAME。
098	CICSTASK	当前任务 LUW 的标识符的唯一表示法。
101	CICSDEST	任务等待 TD I/O 的经过时间。
103	CICSCICS	任务等待 TS 空间或内存的经过时间。
108	CICSSTOR	用户任务的文本段占有。该字段在 LINK、XCTL 或 LOAK 请求和任务退出之前更新。
109	CICSTASK	事务的优先级。事务的优先级确定哪个事务在可用时首先使用资源，以及事务多快执行。

表 39. 监控字段表 (续)

字段标识符	组	说明
112	CICSCICS	引起写监控记录的条件。可能是下列之一： <ul style="list-style-type: none"> • C（会话式） • D（ULM 写请求） • T（任务结束）
113	CICSPROG	任务记录的第一个异常终止代码。
114	CICSPROG	任务记录的最近异常终止代码与任务记录的第一个异常终止代码不同。
115	CICSPROG	等待加载程序的经过时间。
200	CICSFILE	文件管理器耗费的经过时间（SFS）。
202	CICSSTOR	由任务发出的 FREEMAIN 请求的总数。
203	CICSSTOR	从 GETMAIN 请求获得的内存的总量。
207	CICSOMS	任务等待 SNA 链接的经过时间。
208	CICSOMS	任务等待 TCP/IP 链接的经过时间。
209	CICSOMS	任务接收的 ISC 信息的数量。
210	CICSOMS	任务发送的 ISC 信息的数量。
211	CICSTASK	任务处于正在执行的内核空间中的 CPU 时间。
212	CICSTASK	由于该进程在其时间片完成之前自愿放弃系统资源，上下文切换发生的次数。
213	CICSTASK	由于高优先级处理线程可运行或因为当前进程超过了其时间片，上下文切换发生的次数。
214	CICSTASK	任务接收到的信号的数量。
215	CICSTASK	任务活动时文件系统 I/O 的操作数。该数只为实 I/O 记录；高速缓存机制提供的数据仅负责首先读取或写入数据的进程。
216	CICSTASK	任务耗费在 CICS 空间的系统时间或用户时间。
217	CICSUSER	耗费在监控 ULM 的经过时间。

如何控制 CICS 监控

要使用监控：

- 保证监控是为系统定义并且资源定义作任何改变后要冷启动该区域。
- 保证监控已在如下处调整：
 - 在系统初始化时。要指定 CICS 初始化时监控是开还是关，必须用 **cicsupdate** 命令来设置永久 MD 中的 **MonitorStatus** 属性为 **yes** 或 **no**。或者，可使用 SMIT。请参阅 第143页的『更改运行时中的资源定义』。
 - 在运行期间。在 CICS 初始化后设置 CICS 监控设施为开或关，要使用 CEMT SET MONITOR 事务处理。

要指定希望报告哪个监控数据，必须使用 CICS 资源管理命令来改变 MD 中的 **Include** 和 **Exclude** 属性。这些改变在下一次冷启动区域后实现。

CICS 监控设施输出的处理

CICS 提供名为监控数据格式化器 (**cicsmfmt**) 的脱机实用程序，是应用程序的形式。该实用程序读取、格式化和打印监控瞬时数据队列。它作为一个样本程序，当需要编写自己的程序来分析数据集时可以把它用作一个骨架。

另外，您可能想编写自己的应用程序来处理 CICS 监控设施的输出。

相关信息

CICS 管理参考大全 中的“cicsmfmt - 监控数据格式化器”，“区域定义 (RD)”和“产品定义 (XAD)”。

跟踪 CICS 执行性能

外部跟踪

是由操作系统 AIX 提供的跟踪设施。该跟踪设施称为事件性能跟踪设施 (EPTF)，可以马上从所有 AIX 产品收集跟踪。它提供纳秒时间戳记以至可以估计哪个产品或进程正在占有处理器的时间。对于写至 EPTF 的 CICS，必须设置区域定义 (RD) 中的 **ExternalTrace** 属性。要获取如何使用 EPTF 的详细信息，请参阅 AIX 信息。

跟踪信息可在主存缓冲区中记录，但在大多数情况下主存缓冲区都很小且输入项可能被覆盖。要得到 CICS 不覆盖跟踪入口的 CICS 活动的跟踪，可以使用称为**辅助跟踪**的进一步的记录方法。CICS 把跟踪入口记录在另外的顺序文件中。辅助跟踪文件中的跟踪入口的格式与主存缓冲区中的相同。

可以使用一个或两个辅助跟踪文件。如果定义了两个文件并且第一个文件满了，可以在 CICS 运行时切换至第二个文件。

由于它产生的输出卷，所以应该仅在有限的时间周期激活辅助跟踪，如由于主存缓冲区未提供有效的信息而需要重复问题情况时。

可以各自激活主存缓冲区和辅助跟踪，无论先后。然而，除非主存缓冲区同时为活动的，否则 CICS 不记录辅助跟踪入口。

CICS 在文件分配结束时自动终止辅助跟踪输出。可以把输出导向两个文件中的任何一个，使输出在 CICS 打印其它文件时可进入另一个文件。然而，必须手工在文件间进行切换，而且 CICS 可能在该进程期间丢失一些跟踪。

在完成辅助跟踪且 CICS 关闭跟踪文件后，可以用提供读取、格式化和把跟踪数据打印进标准输出功能的跟踪格式化器 (**cicstfmt**) 来格式化和打印信息。直接从操作系统调用 **cicstfmt**。

使用辅助跟踪的一般注解

CICS 辅助跟踪是审查 CICS 性能问题的强大助手。然而，激活跟踪可能带来显著的附加系统开销，原则上是因为它带来了与 CICS 主存缓冲区相比额外的输入和输出操作

(辅助跟踪需要它来作为先决条件)。这意味着, 如果在最高加载时有性能问题, 开启辅助跟踪可能使其恶化。除了这点, 从辅助跟踪获得的数据财富可能对分析问题有帮助, 并且可能仍然承认系统的临时降低。

然而在计划得到辅助跟踪的数据之前, 应该检查所有可用信息的其它来源: 来自关机统计的 CICS 数据, CICS 监控设施等等。

— 相关信息 —

CICS 管理参考大全 中的 “*cicstfmt* - 跟踪格式化器”。

CICS 问题确定指南 中的 “使用 CICS 跟踪”提供有关所有 CICS 跟踪设施方面的信息, 包括操纵两个跟踪类型和脱机实用程序 **cicstfmt**。

附录A. CICS 环境中的安全性

CICS 的安全维护是通过:

- 请求至区域的连接的用户认证。
- 事务级安全性 (TSL) 和资源级安全性 (RSL) 访问的授权事务和资源。

本章描述了如何实现认证、TSL 和 RSL 安全性, 并描述了本地环境中使用的用户 CICS 安全性的基本概念。这包括区域 (服务器) 和终端客户间的安全性。建立在这些概念上的 *CICS 内部通信指南* 描述了如何在系统间通信环境中使用安全性设施。详情请参阅 *CICS 内部通信指南* 中的“系统间安全性配置”。

表 40. 安全性信息的导向图

若需...	参考...
确认 CICS 用户已认证。	第205页的『认证用户访问 CICS』。
使用 DCE 存取控制列表 (ACL) 和远程过程调用 (RPC) 来维护 CICS 安全。	第210页的『DCE RPC 和 ACL 如何维护系统的安全』。
限制至 CICS 事务的访问。	第214页的『授权访问 CICS 事务』。
限制至 CICS 资源的访问。	第218页的『授权用户访问 CICS 资源』。
与 CICS 一起使用外部安全性管理器 (ESM)。	第225页的『使用外部安全性管理器』。

认证用户访问 CICS

用户认证是一个靠 CICS 检查用户身份 (例如通过要求一个口令) 的进程。这是一个重要步骤因为 CICS 仅允许在认证用户后访问区域。

有两种方法来认证用户:

1. 使用 CICS 用户标识符和口令
2. 使用 DCE 委托人和口令

CICS 用户认证概述

要使用 CICS 来认证用户, 需要:

- 设置用区域定义 (RD) **AuthenticationService** 属性来配置的区域为 **CICS**。
- 区域定义 (RD) **RuntimeProtection** 属性设置为 **none**。

在仅 RPC 设置中, **cicsterm** 和 **telnet** 客户以区域缺省用户注册。如果用户不想以区域缺省用户方式操作, 那么必须使用事务 **CESN** 来提供一个不同的 CICS 用户标识符, 它可能需要一个取决于用户如何定义的口令。

也可使用 **CESN** 来更改口令, 在这种情况下, 口令被加密且存储在用户定义 (UD) 中。注意如果 **CESN** 失败了, 用户仍将以缺省用户标识符方式访问。

可以用 **-t** 选项启动 **cicsterm** 和 **cicsteld** 从而在一个步骤中调用 **CESN**。

DCE 用户认证概述

当配置一个区域以使用 DCE 安全性服务时，区域定义（RD） **AuthenticationService** 属性设置为 **DCE** 且为区域和区域用户创建 DCE 委托人。如果 **RuntimeProtection** 设置为大于 **none** 的值，用户必须首先注册 DCE 委托人来使用 CICS。

DCE 安全性服务认证用户并在 DCE RPC 中创建一个启用认证的时间戳记票券。时间戳记指出 RPC 服务器接受 RPC 请求中存在一个有限间隔。如果超过了间隔，DCE 需要另一个注册。CICS 不访问票券因为 DCE 认证在 CICS 外获得、存储、删除和重获得票券。

可以用 **-t** 选项启动 **cicsterm** 和 **cicsteld** 从而在一个步骤中调用 CESN。也可使用 CESN 来更改口令，在这种情况下，口令被加密且存储在用户定义（UD）中。注意如果 CESN 失败了，用户仍将以缺省用户标识符方式访问。

注：

1. 系统管理员要防止欺骗性使用 DCE 委托人帐户，应该注意删除一个帐户或更改帐户口令和『Good Since Date』没有妨碍委托人使用现存的服务票券。
2. 如果使用 CICS 控制程序（**cicscp**）来配置 DCE，则单元管理员委托人将与缺省口令一起创建（除非设置 **cell_admin_pw** 为环境变量）。

使用 **cicsterm** 时的 DCE 认证

要连接至使用 **cicsterm** 的区域，当 **RuntimeProtection** 大于 **none** 时，用户首先注册至 DCE。然后，当使用了 **cicsterm** 时，扫描用户定义（UD）输入项看该用户的 DCE 委托人是否存在。如果找到了委托人，则该用户以该委托人定义的 CICS 用户注册。虽然 CICS 在用户定义（UD）项中保持了 DCE 委托人的记录，但口令只有 DCE 和用户知道。

使用 **cicsteld** 时的 DCE 认证

当使用 Telnet 连接至 CICS 区域时，启动 **cicsteld** 进程时指定的委托人用于认证。这样，进入该端口的所有 Telnet 用户都使用相同的 DCE 委托人和 CICS 用户标识符。（参阅第74页的『Telnet 客户和 **cicsteld**』和第208页的『Telnet 客户的安全性考虑』以获得更多信息。）

缺省用户标识符如何用于 DCE 认证

每个 CICS 区域都有一个特殊的用户标识符叫作缺省用户标识符。这用户标识符由已注册到 DCE，但没有已定义的 CICS 用户标识符的用户使用。用区域定义（RD） **DefaultUserId** 属性来定义缺省用户标识符。参阅第233页的『创建区域的缺省用户标识符』，以获得更多信息。

任何不与 CICS 用户标识符关联的 DCE 用户都能以缺省用户标识符连接至 CICS。这样，用户定义（UD）中为缺省用户标识符定义的安全性密钥应该有限制性。这在第215页的『运行安全事务的实例』中有所描述。

如果 **RuntimeProtection** 设置为 **none**，则没有注册到 DCE 的用户可以以缺省用户标识符注册。必须注意要保证缺省用户标识符已定义为限制访问，例如，仅至公用资源。

终端保护

当连接 CICS 终端至 CICS 区域时，RPC 在区域的运行时间保护级别，或更高终端的运行时间保护级别。

可以用区域定义 (RD) **RuntimeProtection** 属性来设置区域运行时间保护级别。可以用终端定义 (WD) **TerminalProtection** 属性来设置终端运行时间保护级别。参阅 *CICS 管理参考大全* 中的“终端定义 (WD)”和“区域定义 (RD)”以获得这些属性的描述。

在事务路由选择情况 (已装运了终端) 中，终端的保护级别滞留本地至区域，且被路由的 WD 使用远程区域的运行时间保护级别。如 *CICS 内部通信指南* 中的“系统间安全性配置”所建议的，两个区域应该处于相同的保护级别。

用 DCE 委托人定义 CICS 用户

当为使用 DCE 安全服务的区域创建 UD 项时，CICS 管理设施为该用户创建 DCE 委托人和帐户。CICS 用户以两种方式之一连接至区域

1. 使用用 DCE 委托人定义的用户定义 (UD) 项，该委托人用 DCE 注册。
2. 使用区域的缺省用户标识符 (用 区域定义 (RD) **DefaultUserId** 属性定义)。例如，当用户没有已定义的 UD 项或当区域定义 (RD) **RuntimeProtection** 属性设置为 **none** 以及用户没有注册到 DCE 时，使用该 ID。

CICS 用户标识符的最大长度为 8 个字符。

CICS 用户标识符的 DCE 口令的最大长度为 8 个字符以允许用 CESN 来注册。

可以为用户帐户和口令设置到期日期。可以为每个个别的用户设置这些日期，或使用 DCE **dcecp** 命令将它们作为位置策略。

dcecp 命令能从 scripts 运行，这对于重复任务 (如更改口令) 很有用。

多注册用户的安全性

当 CICS 通过调用 CESN 事务和输入不同的用户标识符及口令或者通过使用 EXEC CICS SIGNON 命令来运行时，用户可以更改至另一个用户标识符。用户标识符和口令组合由 DCE 安全性服务或 CICS 根据区域定义 (RD) **AuthenticationService** 属性的值来验证。新的用户标识符和它的相关权限仍然与该终端关联，直到用户注销或注册至另一个用户标识符。

如果用户没有在启动 **cicsterm** 前注册至 DCE (仅当 **RuntimeProtection** 设置为 **none** 才可能)，则用户可能使用 CESN 将连接上的注册从缺省用户标识符更改至有适当访问权的用户。在仅 RPC 设置中经常出现这种情况。

DCE 安全性服务使您能为每个用户指定一个生命周期。周期结束时，用户失去网络上的认证，并被迫注销。用户必须在被允许返回 CICS 前再次 **dce_login**。该生命周期有最小值 (一分钟)，且可能存在系统大范围缺省值或个别设置选项。可以通过使用 **rgy_edit** 或 **dcecp** 命令来指定生命周期。

注：当正在使用 DCE 认证时，用户从 DCE 登录维护原始 DCE 上下文，忽略任何使用 CESN 的注册更改。

相同的用户标识符可以在多于一个并存终端上注册。CICS 不检查以前的注册中是否已使用了特定用户标识符和口令，DCE 也同样不查。

至区域的注册尝试无论成功或失败，都有一个记录保存在 `/var/cics_regions/regionName/console.nnnnnn` 中。

Telnet 客户的安全性考虑

如果系统为用户认证使用 DCE，则转至第208页的『使用 DCE 认证时的 Telnet 安全性』。如果系统为用户认证使用 CICS，则转至第208页的『使用 CICS 认证时的 Telnet 安全性』。

使用 DCE 认证时的 Telnet 安全性

当启动 Telnet 服务器 (**cicsteld**) 时，DCE 委托人能以参数之一 (**cicsteld -p principal**) 的方式经由它传送。当 Telnet 客户请求至区域的连接时，**cicsteld** 读取存储委托人和口令的键表文件，并传送委托人至 CICS 就象用户在使用 **cicsterm** 一样。然后区域根据从 **cicsteld** 传送至它的委托人来搜索匹配的用户定义 (UD)，如果找到的话使用该用户标识符以注册到该区域。用户然后被赋予至那些事务和资源 (用 UD 中用户标识符的 **TSLKeyList** 和 **RSLKeyList** 属性来指定) 的访问权。

这点的有效之处在于，访问区域上的 CICS 事务和资源已被授予任何注意到 **cicsteld** 正在侦听该端口的 Telnet 客户机用户。这样，如果您选择了一个不适当的委托人，会发现您的事务和资源都暴露给 Telnet 客户了。

要创建一个安全的环境，建议您执行下列操作：

- 使 Telnet 客户注册到 CICS，保证 CESN 事务作为 **cicsteld** 调用的第一个事务。使用 **cicsteld -t** 选项 (即 **cicsteld -p principal -t CESN**) 来达到这一点。

注：如果 CESN 失败，用户用缺省用户标识符注册。因而，应该用受限访问来定义缺省用户标识符。

- 限制访问 **cicsteld** 程序及小的受控制的用户集。可以改变可执行的 **cicsteld** 上的所有者和组许可权来执行这点。保证任何设置 **cicsteld** 客户程序的人都注意到所使用的 DCE 委托人的安全性蕴含式。
- 设置 **cicsteld** 以与 DCE 委托人一起运行，该 DCE 委托人是 UD 项中为区域的缺省用户标识符定义的。这意味着 Telnet 客户程序用户与未认证的用户有相同的存取权。

注：区域的缺省用户标识符由区域定义 (RD) 中 **DefaultUserId** 属性定义。

使用 CICS 认证时的 Telnet 安全性

如果区域为用户认证使用 CICS (如果区域没有 DCE 委托人)，那么如下显示启动 **cicsteld** 服务器：

```
cicsteld -t CESN
```

它调用 CESN 事务作为第一个事务，在这种情况下，任何在为 **cicsteld** 服务器定义的端口上使用 Telnet 的人员都需要为区域定义的 CICS 用户标识符和为该用户标识符定义的口令。CICS 口令是用用户定义 (UD) **CICSPassword** 属性定义的。当存储在资源定义数据库中时它被加密。

注：如果 CESN 失败了，用户仍将以缺省用户标识符方式访问。

cicsterm 客户的安全性考虑

要创建一个使用 DCE 的安全环境，建议您执行下列操作：

- 使 **cicsterm** 用户注册到 CICS，保证 CESN 事务作为 **cicsterm** 调用的第一个事务。可以使用 **cicsterm -t** 选项来做到这一点。

注：如果 CESN 失败，用户用缺省用户标识符注册。因而，应该用受限访问来定义缺省用户标识符。

- 限制访问 **cicsterm** 程序及小的受控制的用户集。可以改变可执行的 **cicsterm** 上的所有者和组许可权来执行这点。
- 设置 **cicsterm** 以与 DCE 委托人一起运行，该 DCE 委托人是 UD 项中为区域的缺省用户标识符定义的。这意味着 **cicsterm** 客户程序用户与未认证的用户有相同的存取权。保证任何设置 **cicsterm** 的人都注意到所使用的 DCE 委托人的安全性蕴含式。

注：区域的缺省用户标识符通过使用 **DefaultUserId** 属性在区域定义（RD）中定义。

XA 生效环境的安全性考虑

在 XA 生效的环境中从 CICS 存取的关系数据库的颗粒度处于区域级别。所有基于 XA 生效的数据库运行的事务用相同的用户标识符访问该数据库。

这是一个 XA 体系结构的蕴含式，其中 RM（资源管理器）认证在发出 **xa_open** 调用的点处发生。对于 CICS，在应用程序服务器初始化期间发出 **xa_open** 调用。在该点，用特定数据库用户标识符建立了至关系数据库的连接。

基于该特定应用程序服务器的每个事务都继承了已建立的数据库连接，并通过蕴含式继承已建立的数据库权限。同样，特定区域中的每个 CICS 应用程序服务器用相同的用户标识符连接至数据库（因为它使用相同的 **xa_open()** 字符串）。因而，每个执行 EXEC SQL 调用的事务使它们处于该特定数据库用户标识符的权限下。

应该将安全性策略基于启用 XA 的 CICS 环境中的 CICS 安全性设施。CICS 有控制用户访问系统和特定资源的设施，例如至它们的可用事务。这在第218页的『授权用户访问 CICS 资源』中有所描述。

使用 DB2，可以联接程序（或包）至数据库。对于使用静态的 SQL 语句的 CICS 事务，使用联接器的特权而非调用者的特权来访问数据库。因为所有事务都在相同的数据库用户标识符下执行，该用户标识符对涉及 CICS 事务的包必须仍被授予 EXECUTE 特权。使用动态 SQL 的 CICS 事务在运行时间已完成他们的访问认证，其它支持的 RDBMS 系统也使用同样方法完成。要获得 XA 环境中 DB2 安全性的进一步信息，请参阅 *DB2：管理指南*。

相关信息

第214页的『授权访问 CICS 事务』。

第218页的『授权用户访问 CICS 资源』。

第225页的『使用外部安全性管理器』。

第21页的『第2章 创建和配置 CICS 区域』。

第65页的『第4章 配置客户程序』。

CICS 内部通信指南 中的“系统间安全性配置”。

要获得下列信息，请参阅 *CICS 管理参考大全*：

- “cicsteld - 将 telnet 客户程序连接至区域”
- “cicsterm - 调用 CICS 3270 终端仿真器”
- “用户定义 (UD)”
- “文件定义 (FD)”
- “区域定义 (RD)”
- “CESN (注册)”

DCE RPC 和 ACL 如何维护系统的安全

CICS 为 区域、SFS 和客户间的通信使用远程过程调用 (RPC)。当区域定义 (RD) 中的认证服务属性设置为 **DCE** 时，可以使用已认证 RPC 来限制或禁止至区域的访问。已认证的 RPC 提供一个保护级别范围，从仅在 RPC 会话开始的校验认证到数据的加密。

用于设置已认证 RPC 保护级别的属性

RuntimeProtection 属性（区域定义 (RD)）指定了保护的最小级别，该保护用于 CICS 区域、CICS 服务器和文件间的 RPC 调用 RPC。另外，当 **RuntimeProtection** 不设置为无，资源保护属性如在第211页的表 42 中列出的 **LogicalTDQProtection**(区域定义 (RD)) 和 **FileProtection**(文件定义 (FD))，就能用于增加认证级别（用于 RPC 调用资源），但不能用于降低认证级别。如果 **RuntimeProtection** 设置为无，则忽略这些属性。

这些属性的每个都能设置为第211页的表 41 中描述的 DCE 认证值。注意，保护级别越高，对性能的影响越大。当 CICS 已初始化配置，则所有定义保护级别的值设置为 **none**。如果 **AuthenticationService** 属性设置为 **CICS**，则这些保护级别不必更改其缺省值。

注：当使用 **none** 时，没有用户定义 (UD)（为它们而定义）的用户以区域缺省用户标识符形式注册。这样，应该注意保证不赋予缺省用户受保护资源的访问权。

表 41. 已认证 RPC 的保护级别

保护级别	说明
none	DCE 不尝试认证 RPC。所有详细了解 DCE 的用户都能假冒一个 cicsterm 会话、区域或 SFS，并能观察正在委托人间网络上发送的所有数据。
连接	任何委托人第一次通信时，它们使用 cryptographically-secure 协议来认证双方。对随后的流动，每个委托人有一个服务号，并提供双方都同意的序列号，从而接受流动。
调用	每次一对委托人通信时，它们证实各自的身份给对方。所有数据都能象清除文本那样被发送。那是指应用程序（例如网络网桥或路由程序）能不被检测地改变发送中的数据，但不能插入额外的信息。
pkt	在整个 RPC（从客户到服务器）期间认证。
pkt_integ	每次一对委托人通信时，它们证实各自的身份给对方，并交换数据的 cryptographically-secure 校验和。所有数据都能象清除文本那样被发送，但任何介入应用程序都不能不被检测地改变信息。
pkt_privacy	如所有以前级别所指定地执行保护，并加密每个远程过程调用变量值。这个保护级别仅当 DCE 私有关支持安装时才可用。参阅本表底部的注意。
缺省值	如果没有明确指定其它保护级别，则以此为适用的保护级别。缺省保护级别由 DCE 供应商设置。
注： 1. 尽管 CICS 支持 pkt_privacy 的使用，该安全性级别的 DCE 软件可能在系统上没有。参考 DCE 产品信息。 2. 对于链路（它的通信定义（CD）有 ppc_tcp 的 ConnectionType ）上的通信，'connect' 和 'call' 设置为 'pkt'。 注：	

下表描述了对下列项如何设置保护级别：

- 区域
- 文件
- SFS
- 终端
- 网关服务器

表 42. 可用于为区域定义（RD）设置已认证 RPC 保护的属性

资源类	属性	在...间为 RPC 设置认证级别
区域（RD）	LogicalTDQProtection 逻辑可恢复的瞬时数据	当处理存储逻辑可恢复 TDQ 文件时的区域和 SFS。
区域（RD）	PhysicalTDQProtection 物理可恢复的瞬时数据	当处理存储物理可恢复 TDQ 文件时的区域和 SFS。
区域（RD）	NonRecTDQProtection 不可恢复的瞬时数据	当处理存储不可恢复 TDQ 文件时的区域和 SFS。
区域（RD）	RecTSQProtection 可恢复临时存储器	当处理存储可恢复 TDQ 文件时的区域和 SFS。
区域（RD）	NonRecTSQProtection 不可恢复的临时存储器	当处理存储不可恢复 TDQ 文件时的区域和 SFS。
区域（RD）	LocalQProtectProtection 可恢复队列启动	当处理存储本地队列受保护 START 的文件的区域和 SFS。
区域（RD）	LocalQProtection 不可恢复队列启动	当处理存储本地队列不受保护 START 的文件的区域和 SFS。
区域（RD）	RuntimeProtection RPC 运行时间	控制从一个区域至另一个区域的 RPC 认证。

表 42. 可用于为区域定义 (RD) 设置已认证 RPC 保护的属性 (续)

资源类	属性	在...间为 RPC 设置认证级别
文件 (FD)	FileProtection RPC 认证级别	当处理已命名文件时的区域和 SFS。
SFS (SSD)	ProtectionLevel RPC 认证级别	SFS 和任何需要 SFS 的。
终端 (WD)	TerminalProtection RPC 认证级别	区域和终端。
网关服务器 (GSD)	ProtectionLevel RPC 认证级别	区域和 PPC 网关服务器。

了解 DCE 组和缺省 ACL

当区域设置为使用 DCE DCE 安全性服务时, CICS 设置存取控制列表 (ACLs) 来保护您的资源。ACL 是控制存取至 DCE 对象的授权机制。它是一个列表, 该列表指定每个列出的用户都允许执行一个对象的操作。可能需要为下列原因之一更改 ACL :

- 如果要增加安全性
- 如果要增加 RPC 保护级别
- 如果 SFS 已自动启动且已更改保护级别
- 如果要另一个 DCE 单元的用户访问服务器

要更改 ACL 并正确设置它们, 必须了解 CICS 区域和 SFS 是如何使用 DCE 委托人及组的。

为 CICS 配置 DCE 安全性服务器 (使用 **cicssetupdce** 命令) 时, 定义下列 DCE 安全性组用于那个 DCE 单元:

cics_admin

包含帐户的委托人, 该帐户被允许去管理 CICS 区域和服务, 例如 DCE 委托人 **cell_admin**。例如, 要将一个模式用于 SFS, 您必须作为 **cics_admin** 组的成员被认证。

cics_regions

包含每个区域的委托人。

cics_users

包含用户的委托人。

cics_sfs

包含所有 SFS 的委托人。

cics_ppcgwy

包含 PPC 网关服务器的委托人。

当已为 CICS 客户配置了第一个 DCE 客户时, 就在每个机器上创建了一个附加的 DCE 安全性组 (参阅第72页的『为与 CICS 客户机使用建立 DCE』)。每个组的 ACL 都已设置, 以允许这些组的成员去执行特定任务。

对于区域

每个 CICS 区域建立一个 DCE 委托人, 该委托人名由区域名和前缀 **cics/** 形成。例如, 区域 **SALES** 的委托人, 它的区域名将是 **cics/SALES**。这个委托人的帐户成为 DCE 组 **cics_users** 和 **cics_regions** 的一员。当该区域启动时, 该帐户的口令随机

产生并以键表文件的加密形式存储，而且口令按需要周期性地重新产生。任何用户（除了 root）或程序都不能存取键表文件来获得区域的委托人口令。

对于 SFS 和 SFS 文件

当启动了 SFS，且它的 **ProtectionLevel** 属性设置为需要认证的级别之一，则设置 SFS 文件的 ACL 以便于仅允许组 **cics_regions** 的成员存取。这意味着用户不能直接存取文件。必须使用 CICS，因为只有 CICS 区域可以访问文件。设置 SFS 数据文件的 ACL（瞬时数据队列、临时存储器队列和未期满 START 请求）来仅允许区域委托人存取。

如果非认证地运行 SFS，ACL 设置成允许任何委托人和非认证用户访问服务器。SFS 数据文件也允许任何委托人访问。

注：当配置了一个区域给 SFS 服务器时，**RuntimeProtection**（区域定义（RD））属性与 **SFS ProtectionLevel**（结构化文件服务器定义（SSD））属性一起用来为 SFS 服务器上创建的 CICS 文件确定 ACLS 设置。**RuntimeProtection** 属性（区域定义（RD））应该设置成与为 SFS 服务器设置的 **ProtectionLevel** 相同或更高层次的认证。

认证策略

当区域设置来使用 DCE 安全性服务时，DCE 创建一组可应用于跨越该单元的安全性标准或规则。DCE 单元的安全性标准设置称为认证策略。

例如，可以应用认证策略于指定用户单元：

- 帐户的生命周期
- 口令的最大和最小长度
- 口令必须隔多少时间新建

可以修改缺省认证策略并创建仅适用于特定单元用户组的附加策略。（要获得更多有关 DCE 委托人、组和组织的详细信息，请参阅 DCE 文档编制。）

假定区域委托人的认证策略指定了最大口令周期。如果正在运行 CICS 区域时委托人的口令期满了，口令将由 CICS 自动更改。如果停止 CICS 区域时区域委托人的口令期满了，则区域无法启动。

要启动区域必须或者调出后再重调入回区域，或者手动更改区域委托人的口令。

如果客户委托人的认证策略指定了最大口令周期且口令期满了，CICS 将不更改口令。

要生成客户委托人新的随机口令，需要运行 **cicssetupclients** 命令。

注：如果要防止欺骗性使用 DCE 委托人帐户，应该注意删除一个帐户或更改帐户口令和『Good Since Date』没有妨碍委托人使用现存的服务票券。

赋予访问权给另一个单元的区域

要允许当前 DCE 单元以外的区域存取文件，必须更改 SFS 文件的 ACL 并记录文件以便于它们能接受另外区域的委托人。

使用带外部安全性管理器的 DCE ACL 外部安全性管理器 (ESM)

CICS 不能使用 ACL 和 RPC 来限制访问特定用户。它却可使用 RSL 和 TSL 密钥来确定哪些用户可以运行哪些事务及哪些资源。你也可以写入一个使用 ACL 的 ESM，以授权特定用户运行事务和访问资源。参阅第226页的『定义 CICS 和 ESM 间的接口』，以获得更多信息。

授权访问 CICS 事务

本节描述了 CICS 提供的安全性机制限制访问指定事务。

访问用户定义事务

CICS 执行安全性检查来确定是否应该允许用户或打印机终端使用两种方法之一访问特定事务：

- 通过运行用户提供的外部安全性管理器 (ESM) 程序
- 通过运行内部 CICS 检查

可以定义事务处理定义 (TD) **TSLCheck** 属性使用的方法。

当 **TSLCheck=external** 时，CICS 传送安全性检查至 ESM。如果 **ESMLoad=no**，当区域开始警告 ESM 未装入且对事务缺省为 CICS 内部检查不安全性时 CICS 发出一个信息。这也适用于区域生命期内已定义的事务。

要获得 ESM 的详细细节，请参阅第225页的『使用外部安全性管理器』。

当 **TSLCheck=internal** 时，CICS 内部安全性检查使用 TD **TSLKey** 属性和用户定义 (UD) 及终端定义 (WD) **TSLKeyList** 属性来检查用户和终端是否应该被允许访问事务。

TD **TSLKey** 属性定义了值范围从 1 至 64。UD 和 WD **TSLKeyList** 属性被赋予一个值列表 (从 1 至 64)。**TSLKeyList** 必须有相同的定义值，以便于被授予许可权去访问事务。

例如，如果赋予事务 **TSLKey=4**，赋予用户 **TSLKeyList=3|5|6**，则用户被拒绝访问事务。

当终端是由 **cicstermp** 管理的打印机时，使用终端定义 (WD) **TSLKeyList** 属性。

可以定义事务为有公用访问 (**TSLKey=public** 或 **TSLKey=1**)。在这种情况下，为每个访问事务的尝试 (是或否) 假设值 **public** 和 **1**：

- **TSLKeyList** 中包括 **1**。
- 存在用户的 UD 项。
- 用户标识符或终端与事务关联

打印机终端的事务安全性

CICS 允许您联系 TSL 密钥和 RSL 密钥至终端定义 (WD)。这是指事务能在无用户注册的终端上运行。这用于打印机连接至 **cicstermp** 终端 (仅输出)。

可以定义打印机终端为仅输出设备。通过运行 **cicstermp** 给出终端网络名、区域名和任何打印机选项来启动打印机终端。

一旦启动了 **cicstermp**，事务可直接通过自动事务启动（ATI）（例如显示 EXEC CICS START TRANSID 命令）或触发器 TDQ 在该终端上运行。要决定事务是否能启动，系统比较事务的 TSL 密钥和终端的 TSL 密钥表。如果事务的 TSL 密钥在终端的列表中，事务能在打印机终端上启动。

假设有一个事务 PAYR（TSL 密钥为 15），且有一个打印机连接至 termid 为 LPT1 的终端（有 1、8、15 和 23 的 TSLKeyList）。如果输入下列命令，则执行安全性检查，且因为终端有正确的 TSL 密钥，事务运行：

```
EXEC CICS START TRANSID(PAYR) TERMID(LPT1)
```

仅当没有用户注册时才需要终端密钥，这样只有打印机连接的终端才应该有定义的终端安全性密钥。

自动事务启动的安全性

可以为事务指定安全性，它自动通过瞬时数据队列或 CICS 间隔控制程序启动至终端。（参阅 *CICS 内部通信指南* 中的“自动事务启动（ATI）”。）要保证可以无安全性违例地启动所有自动启动事务，则：

- 如果事务要在用户注册的终端上运行，确认该终端的用户有正确的 TSL 密钥来运行该事务。
- 如果事务要在无用户注册的终端（如打印机）上运行，确认该终端有定义在终端定义中正确的 TSL 密钥来运行事务。
- 如果事务不在终端上运行，注意无终端 ATI 能拥有与之相关联的用户标识符。安全性密钥是从可能的用户标识符取得的。否则仅公用事务和资源是可用的。启动 START 的用户的 **TSLKeyList** 必须包含要启动事务的 TSL 密钥。

注：START API 检查事务的 TSL 密钥以保证能发出 START，而不启动事务本身。

对于由瞬时数据队列触发器启动的事务，发出触发队列的写请求的用户必须有运行该事务的正确 TSL 密钥。参阅第223页的『使用 EXEC CICS START 或 EXEC CICS CANCEL 的事务』，以获得更多信息。

运行安全事务的实例

假设用户的定义包含 TSL 密钥 1、2 和 4，用户系统中两个事务是 ACCO 和 PERS。ACCO 的 TSL 密钥为 2，PERS 的 TSL 密钥为 5。

如果用户调用 ACCO 事务，CICS 比较事务 ACCO 的 TSL 密钥和用户 TSL 密钥列表中的密钥。因为列表中 ACCO 的 TSL 密钥为 2，所以用户被特许运行事务。

如果用户尝试调用 PERS 事务，CICS 比较 PERS 的 TSL 密钥（5）和用户 TSL 密钥列表中的密钥。CICS 在用户列表中没有找到能匹配 PERS 事务的 TSL 密钥，则执行下列步骤：

- CICS 不运行事务。
- 事务 PERS 以异常终止码 A146（安全性违例）异常终止。
- CICS 带适当细节地发送信息 ERZ014016 至终端。
- CICS 将错误信息 ERZ014016 和 ERZ014001 写入 console.nnnnnn 日志。这让检查记录的操作员知道哪个用户导致违例。

- CICS 将错误信息 ERZ045004 写入 CSMT 日志。这指定了它是一个事务安全性违例，也指定了试图运行 PERS 的用户和终端。

授权访问 CICS 提供的事务

CICS 提供了很多事务给辅助控制和使用。这些事务的错误使用能严重影响操作。

CICS 提供的事务用事务处理定义 (TD) 中的标准事务安全性值 **1** 或 **public** 定义。这简化了 CICS 的初始安装，但有缺点，即它允许所有的 CICS 用户调用这些事务。

CICS 注册功能

除了运行 CRTE 来路由事务至另一个系统外（从而在这儿注册），用户仅需要 CESN 来更改他（或她）的权限。例如，缺省时所有用户都能以相同权限级别运行，但偶尔可能需要使用不同的 ID 来执行安全事务。

注：如果正在使用 DCE 认证，则用户从 DCE 登录维护原始 DCE 上下文，忽略任何使用 CESN 的注册更改。

CESN 和注销事务 CESF 缺省时定义 **TSLKey** 为 **1**。因为大多数用户喜欢使用这些事务，所以不必常规更改这两个事务的缺省安全性设置。

系统事务的安全性

事务如 CEMT（主终端）、CECI（命令解释器）和 CEDF（执行诊断设施）执行一些功效强大的功能。考虑应用安全性至这些事务，并需要特权用户的特定注册来运行它们。

例如，程序员能使用 CECS 事务来交互检查命令语法。他们也可以使用 CECI 事务来执行命令，而不用写程序。因为用该设施很容易更改系统中的数据，所以您可能想要控制 CECI 的使用。

在产品系统中，可以提供 CEDF 和 CECI，但必须适当地控制和监督它们。应该使用事务安全性和资源安全性检查来严密控制它们，当遇到异常问题或调试不能进行时，系统程序员将保留使用它们。

授权访问 CEDF 调试事务

如果事务在调试方式（`cicstran -e`）中编译，则可以使用 CEDF 事务来调试另一个事务。

如果 CICS 系统赋予您访问 CEDF 事务权，CICS 执行事务安全级（TSL）验证来验证您是否有足够的权限来调试事务，如需要的话，为正在运行事务的用户和作为正在调试事务的 CEDF 用户的您执行随后的资源安全级（RSL）验证。

当验证 CEDF 调试用户时，CICS 执行 TSL 检查稍有不同。CEDF 事务和调试事务需要有效的 TSL 安全性值，因此必须保证 CEDF 事务和正在调试的事务在事务处理定义（TD）中有相同的 **TSLCheck** 属性（**internal** 或 **external**）值。

如果 CEDF 需要资源验证（TD 中的 **RSLCheck** 属性不等于 **none**），则保证 TD 中需求 CEDF 和被调试的事务有相同的 **RSLCheck** 值（**internal**、**external** 或无）。如果满足了所有这些需求，则应该能从正在运行 CEDF 的终端运行事务。

如果正在使用 CEDF 且安全性检查不成功，则在屏幕上显示下列信息：

Security violation: Security Check does not allow debug access

CICS 通过使用您的用户标识符、终端标识符和链路标识符（如果是事务路由选择）为您执行 RSL 验证，并以相同方式作用于事务用户。类似地，对于外部检查，CICS 能传递信息至 外部安全性管理器（ESM），描述用户标识符、终端以及适当的话，来自事务路由选择的链路。

事务安全性检查表

1. 决定事务安全性的程度。它应该适用于每个事务、每个事务组或仅为选定的系统？
事务安全性越复杂，管理和维护越困难。平衡需要以保护使用需要。
2. 分配事务安全性密钥（1-64）给涉及的事务标识符。
3. 检查事务定义（用户和 CICS 提供的）。按需要更改 TSL 密钥为正确值。
4. 如果正在使用事务的外部安全性，请确认将使用外部安全性的事务和资源的定义 RSLCheck 属性设置为 **external**。
5. 建立您的 CICS 用户列表。需要：
 - 用户名称
 - 唯一的用户标识符
 - 关联口令
 - 适当的安全性密钥列表，匹配用户要访问的任务
6. 为更新用户信息建立一个过程。可以建立表格描述：
 - 请求者（名称、电话号码、位置）
 - 要更改什么（要添加或删除的任务、口令复位）
 - 注册用户标识符和口令（仅安全性 admin 字段）
 - 管理协议

记住使用 SMIT 至 CICS 的定义用户也可以定义它们至 DCE。CICS 用户属性必须在 SMIT 中更改，但非 CICS 用户属性（如票券周期）应该在 DCE 的 rgy_edit 中更改。
7. 告诉用户：
 - 用户标识符和口令
 - 他们能使用什么事务
 - 安全性何时实现（如果没有预先完成）
 - 发生困难时应与谁联系，例如安全性管理员
8. 如果有任何打印机连接至 CICS，且连接的终端有允许访问安全事务的 TSL 或 RSLKeyList，则该打印机应该在安全的区域中。
9. 确认监控尝试违例信息的适当日志，记录下原因。参阅
`/var/cics_regions/regionName/console.nnnnnn` 和
`/var/cics_regions/regionName/data/CSMT.out`。
10. 如果从测试系统移动至产品系统，检查什么用户（若有的话）需要访问 CEMT、CEDF 和 CECI。确认没有人（除了那些非常需要使用这些事务的人）能访问它们。
11. 检查每个用户的注册以保证他们能：

- 以他们的用户标识符和口令注册。
- 使用允许他们使用的事务和资源。

相关信息

第225页的『使用外部安全性管理器』。

CICS 管理参考大全 中的“事务定义 (TD)”包含有关 **TSLCheck** 和 **RSLCheck** 属性的信息。

请在 *CICS 管理参考大全* 中参阅下列信息:

- “终端定义 (WD)”
- “用户定义 (UD)”
- “用户定义 (UD)”
- “CICS 提供的事务处理”
- “CESN (注册)”

授权用户访问 CICS 资源

资源安全级 (RSL) 验证为来自应用程序的运行时间资源访问提供安全性。

RSL 验证也适用于 CICS 设施，该验证是基于事务安全性相似概念之上的。要获得有关事务安全性的信息，请参阅第214页的『访问用户定义事务』。

运行时间以外的访问

运行时间区域以外有几种可用的 CICS 设施。这些设施包括:

- CICS 转换器 **cicstran**
- 基本映像支持 (BMS) 处理器 **cicsmap**
- 资源定义文件
- Telnet 客户程序的接口命令 **cicsteld**
- 下列 CICS 资源管理命令: **cicsadd**、**cicsdelete**、**cicsupdate** 和 **cicsget**

要点: 当区域使用 CICS 口令检查代替 DCE 安全性服务 (如果区域定义 (RD) 中 **AuthenticationService=CICS**) 时，加密的 CICS 口令存储于用户定义 (UD)。可以使用 **cicsupdate** 和 **cicsadd** 来更改 CICS 口令。任何 cics 组中的用户都可以更改口令。

至所有这些设施的访问权限由 AIX 而不是 CICS 提供。根据站点上需要的安全性级别，可以改变 CICS 安装这些资源时赋予它们的许可权。

授权访问启动和关闭程序

必须以 root 身份注册才能使用任何启动和关闭方法。这些方法是:

- CICS 控制程序 (**cicscp**)
- **cicsstart** 或 **cicsstop** 命令

- 系统管理界面工具（SMIT）

授权用 CICS 命令访问资源定义

一些 CICS 资源管理命令用来更新永久和运行时间数据库中的资源定义，（如 第21页的『第2章 创建和配置 CICS 区域』中所描述的）。可以使用文件和目录的许可权来防止未授权的用户调用个别命令。例如 **cicsupdate**。

另外，也可以通过调用系统管理界面工具（SMIT）界面执行 CICS 资源的联机定义，从而发出 CICS 资源管理命令。可以使用 AIX 许可权来仅允许特许的用户调用 SMIT。

因为可能需要限制一些 SMIT 选项的使用到特定用户，该方法可能不令人满意。在这种情况下，应该使用个别命令上的 AIX 许可权。请注意如果使用 SMIT 对话框来调用未授权的命令，则当试图调用实际命令时允许执行整个对话框。在这个阶段，因为安全性违例而不执行该命令。

使用操作系统的许可权来控制何种用户能执行何种文件操作，及那些操作应该是什么。例如，可能需要：

- 允许系统管理员读、添加、更新和删除记录。
- 允许特定组成员读和更新资源定义文件。
- 允许所有用户读资源定义文件。

为了防止资源运行时间定义的未授权更新，必须为执行该任务的 CICS 私有事务 CURD 设置事务安全性。有关进一步信息，请参阅第214页的『授权访问 CICS 事务』。

授权访问运行时间区域以外的资源

CICS 赋予它使用的对象下列许可权：

表 43. CICS 产品文件必需的许可权

文件类型	许可权	组
资源定义文件	rw-rw----	cics
可执行的运行时间	r-x-----	cics
运行时间创建和使用的文件	所需的 r----- 或 rw-----	cics
可执行的非运行时间，除了： cicstran 、 cicsmap 、 cicstcl 、 cicsterm 、 cicstermp 和 cicsteld	r-xr-x---	cics
cicstran 、 cicsmap 和 cicstcl	r-xr-xr-x	cics
cicsterm 、 cicstermp 和 cicsteld	r-xr-xr-x	cicsterm
客户的 DCE 键表文件	rw-rw----	cicsterm

如果区域为安全性使用 DCE，则 CICS 终端程序的所有用户必须拥有对 /var/cics_clients 目录下 DCE 键表文件的读和写访问权。

从应用程序授权访问 CICS 资源

CICS 根据您是否对给定的资源有访问许可权，通过使用安全性密钥来执行安全性检查。可以为每个资源指定 CICS 是否通过事务在 TD 项中设置 **RSLCheck** 属性，从而在事务上执行这些检查。

由 CICS 所做的 RSL 安全性检查可以或者是由 CICS 执行的内部检查 (**RSLCheck=internal**)，或者是由用户提供外部安全性管理器 (ESM) 程序所执行的外部检查 (**RSLCheck=external**)。

资源安全级 (RSL) 验证可能会执行的资源是：

- 事务
- 程序
- 文件
- 瞬时数据队列
- 临时存储器队列
- 日志

也可以指定对于特定事务完全不需要 RSL 验证。这通过在所需的 TD 项中设置 **RSLCheck** 属性为 **none**。

通过比较 TD 中 **RSLCheck** 属性和区域定义 (RD) 中挂起资源的特定 **RSLCheck** 属性来定义给定资源所执行的安全性检查。这些属性在 RD 中如下定义：

- **TransactionRSLCheck**
- **ProgramRSLCheck**
- **FileRSLCheck**
- **TransientDataRSLCheck**
- **TemporaryStorageRSLCheck**
- **JournalRSLCheck**

当两种资源类型都在 RD 中且事务指定所需的外部验证时，ESM 仅执行资源级验证。

如果已为事务在 TD 项中指定了 **RSLCheck=external**，并在 RD 中设置了 **ESMLoad=yes**，且装入 ESM 失败，则区域不能启动。如果已为事务在 TD 项中指定了 **RSLCheck=external**，并在 RD 中设置了 **ESMLoad=no**，这样就不存在 ESM 程序，则出现启动区域时未装入 ESM 的警告信息，区域的资源检查缺省为内部检查。这也适用于区域生命期内已定义的事务。详情参阅第225页的『使用外部安全性管理器』。

如果 ESM 程序的返回码指明赋予访问权给资源失败，CICS 会终止应用程序服务器和事务。

如果需要内部安全性检查，CICS 安全性模块实现一个验证方案。对于每个资源都有一个 **RSLKey** 属性包含在为特定资源提供的定义中。可以赋值下列 **RSLKey** 值给每个资源：

- 由数字表示的密钥 (1 至 24)
- **public**
- **private**

关联用户标识符的用户定义 (UD) 项包括一个 **RSLKeyList** 属性。该属性定义了允许用户访问什么资源。可以在 UD 的资源安全性键表中最多指定 24 个键。

当用户试图访问给定资源时，CICS 从适当的资源定义的运行时间版本获得与该资源相关的 **RSLKey**。如果键在用户 **RSLKeyList** 中指出，则 CICS 允许用户访问资源。如果没有，则 CICS 阻止用户访问资源并指出该请求的错误。

终端定义 (WD) 中包含了相似的 **RSLKeyList** 以在没有关联事务的用户标识符处执行安全性检查。

CICS 在特定环境中使用终端键。这种情况，如终端是由特许用户启动的 **cicsternp** 进程管理的打印机。

在所有其它情况下，用户键由 CICS 使用。

适当的资源定义中的某些特殊的 **RSLKey** 值由 CICS 用来指出安全性检查总能成功。可以为资源提供 **RSLKey=public** (赋予任何用户以访问权)，或 **RSLKey=private** (赋予任何用户以访问权)。**RSLKey=private** 密钥保证资源仅能由 TD 项中拥有属性 **RSLCheck=none** 的可信任事务来访问。要访问 **public** 资源，用户不需要在他们的 **RSLKeyList** 中有任何键。

用户不能直接访问 **private** 资源。要经由可信任事务 (**RSLCheck=none**) 访问 **private** 资源，用户需要适当的 **TSLKey** 或 **RSLKey** 来使用该事务，然后事务就能更新 **private** 资源。

使用 TSLKeyList 和 RSLKeyList 属性

基于 **TSLKeyList** 和 **RSLKeyList**，事务和资源在独立的 CICS 系统中由 CICS 检查。该系统由 用户定义 (UD) 或终端定义 (WD) 建立。

当用户成功建立了一个 CICS 会话，CICS 就获得了与 UD 的用户标识符相关的安全性密钥。CICS 使用这些密钥来为事务和资源执行安全性验证，这些事务和资源与他（或她）正在使用的终端设备相关联。然而，一些事务与用户不能注册的设备相连接，例如打印机。尽管设备的 UD 中没有项，仍能提供事务和资源安全性，在 WD 中定义事务和资源安全性密钥。终端密钥仅在不能执行注册处使用，例如正在执行事务的打印机设备。当为这些设备执行安全性验证时，CICS 使用这些密钥。

资源安全性的更多信息

资源安全级 (RSL) 是 CICS 安全性检查的第二层，该安全性检查扩展事务安全性概念到资源，如文件、瞬时数据队列和应用程序。

要关联安全性和系统中的资源，指定是否对于给定事务执行 RSL 检查。这是指为每个资源建立一个 RSL 密钥，当用户输入一个事务标识符时，用户密钥适用于任何该事务访问的资源。

定义资源安全性包含三个步骤。必须：

- 明确说明要为事务执行 RSL 检查。
- 为事务访问的每个资源定义一个 RSL 值（或 **public**，或 **private**）。
- 限制用户和终端的 RSL 密钥于用户访问的资源关联的密钥值。

事务的资源安全性

首先必须定义想要在事务运行的任何时间内执行 RSL 检查。通过在事务处理定义（TD）的 **RSLCheck** 字段中设置适当的值来做到这一点。**RSLCheck** 字段有三种可能状态，**none**、**internal** 或 **external**。

要使用内部安全性 RSL 密钥来保护资源，应该确认事务的 **RSLCheck** 字段设置为 **internal**。（要使用外部安全性管理器来检查资源，则设置事务的 **RSLCheck** 字段为 **external**。）

对于提供的事务（例如 CECI、CPMI、CEDF、CEBR、CSM1 和 CSMI）**RSLCheck=internal** 为您缺省指定。这是指在命令解释器、执行诊断或内部通信支持会话期间的所有被访问的资源都必须定义适当的 RSL 密钥。如果这不可接受，可以改变这些事务的定义，指定 **RSLCheck=none**。然后就能依靠事务安全性来控制使用。

如果事务不与终端（例如，运行在后台的 ATI 任务）关联，但与用户关联，那么资源安全性检查将与用户关联。如果事务不与终端或用户关联，则它不应该使用资源安全性检查。

资源的资源安全性

在为事务指定了 RSL 检查后，下一步是将 RSL 密钥和事务访问的每个资源关联起来。**RSLKey** 属性将 RSL 检查与下列资源关联起来：

- 文件定义（FD）中定义的文件
- 瞬时数据定义（TDD）中定义的瞬时数据队列
- 临时存储器定义（TSD）中定义的临时存储器队列
- 日志定义（JD）中定义的日志
- 程序定义（PD）中定义的应用程序
- 事务处理定义（TD）中定义的事务

注：关系数据库管理器（RDBM）保持的资源不受 RSL 检查控制，应该使用 RDBM 提供的安全性机制来保护它。

每个资源可能仅有一个赋值给它的 RSL 密钥。相同的密钥必须包含在用户 **RSLKeyList** 中以允许用户访问该资源。

参阅下表以获得 **RSLCheck** 和 **RSLKey** 属性各种值的允许访问的类型。（如果 **RSLCheck** 是 **external**，则与 **RSLKey** 值无关。带关联用户标识符的非终端事务（例如功能装运镜像）与安全性检查的终端连接事务一样操作。没有用户标识符的非终端事务仅能访问公用资源。

RSL key 值	RSLCheck=none（不检查）	RSLCheck=internal（检查）
RSLKey=private	允许访问	拒绝访问
RSLKey=public	允许访问	允许访问
RSLKey=1-24	允许访问	允许访问（如果键匹配）

RSLKey 的 **private** 值是一个特殊的值，它表示资源仅能由拥有 **RSLCheck** 为 **none** 值的事务来访问。

如果用户尝试访问一个资源，但该资源安全性密钥不匹配，则提出 NOTAUTH 条件。可以在应用程序中提供您自己的代码来处理 NOTAUTH 条件，例如通过显示一个信息，说明用户不被特许地访问该资源。信息 ERZ045001 被写入 CSMT 日志以处理当系统管理员试图启动资源的未授权访问。（详情请参阅 *CICS 应用程序设计指南* 中的“处理错误条件”。）

文件、瞬时数据和临时存储器

在很多安装中，最重要的资源是使用的数据文件。内部安全性 **RSLKey** 属性给予您控制存取那些文件的机会。

对于 FD 中定义的每个文件，应该在访问该文件的用户的 UD 或 WD 中指定 **RSLKey** 值（从 1 至 24）。只要事务访问文件有 **RSLCheck=internal**，该文件就受保护。

瞬时数据队列用于为进一步处理排列数据。这些队列可能也包含敏感数据。对于这些内分区和附加分区瞬时数据队列，可以定义在它们的 TDD 定义中定义 **RSLKey** 值。

临时存储器可能包含敏感数据，所以应该在这些队列的 TSD 定义中设置适当的 **RSLKey** 值来保护它们。

当从 TSD 中的模板创建临时存储器队列时，临时存储器从模板继承该保护设置。

应用程序

也可以将应用程序考虑为一个资源。记住将每个事务定义与应用程序关联起来从而为该事务启动数据处理。不过使用 LINK 和 XCTL 命令，一个程序能与其它程序一起通信。

应用资源安全性于 CICS 应用程序以保证在适当的方式中控制它们的使用。程序只能 LINK 或 XCTL 到用户在 **RSLKeyList** 中有 RSL 密钥的其它程序。可以在程序定义（PD）中为每个程序定义 RSL 密钥。

使用 EXEC CICS START 或 EXEC CICS CANCEL 的事务

这适用于面向时间启动（EXEC CICS START）和取消（EXEC CICS CANCEL）请求。

当要事务执行资源安全性检查时，必须设置它的 **RSLCheck** 值为 **internal**。这是指事务拒绝访问那些不为用户（或终端，如果没有关联用户）匹配值的带 RSL 密钥的资源。

现在假设那些资源中的一个为另一个事务。仅当应用程序为另一个事务标识符发出 START 或 CANCEL 命令时，CICS 对事务执行访问的资源级安全性检查。RSL 检查是针对目标事务的 RSL 密钥的。RSL 检查是当为事务发出 START 或 CANCEL 命令时所做的。

例如，假定有一个运行程序 **persprg** 的事务 PERS。事务 PERS 的 **RSLCheck** 字段定义为 **internal**。这是指当事务 PERS 正在运行时（即程序 **persprg** 正在运行），资源安全性检查在 PERS 访问的资源上已完成。当程序 **persprg** 为不同的事务（例如事务 PAYR）发出 START，PAYR 事务的 **RSLCheck** 字段在该点无关。因为 PERS 有 **RSLCheck=internal**，PAYR 必须有适当的 RSL 密钥以便执行。它必须有 **RSLKey=public** 或者有 **RSLKey=nn**，nn 是在 **RSLKeyList** 中用户运行 PERS 时使用的 1 和 24 间的数字。

仅在成功 RSL 检查后，CICS 才尝试运行事务。然后执行 TSL 检查。启动 PAYR 后，PAYR 的 TD 中的 **RSLCheck** 字段才确定它是否当 PAYR 访问资源时应用资源安全性检查。

在已为事务指定 **RSLCheck** 为内部后，必须在事务使用的所有资源上有适当的 **RSLKey** 值。如果任何那些资源的 **RSLKey** 为 **internal**，资源的所有已尝试访问将被拒绝。

日志的资源安全性

当与其它资源一起时，日志将受资源安全性检查控制，这样就能为每个日志设置 **RSLKey** 值。

确认用户有适当的 AIX 安全性级别，因为日志以平面文件形式驻留在文件系统中。

用户的资源安全性

当已为资源安全性设置好所有事务和资源后，最后步骤是建立赋予访问给事务和资源的各种的 RSL 密钥。通过在 UD 的 **RSLKeyList** 属性中定义密钥来指定每个用户拥有的密钥。

不同于事务安全性密钥，密钥为 1 与公用访问不同。如果资源定义 **RSLKey** 为 **public**，则用户 **RSLKeyList** 中不需要 RSL 密钥。然而，如果资源的 **RSLKey** 为 1，用户需要 **RSLKey** 为 1 来访问它。

记住如果资源的 **RSLKey** 为 **private**，则用户有什么 **RSLKey** 值都没有关系。访问取决于 **RSLCheck** 值。如果用户想要包含所有可能密钥（1 至 24）的密钥表，则可以设置列表为 **all** 代替 **1|2|3...|24**

终端的资源安全性

连接至打印机的终端应该定义有适当的 **TSLKey** 和 **RSLKey** 值，这样当执行安全事务（发送输出到打印机终端、没有关联用户的终端）时，有权运行它。RSL 密钥赋予终端启动事务必需的访问必需资源的权限。

TSL 和 RSL 两者的终端密钥都能在终端定义中设置。与其它 **TSLKeyList** 和 **RSLKeyList** 一起时，TSL 密钥能从 1 至 64，RSL 密钥能从 1 至 24。

资源安全性检查表

RSL 检查的实现计划比安全事务的实现计划涉及更多。要维护安全的资源（例如，FD 中的文件）可能被很多任务访问。可能既要实现 RSL 检查，又要实现事务检查。

1. 确定维护何种资源的安全。这些可能是在 FD、TSD、TDD、JD、PD 或 TD 中。
2. 分配给每个资源一个资源安全性密钥（1-24）、**public** 或 **private**。如果有很多要维护安全的资源，特别是 CICS 资源和使用它的任务间的交叉引用，可能想要保持一个写记录。当更新用户定义时需要这点。这些数字不以任何方式与 TSL 密钥联系。
3. 更改所有受影响资源的定义来反映他们新的 RSL 密钥。
4. 更改事务定义中的 **RSLCheck** 字段为 **internal**。
5. 如果任何连接打印机将打印安全资源，改变它们的终端定义以赋予它们 RSL 密钥。

6. 更改用户定义以包括适合每个用户的 **RSLKeyList** 。通过检查他们能访问什么事务，以及那些需要 **RSL** 密钥访问的事务访问了什么资源来完成该项。
7. 注意现在可能出现资源安全性违例的记录，当它们出现时应该审查它们。
8. 让您的用户知道新资源的安全性策略。
9. 确认 **CEMT**、**CECI** 和 **CEDF** 使用资源检查。

相关信息

第214页的『授权访问 CICS 事务』。

CICS 内部通信指南 中的“CICS 链路安全性”。

请在 *CICS 管理参考大全* 中参阅下列信息：

- “关闭 CICS 系统”
- “用户定义 (UD)”
- “终端定义 (WD)”
- “通信定义 (CD)”
- “资源定义”

使用外部安全性管理器

可以使用外部安全性管理器（ESM）或与 CICS 内部安全性一起来控制存取 CICS 资源和事务。ESM 是提供给用户的程序，它允许您定义系统自身的安全性机制以防止未授权的用户从应用程序和未授权的 CICS 事务启动访问资源。

如何与 CICS 一起使用 ESM

缺省时 CICS 使用内部安全性检查。要使用外部安全性管理器来代替 CICS，必须：

- 安装外部安全性管理器。
- 更改区域定义（RD）**ESMLoad** 属性为 **yes**。
- 使用 RD **ESMModule** 属性来指定 ESM 程序路径和名称。

资源类型的外部资源检查能在 RD 中通过设置下列任何 RD 属性为 **external** 来设置：

- **FileRSLCheck**
- **JournalRSLCheck**
- **ProgramRSLCheck**
- **TemporaryStorageRSLCheck**
- **TransactionRSLCheck**
- **TransientDataRSLCheck**

访问这些资源的事务必须指定它们要使用 ESM，

ESM 执行的区域的安全性检查码取决于 CICS 内部安全性执行的检查量。可以使用 ESM 来增强或替换内部安全性检查。

CICS 资源或事务上的信息由 CICS 传送至 ESM。ESM 必须使用它，和其它可能的信息来确定权限是否允许。通过接口必须通知 CICS 系统 ESM 的判定。启动时你必须标识 ESM 至 CICS 系统；也要标识 CICS 系统至 ESM。

如果在区域的生命周期更改 ESM，必须执行热启动使 ESM 适用于所有应用程序服务器。否则，更改的 ESM 仅适用于 ESM 更改后启动的应用程序服务器中运行的任务，这些任务将导致区域内不一致的安全性。

最好设计 ESM，使它存储的数据独立于程序本身。任何时候无需重新启动 CICS，就可以访问该数据。

定义 CICS 和 ESM 间的接口

当需要执行定义为外部的安全性检查时，可以为 CICS 安装 ESM 模块来自动调用。CICS 提供该模块的实例实现，来论证写自己的 ESM 时需要符合什么接口。要获得该提供模块的信息，请参阅 *CICS 管理参考大全* 中的“外部安全性管理器用户程序”。

注：CICS 为安全性检查将用户标识符和与之关联的 DCE 委托人从用户定义（UD）传送至 ESM。在找不到 UD 映射的情况下，不传送委托人。如果要 ESM 检查基于 DCE 委托人，而不是等价的 CICS。请执行下列事项：

- 对于每个 DCE 委托人，保证有与系统定义的相同名称相应的 CICS 用户标识符。
- 在 ESM 中实现映射机制以映射每个 CICS 用户标识符至关联的 DCE 委托人。

ESM 是各自编译的模块，它在接口中实现三种功能至 CICS 安全性模块：初始化功能、**RSLCheck** 和 **TSLCheck**。如果需要这个模块，则 load 子程序在 CICS 启动时将它装入 CICS 系统。ESM 的初始化功能称为 **CICS_ESM_Init**。

三种功能名只用于参考目的。CICS 系统仅从安全性模块而非调用命名功能，通过地址间接访问这些功能。这样当实现一个替代 ESM 时，就可以在 ESM 模块中定义实际使用的名称。这些功能在下面描述：

初始化函数

CICS 启动进程调用初始化函数以使 ESM 能执行任何系统特定初始化，并建立 ESM 回叫功能函数地址，该函数执行事务级和资源级安全性检查。该函数是装入模块的入口点。load 调用返回该函数的地址给 CICS，所以它不必链接至 CICS。CICS 也提供该函数描述 CICS 区域的数据。执行事务级和资源级安全性检查的两个回叫函数的地址以数据形式返回。

传送至初始化函数的参数为：

- RegionData
- FnBuf

TSLCheck 功能

CICS 调用事务安全性回叫函数以执行外部事务安全级（TSL）验证。CICS 提供该函数以描述 CICS 区域、用户、事务、终端和程序的适当数据。

传送至 TSLCheck 函数的参数为：

- RegionData
- ContextData

RSLCheck 功能

CICS 调用资源安全性回叫函数以执行外部资源安全级（RSL）验证。CICS 提供该函数以描述 CICS 区域、用户、事务、终端、程序和资源的适当数据。

传送至 RSLCheck 函数的参数为:

- RegionData
- ContextData
- ResourceData
- ResourceClass
- TaskData
- EIB_Fn

注: 当在已定义事务上使用 CEDF 以使用 ESM 时 (**RSLCheck** 和 **TSLCheck** 设置为 **external**)，将为正在诊断的事务和 CEDF 传送上下文和资源至 ESM。

适于 ESM 调用的资源数据由 CICS 以一系列结构引用变量 (每个相关资源一个) 方式提供。所有的结构包含至少一个公共标识符字段, CICS 资源名。每个结构类型由各种字段组成, 该字段包含适于问题中资源类型的信息。这些资源结构的使用允许您无须中断现存的 ESM 代码进行进一步更改。

ESM 功能参数

RegionData

它描述 CICS 区域至 ESM。它包含 CICS 区域的系统标识符和 CICS 发行版本号。

FnBuf 该结构是 ESM 函数缓冲区, 它用于返回执行 ESM 验证的回叫函数的地址。该结构使您能添加新的 ESM 函数。

ContextData

该结构包含描述当前事务、用户、终端、程序、DCE 委托人和链路区域的子结构。因为这时没有程序在运行, 所以 TSL 检查的程序子结构为空。链路区域子结构为空, 除非事务从远程系统预定。

ResourceData

CICS 将各种不同的资源定义数据传送至 ESM 以使它能验证至特定资源的访问。该数据的转换取决于资源类。ESM 调用中传送一个类属资源数据值。ESM 使用资源类变量来分解它成特定类型。资源类为:

事务 资源数据中事务包括的字段是:

- 事务标识符。
- 内部/远程标识符。
- 远程事务标识符。
- 远程系统标识符。

程序 资源数据中程序包含的唯一字段是程序标识符。

文件 资源数据中文件包括的字段是:

- 文件标识符。
- 内部/远程指示符。
- 远程文件标识。
- 远程系统标识符。
- SFS 文件卷。
- SFS 文件名。

- SFS 文件索引。
必须定义 SFS 文件名组件的长度，该文件名组件用于传送文件数据结构至这儿的 ESM。卷名当前定义为 256 个字符长，文件名 256 个字符，索引名 16 个字符。

瞬时数据队列

资源数据中瞬时数据队列包括的字段是:

- 瞬时数据队列标识符。
- 瞬时数据队列类型。它向 ESM 指出队列类型，及什么字段（远程瞬时数据标识符、远程系统标识符、附加分区和间接名称）是有效的。它保持值 INTRA、EXTRA、REMOTE 或 INDIRECT 之一。
- 远程瞬时数据队列标识符。
- 远程系统标识符。
- 附加分区名称。
- 间接名称。

临时存储器队列

资源数据中临时存储器队列包括的字段是:

- 临时存储器队列标识符。
- 临时存储器队列类型。它向 ESM 指出队列类型，及临时存储器队列标识符、远程系统标识符是否有效。它保持值 MAIN、REMOTE、AUXILIARY 和 TEMPLATE 之一。
- 远程临时存储器队列标识符。
- 远程系统标识符。

日志 资源数据中日志包含的唯一字段是日志标识符。

ResourceClass

CICS 传送该属性至 ESM RSL 验证函数以分解要验证的资源标识符类型和资源特定数据。

TaskData

CICS 提供 ESM 两个用户数据区域的地址，公共工作区（CWA）和用户区（TCTUA）。这些地址在结构中以指针方式传送。在任务上下文不可用的情况下，它可能为空指针。

EIB_Fn

这是 EXEC CICS 操作的双字节码，该操作执行从 EXEC 接口块（EIB）的 FN 字段获得的资源访问。

CICS 在该点提供各种标识符的长度和名称字段（不包括末端的空字节）给 ESM。

长度字段和它们的缺省值在下列表中显示:

表 44. 缺省 ESM 参数字段长度

字段名	缺省长度
文件标识符长度	8 字节
系统标识符长度	4 字节
瞬时数据队列标识符长度	4 字节

表 44. 缺省 ESM 参数字段长度 (续)

字段名	缺省长度
附加分区名称长度	127 字节
间接名称长度	4 字节
临时存储器队列标识符长度	8 字节
日志标识符长度	2 字节
程序标识符长度	8 字节
事务标识符长度	4 字节
终端标识符长度	4 字节
用户标识符长度	8 字节
发行版标识符长度	4 字节

从初始化函数返回的代码是整数零表明成功，任何其它值表明发生了错误。

从 **TSLCheck** 和 **RSLCheck** 函数返回的指示符可能是下列三种值的一种:

CICS_ESM_RETURN_OK

这是指访问给定的事务或资源是有效的。

CICS_ESM_RETURN_INVALID

这是指访问给定的事务或资源是无效的（即 NOTAUTH）。

CICS_ESM_RETURN_FAIL

这是指给定的事务或资源不能外部验证。信息 ERZ045014 是为 **TSLCheck** 函数所发，ERZ045013 是为 **RSLCheck** 函数所发。CICS 终止任务应用程序服务器。这意味着验证中发生了一个错误（ERROR）。当应用程序服务器终止时，一些 ERZ087001 信息写入了 console.nnnnnn。这些信息指出与应用程序服务器端接相关联的 RPC 异常。

如果当出现应用程序服务器的异常终止信号时，由 ESM 请求安全性检查的任务保持任何关键锁定，则任务的终止可能导致区域的终止。

启动时定义 ESM

在每个应用程序服务器初始化期间，CICS 检查区域定义（RD）中 **ESMLoad** 属性值。如果 **ESMLoad** 设置为 **yes**，CICS 将 ESM 模块装入应用程序服务器存储区。如果 ESM 装入失败，区域将不能启动。如果 ESM 程序的返回码指明失败，CICS 会终止应用程序服务器和事务。

RD 中定义的属性 **ESMModule**，用于 load 调用的变量。**ESMModule** 或者是至 ESM 模块的绝对路径名，或者是相对名。如果使用了相对路径名，那么 LIBPATH 环境变量必须包含含有 ESM 模块的目录。

load 调用返回 ESM 初始化函数的入口点地址。当建立 ESM 模块时，必须定义该入口点地址。CICS 安全性模块调用 ESM 的初始化函数,允许 ESM 执行任何特定初始化。ESM 初始化函数使用一个返回码来表明初始化的成功。如果 ESM 安装失败，启动进程没有完成，且区域关闭。

如果 **ESMLoad** 设置为 **no**，则发出事务或资源，该事务或资源已定义为需要外部安全性缺省使用内部安全性检查和警告信息。因此，建议任何事务或资源的 **TSLKey** 和 **RSLKey** 属性（可能受其影响的）应该设置某些值，这些值当区域不使用 ESM 的启动时也适合于提供保护。

定义 CICS 至 ESM

可能需要定义 CICS 系统的组件至 ESM。该定义方法是 ESM 的责任，但您可能需要保证该定义在 ESM 能用于 CICS 安全性检查前已完成。

用 CICS 系统事务来使用 ESM

CICS 运行它自己的一些 CICS 用户标识符下的系统事务，该用户标识符与区域名相匹配，且运行空的 CICS 用户标识符下的其它系统事务。请确认您的 ESM 不禁止至这些事务的访问。

相关信息

第214页的『授权访问 CICS 事务』。

第218页的『授权用户访问 CICS 资源』。

CICS 管理参考大全 中的“区域定义 (RD)”包含有关 RD 中 **ESMLoad** 和 **ESModule** 属性的信息。

请在 *CICS 管理参考大全* 中参阅下列信息：

- “外部安全性管理器用户程序”
- “事务定义 (TD)”
- “程序定义 (PD)”
- “文件定义 (FD)”
- “临时存储器定义 (TSD)”
- “瞬时数据定义 (TDD)”
- “日志定义 (JD)”

附录B. 使用 CICS 命令的 CICS 系统管理过程

表 45. 使用 CICS 命令的 CICS 系统管理过程的道路映象

若需...	参考...
设置 DCE 职员和 CICS 客户。	第232页的『用 cicssetupclients 设置 DCE 和 cicsterm 客户』。
设置一个使用用于 CICS 队列和文件管理的本地的 SFS 的区域。	以该次序: <ol style="list-style-type: none">1. 第233页的『创建区域的缺省用户标识符』。2. 第233页的『使用 cicsdefault 定义区域』或 第234页的『使用 cicsexport 和 cicsimport 定义区域』。3. 第235页的『为区域设置 CICS_HOSTS』(仅 RPC 设置)。4. 第236页的『创建 SFS 的用户标识符』。5. 第236页的『准备 SFS 的物理存储设备』。6. 第237页的『使用 cicssfscree 定义 SFS』。7. 第238页的『设置 SFS 的联接文件』(仅 RPC 设置)。8. 第239页的『使用 cicssfss 启动 SFS』。9. 第239页的『使用 cicssfssconf 设置队列』。10. 第247页的『从命令行启动区域』。
设置一个使用用于 CICS 队列和文件管理的远程的 SFS 的区域。	以该次序: <ol style="list-style-type: none">1. 第233页的『创建区域的缺省用户标识符』。2. 第233页的『使用 cicsdefault 定义区域』或 第234页的『使用 cicsexport 和 cicsimport 定义区域』。3. 第235页的『为区域设置 CICS_HOSTS』(仅 RPC 设置)。4. 第240页的『使用本地命令设置队列』。5. 第247页的『从命令行启动区域』。 <p>这些过程假设 SFS 是最初设置为用于本地区域的。</p>
设置使用用于 CICS 队列和文件管理的 DB2 版本 2 数据库的区域。	以该次序: <ol style="list-style-type: none">1. 第233页的『创建区域的缺省用户标识符』。2. 第233页的『使用 cicsdefault 定义区域』或 第234页的『使用 cicsexport 和 cicsimport 定义区域』。3. 第235页的『为区域设置 CICS_HOSTS』(仅 RPC 设置)。4. 第243页的『为队列和文件管理设置 DB2』。5. 第246页的『用 cicsdb2conf 配置资源和 DB2』。6. 第247页的『从命令行启动区域』。
从命令行关闭区域。	第247页的『从命令行关闭区域』。

表 45. 使用 CICS 命令的 CICS 系统管理过程的道路映象 (续)

若需...	参考...
使用 <code>cicssfssshut</code> 关闭 SFS。	第138页的『使用 <code>cicscp stop sfs_server</code> 关闭 SFS』。
使用 <code>cicsdestroy</code> 删除区域。	第249页的『使用 <code>cicsdestroy</code> 删除区域』。
使用 <code>cicssfdestroy</code> 删除 SFS。	第249页的『使用 <code>cicssfdestroy</code> 删除 SFS』。

有关使用基本命令来设置 CICS 的信息

该章节描述了如何使用以 SMIT 或 **cicscp** 命令所使用的基本命令来配置 CICS。

cicscp 是设置 CICS 所推荐的方式。不过，当您正进行令人烦恼的配置时您可使用基本命令，在该情况下，它可帮助一步步设置每个 CICS 组件。

对于更多信息，参阅 第21页的『第2章 创建和配置 CICS 区域』。

使用基本命令的用来设置 CICS 的过程

该章节包含了使用基本命令设置区域和文件管理器的过程。按照 第231页的表 45 中的指导使用这些过程。

用 **cicssetupclients** 设置 DCE 和 **cicsterm** 客户

该过程描述了如何使用 **cicssetupclients** 命令来配置 **cicsterm** 客户。

客户当它安装时进行最初配置。仅当需要为客户重新配置 DCE 时需要该过程。对于更多信息，参阅 第251页的『附录C. 使用 DCE』。

1. 登录一个有超级用户（root）特权并如 第3页的『建立 CICS 系统管理员』中所描述而设置的用户标识符。
2. 如果正在一个 DCE 单元上工作，那么作为单元管理员（`dce_login cell_admin`）登录 DCE。
3. 使用或者步骤 第232页的 (3.a) 或者步骤 第232页的 (3.b)，这取决于您是否在 DCE 单元中工作：
 - a. 在 DCE 单元，使用该命令：
cicssetupclients
 - b. 在仅 RPC 设置中，使用该命令：
cicssetupclients -m
4. 按照 第73页的『设置客户的语言』。

对于更多信息，参阅在 *CICS 管理参考大全* 中的“**cicssetupclients** - 为 CICS 主机配置 DCE”。

创建区域的缺省用户标识符

使用该过程来创建区域的用于没有与已登记用户标识符相关联的任务以及一些用户任务的缺省用户标识符。用户任务为:

- 当尝试使用 CESN 登记失败时
- 当使用 CRTE 布线至另一个系统, 并且使用 CESF 注销时, 将使用缺省用户标识符直到使用 CESN 返回登录

用户作为缺省用户标识符登记至 CICS 区域当:

- 在 区域定义 (RD) 中 **AuthenticationService=CICS** 时
- 在 RD 中 **AuthenticationService=DCE** 并 **RuntimeProtection=none**, 并且用户不是 DCE 登录的时
- **AuthenticationService=DCE** 并 **RuntimeProtection** 大于 **none**, 并且 DCE 用户没有 CICS 用户定义 (UD) 项时

对于系统间的请求, 以下属性与缺省用户标识符一起使用:

- 通信定义 (CD) **LinkUserId** 属性以及定义在 UD 项目中的用于用户标识符的以该属性指定的 TSL 和 RSL 关键字
- CD **RemoteSysSecurity** 属性
- CD **TSLKeyMask** 和 **RSLKeyMask** 属性

要创建区域缺省用户标识符:

1. 定义一个用户
2. 设置 RD **DefaultUserId** 属性为 UD 项目的那个用户的关键字。

有关缺省用户标识符和安全性: 因为缺省用户标识符典型地用作对于一个非认证用户的注册, 所以当定义 UD 项时要小心以确保 **RSLKeyList** 和 **TSLKeyList** 属性仅允许对公共事务的存取。

使用 cicsdefault 定义区域

本过程的目的是使用 **cicsdefault** 命令定义区域。该命令使用缺省区域定义。

使用该过程以前, 应该复查在 *CICS 管理参考大全* 中的“cicsdefault - 创建系统默认区域的副本”。

使用该过程:

1. 登录一个有超级用户 (root) 特权并如 第3页的『建立 CICS 系统管理员』中所描述而设置的用户标识符。
2. 如果正在一个 DCE 单元上工作, 那么作为单元管理员 (dce_login cell_admin) 登录 DCE。
3. 使用以下例子之一:

在 DCE 单元

使用该命令:

```
cicsdefault -r regionName
```

该命令:

- 创建包括缺省区域资源定义的区域目录结构。在区域的目录结构中的文件的所有者是组 **cics**。
- 创建区域的 DCE 委托人和帐户
- 创建区域的键表文件。
- 创建定义在区域的资源定义中的所有用户的 DCE 委托人和帐户。

在 仅 RPC 设置中

使用该命令:

```
cicsdefault -r regionName -S
```

该命令:

- 创建包括缺省区域资源定义的区域目录结构。在区域的目录结构中的文件的所有者是组 **cics**。
- 设置 区域定义 (RD) **NameService** 属性从 **DCE** 至 **NONE** 并且设置 **AuthenticationService** 属性从 **DCE** 至 **CICS**。

使用 **cicsexport** 和 **cicsimport** 定义区域

用该过程以使用一个现存的区域定义设置来创建一个新的区域。该过程使用详细描述在 *CICS 管理参考大全* 中“**cicsexport** - 导出 CICS 资源定义”和“**cicsimport** - 导入 CICS 资源定义”中的 **cicsexport** 和 **cicsimport** 命令。在使用这些命令以前, 应该复查在那些章节中的信息。

注意: 该过程创建在 `/var/cics_regions` 中的区域的一个拷贝。在在使用该命令以前, 验证在 `/var` 中有用于拷贝的足够的空间。

1. 登录一个有超级用户 (root) 特权并如 第3页的『建立 CICS 系统管理员』中所描述而设置的用户标识符。
2. 如果正在一个 DCE 单元上工作, 那么作为单元管理员 (dce_login cell admin) 登录 DCE。
3. 使用步骤 第234页的 (3.a) 或 第235页的 (3.b) :

- a. 如果用于旧区域的 DCE 配置是与用于新的区域的相同的:

- 1) 使用 **cicsexport** 命令进行现存区域的拷贝。例如, 下列命令输出 **regionA** 配置至命名为 **TEMPFILE** 的文件:

```
cicsexport -r 'regionA' -o 'TEMPFILE'
```

- 2) 使用 **cicsimport** 命令从档案文件定义新的区域。参阅以下例子:

```
cicsimport -r 'regionB' -i 'TEMPFILE'
```

该命令:

- 建立下列区域目录结构:

```
/var/cics_regions/regionB
```

在区域的目录结构中的文件的所有者是组 **cics**。

- 用新的区域名称修改 **TEMPFILE** 中的资源定义。
- 如果在 DCE 单元中配置 **regionA**:
 - 为 **regionB** 创建 DCE 委托人和帐户。

- 为 regionB 创建键表文件。
 - 为定义至 regionA/regionB 的所有用户创建 DCE 委托人和帐户。
- b. 对于仅 RPC 设置配置新的区域:
- 1) 使用 **cicsexport** 命令进行现存的区域的拷贝。例如, 下列命令输出 regionA 配置至命名为 TEMPFILE 的归档文件:


```
cicsexport -r 'regionA' -o 'TEMPFILE'
```
 - 2) 使用 **cicsimport** 命令从档案文件定义新的区域。参阅以下例子:


```
cicsimport -r 'regionB' -i 'TEMPFILE' -S
```

该命令:

- 建立下列区域目录结构:

```
/var/cics_regions/regionB
```

在区域的目录结构中的文件的所有者是组 **cics**。

- 用新的区域的名称修改 TEMPFILE 中的资源定义并设置 区域定义 (RD) **NameService** 属性为 **NONE** 以及 **AuthenticationService** 属性为 **CICS**。

为区域设置 CICS_HOSTS

本过程的目的是对于仅 RPC 设置设置 CICS_HOSTS 环境变量。

使用该过程:

- 必须配置区域以设置 /var/cics_regions/*regionName*/environment 中的环境变量。
- CICS 客户没有先决条件任务或条件。

CICS_HOSTS 环境变量的语法为:

```
CICS_HOSTS='host1 host2 hostn ... hostn'
```

在 CICS_HOSTS 中列出的主机为:

- 全部有客户需要连接至的 CICS 区域的主机
- 本地主机 (如果在它上面有区域)
- 所有带有 Encina 服务器的主机 (PPC 网关服务器 或 SFS)
- 区域需要连接至的任何其它 CICS 区域。

例如, 如果 CICS_HOSTS='hostA hostB hostC', 那么 CICS 对于它正在寻找的区域或服务器的字符串联接信息将搜索 hostA。如果它不存在于那主机上, 则搜索 hostB, 然后是 hostC。

设置该环境变量于:

- CICS 区域的 /var/cics_regions/*regionName*/environment 中。
- CICS 客户的用户的环境或 /etc/environment 中。

如果没有设置 CICS_HOSTS, 那么假设为字符串联接信息并且因此和服务器一起都在本地主机上。

-h 选项也可与 **cicsterm**、**cicsteld** 以及 **cicsternp** 一起使用来指定主机。该选项覆盖 CICS_HOSTS 环境变量的内容。

创建 SFS 的用户标识符

该过程描述了如何创建 SFS 的操作系统用户标识符。SFS 用户标识符的请求为:

- 如果使用 **cicscp** 配置 SFS, 那么 SFS 用户标识符的名称必须为以 S 开头的主机名的开始七个字符。例如, 如果主机名为 **hostABCD**, 那么 SFS 用户标识符的名称必须为 **ShostABC**。

如果使用资源定义命令或 SMIT 替代 **cicscp** 以配置 SFS, 那么 SFS 用户标识符的名称必须是 **SFS_SERV**。

- 用户标识符的主组必须为 **cics**。
- SFS 用户标识符的自用目录必须为:

```
/var/cics_servers/SSD/cics/sfs/serverId
```

其中 *serverId* 为 SFS 的唯一的标识符。 *serverId* 使用的缺省值为主机名。

- 核心文件可代表 SFS 用户标识符创建。要确保捕捉全部核心文件, SFS 用户标识符的最大的核心文件大小值应该设置为 81950 块。

注意, 如果创建核心文件, 那么更改该值至 81950 可能需求多至 40Mb 的磁盘空间, 如第236页的『准备 SFS 的物理存储设备』中描述。

1. 登录一个有超级用户 (root) 特权并如 第3页的『建立 CICS 系统管理员』中所描述而设置的用户标识符。
2. 使用 **mkuser** 命令或 SMIT 来创建 SFS 用户标识符。例如, 如果使用 **cicscp** 配置 SFS, 并且如果主机名为 **hostABCD**, 那么您可使用下列命令:

```
mkuser pgrp='cics' home='/var/cics_servers/SSD/cics/sfs/hostABCD \
      core='2097152' ShostABC
```

在该例子中, **cics** 为主组, **/var/cics_servers/SSD/cics/sfs/hostABCD** 为自用目录, 并且 **ShostABC** 为用户标识符。

3. 如果要能搜集并阅读 SFS 的磁心信息转储, 确保用户的最大的核心文件大小是适当的。
4. 用户标识符定义必须不是来自网络信息服务 (NIS)。

准备 SFS 的物理存储设备

Encina SFS 需要两个逻辑卷; 一个用于日志数据并且一个用于队列和文件数据。SFS 逻辑卷的大小应该为至少每个 64 兆字节 (MB)。附加的 64 兆字节 (MB) 需要用于每个附加的区域。

结构化文件服务器定义 (SSD) **DataVolume** 和 **LogVolume** 属性用来指定 SFS 逻辑卷的名称。缺省值为 **sfs_shortName** 和 **log_shortName**。SSD **ShortName** 属性的缺省值取决于您如何配置 SFS:

- 如果使用 **cicscp** 配置 SFS, 那么 **ShortName**, 缺省地, 设置为以字母 S 开头的主机名的开始七个字符。例如, 如果主机名是 **hostABCD**, 那么, SFS 的缺省短名称为 **ShostABC**。逻辑卷的缺省名称于是为 **log_ShostABC** 和 **sfs_ShostABC**。
- 如果使用 **cicssfscreate** 或 SMIT 来配置 SFS, 那么 **ShortName**, 缺省地设置为 **SFS_SERV**。逻辑卷的缺省名称于是为 **log_SFS_SERV** 和 **sfs_SFS_SERV**。

如果准备在配置在 SFS 时使用缺省值, 则当设置逻辑卷时您需要使用缺省值。

使用该过程:

- 必须作为超级用户 (root) 登录并且超级用户 (root) 必须如 第3页的『建立 CICS 系统管理员』中所描述而建立。
- SFS 用户标识符必须如 第236页的『创建 SFS 的用户标识符』中所描述进行创建。

设置 SFS 逻辑卷:

1. 使用操作系统命令或 SMIT 来创建逻辑卷。例如, 如果主机名为 **hostABCD**, 那么您可使用下列命令:

```
mklv -y log_ShostABC rootvg 16
mklv -y sfs_ShostABC rootvg 16
```

在该例子中, **rootvg** 为卷组, 并且 **16** 为逻辑分区的推荐数。逻辑分区为 4 兆字节 (MB)。

该命令放置逻辑卷在 /dev 目录中。

2. 使用 **chown**、**chmod** 和 **chgrp** 命令来设置存取许可权以及逻辑卷的所有权。例如:

```
chown ShostABC /dev/log_ShostABC
chown ShostABC /dev/sfs_ShostABC
chown ShostABC /dev/rlog_ShostABC
chown ShostABC /dev/rsfs_ShostABC
chmod 777 /dev/log_ShostABC
chmod 777 /dev/sfs_ShostABC
chmod 777 /dev/rlog_ShostABC
chmod 777 /dev/rsfs_ShostABC
chgrp cics /dev/log_ShostABC
chgrp cics /dev/sfs_ShostABC
chgrp cics /dev/rlog_ShostABC
chgrp cics /dev/rsfs_ShostABC
```

使用 **cicssfscreate** 定义 SFS

cicssfscreate 命令的目的为:

- 使用要么缺省值要么 **cicssfscreate** 命令指定的值来创建 SFS 的结构化文件服务器定义 (SSD)。
- 创建 SFS 的子系统项。
- 如果您指示 DCE 将用于安全性, 那么该命令创建:
 - SFS 的 DCE 委托人
 - 存取控制列表 (ACL)
 - 键表文件

第236页的『准备 SFS 的物理存储设备』过程必须在该过程可做以前进行。

定义 SFS:

1. 登录一个有超级用户 (root) 特权并如 第3页的『建立 CICS 系统管理员』中所描述而设置的用户标识符。
2. 作为单元管理员 (dce_login cell_admin) 登录 DCE。
3. 使用 **cicsdefaultservers** 命令创建 SFS 目录结构。
4. 取决于您是否在 DCE 单元上工作, 使用步骤 第238页的 (4.a) 或 第238页的 (4.b) :

- a. 在仅 RPC 设置, 使用该命令:

```
cicssfscree -S ./cics/sfs/serverId \  
ShortName=shortName
```

- b. 在 DCE 单元, 使用该命令:

```
cicssfscree ./cics/sfs/serverId \  
ShortName=shortName \  
ProtectionLevel=level
```

在这些例子中, 没有指定用户标识符和逻辑卷。逻辑卷的缺省名称为 **log_shortName** 和 **sfs_shortName** 并且 SFS 用户标识符的缺省名称为短名称。如果没有设置物理设备来使用这些名称 (参阅 第236页的『创建 SFS 的用户标识符』) 那么使用 **LogVolume** 和 **DataVolume** 属性来更改缺省值。如果没有使用短名称设置用户标识符 (参阅 第236页的『准备 SFS 的物理存储设备』) 那么使用 **UserID** 属性来更改缺省值。例如:

```
cicssfscree ./cics/sfs/hostABCD \  
ShortName=ShostABC \  
ProtectionLevel=connect \  
LogVolume="accountlog" \  
DataVolume="accountdata" \  
UserID="accountsfs"
```

对于用来定义 SFS 至 CICS 的所有属性的完整说明, 参阅在 *CICS 管理参考大全* 中的“结构化文件服务器定义(SSD)”。

设置 SFS 的联接文件

本过程的目的为在仅 RPC 配置中设置 SFS 的一个联接文件。

设置联接信息包含了把信息添加至:

```
/var/cics_servers/server_bindings
```

注:

1. 如果该 Encina 服务器是在该机器上由 CICS 为其使用而配置的第一个服务器, 该文件可能不存在。
2. 检查 `server_bindings` 文件是否有 `cics:cics` 的所有权以及 660 许可权。

```
chown cics server_bindings  
chgrp cics:cics server_bindings  
chmod 660 server_bindings
```

联接文件用来指定服务器名以及字符串联接。在联接文件中的服务器名必须为全限定名。字符串联接包含一个协议序列、网络地址或主机名, 以及一个端点。例如, `hostABCD` 项可为:

```
./cics/sfs/hostABCD ncadg_ip_udp:hostABCD[1234]
```

该字符串联接显示了一个数据报协议 (UDP/IP) 序列的例子。网络地址为主机的名称。在方括号中的数字为端点。

以下为使用 TCP 序列的联接字符串的一个例子:

```
./cics/sfs/hostABCD ncacn_ip_tcp:hostABCD[4529]
```

端点为系统相关的。对于 TCP/IP 协议, 端点为端口号。咨询 `/etc/services` 文件以确定哪一个端口号还没有在使用, 然后选择一个没有列出的号码。一旦已选择一个端点, 您应该把一个项目添加至 `/etc/services` 中。

1. 登录一个有超级用户（root）特权并如 第3页的『建立 CICS 系统管理员』中所描述而设置的用户标识符。
2. 设置 ENCINA_BINDING_FILE 环境变量为 /var/cics_servers/server_bindings。
3. 把联接字符串添加至 /var/cics_servers/server_bindings 并更新 /etc/services 文件。

使用 **cicssfs** 启动 **SFS**

该过程的目的是启动 **SFS**，这样可把 **CICS** 队列和文件添加至 **SFS** 逻辑卷。

以下过程必须在该过程可做以前进行：

- 第236页的『创建 **SFS** 的用户标识符』
- 第236页的『准备 **SFS** 的物理存储设备』
- 第237页的『使用 **cicssfscreate** 定义 **SFS**』
- 第238页的『设置 **SFS** 的联接文件』（仅 **RPC** 设置）

注意：

冷启动 **SFS** 消除了正在 **SFS** 逻辑卷上的文件。以下过程告诉您何时冷启动以及何时自动启动。

要启动 **SFS**：

1. 登录一个有超级用户（root）特权并如 第3页的『建立 **CICS** 系统管理员』中所描述而设置的用户标识符。
2. 如果正在一个 **DCE** 单元上工作，那么作为单元管理员（**dce_login cell_admin**）登录 **DCE**。
3. 使用 **cicssfs** 命令启动 **SFS**。例如：

```
cicssfs /./cics/sfs/hostABCD
```

对于有关当启动 **SFS** 时使用的参数的更多信息，参阅在 **CICS** 管理参考大全 中的“结构化文件服务器定义(SSD)”和“**cicssfs** - 启动 **CICS SFS**”。

也可参阅 第123页的『第7章 启动 **CICS** 系统』。

使用 **cicssfsconf** 设置队列

该过程的目的是使用 **cicssfsconf** 命令把文件添加至用于 **CICS** 队列管理的 **SFS** 逻辑卷中。

以下过程必须在该过程可做以前进行：

- 第236页的『创建 **SFS** 的用户标识符』
- 第236页的『准备 **SFS** 的物理存储设备』
- 第237页的『使用 **cicssfscreate** 定义 **SFS**』
- 第238页的『设置 **SFS** 的联接文件』
- 第239页的『使用 **cicssfs** 启动 **SFS**』

1. 登录一个有超级用户（root）特权并如 第3页的『建立 CICS 系统管理员』中所描述而设置的用户标识符。
2. 作为单元管理员（dce_login cell_admin）登录 DCE。
3. 确保 区域定义（RD） **DefaultFileServer** 属性设置为用于该区域的队列和文件数据的 SFS 的名称。例如，如果对于该配置已正使用准备一个命名为 `/./cics/sfs/hostABCD` 的 SFS 的过程，则检查是否 **DefaultFileServer=/./cics/sfs/hostABCD**。
4. 使用 **cicssfscnf** 命令来把队列添加至区域。例如：

```
cicssfscnf wc regionA
```

对于有关使用该命令的细节，查看在 *CICS 管理参考大全* 中的“cicssfscnf - 配置 CICS SFS”。

有关安全性的注释： CICS 有保护从未授权的用户来的对队列的存取的设施（如 第218页的『授权用户访问 CICS 资源』中所述）。DCE 可用来增强该安全性。参阅 第210页的『DCE RPC 和 ACL 如何维护系统的安全』。

使用本地命令设置队列

每个区域需要把七个文件添加至 SFS。五个文件是需要用于 CICS 临时存储器队列和瞬时数据队列管理设施的。两个文件是需要用于发送 EXEC CICS START 请求至其它系统的控制的。

使用该过程来添加文件。

在使用该过程以前，必须为 CICS 的使用而设置并启动 SFS 并且必须已配置好区域。

1. 从在其上有区域的主机，登录一个有超级用户（root）特权并如 第3页的『建立 CICS 系统管理员』中所描述而设置的用户标识符。
2. 使用下列命令来收集有关文件的信息。

```
cicsget -c rd -r regionName
```

以下列表通过信息类别描述了，包含适当的信息的属性以及当创建文件时在哪儿适合使用 **sfsadmin** 命令选项：

- 该区域的 SFS 的名称必须以 **DefaultFileServer** 属性设置。
- 未到期启动（非受保护的）：
 - LocalQFile（文件的名称）
 - LocalQIndex（主索引名称）
 - LocalQMaxRecs（**-recordlimit** 选项）
 - LocalQPrePages（**-preallocate** 选项）
 - LocalQProtection（保护级别）
 - LocalQVol（逻辑卷名称）
- 未到期的启动（受保护的）：
 - LocalQProtectFile（文件的名称）
 - LocalQProtectIndex（主索引名称）

LocalQProtectMaxRecs (**-recordlimit** 选项)

LocalQProtectPrePages (**-preallocate** 选项)

LocalQProtectProtection (保护级别)

LocalQProtectVol (逻辑卷名称)

- 逻辑可恢复瞬时数据队列:

LocalTDQFile (文件的名称)

LocalTDQIndex (主索引名称)

LocalTDQMaxRecs (**-recordlimit** 选项)

LocalTDQPrePages (**-preallocate** 选项)

LocalTDQProtection (保护级别)

LocalTDQVol (逻辑卷名称)

- 物理可恢复瞬时数据队列:

PhysicalTDQFile (文件的名称)

PhysicalTDQIndex (主索引名称)

PhysicalTDQMaxRecs (**-recordlimit** 选项)

PhysicalTDQPrePages (**-preallocate** 选项)

PhysicalTDQProtection (保护级别)

PhysicalTDQVol (逻辑卷名称)

- 不可恢复瞬时数据队列:

NonRecTDQFile (文件的名称)

NonRecTDQIndex (主索引名称)

NonRecTDQMaxRecs (**-recordlimit** 选项)

NonRecTDQPrePages (**-preallocate** 选项)

NonRecTDQProtection (保护级别)

NonRecTDQVol (逻辑卷名称)

- 不可恢复临时存储器队列:

NonRecTSQFile (文件的名称)

NonRecTSQIndex (主索引名称)

NonRecTSQMaxRecs (**-recordlimit** 选项)

NonRecTSQPrePages (**-preallocate** 选项)

NonRecTSQProtection (保护级别)

NonRecTSQVol (逻辑卷名称)

- 可恢复临时存储器队列:

RecTSQFile (文件的名称)

RecTSQIndex (主索引名称)

RecTSQMaxRecs (**-recordlimit** 选项)

NonRecTSQPrePages (**-preallocate** 选项)

RecTSQProtection (保护级别)

RecTSQVol (逻辑卷名称)

3. 使用以下步骤来把文件添加至 SFS:

- a. 从在其上有 SFS 的主机，登录一个有超级用户（root）特权并如 *Encina 管理指南 卷 2: 服务器管理* 书中所描述而设置的用户标识符。
- b. 如果在一个 DCE 单元上配置 SFS，那么作为单元管理员（dce_login cell_admin）登录 DCE。
- c. 设置下列环境变量：

```
ENCINA_TK_SERVER 为 ./cics/sfs/serverId
ENCINA_SFS_SERVER 为 ./cics/sfs/serverId
ENCINA_CDS_ROOT 为 ./cics
```

- d. 使用 **sfsadmin** 命令如下所示创建文件：

- 未到期启动（非受保护的）：

```
sfsadmin create clusteredfile regionNamecicsnlqfile \
  2 key byteArray 8 data varLenByteArray 32768 \
  cicsnlqidx -unique 1 key sfs_shortName \
  -preallocate 10 -recordlimit 1000000
```

- 未到期的启动（受保护的）：

```
sfsadmin create clusteredfile regionNamecicsplqfile \
  2 key byteArray 8 data varLenByteArray 32768 \
  cicsplqidx -unique 1 key sfs_shortName \
  -preallocate 10 -recordlimit 1000000
```

- 逻辑可恢复瞬时数据队列：

```
sfsadmin create clusteredfile regionNamecicstdqlgfile \
  2 itqkey byteArray 12 itqdata varLenByteArray 32768 \
  cicstdqlgidx -unique 1 itqkey sfs_shortName \
  -preallocate 10 -recordlimit 1000000
```

- 物理可恢复瞬时数据队列：

```
sfsadmin create clusteredfile regionNamecicstdqphfile \
  2 itqkey byteArray 12 itqdata varLenByteArray 32768 \
  cicstdqphidx -unique 1 itqkey sfs_shortName \
  -preallocate 10 -recordlimit 1000000
```

- 不可恢复瞬时数据队列：

```
sfsadmin create clusteredfile regionNamecicstdqnofile \
  2 itqkey byteArray 12 itqdata varLenByteArray 32768 \
  cicstdqnoidx -unique 1 itqkey sfs_shortName \
  -preallocate 10 -recordlimit 1000000
```

- 不可恢复临时存储器队列：

```
sfsadmin create clusteredfile regionNamecicsnrectsqfil \
  2 tsqkey byteArray 11 tsqdata varLenByteArray 32768 \
  cicsnrectsqidx -unique 1 tsqkey sfs_shortName \
  -preallocate 10 -recordlimit 1000000
```

- 可恢复临时存储器队列：

```
sfsadmin create clusteredfile regionNamecicsrectsqfile \
  2 tsqkey byteArray 11 tsqdata varLenByteArray 32768 \
  cicsrectsqidx -unique 1 tsqkey sfs_shortName \
  -preallocate 650 -recordlimit 1000000
```

4. 对于其保护级别设置为 **none** 的任何文件，使用下列命令来创建 ACL：

```
acl_edit ./cics/sfs/serverId/fileName
sec_acl_edit> m unauthenticated:ADIQRUE
sec_acl_edit> m any_other:ADIQRUE
sec_acl_edit> exit
```

对于其保护级别大于 **none** 的任何文件，使用下列命令以创建 ACL：

```

acl_edit ./cics/sfs/serverId/fileName
sec_acl_edit> m user:cell_admin:ADIQRUE
sec_acl_edit> m user:cics/regionName:ADIQRUE
sec_acl_edit> m group:cics_admin:ADIQRUE
sec_acl_edit> exit

```

5. 在带有区域的主机上，使用下列命令来扩充 区域定义（RD）**%S** 符号为 SFS 的短名称:

```

cicsupdate -c rd -r regionName DefaultFileServer="serverId" \
LocalQVol="sfs_shortName" \
LocalQProtectProtectVol="sfs_shortName" \
LocalTDQVol="sfs_shortName" \
PhysicalTDQVol="sfs_shortName" \
NonRecTDQVol="sfs_shortName" \
NonRecTSQVol="sfs_shortName" \
RecTSQVol="sfs_shortName"

```

为队列和文件管理设置 DB2

该过程描述了如何设置 DB2 数据库，这样可为 CICS 文件管理配置它。

在使用该过程以前

- 您首先检查在 CICS CD 上 README 文件中发行本所特有的信息是很重要的，因为该文件包含了对以下过程的更改。
- 您同样应该有下列 DB2 文档：
 - DB2: 命令说明
 - DB2: 管理指南
 - DB2: 安装指南
- DB2 必须在 CICS 支持的层次上安装。
- 如果当 DB2 安装时没有运行 **db2ln** 脚本，则现在运行它。参阅 DB2: 安装指南。
- 必须设置并启动 DB2 实例。参阅 DB2: 管理指南。

注:

1. 如果 DB2 返回在运行时的一个 XA 资源管理器错误（它可能意味着在配置中有错误），则 CICS 把错误添加至 CICS console.nnnnnn 文件中。
2. DB2 提供了两种类型表格空间的选择：
 - 系统管理空间（SMS）表格空间，其中使用操作系统文件管理器调用来控制存储空间。
 - 数据库管理空间（DMS）表格空间，其中数据库管理器控制存储空间。

缺省地，DB2 使用 SMS 表格空间。不过，推荐把 DMS 表格空间与 CICS 一起使用，一般它可提供性能的改进。

对于有关如何创建 DMS 表格空格，参阅 DB2: 管理指南。对于有关 CICS 和数据库性能的更多信息，参阅 第185页的『改进访问数据库的性能』。
3. 如果希望指定一个操作的锁定超时值，或要重新配置 DB2 队列和文件管理的死锁超时间隔，那么这通过设置 LOCKTIMEOUT 和 DLCHKTIME 参数归档于数据库层。缺省地，不设置操作的锁定超时值。
4. 数据库实例所有者必须在 CICS 组中。

5. 因为所有队列被处理为可恢复的，所以使用 DB2 文件控制增加了死锁的风险。要帮助避免死锁，定义事务定义属性 DeadLockTimeout。缺省值为 0，它意味着没有超时。

尝试使用 DB2 变量 DB2_RR_TO_RS 来关掉下一关键锁定。这可能避免不需要的死锁，但如果使用该变量，DB2 将获取更多内部的锁定。在该情况下，locklist 或 maxlocks 数据库参数可能需要增加以容纳额外的内部锁定。如果 locklist 不够，将发生锁定逐步增加并且并发性将受损。把 DB2_RR_TO_RS 设置为 ON，应用程序可能不能使用可重复读的光标也是一个限制。CICS DB2 文件控制不使用可重复读光标。

6. DB2 允许从以用户 cics 隐含认证的或以另一些用户显式认证的 CICS 来的连接。隐式或显式的认证的选择取决于您的 DB2 环境。例如，如果 CICS 和 DB2 服务器安装在相同的机器上，则以用户 cics 的隐式认证易于被支持。

一个认证的 DB2 用户在 DB2 数据库中可以有不同的认证级别。例如，在 DB2 下，用户 cics 可能配置为有系统管理 (SYSADM) 或数据库管理 (DBADM) 权限以及任何数量的特权、权限和授权。

如果有多个用户要对外部文件句柄有存取权，那么他们需要给予对 /usr/lpp/cics/etc 目录中的下列文件的执行存取：

- cics101e
- cics101f
- cics101j
- cics101l
- cics101s
- cics101u

使用下列命令来授予该存取：

```
db2 grant execute on <package> buildadm.<package name> to <userid>
```

为了使用用于队列和文件管理的 DB2，在 CICS 连接至 DB2 的用户标识符应该至少有对适当的数据库的 CONNECT 特权。

对于更多信息，参阅 *DB2: 管理指南*。

注：*Privileges* 使得用户能创建或存取数据库资源。*Authority* 级别提供了一个分组特权和高级别数据库管理器维护以及实用程序操作的方式。这些操作共同进行来控制对数据库管理器和它的数据库对象的存取。用户仅可以存取他们有适当的 *authorization* 的对象，也就是说他们需要特权或权限。

1. 登录一个有超级用户 (root) 特权并如 第3页的『建立 CICS 系统管理员』中所描述而设置的用户标识符。
2. 使用下列命令来创建在运行时间由 CICS COBOL 运行时间、CICS 事务以及 DB2 交换装入文件引用的 DB2 共享对象：

```
cd /usr/lpp/cics/DB2Version/lib
ar -vx libdb2.a
mv shr.o db2.o
```

其中 *DB2Version* 反映您正在使用的 DB2 发行本

3. 使用下列命令创建从 /usr 至 db2.o 的符号链接：

```
ln -s /usr/lpp/cics/db2_02_01/lib/db2.o /usr/lib/db2.o
```


4. 使用 **db2 CREATE DATABASE *database*** 命令来创建用于 CICS 的 DB2 数据库。（如果您要使用一个现存的数据库，该步骤可跳过。）

如果您正在使用 SMS 表格空间，则使用以下命令：

```
db2 create database database
```

如果您正在使用 DMS 表格空间，那么：

- a. 创建用于表格空间的 逻辑卷。
- b. 当创建数据库时，使用 **tablespace** 选项。

例如，以下命令创建一个带有 10 个逻辑分区的逻辑卷以及一个带有 10000 页面的表格空间：

```
mk1v -y logvolABC rootvg 10
db2start
db2 create database CHARLEY user tablespace \
    managed by database using (device '/dev/rlogvolABC' 10000)
```

注：在使用 **db2start** 启动数据库管理器以前，推荐设置 DB2_RR_TO_RS 环境变量为 ON。这改变了 DB2 索引管理器的行为来避免下一个关键锁定，它可能对使用用于队列和文件管理的 DB2 的 CICS 区域有一些性能增进。

不过，设置 DB2_RR_TO_RS 为 ON 需要您在任何使用该 db2 实例的 DB2 程序中避免使用可重复读的光标。使用用于队列和文件管理的 DB2 的区域不使用这种类型的光标。因为该限制，建议仅对于为 CICS 所专门使用的实例，设置该环境变量。

5. 输入命令 db2 来访问 DB2 命令行处理器。使用 **CONNECT TO *database*** 命令连接至数据库。
6. 授予在新的数据库上的适当的特权。例如，如果您计划隐式允许 CICS 以用户 cics 连接至数据库，并且如果用户 cics 在数据库中没有 SYSADM 或 DBADM 权限，那么使用 **GRANT CONNECT ON DATABASE TO USER CICS** 命令来授予适当的特权。输入 **quit** 来退出 DB2 命令行处理器。
7. 在该点上，您可能希望配置记录中所提到的 LOCKTIMEOUT 和 DLCHKTIME 参数。同样，推荐您设置 DB2 数据库管理器 TP_MON_NAME 配置参数为 **CICS**。对于更多有关配置参数的信息，参阅 *DB2: 管理指南*。
8. 运行 **cicsdefaultservers** 以设置包含用于定义数据文件至 DB2 的缺省模式的文件系统。参阅 第45页的『使用 cicsdb2import 建立 DB2 上的数据文件』来发现如何使用图表。

对于有关命令行处理器（CLP）**db2** 命令，启动 DB2（**db2start**）命令，**create database** 命令，以及 **grant** 命令的信息，参阅 *DB2: 命令说明*。

用 cicsdb2conf 配置资源和 DB2

该过程的目的是配置资源定义并把表格添加至将用于 CICS 队列和文件管理的 DB2 数据库中。该过程使用 **cicsdb2conf** 命令。

注意:

该过程创建把该数据库定义至区域的 产品定义 (XAD)。若已经有为该数据库定义的 XAD, 则使用 **cicsdelete** 命令或当使用 **cicsdb2conf** 命令时使用 **-s** 选项。否则, 该数据库可能有两个 XAD 项, 产生不可预测的结果。

注: 如果 DB2 返回一个在运行时的 XA 资源管理器错误 (它可能意味着在配置中有错误), 则 CICS 把错误添加至 `/var/cics_regions/regionName/console.nnnnnn` 中。

以下过程必须在该过程可做以前进行:

- 必须配置区域。
- 第243页的『为队列和文件管理设置 DB2』。
- 若正在使用显式的 DB2 认证将 CICS 连接到 DB2:
 - 必须建立用户, 并授予最小许可权。
 - 若不希望提示口令, 则将 `CICSDB2PASSWD` 环境变量设置为数据库的用户标识符口令。
- 1. 如 *CICS 管理参考大全* 中 “cicsdb2conf - 配置 DB2” 所述使用 **cicsdb2conf** 命令。例如, 下列命令:

```
cicsdb2conf -r regionA -C -i INSTANCE -a ALIAS -u DB2USER
```

- a. 设置区域的 区域定义 (RD) 中的 **RDBMSInstance** 属性为 **INSTANCE** 以及 **DefaultFileServer** 属性为 **ALIAS**。
- b. 创建一个指定 CICS 与 DB2 的这个实例使用两阶段提交通过 XA-接口进行相互间操作的 XAD 项。要使用单阶段提交优化, 使用 **cicsdb2conf -n** 选项。
- c. 设置 XAD **SwitchLoadFile** 为 **cicsxadb2**, 它是缺省的交换装入文件。

注: 如果正在使用您自己的交换装入文件, 使用 **cicsdb2conf -s** 选项来抑制系统定义的 XAD 项的创建。然后使用 **cicsadd** 来创建该数据库的 XAD (如果还没有一个存在的话) 并设置 **SwitchLoadFile** 属性为您的交换装入文件的名称。

- d. 设置 XAD **XAOOpen** 属性为 **ALIAS,DB2USER**。
- e. 在为区域定义的 文件定义 (FD) 中设置所有项的 **FileServer** 属性为空。
- f. 调用 DB2 认证服务来验证是否允许用户 **DB2USER** 存取该数据库。如果 `CICSDB2PASSWD` 设置为 **DB2USER** 的正确口令, 就认证 **DB2USER**。如果 `CICSDB2PASSWD` 没有设置, 则提示要口令。
- g. 在 DB2 数据库中为下列每个事物创建一个表格:
 - 可恢复辅助临时存储器队列。表名为 `%Rreqtsq#`。
 - 不可恢复辅助临时存储器队列。表名为 `%Rnreqtsq#`。
 - 逻辑可恢复瞬时数据队列。表名为 `%Rlogtdq#`。
 - 物理可恢复瞬时数据队列表名为 `%Rphtdq#`。
 - 不可恢复瞬时数据队列。表名为 `%Rnreqtdq#`。
 - 本地排队受保护的 STARTS。表名为 `%Rlqprot#`。

- 本地排队非受保护的 STARTS。表名为 %Rlque#。

由于 DB2 限制，所有队列为可恢复和受保护的。

从命令行启动区域

要启动 CICS 区域，必须注册为一个有超级用户（root）特权的用户标识符。

有关冷启动区域的警告：如果冷启动区域，将丢弃下列信息：

- 包含非固定资源定义的运行时间数据库。
- 任何进行中的事务状态，临时存储器队列，事务数据队列，或当区域关闭时没有清除的 START 请求。这些信息在正常关闭期间被清除。

对于更多信息，参阅 第124页的『冷启动和热启动』。

注：该信息不覆盖 AIX **startsrc** 命令的全部的功能，并仅关心 CICS 启动时它的使用。

对于该命令的完整描述，请阅读有关 AIX 文档。该命令包含于 AIX 基本操作系统（BOS）的运行时间中。

您可使用 **startsrc** 设施来启动一 CICS 区域，虽然 CICS 控制程序是更可取的方式（参阅 第125页的『对于启动使用 cicscp』）。一旦 CICS 成功地启动并运行站初试化程序，CICS 就允许进行事务处理。如果设置了环境变量 CICSTRACE，则 CICS 自动启动产品跟踪。

如下所示使用 **startsrc -s** 选项来指定区域的名称：

```
startsrc -s cics.regionName
```

其中 *regionName* 是区域的名称。这个命令使用在 区域定义（RD）中指定的属性值来启动区域。可用 **startsrc -a** 选项更改在启动时使用的 RD 属性值。语法为：

```
startsrc -s cics.regionName -a "attribute=value"
```

如果要更改多于一个属性值，那么以空格分开属性值，如下所示：

```
startsrc -s cics.regionName
-a "attribute=value attribute=value"
```

例如，下列命令可用来显式指定 regionA 的冷启动：

```
startsrc -s cics.regionA -a 『StartType=cold』
```

可通过指定不同的区域名称以拥有多个 CICS 实例。CICS 阻止相同的区域的多实例的启动。

从命令行关闭区域

注：该信息不覆盖 AIX **stopsrc** 设施的全部的功能，并仅关心 CICS 关闭时该设施的使用。对于 **stopsrc** 设施的完整描述，请阅读有关 AIX 的文档编制。**stopsrc** 设施为 AIX 基本操作系统（BOS）运行时间中的一部分。

1. 登录一个有超级用户（root）特权并如 第3页的『建立 CICS 系统管理员』中所描述而设置的用户标识符。

2. 如果正在一个 DCE 单元上工作，那么作为单元管理员（dce_login cell_admin）登录 DCE。
3. 对于正常关闭使用下列命令：

```
stopsrc -s cics.regionName
```

其中 *regionName* 是区域的名称。正常的关闭完成所有当前进程并在关闭区域以前释放资源。区域不接受新的工作请求。

如果需要，您可使用 **-c**（取消）或 **-f**（强制）选项来关闭区域。

当指定了 **-c** 时，**stopsrc** 发送一个 SIGTERM 请求至区域并尝试停止所有进程并在关闭区域以前释放所有资源。区域释放所有资源并关闭。如果区域不在某些宽限期（在区域的子系统对象类中指定）内关闭，**stopsrc** 发送一个 SIGKILL 信号以关闭区域。当这发生时，一些由区域分配的资源可能不释放。如果是该情况，您可运行 **cicsnotify** 命令来释放这些资源，并删除 *region_lock* 文件。

对于该类型的关闭使用下列命令：

```
stopsrc -s cics.regionName -c
```

当指定了 **-f** 时，**stopsrc** 在关闭区域以前释放资源但不等待进程停止。当带有 **-c** 时，一些由区域分配的资源可能不释放。如果是该情况，您可运行 **cicsnotify** 命令来释放这些资源，并删除 *region_lock* 文件。

对于该类型的关闭使用下列命令：

```
stopsrc -s cics.regionName -f
```

关闭信息被发送至 */var/cics_regions/regionName/console.nnnnnn* 中。

正常关闭可能要较长时间来完成。可让用户结束较长运行任务，或使用 CEMT 清除任务，并允许关闭正常完成。

使用 **cicssfsshut** 关闭 SFS

用该过程来使用 **cicssfsshut** 关闭 SFS。

注意：

不要关闭 SFS 直到使用 SFS 的所有区域已首先被关闭。

1. 登录一个有超级用户（root）特权并如第3页的『建立 CICS 系统管理员』中所描述而设置的用户标识符。
2. 如果正在一个 DCE 单元上工作，那么作为单元管理员（dce_login cell_admin）登录 DCE。
3. 使用下列命令：输入：

```
cicssfsshut ./cics/sfs/serverId
```

其中 *serverId* 是该服务器的基名。

对于有关使用该命令的更多信息，参阅在 *CICS 管理参考大全* 中的“**cicssfsshut** - 关闭 CICS SFS”。

使用 **cicsdestroy** 删除区域

如果对于区域要从固定数据库中删除一个区域的资源定义，那么使用 **cicsdestroy** 来删除与区域相关的所有资源定义。例如：

```
cicsdestroy -r regionA
```

删除除了那些为 SFS 定义的以外的所有资源定义。

您也可删除区域使用的 SFS 文件。对于有关这个的更多信息，参阅在 *CICS 管理参考大全* 中的“**cicsdestroy** - 破坏 CICS 区域”中 **-s** 选项的描述。

使用 **cicssfsdestroy** 删除 SFS

用该过程来使用 **cicssfsdestroy** 删除 SFS

本过程不删除为 SFS 创建的操作系统用户标识符。

1. 登录一个有超级用户（root）特权并如 第3页的『建立 CICS 系统管理员』中所描述而设置的用户标识符。
2. 如果正在一个 DCE 单元上工作，那么作为单元管理员（**dce_login cell_admin**）登录 DCE。
3. 使用下列命令来停止 SFS:

```
cicssfsshut /./cics/sfs/serverId
```

4. 使用下列命令来消除 SFS:

```
cicssfsdestroy /./cics/sfs/serverId
```

该命令:

- 从 CICS 服务器数据库删除 SFS 定义
- 删除操作系统机制中用于启动和停止较长运行进程的项目
- 删除 SFS 的 DCE 委托人、其 DCE 帐户和其 DCE 密钥表项（若存在）。

5. 删除 SFS 目录:

```
/var/cics_servers/SSD/cics/sfs/serverId
```

6. 若正在 DCE 单元中工作，则可通过再次注册入 DCE 或使用 DCE 命令 **kinit** 来对于任何已经访问服务器的 **dce_login** 上下文重新认证 DCE 票券。这将删除旧服务器的任何高速缓存的票券，并避免产生问题（若用与刚消除的那个相同的名称创建新 SFS）。

对于有关使用该命令的更多信息，参阅在 *CICS 管理参考大全* 中的“**cicssfsdestroy** - 破坏 CICS SFS”。

附录C. 使用 DCE

表 46. 重新配置 DCE 的导向图

若需...	参考...
设置 CICS 的 DCE。	第251页的『DCE 先决条件』
在现存 DCE 单元中添加主机。如果没有已配置过的 DCE 客户或者如果要更改 DCE 客户配置，则使用该过程。	第252页的『将主机添加到现存的 DCE 单元中』。
为 CICS 客户设置 DCE 请求。如果要配置或重新配置 CICS 客户，则可使用该过程，但不要更改 DCE 配置。	第254页的『设置 CICS 客户的 DCE 请求』
为仅 RPC 设置配置 DCE 客户。	第255页的『为仅 RPC 设置配置 DCE 客户』。
作为 DCE 单元管理员注册到 DCE 。	第253页的『将自身认证为 DCE 单元管理员』。
将与 CICS 有关的组和对象添加到单元中。在单元中只需要做一次，并且它被包含在大多数 DCE 安装过程中。本过程只在（出于某些原因）删除与 CICS 相关的 DCE 组和目标时才需要。	第254页的『添加与 CICS 有关的组和对象』。
消除使用 cicscp 的 DCE 配置	第255页的『消除使用 cicscp 的 DCE 配置』。
使用 DCE 命令取消配置 DCE 客户。	第255页的『从 DCE 客户主机中删除 DCE 客户』。
使用 DCE 单元取消配置 DCE 客户。	第256页的『删除单元的 DCE 配置』。
从 DCE 单元移植到仅 RPC 的设置。	第257页的『将 CICS 从 DCE 单元移植到仅 RPC 设置』。
从仅 RPC 的设置移植到 DCE 单元。	第265页的『将 CICS 从仅 RPC 设置移植到 DCE 单元』。

DCE 先决条件

在配置 CICS 区域、客户或 Encina 服务器前必须先设置 DCE 。DCE 请求是：

- 在 DCE 单元中：
 - DCE 单元必须与 DCE 安全服务器和 DCE 单元目录服务器一同设置。
 - RPC 端点映像程序（精灵程序），安全性职员和 CDS 职员必须在单元的每个主机上都配置。
 - 与 CICS 相关的 DCE 组和对象必须添加到单元中。DCE 组是：

cics_admin
cics_sfs
cics_ppcgwy
cics_regions
cics_users

与 CICS 相关的 DCE 对象是：

././cics/sfs
././cics/ppc
././cics/ppc/gateway
././cics/trpc
././cics/tmxa
././cics/profile

使用下列命令在单元中列出对象:

```
cdscp list dir ././cics/*
```

列表应该包括:

```
sfs  
ppc  
trpc  
tmxa  
profile
```

使用本机 DCE 命令列出组。下列示例使用了 **rgy_edit**:

```
rgy_edit  
  
domain group  
  
view
```

显示出的列表应该包含与 CICS 相关的 DCE 组。

- 在仅 RPC 设置中
 - RPC 端点映像程序（精灵程序）必须在每个具有 CICS 区域、客户或 Encina 服务器（它们之间相互通信）的主机上设置。
 - Encina 服务器的主机需要在 `/var/cics_servers/server_bindings` 文件中的端点信息，因此区域可以联接到它们。
 - CICS 区域和客户的主机要求 CICS_HOSTS 环境文件设置到区域和客户与之通信的主机列表中，如具有区域或 Encina 服务器的主机。

相关信息

第72页的『为与 CICS 客户机使用建立 DCE』。

同样参阅 AIX 文档以得到关于 DCE 的信息。

DCE 设置过程

本节包含 DCE 设置的过程。按照第251页的表 46 中的指导使用这些过程。

将主机添加到现存的 DCE 单元中

使用该过程将主机添加到现存 DCE 单元中。该过程假设:

- 设置了 DCE 单元。
- 已安装了 DCE 客户软件（参阅 *快速入门* 中的“IBM TXSeries 概述”）。
- 安装了 CICS。

注意:

该过程消除了本地主机上的任何 DCE 配置。

在使用该过程前，先验证本地主机的日期和时间是否与 DCE 安全性服务器 主机的相同。可以使用下列几种方法之一:

- 如果在单元中配置了分布式时间服务（DTS），则包括 DCE 客户配置中的 DTS 客户。这是推荐的方式。

如果配置了分布式时间服务（DTS），则 **cicscp** 命令自动配置 DTS 客户。

- 使用其它一些时间同步化服务。
- 如果没有使用时间同步化服务，则手工设置本地主机的日期和时间为 DCE 安全性服务器主机的日期和时间。

注意：

如果在单元主机间发生时间漂移，则 **DCE** 失败。因此，不推荐手工将日期和时间同步。

1. 需要知道单元的名称，DCE CDS 所在主机及 DCE 安全服务器所在主机的名称。使用 **getcellname** 命令从一已在单元中的主机中找出单元的名称。向 DCE 管理员询问 DCE 服务器主机的名称。
2. 注册到一个不拥有超级用户特权的用户标识符，该用户标识符如 第3页的『建立 CICS 系统管理员』中描述的那样设置。
3. 按照下列步骤在本地主机上删除任何现存的 DCE 配置：

第255页的『消除使用 cicscp 的 DCE 配置』。

第255页的『从 DCE 客户主机中删除 DCE 客户』。

第256页的『删除单元的 DCE 配置』（如果在本地主机上有 DCE 服务器）。

4. 如果 DCE CDS 和 DCE 安全服务器都在相同的主机上。使用这个命令：

```
cicscp -v -l logFile create dce -n cellName [-s serverHost]
```

这里 *cellName* 是单元的名称，而 *serverHost* 运行 DCE CDS 和 DCE 安全服务器机器的主机名。

如果 DCE CDS 和 DCE 安全服务器不在相同主机上：

- a. 使用这个命令：

```
mkdce -n cellName -s securityServerHost -c CDSHost sec_cl cds_cl
```

- b. 与 CICS 有关的 DCE 组 and 用户还没有添加到 DCE CDS 中，则使用下列命令：

```
cicssetupdce
```

注：在 DCE 服务器或 DCE 客户使用 **cicscp** 进行配置时，添加与 CICS 有关的 DCE 组 and 用户。

要得到在此过程中使用的 CICS 命令详细的信息，参阅 *CICS 管理参考大全* 中的“cicscp - DCE 命令”和“cicssetupdce - 在一个 DCE 单元创建目录和组”。

将自身认证为 DCE 单元管理员

在设置 DCE 满足 CICS 需求时，必须先认证为 DCE 委托人 *cell_admin*。

要将自身认证为 DCE 委托人 *cell_admin*：

1. 输入 **dce_login cell_admin**。
2. 输入 DCE 委托人 *cell_admin* 的口令。

如果没有更改缺省的口令，则显示『Password must be changed!』。

使用 **klist** 命令校验是否已认证为 DCE 委托人。由『**Default Principal:**』开始的行显示了当前委托人 **cell_admin** 的名称。

设置 CICS 客户的 DCE 请求

使用该过程将对象添加到 CICS 客户的单元中，并创建 CICS 客户密钥表文件。如果没有在本地主机上配置 DCE 客户，则第252页的『将主机添加到现存的 DCE 单元中』。

该过程假设：

- 设置了 DCE 单元。参阅第251页的『附录C. 使用 DCE』。
 - 设置了 DCE 客户。参阅第252页的『将主机添加到现存的 DCE 单元中』。
 - 安装了 CICS。
1. 注册到一个不拥有超级用户特权的用户标识符，该用户标识符如第3页的『建立 CICS 系统管理员』中描述的那样设置。
 2. 验证了正在运行的 DCE 服务器。
 3. 作为单元管理员（**dce_login cell_admin**）注册到 DCE 中。
 4. 使用这个命令：

cicssetupclients

5. 如果这是第一个为单元中 CICS 配置的数据库，则用以下命令将与 CICS 有关的 DCE 组和对象添加到单元中：

cicssetupdce

要得到此过程使用的命令的详细信息，参阅 *CICS 管理参考大全* 中的“**cicssetupdce** - 在一个 DCE 单元创建目录和组”和“**cicssetupclients** - 为 CICS 主机配置 DCE”。

添加与 CICS 有关的组和对象

使用该过程将与 CICS 有关的组和对象添加到单元中。

注：在 DCE 服务器或 DCE 客户使用 **cicscp** 进行配置时，添加与 CICS 有关的 DCE 组和用户。因此，该过程仅在如果，出于某些原因，从单元中删除了与 CICS 有关的 DCE 组和用户时才需要使用。

该过程假设：

- 设置了 DCE 单元。参阅第251页的『附录C. 使用 DCE』。
 - 设置了 DCE 客户。参阅第252页的『将主机添加到现存的 DCE 单元中』。
 - 安装了 CICS。
1. 注册到一个不拥有超级用户特权的用户标识符，该用户标识符如第3页的『建立 CICS 系统管理员』中描述的那样设置。
 2. 验证了正在运行的 DCE 服务器。
 3. 作为单元管理员（**dce_login cell_admin**）注册到 DCE 中。
 4. 使用这个命令：

cicssetupdce

需要更多信息，请参阅 *CICS 管理参考大全* 中的“cicssetupdce - 在一个 DCE 单元创建目录和组”。

为仅 RPC 设置配置 DCE 客户

使用该过程配置只在仅 RPC 设置中使用的端点映像程序。该过程假设：

- 已安装了 DCE 客户软件（参阅 *快速入门* 中的“IBM TXSeries 概述”）。
- 安装了 CICS。

注意：

该过程消除了本地主机上的任何 DCE 配置。

1. 注册到一个不拥有超级用户特权的用户标识符，该用户标识符如 第3页的『建立 CICS 系统管理员』中描述的那样设置。
2. 使用 **rmddce all** 命令删除任何现存的 DCE 配置。
3. 使用这个命令：

```
cicscp -v -l logFile create dce -R
```

需要更多信息，请参阅 *CICS 管理参考大全* 中的“cicscp - DCE 命令”。

消除使用 cicscp 的 DCE 配置

使用该过程消除使用 **cicscp** 的 DCE 配置。

如果正在更改 DCE 配置，遵照 第257页的『将 CICS 从 DCE 单元移植到仅 RPC 设置』或 第265页的『将 CICS 从仅 RPC 设置移植到 DCE 单元』。

cics destroy dce 命令停止了 DCE 并消除了机器的 DCE 配置。如果在客户机上运行此命令且 DCE CDS 是可用的，则从单元中删除与 CICS 有关的 DCE 组和对象。如果 DCE CDS 不可用，则只删除本地配置。

注意：

在使用该过程前，关闭任何在本地主机上的区域或 **SFS**。否则，如果没有配置过 DCE，将不得不迫使区域或 **SFS** 关闭，则在启动系统时可能需要一次冷启动。

1. 注册到一个不拥有超级用户特权的用户标识符，该用户标识符如 第3页的『建立 CICS 系统管理员』中描述的那样设置。
2. 作为单元管理员（dce_login cell_admin）注册到 DCE 中。
3. 使用下列命令：

```
cicscp destroy dce
```

参阅 *CICS 管理参考大全* 中的“cicscp - DCE 命令”，以获得更多有关使用该命令的信息。

从 DCE 客户主机中删除 DCE 客户

从本地主机上删除 DCE 客户配置，并从单元上删除该单元的对象。如果在本地主机上配置了任何 DCE 服务器，则不要使用该步骤。要取消配置 DCE 服务器，参阅 第256页的『删除单元的 DCE 配置』。

注：如果在工作的主机上安装了 CICS，则遵照 第255页的『消除使用 cicscp 的 DCE 配置』。

1. 注册到具有超级用户权限的用户标识符中并：

- 在要取消配置 DCE 客户的主机上
- 已在适当的环境变量中设置了下列路径：

a. PATH 应该包括：

```
/usr/lpp/dce/bin  
/usr/lpp/dce/etc
```

b. LIBPATH 应该包括：

```
/usr/lpp/dce/lib
```

c. NLSPATH 应该包括：

```
/usr/lpp/dce/lib/nls/msg/%L/%N
```

如果要在没有使用 CICS AIX 版 的主机上配置 DCE，则请参见产品自身的文档。

2. 若正在 DCE 单元中工作（如果这不是仅 RPC 设置），则作为管理员注册到（dce_login cell_admin）DCE。

注：即使不能注册到 DCE 中，也可以取消配置本地主机。然而，这样就不能从单元中删除主机的对象。下列内容描述了在这种情况下该如何处理。

3. 下列指导只是针对 CICS AIX 版 的。

使用下列命令：

```
rmcdce -o full all-cl
```

该命令从单元中删除了客户对象，并从本地主机中删除了客户配置。

如果不能建立到单元的连接以删除对象，则使用以下命令：

a. 从客户主机中：

```
rmcdce -o local all-cl
```

b. 从 CDS 服务器主机中：

```
rmcdce -o admin -h clientHost all-cl
```

这里 *clientHost* 是要取消配置的 DCE 客户的主机名。

如果要在没有使用 CICS AIX 版 的主机上配置 DCE，则请参见产品自身的文档。

删除单元的 DCE 配置

使用该过程消除一个 DCE 单元，这里 CDS 服务器和安全服务器都在相同的主机上。若需其它的配置，参考您的 DCE 文档。

注：如果在具有 DCE 服务器的主机上安装了 CICS，则遵照 第255页的『消除使用 cicscp 的 DCE 配置』。

1. 停止并取消配置所有在单元中的 DCE 客户（按照 第255页的『消除使用 cicscp 的 DCE 配置』或 第255页的『从 DCE 客户主机中删除 DCE 客户』）。

2. 注册到具有超级用户权限的用户标识符中并：

- 在要取消配置 DCE 服务器的主机上

- 已在适当的环境变量中设置了下列路径:
 - a. PATH 应该包括:
`/usr/lpp/dce/bin`
`/usr/lpp/dce/etc`
 - b. LIBPATH 应该包括:
`/usr/lpp/dce/lib`
 - c. NLSPATH 应该包括:
`/usr/lpp/dce/lib/nls/msg/%L/%N`
- 3. 作为单元管理员 (dce_login cell_admin) 注册到 DCE 中。
- 4. 下列指导只是针对 CICS AIX 版的。
使用下列命令:
`rmddce -o full all`

如果要在没有使用 CICS AIX 版的主机上取消配置 DCE, 则请参见产品自身的文档。

将 CICS 从 DCE 单元移植到仅 RPC 设置

这些过程描述了如何将 DCE 单元中的区域、客户和 Encina 服务器移植到仅有 RPC 的设置中。该进程中包含两个步骤。第一步, 删除 DCE 配置。第二步, 配置仅 RPC 设置的 CICS 系统。

第一步有三个选项, 取决于要删除的 DCE 配置的数量。参考 第258页的图 5, 第259页的图 6, 和 第260页的图 7。

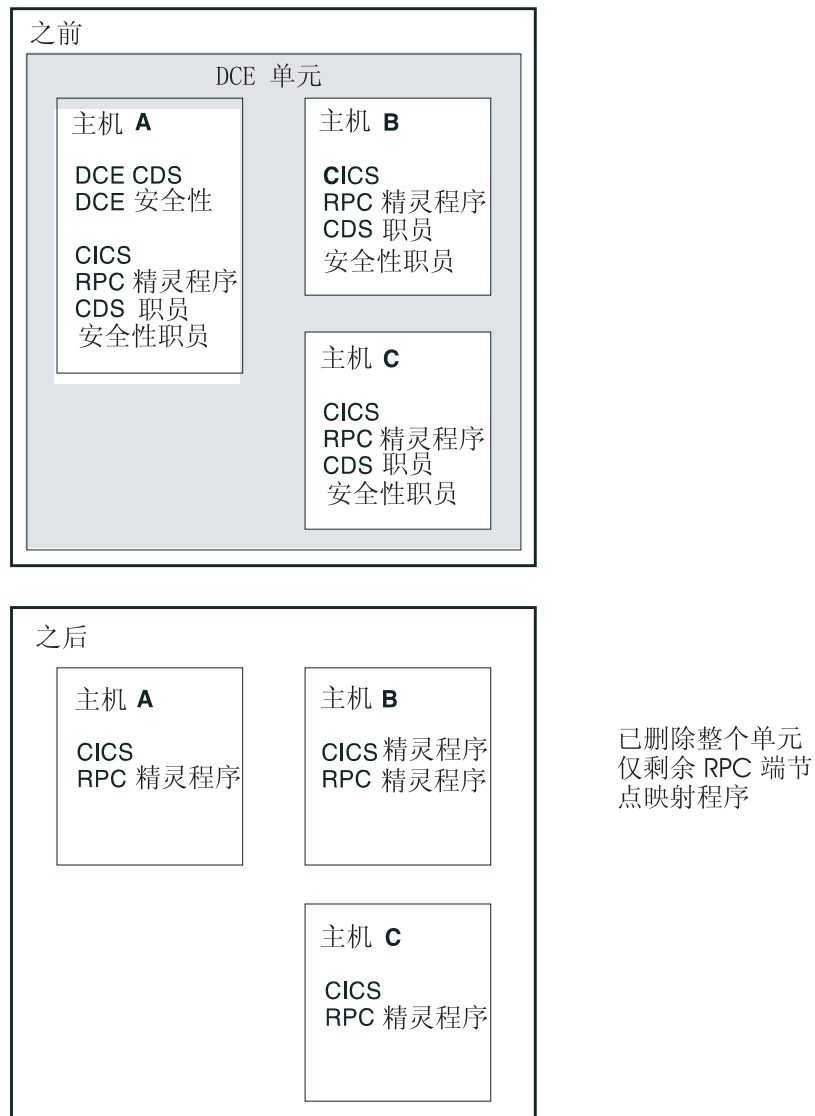
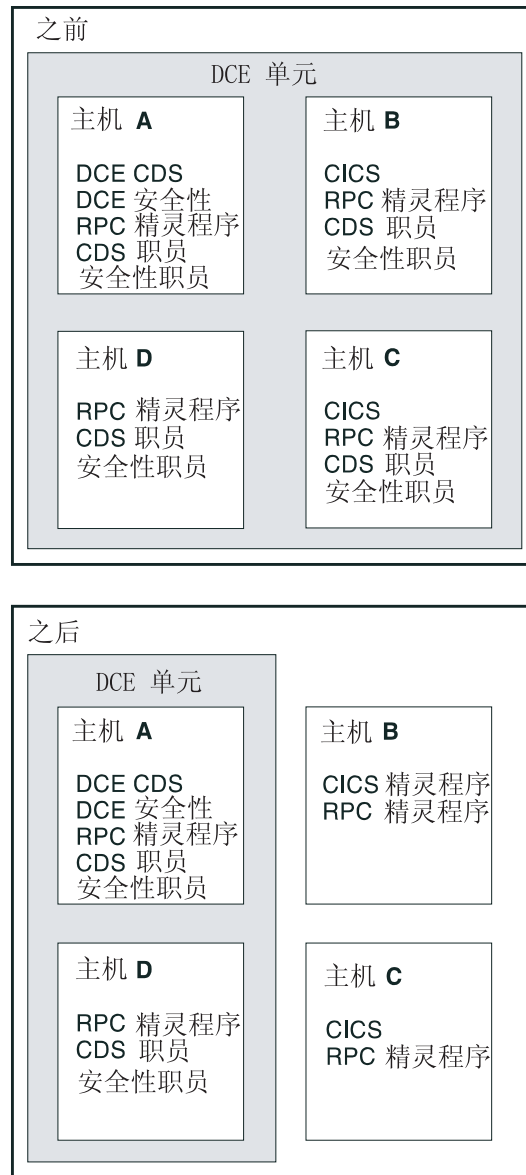


图 5. 方案 1

在该方案中，取消配置了整个 DCE 单元，包括 DCE 服务器。将 第260页的『消除单元』用于此方案。然后使用第262页的『配置仅 RPC 设置』来重新配置 CICS。



从单元内将两个 CICS 主机删除。
在本方案中，CICS 相关的 DCE
组和对象能从 CDS 中删除。

图 6. 方案 2

在该方案中，从单元中删除了两个 CICS 主机，但单元却完整无缺。因为在单元中没有 CICS 主机，从单元中删除与 CICS 有关的 DCE 组和对象。将第261页的『从单元中删除所有的 CICS 主机』用于此方案。然后使用第262页的『配置仅 RPC 设置』来重新配置 CICS。

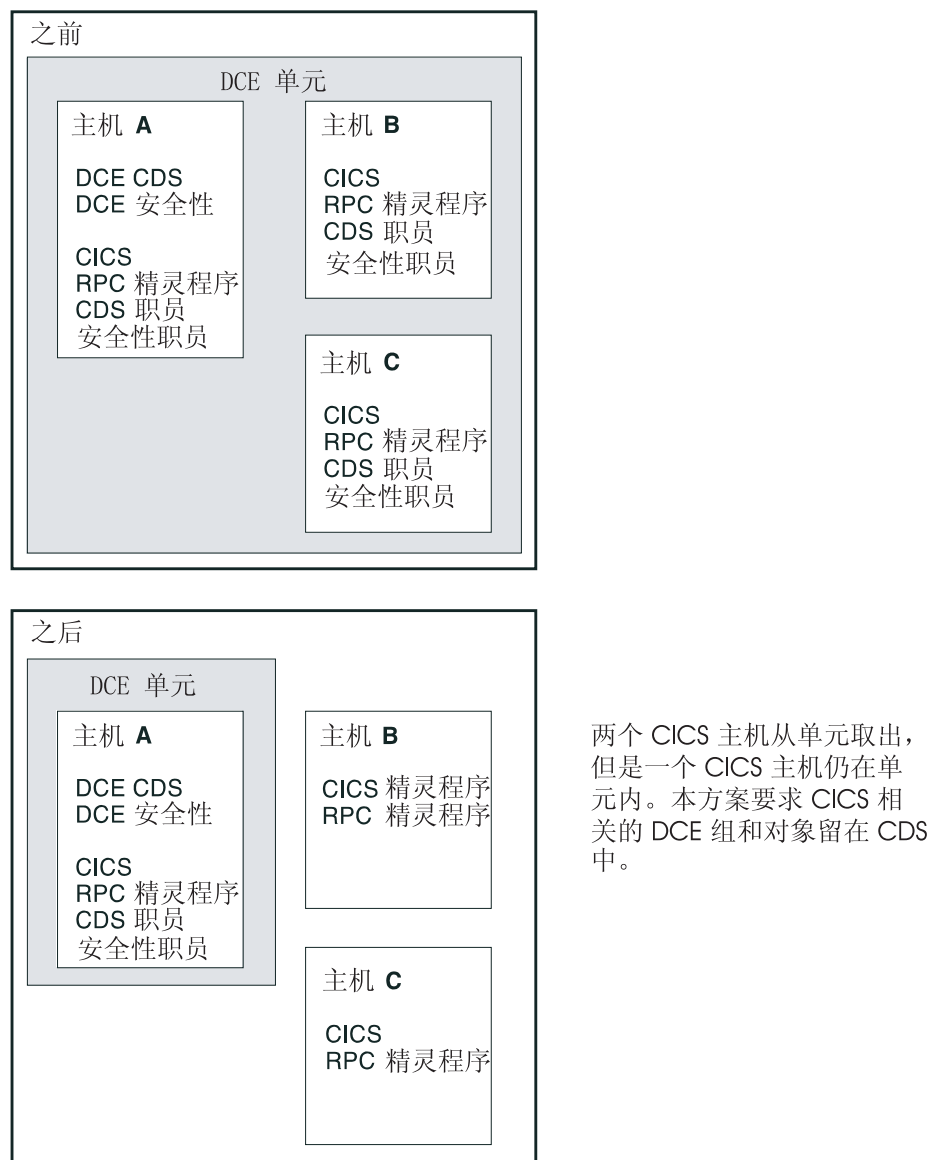


图 7. 方案 3

在该方案中，从单元中删除了两个 CICS 主机，但单元却完整无缺，并且与 CICS 有关的组和对象没有从单元中删除。将第262页的『从单元中删除一些 CICS 主机』用于此方案。然后使用第262页的『配置仅 RPC 设置』来重新配置 CICS。

消除单元

使用该过程删除任何 DCE 配置，包括 DCE 服务器，为配置仅 RPC 设置做准备。

需要使用本过程:

- 必须注册到一个不拥有超级用户特权的用户标识符，该用户标识符如第3页的『建立 CICS 系统管理员』中描述的那样设置。
- 必须作为单元管理员 (dce_login cell_admin) 注册到 DCE 中。

- 所有在单元中的区域和 Encina 服务器都必须关闭。使用正常的关机过程，因此可以在完成移植后热启动。

注：非常重要：要在开始前关闭区域和服务。在关机前消除 DCE 配置需要一次强迫关机，这说明可能不能热启动。

在单元内每个主机上完成以下步骤，完整地删除所有的 DCE 配置。对 DCE 服务器的主机的操作放在最后进行。

1. 删除下列文件，如果它们存在的话：

```
/var/cics_regions/regionName/keytab
/var/cics_servers/SSD/cics/sfs/serverId/key
/var/cics_servers/GSD/cics/ppc/gateway/serverId/key
/var/cics_clients/numbers.key
```

使用 **rm dce -o local all**

从单元中删除所有的 CICS 主机

使用该过程删除任何 DCE 配置，为配置仅 RPC 设置做准备。该过程从单元中删除了所有与 CICS 有关的 DCE 配置，但并不消除单元。

需要使用本过程：

- 必须注册到一个不拥有超级用户特权的用户标识符，该用户标识符如 第3页的『建立 CICS 系统管理员』中描述的那样设置。
- 必须作为单元管理员（dce_login cell_admin）注册到 DCE 中。
- 所有在单元中的区域和 Encina 服务器都必须关闭。使用正常的关机过程，因此可以在完成移植后热启动。

注：非常重要：要在开始前关闭区域和服务。在关机前消除 DCE 配置需要一次强迫关机，这说明可能不能热启动。

1. 在每个主机上完成以下步骤，删除所有的 DCE 配置。**不要对 DCE 服务器主机进行这些操作。**下列指导只是针对 CICS AIX 版的。
 - a. 使用 **cicscleandce** 命令从单元上删除与 CICS 有关的 DCE 组和对象。在单元中只需要做一次。
 - b. 使用 **/etc/dce.clean** 停止 DCE。
 - c. 使用 **rm dce all** 删除本地 DCE 配置。

如果要在不是 CICS AIX 版 主机的主机运行 DCE，则请参见产品自身的文档。

2. 如果主机有一个设置用于 CICS 的 Encina PPC 网关服务器，并且在主机上没有安装 CICS，则需要使用 DCE **rgy_edit** 设施或 **dcecp** 删除服务器的委托人。该示例使用了 **rgy_edit**（要得到关于如何使用 dcecp 的信息，请参阅 AIX 文档。）输入：
 - a. **rgy_edit**
 - b. 域委托人
 - c. **del /cics/ppc/gateway/serverId**
 - d. **quit**

然后使用适当的平台命令停止 DCE 并删除本地配置。

从单元中删除一些 CICS 主机

使用该过程删除任何 DCE 配置，为配置仅 RPC 设置做准备。该过程从一些主机的单元中删除了 DCE 配置，但在 CICS 主机中保留了与 CICS 有关的组和对象（是那些没有移植到仅 RPC 设置中的）。

需要使用本过程：

- 必须注册到一个不拥有超级用户特权的用户标识符，该用户标识符如 第3页的『建立 CICS 系统管理员』中描述的那样设置。
- 必须作为单元管理员（dce_login cell_admin）注册到 DCE 中。
- 所有在单元中的区域和 Encina 服务器都必须关闭。使用正常的关机过程，因此可以在完成移植后热启动。

注：非常重要：要在开始前关闭区域和服务。在关机前消除 DCE 配置需要一次强迫关机，这说明可能不能热启动。

1. 在每个主机上完成以下步骤，删除 DCE 配置。 **不要对 DCE 服务器主机进行这些操作。** 下列指导只是针对 CICS AIX 版的。

- a. 使用 **/etc/dce.clean** 停止 DCE。
- b. 使用 **rm dce all** 删除本地 DCE 配置。

如果要在不是 CICS AIX 版 主机的主机运行 DCE，则请参见产品自身的文档。

2. 如果主机有一个设置用于 CICS 的 Encina PPC 网关服务器，并且在主机上没有安装 CICS，则需要使用 **dcecp** 删除服务器的委托人。该示例使用 **rgy-edit** 输入：

- a. **rgy_edit**
- b. **domain principal**
- c. **del /cics/ppc/gateway/serverId**
- d. **quit**

然后使用适当的平台命令停止 DCE 并删除本地配置。

配置仅 RPC 设置

使用该过程在删除 DCE 配置后进行仅 RPC 的设置。在此过程中，首先操作 Encina 服务器、然后是区域、再然后是客户。

需要使用本过程：

- 必须注册到一个不拥有超级用户特权的用户标识符，该用户标识符如 第3页的『建立 CICS 系统管理员』中描述的那样设置。
- 必须已完成 第260页的『消除单元』、第261页的『从单元中删除所有的 CICS 主机』或 第262页的『从单元中删除一些 CICS 主机』。

1. 在具有 SFS 的主机上：使用下列命令配置 RPC 端点映像程序（**在每个主机上只需要做一次**）。

```
cicscp -v -l logFile create dce -R
```

- a. 对于主机上所有永久和运行时的 结构化文件服务器定义（SSD）项，将 **NameService** 属性设置为 **NONE** 并且将 **ProtectionLevel** 属性设置为 **none**。

```
cicsupdateclass -w -c ssd -a NameService -n NONE
```

```
cicsupdateclass -w -c ssd -a ProtectionLevel -n none
```

- b. 保证没有设置下列环境变量:

```
ENCINA_AUTHN
```

```
ENCINA_MIN_AUTHN
```

```
ENCINA_AUTHZ
```

- c. 按照在第238页的『设置 SFS 的联接文件』中描述的过程做。

- d. 要运行在本地主机上的 Encina 命令, 则必须设置下列环境变量。

```
export ENCINA_BINDING_FILE=/var/cics_servers/server_bindings
```

应该考虑将它添加到 /etc/environment 文件中使它更永久。

- e. 使用下列命令热启动 SFS:

```
cicscp -v start sfs_server ./cics/sfs/serverId
```

2. 对于每个安装有 CICS 主机的 PPC 网关服务器 使用下列命令配置端点映象程序 (在 每个主机上只需做一次):

```
cicscp -v -l logFile create dce -R
```

- a. 对于主机上所有永久和运行时的 网关服务器定义 (GSD) 项, 将 **NameService** 属性设置为 **NONE** 并且将 **ProtectionLevel** 属性设置为 **none**。

```
cicsupdateclass -w -c gsd -a NameService -n NONE \
```

```
cicsupdateclass -w -c gsd -a ProtectionLevel -n none \
```

- b. 保证没有设置下列环境变量:

```
ENCINA_AUTHN
```

```
ENCINA_MIN_AUTHN
```

```
ENCINA_AUTHZ
```

- c. 按照下列过程, 如 *CICS 内部通信指南* 中的“在 PPC 网关服务器中设置字符串联接文件”所示。

- d. 要运行在本地主机上的 Encina 命令, 则必须设置下列环境变量。

```
export ENCINA_BINDING_FILE=/var/cics_servers/server_bindings
```

- e. 使用下列命令热启动 PPC 网关服务器:

```
cicscp -v start ppcgwy_server ./cics/ppc/gateway/serverId
```

3. 对于没有安装 CICS 的主机上的 PPC 网关服务器: 配置 RPC 端点映象程序。按照 *Encina 管理指南 卷 2: 服务器管理* 的步骤。

- a. 保证没有设置下列环境变量:

```
ENCINA_AUTHN
```

```
ENCINA_MIN_AUTHN
```

```
ENCINA_AUTHZ
```

- b. 按照下列过程, 如 *CICS 内部通信指南* 中的“在 PPC 网关服务器中设置字符串联接文件”所示。

- c. 要运行在本地主机上的 Encina 命令, 则必须设置下列环境变量。

```
export ENCINA_BINDING_FILE=/var/cics_servers/server_bindings
```

- d. 使用 **ppcgwy** 命令热启动服务器。例如:

```
$ nohup ppcgwy -n /./cics/ppc/gateway/gateABCD \
-v /var/cics_servers/GSD/cics/ppc/gateway/gateABCD/restart:\
/var/cics_servers/GSD/cics/ppc/gateway/gateABCD/restart.bak \
-l log_PgateABC/logfile \
-N 1 \
-Z 0 \
-T all:/var/cics_servers/GSD/cics/ppc/gateway/serverId/msg \
-t admin:0:dceutils:0:trdce:0:vol:0:trpc:0 \
>/dev/null 2>&1 &
```

-l 选项用来指定用于恢复 PPC 网关服务器的日志文件。通常，日志文件的名称是 **logfile**。要找到日志文件的名称，使用下列命令：

```
tkadmin query logvol -s serverName logVolName
```

这里 *serverName* 是服务器名，如 */./cics/sfs/hostABCD*，并且 *logVolName* 是日志逻辑卷的名称，如 **log_ShostABC** 或 **log_SFS_SERV**。

- e. 当服务器将 **Initialized ...** 写入信息文件时，使用下列命令，允许服务器启动 CICS 区域的处理请求：

```
tkadmin clear exclusiveauthority -s serverName
```

4. 在具有区域的主机上：使用下列命令配置 RPC 端点映像程序（在每个主机上只需要做一次）。

```
cicscp -v -l logFile create dce -R
```

- a. 在永久的运行时数据库中，将 区域定义（RD） **AuthenticationService** 属性设置为 **CICS**， **NameService** 属性设置为 **NONE** 且 **RuntimeProtection** 属性设置为 **none**。使用下列命令：

```
cicsupdateclass -r regionName -w -c rd -a AuthenticationService -n CICS
cicsupdateclass -r regionName -w -c rd -a NameService -n NONE
cicsupdateclass -r regionName -w -c rd -a RuntimeProtection -n none
```

- b. 对于所有使用口令校验的永久运行时 用户定义（UD），都需要设置 **CICSPassword** 属性。在存储到资源定义数据库时，加密该口令。参阅下列命令示例：

```
cicsupdateclass -r regionName -w -c ud -a CICSPassword -n passWord -k udKey
```

这里 *udKey* 是要更改的 UD 项的密钥，而 *passWord* 是添加到项目中的口令。

- c. 按照在 第235页的『为区域设置 CICS_HOSTS』中描述的过程做。
- d. 验证已启动了区域使用的文件管理器，然后使用下列命令热启动区域：

```
cicscp -v start region regionName
```

5. 对于 CICS 客户：

- a. 使用下列命令配置 RPC 端点映像程序（在每个主机上只需要做一次）。

```
cicscp -v -l logFile create dce -R
```

- b. 按照在快速入门 中说明的执行。

相关信息

参阅 *CICS 管理参考大全* 中的下列内容:

- “cicscp - DCE 命令”
- “cicscp - PPC 网关服务器命令”
- “cicscp - 区域命令”
- “cicscp - SFS 命令”
- “区域定义 (RD)”
- “结构化文件服务器定义(SSD)”
- “网关服务器定义 (GSD)”

同样参阅 AIX 文档以得到关于 DCE 的信息。

将 CICS 从仅 RPC 设置移植到 DCE 单元

这些过程描述了如何将仅 RPC 设置中的区域、客户和 Encina 服务器移植到 DCE 单元中。

注意:

正常的关机过程保存了关于应用程序处理的信息，因此处理可在关机发生处继续。然而，该移植进程需要冷启动，因为不能在 DCE 环境中重新启动服务器而没有先前运行时保存的 DCE 信息。

如果可能，在关闭 CICS 系统前允许应用程序完成处理。这对于 **particularly important with PPC 网关服务器** 尤其重要。

首先，遵照 第265页的『移植具有 DCE 服务器的 CICS 主机』。在配置 DCE 服务器后，对于每个添加到单元中的主机，遵照 第266页的『移植只具有 DCE 服务器的 CICS 主机』或 第267页的『移植没有 CICS 的 PPC 网关服务器』。to the cell. 当移植了所有在单元中的 第267页的『启动 CICS 系统』。

移植具有 DCE 服务器的 CICS 主机

使用该过程移植主机（将具有 DCE 服务器）上的 CICS。

需要使用本过程:

- 必须注册到一个不拥有超级用户特权的用户标识符，该用户标识符如 第3页的『建立 CICS 系统管理员』中描述的那样设置。
 - 关闭所有主机（要添加到单元中的）的区域和 Encina 服务器。将 CICS 用这种方法（可以冷启动）关闭是十分重要的。
 - 验证 DCE 服务器和客户软件是否已如 *快速入门* 描述的那样安装。
 - 假设已安装了 CICS。如果想要放置 DCE 服务器的主机上没有安装 CICS，则使用 DCE 文档来设置服务器。
1. 使用 **rm dce all** 命令清除任何在本地主机上的 DCE 配置。
 2. 删除 CICS_HOSTS 和 ENCINA_BINDING_FILE 环境变量。

3. 使用下列命令设置 CICS 和新的 DCE 单元。这包括:

- 设置 CICS 客户
- 设置 DCE CDS
- 设置 DCE 安全服务器
- 设置 DCE RPC 精灵程序和 DCE CDS , 以及安全客户进程。
- 将与 CICS 有关的组和对象添加到单元中。DCE 组是:

```
cics_admin
cics_sfs
cics_ppcgwy
cics_regions
cics_users
```

CDS 目录是:

```
././cics
././cics/sfs
././cics/ppc
././cics/ppc/gateway
././cics/trpc
././cics/tmxa
././cics/profile
```

```
cicscp -v -l logFile create dce [-n cellName]
```

这里的 *cellName* 是单元的名称。单元名称是可选的。缺省的单元名称是:

```
dce_cell.hostName.domainName
```

4. 使用下列命令, 作为单元管理员注册到 DCE 中。

```
dce_login cell_admin
```

5. 如果在本地主机上有任何区域或 Encina 服务器, 使用下列命令:

```
cicsusedceservers -A -I
```

cicsusedceservers 创建必需的 DCE 委托人和密钥表文件, 如 *CICS 管理参考大全* 中的 “cicsusedceservers - 迁移 CICS 区域和 Encina 服务器以使用 DCE” 描述的那样。它同样将区域和服务器的 **NameService** 和 **AuthenticationService** 属性更改为 **DCE**。没有更改指定 RPC 保护保护级别的各种属性。如果需要高于 **none** 的保护级别, 使用 **cicsupdate** 在启动区域或 Encina 服务器前更改有关的值。例如:

```
cicsupdate -c rd -r regionName -P RuntimeProtection='connect'
cicsupdate -c ssd ././cics/sfs/serverId -P ProtectionLevel='connect'
cicsupdate -c gsd ././cics/ppc/gateway/serverId -P ProtectionLevel='connect'
```

-I 选项继续移植其它服务器, 而不管其它服务器、区域和网关是否失败。

6. 如 第267页的『启动 CICS 系统』那样冷启动系统。

移植只具有 DCE 服务器的 CICS 主机

使用该过程移植主机 (将只具有 DCE 服务器) 上的 CICS。

1. 注册到一个不拥有超级用户特权的用户标识符，该用户标识符如 第3页的『建立 CICS 系统管理员』中描述的那样设置。
2. 使用 **rmddce all** 命令删除任何在本地主机上的 DCE 配置。
3. 使用下列命令将主机添加到单元中。这包括：

- 设置 CICS 客户
- 在本地主机上设置 DCE RPC 精灵程序和 DCE CDS，以及安全客户进程。

```
cicscp -v -l logFile create dce -n cellName -s serverHost
cicscp -v -l logFile start dce
```

这里 *cellName* 是单元的名称，而 *serverHost* 是运行 DCE CDS 和 DCE 安全服务器机器的主机名。

4. 使用下列命令，作为单元管理员注册到 DCE 中。
5. 如果在本地主机上有任何区域或 Encina 服务器，使用下列命令：

```
cicsusedceservers -A -I
```

cicsusedceservers 创建必需的 DCE 委托人和密钥表文件，如 *CICS 管理参考大全* 中的“cicsusedceservers - 迁移 CICS 区域和 Encina 服务器以使用 DCE”描述的那样。它同样将区域和服务器的 **NameService** 和 **AuthenticationService** 属性更改为 **DCE**。指定 RPC 保护级别的各种属性都设置成 **none**，然而，所以，如果需要更高的保护级别，则需要使用 **cicsupdate** 在更改区域和 Encina 服务器前更改设置选项。例如：

```
cicsupdate -c rd -r regionName -P RuntimeProtection='connect'
cicsupdate -c ssd ./cics/sfs/serverId -P ProtectionLevel='connect'
cicsupdate -c gsd ./cics/ppc/gateway/serverId -P ProtectionLevel='connect'
```

移植没有 CICS 的 PPC 网关服务器

使用该过程移植主机（没有安装 CICS）上的 PPC 网关服务器。

1. 注册到一个不拥有超级用户特权的用户标识符，该用户标识符如 第3页的『建立 CICS 系统管理员』中描述的那样设置。
2. 使用 DCE 文档配置主机上完整的 DCE 客户。
3. 作为单元管理员（`dce_login cell_admin`）注册到 DCE 中。
4. `dcecp` 要添加一个名为 `cics/ppc/gateway/serverId` 的 DCE 委托人，它属于 `cics_ppcgwy` 组。
5. 将此 DCE 委托人的口令添加到服务器的密钥表文件中：

```
/var/cics_gateways/serverId/key
```

启动 CICS 系统

当移植了所有单元中的主机时，以下列次序启动 CICS 系统：

1. 使用下列命令冷启动在 CICS 主机上的所有 PPC 网关服务器：
2. 注册到一个不拥有超级用户特权的用户标识符，该用户标识符如 第3页的『建立 CICS 系统管理员』中描述的那样设置。

```
cicscp -v start ppcgwy_server ./cics/ppc/gateway/serverId \
    StartType=cold
```

3. 冷启动所有主机（没有安装 CICS 的）上的 PPC 网关服务器，如 *Encina* 管理指南卷 2: 服务器管理 中描述的那样。
4. 对于每个 SFS:
 - a. 使用下列命令冷启动服务器:

```
cicscp -v start sfs_server /./cics/sfs/serverId StartType=cold
```
 - b. 使用下列过程配置区域的 SFS:
 - 对于相同主机上的区域来说，第240页的『使用本地命令设置队列』。
 - 对于在远程主机上的区域，第239页的『使用 cicssfscnf 设置队列』。
 - c. 使用下列命令冷启动区域:

```
cicscp -v start region regionName StartType=cold
```

相关信息

参阅 *CICS 管理参考大全* 中的下列内容:

- “cicsupdate - 修改 CICS 资源定义”
- “cicscp - DCE 命令”
- “cicscp - PPC 网关服务器命令”
- “cicscp - 区域命令”
- “区域定义 (RD)”
- “结构化文件服务器定义(SSD)”
- “cicsupdateclass 命令”
- “网关服务器定义 (GSD)”

同样参阅 AIX 文档以得到关于 DCE 的信息。

附录D. 配置 DBCS/MBCS 环境的 CICS

本章讲述了如何配置 CICS 用于 DBCS/MBCS 环境中。

表 47. 配置 DBCS/MBCS 环境 CICS 的导向图

若需...	参考...
在 SFS 文件中建立数据类型。	第269页的『使用 nlsString 数据类型』
选择或定义一个终端。	第270页的『建立和使用终端』
建立 cicsteld	第270页的『建立 cicsteld』
设置打印机	第271页的『设置打印机』

使用 nlsString 数据类型

如果计划在 SFS 文件中使用 nlsString 数据类型，必须在启动 SFS 前，将整理语言字段设置成合适的值（如， ja_JP ）。当创建 SFS 时，该值由结构化文件服务器定义（SSD）中的 **CollatingLanguage** 属性设置。该值可以改变，只须通过使用 **cicssfsimport** 命令来调入它即可，如 *CICS 管理参考大全* 中的“cicssfsimport - 建立 SFS”元数据”所示。

一旦冷启动服务器，将不能更改此值，除非再次冷启动服务器。

或者，可以使用快速路径 **smitty cicscstartssd** 来显示包含『Collating language』字段的面板。参阅 *快速入门* 以得到详细信息。

Cold Start an Encina SFS Server

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

* SFS Server Identifier

Resource description

Short name used for SRC

AIX user ID for server

Buffer pool size in Kbytes

Collating language

]

Idle timeout

Number of Log writes per checkpoint interval

Protectionlevel

Number of threads devoted to normal operations

Number of threads devoted to resource operations

First Volume (for cold start)

[EntryFields]

././cics/sfs/sfsname

SFS Server Definition

[SFS_SERV]

%S

[1000]

[Ja_jp]

[300]

[5000]

none

[12]

[3]

[sfs_%S]

F1=Help

F2=Refresh

F3=Cancel

F4=List

F5=Reset

F6=Command

F7=Edit

F8=Image

F9=Shell

F10=Exit

Enter=Do

注：等价的 Encina **sfs** 命令选项是 **-c**。参阅 *参考 Encina 文档* 以得到详细信息。

设置 LANG 变量

在执行这些步骤前不必关闭 CICS 。要建立环境:

1. 将缺省场所更改为适当的 DBCS/MBCS 语言，（通过在 **/var/cics_regions/region/environment** 文件中设置 LANG 变量）。

例如，要将缺省场所更改为 Ja_JP，在环境文件加入以下内容：

```
LANG=Ja_JP
```

2. 调出 LANG 环境变量。例如，要调出 ja_JP（Japanese EUC MBCS），输入：

```
export LANG=ja_JP
```

建立和使用终端

可以在 DBCS/MBCS 环境中使用下列 CICS 提供的终端：

- MD07: (hft-mb) High-function multi-byte color
- MD09: (mft-mb) High-function multi-byte monochrome
- MD011: (lft-mb) Low-function multi-byte monochrome

或者，可以创建自己的终端定义。要定义一个 DBCS/MBCS 环境的终端，在终端定义（WD）中设置它的 **SOSI** 和 **ProgramSymbol** 属性为 **yes**。可以使用 **cicsupdate** 命令做到这点。

例如，如果终端键是 MINE，并且区域名称是 regionA，可以使用下列示例来更新运行时和更终端定义数据库：

```
cicsupdate -c wd -r regionA -B MINE SOSI=yes ProgramSymbol=yes
```

要设置 DBCS/MBCS 环境的 CICS Telnet 服务器，以及键的 **sosiswitch** 功能，则通过添加下列语句到 3270.keys 文件中：

```
bind
sosiswitch
『 -e[013q』
shift-F1
```

建立 cicsteld

必须用以下列出的模型及场所之一来运行 **cicsteld**。场所用 **-l** 参数指定而模型用 **-e** 参数指定。

- **-l** Ja_JP
- **-l** ja_JP
- **-l** ko_KR
- **-l** zh_CN
- **-e** hft-mb
- **-e** mft-mb
- **-e** lft-mb

例如，要用运行 **cicsteld** 用一低级功能、多字节、单色终端和一个 ja_JP 场所，一个叫 regionA 的区域和 9006 的端口赋值，则输入：

```
cicsteld -r regionA -P 9006 -l ja_JP -e lft-mb &
```

如果设置了环境变量 LANG，则不用指定 **-l** 参数。

设置打印机

当终端定义为 DBCS 打印机时，需要设置下列属性：

- **CanStartATIs=yes**
- **ExtDS=yes**
- **IsPrinter=yes**
- **NumColumns=132**
- **NumLines=64**
- **ProgramSymbols/SOSI=yes**

如需详细信息，参阅 *CICS 管理参考大全* 中的“终端定义 (WD)”和“cicsternp - 打印机仿真”。

附录E. 使用 HACMP 的示例脚本

有关 **HACMP** 支持的重要提示: 请参阅 *程序目录* 或 *用户备忘* 以得到 CICS AIX 版本 2 发行本 1 提供的 HACMP 支持。

这是一系列可用于建立和运行 HACMP 的脚本。示例是参阅在 第164页的『使用一个高可用性簇多重处理 (HACMP)』中所描述的内容。

每个示例都有三个可用于 HACMP 环境的脚本。

- 在建立 HACMP 环境时运行的准备脚本
- 初始脚本在备份期间使用并且在系统故障时触发。
- 恢复脚本, 在接管进程后将系统返回到正常状态。

HACMP 示例脚本 1 (APPLID1)

```
#!/bin/ksh

echo "0: Initialize environment"
$CICSBINPATH=/usr/lpp/cics/bin
$HACMPBIN=/st/AD/src/hacmp
$CLUSTERBIN=/usr/sbin/cluster

echo "0: Initialization complete"

#
# Stage 1: Configure cluster environment
#

print "0: Stage 1, Configuring cluster environment"

$HACMPBIN/hacmpsetup

$CLUSTERBIN/clsetup -a -p 50 -n'2' -t'1' \
    $SERVICE_LABEL= \
    $FILESYSYEM='/var/server1_servers
                  /var/server1_regions
                  /var/dce
                  /etc/dce
                  /krb5
                  /dcshare' \
    $EXPORT_FILESYSTEM= \
    $MOUNT_FILESYSTEM= \
    $VOLUME_GROUP='shareserver1vg' \
    $DISK= \
    $APPLICATIONS='applid1'

print "0: Stage 1 complete"
```

图 8. APPLID1 - 准备

```

#!/bin/ksh

echo "0: Initialize environment"
$CICSBINPATH=/usr/lpp/cics/bin
$HACMPBIN=/st/AD/src/hacmp
export $KRB5CCNAME='dce_login_noexec cell_admin pwd'

echo "0: Initialization complete"

#
# Stage 1: Close down the current workload
#

echo "0: Stage 1, close down current workload"
echo "0: Closing down CICS regions"
stopsrc -fs cics.regionName2
$HACMPBIN/ctermenate regionName2
$HACMPBIN/locate_terminate cicsas
$HACMPBIN/locate_terminate cicsrl
$HACMPBIN/locate_terminate cicsic
$HACMPBIN/locate_terminate cicsld
$HACMPBIN/locate_terminate cicsam

echo "0: Closing down Encina servers"
$CICSBINPATH/cicssfsshut ./cics/sfs/hostB
$HACMPBIN/locate_terminate sfs

echo "0: Closing down DCE clients"
dce.clean
$HACMPBIN/locate_terminate dce

echo "0: Stage 1 complete"

```

图 9. *APPLID1* 启动 (1/3)

```

#
# Stage 2: Modify the necessary file
#

echo "0: Stage 2, Modify local files"
echo "0: Backing up DCE files"
cp -p /etc/rc.dce rc.dce.old
cp -p /etc/dce_cf.db /etc/dce_cf.db.old
cp -p /etc/rc.dts /etc/rc.dts.old
cp -p /etc/mkdce.data /etc/mkdce.data.old
cp -p /opt/dcelocal/var/rpc/rpcdep.dat /opt/dcelocal/var/rpc/rpcdep.dat.old
cp -p /opt/dcelocal/var/rpc/rpcdllb.dat /opt/dcelocal/var/rpc/rpcdllb.dat.old

echo "0: Tidying up unnecessary DCE files"
rm /opt/dcelocal/var/rpc/rpcdep.dat
rm /opt/dcelocal/var/rpc/rpcdllb.dat
rm /opt/dcelocal/var/security/creds/*

echo "0: Installing new versions of DCE files"
cp -p /dceshare/rc.dce /etc/rc.dce
cp -p /dceshare/dce_cf.db /etc/dce_cf.db
cp -p /dceshare/rc.dts /etc/rc.dts
cp -p /dceshare/mkdce.data /etc/mkdce.data

echo "0: Backing up CICS Encina Server files"
cp -p /var/cics_servers/SSD/SSD.stanza /var/cics_servers/SSD/SSD.stanza.old
cp -p /var/cics_servers/SSD/AUTO.stanza /var/cics_servers/SSD/AUTO.stanza.old

echo "0: Installing new versions of CICS Encina Server files"
cp -p /var/server1_servers/SSD/SSD.stanza /var/cics_servers/SSD/SSD.stanza
cp -p /var/server1_servers/SSD/AUTO.stanza /var/cics_servers/SSD/AUTO.stanza

echo "0: Stage 2 complete"

```

图 9. APPLID1 启动 (2/3)

```

#
# Stage 3: Initiate the required workload
#

echo "0: Stage 3, Initiate required workload"
echo "0: Starting dce"
rc.dce
export $KRB5CCNAME='dce_login_noexec cell_admin pwd'

echo "0: Starting sfs server server1_ha"
$CICSBINPATH/cicssfslock -u ./cics/sfs/server1_ha
$CICSBINPATH/cicssfs ./cics/sfs/server1_ha StartType=auto

echo "0: Starting region region1"
$CICSBINPATH/cicsrlck -r regionName1
startsrc -s cics.regionName1 -a StartType=cold

echo "0: Stage 3 complete"

```

图 9. APPLID1 启动 (3/3)

```

#!/bin/ksh

print "0: Initialize environment"
$CICSBINPATH=/usr/lpp/cics/bin
$HACMPBIN=/st/AD/src/hacmp
export $KRB5CCNAME='dce_login_noexec cell_admin pwd'

print "0: Initialization complete"

#
# Stage 1: Close down the current workload
#

print "0: Stage 1, close down current workload"
print "0: Closing down CICS regions"
stopsrc -fs cics.regionName1
$HACMPBIN/ctermenate regionName1
$HACMPBIN/locate_terminate cicsas
$HACMPBIN/locate_terminate cicsrl
$HACMPBIN/locate_terminate cicsic
$HACMPBIN/locate_terminate cicsld
$HACMPBIN/locate_terminate cicsam

print "0: Closing down Encina servers"
$CICSBINPATH/cicssfsshut ././cics/sfs/server1_ha
$HACMPBIN/locate_terminate sfs

print "0: Closing down DCE clients"
dce.clean
$HACMPBIN/locate_terminate dce

print "0: Stage 1 complete"

```

图 10. APPLID1 恢复 (1/2)

```

#
# Stage 2: Modify the necessary file
#

print "0: Stage 2, Modify local files"
print "0: Recovering DCE files"
cp -p /etc/rc.dce.old /etc/rc.dce
cp -p /etc/dce_cf.db.old /etc/dce_cf.db
cp -p /etc/rc.dts.old /etc/rc.dts
cp -p /etc/mkdce.data.old /etc/mkdce.data
cp -p /opt/dcelocal/var/rpc/rpcdep.dat.old /opt/dcelocal/var/rpc/rpcdep.dat
cp -p /opt/dcelocal/var/rpc/rpcdllb.dat.old /opt/dcelocal/var/rpc/rpcdllb.dat

print "0: Tidying up unnecessary DCE files"
rm /opt/dcelocal/var/security/creds/*

print "0: Recovering CICS Encina Server files"
cp -p /var/cics_servers/SSD/SSD.stanza.old /var/cics_servers/SSD/SSD.stanza
cp -p /var/cics_servers/SSD/AUTO.stanza.old /var/cics_servers/SSD/AUTO.stanza

print "0: Stage 2 complete"

#
# Stage 3: Initiate the required workload
#

print "0: Stage 3, Initiate required workload"
print "0: Starting dce"
rc.dce
export $KRB5CCNAME='dce_login_noexec cell_admin pwd'

print "0: The environment was non-essential, the user must restart"
print "    encina servers and regions as required."

print "0: Stage 3 complete"

```

图 10. APPLID1 恢复 (2/2)

HACMP 示例脚本 2 (APPLID2)

```
#!/bin/ksh
#
#
echo "0: Initialise environment"
CICSBINPATH=/usr/lpp/cics/bin
HACMPBIN=/st/AD/src/hacmp
CLUSTERBIN=/usr/sbin/cluster

echo "0: Initialization complete"

#
# Stage 1: Configure cluster environment
#

print "0: Stage 1, Configuring cluster environment"

HACMPBIN/hacmpsetup

CLUSTERBIN/clsetup -a -p 50 -n'2' -t'1' \
    SERVICE_LABEL= \
    FILESYSTEM='/var/hostB_servers' \
    /var/hostB_regions' \
    EXPORT_FILESYSTEM= \
    MOUNT_FILESYSTEM= \
    VOLUME_GROUP='sharehostBvg' \
    DISK= \
    APPLICATIONS='applid2'

print "0: Stage 1 complete"
```

图 11. APPLID2 准备

```

#!/bin/ksh
#
#
print "0: Initialise environment"
CICSBINPATH=/usr/lpp/cics/*/bin
HACMPBIN=/st/AD/src/hacmp
export $KRB5CCNAME='dce_login_noexec cell_admin pwd'

print "0: Initialization complete"

#
# Stage 1: Close down the current workload
#

print "0: Stage 1, close down current workload"
print "0: No processes to close"

print "0: Stage 1 complete"

#
# Stage 2: Modify the necessary file
#

print "0: Stage 2, Modify local files"
print "0: Backing up CICS Encina Server files"
cp -p /var/cics_servers/SSD/SSD.stanza /var/cics_servers/SSD/SSD.stanza.old
cp -p /var/cics_servers/SSD/AUTO.stanza /var/cics_servers/SSD/AUTO.stanza.old

print "0: Merging extra CICS Encina Server files"
cat /var/server1_servers/SSD/SSD.stanza >> /var/cics_servers/SSD/SSD.stanza
cat /var/server1_servers/SSD/AUTO.stanza >> /var/cics_servers/SSD/AUTO.stanza

print "0: Stage 2 complete"

#
# Stage 3: Initiate the required workload
#

print "0: Stage 3, Initiate required workload"

print "0: Starting sfs server server1_ha"
CICSBINPATH/cicssfslock -u ./cics/sfs/server1_ha
CICSBINPATH/cicssfs ./cics/sfs/server1_ha StartType=auto

print "0: Starting region regionName1"
CICSBINPATH/cicsrlck -r regionName1
startsrc -s cics.regionName1 -a StartType=cold

print "0: Stage 3 complete"

```

图 12. APPLID2 启动


```

#!/bin/ksh
#
#
print "0: Initialize environment"
CICSBINPATH=/usr/lpp/cics/bin
HACMPBIN=/st/AD/src/hacmp
export $KRB5CCNAME='dce_login_noexec cell_admin pwd'

print "0: Initialization complete"

#
# Stage 1: Close down the current workload
#

print "0: Stage 1, close down current workload"
print "0: Closing down CICS regions"
stopsrc -fs cics.regionName2
HACMPBIN/ctermiate regionName2

print "0: Closing down Encina servers"
CICSBINPATH/cicssfssht /./cics/sfs/hostB

print "0: Stage 1 complete"

#
# Stage 2: Modify the necessary file
#

print "0: Stage 2, Modify local files"
print "0: Recovering CICS Encina Server files"
cp -p /var/cics_servers/SSD/SSD.stanza.old /var/cics_servers/SSD/SSD.stanza
cp -p /var/cics_servers/SSD/AUTO.stanza.old /var/cics_servers/SSD/AUTO.stanza

print "0: Stage 2 complete"

#
# Stage 3: Initiate the required workload
#

print "0: Stage 3, Initiate required workload"
print "0: No workload to reinstate for scenario 2 testing"

print "0: Stage 3 complete"

```

图 13. APPLID2 恢复

HACMP 示例脚本 3 (APPLID3)

```
#!/bin/ksh

echo "0: Initialize environment"
CICSBINPATH=/usr/lpp/cics/bin
HACMPBIN=/st/AD/src/hacmp
CLUSTERBIN=/usr/sbin/cluster

echo "0: Initialization complete"

#
# Stage 1: Configure cluster environment
#

print "0: Stage 1, Configuring cluster environment"

HACMPBIN/hacmpsetup

CLUSTERBIN/clsetup -a -p 50 -n'2' -t'1' \
    SERVICE_LABEL= \
    FILESYSTEM='/var/server1_servers' \
    /var/server1_regions' \
    EXPORT_FILESYSTEM= \
    MOUNT_FILESYSTEM= \
    VOLUME_GROUP='shareserver1vg' \
    DISK= \
    APPLICATIONS='applid3'

CLUSTERBIN/clsetup -a -p 50 -n'1' -t'2' \
    SERVICE_LABEL= \
    FILESYSTEM='/var/hostB_servers' \
    /var/hostB_regions' \
    EXPORT_FILESYSTEM= \
    MOUNT_FILESYSTEM= \
    VOLUME_GROUP='sharehostBvg' \
    DISK= \
    APPLICATIONS='applid4'

print "0: Stage 1 complete"
```

图 14. APPLID3 准备

```

#!/bin/ksh

print "0: Initialize environment"
CICSBINPATH=/usr/lpp/cics/bin
HACMPBIN=/st/AD/src/hacmp
export $KRB5CCNAME='dce_login_noexec cell_admin pwd'

print "0: Initialization complete"

#
# Stage 1: Close down the current workload
#

print "0: Stage 1, close down current workload"
print "0: No processes to close"

print "0: Stage 1 complete"

#
# Stage 2: Modify the necessary file
#

print "0: Stage 2, Modify local files"
print "0: No local files to be modified"

print "0: Stage 2 complete"

#
# Stage 3: Initiate the required workload
#

print "0: Stage 3, Initiate required workload"

print "0: Starting region regionName2"
CICSBINPATH/cicsrlck -r regionName2
startsrc -s cics.regionName2 -a StartType=cold

print "0: Stage 3 complete"

```

图 15. APPLID3 启动

```

#!/bin/ksh

print "0: Initialize environment"
CICSBINPATH=/usr/lpp/cics/bin
HACMPBIN=/st/AD/src/hacmp
export $KRB5CCNAME='dce_login_noexec cell_admin pwd'

print "0: Initialization complete"

#
# Stage 1: Close down the current workload
#

print "0: Stage 1, close down current workload"
print "0: Closing down CICS regions"
stopsrc -fs cics.regionName2
HACMPBIN/ctermenate regionName2

print "0: Stage 1 complete"

#
# Stage 2: Modify the necessary file
#

print "0: Stage 2, Modify local files"
print "0: No local files to be modified for scenario 3 testing"

print "0: Stage 2 complete"

#
# Stage 3: Initiate the required workload
#

print "0: Stage 3, Initiate required workload"
print "0: No workload to reinstate"

print "0: Stage 3 complete"

```

图 16. APPLID3 恢复

HACMP 示例脚本 4 (APPLID4)

```
#!/bin/ksh

echo "0: Initialize environment"
CICSBINPATH=/usr/lpp/cics/bin
HACMPBIN=/st/AD/src/hacmp
CLUSTERBIN=/usr/sbin/cluster

echo "0: Initialization complete"

#
# Stage 1: Configure cluster environment
#

print "0: Stage 1, Configuring cluster environment"

HACMPBIN/hacmpsetup

CLUSTERBIN/clsetup -a -p 50 -n'2' -t'1' \
    SERVICE_LABEL= \
    FILESYSTEM='/var/server1_servers' \
    /var/server1_regions' \
    EXPORT_FILESYSTEM= \
    MOUNT_FILESYSTEM= \
    VOLUME_GROUP='shareserver1vg' \
    DISK= \
    APPLICATIONS='applid3'

CLUSTERBIN/clsetup -a -p 50 -n'1' -t'2' \
    SERVICE_LABEL= \
    FILESYSTEM='/var/hostB_servers' \
    /var/hostB_regions' \
    EXPORT_FILESYSTEM= \
    MOUNT_FILESYSTEM= \
    VOLUME_GROUP='sharehostBvg' \
    DISK= \
    APPLICATIONS='applid4'

print "0: Stage 1 complete"
```

图 17. APPLID4 准备

```

#!/bin/ksh

print "0: Initialize environment"
CICSBINPATH=/usr/lpp/cics/bin
HACMPBIN=/st/AD/src/hacmp
export $KRB5CCNAME='dce_login_noexec cell_admin pwd'

print "0: Initialization complete"

#
# Stage 1: Close down the current workload
#

print "0: Stage 1, close down current workload"
print "0: Warning: Closing region regionName2"
HACMPBIN/ctermiate regionName2

print "0: Stage 1 complete"

#
# Stage 2: Modify the necessary file
#

print "0: Stage 2, Modify local files"
print "0: Backup up AUTO stanza files only"
cp -p /var/cics_servers/SSD/AUTO.stanza /var/cics_servers/SSD/AUTO.stanza.old

print "0: Modify AUTO stanza files only"
cat /var/hostB_servers/SSD/AUTO.stanza >> /var/cics_servers/SSD/AUTO.stanza

print "0: Stage 2 complete"

#
# Stage 3: Initiate the required workload
#

print "0: Stage 3, Initiate required workload"

print "0: Starting sfs server hostB_ha"
CICSBINPATH/cicssflock -u ./cics/sfs/hostB_ha
CICSBINPATH/cicssfss ./cics/sfs/hostB_ha StartType=auto

print "0: Starting region regionName2"
CICSBINPATH/cicsrlck -r regionName2
startsrc -s cics.regionName2 -a StartType=cold

print "0: Stage 3 complete"

```

图 18. APPLID4 启动

```

#!/bin/ksh

print "0: Initialize environment"
CICSBINPATH=/usr/lpp/cics/bin
HACMPBIN=/st/AD/src/hacmp
export $KRB5CCNAME='dce_login_noexec cell_admin pwd'

print "0: Initialization complete"

#
# Stage 1: Close down the current workload
#

print "0: Stage 1, close down current workload"
print "0: Closing down region regionName2"
ctermenate regionName2

print "0: Closing down Encina servers"
CICSBINPATH/cicssfssht /./cics/sfs/hostB_ha

print "0: Stage 1 complete"

#
# Stage 2: Modify the necessary file
#

print "0: Stage 2, Modify local files"
print "0: Recovering CICS Encina AUTO stanzas only"
cp -p /var/cics_servers/SSD/AUTO.stanza.old /var/cics_servers/SSD/AUTO.stanza

print "0: Stage 2 complete"

#
# Stage 3: Initiate the required workload
#

print "0: Stage 3, Initiate required workload"
print "0: No workload to reinstate for scenario 4 testing"

print "0: Warning: Regions regionName2 should be restarted when the log"
print "0: and SFS servers are running on hostB again."

print "0: Stage 3 complete"

```

图 19. APPLID4 恢复

与以上示例相关联的脚本

下列四个脚本是参考以上示例的。


```
#!/bin/ksh

if [ "1" != "" ]
then
    CICSREGION="1"
fi

if [ "CICSREGION" = "" ]
then
    echo "Usage: 0 region"
    exit
fi

for i in `ps -eaF"pid args" | grep cics | grep CICSREGION | sed "s/^ *//" | cut -f1 -d" "`
do
    echo i
    kill -9 i
done

cicsrlck -r CICSREGION
ps -eaf | grep CICSREGION
```

图 20. *ctermminate*. **ctermminate** 终止了 *CICS* 进程并删除了 *CICS* 区域。应该小心使用它，因为它可能删除操作中关键的进程。

```
#!/bin/ksh

if [ "1" = "" ]
then
    echo "Usage: 0 text"
    exit
fi

for i in `ps -eaF"pid args" | grep 1 | grep -v grep | grep -v 0 | sed "s/^ *//" | cut -f1 -d" "`
do
    print "0: Terminating: i"
    kill -9 i
done
```

图 21. *locate_terminate*. **locate_terminate** 终止了 *CICS* 进程并删除了 *CICS* 区域。应该小心使用它，因为它可能删除操作中关键的进程。

```

#!/bin/ksh

echo "0: Initialize environment"
CICSBINPATH=/usr/lpp/cics/bin
HACMPBIN=/st/AD/src/hacmp
CLUSTERBIN=/usr/sbin/cluster

echo "0: Initialization complete"

#
# Stage 1: Close down the current workload
#

print "0: Stage 1, Cluster setup"
print "0: Define applications"
HACMPBIN/hacmpappl applid1
HACMPBIN/hacmpappl applid2
HACMPBIN/hacmpappl applid3
HACMPBIN/hacmpappl applid4

print "0: Define owned resources for node 2"

CLUSTERBIN/clsetup -o -p 100 -n'2' \
    FILESYSTEM='/krb5
                /var/dce
                /etc/dce
                /dcshare
                /var/server1_servers
                /var/server1_regions' \
    EXPORT_FILESYSTEM= \
    VOLUME_GROUP='shareserver1vg' \
    CONCURRENT_VOLUME_GROUP= \
    DISK= \
    APPLICATIONS=

print "0: Define owned resources for node 1"

CLUSTERBIN/clsetup -o -p 100 -n'1' \
    FILESYSTEM='/var/hostB_regions
                /var/hostB_servers' \
    EXPORT_FILESYSTEM= \
    VOLUME_GROUP='sharehostBvg' \
    CONCURRENT_VOLUME_GROUP= \
    DISK= \
    APPLICATIONS=

print "0: Stage 1 complete"

```

图 22. *hacmpsetup*. **hacmpsetup** is used to configure the HACMP environment.

```

#!/bin/ksh

echo "0: Initialize environment"
CICSBINPATH=/usr/lpp/cics/bin
HACMPBIN=/st/AD/src/hacmp
CLUSTERBIN=/usr/sbin/cluster

echo "0: Initialization complete"

if [ "1" = "" ]
then
    print "Usage: 0 <application>"
    exit
fi

#
# Stage 1: Close down the current workload
#

print "0: Stage 1, Application server definition"
print "0: Checking if it exists"
Output='CLUSTERBIN/cllsserv -n 1'

if [ "Output" != "" ]
then
    print "0: Removing application definition"
    CLUSTERBIN/clrmserv 1
fi

print "0: Adding application definition"
CLUSTERBIN/claddserv -s1 -bHACMPBIN/1_initiate -eHACMPBIN/do_nothing

print "0: Stage 1 complete"

```

图 23. *hacmpappl*. 初始脚本使用 **hacmpappl** 来启动 HACMP 需要的进程。

公告

第一版(1997 年 12 月)

下段不适用于英国或其他当地法律与本条款不一致的国家:

担保与责任

本文档的更改不会另行通知, 而且并不表示承诺 Transarc 公司会以任何方式维护、更新、修改文档或任何相关软件。本出版物是以“AS IS”(即“原样”)方式提供, 不具有任何形式的担保, 无论是明确的还是隐式的, 其中包括但不限于对特定目的之可销售性或适宜性的隐式担保。在任何情况下, 对于本文档的设计、生产或投放市场所涉及的 TRANSARC 及其它任何团体, 对任何间接的、意外的、特殊的损坏, 都不负有责任, 即使已告知存在这些损坏的可能性。

有些州不允许对某些事务的明确或隐式担保推卸责任, 因此上述条款对您或许不适用。

本出版物可能会包含技术上的不精确性或印刷错误。此处提到的信息也会随时更改; 这些信息将会被合并至出版物的新版本中。Transarc 可以在任何时间对本出版物中描述的产品和(或)程序进行改进和(或)更改。

本出版物中可能会包含对有些未在你的国家推出的 Transarc 软件产品、程序设计或服务的引用或信息。这些引用或信息并不意味着 Transarc 打算在你的国家推出这些 Transarc 产品、程序设计或服务。

应该向 Transarc 或 Transarc 特许的代理商请求 Transarc 产品的技术信息。

美国政府用户注意事项

软件和(或)文档的使用、复制或公开受到下列条款的限制: 如果软件和(或)文档是为国防部(“DoD”)提供的, 那么根据联邦获取规则(“DFARS”)的 DoD 补充条例中 252.227-7014 (或任何后续法规), 它归为“商业计算机软件”, 政府仅获得在此处授予的许可证权利(通常提供给非政府用户的许可证权利)。如果软件和(或)文档是为非 DoD 的任何美国政府单位或机构提供的, 它归为“受限计算机软件”, 对软件和(或)文档的政府权利定义在联邦获取规则(“FAR”)的 52.227-19 中(或任何后续法规); 如果软件和(或)文档是为 NASA 提供的, 则政府权利定义在 FAR 的 NASA 补充条例的 18.52.227-86 中(或任何后续法规)。

拒负责任声明

本出版物中对 Transarc 软件程序、文档或服务的引用并不意味着 Transarc 打算使它们在所有 Transarc 运作的国家中都可用。所有对 Transarc 软件程序、文档或服务的引用并不说明或暗示只可以使用 Transarc 软件程序、文档或服务。根据 Transarc 有效知识产权或其它合法的权利, 任何功能上等价的软件程序、文档或服务都可以用来代替 Transarc 软件、文档或服务。在与其它产品结合使用时, 除了那些由 Transarc 明确指定的产品之外, 其评估和验证均由用户自行负责。

对于本文档覆盖的内容材料，Transarc 可能拥有专利或者未决的专利申请。本文档的提供并没有授权或给予您这些专利的任何许可证。您可以用书面方式将许可证查询寄往：

Transarc Corporation
Attn.: Software Licensing
The Gulf Tower
707 Grant Street
Pittsburgh, PA 15219

商标和服务标记

下列术语是 IBM 公司在美国或其他国家的商标或注册商标：

AFP	DATABASE 2	Operating System/2
AIX	DB2	P2P
AIX/6000	Extended Services	POWERparallel
AIXwindows	FAA	RISC System/6000
AS/400	GDDM	RS/6000
AT	HACMP/6000	SAA
BookManager	IBM	SNA
CICS	IBMLink	SP2
CICS OS/2	IMS	SystemView
CICS/400	InfoExplorer	TXSeries
CICS/6000	Library Reader	VSE/ESA
CICS/ESA	MQSeries	VTAM
CICS/MVS	MVS/ESA	VisualAge
CICS/VSE	MVS/XA	Xstation
CICSplex	OS/2	
CT	OS/400	

Transarc、Encina 和 AFS 是 Transarc 公司在美国或其他国家的注册商标； DFS 和 DCE Encina Lightweight Client 是 Transarc 公司在美国或其他国家的商标。

下列术语是其他公司的商标或服务标记：

Microsoft、Windows、Windows NT 和 Windows 95 徽标是 Microsoft 公司的注册商标。

UNIX 是 Open Group 的注册商标。

OSF 和 Open Software Foundation 是 Open Software Foundation 公司的注册商标。

HP、Hewlett-Packard 和 HP-UX 是惠普公司的注册商标。

Orbix 是 IONA 技术有限公司的注册商标； OrbixWeb 是该公司的商标。

Sun、SunLink、Solaris、SunOS、Java、NFS 和 Sun Microsystems 是 Sun Microsystems 公司的商标或注册商标。

其它公司、产品和服务名称可能是其它公司的商标或服务标记。

索引

本索引按汉语拼音，数字，英文字母和特殊字符顺序排列。

[A]

安全性

- 程序 222
- 存取控制列表 210
- 访问 CICS 提供的事务 216
- 服务的恢复 154
- 故障后的恢复 170
- 含有数据的服务 162
- 开发和测试的服务 162
- 来自应用程序的资源 220
- 临时存储器队列 222
- 没有数据的服务 162
- 配置恢复 159
- 启动和关闭程序 218
- 日志 222
- 使用安全性键 221
- 使用 CICS 口令认证 205
- 使用 CICS 命令的资源定义 219
- 使用 CICS 认证 205
- 使用 DCE 口令认证 206
- 使用 DCE 认证 206
- 事务 215, 217, 221, 222
- 事务路由选择事务 207
- 授权访问事务 214
- 瞬时数据队列 222
- 提供的事务处理 222
- 外部 114
- 委托人, DCE 206
- 文件 222
- 性能成本 180
- 硬件故障 163
- 用户 205, 206
- 用户标识符恢复 161
- 用户访问 CICS 205, 218
- 运行时间区域以外的资源 219
- 终端保护 207
- 终端运行时间保护级别 207
- 装运终端 207
- 资源 221, 222, 224
- 子系统项的恢复 160
- ATI 215
- CEBR 222
- CECI 222
- CEDF 222
- CEDF 调试事务 216

安全性 (续)

- CICS 认证 205
- CICS 提供的事务 222
- cicsteld 206
- cicsterm 206, 208, 209
- CPMI 222
- CSM1 222
- CSMI 222
- DB2 的恢复 158
- DCE 认证 206
- DCE 委托人 207
- DefaultUserId, 用于 206
- Encina 服务器 210
- SFS 的恢复 155
- Telnet 208
- XA 生效关系数据库 209

安全性考虑, DB2 28

安全性文件管理器

- 可用性问题 128
- 启动失败 128

安装, CICS AIX 版 客户 232

[B]

保护级别

- 属性 211
- 终端 207

备份和恢复

- 防止数据丢失 163
- 概述 149
- 高可用性簇多重处理 (HACMP) 164
- 恢复服务的不同类型 162
- 媒体归档 157
- 配置的 159
- 前卷 161
- 使用第二处理器 163
- 使用镜像卷 163
- 完全故障后 170
- 硬件故障的容错 163
- 用户标识符的 161
- 子系统项 160
- CICS 子系统项 160
- DB2 的 158
- DCE 服务器 153
- HACMP (高可用性簇多重处理) 164
- SFS 的 155

表格空间, DB2 243

表, DB2 185

不可恢复临时存储器队列 240

不可恢复瞬时数据队列 240

不确定条件 38

[C]

测试系统, 需要的备份和恢复 162

产品定义 (XAD), 装入 4

场所

指定 DBCS/MBCS 环境 270

程序

安全性 222

设置 39

程序定义 (PD)

更改 39

共享资源 22

配置 39

删除 39

添加 39

cicsadd 39

cicsdelete 39

cicsupdate 39

程序定义 (PD) 属性, 用于, RSLKey 222

程序管理, 配置 程序定义 (PD) 39

重路由选择 108

重新启动文件, 在正常关闭时的使用 135

重新启动, 性能开销 180

创建 (cicsddt 命令) 47

存储器

违反 178

性能 177, 178, 181

存储器压力条件 177

存取控制列表 210, 212

[D]

打印机

设置 56

终端 56

打印机仿真, 过程 85

打印机管理, 配置 终端定义 (WD) 56

打印机设置, DBCS 271

单阶段落实优化 88

单元目录服务 (CDS), 为...配置 233, 234

单元目录客户 232

调用 RPC 保护级别 211

动态分布程序连接用户出口 111

动态事务复原 135

动态事务路由选择用户出口 108

端点 238

例子 238

TCP 例子 238

UDP 例子 238

队列管理

配置过程

DB2 28

队列管理 (续)

配置过程 (续)

SFS 24

SQL 28

需求 (SFS) 26

(SQL) 的命名 28

[F]

非请求统计 194

分布式时间服务, DCE 131

符号链接 22

服务类型, 需要的备份和恢复 162

服务, 备份和恢复 154

辅助存储器跟踪 202

辅助跟踪 202

辅助临时存储器队列, 恢复 150

辅助临时存储器, 性能成本 179

[G]

高可用性簇多重处理 (HACMP) 164

跟踪

辅助 202

辅助存储器 (auxiliary storage) 202

外部存储器 (external storage) 202

性能成本 179

主存储器 (main storage) 202

跟踪表 202

跟踪实用程序 (cicstfmt) 202

更改, 口令 83

更改, 区域定义 (RD) 33

更新, 运行时 141

共享任务存储器池, 性能 177, 178

共享资源 22

工作负荷均衡 108

关闭

立即 135

强制异常 135

区域 137

取消 (强制异常) 135

软件启动 135

未控制 136

正常的 134

CICS 区域 137

cicsstop 247

关闭区域 133, 137

关闭统计 136

关机程序, 写时的考虑 118

关机, 性能开销 180

关系数据库管理器

队列和文件管理 243

建立 XA 接口 93

关系数据库管理器 (续)

支持 XA 的接口 90, 96

DB2 和 XA 接口 90, 243

Informix 和 XA 接口 93

Oracle 和 XA 接口 96

关系数据库, 集成支持 XA 的 90, 96

管理区域, 消除区域 35

管理资源

安装永久定义 13

建立区域 21

输出区域 234

输入区域 234

添加区域 234

cicsdefault 命令 233

SFS 的区域定义 (RD) 24

管理 SFS

消除 DCE 配置 255

消除 SFS 36, 249

过程

创建 SFS 联接文件 238

创建 SFS 逻辑卷 236

创建 SFS 用户标识符 236

打印机仿真 85

端点设置 238

对于队列管理配置 DB2 246

配置区域 233

配置 程序定义 (PD) 39

配置 临时存储器定义 (TSD) 51

配置 模式文件定义 (SCD) 44

配置 日志定义 (JD) 54, 55

配置 事务处理定义 (TD) 37

配置 瞬时数据定义 (TDD) 52, 53

配置 文件定义 (FD) 41, 43

配置 用户定义 (UD) 83

配置 侦听程序定义 (LD) 82

配置 终端定义 (WD) 56, 58

配置 DB2 45, 47

配置 SFS 48, 239

字符串联接 235, 238

cicstermp 85

CICS_HOSTS 环境变量 235

SFS

定义 237

配置 46

启动 239

国家语言支持

客户的 73

区域的 34

信息 34, 73

BMS 40

[H]

核心文件

最大的文件大小 236

后初始程序

说明 117

写的考虑 117

环境变量

时间区 (操作系统) 131

字符串联接 235

cell_admin_pw, 使用 125

cell_admin_pw, 用法 9

CICSDB2PASSWD, 使用 246

CICSREGION, 用法 15

CICSTERM, 用法 58, 70

CICSTRACE, 使用 247

CICS_BROWSE_CACHE 188

CICS_HOSTS 235

CICS_SFS_DATA_SIZE, 用法 35

CICS_SFS_DATA_VG, 用法 35

CICS_SFS_LOG_SIZE, 用法 35

CICS_SFS_LOG_VG, 用法 35

CICS_SFS_SIZE, 用法 35

CICS_SFS_VG, 用法 35

DISPLAY, 用法 10

ENCINA_FLT_CLIENT_MAX_FDS, 用法 27

ENCINA_FLT_SERVER_MAX_FDS, 用法 27

ENCINA_FLT_SMS_SERVER_ENABLE, 用法 27

FLT, 用法 27

LANG

设置 4

使用 34, 73

LANG, 使用 270

LIBPATH

设置 4

使用 229

NLSPATH

设置 4

使用 34, 73

PATH, setting 4

TERM, 用法 58, 68, 70

恢复

辅助临时存储器队列的 150

概述 150

功能性 150

内分区瞬时数据的 150

配置的 159

示例 168, 169

用户标识符的 161

子系统项 160

自动安装终端定义的 150

CICS 用户标识符 161

恢复 (续)

- CICS 子系统项 160
- DB2 150, 169
- DB2 的 158
- DCE 服务器 153
- EXEC CICS START PROTECT 请求的 150
- MRA 162
- SFS 150, 168
- SFS 的 155
- 恢复单元, 概述 149
- 恢复和重新启动
 - 概述 149
 - 启动 127
- 恢复数据库, 使用 161, 169
- 会话式事务 61

[J]

- 基本映射支持 (BMS), NLS 40
- 记录了 ERZ0070xxE 消息 128
- 记录了 ERZ021004E 消息 128
- 记录了 ERZ0220xxE 消息 128
- 记录了 ERZ0230xxE 消息 128
- 记录了 ERZ0250xxE 消息 128
- 记录了 ERZ061013E 消息 128
- 间隔统计 194, 196
- 监控定义 (MD), 设置 60
- 监控设施
 - 设置 60
 - 时钟 197
 - 事件监控点 (EMP) 197
 - 数据集合并控制 197
 - 通过应用程序提供的数据 197
 - 性能 197
 - 性能级别 197
 - 用户 197
- cicsmfmt 202
- 监控数据格式化器 (cicsmfmt) 202
- 监控 CICS 的工具, 辅助跟踪 202
- 监控, 性能成本 179
- 简写
 - 用于指定服务器文件名 23
 - 资源定义 7
- 键表文件
 - 包含区域委托人 212
- 仅 RPC, DCE 257
- 镜像卷, 硬件容错 163
- 均衡工作负荷 108

[K]

- 开发系统, 需要的备份和恢复 162
- 开放系统上的 CICS 客户
 - 启用 66

开放系统上的 CICS 客户 (续)

- 使用 66
- 为...配置 66
- 可恢复临时存储器队列 240
- 可恢复资源
 - 逆序恢复概述 150
 - 性能成本 180
- 可用性问题的, 文件管理器 128
- 客户
 - 启用 66
 - 使用 66
 - 为...配置 66
 - 侦听程序处理 80
- 口令验证
 - 使用 CICS 205
 - 使用 DCE 206
- 口令, 更改 83
- 快速本地传送 (FLT)
 - 改进性能 186
 - 环境变量, 设置 27
- 扩展数据流属性, 用法 56

[L]

- 类, 共享 22
- 冷启动
 - 安装资源组 4
 - 使用永久数据库 4
- 立即关闭 135
- 例子, 端点 238
- 连接 RPC 保护级别 211
- 链接, 符号 22
- 两阶段落实 38
- 临时存储器定义 (TSD)
 - 更改 50
 - 共享资源 22
 - 配置 50, 51
 - 删除 50
 - 添加 50
 - cicsadd 50
 - cicsdelete 50
 - cicsupdate 50
- 临时存储器定义 (TSD) 属性, 用于, RSLKey 222
- 临时存储器队列
 - 安全性 222
 - 设置 50
- 浏览高速缓存, 用来 188
- 逻辑工作单元, 概述 149
- 逻辑卷 (SFS), 创建 236
- 逻辑可恢复瞬时数据队列 240

[M]

媒体归档

禁用 157

启用 157

没有数据的 CICS 系统, 需要的备份与恢复 162

密钥表文件

访问 67

命令

关闭 247

区域 (关闭) 137

区域 (启动) 126, 247

CICS 区域 (关闭) 137

CICS 区域 (启动) 126, 247

cicsadd 41, 50, 52

cicscp 8

cicsdb2conf 246

cicsdb2import 45

cicsddt 47

cicsdefault 233

cicsdelete 41, 50, 52

cicssdt 48

cicssfs 239

cicssfsconf 239

cicssfscreate 237

cicssfsimport 46

cicsstop 247

cicsupdate 41, 50, 52

命令, 使用

恢复数据库 161, 169

前卷数据库 161

cicscp 创建 telnet 79

cicscp 破坏 telnet 79

cicscp start all 127

cicscp start dce 126

cicscp start region 126

cicscp start sfs_server 126

cicscp stop all 138

cicscp stop dce 138

cicscp stop region 137

cicscp stop sfs_server 138

cicsmkcobol 61

cicssetupclients 67

cicsterm 67

cicstermp 85

getcellname 72, 253, 255

rmddce all 72, 253, 255

命令, 用来

cicssetupclients 232

cicssfsdestroy 249

cicssfsshut 248

cicsstart 247

命令, 用来 (续)

sar 175

模式文件定义 (SCD)

配置 44

DB2 上的 45

SFS 上的 46

模型, 自动安装 56

[N]

内分区瞬时数据队列, 配置 SFS 52

内分区瞬时数据, 恢复 150

逆序恢复, 可恢复资源的列表 150

[P]

配置

备份和恢复 159

DCE 和 CICS AIX 版 客户 232

[Q]

启动

调用用户编写程序 117

后初始程序 117

恢复和重新启动 127

区域 126

顺序 123

CICS 区域 126

cicsstart 命令 247

启动区域

cicscp 命令 126

启动失败, 文件管理器 128

启动一区域

过程 247

CICS 控制程序 125

cicscp 命令 125

cicsstart 命令 247

前卷数据库, 使用 161

前卷, 备份和恢复 161

强制异常关闭 135

清除 DCE 260

请求统计 194

区域

定义和管理概述 3

多个的性能成本 180

关闭 133, 137

配置 233

启动 126, 247

启动失败 213

缺省用户标识符 233

委托人 212

区域 (续)

字符串联接 235

cicsdefault 命令 233

CICS_HOST 环境变量 235

区域存储器池, 性能 177, 178

区域定义 (RD)

对于 Encina 服务器, 共享的资源定义 23

更改 33

共享资源 22

关闭程序列表 134

设置 33

在启动时的覆盖 247

SFS 的 24

区域定义 (RD) 属性, 用于

DefaultUserId 206

ESMLoad 214, 220

FileRSLCheck 220, 225

JournalRSLCheck 220, 225

LoadDataNumBlocks 177

ProgramRSLCheck 220, 225

RuntimeProtection 207

RuntimeProtectionLevel 207

TaskSHNumBlocks 177

TemporaryStorageRSLCheck 225

TransactionRSLCheck 220, 225

TransientDataRSLCheck 220, 225

TSLCheck 214

区域定义 (RD) 属性, 使用

StartType 247

区域委托人 212

取消配置 DCE 257

取消 (强制异常) 关闭 135

缺省用户标识符

创建 233

定义 206

[R]

认证策略, DCE 安全性 213

认证的 RPC

性能成本 180

认证服务属性 206

认证为 DCE 委托人 253

认证用户访问 CICS 205

日志

安全性 222

设置 54

日志定义 (JD)

更改 54

配置 54, 55

删除 54

添加 54

日志定义 (JD) (续)

cicsadd 54

cicsdelete 54

cicsupdate 54

日志定义 (JD) 属性, 用于, RSLKey 222

日志管理, 配置 日志定义 (JD) 54

容错, 硬件故障 163

软件启动关闭 135

[S]

设置

程序 39

打印机 56

监控定义 (MD) 60

监控设施 60

临时存储器队列 50

区域定义 (RD) 33

日志 54

事务 37

数据文件 41

瞬时数据队列 52

文件 41

性能监控 60

应用程序服务器 61

映像集 39

TDQ 52

TSQ 50

设置打印机, DBCS 271

设置 DCE 251

时间区 131

时间区环境变量 131

时钟

监控设施 197

系统 131

CICS 131

使用安全性键 221

使用外部安全性管理器

定义接口 226

概述 225

使用 外部安全性管理器

定义 CICS 至 外部安全性管理器 230

启动时定义 ESM 229

使用 CICS 认证 205

使用 cicscp 管理资源

消除区域 35

消除 DCE 配置 255

消除 SFS 36

使用 cicssfdestroy 管理资源, 消除 SFS 249

使用 DCE 认证 206

事件监控点 (EMP) 197

事务

安全性 215, 221, 222

- 事务 (续)
 - 管理, 配置 事务处理定义 (TD) 37
 - 设置 37
 - 性能 197
 - 资源安全性 (resource security) 222
- 事务安全级 (TSL), 定义 221
- 事务安全性 217
- 事务处理定义 (TD)
 - 更改 37
 - 共享的资源定义。 22
 - 配置 37
 - 删除 37
 - 添加 37
 - 在关闭时启动 134
 - cicsadd 37
 - cicsdelete 37
 - cicsupdate 37
- 事务处理定义 (TD) 属性, 用于
 - RSLCheck 220, 222, 225
 - RSLKey 222
- 事务路由选择
 - 动态 108
- 事务路由选择事务
 - 安全性 207
- 事务支持
 - C 和 DB2 90
 - C 和 Oracle 96
 - DB2 和 C 90
 - Informix 版本 7 93
 - Oracle 和 C 96
- 授权访问 CICS
 - 概述 218
 - 来自应用程序的资源 220
 - 使用 CICS 命令的资源定义 219
 - 运行时间区域以外的资源 219
- 授权访问 cics 对象, 使用安全性键 221
- 授权访问 CICS 事务 214, 216
- 授权用户访问 CICS、启动和关闭程序 218
- 输出区域 234
- 输入区域 234
- 数据库
 - 安全性 209
 - 改进性能 185
 - 集成 90, 96
- 数据文件
 - 设置 41
- 属性
 - DefaultUserId 233
 - NameService 238
- 属性, 使用
 - ActivateOnStartup 141
- 属性, 用法
 - 动态 108
 - 协议
 - CICSCLI.INI 80
 - LD 80
 - 协议 (LD) 82
 - CICSPassword(UD) 83
 - DefaultUserId (RD) 206
 - ESMLoad (RD) 214, 220
 - FileRSLCheck (RD) 220, 225
 - InitialTransid (CICSCLI.INI) 80
 - IntrospectInterval 180
 - IsPrinter (WD) 86
 - JournalRSLCheck (RD) 220, 225
 - MaxServer 180
 - MinServer 180
 - NetName
 - CICSCLI.INI 80
 - WD 86
 - NumColumns (WD) 86
 - NumLines(WD) 86
 - Port(CICSCLI.INI) 80
 - ProgramCacheSize 184
 - ProgramRSLCheck (RD) 220, 225
 - RemoteName 108
 - RemoteSysId 108
 - RSLCheck (TD) 222
 - RSLCheck (TD) 225
 - RSLKey 222
 - RSLKeyList (UD) 222
 - RuntimeProtection (RD) 207
 - RuntimeProtectionLevel(RD) 207
 - ServerIdleLimit 180
 - StartType (RD) 247
 - TCPAddress (LD) 82
 - TCPService (LD) 82
 - TemporaryStorageRSLCheck (RD) 225
 - TerminalProtection (WD) 207
 - TransactionRSLCheck (RD) 220, 225
 - TransientDataRSLCheck (RD) 220, 225
 - TSLCheck (RD) 214
 - TSLKey (TD) 214
 - TSLKeyList (UD) 214
 - UpperCaseSecurity(CICSCLI.INI) 80
- 属性, 用于
 - DevType(WD) 58
 - ErrorIsolation 188
 - LoadDataNumBlocks (RD) 177
 - RSLCheck (TD) 220
 - TaskSHNumBlocks (RD) 177
- 瞬时数据定义 (TDD)
 - 更改 52

瞬时数据定义 (TDD) (续)

共享资源 22

配置 52, 53

删除 52

添加 52

cicsadd 52

cicsdelete 52

cicsupdate 52

瞬时数据定义 (TDD) 属性, 用于

RSLKey 222

瞬时数据队列

安全性 222

内分区 52

配置 瞬时数据定义 (TDD) 52

配置 CICS 52

设置 52

顺序, 启动 123

[T]

特征字符串 178

提供的程序

功能附带 114

事务路由选择 114

外部安全性管理器(ESM) 114

性能监控程序用户程序 114

用户转换程序 114

终端自动安装 114

IBM cics 客户机 产品自动安装 114

提供的事务, 安全性 222

天结束统计 194

通信定义 (CD), 共享的资源定义 22

同步点

性能成本 180

一般说明 150

统计

产生 193

非请求 194

间隔 194, 196

检索 194

请求的 194

天结束 194

统计资料

关闭 136

脱机实用程序, cicsmfmt 202

[W]

外部安全性管理器 (ESM)

概述 225

外部安全性管理器(ESM)

使用 114

外部呈现接口应用程序

启用访问 80, 81

外部呈现接口应用程序 (续)

侦听程序定义 (LD) 80, 81

外部磁盘, 在备份和恢复中 163

外部调用接口应用程序

启用访问 80, 81

侦听程序定义 (LD) 80, 81

外部跟踪 202

完全故障, 恢复过程 170

网关, 启动失败 128

委托人

删除 262

添加 266

用安全服务器认证 253

委托人, DCE

为...配置 233, 234

伪会话式事务 61

未到期的启动

非受保护的 240

受保护的 240

未控制关闭 136

位置, cicsprCOBOL 61

文件

安全性 222

设置 41

CICSCLI.INI 80

文件定义 (FD)

更改 41

共享资源 22

配置 41, 43

删除 41

添加 41

cicsadd 41

cicsdelete 41

cicsupdate 41

文件定义 (FD) 属性, 用于, RSLKey 222

文件管理

模式文件定义 (SCD) 45, 46

配置过程 (DB2) 28

配置过程 (SFS) 24

配置过程 (SQL) 28

配置 文件定义 (FD) 41

配置 DB2 45, 47

配置 SFS 46, 48

需求 (SFS) 26

(SQL) 的命名 28

文件管理器

可用性问题 128

启动失败 128

文件控制, 性能成本 180

文件系统, CICS 日志 33

物理可恢复瞬时数据队列 240

物理内存用法, 调整性能 181

[X]

- 系统活动报告, 使用 175
- 系统时钟 131
- 限制
 - 侦听程序处理 81
- 限制, 字段名 (DB2) 29
- 响应时间, 症状和解决方案的摘要 179
- 消息
 - ERZ0070xxE 被记录 128
 - ERZ021004E 被记录 128
 - ERZ0220xxE 被记录 128
 - ERZ0230xxE 被记录 128
 - ERZ0250xxE 被记录 128
 - ERZ061013E 被记录 128
- 协议
 - TCP/IP 238
 - UDP/IP 238
- 协议属性
 - CICSCLI.INI 80
 - LD 80
- 协议属性 (LD), 用法 82
- 信息
 - ERZ045001 223
 - ERZ045013 229
 - ERZ045014 229
 - ERZ087001 229
 - NLS (客户) 73
 - NLS (区域) 34
- 性能
 - 存储器 178
 - 存储器压力的症状 177
 - 低劣性能的症状 177
 - 确定性能约束 177
 - 事务 197
 - 响应时间 179
- 性能成本 179
- 性能监控
 - 设置 60
 - 信息 197
- 性能监控程序用户程序
 - 概述 116
- 性能信息, 监控设施 197
- 性能约束, 确定 177
- 许可权, 组 209

[Y]

- 一般安全性考虑, AIX 205
- 移植, DCE 251
- 已认证 RPC
 - 为 CICS 设置 210

- 应用程序服务器
 - 设置 61
 - 说明 61
- 应用程序, 程序和监控 197
- 硬件故障, 容错 163
- 映像集
 - 设置 39
- 永久数据库
 - 更新 区域定义 (RD) 4
 - 共享 22
- 用户安全性 205, 206
- 用户标识符
 - 备份和恢复 161
- 用户标识符 (SFS)
 - 创建 236
- 用户程序
 - 概述 114
 - 功能附带 114
 - 事务路由选择 114
 - 外部安全性管理器 (ESM) 114
 - 性能监控 114
 - 用户转换程序 114
 - 终端自动安装 114, 115
 - IBM cics 客户机 产品自动安装 114
- 用户出口
 - 动态分布程序连接 111
 - 动态事务路由选择 108
 - 概述 105
 - 命名约定 108
- 用户出口命名约定 108
- 用户定义 (UD)
 - 更改 83
 - 配置 83
 - 删除 83
 - 添加 83
 - cicsadd 83
 - cicsdelete 83
 - cicsupdate 83
- 用户定义 (UD) 属性, 用法
 - CICSPassword 83
- 用户定义 (UD) 属性, 用于
 - RSLKeyList 222
 - TSLKeyList 214
- 用户访问 cics 对象, 安全性 221
- 用户访问 CICS 事务, 安全性 214, 216
- 用户访问 CICS, 安全性 205, 218, 219, 220
- 用户管理, 配置 用户定义 (UD) 83
- 用户监控 197
- 用户转换程序
 - 功能附带 114
 - 事务路由选择 114
- 有数据的 CICS 系统, 需要的备份与恢复 162

- 语言方式文件, Micro Focus COBOL 61
- 远程过程调用, 安全性 210
- 运行时
 - 更新 141
- 运行时间保护级别, 终端 207
- 运行时间错误, DB2 243
- 运行时间数据库
 - 共享 22
 - 说明 4

[Z]

- 侦听程序处理, 限制 81
- 侦听程序定义 (LD)
 - 配置 82
 - 外部呈现接口应用程序 80, 81
 - 外部调用接口应用程序 80, 81
 - cics 客户机 产品 80, 81
 - ECI 应用程序 80, 81
 - EPI 应用程序 80, 81
- 侦听程序定义 (LD) 属性, 用法
 - 协议 80, 82
 - TCPAddress 82
 - TCPService 82
- 侦听程序进程, 客户 80
- 正常关闭的停顿阶段 134
- 正常 (控制的) 关闭 134
- 症状, 性能 177
- 整理语言字段
 - 指定 DBCS/MBCS 环境 269
- 支持非 XA
 - 样本程序 87
- 支持 RDBM XA 的接口 90, 96
- 支持 XA 的关系数据库
 - 与 CICS 集成 90, 96
 - 支持 XA 的接口 90, 96
- 终端
 - 保护级别 207
 - 保护, 安全性 207
 - 管理, 配置 终端定义 (WD) 56
 - 设置 56
 - 运行时间保护级别 207
- 终端定义 (WD)
 - 更改 56
 - 配置 56, 58
 - 删除 56
 - 添加 56
 - cicsadd 56
 - cicsdelete 56
 - cicsupdate 56
- 终端定义 (WD) 属性, 用法
 - NumColumns 86

- 终端定义 (WD) 属性, 用法 (续)
 - NumLines 86
- 终端定义 (WD) 属性, 用法
 - DevType 58
 - IsPrinter 86
 - NetName 86
- 终端定义 (WD) 属性, 用于
 - TerminalProtection 207
- 终端类型
 - 指定 DBCS/MBCS 环境 270
- 终端自动安装
 - 说明 114
 - 用户程序 115
- 终端 类型
 - DBCS/MBCS 环境 270
- 终端 I/O 错误, 恢复
 - 取消 (强制异常) 135
 - 终端错误程序立即关闭 135
 - 终端错误程序强制异常关闭 135
- 终止
 - 立即 135
 - 强制异常 135
 - 取消 (强制异常) 135
 - 软件启动 135
 - 未控制 136
 - 正常 (控制的) 关闭 134
- 主存缓冲区 202
- 主临时存储器。性能成本 179
- 专用任务存储器池, 性能 177, 178
- 转储文件, 打印 136
- 装入, 产品定义 (XAD) 4
- 装运终端, 安全性 207
- 资源
 - 安全性 222
 - 安装永久定义 13
 - 的列表 6
 - 共享 22
 - 可恢复的性能成本 180
 - 组 4
- 资源安全级 (RSL), 定义 221
- 资源安全性 (resource security) 224
 - 定义 221
 - 事务 222
- 资源定义
 - 概述 21
- 子系统项
 - 备份和恢复 160
 - 恢复 160
- 自动安装模型 56
- 自动安装终端定义, 恢复 150
- 自动安装, 终端自动安装用户程序 115
- 自动任务启动 (ATI) 119

自动事务启动, 安全性 215
字段名限制, DB2 29
字段名 (DB2), 限制 29
字符串联接 238

组

许可权 209
资源的 4
cics 26, 218
cicsteld 208
cicsterm 67
cics_admin 46

最终用户

启用 66
使用 66
为...配置 66

[数字]

3270.key 文件
指定 SOSI 270

A

A147 异常终止 116
ActivateOnStartup 属性
使用 141
说明 4
AIX安全性 205
aixterm 模型 56
ATI, 安全性 215

B

BufferPoolSize, 改进性能 182

C

C

DB2 的事务支持 90
DB2 事务支持 90
Oracle 的事务支持 96
Oracle 事务支持 96
C 程序, 高速缓存 184
C 事务, 配置 Sybase 100
CDS 名称文件管理器
可用性问题 128
启动失败 128
CEBR, 安全性 222
CECI, 安全性 222
CEDF
安全性 216, 222
性能成本 179

cell_admin 253
cell_admin_pw 环境变量, 使用 125
cell_admin_pw 环境变量, 用法 9
CICS

关闭 138
后初始程序 117
冷启动 247
启动 117, 125, 127, 247
时钟, 设置 131

cics 客户机 产品

启用访问 80, 81
侦听程序定义 (LD) 80, 81

CICS 客户, 刷新 213

CICS 控制程序

安全性, DCE 206
关闭区域 133
介绍 8
启动 125
区域关闭 133
cicscp 创建 telnet, 用法 79
cicscp 命令 8
cicscp 破坏 telnet, 用法 79
DCE 安全系统 206
telnet 服务器 79

CICS 口令, 更改 83

CICS 配置, 备份和恢复 159

CICS 区域

关闭 137
配置 234
启动 126, 247
添加 234

CICS 认证 205

CICS 日志, 文件系统 33

CICS 提供的事务, 安全性 222

CICS 用户标识符

备份和恢复 161
恢复 161

CICS 资源管理命令, 配置 CICS 区域 234

CICS 子系统项

备份和恢复 160
恢复 160

cics, 组 26, 218

cicscdt 命令, 创建 47

CICSCLI.INI 属性, 用法

协议 80
InitialTransid 80
NetName 80
Port 80
UpperCaseSecurity 80

CICSCLI.INI, 文件 80

CICSCLI.INI. 属性, 用法
协议 80

CICSCLI.INI 属性, 用法 (续)

- InitialTransid 80
- NetName 80
- Port 80
- UpperCaseSecurity 80
- cicscp
 - 安全性, DCE 206
 - 创建 telnet, 用法 79
 - 介绍 8
 - 破坏 telnet, 用法 79
 - DCE 安全系统 206
 - telnet 服务器 79
- cicscp 创建 telnet 73
- cicscp 命令 8, 125
- cicscp start all, 使用 127
- cicscp start dce, 使用 126
- cicscp start region, 使用 126
- cicscp start sfs_server, 使用 126
- cicscp stop all, 使用 133, 138
- cicscp stop dce, 使用 138
- cicscp stop region, 使用 137
- cicscp stop sfs_server, 使用 138
- cicsdb2conf 命令 246
- cicsdb2import 命令 45
- CICSDB2PASSWD 环境变量, 使用 246
- cicsddt 命令, 创建 47
- cicsdefault 命令 233
- cicsexport, 共享的资源定义 22
- cicsimport, 共享的资源定义 22
- cicsmap, 要使用的许可权 219
- cicsmfmt 202
- cicsmkcobol
 - 使用 61
- CICSPassword 属性 (UD), 用法 83
- cicsprCOBOL
 - 位置 61
- cicsprCOBOL 文件, Micro Focus COBOL 61
- CICSREGION 环境变量, 用法 15
- cicsrl 189
- cicssdt 命令, 创建 48
- cicssetupclients, 使用 232
- cicssetupclients, 用法 67
- cicssfnt 标题 197
- cicssfns 命令 239
- cicssfnsconf 命令 239
- cicssfnscreate 命令 237
- cicssfnsdestroy, 使用 249
- cicssfnsimport 命令 46
- cicssfnsshutdown, 使用 248
- cicsstart 命令, 使用 247
- cicsstop 命令 247
- cicstcl, 要使用的许可权 219

- cicsteld
 - 安全性 206
 - 服务器 74
 - 使用 inetd 精灵程序注册 75
 - 说明 73
 - 要使用的许可权 219
 - 在 DBCS/MBCS 环境中使用 270
 - 组 208
 - 作为前台进程运行 78
 - DCE 认证 206
- cicsterm
 - 安全性 206
 - 安全性蕴含式 208, 209
 - 使用 67
 - 要使用的许可权 219
 - 组 67
 - DCE 认证 206
- CICSTERM 环境变量, 用法 58, 70
- cicstermp
 - 过程 85
 - 使用 85
 - 要使用的许可权 219
- cicstermp, DBCS 271
- cicstfmt, 跟踪实用程序 202
- CICSTRACE 环境变量, 使用 247
- cicstran, 要使用的许可权 219
- cics_admin 组 46, 212
- CICS_BROWSE_CACHE 环境变量 188
- CICS_ESM_RETURN_FAIL 229
- CICS_ESM_RETURN_INVALID 229
- CICS_ESM_RETURN_OK 229
- cics_log 组 212
- cics_regions 组
 - 说明 212
 - SFS 作为成员 213
- cics_sfs 组 212
- CICS_SFS_DATA_SIZE 环境变量, 用法 35
- CICS_SFS_DATA_VG 环境变量, 用法 35
- CICS_SFS_LOG_SIZE 环境变量, 用法 35
- CICS_SFS_LOG_VG 环境变量, 用法 35
- CICS_SFS_SIZE 环境变量, 用法 35
- CICS_SFS_VG 环境变量, 用法 35
- cics_users 组 79, 212
- console.nnnnnn
 - 存储器压力信息 177
 - 关闭信息 136
- CPMI, 安全性 222
- create (cicssdt 命令) 48
- CSM1, 安全性 222
- CSMI, 安全性 222

D

DB2

- 安全性 209
- 安全性考虑 28
- 备份和恢复 158
- 表空间 185, 243
- 队列管理 28
- 队列和文件管理 243
- 恢复 158
- 可用性问题 128
- 模式文件定义 (SCD) 45
- 配置 45, 47
- 配置过程 28
- 启动失败 128
- 事务支持, C 90
- 事务支持, COBOL 90
- 为队列管理配置 246
- 文件管理 28
- 运行时间错误 243
- 支持 XA 的接口 90
- 字段名限制 29
- cicsdb2conf 命令 246
- cicsdb2import 命令 45
- cicsddt 命令 47
- DLCHKTIME 245
- LOCKTIMEOUT 245
- TP_MON_NAME 245

DBCS

- 打印机设置 271
- 设置打印机 271
- cicstermp 271

DBCS/MBCS 环境 269

DCE

- 安全性组 212
- 分布式时间服务 131
- 服务器的备份 153
- 关闭 138
- 仅 RPC 257
- 启动 126
- 清除 260
- 区域委托人 212
- 取消配置 257
- 删除委托人 262
- 设置 251
- 添加服务器 265
- 添加客户 265
- 添加委托人 266
- 为...配置客户 232
- 移植 251
- group 167
- SFS 安全性 213

DCE 安全性服务, 为...配置 233, 234

DCE 安全性, 认证策略 213

DCE 认证

- 安全性 206
- cicsteld 206
- cicsterm 206

DCE 委托人

- 安全性 207
- 口令期满 213
- 为...配置 233, 234

DefaultUserId 属性 (RD), 使用 233

DefaultUserId 属性 (RD), 用于 206

DevType 属性 (WD), 用法 58

DISPLAY 环境变量, 用法 10

DLCHKTIME, DB2 245

DOS 客户 80, 81

E

ECI 应用程序

- 启用访问 80, 81
- 侦听程序定义 (LD) 80, 81

Encina, 为...创建逻辑卷 236

ENCINA_FLT_CLIENT_MAX_FDS 环境变量, 用法 27

ENCINA_FLT_SERVER_MAX_FDS 环境变量, 用法 27

ENCINA_FLT_SMS_SERVER_ENABLE 环境变量, 用法 27

EPI 应用程序

- 启用访问 80, 81
- 侦听程序定义 (LD) 80, 81

ERZ045001 信息 223

ERZ045013 信息 229

ERZ045014 信息 229

ERZ087001 信息 229

ESM (外部安全性管理器)

- 概述 225

ESM(外部安全性管理器)

- 使用 114

ESMLoad 属性 (RD), 用于 214, 220

EXEC CICS START PROTECT 请求, 恢复 150

ExtDS 属性, 用法 56

F

FD 属性, 用于, RSLKey 222

FileRSLCheck 属性, 用于 220, 225

FLT 环境变量, 用法 27

G

getcellname, 使用 253, 255

getcellname, 用法 72

group
DCE 167

H

HACMP

示例 273

HACMP (高可用性簇多重处理) 164

hft 模型 56

hft-mb 模型 56

hp 模型 56

hpterm 模型 56

I

IBM cics 客户机

产品自动安装 114

DOS 版 80

Macintosh 版 80

OS/2 版 80

Windows 3.1 版 80

Windows 95 版 80

Windows NT 版 80

IBM CICS 客户机 DOS 版 81

IBM CICS 客户机 Macintosh 版 81

IBM CICS 客户机 OS/2 版 81

IBM CICS 客户机 Windows 版 81

IBM RS/6000, 在备份与恢复中使用 163

ibm3151 模型 56

ibm3161 模型 56

ibm3164 模型 56

In 命令, 连接数据库 22

inetd 73

inetd 精灵程序 75

Informix 版本 7

建立 XA 接口 93

事务支持, C 93

InitialTransid 属性 (CICSCLI.INI), 用法 80

IntrospectInterval 180

iostat 命令 175

IsPrinter 属性 (WD), 用法 86

J

JD 属性, 用于, RSLKey 222

JournalRSLCheck 属性 (RD), 用于 220, 225

L

LANG 环境变量

设置 4

使用 34, 73

LANG 环境变量, 使用 270

LD

配置 82

外部呈现接口应用程序 80, 81

外部调用接口应用程序 80, 81

cics 客户机 产品 80, 81

ECI 应用程序 80, 81

EPI 应用程序 80, 81

LD 属性, 用法

协议 80, 82

TCPAddress 82

TCPService 82

lft 模型 56

lft-md 模型 56

LIBPATH 环境变量

设置 4

使用 229

LoadDataNumBlocks attribute (RD), 用于 177

LOCKTIMEOUT, DB2 245

LUW (逻辑工作单元), 概述 149

M

Macintosh 客户 80, 81

MaxServer 180

MD 198

mft 模型 56

mft-mb 模型 56

Micro Focus COBOL

语言方式文件 61

cicsprCOBOL 文件 61

MinServer 180

MRA, 备份和恢复 162

N

NameService 属性, 使用 238

NetName 属性

CICSCLI.INI 80

WD 86

netstat 命令 175

NLS

客户的 73

区域的 34

信息 34, 73

BMS 40

NLSPATH 环境变量

设置 4

使用 34, 73

nlsString 数据类型

准备 SFS 用于 269

NumColumns 属性 (WD), 用法 86

NumLines 属性 (WD), 用法 86

O

OpThreadPoolSize, 改进性能 182

Oracle

事务支持, C 96

事务支持, COBOL 96

支持 XA 的接口 96

OS/2 客户 80, 81

P

PATH 环境变量, 设置 4

PD 属性, 用于, RSLKey 222

pkt RPC 保护级别 211

pkt_integ RPC 保护级别 211

pkt_privacy RPC 保护级别 211

Port 属性 (CICSCLI.INI), 用法 80

PPC 网关服务器

可用性问题 128

启动失败 128

ProgramRSLCheck 属性 (RD), 用于 220, 225

R

RD

对于 Encina 服务器, 共享的资源定义 23

更改 33

共享资源 22

关闭程序列表 134

设置 33

在启动时的覆盖 247

SFS 的 24

RD 属性, 使用

StartType 247

RD 属性, 用于

DefaultUserId 206

ESMLoad 214, 220

FileRSLCheck 220, 225

JournalRSLCheck 220, 225

LoadDataNumBlocks 177

ProgramRSLCheck 220, 225

RuntimeProtection 207

RuntimeProtectionLevel 207

TaskSHNumBlocks 177

TemporaryStorageRSLCheck 225

TransactionRSLCheck 220, 225

TransientDataRSLCheck 220, 225

TSLCheck 214

RemoteName 属性, 共享的资源定义 22

rmnce all, 使用 253, 255

rmnce all, 用法 72

RPC 侦听程序进程 189

RPC, 为 CICS 设置 210

RSLCheck 属性 (TD), 用于 220, 222, 225

RSLKey 属性, 用于 222

RuntimeProtection 属性 210

RuntimeProtection 属性 (RD), 用于 207

RuntimeProtectionLevel 属性 (RD), 用于 207

S

sar, 用来 175

schedtune 182

ServerIdleLimit 180

SFS

安全性 210

备份和恢复 155

创建联接文件 238

创建逻辑卷 236

创建用户标识符 236

定义 237

端点设置 238

队列管理 24, 26

改进性能 27, 186

关闭 138

关机 248

恢复 150, 155

可用性问题 128

快速本地传送 (FLT) 27, 186

模式文件定义 (SCD) 46

配置 46, 48, 239

配置过程 24

配置 TSQ 50

启动 126, 239

启动失败 128

文件管理 24, 26

需求 26

字符串联接 238

cicssdt 命令 48

cicssfs 命令 239

cicssfsconf 命令 239

cicssfscreate 命令 237

cicssfsimport 命令 46

TSQ 配置 50

SFS 的 RD 24

SOSI 切换键

指定 DBCS/MBCS 环境 270

SOSI 属性

指定 DBCS/MBCS 环境 270

SOSI 属性, 用法 56

SQL

队列管理 28

命名 28

配置过程 28

嵌入调用 87

文件管理 28

StartType attribute (RD), 使用 247

SwitchLoadFile 属性

用 DB2 28

Sybase

C 事务支持的配置 100

Symrecs 文件 136

T

TaskSHNumBlocks attribute (RD), 用于 177

TCP 例子, 端点 238

TCPAddress 属性 (LD), 用法 82

TCPService 属性 (LD), 用法 82

TCP/IP

协议 238

TD

更改 37

配置 37

删除 37

添加 37

在关闭时启动 134

cicsadd 37

cicsdelete 37

cicsupdate 37

TD 属性, 用于

RSLCheck 220, 222, 225

RSLKey 222

TSLKey 214

TDD 属性, 用于

RSLKey 222

TDQ

内分区 52

配置 瞬时数据定义 (TDD) 52

配置 CICS 52

设置 52

telnet 服务器

安全性 78

配置 73

启动 73

消除 73

cicscp 73, 79

inetd 73

telnet 客户

使用 73

为...配置 73

security for 208

telnet 客户接口

方法, 建立 74

作为前台进程启动 78

TemporaryStorageRSLCheck 属性 (RD), 用于 225

TERM 环境变量, 用法 58, 68, 70

TerminalProtection 属性 (WD), 用于 207

TP_MON_NAME, DB2 245

TransactionRSLCheck 属性, 用于 220, 225

TransientDataRSLCheck 属性 (RD), 用于 220, 225

TSD 属性, 用于, RSLKey 222

TSLCheck 属性 (RD), 用于 214

TSLKey 属性 (RD), 用于 214

TSLKeyList 属性 (RD), 用于 214

TSQ

配置的 50

设置 50

U

UD

更改 83

配置 83

删除 83

添加 83

cicsadd 83

cicsdelete 83

cicsupdate 83

UD 属性, 用法

CICSPassword 83

UD 属性, 用于

RSLKeyList 222

TSLKeyList 214

UDP 例子, 端点 238

UDP/IP, 协议 238

UpperCaseSecurity 属性 (CICSCLI.INI), 用法 80

UserExitNumber 属性 106

V

vmstat 命令 175

vt100 模型 56

W

WD 属性, 用法

DevType 58

IsPrinter 86

NetName 86

NumColumns 86

NumLines 86

WD 属性, 用于

TerminalProtection 207

Windows 客户 80, 81

X

XA 接口

建立 Informix 版本 7 93

为 DB2 启用 243

- XA 接口 (续)
 - 支持 DB2 90
 - 支持 Oracle 96
 - DB2 90, 243
 - Informix 版本 7 93
 - Oracle 96
- XA 生效关系数据库
 - 安全性 209
- XAD, 装入 4
- xa_open 调用, 用于 209
- xterm 模型 56
- X/Open DTP 实现, 安全性考虑 209
- X/Open XA 接口 87
- %H 简写 7
- %R 简写 7
- %S 简写 7

读者意见表

IBM TXSeries AIX 版
CICS 管理指南
版本 4.2

SC09-3948-00

姓名

地址

单位及部门

电话号码



请沿此线
撕下或折起

折起并封口

请勿使用钉书机

折起并封口

在此
贴上
邮票

Transarc 公司
ATTN: Documentation Group
The Gulf Tower
707 Grant Street
Pittsburgh, PA
15219-1900

折起并封口

请勿使用钉书机

折起并封口

请沿此线
撕下或折起



程序编号: 5697-D17

5697-D18

5697-D19

5697-D20

5697-D21

5697-D22

Printed in China

SC09-3948-00

