

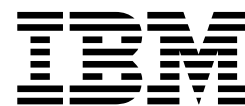
CICS® Transaction Gateway



Gateway 管理 HP-UX 版

版本 4.0

CICS® Transaction Gateway



Gateway 管理 HP-UX 版

版本 4.0

注意！

在使用这些信息和它支持的产品之前，请务必阅读第95页的『附录C. 声明』中的一般信息。

第一版（2001年6月）

本修订版适用于 CICS® Transaction Gateway HP-UX 版本号 4.0，程序号 5724-A75。它还适用于所有后续版本、发行版和修正，直到在新修订版中另有声明为止。

这是基于早期版本的用于其它操作系统的“管理”书籍的新书。页面左边的垂直线表示本修订版中的新资料。

© Copyright International Business Machines Corporation 1996, 2001. All rights reserved.

目录

图	vii
表	ix
更改摘要	xi
关于本书	xiii
本书针对的读者	xiii
本书使用的约定和术语	xiii
先决条件和相关信息	xiii
如何发送意见	xiv
从 IBM 获得书籍	xiv
第1章 概述	1
CICS Transaction Gateway 提供的内容	1
Java™ 技术	2
Java 语言	2
Java 小应用程序	3
Java 小服务程序	3
Java 应用程序	4
JavaBean™	4
防火墙	4
Web 浏览器和网络计算机	5
Web 服务器	5
CICS Transaction Gateway 如何访问 CICS	5
CICS Transaction Gateway: 线程模型	6
外部访问接口 (EPI, ECI, ESI)	8
外部显示接口 (EPI)	9
外部调用接口 (ECI)	9
外部安全性接口 (ESI)	9
网络安全性	10
安全套接字层 (SSL)	10
HTTPS	10
密钥和证书	11
安全性出口	11
CICS Transaction Gateway 和对象请求代理	11
其它功能	11
欧元支持	12
可访问性	12
第2章 安装前的规划	13
硬件需求	13
支持的软件	13
第3章 安装	17
安装 “CICS® Transaction Gateway HP-UX 版”	17
国家语言支持	17
从远程系统使用 X-Windows	18
卸载 CICS Transaction Gateway	18

第4章 配置	19
配置 CICS Transaction Gateway 的编程环境	19
在 HP-UX 上设置 CLASSPATH	19
使用配置工具	19
配置工具界面	20
配置 CICS Transaction Gateway 设置	22
配置客户机设置	26
配置服务器设置	28
跟踪设置	31
配置转换工具	33
使用转换工具	33
编辑配置文件	34
GATEWAY 部分	34
CLIENT 部分	35
SERVER 部分	35
DRIVER 部分	35
重命名配置和映射文件	36
配置客户机键盘映射	36
键盘映射文件语法	36
键盘映射文件	37
定制屏幕颜色	40
颜色映射语法	41
颜色映射文件	42
准备使用本地 CICS Transaction Gateway 支持	44
第5章 安全性	45
概述	45
什么是加密?	46
数字签名和数字证书	46
获得数字证书	47
KeyRing	47
SSL 与认证	48
HTTPS	49
ctgikey 工具	49
将 iKeyman 分发到客户机工作站	50
使用在外部签署的证书 (SSLight)	50
配置 SSL 服务器	51
配置 SSL 客户机	53
使用自签证书 (SSLight)	54
配置 SSL 服务器	54
配置 SSL 客户机	55
限制对服务器 KeyRing 的访问	57
将 CICS Transaction Gateway 配置为 SSL 和 HTTPS	57
指定客户机 KeyRing	57
第6章 CICS Transaction Gateway 操作	59
启动 Gateway	59
用预置选项启动 Gateway	59
用用户指定的选项启动 Gateway	59
停止 Gateway	61
第7章 CICS Transaction Gateway 终端小服务程序	63

什么是“CICS Transaction Gateway 终端小服务程序”？	63
安装和配置“终端小服务程序”	64
配置 Web 服务器的 CLASSPATH 和 PATH 设置	65
将“终端小服务程序”添加到 Web 服务器的配置中	65
配置小服务程序初始化参数	65
考虑其它配置选项	68
装入“终端小服务程序”	69
使用“终端小服务程序”	69
连接到 CICS 并启动事务	69
调用“终端小服务程序”	69
下一步将发生什么？	71
显示屏幕和字段	71
将屏幕发送回 CICS	72
设置 AID	72
断开连接	73
特性和参数参考	73
小服务程序配置特性	73
页面映射特性	75
请求参数	75
可显示特性	76
CICS [®] Transaction Server OS/390 [®] 版 Web 界面	77
第8章 问题确定和问题解决	79
初步检查	79
下一步操作	79
消息	79
跟踪	80
程序支持	82
诊断常见问题	82
安装问题	82
Java 异常	82
“网关”作为 Windows NT [®] 服务运行时的问题	83
java.lang.OutOfMemory 异常	83
如果进程锁定	84
代码页问题	84
SSL 异常	84
JDK [™] AppletViewer 问题	85
附录A. “CICS 通用客户机”数据转换	87
受支持的转换	87
附录B. CICS Transaction Gateway 和“CICS 通用客户机”书籍库	91
CICS Transaction Gateway 书籍	91
“CICS 通用客户机”书籍	92
CICS 系列出版物	92
书籍文件名	93
样本配置文档	93
其它出版物	93
查看联机文档	94
查看 PDF 书籍	94
附录C. 声明	95
商标	96

索引 97



1.	CICS Transaction Gateway	1
2.	用于 TCP/IP 和 SSL 的 CICS Transaction Gateway 线程模型	7
3.	用于 HTTP/HTTPS 的 CICS Transaction Gateway 线程模型	8
4.	配置工具	21
5.	“CICS 通用客户机” 样本键盘映射文件	38
6.	样本颜色映射文件	43
7.	带有服务器认证的 SSL 握手	49
8.	CICS Transaction Gateway 使用由 URL 调用的“终端小服务程序”	64
9.	样本 Java 堆栈跟踪	81
10.	样本 Java 堆栈转储	82
11.	作为 Windows NT® 服务运行的“网关”的跟踪	83

表

1.	CICS Transaction Gateway 平台上的线程限制	8
2.	支持的产品	13
3.	可映射的“CICS 通用客户机”键	37
4.	小服务程序初始化参数	66
5.	CICS Transaction Gateway 和“CICS 通用客户机”书籍以及文件名	93

更改摘要

以下是对 CICS Transaction Gateway 版本 4.0 所做的功能更改:

- 支持 TCP62。
- 已经除去了下列各项的支持:
 - Cobol
 - PL/I
 - REXX
 - CICSTELED

这是基于早期版本的用于其它操作系统的“管理”书籍的新书。页面左边的垂直线表示本修订版中的新资料。

关于本书

本书包含下列各章:

- 第1章介绍了 CICS Transaction Gateway 并概述了使用它的优点和它提供的功能。
- 第2章讨论规划问题, 包括 CICS Transaction Gateway 的硬件和软件需求。
- 第3章描述了如何安装 CICS Transaction Gateway。
- 第4章描述了如何配置 CICS Transaction Gateway。
- 第5章描述了如何为 CICS Transaction Gateway 中的 SSL 和 HTTPS 协议设置安全性。
- 第6章描述了如何操作 CICS Transaction Gateway, 包括启动和停止 CICS Transaction Gateway。
- 第7章介绍了与 CICS Transaction Gateway 一起提供的终端小服务程序。
- 第8章描述了 CICS Transaction Gateway 的问题确定。
- 附录B描述了如何查看 CICS Transaction Gateway 和“CICS 通用客户机”库中的联机信息, 以及如何打印和订购书籍。

本书针对的读者

本书供那些涉及规划、安装、定制或操作 CICS Transaction Gateway 的用户使用。

假设您已熟悉运行 CICS Transaction Gateway 的操作系统。

理解因特网术语也是有用的。

本书使用的约定和术语

在本书中, 术语“CICS 通用客户机”指 CICS Transaction Gateway 的“客户机”组件。

『System/390[®] 上的 CICS』用于指下列 CICS 服务器产品:

- CICS[®] MVS/ESA[™] 版
- CICS[®] Transaction Server OS/390[®] 版
- CICS[®] Transaction Server VSE/ESA[™] 版
- CICS/VSE[®]

CICS Transaction Gateway 运行在 Windows NT[®] 和 Windows 2000 上。本书中对 Windows[®] 的引用是指 Windows NT 和 Windows 2000, 除非指定特定版本的 Windows。

文本中对 OS/390 的引用包括 OS/390 和 z/OS 操作系统。

先决条件和相关信息

有关本产品的可用书籍信息, 请参考第91页的『附录B. CICS Transaction Gateway 和“CICS 通用客户机”书籍库』。本章还提供了如何查看和打印软拷贝书籍及如何向 IBM 订购印刷副本的详细信息。

如何发送意见

您的反馈对于帮助我们提供最精确和高质量的信息非常重要。如果您对本书或任何其它 CICS 文档有任何意见:

- 请访问我们的 Web 站点:

www.ibm.com/software/ts/cics/

并跟着**书库**链接到我们的反馈表。

在此, 您将看到反馈页面, 您可以在其中输入并提交您的意见。

- 通过电子邮件将意见发送至 idrpf@hursley.ibm.com
- 将意见传真至:

+44 1962 842327 (如果您在英国之外)
01962 842327 (如果您在英国)

- 将意见邮寄至:

User Technologies
Mail Point 095
IBM United Kingdom Limited
Hursley Park
Winchester
Hampshire
SO21 2JN
United Kingdom

不管您使用何种方式, 请确保包含:

- 书名
- 书的表单号
- 产品的版本 (如果适用)
- 您意见所指的文本的特定位置, 例如: 页数或表格号。

当您把信息发送给 IBM 之后, 就授予了 IBM 一项非专用权, 允许 IBM 以它认为合适的任何方式使用或分发这些信息, 而不需要对您负任何责任。

从 IBM 获得书籍

有关可以下载的书籍信息, 请访问我们的 Web 站点:

www.ibm.com/software/ts/cics/

并遵循**书库**链接。

可以订购硬拷贝书籍:

- 通过 IBM 代理或提供本地服务的 IBM 分部。
- 在美国拨打电话 1-800-879-2755。
- 从 Web 站点:

www.ibm.com/shop/publications/order

第1章 概述

IBM CICS Transaction Gateway 通过在一系列配置中使用标准因特网协议，提供了从 Web 浏览器和网络计算机对在 CICS Transaction Server 和 TXSeries™ 服务器上运行的业务关键型应用的安全、方便的访问。

CICS Transaction Gateway 是对 Web 服务器强健和可伸缩的补充，正因为如此，它可以作为 IBM WebSphere™ 的 (Java™ 小服务程序的运行时环境) 电子商务连接器实现。

提供了 Windows NT®、AIX®、Linux、Solaris、HP-UX 和 OS/390® 等平台版本 CICS Transaction Gateway。

图1显示了 Web 客户机是如何访问 CICS 程序和数据的。请注意，显示的 CICS Transaction Gateway 是安装在 Web 服务器机器上的。这只在您将 CICS Transaction Gateway 和 Java 小应用程序一起使用时才必需。

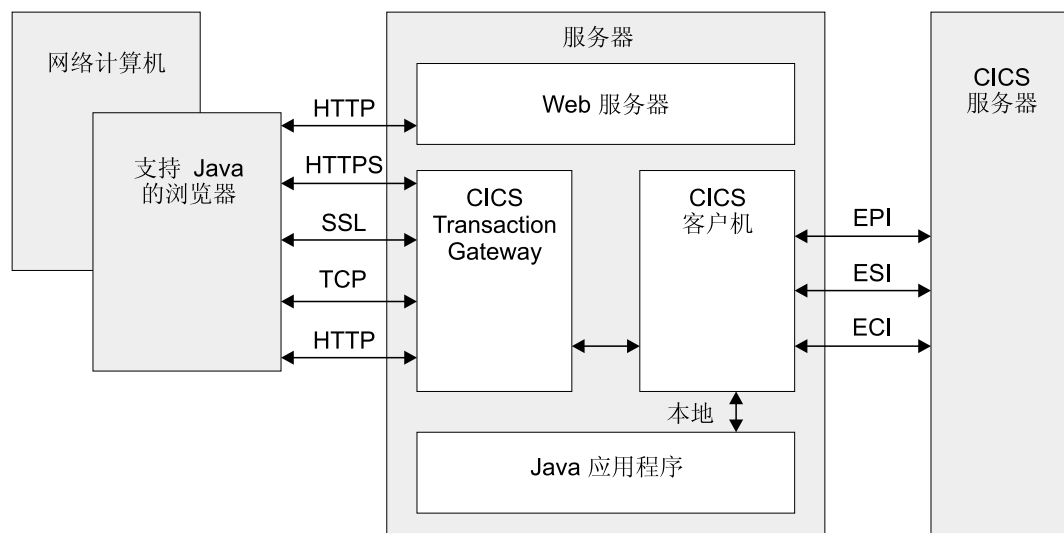


图1. CICS Transaction Gateway

与 CICS Transaction Gateway 进行的通信基于以下协议:

- TCP/IP 套接字
- 超文本传送协议 (HTTP)
- 安全套接字层 (SSL)
- SSL 上的 HTTP (HTTPS)

TCP/IP 套接字和 SSL 提供了内部网应用程序通信的一种有效方法。在有防火墙存在的地方，HTTP 及其安全备用 HTTPS 是因特网应用程序的有效通信协议 (请参阅第10页的『网络安全性』)。

CICS Transaction Gateway 提供的内容

CICS Transaction Gateway 提供了以下内容:

概述

1. 一个通常位于 Web 服务器工作站上（出于安全性原因）的 **Java 网关应用程序**。它通过“CICS 通用客户机”提供的 ECI（外部调用接口）、EPI（外部显示接口）或 ESI（外部安全性接口）接口与在 CICS 服务器中运行的 CICS 应用程序进行通信。该 Java 应用程序以前在 IBM CICS Gateway Java 版 中提供。
2. 一个提供 ECI、EPI 和 ESI 接口，以及终端仿真功能的“**CICS 通用客户机**”。**ECI 接口**可以让非“CICS 客户机”应用程序同步或异步地将 CICS 程序作为子例程调用。**EPI 接口**可以让非“CICS 客户机”应用程序充当逻辑 3270 终端，因此可以控制 CICS 3270 应用程序。**ESI 接口**可以让非“CICS 客户机”应用程序调用由高级程序间通信（APPC）密码到期管理（PEM）提供的服务。根据不同的操作系统，“CICS 通用客户机”允许使用 TCP/IP、APPC 和其它协议与“CICS 服务器”进行通信。
3. 一个 **CICS Java 类库**，包括一些提供应用程序编程接口（API）的类，这些类用于 Java 网关应用程序和 Java 应用程序（小应用程序或小服务程序）之间进行的通信。**JavaGateway** 类用于建立与“网关”进程的通信，它使用 Java 套接字协议。**ECIRequest**、**EPIRequest** 和 **ESIRequest** 这三类分别用于指定流向网关的 ECI、EPI 和 ESI 调用。
4. 一个**终端小服务程序**，允许您将 Web 浏览器作为在任何 CICS 服务器上运行的 3270 CICS 应用程序的仿真器使用。终端小服务程序可以和 Web 服务器或小服务程序引擎一起使用，它们提供的支持等价于 Java Servlet Development Kit (JSDK) 版本 1.1 或更新版本。这是早先由“IBM CICS 客户机”版本 2 中 CICS Internet Gateway 提供的增强版本的功能。终端小服务程序提供了使用 EPI 编程接口的一种替代手段。
5. 一组 **Java EPI bean**，用于在无需任何编程的情况下为现有 CICS 3270 应用程序创建 Java 前端。

CICS Transaction Gateway 可以并行地管理许多到已连接 Web 浏览器的通信链路，而且可以控制与多个 CICS 服务器系统的异步对话。CICS Transaction Gateway 的多线程体系结构可以使单个“网关”支持多个并行连接的用户。

Java™ 技术

本节讨论 Java 语言，包括可以开发的程序类型以及安全性相关事项。

Java 语言

Java 语言可以用于构造 Java 小服务程序、Java 小应用程序和 Java 应用程序。

Java 是一种解释型面向对象语言，它与 C++ 类似，后者用于同时以源码和对象的形式构建与平台无关的程序。它具有跨越 Web 浏览器和 Web 服务器的一种独特的操作特征，这种特征能够在因特网应用程序中启用新的和功效强大的功能。

为实现平台无关性，Java 语言不允许任何与平台相关的操作，并将一些 C++ 功能排除在外，例如预处理器、运算符重载、多继承和指针等。所有 Java 编程都封装在类中，并且“Java 开发工具箱 (JDK)”包括了一些对确保平台无关性非常特殊的特殊类，包括 GUI 功能、输入/输出功能和网络通信。

Java 编译器产生与机器无关的一种中间字节码格式。然后，它在执行时由 Java 解释器处理。解释器还在执行时检查字节码来确保它的有效性以及对机器环境来说是否安全。因为 Java 解释器提供了隔离，所有它有时也指“Java 虚拟机 (JVM)”。

Java 小应用程序

Java 小应用程序是一种下载到支持 Java 的 Web 浏览器或网络计算机，并在其上运行的小型应用程序。Java 小应用程序通常执行客户机代码在客户机 / 服务器体系结构中执行这类操作。它编辑输入、控制屏幕，并向执行数据或数据库操作的服务器发送事务。

小应用程序是使用 `<applet>` HTML 标记开始的；它为小应用程序提供了控制，并指定由小应用程序所使用的显示区域。当支持 Java 的服务器在下载页面时遇到该标记，它也下载小应用程序字节码，使用的方式和下载由 HTML `image` 标记引用的图像一样。然后，支持 Java 的浏览器解释并执行小应用程序字节码。小应用程序会编辑屏幕输入、生成屏幕输出，并与从中下载它的计算机进行通信。多个小应用程序可以并行执行。

小应用程序处理的一个示例：小应用程序与一个服务器保持不间断的通信来接收股票交易信息，它会更新屏幕上窗口中的信息。

下载小应用程序不应该对终端用户的响应时间有显著的性能影响，因为小应用程序一般都不是很大。小应用程序甚至还可以通过消除与 Web 服务器的迭代来提高浏览器的整体性能。而且，如同图像可以在 Web 浏览器中被高速缓存一样，小应用程序也可以被高速缓存，这就将小应用程序的下载频率减到最小。

Java 小服务程序

Java 小服务程序是在 Web 服务器机器上运行的小型 Java 应用程序，与 Java 小应用程序不同，后者是下载到 Web 浏览器的。

Java 小服务程序现已非常普遍，它是 CGI（公共网关接口）程序的替代。Java 小服务程序胜过 CGI 应用程序的优势在于它们可以更快速地执行，因为它们是作为守护程序中的线程调用的，这意味着它们在内存中是持久的，可以完成多个请求。

运行小服务程序有三种方法：

通过 URL（统一资源定位器）调用小服务程序

小服务程序的 URL 通常具有以下格式：

```
http://machine-name:port/servlet/servlet-name
```

在您不需要用户输入任何信息时使用这种方法。请求参数可以以查询字符串的格式在 URL 中编码。

使用表调用小服务程序

这种方法需要在 Web 页面中创建一个 HTML 表单：

```
<FORM METHOD="GET" ACTION="/servlet/servlet-name">  
  attributes  
</FORM>
```

在您需要用户输入信息时使用这种方法。

使用服务器引用调用小服务程序

服务器引用是在将 Web 页面发送给用户之前由 Web 服务器处理的。以 HTML 源码格式表示的服务器引用如下所示：

概述

```
<SERVLET NAME="TerminalServlet" >
<PARAM NAME="request" VALUE="send">
<PARAM NAME="transaction" VALUE="CECI">
<PARAM NAME="display" VALUE="none">
</SERVLET>
```

Java 应用程序

Java 应用程序是一种在计算机上本地执行的程序。除小应用程序具有的性能外，它还具有与平台无关的能力。它可以访问本地文件、创建和接受常规网络连接，以及调用特定于机器的库中的本机 C 或 C++ 函数。

JavaBean™

JavaBeans API 由 Sun 公司开发，允许您使用 Java 编写组件软件。组件是自包含、可重用软件单元，您可以使用视觉应用程序构建器工具（例如 VisualAge® for Java）来以可视方式将它们构成小应用程序或小服务程序。任何与 JDK 版本 1.1 兼容的浏览器或工具都支持 JavaBean。

JavaBean 组件称为 bean。大多数构建器工具都允许您在调色板或工具箱中维护 bean。可以从工具箱中选择某个特别的 bean、将它放在表单上、修改其外观和行为，然后定义它如何与其它 bean 交互。可以将您选中的 bean 和其它 bean 一起构成小应用程序、小服务程序或新的 bean，而所有这些都可以在不编写任何代码的情况下做到。有关 JavaBean 的详细信息，请参阅 Sun Web 站点 (www.java.sun.com)。

CICS Transaction Gateway 提供了基于高级 EPI 接口的 **EPI Java bean**。这些 bean 可以让您在无需任何编程的情况下轻松地创建访问现有 CICS 3270 应用程序数据的 Java 程序（小应用程序和应用程序）。使用 bean 构成器工具（例如 VisualAge for Java），可以迅速而轻松地创建可以与 CICS 连接、运行事务、显示 3270 屏幕数据以及将用户输入发送回 CICS 服务器的新 Java 前端。有关详细信息，请参阅 *CICS Transaction Gateway: Gateway Programming* 一书。

防火墙

目前，在使用 Java 小应用程序通信设计方面的一个考虑因素是防火墙的影响。防火墙是一个用在软件配置上的术语，它可以防止可信网络和不可信网络之间出现未授权的通信量。适当地放置防火墙可以保护公司资产免遭外界侵犯，但是防火墙同样也会限制合法通信。防火墙功能可以通过以下两种方法来实现：

1. 外部用户对服务器的一般访问能力 - 入站限制
2. 防火墙内的最终用户在防火墙外执行特定的网络功能的能力 - 出站限制。

CICS Transaction Gateway 配置非常合适于避免第一点所述的问题，因为“网关”处理器可以放置在防火墙外，并通过防火墙和 CICS 服务器相连。但最终用户不得不出站防火墙将是一个问题。一个大公司可能会使用防火墙来限制某些可用的连接类型及协议。

在内部网（因特网的本地实现）上能很好地使用 Java，因为防火墙通常不是考虑因素。但是，当为公司外部的最终用户设计因特网应用程序时，您应当确定最终用户防火墙是否是一个影响实现的因素。如果是，那么有必要为这些用户进行一些替代处理，例如将 Java 代码作为 Web 服务器上的 Java 小服务程序执行。您还应该考虑使用 CICS Transaction Gateway 所支持的 HTTP 和 HTTPS 协议，请参阅第10页的『安全套接字层 (SSL)』和第10页的『HTTPS』。

Web 浏览器和网络计算机

CICS Transaction Gateway 需要一个支持 Java 的 Web 浏览器，即支持 JDK 版本 1.1，例如 Windows 和 AIX 上的 Netscape Communicator 4.5.1 或更新版本。有关详细信息，请参考第13页的『支持的软件』。

Web 浏览器使用“超文本传送协议 (HTTP)”与 Web 服务器进行通信，请求下载“超文本标记语言 (HTML)”页面。这些 HTML 页面可以包括对 Java 小应用程序（请参阅第3页的『Java 小应用程序』）或小服务程序的调用；多个小应用程序可以并行运行。

您可能会发现，对于相同的信息，每种浏览器显示的方式可能不同。

网络计算机是用于因特网用户的一种低成本计算机，它和 Web 浏览器执行相同的功能。

Web 服务器

Web 服务器是一种软件程序，它对由 Web 浏览器生成的信息请求做出响应。接收到来自浏览器的请求时，Web 服务器处理请求来确定将要采取的操作。

- 返回请求的文档。
- 拒绝请求。
- 由外部应用程序传递请求，作进一步的处理。例如，请求可以是对数据库发出以执行搜索请求的，也可以是对更为动态的信息传递格式（例如 Lotus Domino™）发出的。

Web 服务器和外部应用程序之间的通信是透明的，您只需要知道 Web 服务器的 URL 以将请求指向它即可。另外，所有 Web 服务器都可以并行地处理来自多个浏览器的请求。

也可以配置专门的服务器来对一组受限用户限制访问，或者为购买商品或服务提供安全性。

几乎每种平台都有 Web 服务器存在，而且它们可以从众多供应商处获得。有关 CICS Transaction Gateway 支持的 Web 服务器的信息，请参阅第13页的『支持的软件』。

CICS Transaction Gateway 如何访问 CICS

这一节将描述 CICS Transaction Gateway 是如何允许访问 CICS 程序和数据的。

对于 **Java 小应用程序**，访问是通过如下方式完成的：

1. Web 浏览器或网络计算机通过使用 HTTP（超文本传送协议）协议请求来自 Web 服务器的 HTML 页面。
2. Web 服务器返回包含标识 Java 小应用程序标记的 HTML 页面。
3. 浏览器开始请求来自 Web 服务器的有关 Java 类。
4. Web 服务器返回 Java 类，包括请求的 CICS Transaction Gateway 类。
5. 返回类以后，Java 小应用程序启动。
6. Java 小应用程序创建一个 **JavaGateway** 对象来与 CICS Transaction Gateway 连接。这会在浏览器和长时间运行的“网关”进程之间使用 Java 的套接字协议建立通信。

7. Java 小应用程序创建分别包含 ECI、EPI 或 ESI 调用的 **ECIRequest**、**EPIRequest** 或 **ESIRequest** 对象，并使用 **JavaGateway.flow** 方法将它发送给“网关”。
8. “网关”接收到请求，将它解包，然后对“CICS 通用客户机”执行相应的 ECI、EPI 或 ESI 调用。
9. “CICS 通用客户机”将 ECI、EPI 或 ESI 调用传递给预期的 CICS 服务器。
10. CICS 服务器处理调用，包括在需要时执行的用户标识和密码的验证，然后将控制和用户数据传递给 CICS 应用程序。
11. 当 CICS 应用程序完成处理之后，它将控制和数据返回给 CICS。然后，CICS 将控制和数据返回给“客户机”，“客户机”再将它返回给“网关”。
12. “网关”将这些结果进行包装，然后将它们返回给在 Web 浏览器上运行的 Java 小应用程序。

对于 **Java 小服务程序**，访问是通过如下方式完成的：

1. Web 服务器装入并初始化小服务程序。这是在 Web 服务器启动时，或者第一次对小服务程序发出请求时执行的。这时，小服务程序创建一个 **JavaGateway** 对象来与 CICS Transaction Gateway 连接。
2. 当小服务程序由相应的 HTTP 请求调用时，Web 服务器使用请求的详细信息调用其 **Service** 方法。
3. Java 小服务程序创建分别包含 ECI、EPI 或 ESI 调用的 **ECIRequest**、**EPIRequest** 或 **ESIRequest** 对象，并使用 **JavaGateway.flow** 方法将它发送给“网关”。
4. “网关”接收到请求，将它解包，然后对“CICS 通用客户机”执行相应的 ECI、EPI 和 ESI 调用
5. “CICS 通用客户机”将 ECI、EPI 和 ESI 调用传递给预期的 CICS 服务器。
6. CICS 服务器处理调用，包括在需要时执行的用户标识和密码的验证，然后将控制和用户数据传递给 CICS 应用程序。
7. 当 CICS 应用程序完成处理之后，它将控制和数据返回给 CICS。然后，CICS 将控制和数据返回给“客户机”，“客户机”再将它返回给“网关”。
8. “网关”将这些结果进行包装，然后将它们返回给 Java 小服务程序。
9. 当小服务程序接收到对 CICS Transaction Gateway 发出的请求的结果时，它生成将返回给 Web 浏览器的 HTTP 响应。

注：不是所有 ECI、EPI 和 ESI 调用都分别受 **ECIRequest**、**EPIRequest** 和 **ESIRequest** 支持。有关详细信息，请参考 *CICS Transaction Gateway: Gateway Programming*。

CICS Transaction Gateway: 线程模型

CICS Transaction Gateway 为处理网络连接以及对 Java 客户机的请求和应答分配线程提供了多线程模型。多线程模型涉及以下组件：

ConnectionManagers

ConnectionManager 管理所有来自特定 Java 客户机（小应用程序或应用程序）的连接。接收到请求时，它从可用的 Worker 线程池中分配一个 Worker 线程来执行请求。初始 **ConnectionManager** 资源池的大小是由初始“连接管理器”线程数配置设置定义的。可以使用最大“连接管理器”线程数设置指定

ConnectionManager 池的最大大小，请参阅第19页的『使用配置工具』。也可以在启动 CICS Transaction Gateway 时指定这些限制，请参阅第59页的『用用户指定的选项启动 Gateway』。

Worker

Worker 是实际执行 Java 客户机请求的对象。每个 Worker 对象都有其自己的线程，这些线程是在要执行某些工作时激活的。当 Worker 线程完成后，它返回到可用 Worker 线程池中。和 ConnectionManager 一样，Worker 资源池有一个使用初始“工作程序”线程数设置指定的初始大小。可以使用最大“工作程序”线程数设置指定 Worker 池的最大大小，请参阅第19页的『使用配置工具』。也可以在启动 CICS Transaction Gateway 时指定这些限制，请参阅第59页的『用用户指定的选项启动 Gateway』。

在下图中说明了线程模型：

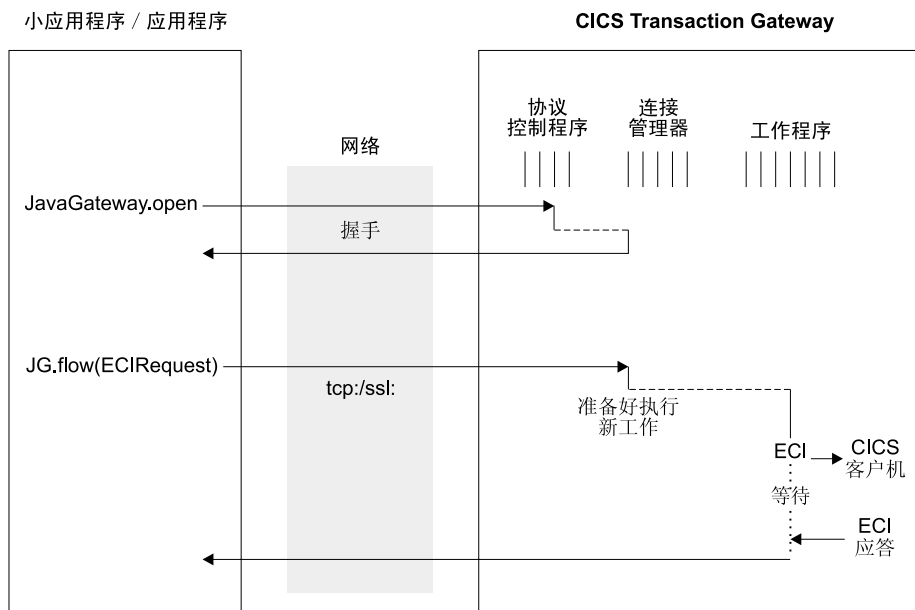


图 2. 用于 TCP/IP 和 SSL 的 CICS Transaction Gateway 线程模型。这些协议有一个持久套接字。

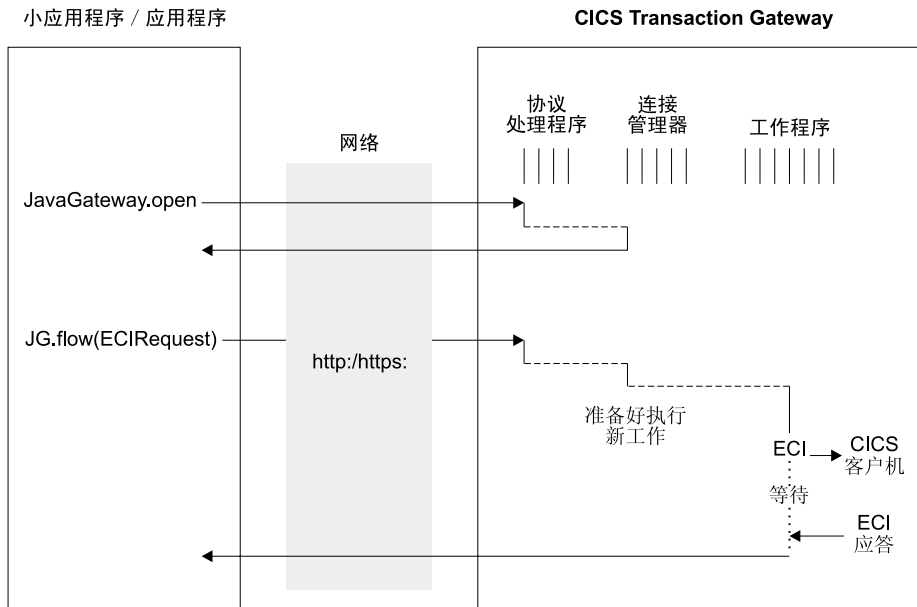


图 3. 用于 HTTP/HTTPS 的 CICS Transaction Gateway 线程模型. 这些协议没有持久套接字。

表1显示了在各种平台上设置“连接管理器”和 Worker 线程数时应该考虑的线程限制。

表 1. CICS Transaction Gateway 平台上的线程限制

操作系统	最大线程数系统范围的限制	线程数的进程限制
HP-UX	无限制 (30 000 个内核线程)	30 000 (请参考 SAM 实用程序中的“可配置内核参数”)

有关 Java 为线程分配内存所使用的方法的详细信息，请参考 *CICS Transaction Gateway: Gateway Programming* 一书中的『Performance issues』。

外部访问接口 (EPI, ECI, ESI)

外部访问接口允许非 CICS 应用程序通过启动 CICS 事务或调用 CICS 程序来访问和更新 CICS 资源。当结合 CICS 通信设施一起使用时，它们可以让非 CICS 程序访问和更新任何 CICS 系统上的资源。它支持例如为 CICS 应用程序开发图形用户界面 (GUI) 前端，以及允许对 CICS 系统和非 CICS 系统进行集成这样一些活动。

外部显示接口 (EPI) 允许您为现有的 CICS 系统或新应用程序开发 GUI。这对于无需更改而为现有 CICS 事务开发新的 GUI 前端特别有用。应用程序可以使用 EPI 和一个 CICS 事务进行通信，而且可以利用客户机系统的显示设施来与最终用户进行通信。

CICS 和非 CICS 系统的集成通常涉及到在非 CICS 系统的程序和一个 CICS 程序之间传递用户定义数据，并可以对其使用外部调用接口 (ECI)。

在任何一种情况下，选择 EPI 还是 ECI 不总是一目了然的，因为这两种接口都可以用来在非 CICS 应用程序和 CICS 程序之间传递数据。不过，两种情况中的机制并不相同：对于 EPI，使用 3270 数据流；对于 ECI，使用 COMMAREA 中应用程序定义的格式。

外部显示接口 (EPI)

EPI 允许将非 CICS 应用程序由与之连接的 CICS 服务器系统作为 3270 终端查看；EPI 应用程序和 CICS 终端都可以在 CICS 服务器中调度事务。应用程序可以同时使用几个服务器上的设施，并可以象有许多不同的 3270 终端那样操作。应用程序可以调度 CICS 事务，并且对于这些事务来说，它是主要设施。

由于 CICS 服务器支持通过 EPI 进行的访问，所以在服务器中运行的其它 CICS 事务可以使用 CICS START 命令来调度使用非 CICS 应用程序作为其启动终端的事务。当非 CICS 应用程序使用 EPI 启动 CICS 服务器中的事务时，3270 数据流和事件在服务器和应用程序之间传递。应用程序可以向其用户以适合于应用程序操作环境的方式显示终端 I/O 的内容。事务可以通过标准事务路由路由到其它 CICS 系统。其它 CICS 系统上的资源可以通过函数装运访问。

请注意，服务器事务可以是使用 3270 输入和输出的现有事务（有一些限制）。

外部调用接口 (ECI)

ECI 允许一个非 CICS 应用程序调用一个在 CICS 服务器中的 CICS 程序。应用程序可以同时和多个服务器相连，也可以同时有多个未完成的程序调用。

CICS 程序不能执行终端 I/O，但它可以访问和更新所有其它的 CICS 资源。同一个 CICS 程序既可以由一个非 CICS 应用程序使用 ECI 来调用，也可以由一个 CICS 程序使用 EXEC CICS LINK 来调用。在两个程序之间可以用 COMMAREA 方法交换数据，与 CICS 的方法类似。用户可以指定 COMMAREA 数据的长度以优化性能。

发出的调用既可以是同步的也可以是异步的。同步调用在所调用的程序完成时返回控制，返回的信息立即可以使用。异步调用返回控制时无须考虑所调用的程序是否完成，应用程序可以要求在其信息可用时通知它。

调用还可以扩展。也就是说，虽然每个逻辑工作单元在任何时刻只能有一个调用是活动的，但是单个逻辑工作单元可能会涉及两个或更多的连续调用。如果它使用异步调用，应用程序可以并行地管理许多逻辑工作单元。

被调用程序可以更新自身系统上的资源，它可以使用分布式程序链接 (DPL) 来调用其它系统上的 CICS 程序，并可以通过函数装运、通过分布式事务处理 (DTP) 或（在 CICS[®] Transaction Server OS/390[®] 版 或 CICS[®] Transaction Server VSE/ESA[™] 版[™] 环境中）通过前端编程接口 (FEPI) 来访问其它 CICS 系统上的资源。

有关外部访问接口的详细信息，请参阅 *CICS[®] Family: Client/Server Programming*。

外部安全性接口 (ESI)

ESI 允许非 CICS 应用程序调用由高级程序间通信 (APPC) 密码到期管理 (PEM) 提供的服务。

APPC PEM 和 CICS 一起提供了对 APPC 体系结构化的注册事务支持，该事务通过以下方式向 CICS 服务器注册用户标识并处理密码更改请求：

- 标识用户并认证用户身份
- 在认证特定用户的密码已过期时通知他们
- 让用户在其密码过期时（或之前）更改密码

- 告诉用户他们当前的密码在多长时间保持有效
- 提供有关使用特定用户标识访问服务器未授权的尝试的信息。

要使用 APPC PEM, “CICS 通用客户机” 必须通过 APPC 与 CICS 服务器连接。诸如资源访问控制设施 (RACF®) 这样的外部安全性管理器 (ESM) 也必须可用于 CICS 服务器。ESI 调用可以包括在 ECI 或 EPI 应用程序中。只有由 **CICS_EciListSystems** 和 **CICS_EpiListSystems** 函数返回的 CICS 服务器才是可接受的。

网络安全性

CICS Transaction Gateway 支持使用“安全套接字层 (SSL)”和 HTTPS 协议来提供对于成功因特网操作非常关键的安全通信。

第45页的『第5章 安全性』中详细讨论了网络安全性及其在 CICS Transaction Gateway 上的实现; 以下各节概述了 SSL 和 HTTPS 提供的功能。

安全套接字层 (SSL)

SSL 是一个**握手协议**, 开发这个协议的目的是在因特网上提供安全性和保密性。使用 SSL 协议确保:

机密性 要在客户机和服务器之间交换的数据是经过加密的, 因此只有该客户机 (您的应用程序或小应用程序) 和该服务器 (CICS Transaction Gateway) 才知道数据的含义。

SSL 使用公用密钥加密作为一种安全机制在服务器和客户机之间分发密钥。公用密钥加密技术使用一对非对称密钥进行加密和解密。在使用 SSL 的情况下, 秘 (对称) 钥在客户机和服务器之间进行传递 (使用公用密钥密码术), 然后使用这个秘钥来加密和解密所有 SSL 连接上的通信量。这种加密可以保护数据不会被其它各方窃取, 因为其它各方没有解密数据所需的秘钥。这样就确保了专用信息, 例如信用卡号, 能被安全传输。

完整性 消息传送包括了基于安全散列算法的消息完整性检查。这种算法将在发送消息时执行, 并在接收消息时再次执行。如果两个散列值不匹配, 将警告接收方该消息可能已被篡改。

可说明性

可说明性是由数字签名确保的, 因此如果有任何错误, 可以标识应该由哪一方负责。

认证 CICS Transaction Gateway 的 SSL 协议实现提供了**服务器认证**, 因此当客户机建立与 CICS Transaction Gateway 的连接时, 需要认证服务器的详细信息。也可以启用**客户机认证**, 在这种情况下, 服务器将认证客户机的详细信息。

这种认证机制是基于数字证书 (X.509v3 证书) 的交换。这些数字证书中包含了有关如系统名以及公用密钥这样的实体信息, 还有有关服务器数字签名的信息。数字证书由“认证中心 (CA)”颁发, 并使用 CA 的专用密钥进行加密。如果能使用 CA 的公用密钥解密证书, 就能知道证书中所包含的信息都是可信的, 也就是说, 该证书确实属于声称拥有它的那个所有者。

HTTPS

当前大多数浏览器都支持一种 URL 访问法, 即 HTTPS, 用于使用 SSL 与 HTTP 服务器进行的连接。HTTPS (HTTP+SSL) 是用于处理安全事务的一种 HTTP 变体。

安全连接通常与一个类似于“https://someAddress”这样的 URL 建立的。缺省 HTTPS 端口号是 443，它是由“因特网号码分配机构”(Internet Assigned Numbers Authority) 分配的。

密钥和证书

SSL 协议使用公用密钥密码术，又称为 X.509 协议，建议将其和 ISO 认证框架一起使用。这个框架提供了跨网络的认证。

X.509 中最重要的部分是其公用密钥证书的结构。可信的“认证中心 (CA)”为每个用户指定一个唯一的名称，然后签发一个包含有名称和用户公用密钥的已签名证书。

CICS Transaction Gateway 允许您从 CA 处获得外部签名证书，或者将自身作为 CA 创建以允许颁发“自签名”的 x.509 证书。外部签名证书更适合于因特网使用，而自签名证书则适合于组织内部的使用。

将 X.509 数字证书“封装”到 Java 类文件中，这样 SSL 和 HTTPS 协议就可以使用它们了。这些 Java 类文件称为 *KeyRing* 类。

CICS Transaction Gateway 包括一个 *iKeyman* 工具，用于管理 *KeyRing* 类。可以使用这种工具来创建 *KeyRing* 文件、生成证书、将证书存储在 *KeyRing* 中，以及执行各种其它管理功能。有关详细信息，请参阅第45页的『第5章 安全性』。

安全性出口

提供了安全性出口，使用户可以定义诸如公用密钥加密等安全性操作。它们还可以用在数据压缩上。我们提供了一些示例源文件来演示这些功能。

安全性出口机制也可以用于在启用客户机认证时认证 / 分析 X.509 客户机证书。

有关详细信息，请参考 *CICS Transaction Gateway: Gateway Programming* 一书。

CICS Transaction Gateway 和对象请求代理

某些 Web 服务器包含了符合“对象管理组 (OMG)”“公共对象请求代理 (CORBA)”标准的对象请求代理 (ORB)。如果有这样的 Web 服务器，可以开发能从远程浏览器使用 CORBA “因特网 InterORB 协议 (IIOP)”调用的小服务程序对象。使用 HTTP 将 IIOP 发送到浏览器。

某些浏览器也可以包括 ORB，允许它们运行发出 IIOP 请求的程序。这种能力由通过使用 Java “远程方法调用 (RMI)”进行通信的 Java 小应用程序使用；RMI 是最终映射成 IIOP 的一种绑定机制。使用这种技术，小应用程序可以通过使用 IIOP 与小服务程序通信，并让小服务程序调用 CICS 事务。

其它功能

支持 **Host on-Demand**，它提供了通过位于支持 Java 的 Web 浏览器上的 EPI 访问 CICS 3270 应用程序的一种方法。必需 IBM eNetwork™ Host on-Demand 版本 2.0 或更新版本。

提供了对本地 **CICS Transaction Gateway** 的支持。它可以让 Java 程序与本地安装的 CICS Transaction Gateway 进行通信，而无需使用网络。有关如何配置本地 CICS Transaction Gateway 的信息，请参考第44页的『准备使用本地 CICS Transaction Gateway 支持』。

欧元支持

CICS Transaction Gateway 支持数据流中欧元字符的使用。CICS Transaction Gateway 类 CicsCpRequest 将尝试返回适合于支持欧元的 JDK 的 Java 编码名，以及与“CICS 通用客户机”传递给 CICS 服务器等价的代码页 / 编码字符集标识 (CCSID)

例如，如果“CICS 通用客户机”传递给“CICS 服务器”的是代码页 5348（代码页 1252 的一种支持欧元的版本），那么返回给 Java 客户机的 Java 编码将是 Cp1252。如果 JDK 本身是支持欧元的，那么 Java 编码 Cp1252 也是支持欧元的。有关对 Java 程序欧元支持的进一步信息，请参考适当的 Java 文档。

可访问性

仅在 Windows 操作系统上测试了 CICS Transaction Gateway 的可访问性。除了配置工具以外，产品都是可访问的。变通办法是建议用户手工编辑 ini 文件。可以将它们读入符合可访问性需求的文本编辑器中。在将来的发行版中将会说明这个问题。

第2章 安装前的规划

本章通过讨论硬件和软件需求（包括支持的 Web 服务器 和浏览器）来帮助您规划 CICS Transaction Gateway 的安装。列出了 CICS Transaction Gateway 可连接的 CICS 服务器。

有关所有对硬件和软件需求的最新更改，请参考产品的“自述文件”。

硬件需求

CICS Transaction Gateway 在任何可运行相关操作系统和其它必备软件的硬件上运行。

CICS Transaction Gateway（仅包括美式英语文档）的硬盘需求是 **30 MB**。

需要另外 30 MB 用于安装期间使用的临时文件。在 Windows 驱动器上必须至少有 12MB 用于临时文件。

支持的软件

支持以下产品。如果指出，则表明已经在这些产品上测试过 CICS Transaction Gateway。

表 2. 支持的产品

操作系统

CICS Transaction Gateway 在以下操作系统上运行，并经过测试：

- 带有服务包 25 的 AIX V4.3.3（包括“UNIX 到 UNIX 复制程序”：bos.net.uucp）
- HP-UX 11.0
- Linux for S/390® - SuSe 7.0（内核级别 2.2.16）
- OS/390 V2.8
- OS/390 V2.10
- Solaris 7（32 位方式和 64 位方式）
- Solaris 8（32 位方式和 64 位方式）
- TurboLinux Server 6 for zSeries 和 TurboLinux Server 6 for S/390
- 带有服务包 6a 的 Windows NT Workstation V4.0 或带有服务包 6a 的 Windows NT Server V4.0。
注：不支持“Windows NT 终端服务器”。
- 带有服务包 1 的 Windows 2000 Professional 或带有服务包 1 的 Windows 2000 Server（具有“Windows 2000 Terminal Service”特性）
- z/OS V1

表 2. 支持的产品 (续)

Web 浏览器

- HTML/HTTP 功能: 任何支持 HTML V1.0 的浏览器都应该可以使用本产品。
- Java™ 功能: 任何与 JDK™ 1.1 兼容的 Web 浏览器都应该可以使用本产品。

已经用以下浏览器测试过 CICS Transaction Gateway:

- HotJava™ 浏览器 V 1.1 (Solaris 版)
- Microsoft® Internet Explorer 5.0
- Netscape Communicator 4.7 和 6.0

注: 在与 Netscape Communicator 浏览器一起提供的 JVM 中运行的小应用程序无法成功地连接到 CICS Transaction Gateway。消息号 CCL6652E 返回到“小应用程序”。目前该问题还没有解决方案。

Telnet 客户机

如果通过 telnet 运行 CICS 客户机, 您可能会发现某些 telnet 客户机会出现显示问题。例如, 当消息行超过一定长度时, telnet 会话可能会截断消息行。这通常是您正在使用的 telnet 客户机的问题, 或者是您登录时所用的终端类型的问题。目前该问题还没有解决方案。

JDK 级别

CICS Transaction Gateway 支持以下 JDK 级别:

- OS/390: IBM Java SDK 1.3
- AIX: IBM Java SDK 1.3 (服务发行版 6 或更高版本)
- HP-UX: Java SDK 1.3
- Linux for S/390: IBM Java SDK 1.2.2
- Solaris: Sun Java SDK 1.3
- Windows: IBM Java SDK 1.3 (服务发行版 6 或更高版本)
- 在除 OS/390 以外的所有操作系统上, 还支持 IBM Java SDK 1.2.2 (服务发行版 10 和 10a)

注: CICS Transaction Gateway 支持在与 JDK 1.1 兼容的 Web 浏览器中运行的小应用程序。

表 2. 支持的产品 (续)

CICS 服务器	<p>“CICS 服务器”运行实时多用户应用程序并管理相关的资源 and 数据。已经用以下服务器测试过 CICS Transaction Gateway:</p> <ul style="list-style-type: none"> • CICS[®] MVS/ESA[™] 版 V4.1, 具有通过 SNA 和 TCP62 进行通信的功能 (带有 APAR PQ30167, 用于支持注册的终端, 已应用) • CICS[®] OS/400[®] 版 V4.4 • 带有 CSD 3 的 CICS[®] Transaction Server OS/2[®] 版 V4.1 • CICS[®] Transaction Server OS/390[®] 版 V1.2 (带有 APAR PQ30168, 用于支持注册的终端, 已应用) • CICS[®] Transaction Server OS/390[®] 版 V1.3 (带有 APAR PQ30168, 用于支持注册的终端, 以及 OS/390[®] APAR PQ38644, 用于 EXCI, 已应用) • CICS[®] Transaction Server VSE/ESA[™] 版 1.1.0 (带有 APAR PQ30170, 用于支持注册的终端, 已应用) • CICS[®] Transaction Server VSE/ESA[™] 版 1.1.1 • CICS/VSE 2.3 (带有 APAR PQ30169, 用于支持注册的终端, 已应用) • TXSeries V4.2 + PTF 9 (HP-UX) • TXSeries V4.3 + PTF 4 (Windows[®] NT、AIX[®]、Solaris) • 带有 CSD 3 的 VisualAge[®] CICS Enterprise Application Development (OS/2[®] 和 Windows NT) (只用于应用程序开发) V3.1
Web 服务器	<p>已经用以下 Web 服务器测试过 CICS Transaction Gateway:</p> <ul style="list-style-type: none"> • Apache HTTP Server 1.3.12 (SuSe V7.0) • Domino[™] Go Webserver V4.6.2 (Windows、AIX 和 Solaris) • IBM HTTP Server 1.3 (Windows、AIX 和 Solaris) • Microsoft[®] Internet Information Server V5.0 (Windows)
应用程序服务器	<p>已经用以下 Web 服务器测试过 CICS Transaction Gateway:</p> <ul style="list-style-type: none"> • Windows 和 AIX 上的 IBM WebSphere[™] Application Server V3.5.3 企业版 (带有 PTF 3) • Windows 和 AIX 上的 IBM WebSphere[™] Application Server V3.5.3 高级版 (带有 PTF 3) <p>注: 还用以上“应用程序服务器”测试过终端小服务程序 (以及小服务程序样本)。</p>
TCP/IP 通信	操作系统提供 TCP/IP 支持。
TCP62 通信	CICS Transaction Gateway 提供 TCP62 支持。

表 2. 支持的产品 (续)

编译器与应用程序开发工具 已经用以下编译器和应用程序开发工具测试过 CICS Transaction Gateway:

AIX

- IBM C for AIX V4.4
- IBM C/Set++ for AIX V3.6.6
- IBM VisualAge C++ V4.0
- IBM VisualAge C++ V5.0

HP-UX

- HP ANSI C B.11.01.07
- HP aC++ B.11.01.06

Linux for S/390

- GNU 编译器 C 运行时 V2.95.2
- GNU 编译器 C++ 运行时 V2.95.2

Solaris

Sun WorkShop C++ V5.0

注: C++ 编译器必须应用了所有可用的服务补丁。

Windows

- IBM VisualAge for Java V3.5.3
- IBM VisualAge Interspace™ V5.2
- Microsoft Visual C++ V6.0
- Microsoft Visual Basic V6.0
- Microsoft Visual Basic Script (VBScript) V5.0

注: CICS Transaction Gateway 不支持带有 MTS 的 COM 库或 COM+ 的 MTS 组件的使用。

其它工具

CICS Transaction Gateway 支持以下工具:

- Adobe Acrobat

第3章 安装

本章描述如何安装“CICS® Transaction Gateway HP-UX 版”。

“CICS® Transaction Gateway HP-UX 版”是以包含可执行脚本文件的压缩 tar 文件分发的。当运行时，脚本文件创建库文件、命令、消息和定制 (.INI) 文件。

安装“CICS® Transaction Gateway HP-UX 版”

要安装“CICS® Transaction Gateway HP-UX 版”：

1. 作为 root 用户登录。
2. 在分发媒体中找到 tar 文件。它被称为 ctg-400h.tar.Z。
3. 将文件 ctg-400h.tar.Z 复制到硬盘上的工作目录中。
4. 解压文件并抽取文件：

```
uncompress ctg-400h.tar
tar -xvf ctg-400h.tar
```

这在当前目录中创建一个名为 **ctg_install** 脚本文件。

5. 输入以下命令安装 CICS Transaction Gateway：

```
ctg_install
```

注：安装进程使用 *uudecode* 函数，它是 *UNIX* 至 *UNIX* 复制程序的一部分。在安装 CICS Transaction Gateway 之前，请确保系统包含这个程序。

注：请参考 *CICS Transaction Gateway: 客户机管理 HP-UX 版* 中的『UNIX®系统中的内部客户机通信』以获取关于配置 HP-UX 系统允许大客户机数据流的建议。

国家语言支持

可以通过输入命令，选择“CICS® Transaction Gateway HP-UX 版”用以显示用户消息的语言：

```
mkclimsgs XX Code Set
```

其中 *XX* 是双字符消息语言。要获取可用语言的列表，请输入不带参数的命令：

“CICS 客户机 HP-UX 版” - 用户消息

	Language	Locale	Code Set
us	US English	en_US EN_US	iso81 utf8
fr	French	fr_FR FR_FR	iso81 utf8
de	German	de_DE DE_DE	iso81 utf8
it	Italian	it_IT IT_IT	iso81 utf8
es	Spanish	es_ES ES_ES	iso81 utf8
tr	Turkish	tr_TR TR_TR	iso89 utf8
jp	Japanese	ja_JP.SJIS ja_JP JA_JP	sjis eucJP utf8
kr	Korean	ko_KR KO_KR	eucKR utf8
cn	Simplified Chinese	zh_CN ZH_CN	hp15CN utf8

语法: mkclimsgs <Language> <Code Set>

从远程系统使用 X-Windows

当从远程系统使用 X-Windows 时，例如访问配置工具和 iKeyman，必须设置 DISPLAY 环境变量以允许应用程序在该系统上显示其窗口。

在显示系统上（将显示窗口的系统），输入命令：

```
xhost +appl
```

其中 *appl* 是正在用于运行应用程序的系统的网络名称。

在应用系统系统上运行应用程序之前，输入命令：

```
DISPLAY=disp:0
```

后跟

```
export DISPLAY
```

其中 *disp* 是将显示窗口的系统的主机名（后跟冒号和显示器标识--通常为 0）。然后，在 *disp* 系统上显示应用程序窗口。

卸载 CICS Transaction Gateway

要卸载 CICS Transaction Gateway，请输入：

```
mkcicscli -u
```

第4章 配置

本章描述下列内容:

- 『配置 CICS Transaction Gateway 的编程环境』
- 『使用配置工具』
- 第33页的『配置转换工具』
- 第34页的『编辑配置文件』
- 第36页的『重命名配置和映射文件』
- 第36页的『配置客户机键盘映射』.
- 第40页的『定制屏幕颜色』.
- 第44页的『准备使用本地 CICS Transaction Gateway 支持』.

配置 CICS Transaction Gateway 的编程环境

“Java 虚拟机” (JVM) 使用 CLASSPATH 环境变量来查找类和包含类的 zip 或 jar 档案。要允许 JVM 访问类文件, 必须指定包含类文件或档案的目录的完整路径。

要编译并运行用于 CICS Transaction Gateway 的 Java 应用程序, 请将下列文件的完整路径添加到 CLASSPATH 环境变量中:

- ctgclient.jar
- ctgserver.jar (如果正在使用本地网关)
- cfwk.zip (如果正在使用到网关的 SSL 连接)

这些档案位于安装 CICS Transaction Gateway 的目录的 classes 子目录中。

在 HP-UX 上设置 CLASSPATH

使用如下外壳命令来设置 CLASSPATH 环境变量:

```
export CLASSPATH=./opt/classes:/opt/ctg/classes/ctgclient.jar:${CLASSPATH}
```

这个外壳命令使 JVM 能访问当前目录、目录 /opt/classes 和档案 /opt/ctg/classes/ctgclient.jar 中的类。

可以将外壳命令放置在主目录下的 .profile 文件中, 这样当登录系统时, 将自动设置 CLASSPATH 环境变量。

使用下列外壳命令来显示 CLASSPATH 环境变量的值:

```
echo $CLASSPATH
```

使用配置工具

使用配置工具来设置 CICS Transaction Gateway 和 “CICS 通用客户机” 的配置参数。

启动配置工具之前, 使下列信息可用:

- 将用于连接 CICS 服务器的协议 (TCP/IP、TCP62)
- 服务器的主机名或 IP 地址

配置

- CTG 应该连接的服务器的端口号
- 将用于 Gateway 的协议 (TCP、SSL、HTTP 和 HTTPS)

要远程使用配置工具，必须使用第18页的『从远程系统使用 X-Windows』中描述的命令导出显示器。

要启动配置工具，输入 **ctgcfg** 命令。

任务向导

第一次运行配置工具时，“任务向导”帮助您定义关于系统的基本信息。可以输入 CICS 服务器的详细信息，并指定将使用的协议。以后可以通过配置工具更改这里输入的任何设置。可以绕过“任务向导”，通过在“任务向导”的打开窗口上单击“取消”而直接使用配置工具。

关于完成“任务向导”上任何屏幕的帮助，请参考联机帮助或『配置工具界面』中的相关部分。

配置文件

缺省情况下，将配置详细信息存储在安装 CTG 的 bin 子目录中的 CTG.INI 文件中。建议仅通过配置工具来编辑这个文件。

可以通过设置 CICSCLI 环境变量为配置文件指定不同的位置。CICS Transaction Gateway 日志会指示 CTG.INI 的位置。如果未找到 CTG.INI，将不运行 CICS Transaction Gateway。

如果在 bin 目录中找不到 CTG.INI，仍将启动“任务向导”。如果为配置文件指定一个不同的位置，并想编辑现有配置，则单击**取消**来绕过“任务向导”。一旦已经启动了配置工具，则打开配置文件。

配置工具界面

屏幕有 4 个区域 (以下分别描述)：

1. 菜单栏 (由“文件”、“编辑”、“工具”和“帮助”下拉菜单组成)
2. 带有下列图标的工具栏：“新建”、“打开”、“保存”、“剪切”、“复制”、“粘贴”、“新建服务器”、“跟踪设置”
3. 树状结构
4. 设置面板

第 5 区域是状态栏，显示了正在使用的配置文件和操作系统。状态栏只是显示。

使用 F10、Escape 和 Tab 键在 4 个输入区之间移动。

- 按 F10 激活菜单。
- 按 Escape，关闭下拉菜单而不做选择。
- 使用 Tab 键在工具栏、树状结构和设置面板之间移动。重复按 Tab 键移动到设置面板上的每个对象，然后移动到每个图标，最后回到树状结构。
- 如果按 Tab 键无法移动到下一个区域，则按 F10，然后在再次按 Tab 键之前，按 Escape (以激活并关闭菜单)。

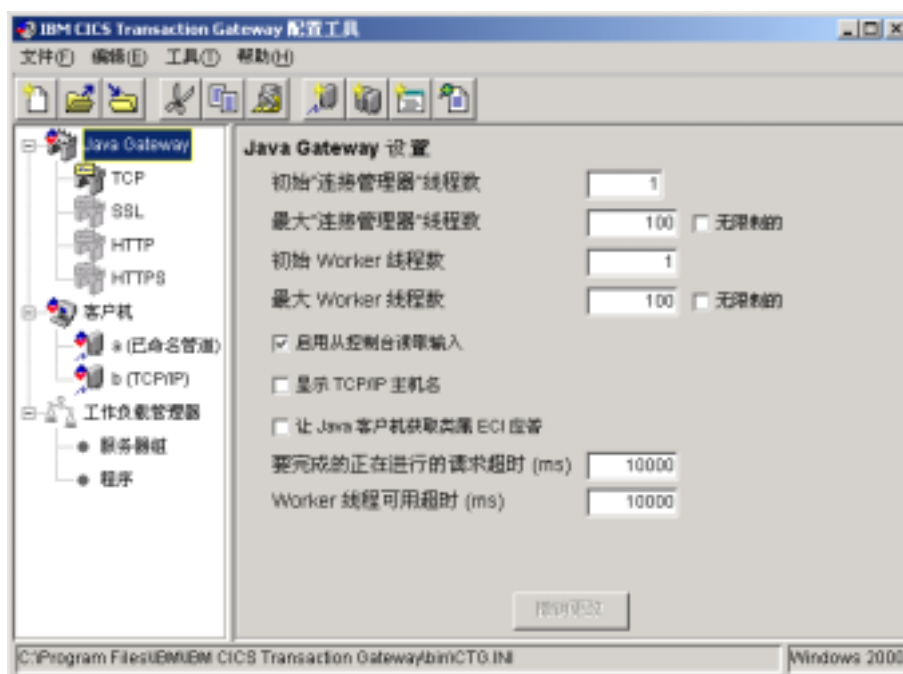


图 4. 配置工具

注：在您的平台上，配置工具可能不完全象图4中显示的那样出现。

树状结构

树状结构（参阅图4）允许浏览配置中的所有设置。根节点的类型如下：

Java Gateway

最多包含 4 个子节点，即，每种 CICS Transaction Gateway 可以使用的协议（TCP、SSL、HTTP 和 HTTPS）各包含一个。

客户机

包含每个服务器定义的子节点。

可访问性：使用向上和向下箭头键在节点和子节点之间移动。当打开“配置工具”时，扩展所有节点。要收缩节点，移动到其上然后按向左箭头键。要扩展它，按向右箭头键。

菜单栏

菜单栏包含文件、编辑、工具和帮助下拉菜单。

文件下拉菜单有下列选项：

- | | |
|-----|--|
| 新建 | 创建新的配置。 |
| 打开 | 打开一个现有配置。 |
| 保存 | 保存当前配置。如果已经创建了新配置，则文件名为 CTG.INI（在 bin 子目录中）。 |
| 另存为 | 允许覆盖配置文件的缺省路径和名称。 |
| 退出 | 退出配置工具。询问您是否想保存配置。 |

编辑下拉菜单中有剪切、复制和粘贴选项，每个选项都执行标准的文本功能。

配置

工具下拉菜单中有下列选项:

- 跟踪设置** 显示“跟踪设置”对话框。
- 新服务器** 启动“任务向导”以创建新建的服务器定义。
- 删除服务器** 从树状结构中删除服务器节点。

帮助下拉菜单具有下列选项:

- 上下文帮助** 根据当前屏幕上下文显示联机帮助信息, 例如, 对于特定配置设置。
- 目录** 显示联机帮助信息的目录列表。
- 索引** 显示联机帮助信息的主题索引。
- 关于** 显示关于配置工具的版本号和其它信息。

工具栏

工具栏与菜单栏上的工具栏具有相同功能, 但是它不包含**另存为**或**退出**选项。图标具有弹出帮助, 所以当光标置于其上时, 将出现描述选项的文本框。

设置面板

选择树状结构中的节点时, 显示适当的设置面板。这些面板中的设置与 CTG.INI 文件中的参数相对应。

在每个设置面板上, 都有**撤销更改**按钮, 允许撤销所作的更改。

配置 CICS Transaction Gateway 设置

要显示“Gateway 设置”面板, 可选择树结构中的 Gateway 节点。设置映射为 CTG.INI 文件的 Gateway 部分中的参数。

使用配置工具, 可以为任何使用 Gateway 命令行选项指定的参数提供预置值。

如果使用配置工具定义的参数在 Gateway 启动时经由相关的命令行选项指定, 则命令行设置的**优先级高**。

常规 Gateway 设置

常规 Gateway 设置是:

初始“连接管理器”线程数: 输入 ConnectionManager 线程的初始数目。缺省值是 1。

可使用 `ctgstart -initconnect` 命令覆盖该设置。

最大“连接管理器”线程数: 输入 ConnectionManager 线程的最大数目。缺省值是 100。

如果选择**无限制**复选框, 则 ConnectionManager 线程数将没有限制。

可用 `ctgstart -maxconnect` 命令覆盖该设置。

关于线程限制的信息, 请参阅第8页的表1。

初始“工作程序”线程数: 输入“工作程序”线程的初始数目。缺省值为 1。

可使用 `ctgstart -initworker` 命令覆盖该设置。

最大“工作程序”线程数: 输入工作程序线程的最大数目。缺省值是 100。

如果选择**无限制**复选框，则工作程序线程数将没有限制。

可使用 `ctgstart -maxworker` 命令覆盖该设置。

关于线程限制的信息，请参阅第8页的表1。

启用从控制台读取输入: 选择该复选框以启用从控制台读取输入。

缺省情况下启用从控制台读取输入。

可使用 `ctgstart -noinput` 命令覆盖该设置。

显示 TCP/IP 主机名: 选择该复选框以启用在消息中显示 TCP/IP 主机名。

这个选项允许选择在消息中如何显示 TCP/IP 地址。缺省情况下，CICS Transaction Gateway 在消息中用数字格式显示 TCP/IP 地址。如果启用该选项，CICS Transaction Gateway 将使用“域名系统”(DNS)将消息中的数字 TCP/IP 地址转换成符号 TCP/IP 主机名。这使读取该消息更加简便。

注: 选择这个选项可能在一些系统上导致严重的性能下降。

让 Java 客户机获取类属 ECI 应答: 如果希望 Java 客户机能够从 CICS Transaction Gateway 获取类属 ECI 应答，请选中该复选框。

类属应答是使用 `Call_Type: ECI_GET_REPLY` 或 `ECI_GET_REPLY_WAIT` 获取的。使用 `Call_Type: ECI_GET_SPECIFIC_REPLY` 或 `ECI_GET_SPECIFIC_REPLY_WAIT` 获取特定应答。该设置不适用本地 Gateway。

要完成的进行中请求的超时: 输入以毫秒为单位的值。

Java 客户机程序从 Gateway 断开连接时，Gateway 可能仍在代表该程序处理请求。如果该工作仍在进行，代表 Java 客户机管理请求的 `ConnectionManager` 将等待正在进行的请求完成直到超时为止。如果在该时限后，仍有请求在进行中，`ConnectionManager` 将继续其处理并且将自身标记成可用于新的连接。缺省的超时设置为 10000 毫秒，但可以用输入值覆盖缺省值。

如果该值设置为 0，则 `ConnectionManager` 将不等待正在进行的请求完成。

“工作程序”线程可用超时: 输入以毫秒为单位的值。

`ConnectionManager` 接受请求时，必须分配一个“工作程序”线程来执行该请求。但是，如果在超时周期内无法使用 `Worker`，将发送一个拒绝该请求的错误消息并且不执行请求。缺省的超时设置为 10000 毫秒，但可以用输入值覆盖缺省值。

如果将该值设置为 0，则若不能立即使用“工作程序”将拒绝请求。

TCP 协议设置

选择 **TCP** 子节点来显示 TCP 的设置:

启用协议处理程序: 选择这个复选框来配置使用该协议的 CICS Transaction Gateway。

端口: 输入协议的 TCP/IP 端口号:

配置

缺省的 TCP/IP 端口是 2006。

可使用 `ctgstart -port` 命令覆盖 TCP 协议的这个设置。

处理程序苏醒超时: 输入以毫秒为单位的值。

该设置控制协议处理程序在接受入站连接后的苏醒频率。苏醒后，它检查 Gateway 是否停止，因此该值影响 Gateway 完全关闭所需的时间。如果将该值设置为 0，则只有当接受了新的连接后处理程序才能苏醒，直到此时 Gateway 才能完全关闭。

连接超时: 输入以毫秒为单位的值。

接受新的连接后，该值指定协议处理程序等待 ConnectionManager 线程可用所需的时间。如果在这段时间内不能使用 ConnectionManager 线程，将拒绝连接。如果将该值设置为 0，则若不能立即使用 ConnectionManager，将拒绝连接。

空闲超时: 输入以毫秒为单位的值。

该设置指定连接允许保持休眠的时间。空闲超时周期从请求上次流过连接时开始计算。空闲超时期满后，会断开连接，但如果连接上仍有工作进行，则保留连接。如果未设置该值，或设置成 0，则将不会断开空闲连接。

如果服务器已经安装了这个功能的支持，那么清除处理只有在超时后才生效。

Ping 时间频率: 输入以毫秒为单位的值。

该值指定 Gateway 向连接的客户机发送用于检查客户机是否仍活动的 PING 消息的频率。如果发送下一个 PING 消息之前仍未接收到 PING 响应，则断开连接。同样地，如果连接上仍有工作进行，则可能保留连接（取决于断开工作连接值设置）。如果未设置该值，或设置成 0，则不会发送 PING 消息。

断开正在工作的连接: 选择该复选框来指定因空闲超时或 PING/PONG 故障而要断开的连接，**即使**该连接上仍有工作进行。

SO_LINGER 设置: 输入该处理程序使用的任何套接字的 SO_LINGER 设置。SO_LINGER 设置指定以秒为单位的关闭套接字延迟值。如果未输入该值，或设置成 0，则对该协议处理程序使用的任何套接字都将禁用 SO_LINGER。

如果启用 SO_LINGER，并且还未完成数据传输，“关闭”调用将阻塞该调用程序因此直到数据被发送或连接超时为止。如果禁用 SO_LINGER，返回“关闭”调用而不阻塞调用程序，TCP/IP 仍然尝试发送数据。通常这个传送可成功实现，但是无法保证，因为 TCP/IP 仅在指定的时间段内重复“发送”请求。

需要客户机使用安全性类: 如果想让 Gateway 只接受使用安全性类的连接，则选择该复选框。

Java 客户机程序连接到 Gateway 时，它可指定一对应用于连接的安全性类。但是，缺省情况下 Gateway 也接受来自未指定这对安全性类的程序的连接。

可通过控制 Gateway 能访问的 xxxServerSecurity 类集合来控制哪些安全性类是有效的。

关于存在的 CICS Transaction Gateway 安全性的详细信息，请参阅第11页的『安全性出口』。

SSL 协议设置

选择 **SSL** 子节点来显示 SSL 的设置。除了 SSL 还有下列设置外，其设置与 TCP 的设置相同：

使用客户机认证： 选择该复选框来启用 CICS Transaction Gateway 的客户机认证。缺省值是禁用客户机认证。

启用客户机认证时，任何尝试与 ssl: 或 https: 处理程序的连接都需要客户机显示其自身的“客户机证书”（也称为“数字标识”）。

关于如何获取客户机的“客户机证书”，请参阅第53页的『配置 SSL 客户机』。

KeyRing 类名： 输入服务器 KeyRing 的类名。

服务器 KeyRing 应包含有效的 x.509 证书，用来标识这个连接客户机的服务器。使用本产品提供的 SSL 工具生成这个 KeyRing 类。

关于 SSL 及其关联的 KeyRing 类的详细信息，请参阅第45页的『第5章 安全性』。

KeyRing 密码： 输入在创建进程中指定的服务器 KeyRing 类的“密码”。

SSL 的缺省端口是 8050。

HTTP 协议设置

选择 **HTTP** 子节点来显示 HTTP 的设置。其设置与 TCP 的设置相同。

HTTP 的缺省端口是 8080。

HTTPS 协议设置

选择 **HTTPS** 子节点来显示 HTTPS 的设置。其设置与 HTTP 的设置相同，除了 HTTPS 还有使用客户机认证、**KeyRing 类名**和 **KeyRing 密码**设置。参阅『SSL 协议设置』以获取详细信息。

HTTPS 的缺省端口是 443。

配置

配置客户机设置

要显示“客户机设置”面板，请在树状结构中选择“客户机”节点。设置映射为 CTG.INI 文件的 Client 一节中的参数。

缺省服务器

从下拉列表中选择缺省服务器。

Applid

最多输入 8 个字符，或保留该字段为缺省值 “*”。

该值指定了“CICS 通用客户机”工作站的 applid，其形式是它将作为一个系统在 CICS 服务器上自动安装。该名称在 CICS 服务器系统中必须是唯一的。“*”值自动生成一个被确保是唯一的名称。

注：如果客户机将自动安装到多个 CICS 服务器上，并且您为小应用程序输入了特定名称，则对于所有它连接的服务器来说该名称都必须是唯一的。如果该名称不是唯一的，则可能拒绝连接服务器的尝试，因为已经使用相同的名称安装了另一个客户机。如果使用名称 “*”，则在每个服务器上该客户机都可能具有一个不同的唯一名称。

最大缓冲区大小

输入从 4 到 32 范围内的一个千字节数。缺省值是 32 KB。

该值指定应用程序或终端数据将流入的传输缓冲区的大小。该值应该足够大，以满足使用最大可能的 COMMAREA 或终端输入/输出区域 (TIOA) 的需要。该值不包括某些协议的客户机所需的 512 字节的开销。

只有客户机正在限制内存的环境下运行时，才要改变这种缺省设置。

终端出口

输入 1 到 4 个字符的字符串。缺省值是 EXIT。

任何时刻在可以输入事务名称的地方在终端仿真器上输入该字符串将引起终端仿真器终止。该字符串不可以包含任何空白字符。

字符串是区分大小写的。如果终端仿真器在其 CICS 终端定义中具有大写转换，则应该输入大写的字符串。

最大服务器数

输入一个 1 到 256 范围内的值。缺省值是 10。

该值指定可从客户机并行访问的服务器的最大数。

最大请求数

输入一个 1 到 10 000 范围内的值。缺省值是 256。

该值指定可以同时为客户机上执行的最大项数，其中的项为下列之一：

- 终端仿真器
- EPI 终端
- ECI 工作单元

- ESI 工作单元

它是用来检测应用程序在出错时向服务器提交过多请求的失控情况。如果其它操作系统限制（例如，内存约束或通信会话）生效，则实际限制可能少于该设置。

打印命令

输入 1 到 256 个字符长度的字符串。

指定的字符串是运行客户机的操作系统的特定命令。客户机接收到打印请求时，会对每个打印请求生成具有唯一名称的临时文件。

参数字符串附加了临时文件名，并执行合成的命令。例如，这允许将打印请求复制到文件，定向到本地打印机或为包含进文档而进行格式化等。

“打印命令”负责在完成处理后删除临时打印文件。

使用后随删除命令 (**rm**) 的打印命令（例如，**lpr**）的 shell 脚本。

关于更多信息，请参阅打印文件中的描述。

打印文件

输入 1 到 256 个字符长度的字符串。

仅当省略“打印命令”设置时才可应用该选项。

指定的字符串标识客户机接收到的打印请求的输出将被定向到哪个文件。每个打印请求都附加到当前文件的结束处。

注：该设置仅作为缺省值。终端和打印仿真器提供覆盖该值的选项。

代码页标识重设项

输入表示“编码字符集标识” (CCSID) 的值以覆盖本地代码页标识。

如果您的平台已经更新成支持欧元并且“CICS 服务器”支持欧元，则使用该设置。例如，对于拉丁-1 国家，使用 858 作为 CCSID 值表示代码页 850 包含欧元支持。对于代码页 1252，指定 5348 作为 CCSID 值。

注意，**cicsterm** 总是基于工作站的本地代码页显示字符，而不论**代码页标识重设项**设置指定何值。

还应注意，如果使用 CCSID 来更改使用的代码页标识，则以前存储在服务器中的数据如果包含了代码点会生成不同字符的字符，那么数据在检索到“客户机”时可能被修改。

关于 CCSID 和数据转换的详细信息，请参考第87页的『附录A. “CICS 通用客户机”数据转换』。

服务器重试间隔

输入秒值。这是客户机尝试重新连接到它先前连接的服务器的时间间隔（以秒为单位）。缺省值是 60 秒。

当客户机知道它先前连接的服务器不再活动时，它试图在服务器变为非活动后 1 秒重新连接它。随后的重试尝试以**服务器重试间隔**定义的时间间隔发生。

配置

日志文件

输入用于问题诊断的日志文件名称。

若未指定，缺省的日志文件名是 CICSCLI.LOG，在 /var/cicscli 子目录中。

配置服务器设置

要显示“服务器设置”面板，请在树状结构中选择“服务器”节点。设置映射为 CTG.INI 文件的 Server 一节中的参数。

服务器名称

输入一个 1 到 8 个字符的名称。这为客户机的本地服务器提供了一个独立于通信协议的名称。

从 ECI、EPI、ESI 或终端仿真器请求访问服务器都通过该名称引用服务器。

描述

输入有关服务器的描述，长度在 1 到 60 个字符之间。该描述是任选的。

经由 CICS_EpiListSystems 和 CICS_EciListSystems 函数将该描述返回到运行在客户机上的应用程序。（参阅 *CICS® Family: Client/Server Programming* 一书）。

初始事务

输入 1 到 128 个字符的事务标识。

该字符串是区分大小写的，并标识终端仿真器连接到服务器时要运行的初始事务（和任何参数）。如果没有输入任何信息，则没有初始事务要运行。把字符串中的前四个字符或第一个空格前的字符作为事务。其余数据在调用时传递到事务。

终端型号定义

输入 1 到 16 个字符的字符串。

该字符串是区分大小写的，并指定服务器上终端型号定义的名称，以标识从客户机自动安装的终端特征。如果在服务器上找不到型号，或没有输入任何信息，则使用缺省终端定义。缺省值是服务器特定的。

终端型号定义设置的解释是服务器特定的。例如，对于 TXSeries AIX 版服务器，该值是 1 到 16 个字符，并且是用作型号的 CICS 终端定义项的 DevType。

使用大写安全性

选择该复选框，以指定将所有从 ECI 应用程序或用户提示所提供的用户标识或密码转换成大写。

缺省情况是禁用该设置。

网络协议

从下拉列表中选择要用于服务器连接的协议：

- TCP/IP
- TCP62

在面板上显示的协议设置根据您选择的协议而更改。

TCP/IP 设置

主机名或 IP 地址: 输入运行 CICS 服务器的主机的字符或数字 TCP/IP 标识。例如，cicssrv2.company.com（主机名）或 9.20.4.1（IP 地址）。

“主机名”通过名称服务器或 etc 子目录中的 hosts 文件映射为 IP 地址。但在 IP 地址改变的情况下，最好使用“主机名”。

端口: 输入一个 0 到 65 535 范围内的数值，它定义服务器上客户机应与之连接的端口号。缺省值是 0。

如果在 services 文件中未找到任何项，则假设值为 1435。这是在 TCP/IP 体系结构中指派给“CICS 通用客户机”的端口号。

连接超时: 输入一个 0 到 3600 范围内的值，以秒为单位指定建立连接所允许的最大时间限制，缺省值 0 表示客户机未设置时限。

如果连接的建立超过指定的时间则发生超时。将关闭 TCP/IP 套接字，传回客户机应用程序的返回码是 ECI_ERR_NO_CICS 或 CICS_EPI_ERR_FAILED。

如果服务器已经安装了这个功能的支持，那么清除处理只有在超时后才生效。

发送 TCP/IP Keepalive 包: 如果想让 TCP/IP 定期向服务器发送用来检查连接的 KeepAlive 包，则选择该复选框。

TCP62 设置

hosts 文件: 当配置 TCP62 CICS 服务器时，还必须更新本地 hosts 文件。文件路径如下：

Unix 系统
/etc/hosts

文件将 IP 地址映射成主机名。保持每项单独一行。项使用下列格式：

```
9.20.101.6          IYCQST34.GBIBMIYA.hursley.ibm.com
```

下面解释项的每个元素：

9.20.101.6

CICS 系统的 IP 地址。它必须从第一列开始，并且至少用一个空格将它与项的其余部分隔开。

IYCQST34.GBIBMIYA.

按逆向顺序的“伙伴 LU”名称。在上面的示例中，GBIBMIYA.IYCQST34 是在“配置”工具中输入的“伙伴 LU”名称。

hursley.ibm.com

Anynet 域名后缀。

伙伴 LU 名称: 输入客户机上 SNA 配置已知的服务器的 LU 名称。这必须是一个限定的 17 个字符名称，例如 ABC3XYZ4.PQRS1234。

本地 LU 名称或模板: 输入连接 CICS 服务器时使用的本地 LU 的实际 LU 名称或 LU 名称的模板。

配置

如果输入模板，则必须填写 **LU 名称模板的 IP 地址掩码（可选的）** 字段。

模板包含模板置换字符，即星号 (*)，它用于动态生成 LU 名称。例如，模板 CTS8BC** 表示必须动态生成名称来替换两个星号。使用基于模板、IP 地址掩码和工作站 IP 地址的算法来创建唯一的本地 LU 名称，例如，CTS8BC38。

使用动态生成本地 LU 名称，可以在所有 CICS 客户机机器上使用相同的配置，因为可以根据客户机的 IP 地址生成不同的本地 LU 名称。因此，不必在 VTAM® 内定义多个 LU。

如果有几百个客户机使用动态生成 LU 名称以连接 CICS 服务器，则在“本地 LU”名称中应有更多的模板置换字符，例如，CTS8****。事实上，CICS® Transaction Server OS/390® 版使用 LU 名称中的最后四个字符作为连接名，因此这些字符必须是唯一的。

LU 名称模板的 IP 地址掩码（可选的）：以大尾数法输入 IP 地址掩码，即，最高位的字节排在最前面。它必须是十六进制数，例如 FFFFFFF80。

在动态生成 LU 名称时使用该地址掩码。关于更多信息，请参阅**本地 LU 名称或模板**中的描述。

如果**本地 LU 名称**不是模板，则忽略该参数。缺省值是 00000000。

方式名：输入 1 到 8 个字符，以指定连接到服务器时要使用的方式名。如果未指定“方式名”，将使用缺省的方式名 TCP62。

可使用**最大逻辑 SNA 会话数**、**最大 SNA RU 大小**和 **SNA 调步大小**设置来定义由方式名指定的方式。

公共 TCP62 设置

全限定 CP 名称或模板：输入要启动的节点的全限定 CP 名称或全限定 CP 名称的模板。

如果输入模板，则必须填写 **CP 名称的 IP 地址掩码（可选的）** 字段。

模板包含模板置换字符，即星号 (*)，它用于动态生成 CP 名称。例如，模板 USIBMJKA.CP62**** 表示必须动态生成名称来替换四个星号。使用基于模板、IP 地址掩码和工作站 IP 地址的算法来创建唯一的 CP 名称，例如，USIBMJKA.CP62AH38。

使用动态生成的 CP 名称，可以在所有“CICS 客户机”上使用相同的配置，因为可以根据客户机的 IP 地址生成不同的 CP 名称。因此，不必将多个 CP 定义到 VTAM。

如果有几百个客户机使用动态生成 CP 名称，则在模板中应有更多的模板置换字符，例如 USIBMJKA.CP*****。

模板的网络标识部分（第一部分）一定不能包含任何模板置换字符。

CP 名称的 IP 地址掩码（可选的）：以大尾数法输入 IP 地址掩码，即，最高位的字节排在最前面。它必须是十六进制数，例如 FFFF8080。

在动态生成 CP 名称时使用该地址掩码。关于更多信息，请参阅**全限定 CP 名称或模板**中的描述。

如果全限定 **CP 名称或模板** 不是模板，则忽略该参数。缺省值是 00000000。

AnyNet 域名后缀： 输入 AnyNet® 域名后缀。该参数与伙伴（伙伴 **LU 名称**）一起用来确定与该 LU 有关的主机 IP 地址。

例如，如果域名后缀是 sna.ibm.com，伙伴 LU 名称是 NETID.LUA，则对主机名 LUA.NETID.sna.ibm.com 发出 gethostbyname 调用来获取 LU 名称为 NETID.LUA 的节点 IP 地址。

缺省值是 sna.ibm.com。

TCP62 端口： 输入 0 到 65535 之间的一个端口号以定义客户机应该连接的服务器上的端口号。缺省值是 0，它表示应该使用 TCP/IP 服务 MPTN 的端口（在 services 文件中定义）。

远程节点不活动轮询时间间隔： 输入一个 0 到 65535 之间的秒数值。缺省值是 60 秒。0 值关闭轮询。

这个配置设置控制“CICS 通用客户机”轮询不活动连接之前等待的时间。

高级 TCP62 设置

最大逻辑 SNA 会话数： 输入一个 1 到 255 范围内的值。缺省值是 8。

该参数用于 TCP62 定义逻辑 SNA 会话的数目。

最大 SNA RU 大小： 输入一个 256 到 4096 范围内的值。缺省值是 1024。

该参数用于 TCP62 定义由方式名设置指定的方式。

SNA 调步大小： 输入一个 0 到 63 范围内的值。缺省值是 8。

该参数用于 TCP62 定义由方式名设置指定的方式。

跟踪设置

要配置跟踪设置，则从工具菜单中选择跟踪设置选项。

跟踪设置

选中复选框来指定当跟踪打开时将跟踪的 CICS Transaction Gateway 和“CICS 通用客户机”组件。

跟踪全部	所有组件。
客户机 API 级别 1	客户机 API 层（级别 1）。
客户机 API 级别 2	客户机 API 层（级别 1 和 2）。
CICSCLI 命令行	cicscli 命令接口。
CICSTERM 和 CICSPRINT	cicsterm 和 cicsprnt 仿真器。
CPP 类	C++ 类库。
客户机守护程序	“CICS 通用客户机”守护程序。
传输层	进程间通信。
Gateway	CICS Transaction Gateway。

配置

JNI 跟踪	要启用“JNI 跟踪”，可设置环境变量 <code>CTG_JNI_TRACE=FILENAME</code> ，其中， <code>FILENAME</code> 对应于将写入跟踪的文件。
协议驱动程序级别 1	协议驱动程序（例如，TCP）。它跟踪已发送和已接收的数据并提供有关失败的补充信息。
协议驱动程序级别 2	它跟踪经由协议驱动程序的内部流并与其它软件组件交互。目前，只有 CCLTCP62 协议驱动程序支持它。

注：上面的任何一种设置都不能启用 JNI 跟踪。要启用它，必须将环境变量 `CTG_JNI_TRACE` 设置成指向要存储信息的文件。

也可通过在 `cicscli` 命令上使用 `-m` 参数来指定跟踪组件（除了 Gateway 组件）。任何使用 `cicscli` 指定的跟踪组件都会覆盖用配置工具指定的组件。如果跟踪组件不是由 `cicscli` 命令或配置工具指定的，则会跟踪缺省组件集，名为：**协议驱动程序、客户机守护程序和客户机 API 级别 1**。如果选中任何一个复选框，这将覆盖缺省组件集。

对于 API 组件，可以指定要跟踪的信息级别。**客户机 API 级别 1** 复选框指定跟踪与 API 相关的基本信息，例如，在 ECI、EPI 和 ESI 调用之前和之后。**客户机 API 级别 2** 复选框指定除了跟踪级别 1 的信息外，还要产生附加的 API 跟踪项。

注：`cicscli -d=nnn` 命令用于设置要跟踪的最大数据区大小。如果将 `nnn` 设置成小于期望的数据大小，则跟踪数据可能会被截断。

Gateway 跟踪文件

如果启用跟踪，则输入将写入跟踪消息的跟踪文件的路径名。如果未指定路径，则跟踪写入安装了 CICS Transaction Gateway 的 `bin` 子目录中的 `CTG.TRC` 文件。

还可在 `ctgstart` 命令上使用 `-file` 选项指定跟踪文件。

每次 CICS Transaction Gateway 启动时，并不附加至跟踪文件后；而是覆盖该文件。

性能：打开 Gateway 跟踪对性能有重要的影响。建议在产品环境中不启用 Gateway 跟踪。

Gateway 跟踪环绕大小

输入 0 到 1 000 000 千字节范围内的值。缺省值是 0。

该值以千字节为单位指定网关跟踪文件能达到的大小。后续跟踪项将继续从文件开始处写入。

客户机跟踪文件

如果启用跟踪，请输入将写入跟踪消息的跟踪文件的路径名。

不必输入文件扩展名，因为总会生成 `.BIN` 类型的文件（或在跟踪文件回绕时生成 `.WRP` 文件）。

如果不指定路径，则将跟踪写入 `CICSCLI.BIN` 文件。它在 `/var/cicscli` 子目录中。

要最小化任何性能影响，将客户机跟踪文件用二进制格式写出。要阅读它，必须使用 `cicsftrc` 命令将这个文件转换为 ASCII。

有关跟踪的更多信息，请参阅针对您的操作系统的 *CICS Transaction Gateway 客户机管理* 书籍。

最大客户机环绕大小

输入一个 0 到 999 千字节范围内的值。缺省值是 0。

该值指定环绕之前磁盘上将生成的跟踪文件大小。如果指定缺省值 0，则禁用跟踪环绕。

配置转换工具

使用配置转换工具 (ctgconv) 将 “CICS Transaction Gateway”、“IBM CICS 通用客户机” 和 “CICS Gateway Java 版” 以前版本的配置文件转换为新格式的 “CICS Transaction Gateway 版本 4.0” 配置文件。

使用以 CICS Transaction Gateway、“IBM CICS 通用客户机” 和 “CICS Gateway Java 版” 版本 3.1 之前的版本创建的配置文件的转换工具。如果正从版本 3.1 升级，则不需要使用它。

转换工具转换下列：

Gateway.properties	CICS Transaction Gateway 版本 3.0 和 “CICS Gateway Java 版” 的特性文件。
CICSCLI.INI	“CICS 客户机” 版本 2 和版本 3.0 的 “客户机初始化文件”。

缺省情况下，转换工具生成名为 CTG.INI 的输出文件。这个文件的样本在第34页的『编辑配置文件』中显示。

用 .BAK 扩展名重命名旧的文件，但是插入一个表示它们过时的标志。

使用转换工具

ctgconv 的参数是：

```
ctgconv -c=file [-g=file] [-o=file]
```

对于每个参数，*file* 可能是：

- 带有扩展名的文件名，在这种情况下假设为 /opt/ctg/bin 目录。扩展名必须是 .INI，除了 CICS Transaction Gateway 输入文件的扩展名为 .properties。
- 完整路径名。
- 目录名（在结尾不带 /），在这种情况下，假设是缺省的文件名。

-c=file

指定要转换的 “客户机初始化文件”。

如果 *file* 指定一个目录，则假设文件名为 CICSCLI.INI。

如果未指定参数，将使用 CICSCLI 环境变量来定位 “客户机初始化文件”。如果未设置 CICSCLI，则假设文件 假设 /opt/ctg/bin/CICSCLI.INI。

-g=file

指定要转换的 Gateway.properties 文件。

如果未指定 -g 参数，则在 CTG.INI 中不创建 Gateway 一节。

配置

-o=file

指定已转换文件的路径名。缺省值是 /opt/ctg/bin/CTG.INI。

要获得有关使用 `ctgconv` 的帮助，请输入：`ctgconv -?`

在转换期间，重命名输入文件，这样文件扩展名的最后 3 个字符变为 `.BAK`。带有该名称的任何文件都被覆盖，并且将按相同方式重命名转换之前存在的任何输出文件。

从旧配置文件中除去冗余参数，并在已转换的文件中给予其它参数新的名称。

编辑配置文件

虽然建议使用配置工具，但是也可以通过编辑配置文件来执行配置。

配置文件可用于“CICS 通用客户机”和 CICS Transaction Gateway，并且包含下列部分：

1. GATEWAY
2. CLIENT
3. SERVER
4. DRIVER

这些部分的格式为：

```
SECTION sectionname [=value]
    property1=value
    property2=value
    ...
    propertyN=value
ENDSECTION
```

必须重新启动“CICS 通用客户机”和 CICS Transaction Gateway 以使对配置文件所作的更改生效。

下列每个部分中都给出了特性示例。有关特性的信息，请参考配置工具中相应设置的描述。

注：如果用来表示注释，# 字符必须在行的开始处或者跟在空格或 `tab` 字符之后。这是因为某些有效名称（例如，方式名）可能以 # 字符开头。

GATEWAY 部分

```
SECTION GATEWAY
# -----
# Properties file for the CICS Transaction Gateway
# -----

    initconnect=1 # Initial number of ConnectionManager threads
    maxconnect=100 # Maximum number of ConnectionManager threads
    initworker=1 # Initial number of Worker threads
    maxworker=100 # Maximum number of Worker threads
    trace=on # Enable extra tracing messages
    notime=on # Disable timing information in messages
    noinput=on # Disable the reading of input from the console
    nonames=on # Do not display TCP/IP hostnames

# -----
# Entries to define the protocols which this CICS Transaction Gateway daemon will
# accept.
#
```

```

# Entries for a particular protocol begin with protocol@<protocol-name>.
# followed by which property is being set. The following properties MUST be set
# for each protocol handler to be used :
#
# (1) protocol@<protocol-name>.handler = ....
#     The name of the class which provides the handler for this protocol
#
# (2) protocol@<protocol-name>.parameters = ....
#     The parameters passed to the protocol handler. The possible parameters
#     vary upon the protocol handler.
#
#
protocol@tcp.handler=com.ibm.ctg.server.TCPHandler
Protocol@tcp.parameters=port=2006;sotimeout=1000;
connecttimeout=2000;idletimeout=600000;pingfrequency=60000

protocol@http.handler=com.ibm.ctg.server.HttpHandler
Protocol@http.parameters=port=8080;sotimeout=1000;connecttimeout=2000;
idletimeout=120000;requiresecurity

protocol@ssl.handler=com.ibm.ctg.server.SslHandler
Protocol@ssl.parameters=port=8050;sotimeout=1000;connecttimeout=2000;
Idletimeout=600000;pingfrequency=60000;keyring=ServerKeyRing;keyringpw=default;
clientauth=on;

Protocol@https.handler=com.ibm.ctg.server.HttpsHandler
Protocol@https.parameters=port=443;sotimeout=1000;
connecttimeout=2000;Idletimeout=120000;keyring=ServerKeyRing;keyringpw=default;
clientauth=off;

# -----
# Advanced settings.

ecigenericreplies=off
workertimeout=10000
closetimeout=10000

ENDSECTION

```

CLIENT 部分

```

SECTION CLIENT = *
# -----
# Client section - This section defines the local CICS client. There
# should only be one Client section.
#
#           MaxServers = 1           # Only allow one server connection
#           MaxRequests = 256        # Limit the maximum server interaction
#           MaxBufferSize = 32      # Allow for a 32K maximum COMMAREA
#           CCSID=850
#
ENDSECTION

```

SERVER 部分

```

SECTION SERVER = cicstcp1
# -----
# Server section - This section defines a server to which the client may
# connect. There may be several Server sections.
#
#           Description = A TCP/IP Server # Arbitrary description for the server
#           Protocol = TCPIP              # Matches with a Driver section below
#           NetName = cicstcp.ibm.com     # The server's TCP/IP name or address
#           Port = 0                      # Use the default TCP/IP CICS port
#
ENDSECTION

```

DRIVER 部分

```

SECTION DRIVER = TCPIP           ; Matches the Server's Protocol value
# -----
# Driver section - This section defines a communications protocol DLL
# used to communicate with a server. There may be

```

配置

```
#                several Driver sections.  
                DriverName = CCLIBMIP           #Use the IBM TCP/IP communications library  
ENDSECTION
```

注: DRIVER 部分不对应于配置工具中的任何设置。配置工具自动选择正确的协议驱动程序。

重命名配置和映射文件

缺省情况下, 在 /opt/ctg/bin 目录中与“CICS 通用客户机”一起提供了下列文件:

CTGSAMP.INI	客户机配置文件的样本。(缺省的配置文件名为 CTG.INI。)
CICSKEY.INI	键盘映射文件。(还提供了样本文件 CICSKEYSAMP.INI。)
CICSCOL.INI	颜色映射文件。(还提供了样本文件 CICSCOLSAMP.INI。)

建议用不同的名称创建这些文件的定制版本, 因为服务更新的安装可能覆盖文件并导致丢失任何定制。

通过下列环境变量来引用定制文件:

文件	环境变量
客户机配置文件	CICSCLI
键盘映射文件	CICSKEY
颜色映射文件	CICSCOL

配置客户机键盘映射

在键盘映射文件中定义了终端仿真器操作的键盘映射。缺省情况下, 提供的文件为 /opt/ctg/bin 目录中的 CICSKEYSAMP.INI。不管怎样, 建议创建自己定制的映射文件。

可以通过下列方式指定键盘映射文件:

- `cicsterm` 命令的 `-k` 选项, 它使用特定终端来标识键盘映射文件(请参阅页『配置客户机键盘映射』)。
- `CICSKEY` 环境变量。例如:

```
export CICSKEY=/var/cicscli/MYKEYS.INI
```

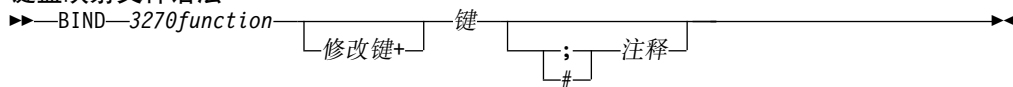
如果这二者均未指定, 则假设为 /opt/ctg/bin 目录中名为 `CICSKEY.INI` 的文件。

可以在任何时候更改键盘映射文件, 虽然直到下次启动终端仿真器时, 更改才会生效。

键盘映射文件语法

本部分描述了键盘映射文件的语法。必须为每个需要的键提供一条语句, 因为没有缺省赋值(除了字母和数字键)。它们是不区分大小写的, 因此可以用大写、小写或混合的方式输入关键字和值。每个绑定必须在单独的行, 使用下列格式:

键盘映射文件语法



例如，要将 3270 功能 EraseEof 映射成同时按下 Ctrl+Delete 键，绑定如下：

```
bind EraseEof Ctrl+Delete ;erase to end of field
```

键盘映射文件

在映射文件中，*3270function* 可以是任何下列之一：

backspace	pa1	pf1	pf13
backtab	pa2	pf2	pf14
clear	pa3	pf3	pf15
cursordown		pf4	pf16
cursorleft	printscreen	pf5	pf17
cursorright	reset	pf6	pf18
cursorselect	tab	pf7	pf19
cursorup		pf8	pf20
delete	ignore	pf9	pf21
enter		pf10	pf22
eraseeof		pf11	pf23
eraseinput		pf12	pf24
home			
insert			
newline			

提供 ignore 值以允许忽略键盘上不需要的控制键。（不生成不需要的符号。）

修改键可以是任何下列之一：

```
Ctrl
Shift
```

键可以是表3 中显示的任何一個键，但是不支持某些修改键加其它键的组合：

表 3. 可映射的“CICS 通用客户机”键

组	键
转义键	转义
功能键	f1 f2 f3 f4 f5 f6 f7 f8 f9 f10 f11 f12
数字键	0 1 2 3 4 5 6 7 8 9
字母键	a b c d e f g h i j k l m n o p q r s t u v w x y z
Tab 键	Tab
移动键	newline backspace insert home pageup delete end pagedown up left down right

配置客户机键盘映射

表 3. 可映射的“CICS 通用客户机”键 (续)

组	键
小键盘键	keypad/ keypad* keypad- keypad7 keypad8 keypad9 keypad4 keypad5 keypad6 keypad+ keypad1 keypad2 keypad3 keypad0 keypad. keypadenter

注: 对于 CICS 通用客户机 HP-UX 版, Ctrl/Act、Print Screen、Scroll Lock 和 Pause 键都不可用。缺省情况下, 3270 功能“清除”被指派成 Esc 键。

键组合

修改键和键的下列组合可以被映射:

- 无修改键** 所有键都可映射。
- Ctrl 修改键** 只有功能键、移动键、字母键、tab 键和小键盘键能被映射。
- Shift 修改键** 只有功能键、数字键、tab 键和字母键能被映射。

在图5中显示了键映射文件的样本。

```
*****  
;* IBM CICS Client - 3270 Emulator Keyboard Binding File *  
*****  
  
; Format:  
;   bind 3270_key [modifier+]workstation_key  
;  
; Refer to the "IBM CICS Clients Administration" book for full details  
; of available values.  
  
; Note: There are no default key bindings, ensure all required 3270  
;       keys are mapped. "Enter" and "Clear" are particularly  
;       important.
```

图 5. “CICS 通用客户机”样本键盘映射文件 (1/3)

```
; Miscellaneous keys
bind Enter      Newline
bind Enter      KeypadEnter
bind Enter      Ctrl+KeypadEnter
bind Clear      Escape
bind Reset      Ctrl+R
bind Newline    Ctrl+N
bind Tab        Tab
bind Backtab    Shift+Tab
bind Backspace  Backspace
bind Delete     Delete
bind Delete     Keypad.
bind Insert     Insert
bind Insert     Keypad0
bind EraseEof   Ctrl+E
bind EraseInput Ctrl+A
bind Ignore     End
bind PrintScreen Ctrl+P

; Cursor movement
bind CursorUp   Up
bind CursorUp   Keypad8
bind CursorDown Down
bind CursorDown Keypad2
bind CursorLeft Left
bind CursorLeft Keypad4
bind CursorRight Right
bind CursorRight Keypad6
bind CursorSelect Ctrl+F3
bind Home       Home
bind Home       Keypad7
```

图 5. “CICS 通用客户机” 样本键盘映射文件 (2/3)

定制屏幕颜色

```
; PF and PA keys
bind Pf1      F1
bind Pf2      F2
bind Pf3      F3
bind Pf4      F4
bind Pf5      F5
bind Pf6      F6
bind Pf7      F7
bind Pf7      Keypad9
bind Pf7      PageUp
bind Pf8      F8
bind Pf8      Keypad3
bind Pf8      PageDown
bind Pf9      F9
bind Pf10     F10
bind Pf11     F11
bind Pf12     F12
bind Pf13     Shift+F1
bind Pf14     Shift+F2
bind Pf15     Shift+F3
bind Pf16     Shift+F4
bind Pf17     Shift+F5
bind Pf18     Shift+F6
bind Pf19     Shift+F7
bind Pf20     Shift+F8
bind Pf21     Shift+F9
bind Pf22     Shift+F10
bind Pf23     Shift+F11
bind Pf24     Shift+F12
bind Pa1      Ctrl+Insert
bind Pa2      Ctrl+Home
bind Pa3      Ctrl+PageUp
```

图 5. “CICS 通用客户机” 样本键盘映射文件 (3/3)

定制屏幕颜色

在颜色映射文件中定义了屏幕颜色和属性。提供了可用于编辑的样本。缺省情况下，提供的文件是 /opt/ctg/bin 目录中的 CICSCOL.INI。建议创建自己定制的映射文件。

可以通过下列方式来标识颜色映射文件：

- `cicsterm` 命令的 `-c` 选项，它标识特殊服务器的颜色映射文件（请参阅页『定制屏幕颜色』）。
- `CICSCOL` 环境变量：例如：

```
export CICSCOL=/var/cicscli/MYCOLS.INI
```

如果这二者均未指定，则假设为当前目录中名为 `CICSCOL.INI` 的文件。

如果在硬件环境下不能精确复制 3270 屏幕属性（例如，闪烁或下划线），则使用颜色映射文件来提供硬件环境下的替代显示法。因此颜色映射文件定义了是客户机硬件上如何仿真 3270 屏幕属性。

颜色映射文件是可选的。但是，对于大多数硬件环境，如果仿真器需要闪烁或下划线支持，则需要映射文件。

注：

1. 如果颜色映射文件指定了属性的映射，那么即使正在运行客户机的硬件实际上支持屏幕属性，也会使用这个映射。

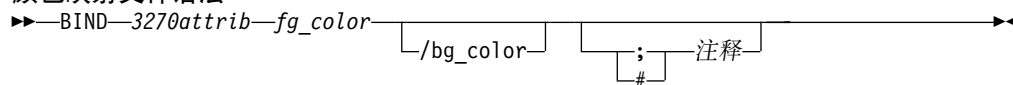
2. 如果应用程序需要用 3270 字段显示，例如，下划线，但是未指定仿真设置并且硬件无法显示下划线，则显示字段时，完全不使用任何突出显示。

可以在任何时候更改颜色映射文件，虽然直到下次启动终端仿真器时，更改才会生效。

颜色映射语法

颜色映射文件的语法如下。不区分大小写，因此可以用大写、小写或混合的方式输入关键字和值。每个绑定必须在单独的行，使用格式：

颜色映射文件语法



定制屏幕颜色 颜色映射文件

在颜色映射文件中, *3270attrib* 可以是任何下列之一:

```
normal_protected    intensified_protected
normal_unprotected  intensified_unprotected

default    blinking_default    underscored_default
blue      blinking_blue      underscored_blue
green     blinking_green     underscored_green
cyan      blinking_cyan      underscored_cyan
red       blinking_red       underscored_red
magenta   blinking_magenta   underscored_magenta
white     blinking_white     underscored_white
yellow    blinking_yellow    underscored_yellow

default_highlight

operator_information_area
```

每个 *fg_color* 和 *bg_color* (前景颜色和背景颜色) 可以是任何下列之一:

```
black    light_gray
blue     light_blue
brown    yellow
cyan     light_cyan
green    light_green
magenta  light_magenta
red      light_red
gray     white
```

如果省略了 *bg_color*, 则采用黑色的缺省值。

在第43页的图6 中显示了样本颜色映射文件。

```

;*****
;* IBM CICS Client - 3270 Emulator Color Binding File *
;*****

; Format:
;   bind 3270_field foreground_color/background_color
;
; Refer to the "IBM CICS Clients Administration" book for full details
; of available values.

; Operation information area at the bottom of the screen.
bind operator_information_area  black/green

; Color used as default for a terminal defined as monochrome, or to display
; characters displayed in the default color when character or field
; attributes are used
bind default                    light_green/black

; Color used for an intense field when either the terminal is defined as
; monochrome, or the screen has been formatted with extended attributes but
; the data to be displayed has default color and highlight.
bind default_highlight          white/black

; Colors used for datastream formatted with field attributes when no
; extended character or field attributes are in use
bind normal_unprotected         light_green/black
bind intensified_unprotected    light_red/black
bind normal_protected           light_cyan/black
bind intensified_protected      white/black

; Colors used for datastream formatted with extended field attributes
; or datastream formatted with character attributes.
bind red                        light_red/black
bind green                      light_green/black
bind blue                      light_blue/black
bind magenta                   light_magenta/black
bind cyan                      light_cyan/black
bind yellow                    yellow/black
bind white                     white/black

; The following bindings provide a mapping for 3270 blink and under-
; score attributes, as most workstation displays do not provide blink
; or underscore capabilities. For those that do these bindings could
; be deleted.
bind blinking_default           light_green/gray
bind blinking_red              light_red/gray
bind blinking_green            light_green/gray
bind blinking_blue             light_blue/gray
bind blinking_magenta          light_magenta/gray
bind blinking_cyan             light_cyan/gray
bind blinking_yellow           yellow/gray
bind blinking_white            white/gray

```

图 6. 样本颜色映射文件 (1/2)

定制屏幕颜色

```
bind underscored_default      light_green/light_gray
bind underscored_red          light_red/light_gray
bind underscored_green        light_green/light_gray
bind underscored_blue         light_blue/light_gray
bind underscored_magenta      light_magenta/light_gray
bind underscored_cyan         light_cyan/light_gray
bind underscored_yellow       yellow/light_gray
bind underscored_white        white/light_gray

;* End of file                                     *
```

图 6. 样本颜色映射文件 (2/2)

准备使用本地 CICS Transaction Gateway 支持

在能够运行使用本地 CICS Transaction Gateway 支持的应用程序之前，必须确保已经正确配置了您的环境。需要下列：

1. 正确配置的 Java 环境

利用本地 CICS Transaction Gateway 支持的 Java 应用程序可以被视为正常的 Java 程序，所以可以使用标准的 JDK java 可执行文件来执行。因此，必须正确设置环境以找到 Java 二进制文件、库和类。应该参考 JDK 档案以获取正确的环境设置。

2. 允许找到 CICS Transaction Gateway 二进制文件的正确设置

需要可用的 CICS Transaction Gateway .class 文件。应该将 CLASSPATH 环境变量设置成包含 CICS Transaction Gateway 的 ctgserver.jar 和 ctgclient.jar 档案。相应的二进制文件也必须是可用的。

将 SHLIB_PATH 环境变量设置成包含 CICS Transaction Gateway 的 bin 子目录，这样应用程序可以找到 libCTGJNI.so。

要运行应用程序： 使用带有适当选项的 JDK java 命令。

第5章 安全性

本章描述如何设置 CICS Transaction Gateway 来使用网络安全协议 SSL 和 HTTPS。

- 『概述』提供网络安全概念和术语的概述。它介绍了密钥、数字证书和 KeyRing 的概念。
- 第48页的『SSL 与认证』描述了 SSL 协议及其提供的认证类型。
- 第49页的『ctgikey 工具』介绍了 iKeyMan, 它是由 CICS Transaction Gateway 提供用于管理数字证书的工具。
- 第50页的『使用在外部签署的证书 (SSLight)』描述如何获得外部签署的数字证书并将它们存储在 KeyRing 文件中, 以供 CICS Transaction Gateway 使用。
- 第54页的『使用自签证书 (SSLight)』描述如何生成自签署的数字证书, 以供 CICS Transaction Gateway 使用。
- 第57页的『将 CICS Transaction Gateway 配置为 SSL 和 HTTPS』描述如何配置 CICS Transaction Gateway 来使用安全协议。

由 CICS Transaction Gateway 提供的 SSL 实现以纯 Java 编写, 称为 **SSLight**。另一个实现 **System SSL** 仅应用于 OS/390 上的 CICS Transaction Gateway, 并且只能用于该平台的 SSL 服务器。

概述

CICS Transaction Gateway 为那些对成功的因特网操作十分重要的安全通信提供全面支持。安全网络协议允许客户机小应用程序和应用程序使用 SSL 来安全地与您的 CICS Transaction Gateway 进行通信。第1页的『第1章 概述』介绍了 SSL 和 HTTPS 协议; 本章则提供了设置 CICS Transaction Gateway 来使用这些协议的更详细的信息。

以下部分描述了安全通信方面的特征:

- **机密性**

机密性意味着在因特网或内部网中传递消息时, 消息内容是保密的。机密性通过对消息加密来保证。

- **完整性**

完整性意味着在传输消息时不改变消息。当消息通过传输路线中的路由器时, 任何路由器都可以插入或删除文本或者使消息错乱。没有完整性, 您就无法保证发送的消息与接收到的消息相符。完整性通过加密和数字签名来保证。

- **可说明性**

可说明性意味着发送方和接收方都同意进行信息交换。如果没有可说明性, 接收方可以说消息从未到达过。可说明性通过数字签名来保证, 这样, 如果出错, 您可以确定谁应该对此负责。

- **真实性**

真实性意味着您知道您在和谁说话, 并且您能够信任那个人。真实性需要验证标识, 以便您可以确保其他人正是他们自称的那些人。真实性通过使用数字签名和数字证书实现。

什么是加密？

加密确保在因特网上发送的传输中的机密性。最简单的加密是对消息进行加扰使其无法被阅读，直到稍后由接收方解扰为止。发送方使用算法模式或密钥来加密消息，接收方使用解密密钥来解扰消息。

有两种密钥可用来进行加密（还可用于数字签名和认证）：

1. 对称
2. 非对称

在对称密钥方法中，发送方和接收方共享某种模式，发送方使用该模式来加密消息，接收方使用它来解密消息。对称密钥的风险在于：当和要与之进行通信的人共享密钥时，必须找到一种安全的传输方法来使用。

在非对称密钥方法中，要创建一个密钥对。这个密钥对由一个公开密钥和一个专用密钥组成。与对称密钥不同，这两个密钥彼此不同，并且专用密钥比公开密钥具有更多的安全加密模式。

发送方可以向它要与之安全通信的所有人广播其公开密钥。它保留专用密钥并用密码加以保护。只有发送方可以解密接收到的、用其公开密钥加密的消息，因为只有它才有专用密钥。

象 SSL 这样的协议既使用非对称（也称为公开密钥）密码术，也使用对称密钥密码术。公开密钥密码术用于 TCP/IP 握手。在握手期间，客户机向服务器传递主密钥。客户机和服务器使用主密钥来生成它们自己的会话密钥。然后，使用会话密钥为余下的会话来加密和解密数据。

数字签名和数字证书

数字签名是确保可说明性的、唯一用数学方法计算出来的签名。

数字证书允许对实体进行唯一的标识，它实际上是一张由可信的第三方签发的电子标识卡。

数字证书有两个作用：它建立所有者的标识，并使所有者的公开密钥可用。数字证书由可信的权威机构，即认证中心 (CA)（如 VeriSign Inc, Thawte），签发。证书签发后只在有限期间内有效，当超过其到期日时，必须将其替换。

数字证书由以下部分组成：

- 被认证人员的公开密钥。
- 被认证人员的名称和地址，也称为专有名称 (DN)。
- CA 的数字签名。
- 签发日。
- 到期日。

“专有名称”是人或组织的名称与地址。在申请证书时，必须输入您的“专有名称”。通过数字方式签发的证书不仅包括您自己的“专有名称”，还包括允许对 CA 进行验证的 CA 的“专有名称”。

要安全地通信，传输中的接收方必须信任签发发送方正在使用的证书的 CA。因此，任何时候当发送方签署消息时，接收方必须具有相应 CA 的签署证书和指定为可信根密钥

的公开密钥。例如，您的 Web 浏览器就会有一个可信 CA 的签署证书缺省列表。如果要信任其它 CA 签发的证书，则必须从该 CA 接收证书并将其指定为可信根密钥。

如果将包含您的公开密钥的数字证书发送给其他人，有什么方法可以防止那个人滥用您的数字证书并假冒您呢？答案是：您的专用密钥。仅有数字证书永远不能证明任何人的身份。数字证书通过提供检查所有者数字签名所需的公开密钥，仅允许验证所有者的身份。因此，数字证书所有者必须保护与数字证书中的公开密钥配对的专用密钥。否则，如果专用密钥丢失，任何人都可以假冒成该数字证书的合法所有者。

获得数字证书

可以用两种方法获得证书：

1. 从 CA 购买证书
2. 为自己签发证书，即，作为自己的 CA。

从 CA 购买证书

如果计划在因特网上从事商业活动，则应该从 CA（如 VeriSign Inc，它的主页是 <https://www.verisign.com/>）购买一个服务器证书。

向 VeriSign 提交证书请求后，CA 会在为您签发证书之前让您证明您的身份。虽然这种证明过程对于保护请求者、请求者的组织以及 VeriSign 是必要的，但它花费的时间可能比您希望的时间长。VeriSign 将以数字方式签署您的证书请求，并通过电子邮件将唯一的证书返回给您。

第50页的『使用在外部签署的证书 (SSLight)』中描述了如何获得在外部签署的证书。

注：VeriSign 服务器证书无法在不同机器上的服务器之间共享。

自己签发证书

如果自己作为 CA，则可以签署自己或其他任何人的证书请求。如果只需要在自己的组织内部使用证书，而不在外部因特网商业环境中使用，则这是个很好的选择。在这种情况下，您可能想仅允许对内部网中小心控制的关键人员小组进行访问。

关键人员将拥有象 Netscape Navigator 这样的浏览器，它可以接收您自签的 CA 证书并将其指定为可信根。然后，他们就可以信任您的通信并安全地共享信息。

第54页的『使用自签证书 (SSLight)』中描述了如何生成自签证书。

KeyRing

那么，在何处存放数字证书及其相关密钥呢？答案是：公开密钥、专用密钥、证书和可信根密钥被存放在 *KeyRing* 文件中。

CICS Transaction Gateway 使用 Java 类在 SSL 服务器和 SSL 客户机上存放证书和密钥数据。CICS Transaction Gateway 守护程序（它作为 SSL 服务器）使用服务器 KeyRing（例如 **ServerKeyRing.class**），而所有 SSL 客户机则使用客户机 KeyRing（例如 **ClientKeyRing.class**）。SSL 和 HTTPS 协议要求访问这些 Java 类以建立安全的连接。在配置 CICS Transaction Gateway 时建立该访问，请参阅第57页的『将 CICS Transaction Gateway 配置为 SSL 和 HTTPS』。

本章的以后各节将告诉您如何：

- 创建 KeyRing 文件。
- 获得数字证书
- 将数字证书接收到 KeyRing 文件中。

SSL 与认证

SSL 允许客户机认证服务器的身份，这称为*服务器认证*。

SSL 版本 3 还允许服务器认证客户机，这称为*客户机认证*。如果服务器需要在响应之前确认客户机的身份，则需要该功能。如果设置了 SSL 客户机认证，则每当客户机进行 SSL 连接时，服务器都请求客户机的证书。服务器在为客户机提供文档之前，将客户机请求中的 DN 信息与客户机证书中的 DN 信息进行对比确认。

SSL 使用安全性握手来初始化客户机和服务器之间的 TCP/IP 连接。在握手期间，客户机和服务器就将在会话中使用的安全性密钥和将在加密过程中使用的算法达成一致。客户机认证服务器。另外，如果客户机请求的文档受 SSL 客户机认证的保护，则服务器还要请求客户机的证书。握手之后，使用 SSL 加密和解密客户机请求和服务器响应中的所有信息，包括：

- 客户机正在请求的 URL
- 正在提交的任何表单的内容
- 如用户名和密码等的访问权限信息
- 在客户机和服务器之间发送的所有数据

第49页的图7中演示了 SSL 握手。

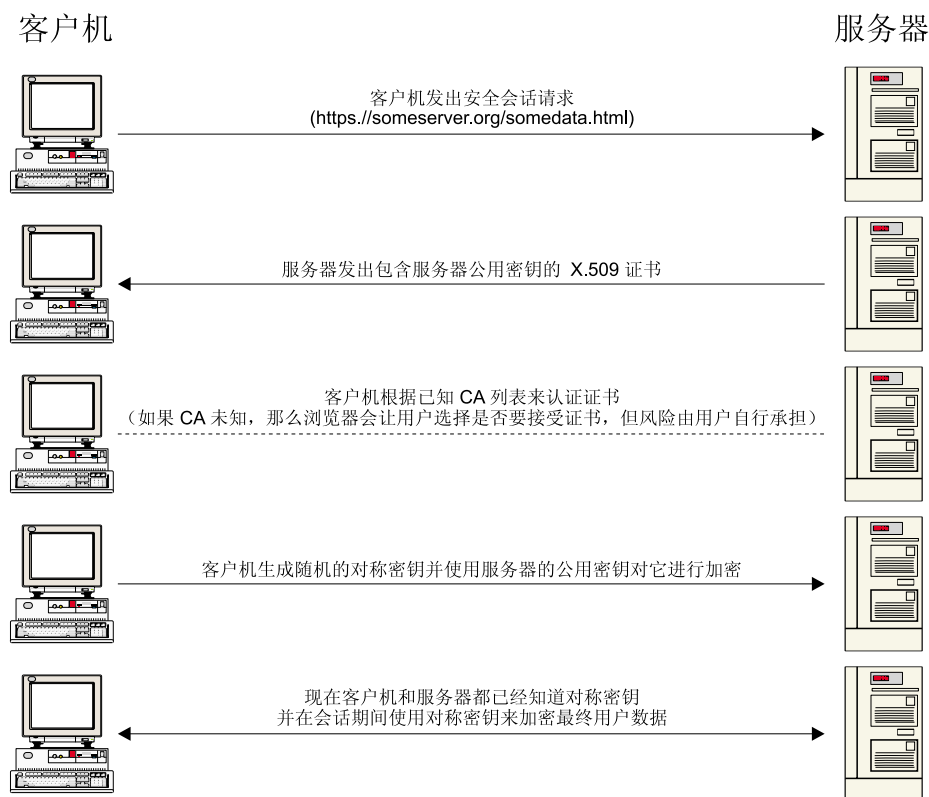


图 7. 带有服务器认证的 SSL 握手

HTTPS

HTTPS 是唯一将 SSL 和 HTTP 组合在一起的协议。需要在链接到受 SSL 保护的文档的 HTML 文档中指定 https:// 作为锚点。客户机用户也可以通过指定 https:// 来打开 URL, 以请求受 SSL 保护的文档。

因为 HTTPS (HTTP + SSL) 和 HTTP 是不同的协议, 并且通常使用不同的端口 (分别是 443 和 8080), 所以可以同时运行 SSL 和非 SSL 请求。因此, 您可以选择为所有使用不安全性的用户提供信息, 并只对发出安全请求的浏览器提供特定的信息。正因为这点, 因特网上的零售公司才可以允许用户对商品进行非安全性的浏览, 但是接下来用户可以安全地填写订购表单并发送其信用卡号码。

不支持“SSL 上的 HTTP”的浏览器当然无法请求使用 HTTPS 的 URL。非 SSL 浏览器将不允许提交那些需要安全提交的表单。

ctgikey 工具

CICS Transaction Gateway 提供一个工具 **iKeyman** 来维护您的数字证书。可以使用 **ctgikey** 命令来设置正确的环境 (包括 JAVA_HOME 环境变量) 并调用 iKeyman。

使用 iKeyMan, 您可以:

- 从 CA 请求和接收数字证书, 请参阅第50页的『使用在外部签署的证书 (SSLight)』。

安全性概念概述

- 生成自签证书，请参阅第54页的『使用自签证书 (SSLight)』。
- 将证书添加到您的 KeyRing 文件中。
- 更改 KeyRing 密码。
- 将专用密钥设为缺省值。
- 删除密钥。
- 通过将密钥复制到文件来导出密钥。
- 从导出的副本导入密钥并将其添加到 KeyRing 中。

将 iKeyman 分发到客户机工作站

虽然大部分密钥环管理的工作都将在 CICS Transaction Gateway 机器上进行，但是可能还需要为与 SSL 服务器连接的客户机提供 iKeyman 工具。客户机机器必须具有最小级别的 CICS Transaction Gateway Java 支持，即 IBM Java SDK 1.2.2。它们还需要一个脚本或命令文件来调用 iKeyman Java 启动类。

例如，以下是用 JDK 调用 iKeyman 的 Windows NT 命令行：

```
java.exe -classpath
"e:\ikeyman\cfwk.zip;e:\ikeyman\gsk4cls.jar;e:\ikeyman
\swingall.jar;" -Dkeyman.javaOnly=true com.ibm.gsk.ikeyman.Ikeyman
```

或者，用 JRE 调用：

```
jre.exe -classpath
"e:\ikeyman\cfwk.zip;e:\ikeyman\gsk4cls.jar;e:\ikeyman
\swingall.jar;" -Dkeyman.javaOnly=true com.ibm.gsk.ikeyman.Ikeyman
```

这些示例假设客户机工作站在 ikeyman 目录中有以下文件：

- cfwk.zip
- cfwk.sec
- gsk4cls.jar
- swingall.jar
- ikmininit.properties

所有这些文件都可以在安装 CICS Transaction Gateway 的 classes 子目录中找到。

注：必须确保 cfwk.zip 是在 CLASSPATH 设置中第一个出现的由 IBM 提供的档案。

使用在外部签署的证书 (SSLight)

CICS Transaction Gateway 可以充当一个 SSL 服务器，具有认证 SSL 客户机和接受来自“认证中心”（如 VeriSign Inc）的、在外部签署的证书的能力。

本节描述如何用证书管理接口 iKeyMan 来配置 SSL 服务器和 SSL 客户机。这个接口完全用 Java 编写，因此可以将它分发到一些装有合适的 JVM 的客户机/服务器工作站。

配置 SSL 服务器和客户机涉及到以下工作：创建 KeyRing 类、获得数字证书并将它们接收到 KeyRing 类中。

服务器 KeyRing 类包含一个“服务器证书”和相应的专用密钥以及一些签署证书。“服务器证书”是一个用于标识连接客户机的 SSL 服务器的数字证书。

客户机 **KeyRing** 类至少要包含 SSL 服务器的签署证书，如果需要客户机认证，则还需要一个客户机 x.509 证书。

配置 SSL 服务器

本节描述如何从 VeriSign Web 站点 (www.verisign.com) 获得“试用（测试）服务器证书”。VeriSign 允许对“服务器证书”试用 14 天。由于这是一个演示“服务器证书”，因此，它不是由可信的“VeriSign 认证中心”，而是由“VeriSign 测试 CA”签署的。

所有 SSL 客户机都需要将“VeriSign 测试签署证书”根密钥安装在它们的客户机 KeyRing 中，或者，在进行 HTTPS 连接的情况下，需要将资源库安装在浏览器中。

本书中的示例假设您使用的是 Netscape Communicator 版本 4.5。

获得完整的“VeriSign 服务器证书”的步骤与此处所描述的获得试用版本的步骤一样。

创建服务器 KeyRing

第一步是创建一个服务器 KeyRing 类，这个类最终将包含您的签署证书以及您的“服务器证书”（及其相关的专用密钥）。这个资源库是受密码保护的，当用 iKeyMan 创建 .class 时会为您提示密码“强度”。建议使用字母数字字符序列；这使得密码在面对“蛮力字典”攻击时更加强壮。

要获得证书：

1. 启动 ctgikey。
2. 选择密钥数据库文件 --> 新建。
3. 从密钥数据库类型选择 **SSLight** 密钥数据库类。
4. 输入 ServerKeyRing.class 作为文件名
5. 在位置中，输入存储您的 ServerKeyRing.class 的合适位置。
6. 选择确定。

生成的 ServerKeyRing.class 在缺省情况下包含一些常用的签署证书，其中包括 VeriSign 根签署证书及其“测试签署证书”。它还包含允许“服务器”用“VeriSign 客户机证书”验证客户机的“VeriSign Class 1 through 4 Public Certification Authority”签署证书。这在第53页的『配置 SSL 客户机』中有详细的解释。

准备证书请求

在可以从 www.verisign.com 获得“服务器证书”之前，SSL 服务器必须进行“证书请求”并在本地存储它：

1. 从 iKeyMan 选择**创建**。
2. 选择**创建新的证书请求**。
3. 必须填写这个证书请求。有些字段是可选的，但是必须至少填写以下字段（如示例所示）：

密钥标签	verisignServerCert
密钥大小	1024
公共名称	servermachine.hursley.ibm.com
组织	IBM UK
国家	GB

使用在外部签署的证书

4. 指定要存储证书请求的文件名，然后选择**确定**。然后，iKeyMan 将生成一个公开/专用密钥对，根据处理器速度的不同，这可能需要一些时间。

获得“服务器证书”

下一步是连接到 VeriSign Web 站点并请求一个“试用服务器证书”。将您的浏览器指向 <http://www.verisign.com>，然后进入页面 <https://digitalid.verisign.com/server/trial/trialIntro.htm>。现在，您可以登录来申请“试用服务器证书”了。

1. 选择**继续**。
2. 再次选择**继续**，然后将出现**第 2 步（共 5 步）- 提交 CSR**。
3. 将您的证书请求文件的内容粘贴到提供的文本框中。您的证书请求的内容应该类似于：

```
-----BEGIN NEW CERTIFICATE REQUEST-----
MIIB1jCCAT8CAQAwgZUXIDAeBgNVBAMTF2NocmlzdHAuaHVyc2x1eS5pYm0u
Y29tMRswGQYDVQQLEExJDbG11bnRzICsgR2F0ZXdheXMxDzANBgNVBAoTBk1C
TSBVSzETMBEGA1UEBxMKV01uY2h1c3RlcjEOMAwGA1UECBMFSGFudHMxETAP
BgNVBETCFNPMjEgMkpOMQswCQYDVQQGEWJHJQjCBnzANBgkqhkiG9w0BAQEF
AAOBjQAwgYkCgYEAkAth9Ar6k6ijNZ3JxdPGH6yikiwYTuaORZDLZBSpaSEx
4qNKN/CrdF1LgfQYbZcN5NGCeC4sC478NhT+1tf5dnR3pNWbzEzmWn5mN01H
tqJ3oib0UmDui+tQc2J9z6iRBKjkcQwjP1Jp0sp5KKsev1ahAETL7LqmQMiq
pJlzKi0CAwEAAaAAMA0GCSqGSIb3DQEBAUAA4GBAHaMhpizPs8Q3bi3I6dh
4yw0UNhojT1S1+ffiph3hK981MHJumZtr0UMBL1/SZGNw850JRiWuDjGYUQW
inJ0uNH34IUsnygBmt78+W1XT5nJuayg+UrAc5Ao2H8QZpRE5Sfaoc81QcvY
p1TggCdMxpYN7I33LrZD13Po0TT8gjxQ
-----END NEW CERTIFICATE REQUEST-----
```

4. 选择**继续**。
5. 下一页允许您验证证书请求的内容。您还必须提供请求的个人详细信息。如果正在请求完整的“VeriSign 服务器标识”（而不是试用版），则将使用这些详细信息来认证您的申请。
6. 输入了您的详细信息并阅读了 VeriSign 协议之后，请选择**接受**。
7. 下一步（**第 4 步（共 5 步）**）将在用来连接到 SSL 服务器的任何浏览器上安装“测试签署证书”根。在从 Java 小应用程序以 SSL 方式连接到 CICS Transaction Gateway 时，这不是必需的，因为客户机小应用程序需要一个客户机 KeyRing 类，这个类将缺省包含“测试签署证书”。然而，HTTPS 协议使用浏览器中的资源库，因此，您需要安装“测试签署证书”。请注意，如果申请完整的“VeriSign 服务器证书”，则无需安装“测试签署证书”。

“VeriSign 试用服务器证书”将通过电子邮件发送到在个人详细信息中指定的地址。这通常需要 1 到 3 小时。

将“服务器证书”接收到服务器 KeyRing 中

获得了“服务器证书”之后，必须用 iKeyMan 将其“接收”到服务器 KeyRing 中。

1. 首先，用文本编辑器将“服务器证书”数据复制并粘贴到一个空白文本文件中，然后将该文件存为 `verisignServerID.arm`。
2. 从 iKeyMan 密钥数据库内容标签下的下拉菜单选择**个人证书**。
3. 选择**接收...**
4. 找到包含您的“服务器证书”数据的文本文件，该文件应该是 Base64 编码的 ASCII 格式。
5. 选择**确定**。
6. 选择**密钥数据库文件**，然后选择**退出**。

现在, CICS Transaction Gateway 可以使用 **ServerKeyRing.class**。它包含缺省的签署证书和您的“VeriSign 服务器证书”及其相应的专用密钥。

配置 SSL 客户机

在一般(缺省)的操作中, CICS Transaction Gateway 在执行 SSL 握手时只使用服务器认证, 并且客户机只需要接受出示的“服务器证书”。为了使服务器认证正常工作, 客户机 KeyRing 类(或者在 HTTPS 的情况下, 浏览器证书资源库)必须包含“服务器证书”的签署证书。在我们的示例中, 签署证书是“VeriSign 测试签署证书”。

与使用服务器 KeyRing 类一样, 当您使用 iKeyMan 来生成客户机 KeyRing 类时, 它将包含最常用签署证书的一个缺省选择。

除了服务器认证, CICS Transaction Gateway 还支持客户机认证。启用了这个选项之后, 任何尝试到 ssl: 或 https: 处理程序的连接都需要客户机出示自己的“客户机证书”(也称为“数字标识”)。

以下各节描述了如何获得“VeriSign 数字标识”并生成必需的客户机 KeyRing 类。

获得“客户机证书”

与获得“服务器证书”相反, 无需使用 iKeyMan 生成任何形式的证书请求。VeriSign 提供一种使用 Netscape 或 Internet Explorer 浏览器获得“1 类数字标识”的方法。

一旦获得了“客户机证书”并将其安装到浏览器资源库, 就可以将证书数据和相关专用密钥一起导出到称为 #PKCS12 文件的安全 vault 文件中。这个文件是受密码保护的, 并且可以使用 iKeyMan 工具导入到客户机 KeyRing 类。

1. 使用 Netscape Communicator 版本 4.5, 将您的浏览器指向 www.verisign.com, 然后进入链接 **Individual Certificates**。
2. 进入 HTML 页面 <https://digitalid.verisign.com/client/class1Netscape.htm> 之后, 按要求填入详细信息。VeriSign 将使用这些详细信息认证客户机的申请。
3. 选择**接受**。
4. 现在, Netscape 将生成一个专用密钥 - 它将请求一个密码来保护这个专用密钥。
5. VeriSign 将通过电子邮件把下载和安装“1 类数字标识”的信息发送到您在申请时提交的地址。

在安装过程中, Netscape Communicator 允许您将“客户机证书”存储在受密码保护的文件(#PKCS12 格式)中。iKeyMan 工具支持 #PKCS12 格式的文件并允许您将证书(及其专用密钥)导入到客户机 KeyRing 类中。

创建客户机 KeyRing

要创建客户机 KeyRing 类:

1. 启动 ctgikey。
2. 选择密钥数据库文件 --> 新建。
3. 从密钥数据库类型选择 **SSLight** 密钥数据库类。
4. 输入 ClientKeyRing.class 作为文件名。
5. 在位置中, 输入存储您的 ClientKeyRing.class 的合适位置。
6. 选择**确定**。

使用在外部签署的证书

这样就创建了包含如下缺省签署证书的 ClientKeyRing.class:

```
VeriSign Class 1 Public Primary Certification Authority
VeriSign Class 2 Public Primary Certification Authority
VeriSign Class 3 Public Primary Certification Authority
RSA Secure Server Certification Authority
Thawte Personal Basic CA
VeriSign Test CA Root Certificate
Thawte Personal Premium CA
Thawte Premium Server CA
Thawte Personal Freemail CA
Thawte Server CA
```

将“客户机证书”导入到客户机 KeyRing

要导入包含“客户机证书”的 #PKCS12 vault 文件:

1. 从密钥数据库内容标签下面的下拉选择器选择个人证书。
2. 选择导入...
3. 将密钥文件类型设置成 PKCS12 文件。
4. 找到存储的 #PKCS12 文件。
5. 选择确定。
6. 选择密钥数据库文件，然后选择关闭。
7. 选择密钥数据库文件，然后选择退出。

现在，CICS Transaction Gateway 可以使用 **ClientKeyRing.class**。它包含缺省的签署证书和您的“VeriSign 客户机证书（1 类数字标识）”及其相关专用密钥。

使用自签证书 (SSLight)

CICS Transaction Gateway 为您提供一种“自己签署”证书的机制。您将自己设置成自己的“认证中心”，并为服务器 (CICS Transaction Gateway) 和客户机 (浏览器) 端生成 X.509 数字证书。

本节描述如何用证书管理接口 iKeyMan 来配置 SSL 服务器和 SSL 客户机。

配置 SSL 服务器

配置 SSL 服务器涉及以下几个任务: 创建服务器 KeyRing 类、生成自签证书以及将它接收到 KeyRing 类中。

创建服务器 KeyRing

第一步是创建一个服务器 KeyRing 类文件来存放“服务器证书”和密钥信息。

1. 启动 ctgikey。
2. 选择密钥数据库文件 --> 新建。
3. 从密钥数据库类型选择 **SSLight** 密钥数据库类。
4. 输入 ServerKeyRing.class 作为文件名。
5. 在位置中，输入存储您的 ServerKeyRing.class 的合适位置。
6. 选择确定。
7. 输入 KeyRing 文件的密码。

生成“服务器证书”

现在您可以创建自签“服务器证书”并将该证书与其专用密钥一起存储在您的服务器 KeyRing 类中:

1. 从 iKeyMan 密钥数据库内容标签下的下拉菜单选择个人证书。
2. 选择新建自签证书...
3. 必须填写证书的详细信息。有些字段是可选的，但是必须至少填写以下字段（如示例所示）:

密钥标签	exampleServerCert
版本	X509 V3
密钥大小	1024
公共名称	clientmachine.hursley.ibm.com
组织	IBM UK
国家	GB
有效期	365 (天)

4. 选择**确定**。然后，iKeyMan 将生成一个公开/专用密钥对，根据处理器速度的不同，这可能需要一些时间。
5. 当 iKeyman 成功地创建了自签“服务器证书”时，它将出现在“个人证书”窗口中。证书将根据在生成过程中指定的**密钥标签**命名，在本例中是 exampleServerCert。
6. 突出显示 **exampleServerCert**，然后选择**查看/编辑**。请注意，**签发给**和**签发人**文本框中的“证书”信息是相同的，因此，证书请求方（签发给）和签发方（签发人）也是相同的。要与出示该证书的服务器建立 SSL 连接，客户机必须信任 exampleServerCert 的签发人。要这样做，客户机密钥资源库必须包含出示 exampleServerCert 的站点的签署证书。

导出服务器签署证书

现在导出 SSL 服务器的签发人（公开）证书。

1. 突出显示 **exampleServerCert**，然后选择**抽取证书...**
2. 选择导出的证书的数据类型，这通常是 Base64 编码的 ASCII 数据。
3. 输入导出的证书的名称/位置，我们的示例使用 exampleServercert.arm。
4. 选择**确定**。

应该将导出的证书存储在安全的地方，并将其导入到任何需要与特定“SSL 服务器证书”握手的客户机密钥资源库中。

配置 SSL 客户机

如果将 CICS Transaction Gateway 使用的 SSL 处理器配置成只支持服务器认证，则无需生成自签“客户机证书”。在这种情况下，客户机 KeyRing 类只需要包含“服务器”的签署证书，该签署证书是导出的证书文件，在我们的示例中，它是 exampleServercert.arm。

以下步骤描述了创建客户机 KeyRing 以及导入“服务器”签署证书的过程。

创建客户机 KeyRing

1. 启动 ctgikey。
2. 选择密钥数据库文件 --> 新建。
3. 从密钥数据库类型选择 **SSLight** 密钥数据库类。
4. 输入 ClientKeyRing.class 作为文件名
5. 在位置中，输入存储您的 ClientKeyRing.class 的合适位置。
6. 选择**确定**。
7. 输入 KeyRing 的密码。
8. 一旦生成了 ClientKeyRing.class，您就应该看到缺省“签署人”的列表。

导入服务器的签署证书

下一步是导入服务器的签署证书:

1. 选择**添加**。
2. 找到存储的“服务器 Base64 编码的 ASCII”证书文件，在我们的示例中是 exampleServercert.arm。
3. 为这个签署证书授予一个唯一的标签，例如 My Self-Signed Server Authority。
4. 选择**确定**。此时，应该将这个新的签署证书添加到缺省签署人列表中。

生成的 ClientKeyRing.class 可以与配置成支持服务器认证的 CICS Transaction Gateway 的 SSL 协议一起使用。如果使用 HTTPS 协议从小应用程序建立安全连接，则正在运行小应用程序的特定浏览器需要将 exampleServercert.arm 导入到它的密钥/证书资源库中。

有关如何导入 Base64 编码的 ASCII 证书文件的信息，请参考您的浏览器联机帮助。

生成“客户机证书”

客户机认证要求客户机 KeyRing 类也包含一个用来标识所连接客户机的自签“证书”。

遵循与生成自签“服务器证书”相同的步骤来创建（或打开现有的）客户机 KeyRing 类（在我们的示例中是 ClientKeyRing.class），然后:

1. 从 iKeyMan 密钥数据库内容标签下的下拉菜单选择**个人证书**。
2. 选择**新建自签证书...**
3. 填写证书的详细信息。
4. 选择**确定**。然后，iKeyMan 将生成一个公开/专用密钥对，根据处理器速度的不同，这可能需要一些时间。
5. 与 SSL 服务器一样，客户机需要将其签署证书安装在 SSL 服务器的密钥/证书资源库中。这就允许 SSL 服务器验证客户机的详细信息。突出显示 **exampleClientCert**，然后选择**抽取证书...**
6. 选择导出的证书的数据类型，这通常是 Base64 编码的 ASCII 数据。
7. 输入导出的证书的名称/位置，我们的示例使用 exampleClientcert.arm
8. 选择**确定**。

应该将导出的证书存储在安全的地方，并将其导入到任何需要与特定“SSL 客户机证书”握手的客户机密钥资源库中。

限制对服务器 KeyRing 的访问

服务器 KeyRing 的内容是用密码加密的，然而强烈建议您：

- 确保设置了正确的文件许可权
- 在适当情况下，限制对 CICS Transaction Gateway 机器的访问。

在服务器之间共享证书不是好习惯。您不希望服务器共享专用密钥，特别是当它们运行在不同的机器上时更是如此。永远不应该将专用密钥泄漏给别人。

将 CICS Transaction Gateway 配置为 SSL 和 HTTPS

CICS Transaction Gateway 支持与其客户机通信的各种协议，包括 TCP、HTTP、SSL 和 HTTPS。要使用 SSL 和 HTTPS 协议，必须使用配置工具来启用它们（请参阅第19页的『使用配置工具』）。然后，当执行 CICS Transaction Gateway 时，将启动 SSL 和 HTTPS 协议。您可以指定只使用任何一个安全性协议或者使用两个协议。通过这些指定的处理程序，当 CICS Transaction Gateway 启动时，它在端口 8050 上侦听 SSL 请求，在端口 443 上侦听 HTTPS 请求。

特定于 SSL 和 HTTPS 协议的设置是：

KeyRing 类名 该设置指定服务器 KeyRing 类文件的名称。必须设置 CLASSPATH 环境变量，以便可以找到该类。

KeyRing 密码 该设置指定用于解密已加密的服务器 KeyRing 的密码。

使用客户机认证

该设置指定要使用客户机认证。缺省值是使用服务器认证。

使用配置工具将在配置文件中创建必需的项。样本配置文件 CTGSAMP.INI 中包含了 SSL 和 HTTPS 协议项的全部示例。

CICS Transaction Gateway 提供了两种用于建立 SSL 和 HTTPS 连接的缺省 KeyRing 类文件。**ClientKeyRing** 和 **ServerKeyRing** 都是使用密码 **default** 加密的，并且建议只在测试环境中使用。因此，要使用 SSL 和 HTTPS 协议，我们建议您如前面章节中描述的那样生成您自己的 KeyRing。

指定客户机 KeyRing

使用哪个安全协议将确定是否需要客户机 KeyRing。HTTPS 协议被设计成从 Java 小应用程序内部进行安全通信，浏览器（客户机）本身具有建立与 CICS Transaction Gateway（服务器）的安全连接的必要功能。出于这种原因，HTTPS 协议只需要指定服务器方的 KeyRing，客户机方由浏览器软件处理。

SSL 协议是在较低级别上设计的，CICS Transaction Gateway 在此级别中具有以安全方式处理服务器和客户机的代码。SSL 协议需要服务器和客户机的 KeyRing 类文件。

客户机 KeyRing 是通过设置 SslJavaGateway.class 中的静态字段指定的。该类构成 CICS Transaction Gateway 客户机端代码部分。

SslJavaGateway.class 提供两种方法：一种用于“获取”，另一种用于“设置”客户机 KeyRing：

```
public static void setKeyRing(String strSetKeyRing, String strSetKeyRingPW)
public static String getKeyRing()
```

使用自签证书

要将客户机 KeyRing 类设置成由 SSL 协议使用，客户机应用程序或小应用程序将静态调用下列方法：

```
SslJavaGateway.setKeyRing(CLASSname, PASSword);
```

其中：

- CLASSname 表示为客户机生成的 Java KeyRing 类的类名
- PASSword 用于译码嵌入的 X.509 证书。

SslJavaGateway.class 还提供“getter”方法 **getKeyRing()** 来返回当前指定的客户机 KeyRing 的 CLASSname。

使用 SSL/HTTPS 协议来建立从客户机应用程序或小应用程序到 CICS Transaction Gateway 的连接与使用 TCP 或 HTTP 协议来建立的相同。客户机应用程序或小应用程序使用相关的 URL 将它的请求“流到” CICS Transaction Gateway。例如，对于 SSL，应用程序将使用 `ssl://transGatewayMachine:8050`，或者对于 HTTPS，它将使用 `https://transGatewayMachine:443`。

有关设计和实现客户机端程序的进一步信息，请参阅 *CICS Transaction Gateway: Gateway Programming* 一书和 CICS Transaction Gateway 编程接口 HTML 页面。

第6章 CICS Transaction Gateway 操作

本章描述:

- 『启动 Gateway』
- 第61页的『停止 Gateway』

启动 Gateway

在安装了 Gateway 的计算机的操作系统命令提示处启动 Gateway。

首先, 将工作目录设置为 CICS Transaction Gateway 的 bin 目录。

可以用两种方法启动 CICS Transaction Gateway:

- 『用预置选项启动 Gateway』
- 『用用户指定的选项启动 Gateway』

用预置选项启动 Gateway

要用预先定义的选项启动 Gateway: 在命令提示处输入 `ctgstart` 并按 Enter 键, 或选择 **CICS Transaction Gateway** 图标。

当您这样启动 Gateway 时, 将使用缺省配置设置或使用配置工具配置的设置。(请参阅第19页的『使用配置工具』)。

启动 Gateway 控制台会话并显示下列消息:

```
CCL6500I: Starting the Gateway with default values.
```

接下来的行显示正在使用的值:

```
CCL6502I: [ Initial ConnectionManagers = 1, Maximum ConnectionManagers = 100,  
CCL6502I: Initial Workers = 1, Maximum Workers = 100, Port = 2345 ]
```

用用户指定的选项启动 Gateway

启动命令中用户可定义的选项是:

```
-port=port_number  
-initconnect=number  
-maxconnect=number  
-initworker=number  
-maxworker=number  
-trace  
-noinput  
-tfile=pathname  
-x  
-tfilesize=number  
-truncationsize=number  
-dnsnames  
-dumpoffset=number  
-stack
```

其中:

-port

指定用于网关侦听的 TCP/IP 端口号。

-initconnect

指定 ConnectionManager 线程的初始数。

-maxconnect

指定 ConnectionManager 线程的最大数目。如果将该值设置为 *-1*，则 ConnectionManager 线程数将没有限制。

-initworker

指定 Worker 线程的初始数。

-maxworker

指定 Worker 线程的最大数目。如果将该值设置为 *-1*，则 ConnectionManager 线程数将没有限制。

-trace

启用标准跟踪（请参阅第80页的『跟踪』）。缺省情况下，跟踪输出只显示所有数据块（例如 COMMAREA 或网络流）的前 128 个字节。其它有用信息（包括 CLASSPATH 变量的值和代码页）显示在跟踪输出的顶部。

如果使用配置工具来配置 Gateway，则跟踪输出的缺省目的地就是安装 CICS Transaction Gateway 的 bin 子目录中的文件 CTG.TRC。否则，跟踪输出写入 stderr。

-noinput

禁止从控制台读取输入。

-tfile=pathname

如果启用跟踪，跟踪输出将被写入在 *pathname* 中指定的文件。这会覆盖跟踪输出的缺省目的地（请参阅 **-trace** 选项）。

-x 启用完全调试跟踪（请参阅第80页的『跟踪』）。缺省情况下，跟踪输出显示所有数据块（例如 COMMAREA 或网络流）的全部字节。它还显示比标准跟踪多的内部 Gateway 处理信息。有关跟踪输出的目的地信息，请参阅 **-trace** 和 **-tfile** 选项。

调试跟踪会显著地降低性能。

-tfilesize=number

指定跟踪输出文件的最大大小（以千字节为单位）。

-truncationsize=number

值 *number* 指定在跟踪中显示的任何数据块的最大大小。可以将该选项与 **-trace** 或 **-x** 选项一起使用来覆盖缺省大小。任何正整数都是有效的。如果指定了值 *0*，则在跟踪中不会显示任何数据块。

-dnsnames

允许在消息中显示带符号的 TCP/IP 主机名。请参阅第23页的『显示 TCP/IP 主机名』，以获取更多信息。

-dumpoffset=number

值 *number* 指定自任何数据块显示开始的偏移量。如果偏移量大于要显示的数据总长，则将使用偏移量 *0*。

-stack

启用 Java 异常堆栈跟踪（请参阅第80页的『跟踪』）。跟踪 Java 异常，包括在 Gateway 的正常操作期间预期发生的异常。不会创建任何其它跟踪输出。有关跟踪输出的目的地信息，请参阅 **-trace** 和 **-tfile** 选项。

要重设启动缺省值，可在命令提示处输入 **ctgstart**，后跟您需要的启动选项并按 Enter 键。命令行上指定的选项覆盖使用配置工具指定的那些选项。

显示下列启动消息：

```
CCL6501I: Starting the CICS Transaction Gateway with user specified values.
```

接下来的行显示正在使用的值，例如：

```
CCL6502I: [ Initial ConnectionManagers = 10, Maximum ConnectionManagers = 100,  
CCL6502I: Initial Workers = 10, Maximum Workers = 100, Port = 2345 ]
```

要获得启动选项的帮助，可输入：

```
ctgstart ?
```

停止 Gateway

要停止 Gateway：

- 如果没有用 **-noinput** 选项启动 Gateway，则可以通过在 Gateway 控制台会话中输入正确的字符并按 Enter 键来停止它。允许输入的字符可以是您的国家语言字符；允许的缺省字符是 Q 或 - 字符。

您只要在 Gateway 控制台会话中按 Enter 键，就可确定哪些字符将停止 Gateway。消息显示如下：

```
CCL6508I: Type Q or - to stop the CICS Transaction Gateway.
```

- 如果使用了 **-noinput** 选项，则必须使用一些其它方法来停止 Gateway 进程，例如：
 - 使用 **kill** 命令

操作

第7章 CICS Transaction Gateway 终端小服务程序

本章描述了如何使用“终端小服务程序”通过 Web 浏览器来访问 CICS 3270 应用程序。

- 『什么是“CICS Transaction Gateway 终端小服务程序”？』介绍了小服务程序的概念，特别介绍了“CICS Transaction Gateway 终端小服务程序”执行的功能。
- 第64页的『安装和配置“终端小服务程序”』讨论了如何安装和配置“终端小服务程序”。
- 第69页的『使用“终端小服务程序”』描述了使用“终端小服务程序”可执行的操作，以及如何从 Web 页面中调用它。
- 第73页的『特性和参数参考』针对可以为“终端小服务程序”指定的小服务程序特性提供了一个完整参考。
- 第77页的『CICS® Transaction Server OS/390® 版 Web 界面』描述了“终端小服务程序”如何使用生成的由“CICS Web 界面”使用的 HTML 模板。

建议在配置和使用“终端小服务程序”之前先通读本章中的所有内容。

什么是“CICS Transaction Gateway 终端小服务程序”？

小服务程序是在 Web 服务器机器上运行的 Java 程序,在其内部有一个启用 Java 的 Web 服务器或小服务程序引擎,例如 WebSphere Application Server (请参阅第64页的图8)。Java 小服务程序现已广为使用并成为“公共网关接口 (CGI)”程序的替代产品 -- “终端小服务程序”就是随“IBM CICS 客户机版本 2”一起交付的 CICS Internet Gateway (它就是 CGI 程序)的替代产品。

小服务程序会在启动 Web 服务器时自动装入,或在客户机第一次请求小服务程序的服务时装入。与 CGI 程序不同的是,小服务程序是具有持久性的,即一旦装入,它们将保持运行以等待附加的客户机请求。这使得小服务程序的运行速度比 CGI 程序快,因为没有时间浪费在创建和破坏进程上。

作为 Java 程序,小服务程序可以跨操作系统移植。并且由于小服务程序接口是标准的,所以它们可以用于各种不同平台上的不同 Web 服务器中。

“终端小服务程序”允许您使用 Web 浏览器来访问 CICS 3270 应用程序。您可以创建调用“终端小服务程序”的 Web 页面来启动 CICS 事务,显示 CICS 服务器发送的屏幕信息并将屏幕发送回 CICS。

“终端小服务程序”可以:

- 作为一个简单的终端仿真器运行。

CICS 屏幕是作为 HTML 表单的一部分显示在浏览器中的。当按下表单中的一个按钮时,会提交表单并将数据发送到 CICS。
- 将来自 CICS 屏幕的数据替换进 HTML 模板文件。

HTML 模板文件用来确定如何在浏览器中显示 CICS 数据。“终端小服务程序”用 CICS 屏幕信息替换 HTML 模板文件中的变量,并将生成的输出发送到浏览器。如需更多信息,请参阅第71页的『使用变量替换』。
- 使用服务器引用、合并 Web 页面中的变量信息或驱动没有用户交互的终端仿真器。

CICS Transaction Gateway 终端小服务程序

服务器引用是 HTML 文件 (.shtml) 中的语句，在将 Web 页面返回到浏览器之前，Web 服务器会处理这些语句。例如，可以使用服务器引用来调用“终端小服务程序”以显示 CICS 屏幕中的字段值。如需更多信息，请参阅第70页的『使用服务器引用调用小服务程序』。

- 将 CICS 屏幕映射到 Web 页面。

使用“终端小服务程序”的页面映射特性，可以将特定的 Web 页面与特定的 CICS 屏幕相关联。例如，可以指定在终端空闲并且没有运行事务时显示的 Web 页面。如需更多信息，请参阅第75页的『页面映射特性』。

与 EPI 编程类似，“终端小服务程序”允许为现有的 CICS 事务创建新的前端。当然，它还允许您充分发挥 Java 编程的强大功能、灵活性以及平台无关性。

注：“终端小服务程序”不支持 3270 数据流中的 DBCS 字段。

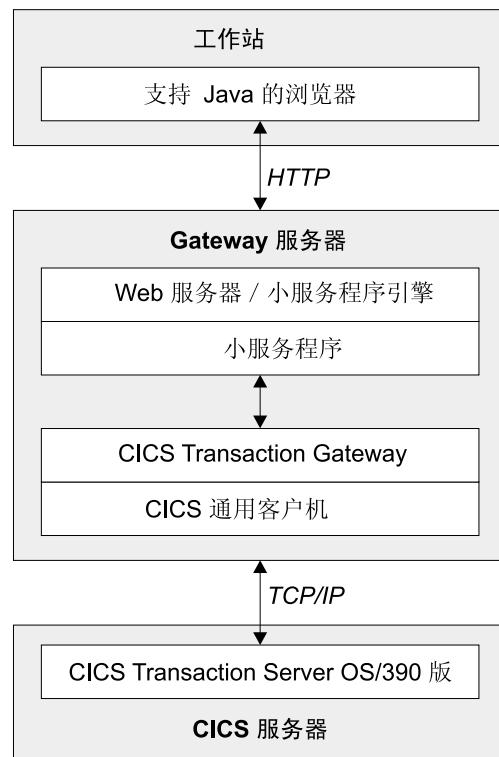


图 8. CICS Transaction Gateway 使用由 URL 调用的“终端小服务程序”

安装和配置“终端小服务程序”

本节讨论了如何在 Web 服务器机器上安装“终端小服务程序”和设置“终端小服务程序”的初始化参数。

根据 Web 服务器的不同，安装和配置“终端小服务程序”的过程也不相同。但是，至少需要执行下列操作：

- 配置 Web 服务器的 CLASSPATH 和 PATH 设置

- 将“终端小服务程序”添加到 Web 服务器的配置中
- 配置小服务程序初始化参数
- 考虑其它配置选项，例如安全性、记录日志以及 Web 服务器是否能够处理服务器引用。

在任何可用的配置文档中都给出了在特殊的 Web 服务器上安装和配置“终端小服务程序”的示例，请参阅第93页的『样本配置文档』。在该文档中描述的原理也应该适用于其它 Web 服务器。

不管使用的 Web 服务器是什么，都应该参考它的文档来指导安装和配置小服务程序。

配置 Web 服务器的 CLASSPATH 和 PATH 设置

必须确保两个 CICS Transaction Gateway jar 文件 (**ctgclient.jar** 和 **ctgserver.jar**) 位于 Web 服务器的类路径中。如果使用的是 WebSphere Application Server，则应该在 WebSphere 的“管理”工具的“Java 引擎”部分中执行此操作。

如果没有正确设置类路径，那么可能会获得一条消息，内容是您无法更新小服务程序，并且无法装入类 **com.ibm.ctg.servlet.TerminalServlet**。

还必须确保 CICS Transaction Gateway bin 子目录位于 Web 服务器的路径中。

将“终端小服务程序”添加到 Web 服务器的配置中

要将“终端小服务程序”添加到 Web 服务器的配置中，必须提供一个“小服务程序名称”并将这个小服务程序实例与“终端小服务程序”类 (**com.ibm.ctg.servlet.TerminalServlet**) 关联起来。如果使用的是 WebSphere Application Server，则应该在 WebSphere 的“管理”工具的“小服务程序配置”部分中执行此操作。

可以配置多个“终端小服务程序”的实例。每个实例都将有一个不同的“小服务程序名称”，但是将引用相同的“小服务程序类”。实际上，必须为要连接的每个 CICS 服务器都创建一个不同的“终端小服务程序”实例。“终端小服务程序”的每个实例可以使用不同的初始化参数，这当配置不同的 CICS 服务器时很有用。

配置小服务程序初始化参数

在将“终端小服务程序”添加到 Web 服务器的配置之后，必须设置“终端小服务程序”的初始化参数。如果使用的是 WebSphere Application Server，则应该在 WebSphere 的“管理”工具的“小服务程序配置”部分中执行此操作。

以下章节深入介绍了“终端小服务程序”是如何工作的，这将帮助您设置适当的小服务程序初始化参数。

“终端小服务程序”现在支持有注册功能的终端。有关可用的初始化参数（包括那些提供此类型终端的参数）的完整集合，请参考第73页的『特性和参数参考』。

CICS Transaction Gateway 终端小服务程序

考虑设置下列参数中的一部分作为起点:

表 4. 小服务程序初始化参数

参数名称	注释
servlet@propertyFile	小服务程序可以从特性文件中装入配置信息。在子目录 /samples/terminalervlet 中提供了样本 <code>servlet.properties</code> 文件。 要使用特性文件, 请将这个参数设置为希望“终端小服务程序”装入的特性文件的全路径名称。如果指定了特性文件, 那么可能希望将这个文件用于所有其它的小服务程序参数。如果是这样, 那么 servlet@propertyFile 就是唯一需要指定的参数。
servlet@extendedLogging	小服务程序将记录信息(例如错误消息)写入标准的小服务程序日志。将这个特性设置为 <code>true</code> 以将更多信息写入日志文件。在第一次配置小服务程序时您可能会发现这个参数很有用, 但是一旦正确设置了小服务程序之后就应该将它关闭。
pool@maxTerminals	将这个特性设置为允许的与 CICS 服务器的最大连接数。应该确保 CICS 服务器有足够的空闲任务来处理这个数目的并行连接。还应该确保“CICS 客户机”的 最大请求数 配置设置至少与这个特性一样大。
pool@serverName	“终端小服务程序”的每个实例仅可以连接到一个 CICS 服务器。这给予您对资源用法和安全性的更多控制权。将这个特性设置为必需的 CICS 服务器的名称, 这个名称与您的配置中由适当的 服务器名称 设置定义的一致。如果没有设置这个特性, 将使用缺省的 CICS 服务器。
pool@gatewayUrl	如果希望小服务程序通过网络连接到 CICS Transaction Gateway, 则设置这个特性。

除特性文件名称之外, 所有小服务程序特性都可以从特性文件装入或作为小服务程序的初始化参数设置。初始化参数覆盖特性文件项。

小服务程序特性名是不区分大小写的。

终端池

“终端小服务程序”使用终端池, 并且小服务程序配置特性允许您控制终端池的行为。缺省行为是: 无论何时用户需要, 都会建立一个新的终端连接到 CICS, 直到到达 **pool@maxTerminals** 限制为止。当用户使用完终端后, 就被断开连接。

如果设置了 **pool@minTerminals** 特性, 那么在小服务程序启动时会有该数目的终端连接到 CICS, 并且在任意时刻最多只能有这么多数量的不在使用的终端连接到 CICS。这就允许立即为新用户分配一个终端而不需要等待连接。

设置 **pool@reusingTerminals** 特性允许重用终端, 即, 当用户会话结束时将终端分配给一个新用户而不是断开连接。如果 **pool@minTerminals** 是 0, 那么这个特性不起作用。在重用终端之前, 将终止终端上运行的任何事务并且清除屏幕。但是, 如果用户已经注册到 CICS (例如使用 CESN), 那么当重用终端时, 终端将仍然注册。出于这个原因, 不要设置这个特性, 除非您确信它适用于您的环境。

pool@idleTimeout 特性允许在一段指定的超时周期后, 如果还没有使用已经分配给用户的终端, 则释放这些终端。这个特性缺省值为 0, 即没有超时。

将值 **pool@connectedTerminals**、**pool@freeTerminals** 和 **pool@terminalsInUse** 用于显示请求，您可以查询连接的、空闲或用于“终端小服务程序”的某个特定实例的终端数。

页面映射

页面映射告诉“终端小服务程序”将为特定的 CICS 屏幕或终端状态显示哪个 Web 页面。

小服务程序选择显示的页面是基于当前的页面映射设置、当前的屏幕处理程序类和终端状态。

“终端小服务程序”首先查询页面映射特性以查看是否为当前会话状态设置了特定页面，如下所示：

会话状态	页面映射特性
发生了错误	page@error
终端空闲 - 没有运行事务	page@idle
终端已断开连接	page@disconnected
当前屏幕有屏幕处理程序	屏幕处理程序的类名称的页面映射

如果没有找到特定页面，将使用页面映射特性 **page@default** 标识的页面。如果没有设置这个特性，那么会显示一个错误消息。

有两种设置页面映射特性值的方式：

1. URL，例如：

```
http://webserver/index.html
```

2. 要装入的模板文件，用于变量替换。这必须以“file://”开始，例如：

```
file://d:/pages/template.html
```

如果设置了 **servlet@templateDir** 特性，那么模板文件名可以指定为模板目录的相对名。

设置页面映射的示例如下：

```
page@error=http://server/errors.html
page@idle=http://server/idle.html
page@disconnected=/ctglab/html/servlet/epissam.html
page@default=file://epissam1.html
```

要设置特定屏幕的页面映射，必须创建一个可识别屏幕的“屏幕处理程序 bean”。然后，必须设置 **pool@handlerPath** 特性以确保“终端小服务程序”装入了“屏幕处理程序 bean”。例如，如果您在名为 testmaps 的包中创建了一个名为 MAP1ScreenHandler 的“屏幕处理程序 bean”，那么该屏幕处理程序的页面映射特性是：

```
page@testmaps.MAP1ScreenHandler.
```

类名是区分大小写的。对于标准页面映射，将特性设置为要显示的 Web 页面的名称，例如：

```
page@testmaps.MAP1ScreenHandler=http://server/test.html
```

CICS Transaction Gateway 终端小服务程序

页面映射还可以在“终端小服务程序”的请求参数中设置，在这种情况下，它仅应用于该请求并且覆盖与正在使用的小服务程序实例关联的任何页面映射。可以使用适当的请求来查询页面映射的当前设置（请参阅第76页的『可显示特性』）。

“屏幕处理程序 bean”和断开终端连接

要成功地断开与终端的连接，必须退出运行事务。“终端小服务程序”实现这个目的的缺省行为是发送 AID（注意标识符）PF3 键到 CICS。但是，如果您将运行的事务无法使用这个行为，那么应该创建一个或多个知道如何退出适当屏幕的“屏幕处理程序 bean”。

可以使用与 CICS Transaction Gateway 一起提供的“BMS 映射转换实用程序 (BMSMapConvert)”来从 BMS 映射自动生成“屏幕处理程序 bean”，或编写自己的类。有关“屏幕处理程序”和如何创建的信息，请参阅 *CICS Transaction Gateway: Gateway Programming* 一书。

要将屏幕处理程序装入小服务程序，请设置 **pool@handlerPath** 特性。尽管将不搜索子目录，但处理程序路径可以包含 .jar 文件、.zip 文件和目录。例如，在小服务程序特性文件中，可以添加：

```
pool@handlerPath=d:/handlers/handlers.jar;d:/handlers/test
```

以从 .jar 文件或目录中装入屏幕处理程序类。有关设置这个特性的更多信息，请参阅样本特性文件。

还可以指定缺省屏幕处理程序（使用特性 **pool@defaultHandler**），如果没有其它屏幕处理程序能够识别屏幕时就使用它来退出屏幕。缺省屏幕处理程序的样本是 **com.ibm.ctg.epi.DefaultScreenHandler**，它发送 AID PF3 到 CICS。在目录 `samples/java/com/ibm/ctg/epi` 下可以找到这个类的源。任何可从类路径或处理程序路径装入的屏幕处理程序 bean 都可用作缺省的屏幕处理程序。

小服务程序不会无休止地尝试退出运行的事务。特性 **pool@exitRetryLimit** 控制尝试退出事务的次数。这个特性的缺省值是 10，但应该确保它对于您的环境足够大。如果在尝试退出这么多次数之后，小服务程序发现终端上仍然有事务在运行，那么会假设这个终端是“死的”并且不会将它断开或分配给其他用户。废弃死终端的唯一方法是卸载小服务程序、停止并重新启动 CICS Transaction Gateway 并再次装入小服务程序。

如果已经正确配置了小服务程序并且提供了“屏幕处理程序 bean”，那么就不应该有任何死终端。

考虑其它配置选项

其它应该考虑的配置选项有：

- **安全性**
可能需要它们来控制对“终端小服务程序”的访问。
- **记录日志**
“终端小服务程序”将日志信息写入标准的小服务程序日志。可以打开或关闭记录日志功能。
- **会话跟踪**
应该配置“终端小服务程序”来使用会话跟踪。
- **服务器引用**

可能希望配置您的 Web 服务器以处理服务器引用。

注：IBM WebSphere Application Server, 版本 1.01 不完全支持小服务程序的 DBCS 数据输入或输出, 其它小服务程序引擎可能也有相同的限制。在这种情况下, “终端小服务程序”将 DBCS 字符显示为 ???? (问号)。

装入“终端小服务程序”

在配置了“终端小服务程序”并保存设置之后, 必须装入它。可以选择是立即装入还是在每次 Web 服务器启动时装入。

如果没有正确配置“终端小服务程序”, 那么当 Web 服务器试图装入“终端小服务程序”时会显示错误消息。有关更多信息, 请参考 *CICS Transaction Gateway: Gateway Messages* 一书的“终端小服务程序”一节。

使用“终端小服务程序”

本节描述了如何使用“终端小服务程序”。它描述了能够调用“终端小服务程序”的不同方式, 以及如何开发使用“终端小服务程序”的 Web 页面。

在 /samples/terminalervlet 子目录中提供了样本文件。在文件 /samples/Samples.txt 中描述了这些样本。

连接到 CICS 并启动事务

“终端小服务程序”的一个实例仅可以连接到一个 CICS 服务器。对于允许访问的每个 CICS 服务器, 都需要创建一个小服务程序的实例。可能还需要创建具有不同初始化参数的小服务程序实例。

要启动 CICS 服务器上的事务, 必须使用一个合适的请求来调用“终端小服务程序”。

调用“终端小服务程序”

有三种调用“终端小服务程序”的方式:

- 使用 URL
- 使用 HTML 表单
- 使用服务器引用

在每种情况下, 都需要知道要使用的小服务程序实例名称, 并将一个或多个请求参数传递到小服务程序来告诉它要采取的操作。所有情况下的参数名称和值都是相同的, 只是指定它们的方式不同。

要启动 CICS 服务器上的事务, 必须提供的参数有:

参数名称	参数值
request	send
transaction	要启动事务的事务标识。 但注意不能为“终端小服务程序”启动 ATI 事务。

如果需要, 小服务程序将为用户分配终端。

使用“终端小服务程序”

请求参数的完整集合列在第75页的『请求参数』中。

通过 URL 调用小服务程序

要调用名为 TerminalServlet 的小服务程序实例，链接到 URL:

```
http://your_webserver/servlet/TerminalServlet
```

其中 *your_webserver* 是 Web 服务器的主机名。

可以在 URL 中以查询字符串的格式编码请求参数，例如:

```
http://your_webserver/servlet/TerminalServlet?request=send&transaction=CECI
```

当不需要用户输入任何信息时，应该使用这个方法调用小服务程序。

使用 HTML 表调用小服务程序

使用这种方法，在 Web 页面中创建一个 HTML 表单。例如:

```
<FORM METHOD="GET" ACTION="/servlet/TerminalServlet">  
(Tags to place text entry areas, buttons, and other prompts go here)  
</FORM>
```

就是一个表单，它将调用名为 TerminalServlet 的小服务程序。METHOD 可以是 GET 或 POST。表单的各种元素都具有名称和值，并且当提交表单时将它们发送到小服务程序。如果您知道要设置一个特定的请求参数，那么将它作为隐藏项添加在表单中，例如: `<input type="hidden" name="request" value="send">`。隐藏输入总是发送到小服务程序。通常，表单元素的名称就是请求参数的名称，它的值就是请求参数的值。

这是将用户输入信息发送到小服务程序的唯一方式。

提供的样本 `epissam3.html` 说明了 HTML 表单方法; 请参阅产品 CD 上的 `Samples.txt` 文件。

使用服务器引用调用小服务程序

服务器引用是在将 Web 页面发送到用户之前由 Web 服务器处理的。调用小服务程序，并且它生成的任何输出都被包含在 Web 页面中以替换服务器引用标记。用户仅看到 Web 页面中的小服务程序输出。

在 HTML 源中，小服务程序的服务器引用看上去类似于:

```
<SERVLET NAME="TerminalServlet" >  
<PARAM NAME="request" VALUE="send">  
<PARAM NAME="transaction" VALUE="CECI">  
<PARAM NAME="display" VALUE="none">  
</SERVLET>
```

可以使用服务器引用将变量信息添加到 Web 页面，或在显示一个 Web 页面时驱动终端执行某些操作，而无需用户介入。

可以在 Web 页面中添加任意多的服务器引用，但是**请注意**:

- 一页中的服务器引用可以并行处理。因此可能无法确保处理它们的次序，尽管通常是以在源中出现的次序处理的。
- 小服务程序存储允许它访问终端（分配给用户的终端）的会话信息。如果用户会话是通过服务器引用调用的，那么相同页面中随后的服务器引用将不能访问会话信息。一旦将页面发送给用户之后，将建立会话，并且具有多个服务器引用的页面将正确工作。

提供的样本 epissam2.shtml 说明了服务器引用方法；请参阅产品 CD 上的 Samples.txt 文件。

下一步将发生什么？

当从表单或通过链接到 URL 来调用小服务程序时，必须将一些输出发送回浏览器。为了决定应该向用户显示哪些内容，小服务程序使用您已经设置的页面映射特性，请参阅第67页的『页面映射』。

显示屏幕和字段

有两种方法可将 CICS 屏幕信息包含在 Web 页面中：

- 服务器引用
- 变量替换。

如果需要，可以混合使用两种方法。

使用服务器引用

必须配置您的 Web 服务器以处理服务器引用。创建包含调用小服务程序的服务器引用的 SHTML 页面。小服务程序的输出将被插入到页面中服务器引用的位置处。

例如，要显示屏幕（作为 HTML 表单的一部分）：

```
<SERVLET NAME="TerminalServlet" >
<PARAM NAME="display" VALUE="screen">
</SERVLET>
```

或在屏幕上显示第 22 个字段的内容：

```
<SERVLET NAME="TerminalServlet" >
<PARAM NAME="display" VALUE="22">
</SERVLET>
```

使用变量替换

创建小服务程序将装入的 HTML 模板文件。小服务程序用 CICS 屏幕信息替换文件中的变量，并将生成的输出发送到浏览器。使用页面映射特性来告诉小服务程序要装入的模板文件（请参阅第67页的『页面映射』）。

例如，CESN 注册屏幕的模板文件可能会如下所示：

```
<html>
<head>
<title>CICS Signon</title>
</head>
<body>
<h1>CICS Signon</h1>
<form method="POST" action="/servlet/TermServlet">
<input type="hidden" name="request" value="send">
<pre>
  Userid: <input type="text" name="USERID" value="&USERID;" size="8" >
  Password: <input type="text" name="PASSWORD" value="&PASSWORD;" size="8" >
</pre>
<input type="submit" name="DFH_ENTER" value="Sign on">
<input type="submit" name="DFH_PF1" value="Help">
<input type="submit" name="DFH_PF3" value="Exit">
</form>
</body></html>
```

使用“终端小服务程序”

其中，变量是 `&USERID;` 和 `&PASSWORD;`。可以使用数字来标识字段 - 要象本例中那样使用字段名称，您必须有“屏幕处理程序 bean”来识别屏幕并根据名称设置字段。

页面映射特性可以是 HTML 模板文件名称或任何 URL。应该将页面映射特性设置为值 `file://template_filename`（例如：`file://d:/html/templates/test.html`），这样小服务程序就会知道该特性标识一个 HTML 模板。

可以显示哪些内容？

任何可以由服务器引用显示的信息片段也都可以用于变量替换，反之亦然。模板中使用的变量名与服务器引用中 **display** 参数的值相同。

通常，可以显示：

- 屏幕，作为 HTML 表单中的一部分
- 字段 - 由数字标识或由名称标识（如果具有“屏幕处理程序 bean”）
- 小服务程序配置设置
- 当前连接的、正在使用的以及空闲的终端数

可显示特性的完整集合列在第73页的『特性和参数参考』中。

将屏幕发送回 CICS

要将屏幕发送回 CICS，必须调用小服务程序并将 **request** 参数设置为“send”。不需要提供 **transaction** 参数 - 如果事务已在运行，将忽略这个参数。要在将屏幕发送回 CICS 之前先用新的信息更新屏幕，需要进一步指定标识字段（通过名称或数字）的请求参数以及它们的值。

例如，以下的服务器引用：

```
<SERVLET NAME="TerminalServlet" >
<PARAM NAME="request" VALUE="send">
<PARAM NAME="USERID" VALUE="sysad">
<PARAM NAME="PASSWORD" VALUE="sysad">
<PARAM NAME="display" VALUE="none">
</SERVLET>
```

将名为 USERID 和 PASSWORD 的字段设置为给定值，然后将屏幕发送到 CICS。在 HTML 表单中，可以使用表单元素 `<input type="text" name="USERID" value="" size="8">`。当用户提交表单时，请求参数 USERID 被设置为输入的值。

设置 AID

缺省的 AID 是 Enter。

要从 HTML 表单中设置 AID，需包含一个 Submit 按钮，可如下指定：

```
<input type="submit" name="DFH_PF3" value="Exit" >。
```

如果用户按下按钮，会将请求参数“DFH_PF3”以及值“Exit”（小服务程序将忽略该值）发送到小服务程序。AID 被设置为 PF3。在服务器引用中，可以如下所示设置 AID：

```
<SERVLET NAME="TerminalServlet" >
<PARAM NAME="request" VALUE="send">
<PARAM NAME="DFH_PF3" VALUE="ignored">
<PARAM NAME="display" VALUE="none">
</SERVLET>
```


请求参数的完整集合列在『特性和参数参考』中。

断开连接

当小服务程序为用户分配了终端之后，会保留这个终端直到：

- 使用设置为“disconnect”的“request”参数调用小服务程序
- 在 Web 服务器上的用户会话到期
- 如果设置了 **pool@idleTimeout** 特性，那么在一段指定的时间周期内没有访问终端，终端就会自动断开连接。
- “CICS 客户机”停止
- 与 CICS 服务器的连接丢失 - 或服务器关闭
- 小服务程序被 Web 服务器卸装。

当不再需要终端时，应该尝试显式地释放终端。

注意，当用户建立了与小服务程序的会话并分配到终端后，用户访问的任何小服务程序实例都能够继续会话。在用户断开连接或出于一些其它原因终止会话之前都使用同一个终端。用于任何特定请求的页面映射和其它特性就是为正在处理请求的小服务程序实例设置的那些特性。

能够设置为请求参数的特性（例如页面映射）会覆盖小服务程序的设置，但通常仅应用到当前请求，并且不影响其它用户发出的请求或同一个用户随后发出的请求。

特性和参数参考

本节描述了“终端小服务程序”特性

小服务程序配置特性

通常，当配置 Web 服务器时这些特性可在特性文件中指定或作为小服务程序初始化参数指定。名称是不区分大小写的。作为小服务程序初始化参数给定的值会覆盖从特性文件装入的设置。除 **servlet@debug** 和 **servlet@extendedLogging** 特性之外，当小服务程序运行时不能更改这些设置。但是，使用请求参数 **display**，可以查询特性的当前设置，请参阅第76页的『可显示特性』。

servlet@coloring

如果设置为 Y，彩色字段将以 3270 数据流中指示的颜色显示。

由于 HTML 中的限制，未保护字段不能是彩色并且总是显示为白底黑字。

如果没有指定这个特性，“终端小服务程序”会以白底黑字方式显示屏幕。

servlet@debug

如果设置为 true，那么将为小服务程序的所有实例打开跟踪。当小服务程序运行时可以打开或关闭跟踪，输出被定向到标准输出。注意，打开跟踪会严重地影响性能！

servlet@extendedLogging

如果设置为 true，会打开扩展记录。输出定向到小服务程序日志。

servlet@propertyFile

要装入的特性文件的全路径名称。这个特性仅在设置为小服务程序初始化参数时才有用。

servlet@templateDir

从中装入 HTML 模板文件的目录名称。设置这个特性可阻止用户查看不在模板目录或其子目录中的文件。如果不设置这个特性，那么小服务程序可以装入它具有访问权的 Web 服务器上的任何文件。

pool@defaultHandler

当没有找到特定处理程序时用来退出屏幕的屏幕处理程序。如果将它设置为从“屏幕处理程序”路径中装入的一个“屏幕处理程序”，那么请确保处理程序路径设置在缺省处理程序之前。

pool@deviceType

使用的终端型号定义。如果没有设置这个特性，则使用缺省值。

pool@encoding

允许指定要使用的“Java 编码”。请参阅 *CICS Transaction Gateway: Gateway Programming* 一书中的“Appendix A. Java encodings”，获得有关这方面的进一步信息。

pool@exitRetryLimit

在终端断开连接时，小服务程序将尝试退出正在运行的事务的次数。这防止小服务程序在无法退出屏幕时出现循环现象。

pool@gatewayUrl

小服务程序应该连接到的 CICS Transaction Gateway 的 URL。缺省值是使用本地“网关”，即 `local://`。

pool@handlerPath

从中装入“屏幕处理程序 bean”的路径。可以包含目录、.jar 文件和 .zip 文件，但不搜索子目录。

pool@idleTimeout

在自动断开终端之前允许终端保持空闲的最大时间（以秒计）。缺省值是 0，表示没有超时。

pool@installTimeout

在失败之前允许安装终端的最大时间（以秒计）。有效范围是 0 到 3600。缺省值是 0，表示没有超时。

pool@maxTerminals

可以并发地连接到 CICS 的最大终端数。缺省值是 5。

pool@minTerminals

当不使用时应保持连接到 CICS 的终端数。缺省值是 0。

pool@password

指定在创建时可注册终端关联的密码。

pool@readTimeout

在超时前允许会话式事务的最大时间（以秒计）。有效范围是 0 到 3600。值为 0 表示没有超时。

pool@reusingTerminals

如果设置为 `true`，则释放的终端将保持连接以用于下一个用户。缺省行为是当用户会话结束时断开终端。还必须设置 **pool@minTerminals** 以便这个特性生效。

pool@serverName

要连接的 CICS 服务器名称。如果没有设置这个特性，将使用在配置中定义的缺省 CICS 服务器。

pool@signonCapability

为扩展终端指定注册能力。有效项是 EPI_SIGNON_CAPABLE 或 EPI_SIGNON_INCAPABLE。

pool@pool@userid

指定在创建时可注册终端关联的用户标识。

screen@cursoring

如果设置为 true，则 JavaScript 设置屏幕光标位置。缺省设置是 false。

screen@NeutralColor

指定屏幕处理程序要将中性的 3270 颜色转换成的 HTML 颜色。

screen@stripBlanks

如果设置为 true，Web 浏览器将格式化字段文本。

screen@stripEmptyRows

如果设置为 true，屏幕处理程序会将不包含数据的行除去。

screen@TrimFields

如果设置为 true，屏幕处理程序会除去任何输入字段的起始和结尾处多余的空白。

页面映射特性

页面映射将 Web 页面与特定的屏幕或终端状态相关联。当配置 Web 服务器时，页面映射可在特性文件中指定或作为小服务程序初始化参数指定。但使用请求参数 **display**，可以查询页面映射的当前设置（请参阅第76页的『可显示特性』）。

标准页面映射特性	值
page@disconnected	当断开终端时显示的页面。
page@error	当发生错误时显示的页面
page@idle	当终端空闲时（即没有运行事务时）显示的页面
page@default	如果没有指定其它页面时显示的页面。
page@screen	为这个屏幕显示的页面。

请求参数

这些参数仅在进行请求时设置。也可以设置任何能在请求中设置的其它特性，例如：**servlet@extendedLogging**。

参数名称	参数值
request	设置为 "send" 以发送屏幕到 CICS，或设置为 "disconnect" 以结束会话 如果需要，将启动新的终端会话。
display	可以显示的特性名称，请参阅第76页的『可显示特性』。

特性和参数参考

参数名称	参数值
translateText	设置为 "true" 以将字符 <、> 和 & 分别转换成 <、> 和 &。这用于显示文本时。
事务	要启动事务的事务标识。这仅在 request 是 "send" 并且终端状态是空闲时使用。 不能为“终端小服务程序”启动 ATI 事务。
transactionData	传递到事务的附加参数。这仅在 request 是 "send" 并且终端状态是空闲的，而且指定了事务时才使用。
DFH_CURSOR	屏幕光标应该放置的字段，可以是字段索引（一个数字）或是字段名称（如果当前屏幕有可用的屏幕处理程序 bean）。 这个参数名称必须是大写。
DFH_ENTER, DFH_CLEAR, DFH_PA1 - 3, DFH_PFI - 24	忽略值。 使用这些参数中的一个来指定发送到 CICS 的 AID 键。它必须是大写。
A field index	在字段中设置的文本。 当知道字段索引时用于设置字段内容。
usingSession	如果不需要终端会话，则将它设置为 false。 如果设置了这个参数，那么任何用于该请求的终端在使用后都被废弃。
matchOnIdle	缺省情况下设置为 false。 通常当终端状态是 'client' 时（即会话式或拟会话式事务正在期待响应），小服务程序仅尝试查找当前屏幕的 ScreenHandler。将这个特性设置为 true 会使得小服务程序在每次屏幕更改时检查当前的 ScreenHandler，而不论状态是什么。如果您具有使用 BMS 映射的事务，但它不是会话式或拟会话式的，那么可能希望使用这个特性。如果有大量的 ScreenHandler，那么当这个特性设置为 true 时小服务程序可能会运行得更慢。出于这个原因，缺省值是 false。

另外，如果“屏幕处理程序 bean”可用于当前屏幕，那么可以通过在请求中指定特性来在 bean 中设置任何特性。对于 BMSMapConvert 生成的 bean，在 BMS 映射中定义的字段名就是可为字段中的必需文本设置的特性。

可显示特性

下表显示了可为 "display" 请求参数指定的值，或用于变量替换的值，请参阅第71页的『使用变量替换』。

显示值	显示
无	什么都不显示。用于抑制来自服务器引用的输出。
screen	屏幕 - 作为 HTML 表单的一部分。
errorMessage	如果发生错误，则显示错误消息。

显示值	显示
errorCause	如果发生错误，则显示有关错误的进一步信息。
DFH_CURSOR	如果可以确定，则是光标位于的字段名称，否则是字段索引。
字段索引	字段文本
屏幕处理程序特性名	如果可以确定，则是特性值。
以上列出的小服务程序配置特性中的任何特性	当前小服务程序实例的特性值。
任何页面映射特性	当前小服务程序实例的特性值。
除 "request" 和 "display" 之外的任何请求参数	当前请求的特性值。
pool@connectedTerminals	对于这个小服务程序实例，当前连接的终端数。
pool@freeTerminals	对于这个小服务程序实例，当前连接的不在使用的终端数。
pool@terminalsInUse	对于这个小服务程序实例，当前分配给用户的终端数。

CICS® Transaction Server OS/390® 版 Web 界面

“CICS Web 界面”（包括在 CICS® Transaction Server OS/390® 版 版本 1.2 和以上版中）是一组支持从 Web 浏览器直接访问 CICS 事务处理服务的事务和程序的集合。

如果已经生成了由“CICS Web 界面”使用的 HTML 模板，那么它们也可以由“终端小服务程序”用作 HTML 模板。您需要进行一些更改以调用小服务程序而不是“CICS Web 界面”，但格式基本上是相同的。

要使用模板文件：

1. 将模板文件复制到 Web 服务器上的目录中。
2. 编辑它，以将 <FORM> 操作更改为 /servlet/TerminalServlet 并添加一个隐藏的 <FORM> 元素，名称为 "request" 且值为 "send"。
3. 为 HTML 模板引用的 BMS 映射生成一个“映射”类和“屏幕处理程序 bean”。
4. 编译生成的 Java 源文件。
5. 将编译的类文件复制到 Web 服务器上适当的目录中。
6. 设置小服务程序特性 **pool@handlerPath**，以便它将装入新的类。
7. 设置一个页面映射特性，以将新的“屏幕处理程序 bean”与 HTML 模板文件相关联。

特性和参数参考

第8章 问题确定和问题解决

虽然在调查问题时可能会找到足以解决问题的信息，但不应该将问题确定与问题解决混淆起来。以下是可能发生的一些问题类型的示例：

- 最终用户错误
- 编程错误
- 配置错误。

初步检查

在调查问题之前，检查是否存在明显的原因：

- 系统以前是否能成功运行？
- 是否对系统配置进行过更改或者添加了新的特性或程序？
- 是否正确设置了 CLASSPATH 路径？

有关设置 CLASSPATH 的信息，请参阅第19页的『配置 CICS Transaction Gateway 的编程环境』。

CLASSPATH 变量显示在 CICS Transaction Gateway 跟踪输出的顶部。有关启动“网关”时启用跟踪的信息，请参阅第59页的『用用户指定的选项启动 Gateway』。

- 是否正确配置了“CICS 通用客户机”环境？

使用命令：

```
cicscli -s=servername
```

其中 *servername* 是 CICS 服务器名，后面跟命令：

```
cicscli -l
```

查看客户机是否能与服务器连接。

- 是否有任何说明故障的消息？
- 是否可以重现故障？

下一步操作

如果您认为问题出在 CICS Transaction Gateway 中，那么需要收集尽可能多的信息并与支持组织取得联系。

如果启动“网关”时没有启用跟踪，则执行以下操作：停止“网关”，重新使用一个跟踪选项启动它，然后重新创建问题。

如果您认为问题出在 CICS 通用客户机 或另一个产品中，那么应该遵照随产品提供的问题确定过程操作。有关“CICS 通用客户机”的问题确定，请参阅 *CICS Transaction Gateway: 客户机管理* 一书。对于“CICS 服务器”，请参阅适当服务器的 *Problem Determination* 一书。

消息

CICS Transaction Gateway 消息带有 CCL 前缀；它一直用于“CICS 客户机”。

下一步操作

有关 CICS Transaction Gateway 生成的消息列表，请参阅 *CICS Transaction Gateway: Gateway Messages* 一书。

跟踪

有三种主要的“网关”和应用程序跟踪级别：

堆栈跟踪

跟踪项只在出现 Java 异常时写入。它们可以帮助确定异常的源。当维持性能非常重要时使用该选项。请参阅第81页的图9。

标准跟踪

跟踪 Java 异常和主要“网关”功能及事件。缺省情况下，“网关”只在跟踪中显示所有数据块（例如 COMMAREA 或网络流）的前 128 个字节。

调试跟踪

比起堆栈或标准跟踪来说，跟踪的 Java 异常和主要“网关”功能及事件更为详细。缺省情况下，“网关”在跟踪中显示所有数据块的全部。只在性能不是很重要或者标准跟踪不能提供足够信息来解决问题时才使用它。

“网关”跟踪

可以在 CICS Transaction Gateway 进程中，或在对“网关”进行调用的用户应用程序中启动跟踪。

在 **ctgstart** 命令上使用选项来启用和控制“网关”中的跟踪。例如，要启用标准跟踪，在启动“网关”时输入以下命令：

```
ctgstart -trace
```

如果使用配置工具来配置“网关”，那么跟踪输出的缺省目的地就是 CICS Transaction Gateway 安装所在的 bin 子目录中的文件 CTG.TRC。否则，跟踪输出写入 stderr。可以通过使用 **ctgstart -tfile** 选项来覆盖跟踪输出的缺省目的地。例如，命令：

```
ctgstart -x -tfile pathname
```

在启动“网关”时启用调试跟踪，并将跟踪输出写入由 *pathname* 指定的文件中。

性能注意事项：跟踪，特别是调试跟踪，会降低性能。在 CICS Transaction Gateway 版本 4.0 中做了下列更改以减少跟踪对性能的影响：

- 如果使用 **-tfile** 选项来指定跟踪输出文件，跟踪输出不写入 stderr。
- 在跟踪输出中的所有数据块中，只有第一行是有时间戳记的。
- **-truncationsize** 和 **-dumpoffset** 选项允许您限制写入跟踪的任何数据块的大小。
- **-tfilesize** 选项允许您限制跟踪输出文件的大小。

有关启动 CICS Transaction Gateway 时带有跟踪选项的详细信息，请参阅第59页的『启动 Gateway』。


```

10:38:12:262 : main: S-C: java.io.IOException
10:38:12:263 : main: S-C: at com.ibm.gskssl.SSLSocketImpl.socketCreate(Native Method)
10:38:12:264 : main: S-C: at com.ibm.gskssl.SSLSocketImpl.create(SSLSocketImpl.java:133)
10:38:12:265 : main: S-C: at com.ibm.gskssl.SSLServerSocket.<init>(SSLServerSocket.java:109)
10:38:12:265 : main: S-C: at com.ibm.gskssl.SSLServerSocket.<init>(SSLServerSocket.java:73)
10:38:12:266 : main: S-C: at com.ibm.ctg.server.GskSslHandler.initialize(GskSslHandler.java:487)
10:38:12:266 : main: S-C: at com.ibm.ctg.server.JGate.main(JGate.java:941)

```

图 9. 样本 Java 堆栈跟踪。

应用程序跟踪

可以使用两种方法来启用应用程序中的跟踪。或者在启动 JVM 时使用 Java 伪指令，或者将调用添加到 CICS Transaction Gateway 跟踪 API 中。在 **java** 命令上使用 **-D** 选项来指定 Java 伪指令。跟踪 API 包括了一些 CICS Transaction Gateway T 类中的静态方法。有关进一步信息，请参阅 *CICS Transaction Gateway: Gateway Programming* 一书中的 “Tracing in Java client programs”（“Java 客户机程序中的跟踪”）。

JNI 跟踪

“网关”跟踪选项不启用 JNI 的跟踪。要启用它，设置环境变量 **CTG_JNI_TRACE**，使它指向希望将输出写入的文件。

小服务程序跟踪和登录到 WebSphere

IBM WebSphere Application Server V3.5 提供了高级管理控制台以简化与管理应用程序相关的任务。

要启用 WebSphere 中的小服务程序跟踪，执行以下操作：

1. 配置 WebSphere，使它将标准输出和标准错误写入您选择的文件：

在“管理控制台”中，选择希望在其中启用跟踪的应用程序服务器。在常规选项卡中有一些标签为**标准输出**和**标准错误**的文本输入框。可以在这些框中输入文件名来覆盖缺省目的地。

2. 启用应用程序中的跟踪：

- 对于“CICS Transaction Gateway 终端小服务程序”，使用 **servlet@debug** 特性来启用跟踪。

有关详细信息，请参阅第73页的『特性和参数参考』。

- 对于用户编写的应用程序，使用 Java 伪指令来启用跟踪。可以将它们输入“管理控制台”中的**命令行自变量**框。对 Java 伪指令使用与在 **java** 命令行调用中相同的格式。例如，要启用完整调试跟踪，将 **-Dgateway.T=on** 作为第一个命令行自变量输入。

有关用于跟踪的 Java 伪指令的进一步信息，请参阅 *CICS Transaction Gateway: Gateway Programming* 一书中的 “Tracing in Java client programs”（“Java 客户机程序中的跟踪”）。

3. 单击**应用**按钮来更新 WebSphere 设置，然后运行小服务程序。

在小服务程序运行时，WebSphere 将日志信息写入与标准输出相同的目的地中。这些信息包括启动时间、停止时间和所有错误消息。对于“CICS Transaction Gateway 终端小服务程序”，使用 **servlet@extendedLogging** 特性来启用扩展日志记录。如果启用了

下一步操作

扩展日志记录，“终端小服务程序”将在每一次接收到请求时写入日志。这可以为诊断问题提供有用的信息，但它也会降低性能。

可以在“终端小服务程序”运行的时候同时启用跟踪和扩展日志记录。有关“终端小服务程序”特性的详细信息，请参阅第73页的『特性和参数参考』。

程序支持

有关与 IBM 支持组织联系的完整详细信息，请参阅《CICS Transaction Gateway: 客户机管理》一书中的“程序支持”。

诊断常见问题

安装问题

安装进程使用 `uudecode` 函数，它是 `UNIX` 至 `UNIX` 复制程序的一部分。如果在尝试安装 CICS Transaction Gateway 时获得以下消息：

```
ERROR: uudecode cannot be found in your PATH,  
please refer to the Gateway Administration book  
Problem Determination section for further information
```

程序可能不在系统上。在安装 CICS Transaction Gateway 之前，确保您的系统包括该程序，并且正确配置了 `PATH`。

Java 异常

如果应用程序或 CICS Transaction Gateway 不处理异常，那么“Java 虚拟机 (JVM)”将写一个 Java 堆栈转储。转储输出的目的地取决于 JVM 实现；有关详细信息，请检查 Java 文档。

禁用 Just-In-Time (JIT) 编译器来增加写入到 Java 堆栈转储的信息。这些信息可能包括发生异常所在的 Java 源代码行。如何禁用 JIT 编译器取决于 JVM 实现；有关详细信息，请检查 Java 文档。

图10显示了一个在禁用了 JIT 编译器情况下的样本 Java 堆栈转储。

```
Exception in thread "main" java.lang.OutOfMemoryError  
  at java.lang.Thread.start(Native Method)  
  at com.ibm.ctg.server.ThreadManager.createObject(ThreadManager.java:345)  
  at com.ibm.ctg.server.ThreadManager.<init>(ThreadManager.java:131)  
  at com.ibm.ctg.server.ManagedResources.<init>(ManagedResources.java:106)  
  at com.ibm.ctg.server.JGate.main(JGate.java:895)
```

图 10. 样本 Java 堆栈转储

如果 CICS Transaction Gateway 处理异常，则只在启用跟踪时写 Java 堆栈转储。尝试在启用跟踪的情况下再现问题。这应该有助于揭示在出现异常之前发生了什么事。有关跟踪的详细信息，请参阅第80页的『跟踪』。

“网关”作为 Windows NT[®] 服务运行时的的问题

如果“网关”没有启动，或者客户机无法与之连接，那么当它作为 Windows NT[®] 服务运行时，可能是因为来自控制台的输入被启用。如图11中的那些项表明“网关”在尝试从控制台中读取：

```
11:37:58:472 : main: S-C: -> [ServerMessages: getKeys] ()
11:37:58:472 : main: S-C: <- [getKeys] = Q-
11:37:58:482 : main: S-C: java.io.IOException: The handle is invalid
11:37:58:482 : main: S-C: at java.io.FileInputStream.readBytes(Native Method)
11:37:58:482 : main: S-C: at java.io.FileInputStream.read(FileInputStream.java:190)
11:37:58:482 : main: S-C: at java.io.BufferedReader.readLine(BufferedReader.java:229)
11:37:58:482 : main: S-C: at java.io.BufferedReader.read(BufferedReader.java:286)
11:37:58:482 : main: S-C: at java.io.FilterInputStream.read(FilterInputStream.java:99)
11:37:58:482 : main: S-C: at java.io.InputStreamReader.fill(InputStreamReader.java:180)
11:37:58:482 : main: S-C: at java.io.InputStreamReader.read(InputStreamReader.java:256)
11:37:58:482 : main: S-C: at java.io.BufferedReader.fill(BufferedReader.java:145)
11:37:58:482 : main: S-C: at java.io.BufferedReader.readLine(BufferedReader.java(Compiled Code))
11:37:58:482 : main: S-C: at java.io.BufferedReader.readLine(BufferedReader.java:368)
11:37:58:482 : main: S-C: at com.ibm.ctg.server.JGate.main(JGate.java:940)
11:37:58:482 : main: S-C: CCL6559E: Unexpected exception occurred. [java.io.IOException: The handle is invalid]
```

图 11. 作为 Windows NT[®] 服务运行的“网关”的跟踪。

要禁用从控制台读取输入，可以执行以下两种操作之一：

- 确保配置工具“常规网关”设置面板上的“启用从控制台读取输入”框没有被选定。请参阅第19页的『使用配置工具』。
- 在 **ctgservice** 命令上使用 **-A** 选项以将 **-noinput** 选项传递给 **ctgstart** 命令。

java.lang.OutOfMemory 异常

如果这在 CICS Transaction Gateway 已运行了很长一段时间后发生，那么可能是由于内存泄漏所导致。如果它只在“网关”负载过重时发生，那么可能对于可用的“Java 虚拟机 (JVM)”内存来说有太多活动的线程。

内存泄漏

对“网关”进程运行内存使用监视器。如果有内存泄漏存在，那么由“网关”使用的内存数量将随时间增加。确保所有与 CICS Transaction Gateway 连接的用户应用程序在它们使用完 JavaGateway() 连接对象之后都关闭了它。如果这不能解决问题，运行“网关”调试跟踪，然后与 IBM 支持组织联系。可以使用 **ctgstart -filesize** 选项来限制跟踪输出文件的大小。也可以使用 **ctgstart -truncationsize** 选项将在跟踪中将要显示的所有数据块的大小减少到零。这将减少对性能的影响。

太多线程

可以使用如下方式估计 JVM 所需的虚拟内存的数量。将以下所有内容相加：

- “连接管理器”线程数
- Worker 线程数
- 由 Java 使用的后台线程数
- 使用中的不同网络协议数

然后将总数乘上由 JVM 分配给每个线程的 Java 堆栈大小。

在计算中，假设活动 Java 后台线程数大约为 5。有关更精确的值，请参考 Java 文档。在 CICS Transaction Gateway 的配置中设置“连接管理器”线程和 Worker 线程的最大数。有关详细信息，请参阅第22页的『常规 Gateway 设置』。

诊断常见问题

Java 分配内存所使用的方法取决于 JVM 实现。大多数 JVM 允许您通过对 Java 堆栈和 Java 堆大小设置限制来控制内存分配。

有关线程限制的详细信息，请参阅第8页的表1。有关 Java 内存分配的详细信息，请参阅 *CICS Transaction Gateway: Gateway Programming* 一书和 Java 文档。

在 OS/390® 和 UNIX® 系统上，可以限制每个进程可用的内存数量。有关详细信息，请参阅操作系统文档。

如果进程锁定

所有操作系统

要确定在哪里发生锁定，在应用程序、“网关”和“客户机”中启用跟踪。有关跟踪“网关”和应用程序的信息，请参阅第80页的『跟踪』。有关跟踪“客户机”，以及如何确定锁定是发生在“客户机”中还是服务器中的信息，请参考 *CICS Transaction Gateway: 客户机管理* 一书。如果锁定发生在“网关”中，请与 IBM 支持组织联系并提供“网关”跟踪。

AIX

在某些“Java 虚拟机 (JVM)”上，可以强制 Java 写堆栈转储，以显示当前线程的状态。例如，在 IBM Java SDK 1.3 上，可以向 Java 进程发送 **SIGQUIT (-3)** 信号，使它将堆栈转储写入 stderr。这显示了当前线程的状态。不要在正在工作的产品系统上执行该操作，而只能在已完全锁定的系统上执行。

代码页问题

“网关”跟踪显示了 JVM 运行所使用的代码页。它在跟踪顶部的 System Properties (系统特性) 部分中显示。有关如何查找“客户机”发送给服务器的代码页的信息，请参考 *CICS Transaction Gateway: 客户机管理* 一书。

SSL 异常

常规情况

在 CICS Transaction Gateway 中启用堆栈跟踪。这将告诉您在出现异常时发生了什么事。它还提供有关配置的信息，例如 CLASSPATH 变量的值。如果它没有提供给您足够的信息来诊断问题，请获得标准跟踪，然后与 IBM 支持组织联系。

启动 CICS Transaction Gateway 时

在启动“网关”时发出以下消息：

```
CCL6525W: Unable to start handler for the ssl: protocol.  
          [java.lang.ClassNotFoundException:  
           server KeyRing class name]
```

该消息显示“网关”无法装入服务器 KeyRing 类。在启动“网关”时启用堆栈跟踪。检查 CLASSPATH 变量，它在跟踪输出的顶部显示，是否有一项是包含类的目录。另外请检查 CLASSPATH 是否包括文件 cfwk.zip 的项。如果“网关”找不到该文件，它无法装入 KeyRing 文件。

在配置工具 SSL 协议设置面板的 **KeyRing 类名** 字段中指定服务器 KeyRing 类的名称。在指定 KeyRing 类名时不要包括目录或文件扩展名。CLASSPATH 应该包括包含 Keyring 类的目录的一项。

在用户应用程序中

API 调用 `SslJavaGateway.setKeyRing("ClientKeyRingName", "thePassword");` 失败，在用户应用程序中有以下错误：

```
java.lang.NoClassDefFoundError: com/ibm/sslight/SSLightKeyRing
```

检查 CLASSPATH 变量是否包括文件 `cfwk.zip` 的项。该文件和 CICS Transaction Gateway 一起提供。

JDK™ AppletViewer 问题

在使用 AppletViewer 运行来自本地文件系统的任何 CICS Transaction Gateway 样本小应用程序时，会显示消息

```
CCL6664E Unable to load relevant class to support the protocol protocol
```

。

要解决这个问题，从 AppletViewer 菜单中选择 **小应用程序**；然后选择 **特性**；然后将 **网络访问** 和 **类访问** 设置成无限制。

附录A. “CICS 通用客户机” 数据转换

ECI 和 EPI 允许运行在客户机系统中的非 CICS 应用程序访问由 CICS 服务器系统管理的 CICS 设施和数据。

字符数据由于在客户机和服务器之间传递，因此可能需要转换；例如，在“CICS 通用客户机”系统上数据以 ASCII 方式编码，在 CICS/390 服务器系统上以 EBCDIC 方式编码。数据转换是由服务器系统执行的。

在 SC09-2190, Character Data Representation Architecture Reference and Registry (CRDA) 一书中详细描述了可能的 ASCII 和 EBCDIC 编码方案。总的来说，每种编码方案都是由“编码字符集标识 (CCSID)”和“代码页全局标识 (CPGID)”标识的，CCSID 定义了一组图形字符集，CPGID 指定用于显示图形字符的代码点。

服务器系统管理的数据可以被几个使用不同的 ASCII 编码方案的客户机系统访问。要支持这种访问，每个客户机系统必须能提供 CCSID “标记”以正确转换数据。

受支持的转换

用来执行数据转换的方法是由服务器平台决定的。受支持的数据转换的范围也取决于平台。下表是从 *Communicating from CICS on System/390 SC33-1697* 中抽取的，ASCII 和 EBCDIC CCSID 被指定成地理组或语言组。

如果两个 CCSID 属于同一组，则支持 ASCII 和 EBCDIC 之间的数据转换。目的是其它 CICS 服务器将提供等价的支持。

阿拉伯语:

	CCSID	CPGID	
ASCII	00864	00864	PC data: Arabic
	01089	01089	ISO 8859-6: Arabic
	01256	01256	MS Windows: Arabic
EBCDIC	00420	00420	Host: Arabic

波罗的海沿岸国家语:

	CCSID	CPGID	
ASCII	00921	00921	PC data: Latvia, Lithuania
	01257	01257	MS Windows: Baltic Rim
EBCDIC	01112	01112	Host: Latvia, Lithuania

西里尔语:

	CCSID	CPGID	
ASCII	00866	00866	PC data: Cyrillic
	00915	00915	ISO 8859-5: Cyrillic
	01251	01251	MS Windows: Cyrillic
EBCDIC	01025	01025	Host: Cyrillic multilingual

爱沙尼亚语:

	CCSID	CPGID	
ASCII	00922	00922	PC data: Estonia
	01257	01257	MS Windows: Baltic Rim
EBCDIC	01122	01122	Host: Estonia

希腊语:

数据转换

	CCSID	CPGID	
ASCII	00869	00869	PC data: Greece
	00813	00813	ISO 8859-7: Grek
	01253	01253	MS Windows: Greek
EBCDIC	00875	00875	Host: Grek

希伯来语:

	CCSID	CPGID	
ASCII	00856	00856	PC data: Hebrew
	00916	00916	ISO 8859-8: Hebrew
	01255	01255	MS Windows: Hebrew
EBCDIC	00424	00424	Host: Hebrew

拉丁语 1:

	CCSID	CPGID	
ASCII	00437	00437	PC data: USA, many other countries
	00819	00819	ISO 8859-1: Latin-1 countries
	00850	00850	PC data: Latin-1 countries
	01252	01252	MS Windows: Latin-1 countries
EBCDIC	00037	00037	Host: USA, Canada, etc
	00273	00273	Host: Austria, Germany
	00277	00277	Host: Denmark, Norway
	00278	00278	Host: Finland, Sweden
	00280	00280	Host: Italy
	00284	00284	Host: Spain, Latin America (Spanish)
	00285	00285	Host: United Kingdom
	00297	00297	Host: France
	00500	00500	Host: International Latin-1
	00871	00871	Host: Iceland
01047	01047	Host: Latin-1 Open Systems	

包含欧元支持的拉丁语 1:

	CCSID	CPGID	
ASCII	00858	00858	PC data: Latin-1 countries
	05348	01252	MS Windows: Latin-1 countries, version 2
EBCDIC	01140	01140	Host: USA, Canada, etc
	01141	01141	Host: Austria, Germany
	01142	01142	Host: Denmark, Norway
	01143	01143	Host: Finland, Sweden
	01144	01144	Host: Italy
	01145	01145	Host: Spain, Latin America (Spanish)
	01146	01146	Host: United Kingdom
	01147	01147	Host: France
	01148	01148	Host: International Latin-1
	01149	01149	Host: Iceland

拉丁语 2:

	CCSID	CPGID	
ASCII	00852	00852	PC data: Latin-2 multilingual
	00912	00912	ISO 8859-2: Latin-2 multilingual
	01250	01250	MS windows: Latin-2
EBCDIC	00870	00870	Host: Latin-2 multilingual

拉丁语 5:

	CCSID	CPGID	
ASCII	00857	00857	PC data: Latin-5 (Turkey)
	00920	00920	ISO 8859-9: Latin-5 (Turkey)
	01254	01254	MS Windows: Turkey
EBCDIC	01026	01026	Host: Latin-5 (Turkey)

拉丁语 9:

	CCSID	CPGID	
ASCII	00923	00923	ISO 8859-15: Latin-9
EBCDIC	00924	00924	Host: Latin-9

日语:

	CCSID	CPGID	
ASCII	00932	00897	PC data: SBCS
		00301	PC data: DBCS
	00942	01041	PC data: extended SBCS
		00301	PC data: DBCS
	00943	00897	PC data: SBCS
		00941	PC data: DBCS for Open environment
	00954	00895	G0: JIS X201 Roman
		00952	G1: JIS X208-1990
		00896	G2: JIS X201 Katakana
		00953	G3: JIS X212
EBCDIC	00930	00290	Host: Katakana, extended SBCS
		00300	Host: DBCS
	00931	00037	Host: Latin-1, SBCS
		00300	Host: DBCS
	00939	01027	Host: Latin-1, extended SBCS
	00300	Host: DBCS	

韩国语:

	CCSID	CPGID	
ASCII	00934	00891	PC data: SBCS
		00926	PC data: DBCS
	00944	01040	PC data: extended SBCS
		00926	PC data: DBCS
	00949	01088	PC data: SBCS, IBM KS code
		00951	PC data: DBCS, IBM KS code
	00970	00367	G0: ASCII
		00971	G1: KSC X5601-1989
	01363	01126	MS Windows: Korean SBCS
		01362	MS Windows: Korean DBCS
EBCDIC	00933	00833	Host: extended SBCS
		00834	Host: DBCS

简体中文:

	CCSID	CPGID	
ASCII	00946	01042	PC data: extended SBCS
		00928	PC data: DBCS
	01381	01115	PC data: extended SBCS, IBM GB
		01380	PC data: DBCS, IBMGB
	01383	00367	G0: ASCII
		01382	G1: GB 2312-80 set
EBCDIC	01386	01114	PC data: SBCS, S-Chinese GBK, T-Chinese IBM BIG-5
		01385	PC data: DBCS, S-Chinese GBK
	00935	00836	Host: extended SBCS
		00837	Host: DBCS

繁体中文:

	CCSID	CPGID	
ASCII	00938	00904	PC data: SBCS
		00927	PC data: DBCS
	00948	01043	PC data: extended SBCS
		00927	PC data: DBCS
	00950	01114	PC data: SBCS, IBM BIG-5
		00947	PC data: DBCS
	00964	00367	G0: ASCII
		00960	G1: CNS 11643 plane 1

数据转换

EBCDIC	00937	00961	G2: CNS 11643 plane 2
		00037	Host: Latin-1 SBCS
		00835	Host: DBCS

附录B. CICS Transaction Gateway 和 “CICS 通用客户机” 书籍库

本章列出了所有 CICS Transaction Gateway、CICS 通用客户机 和相关书籍，并讨论了可以使用它们的各种形式。

本章中的标题有：

- 『CICS Transaction Gateway 书籍』
- 第92页的『“CICS 通用客户机” 书籍』
- 第92页的『CICS 系列出版物』
- 第93页的『书籍文件名』
- 第93页的『样本配置文档』
- 第93页的『其它出版物』
- 第94页的『查看联机文档』

CICS Transaction Gateway 书籍

- *CICS Transaction Gateway: Gateway 管理 Windows[®] 版, SB84-0499*
本书描述了 “CICS[®] Transaction Gateway Windows[®] 版” 的管理。
- *CICS Transaction Gateway: Gateway 管理 AIX[®] 版, SB84-0500*
本书描述了 “CICS[®] Transaction Gateway AIX[®] 版” 的管理。
- *CICS Transaction Gateway: Gateway 管理 Solaris 版, SC34-5934*
本书描述了 “CICS[®] Transaction Gateway Solaris 版” 的管理。
- *CICS Transaction Gateway: Gateway 管理 Linux 版, SB84-0716*
本书描述了 “CICS[®] Transaction Gateway Linux 版” 的管理。
- *CICS Transaction Gateway: Gateway 管理 HP-UX 版, SB84-0725*
本书描述了 “CICS[®] Transaction Gateway HP-UX 版” 的管理。
- *CICS Transaction Gateway: Gateway 管理 OS/390[®] 版, SC34-5935*
本书描述了 “CICS[®] Transaction Gateway OS/390[®] 版” 的管理。
- *CICS Transaction Gateway: Gateway Messages*
这本联机书籍列出并说明了 CICS Transaction Gateway 可能生成的错误消息。
无法购买本书。
- *CICS Transaction Gateway: Gateway Programming, SC34-5938*
本书介绍了使用 CICS Transaction Gateway 进行 Java[™] 编程。
还提供包含编程参考信息的附加 HTML 页面。

“CICS 通用客户机”书籍

- *CICS Transaction Gateway: 客户机管理 Windows® 版, SB84-0495*
本书描述了“CICS 通用客户机 Windows 版”的管理。
- *CICS Transaction Gateway: 客户机管理 AIX® 版, SB84-0496*
本书描述了“CICS 通用客户机 AIX 版”的管理。
- *CICS Transaction Gateway: 客户机管理 Solaris 版, SC34-5942*
本书描述了“CICS 通用客户机 Solaris 版”的管理。
- *CICS Transaction Gateway: 客户机管理 Linux 版, SB84-0497*
本书描述了“CICS 通用客户机 Linux 版”的管理。
- *oCICS Transaction Gateway: 客户机管理 HP-UX 版, SB84-0498*
本书描述了“CICS 通用客户机 HP-UX 版”的管理。
- *CICS Transaction Gateway: Client Messages*
这本联机书籍列出并说明了“CICS 通用客户机”可能生成的错误和跟踪消息。
无法购买本书。
- *CICS Transaction Gateway: C++ Programming, SC34-5945*
本书描述了如何使用 C++ 语言编写面向对象的 ECI 和 EPI 程序。
- *CICS Transaction Gateway COM Automation Programming, SC34-5946*
本书描述了如何根据“组件对象模型”(COM) 标准编写面向对象的 ECI 和 EPI 程序。

CICS 系列出版物

- *CICS® Family: Client/Server Programming, SC34-5947*
本书描述了与 CICS 客户机 / 服务器编程相关联的编程接口--“外部调用接口”(ECI)、“外部显示接口”(EPI) 和“外部安全性接口”(ESI)。它是为想要开发客户机应用程序与 CICS 服务器系统通信的应用程序设计者和程序员而编著的。

书籍文件名

表5 显示了 CICS Transaction Gateway 和 “CICS 通用客户机” 书籍的软拷贝文件名。

表 5. CICS Transaction Gateway 和 “CICS 通用客户机” 书籍以及文件名

书籍标题	文件名
CICS Transaction Gateway: Client Messages	CCLIAB
CICS Transaction Gateway: 客户机管理 AIX® 版	CCLIAD
CICS Transaction Gateway: 客户机管理 Windows® 版	CCLIAF
CICS Transaction Gateway: 客户机管理 Solaris 版	CCLIAG
CICS Transaction Gateway: 客户机管理 Linux 版	CCLIAR
oCICS Transaction Gateway: 客户机管理 HP-UX 版	CCLIAT
CICS Transaction Gateway: Gateway 管理 OS/390® 版	CCLIAI
CICS Transaction Gateway: Gateway Messages	CCLIAJ
CICS Transaction Gateway: Gateway Programming	CCLIAK
CICS Transaction Gateway: Gateway 管理 Windows® 版	CCLIAL
CICS Transaction Gateway: Gateway 管理 AIX® 版	CCLIAN
CICS Transaction Gateway: Gateway 管理 Solaris 版	CCLIAO
CICS Transaction Gateway: Gateway 管理 Linux 版	CCLIAS
CICS Transaction Gateway: Gateway 管理 HP-UX 版	CCLIAU
CICS Transaction Gateway: C++ Programming	CCLIAP
CICS Transaction Gateway COM Automation Programming	CCLIAQ
CICS® Family: Client/Server Programming	DFHZAD
注: 本表中的文件名不包含 2 位数字后缀。	

样本配置文档

许多样本配置文档都提供 “可移植文档格式” (PDF)。

这些文档提供了帮助您的逐步指南，例如，使用各种协议配置 “CICS 通用客户机” 与 CICS 服务器进行通信。它们提供的详细指令扩展了 CICS Transaction Gateway 和 “CICS 通用客户机” 书籍库中提供的信息。

由于更多样本配置文档已经可以使用，可以从我们的 Web 站点下载它们；转至：

www.ibm.com/software/ts/cics/

并遵循书籍库链接。

其它出版物

下列 “国际技术支持组织 (ITSO) 红皮书™” 出版物包含许多客户机 / 服务器配置的示例：

- *Revealed! CICS Transaction Gateway with more CICS Clients Unmasked, SG24-5277*
这本书替代了下列书籍：

其它出版物

- *CICS Clients Unmasked, GG24-2534*

可以通过许多途径获取“ITSO 红皮书”。对于最新信息，请参阅：

www.ibm.com/redbooks/

可以从下列地址找到关于 CICS 产品的信息：

www.ibm.com/software/ts/cics/

查看联机文档

可以访问联机书籍库中与 CICS Transaction Gateway 和“CICS 通用客户机”一起提供的所有文档（仅使用英语）。要使用联机书籍库，需要 Adobe Acrobat Reader 和一个适当的 Web 浏览器（并且可能需要配置它们）。

要访问联机书籍库，运行 **ctgdoc** 脚本，则显示书籍库主页。

联机书籍库允许链接到：

- PDF 格式的 CICS Transaction Gateway 和“CICS 通用客户机”书籍。
- “超文本标记语言” (HTML) 文件中的编程参考文档（仅对于 CICS Transaction Gateway 提供）。
- 自述文件。
- PDF 格式的样本配置文档。
- CICS Web 站点。

还提供了关于使用 Acrobat Reader 的信息。

可能随时提供书籍的更新版本。请检查我们的 Web 站点：

www.ibm.com/software/ts/cics/

并遵循**书籍库**链接。

注：某些书已经被翻译成其它语言。这些书不包括在联机书籍库中，但是可以作为独立的 PDF 文件从以上 Web 站点获取。

查看 PDF 书籍

PDF 信息为下列操作提供了强大功能：

- 浏览信息。在 PDF 文档内有超文本链接，还有到其它 PDF 文档和 Web 页面的超文本链接。
- 搜索特定信息。
- 在 PostScript 打印机上打印所有或部分 PDF 文档。

在 Adobe Web 站点可以找到关于 Acrobat Reader 的更多信息：

www.adobe.com/acrobat/

附录C. 声明

本信息是为在美国提供的产品和服务而开发的。在其它国家，IBM 可能未提供本文档中讨论的产品、服务或特性。请咨询本地的 IBM 代理以获取当前您所在的区域可用的产品和服务的信息。所有对 IBM 产品、程序或服务的引用并不明示或暗示只可以使用 IBM 的产品、程序或服务。任何不侵犯 IBM 知识产权的具有相同功能的产品、程序或服务都可以代替使用。但是，对任何非 IBM 产品、程序或服务的操作评估和验证都由用户自行负责。

IBM 可能已经申请或正在申请与本文档有关的各项专利权。提供本文档并不表示允许您使用这些专利。您可以用书面方式将许可证查询寄往：

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

涉及双字节 (DBCS) 信息的许可证查询，请联系您所在国的“IBM 知识产权部门”或以书面方式将查询寄往：

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

下列篇幅不适用于英联邦国家或任何当地法律与本条款不一致的其它国家： 国际商业机器公司以“仅此状态”的基础提供本出版物，不附有任何形式的担保，包括（但不限于）非侵害性、适销性和适用于某特定用途的默示保证。有些国家禁止对某些事务的明示的或暗示的担保推卸责任，因此上述条款对您或许不适用。

本资料可能会包含技术上的不精确性或印刷错误。此处提到的信息会定期更改；这些更改被合并至本资料的新版本中。IBM 可能会在不作声明的情况下，随时对本文中所说明的产品和（或）程序作改进和（或）更改。

本信息中引用的任何非 IBM Web 站点仅为方便起见，在任何情况下都不能作为对那些 Web 站点的认同。那些 Web 站点中的资料不是 IBM 产品的一部分，使用那些 Web 站点请风险自负。

为了以下目的：(i) 允许在独立创建的程序和其它程序（包括本程序）之间进行信息交换 (ii) 允许对已经交换的信息进行相互使用，而希望获取本程序有关信息的合法用户请与下列地址联系：IBM United Kingdom Laboratories, MP151, Hursley Park, Winchester, Hampshire, England, SO21 2JN。只要遵守适当的条件和条款，包括某些情形下的一定数量的付款，都可获取这方面的信息。

本信息中描述的许可程序及所有可用于它的许可资料都是 IBM 按我们之间的“IBM 客户协议”、“IBM 国际编程许可证协议”及其它同等的协议的各项条款提供。

涉及非 IBM 产品的信息是从这些产品的供应商、他们出版的通告或其它公开可用的途径获得的。IBM 没有测试过这些产品，不能确认其性能的准确度、兼容性或任何与非 IBM 产品相关的其它声明。有关非 IBM 产品功能的问题应该与这些产品的供应商联系。

商标

以下术语是“国际商业机器公司”在美国和/或其它国家的商标:

AIX	Anynet
CICS	eNetwork
IBM	MQSeries
MVS	MVS/ESA
OS/390	OS/400
RETAIN	TXSeries
VisualAge	VSE/ESA
VTAM	WebSphere

Lotus Domino、Lotus Notes 和 Domino Go Webserver 是莲花软件有限公司的商标。

Microsoft、Windows、Windows NT 和 Windows 徽标是微软公司在美国和/或其它国家的商标。

Java 和所有基于 Java 的商标和徽标是 Sun 公司在美国和/或其它国家的商标或注册商标。

UNIX 是由 X/Open 有限公司独家特许在美国和其它国家使用的注册商标。

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

[A]

- 安全套接字层 (SSL) 10
- 安全性 28
 - 安全套接字层 (SSL) 10
 - 安全性出口 11
 - 服务器认证 48
 - 概念 45
 - 公开密钥密码术 46
 - 加密 10, 46
 - 将 iKeyman 分发到客户机工作站 50
 - 可信根密钥 47
 - 客户机认证 48
 - 密钥 46
 - 密钥/证书资源库 56
 - 签署证书 47
 - 嵌入证书 54
 - 权限 10
 - 认证 48
 - 认证中心 (CA) 46
 - 数字签名 46
 - 数字证书 10, 46
 - 外部安全性接口 (ESI) 9
 - 外部签名证书 11
 - 维护数字证书 49
 - 在外部签署的证书 50
 - 证书 11
 - 专有名称 46
 - 自签名证书 11
 - 自签 CA 证书 54
 - ctgikey 49
 - HTTPS 10, 57
 - KeyRing 11, 47
 - KeyRing 文件 47
 - SSL 57
 - SSL (安全套接字层) 10
 - SSL 握手 48
- 安全性出口 11
- 安装
 - 问题 82
 - CICS® Transaction Gateway HP-UX 版 17

[B]

- 本地 CICS Transaction Gateway 11, 44
- 本地 LU 名称或模板 配置设置 29
- 编译器 16
- 变量, 环境 36

[C]

- 操作
 - 启动 Gateway 59
 - 停止 Gateway 61
- 操作系统 13
- 查看联机文档 94
- 超文本标记语言 (HTML) 94
- 程序支持 82
- 出版物, CICS Transaction Gateway 和 “CICS 通用客户机” 书籍库 91
- 初始化文件
 - CICSCOL.INI 40
 - CICSKEY.INI 36
- 初始事务 配置设置 28
- 初始 “工作程序” 线程数 配置设置 22
- 初始 “连接管理器” 线程数 配置设置 22
- 处理程序苏醒超时 配置设置 24

[D]

- 打印命令 配置设置 27
- 打印文件 配置设置 27
- 代码页标识重设项 配置设置 27
- 代码页问题 84
- 定制
 - 键盘 36
 - 屏幕颜色 40
- 动态生成名称 (TCP62) 30
- 端口 配置设置 23, 29
- 断开正在工作的连接 配置设置 24
- 对称密钥 46
- 对象请求代理 (ORB) 11

[F]

- 发送 TCP/IP Keepalive 包 配置设置 29
- 方式名 配置设置 30
- 防火墙 4
- 非对称密钥 46
- 服务器
 - 应用程序 15
 - CICS 15
 - web 15
- 服务器重试间隔 配置设置 27
- 服务器名称 配置设置 28
- 服务器 KeyRing 类文件 47
- 负载均衡 11

[G]

- 跟踪 80
 - 性能注意事项 80

- 跟踪 配置设置 31
- 跟踪, JNI 81
- 公共对象请求代理 (CORBA) 标准 11
- 工具
 - 其它 16
 - 应用程序开发 16
- 公开密钥密码术 46
- 挂起 84
- 规划 13
- 国家语言支持 17

[H]

- 环境变量 36
 - CICSCOL 40
 - CICSKEY 36
 - CLASSPATH 19
 - SHLIB_PATH 44
- 伙伴 LU 名称 配置设置 29

[J]

- 加密 10, 46
- 键盘 12
- 键盘映射文件 36
- 禁用性 12
- 进程锁定 84

[K]

- 开发工具 16
- 可访问性 12
- 可信根密钥 47
- 可移植文档格式 (PDF) 94
- 客户机跟踪文件 配置设置 32
- 客户机 KeyRing 类文件 47
- 空闲超时 配置设置 24
- 快捷键 12

[L]

- 联机书籍, PDF 94
- 联机文档, HTML 94
- 连接超时 配置设置 24, 29
- 浏览器 14
- 路线图 xiii

[M]

- 描述 配置设置 28

[N]

- 内存泄漏 83
- 内存需求 26

[O]

- 欧元 12
- 欧元支持 27

[P]

- 配置 19
 - 编程环境 19
 - 配置工具 19
 - 配置文件 19
 - CLASSPATH 19
 - Gateway.properties 文件 19
 - Web 服务器 19
- 配置工具 19
- 配置设置
 - 本地 LU 名称或模板 29
 - 初始事务 28
 - 初始“工作程序”线程数 22
 - 初始“连接管理器”线程数 22
 - 处理程序苏醒超时 24
 - 打印命令 27
 - 打印文件 27
 - 代码页标识重设项 27
 - 端口 23, 29
 - 断开正在工作的连接 24
 - 发送 TCP/IP Keepalive 包 29
 - 方式名 30
 - 服务器重试间隔 27
 - 服务器名称 28
 - 跟踪 31
 - 伙伴 LU 名称 29
 - 客户机跟踪文件 32
 - 空闲超时 24
 - 连接超时 24, 29
 - 描述 28
 - 启用从控制台读取输入 23
 - 启用协议处理程序 23
 - 全限定 CP 名称或模板 30
 - 让 Java 客户机获取类属 ECI 应答 23
 - 日志文件 28
 - 使用大写安全性 28
 - 使用客户机认证 25
 - 显示 TCP/IP 主机名 23
 - 需要客户机使用安全性类 24
 - 要完成的进行中请求的超时 23
 - 远程节点不活动轮循时间间隔 31
 - 终端出口 26
 - 终端型号定义 28
 - 主机名或 IP 地址 29
 - 最大服务器数 26
 - 最大缓冲区大小 26
 - 最大客户机环绕大小 33

配置设置 (续)

- 最大逻辑 SNA 会话数 31
- 最大请求数 26
- 最大 SNA RU 大小 31
- 最大“工作程序”线程数 23
- 最大“连接管理器”线程数 22
- “工作程序”线程可用超时 23
- Anynet 域名后缀 31
- Applid 26
- CP 名称的 IP 地址掩码 (可选的) 30
- Gateway 跟踪文件 32
- Java 网关跟踪环绕大小 32
- KeyRing 类名 25
- KeyRing 密码 25
- LU 名称模板的 IP 地址掩码 (可选的) 30
- Ping 时间频率 24
- SNA 调步大小 31
- SO_LINGER 设置 24
- 配置文件 19
 - 重命名 36
- 配置转换工具 33

[Q]

- 启动 CICS Transaction Gateway 59
- 启用从控制台读取输入 配置设置 23
- 启用协议处理程序 配置设置 23
- 签署证书 47
- 全限定 CP 名称或模板 配置设置 30

[R]

- 让 Java 客户机获取类属 ECI 应答 配置设置 23
- 认证中心 (CA) 46
- 日志文件 配置设置 28
- 软拷贝书籍, PDF 94

[S]

- 使用大写安全性 配置设置 28
- 使用客户机认证 配置设置 25
- 释放内存 26
- 书籍 91
 - 联机 94
 - 已打印的 94
 - CICS Transaction Gateway 和“CICS 通用客户机”书籍库 91
 - PDF 94
- 数字签名 46
- 数字证书 10, 46
- 锁定 84

[T]

- 停止 Gateway 61

[W]

- 外部安全性接口 (ESI) 9
- 网络计算机 5
- 文档 91
 - HTML 94
 - PDF 94
- 问题确定 79
 - 程序支持 82
 - 初步检查 79
 - 跟踪 80
 - 消息 79
 - 小服务程序跟踪和登录到 WebSphere 81
 - 应用程序跟踪 81
 - 诊断常见问题 82
 - “网关”跟踪 80
 - “CICS 通用客户机”问题 79
 - JNI 跟踪 81
- 问题, 诊断 82

[X]

- 显示 TCP/IP 主机名 配置设置 23
- 线程限制 8
- 消息 79
- 小服务程序 3
- 小服务程序跟踪和登录到 WebSphere 81
- 协议
 - 安全套接字层 (SSL) 10
 - APPC 2
 - HTML 5
 - HTTPS 10
 - TCP/IP 2
- 卸载 CICS Transaction Gateway 18
- 需要客户机使用安全性类 配置设置 24

[Y]

- 颜色映射文件 40
- 样本文本文件 69
- 要完成的进行中请求的超时 配置设置 23
- 因特网 InterOrb 协议 (IIOP) 11
- 应用程序服务器 15
- 应用程序跟踪 81
- 应用程序开发工具 16
- 硬件需求 13
- 硬拷贝书籍 94
- 映射文件
 - 重命名 36
- 语法记号 xiii
- 远程方法调用 (RMI) 11
- 远程节点不活动轮循时间间隔 配置设置 31

[Z]

- 在外部签署的证书 50
- 诊断常见问题 82
 - 代码页问题 84
 - 如果进程锁定 84
 - “网关”作为 Windows NT® 服务运行时 83
 - Java 异常 82
 - java.lang.OutOfMemory 异常 83
 - SSL 异常 84
- 终端出口 配置设置 26
- 终端型号定义 配置设置 28
- 主机名或 IP 地址 配置设置 29
- 专有名称 46
- 自签证书 54
- 最大服务器数 配置设置 26
- 最大缓冲区大小 配置设置 26
- 最大客户机环绕大小 配置设置 33
- 最大逻辑 SNA 会话数 配置设置 31
- 最大请求数 配置设置 26
- 最大 SNA RU 大小 配置设置 31
- 最大“工作程序”线程数 配置设置 23
- 最大“连接管理器”线程数 配置设置 22

[特别字符]

- “工作程序”线程可用超时 配置设置 23
- “网关”跟踪 80

A

- Anynet 域名后缀 配置设置 31
- AppletViewer 85
- Applid 配置设置 26

B

- bean, Java 4
- BMS 映射转换实用程序 68, 76

C

- CICS 服务器 15
- CICS Transaction Gateway
 - 本地 11, 44
 - 负载均衡 11
 - 配置 19
 - 启动选项 59
 - 停止 61
 - 问题确定 79
 - 硬件需求 13
 - 用用户指定的选项启动 Gateway 59
 - 用预置选项启动 CICS Transaction Gateway 59

CICS Transaction Gateway (续)

- Host on-Demand 11
- CICSCLI.LOG 28
- CICSCOL 环境变量 40
- CICSKEY 环境变量 36
- CLASSPATH 19
- CLASSPATH 设置 19, 50
- CORBA (公共对象请求代理) 标准 11
- CP 名称的 IP 地址掩码 (可选的) 配置设置 30
- ctgconv, 转换工具 33
- ctgikey 49
- ctgstart 命令 59

D

- DISPLAY 环境变量 18

E

- ESI (外部安全性接口) 9

G

- Gateway 跟踪文件 配置设置 32
- Gateway.properties 文件 19

H

- Host on-Demand 11
- HTML (超文本标记语言) 94
- HTML 文档, 查看 94
- HTTPS 10

I

- IIOP (因特网 InterOrb 协议) 11

J

- Java
 - 防火墙 4
 - 类 2
 - 小服务程序 3
 - 小应用程序 3
 - 应用程序 4
 - bean 4
 - Java 语言 2
- Java 开发工具箱 14
- Java 网关跟踪环绕大小 32
- Java 异常 82, 83
- Java Servlet Development Kit (JSDK) 2
- java.lang.OutOfMemory 异常 83

JAVA_HOME 环境变量 49
JDK 级别 14
JNI 跟踪 32, 81
JSDK (Java Servlet Development Kit) 2

X

X-Windows 18

K

KeyRing 11, 47
KeyRing 类名 配置设置 25
KeyRing 密码 配置设置 25
KeyRing 文件 47

L

LU 名称模板的 IP 地址掩码 (可选的) 配置设置 30

O

ORB (对象请求代理) 11

P

PDF (可移植文档格式) 94
PDF 书籍, 查看 94
Ping 时间频率 配置设置 24
PostScript 书籍 94

R

RMI (远程方法调用) 11

S

Samples.txt 69, 70, 71
SHLIB_PATH 44
SNA 调步大小 配置设置 31
SO_LINGER 设置 配置设置 24
SSL (安全套接字层) 10
SSL 握手 48
SSL 异常 84

T

TCP62 15
 端口号 31
TCP/IP (传输控制协议/网际协议) 15
Telnet 14

U

uudecode 17, 82

W

Web 服务器 5, 15
Web 浏览器 5, 14



程序编号: 5724-A75

中国印刷

SB84-0725-00



Spine information:



CICS Transaction Gateway Gateway 管理 HP-UX 版

版本 4.0