

# **CA Application Performance Management**

**for IBM CICS Transaction Gateway 指南**  
**版本 9.5**



本文档包括内嵌帮助系统和以电子形式分发的材料（以下简称“文档”），其仅供参考，CA随时可对其进行更改或撤销。

未经 CA 事先书面同意，不得擅自复制、转让、翻印、透露、修改或转录本文档的全部或部分内容。本文档属于 CA 的机密和专有信息，不得擅自透露，或除以下协议中所允许的用途，不得用于其他任何用途：(i) 您与 CA 之间关于使用与本文档相关的 CA 软件的单独协议；或者 (ii) 您与 CA 之间单独的保密协议。

尽管有上述规定，但如果您为本文档中所指的软件产品的授权用户，则您可打印或提供合理数量的本文档副本，供您及您的雇员内部用于与该软件相关的用途，前提是所有 CA 版权声明和标识必须附在每一份副本上。

打印或提供本文档副本的权利仅限于此类软件所适用的许可协议的有效期内。如果该许可因任何原因而终止，您应负责向 CA 书面证明已将本文档的所有副本和部分副本已退还给 CA 或被销毁。

在所适用的法律允许的范围内，CA 按照“现状”提供本文档，不附带任何保证，包括但不限于商品适销性、适用于特定目的或不侵权的默示保证。CA 在任何情况下对您或其他第三方由于使用本文档所造成的直接或间接的损失或损害都不负任何责任，包括但不限于利润损失、投资受损、业务中断、信誉损失或数据丢失，即使 CA 已经被提前明确告知这种损失或损害的可能性。

本文档中涉及的任何软件产品的使用均应遵照有关许可协议的规定且根据本声明中的条款不得以任何方式修改此许可协议。

本文档由 CA 制作。

仅提供“有限权利”。美国政府使用、复制或透露本系统受 FAR Sections 12.212、52.227-14 和 52.227-19(c)(1) - (2) 以及 DFARS Section 252.227-7014(b)(3) 的相关条款或其后续条款的限制。

版权所有 © 2013 CA。保留所有权利。此处涉及的所有商标、商品名称、服务标识和徽标均归其各自公司所有。

# CA Technologies 产品引用

本文档涉及以下 CA Technologies 产品和功能：

- CA Application Performance Management (CA APM)
- CA Application Performance Management ChangeDetector (CA APM ChangeDetector)
- CA Application Performance Management ErrorDetector (CA APM ErrorDetector)
- CA Application Performance Management for CA Database Performance (CA APM for CA Database Performance)
- CA Application Performance Management for CA SiteMinder® (CA APM for CA SiteMinder®)
- CA Application Performance Management for CA SiteMinder® Application Server Agents (CA APM for CA SiteMinder® ASA)
- CA Application Performance Management for IBM CICS Transaction Gateway (CA APM for IBM CICS Transaction Gateway)
- CA Application Performance Management for IBM WebSphere Application Server (CA APM for IBM WebSphere Application Server)
- CA Application Performance Management for IBM WebSphere Distributed Environments (CA APM for IBM WebSphere Distributed Environments)
- CA Application Performance Management for IBM WebSphere MQ (CA APM for IBM WebSphere MQ)
- CA Application Performance Management for IBM WebSphere Portal (CA APM for IBM WebSphere Portal)
- CA Application Performance Management for IBM WebSphere Process Server (CA APM for IBM WebSphere Process Server)
- CA Application Performance Management for IBM z/OS® (CA APM for IBM z/OS®)
- CA Application Performance Management for Microsoft SharePoint (CA APM for Microsoft SharePoint)
- CA Application Performance Management for Oracle Databases (CA APM for Oracle Databases)
- CA Application Performance Management for Oracle Service Bus (CA APM for Oracle Service Bus)
- CA Application Performance Management for Oracle WebLogic Portal (CA APM for Oracle WebLogic Portal)

- CA Application Performance Management for Oracle WebLogic Server (CA APM for Oracle WebLogic Server)
- CA Application Performance Management for SOA (CA APM for SOA)
- CA Application Performance Management for TIBCO BusinessWorks (CA APM for TIBCO BusinessWorks)
- CA Application Performance Management for TIBCO Enterprise Message Service (CA APM for TIBCO Enterprise Message Service)
- CA Application Performance Management for Web Servers (CA APM for Web Servers)
- CA Application Performance Management for webMethods Broker (CA APM for webMethods Broker)
- CA Application Performance Management for webMethods Integration Server (CA APM for webMethods Integration Server)
- CA Application Performance Management Integration for CA CMDB (CA APM Integration for CA CMDB)
- CA Application Performance Management Integration for CA NSM (CA APM Integration for CA NSM)
- CA Application Performance Management LeakHunter (CA APM LeakHunter)
- CA Application Performance Management Transaction Generator (CA APM TG)
- CA Cross-Enterprise Application Performance Management
- CA Customer Experience Manager (CA CEM)
- CA Embedded Entitlements Manager (CA EEM)
- CA eHealth® Performance Manager (CA eHealth)
- CA Insight™ Database Performance Monitor for DB2 for z/OS®
- CA Introscope®
- CA SiteMinder®
- CA Spectrum® Infrastructure Manager (CA Spectrum)
- CA SYSVIEW® Performance Management (CA SYSVIEW)

## 联系技术支持

要获取在线技术帮助以及办公地址、主要服务时间和电话号码的完整列表，请联系技术支持：<http://www.ca.com/worldwide>。

# 目录

---

<b>第 1 章： CTG 扩展概览</b>	<b>9</b>
关于 CTG 扩展 .....	9
CTG 扩展系统要求 .....	10
CTG 扩展组件 .....	10
CTG 扩展和您的 CA Introscope® 环境 .....	11
ProbeBuilder 指令部署 .....	12
Request Exit 概览 .....	15
混合分布式和 z/OS CTG .....	17
<b>第 2 章： 安装 CTG 扩展</b>	<b>19</b>
准备安装 .....	19
解压缩安装存档 .....	20
针对 PBD 安装管理模块和类型视图 .....	21
针对 Request Exit 安装管理模块和类型视图 .....	21
安装 Introscope 代理文件以监控 CTG 服务器 .....	22
使用 AutoProbe 启用 CTG 代码 .....	22
使用 Request Exit 启用 CTG 代码 .....	22
在分布式 CTG 平台上使用手工 ProbeBuilder 启用 CTG 代码 .....	22
准备启用 CTG 服务器代码 .....	23
指定适用于 Windows 的 CTG 服务器 RequestExit 检测 .....	23
输入 CTG 服务器的手工 ProbeBuilding 选项 .....	24
针对 CTG 服务器使用 ProbeBuilder 向导 .....	24
针对 CTG 服务器使用命令行 ProbeBuilder .....	25
在 WebSphere 上启用 CTG 客户端代码 .....	25
修改 z/OS CTG 启动脚本 .....	26
为分布式系统创建已检测的 CTG 启动脚本 .....	27
运行已检测代码 .....	28
恢复到原始代码 .....	28
在分布式 CTG 平台上启用 ChangeDetector .....	29
(可选) 配置 CTG 服务器 RequestExit 和全局统计信息探测 .....	31
(可选) 通过配置文件 ctg.ini 在 z/OS 上配置 CTG 服务器 RequestExit 和全局统计信息探测 .....	31
(可选) 在 Windows 平台上进行配置以分别获取 RequestExit 详细信息和全局统计信息 .....	32
配置 CTG 客户端支持 .....	33
配置独立 CTG 客户端应用程序 .....	33
配置通道和容器度量标准 .....	34
配置 WebSphere 下的客户端应用程序 .....	34
升级 CTG 扩展 .....	34

---

验证安装和配置.....	35
--------------	----

## 第 3 章： 使用 CTG 扩展 37

在 Introscope 中查看 CTG 扩展数据 .....	37
在树视图中查看度量标准数据 .....	38
前端 .....	38
后端 .....	39
分析特定度量标准的历史数据 .....	39
服务器度量标准 .....	39
后端   CTG_Global_Statistics .....	40
“后端   CTG_to_CICS_ECI_IPIC” 节点 .....	41
“后端   CTG_to_CICS_EPI” 节点 .....	42
“后端   JSSE to CTG” 节点 .....	42
“前端   Client_to_CTG_Aggregates” 节点 .....	43
“前端   Client_to_CTG_Details” 节点 .....	45
“前端   Client_to_CTG_Details” 事务视图节点 .....	46
“前端   Client_to_CTG_JSSE 会话” 节点 .....	47
使用 Introscope 调查器选项卡视图 .....	47
在显示板中查看 CTG 扩展数据 .....	48
“CTG 服务器 - 概览” 显示板 .....	49
“CTG 摘要 - 服务器 ECI 活动” 显示板 .....	50
“CTG 服务器 - 全局统计信息” 显示板 .....	51
“CTG 服务器 - ECI/IPIC 请求” 显示板 .....	51
“CTG 服务器 - EPI 请求” 显示板 .....	52
“CTG 服务器 - 连接管理器和工作线程” 显示板 .....	52
“CTG 服务器 - SSL” 显示板 .....	53
“CTG 客户端 - 概览” 显示板 .....	53
“CTG 客户端 - Java 网关及 SSL 会话” 显示板 .....	54
“CTG 客户端 - EPI 请求” 显示板 .....	54
“CTG 客户端 - EPI 终端请求” 显示板 .....	55
“Introscope ChangeDetector” 显示板 .....	55
修改 Introscope 警告/危险报警阈值 .....	56
CTG 事务跟踪器 .....	56
PP CTG 事务跟踪器对特殊字符的处理 .....	56
CTG 事务跟踪器属性的详细列表 .....	57

## 附录 A： CTG 性能度量标准 59

前端度量标准 .....	59
前端客户端到 CTG 聚合 .....	60
前端客户端到 CTG 详细信息 .....	61
“后端” 度量标准 .....	61
后端 CTG_to_CICS_ECI_IPIC 聚合 .....	63

---

后端 CTG 全局统计信息 .....	63
后端 CTG_to_CICS_EPI 聚合度量标准 .....	63
后端 CTG_to_CICS 线程 .....	64
CTG 显示板度量标准 .....	64
“CTG 客户端 - 概览” 显示板 .....	64
“CTG 客户端 - EPI” 显示板 .....	65
“CTG 客户端 - Java 网关及 SSL” 显示板 .....	65
“CTG 客户端 - 终端及终端请求” 显示板 .....	66
“CTG 服务器 - 概览” 显示板 .....	66
“CTG 服务器 - 连接管理器和工作线程” 显示板 .....	66
“CTG 服务器 - ECI 请求” 显示板 .....	67
“CTG 服务器 - EPI 请求” 显示板 .....	68
“CTG 服务器 - 全局统计信息” 显示板 .....	68
“CTG 服务器 - SSL” 显示板 .....	69
Request Exit 度量标准 .....	69
后端   CTG_to_CTG_ECI_IPIC_RequestExit 度量标准 .....	69
“CTG_Global_Statistics_RequestExit” 度量标准 .....	70



# 第 1 章： CTG 扩展概览

---

通过 IBM® CICS® Transaction Gateway 扩展，CA Introscope® 管理员可以监控 CTG 客户端接口。这些接口由 Java/J2EE 与 CTG 服务器中的应用程序使用。

此部分包含以下主题：

[关于 CTG 扩展 \(p. 9\)](#)

[CTG 扩展和您的 CA Introscope® 环境 \(p. 11\)](#)

[混合分布式和 z/OS CTG \(p. 17\)](#)

## 关于 CTG 扩展

CTG 扩展可以实时监控 CICS Transaction Gateway (CTG) 服务器产品和 CTG 客户端 Java。CTG 扩展也监控通过 CTG 调用客户信息控制系统 (CICS) 的 WebSphere CTG 客户端应用程序。此扩展既可以监控 CTG 服务器本身，又可以监控已检测到的调用 CTG 服务器的客户端。CTG 服务器是 Java 和 J2EE 前端与 CICS Transaction Server 后端之间的媒介。监控客户端包括监控 JCA 和基础 API 客户端应用程序。

**注意：** CTG 服务器也称为 CTG 后台进程。本指南将其称为 CTG 服务器，因为它作为服务器更容易理解。然而，在其他技术文档中，它被称为 CTG 后台进程。

CTG 扩展可以帮助 Introscope 用户关联和隔离在其 Java 应用程序、WebSphere 应用程序服务器和 CTG 本身中出现的瓶颈。CTG 扩展监控事务并提供详细的度量标准，让您确定瓶颈存在于 Java 或 J2EE 前端、CTG 服务器中间件还是 CICS 后端。

安装 CTG 扩展后，您可以：

- 查看使用 CTG 的应用程序的性能和活动的图表。无论应用程序是使用 CTG 客户端基类、CCF 接口还是 JCA 接口调用 CTG，都提供这些项目。
- 通过开箱即用型显示板查看 CTG 运行状况的高级概览。
- 查看性能的分层视图和历史视图。

## CTG 扩展系统要求

有关 CTG 系统要求的完整列表，请参阅 CA APM 总目录上的《Application Performance Management 兼容性指南》。产品兼容表提供了所有支持的操作环境的列表。

**注意：** CTG 扩展的日文版只能与 Introscope 9.0.5 结合使用。

## CTG 扩展组件

这些 CTG 扩展组件实现了与 Introscope 的交互：

组件	说明
<i>PPCTGServer_ManagementModule.jar</i>	CTG 扩展管理模块定义应用于度量标准的监控和报告逻辑。这些度量标准包括 Workstation 中提供的显示板，以及定义度量标准阈值的报警。
<i>PPCTGClient_ManagementModule.jar</i>	
<i>PPCTGRequestExit_ManagementModule.jar</i>	CTG 扩展将此模块用于 Request Exit 检测。
<i>PPCTGClient-full.pbd</i> <i>PPCTGClient-typical.pbd</i>	ProbeBuilder 指令文件，控制启用了 Introscope 的 CTG 组件向代理报告的度量标准。 当您使用 PPCTGClient-full.pbd 文件时，Introscope 会显示所有度量标准。如果您不希望产生显示所有度量标准的开销，请使用 PPCTGClient-typical.pbd 文件。
<i>PPCTGServer-full.pbd</i> <i>PPCTGServer-minimal.pbd</i> <i>PPCTGServer-typical.pbd</i>	ProbeBuilder 指令文件，控制启用了 Introscope 的 CTG 组件向代理报告的度量标准。 当您使用 PPCTGServer-full.pbd 文件时，Introscope 会显示所有度量标准。如果您不希望产生显示所有度量标准的开销，请使用 PPCTGServer-minimal.pbd 或 PPCTGServer-typical.pbd 文件。
<i>PPCTGTranTrace.pbd</i>	通过 PPCTGTranTrace.pbd 指令文件，能够跟踪 CTG 事务以及将前端和后端之间的事务进行关联。
<i>PPCTGAgent.jar</i>	此扩展使代理能够将 CTG 服务器度量标准报告给企业管理器。适用于 CTG 版本 6.x 和 7.x。
<i>PPCTGRequestExit.jar</i> <i>PPCTGbmonitor.jar</i>	此扩展使代理能够将 CTG 8.x 服务器度量标准报告给企业管理器。它需要启动 PPrunGlobalStats 批处理作业。适用于 CTG 版本 7.x 和 8.x。

<i>ctg.typeviewers.xml</i>	此企业管理器扩展定义在树中选择了 CTG 组件后，在调查器中可通过选项卡选择的 CTG 特定的视图。
<i>ctg.requestexit.typeviewers.xml</i>	此企业管理器扩展定义 CTG 请求扩展特定的视图。在树中选择了 CTG Request Exit 组件后，在调查器中可通过选项卡选择这些视图。
<i>ChangeDetector-config.xml</i>	此 XML 文件包含 CTG 的默认 ChangeDetector 监控配置条目。
<i>CTG_Tran_Trace_Template.profile</i>	此文件包含可以包含在 <i>IntroscopeAgent.profile</i> 中的 CTG 事务跟踪属性，以便收集 CTG 跟踪特定的度量标准。有关详细信息，请参阅“ <a href="#">CTG 事务跟踪器 (p. 56)</a> ”。
<i>PPrunGlobalStats.bat</i> <i>PPrunGlobalStats.sh</i>	此要求适合于 Windows 平台。您可以在 <i>PPrunGlobalStats.bat</i> 文件中指定 CTG_CLASSES 和 ISCOPE_AGENT 变量，并执行嵌入到 PPCTGbmonitor.jar 的全局统计信息监控。这对于 Windows 是必需的，因为 CTG 作为 Windows 中的一项服务运行。在 Windows 中运行要求使用一个单独的进程来提取全局统计信息。对于 Unix 平台，也存在一个等效的文件。 <i>PPrunGlobalStatus.sh</i> 是上面的 .bat 文件的 Unix/Linux 版本。

## CTG 扩展和您的 CA Introscope® 环境

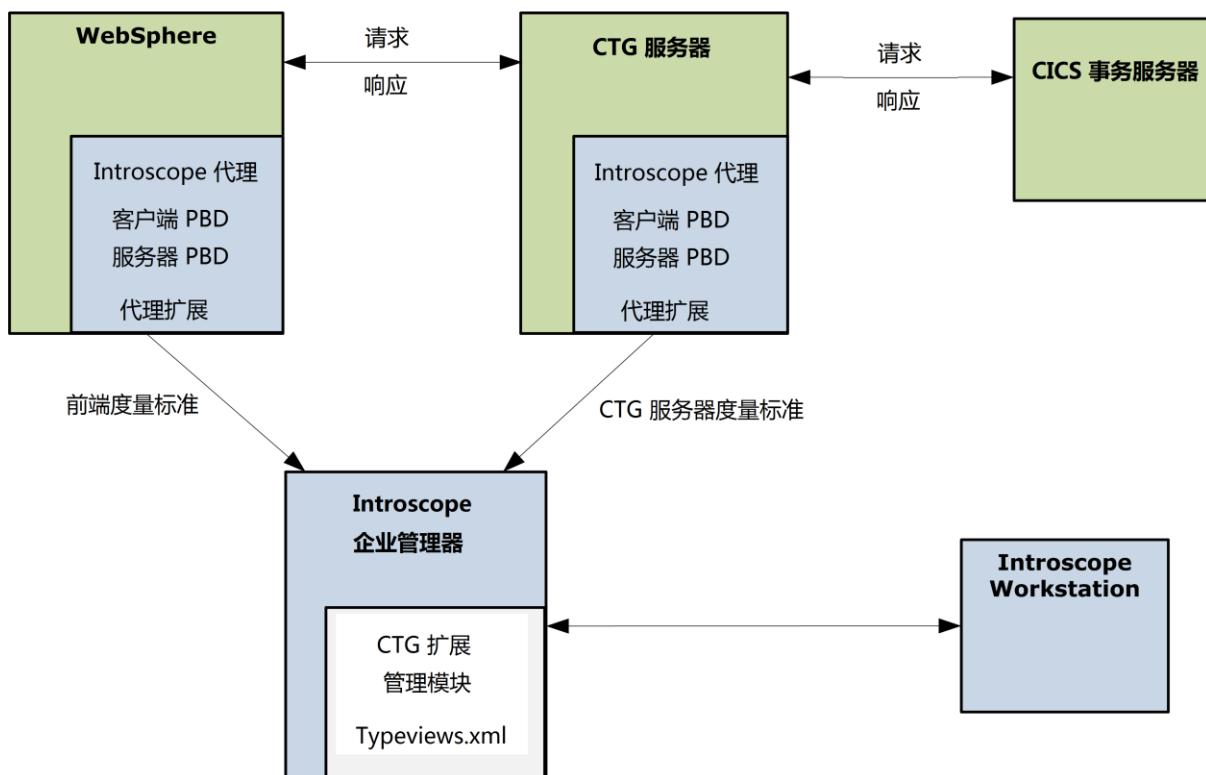
部署 CTG 扩展可能涉及一个或多个计算机，具体取决于您的环境。通过使用 PBD 或 Request Exit 完成部署。部署中涉及的 CA Introscope® 组件 ProbeBuilder 指令包括：

- 用于启用 CTG 客户端库的 ProbeBuilder 指令。这些被部署到承载应用程序的计算机和承载 CTG 服务器的计算机中。
- 用于启用 CTG 服务器库的 CTG 扩展代理扩展和 ProbeBuilder 指令。这些被部署到承载 CTG 服务器的计算机和承载应用程序的计算机中。
- 管理模块。这些被部署到企业管理器中（通常位于不同的计算机上）。

**重要信息！** CA Technologies 不建议对 CTG 启用 ProbeBuilder 和 Request Exit 二者。这样做可能会导致度量标准重叠并提高 CPU 使用率。

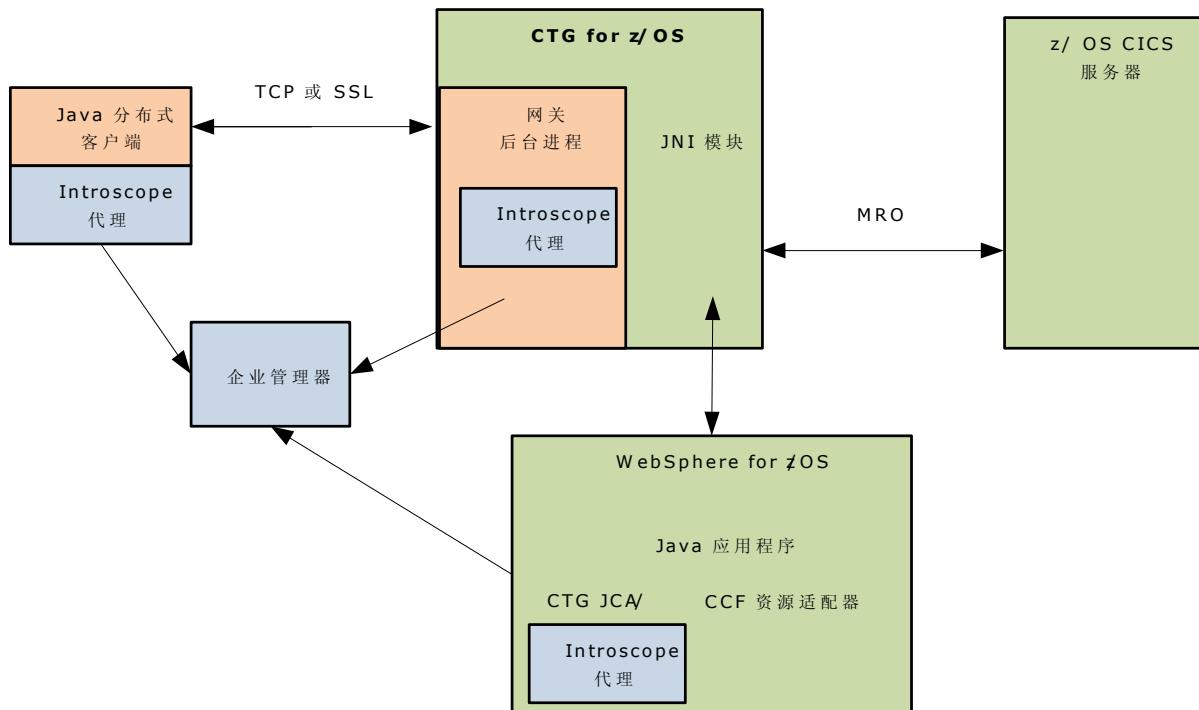
## ProbeBuilder 指令部署

下图显示了 ProbeBuilder 指令用于部署时的总体交互。CTG 扩展既可以在 z/OS 环境中运行，又可以在分布式 (Windows/Linux/UNIX) 环境中运行。



## z/OS 上的 CTG 组件

下图显示了 z/OS 上的 CTG 组件之间的交互：



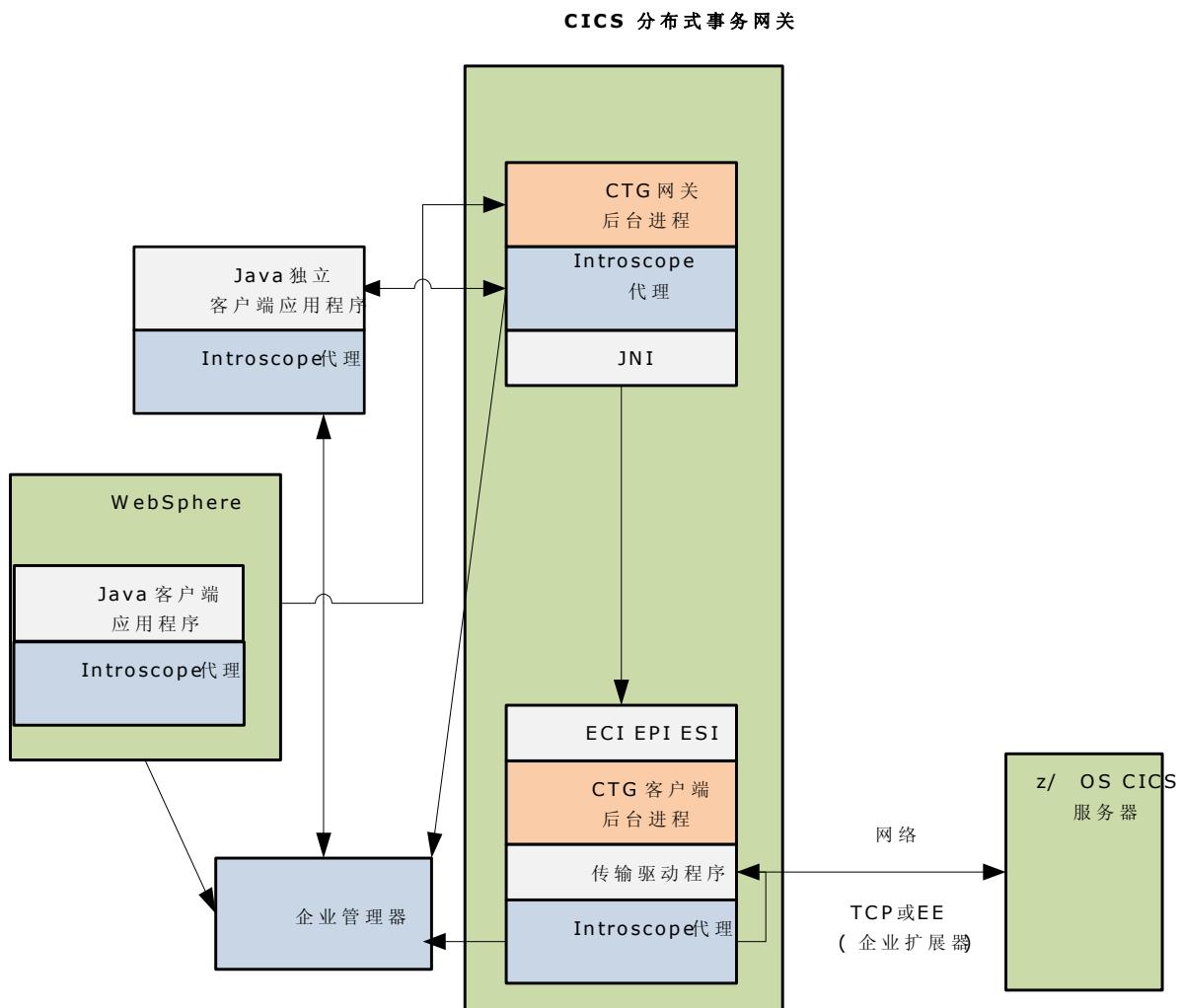
CTG for z/OS 由两个主要的运行时组件构成：

- CTG 网关后台进程，它侦听传入通信，并使用 IBM 的 EXCI 工具将该通信转发到本地 CICS 后端服务器。
- 部署在 WebSphere 运行时环境中的 JCA（或 CCF）资源适配器。

CTG 扩展监控这两个组件。在 Introscope 调查器中，所有客户端度量标准都显示在“前端|Client\_to\_CTG\_Aggregates”和“前端|Client\_to\_CTG\_Details”节点下。所有服务器度量标准都显示在“后端|CTG\_to\_CICS\_xxx”和“后端|JSSE\_to\_CTG”节点下。

## 分布式系统上的 CTG 组件

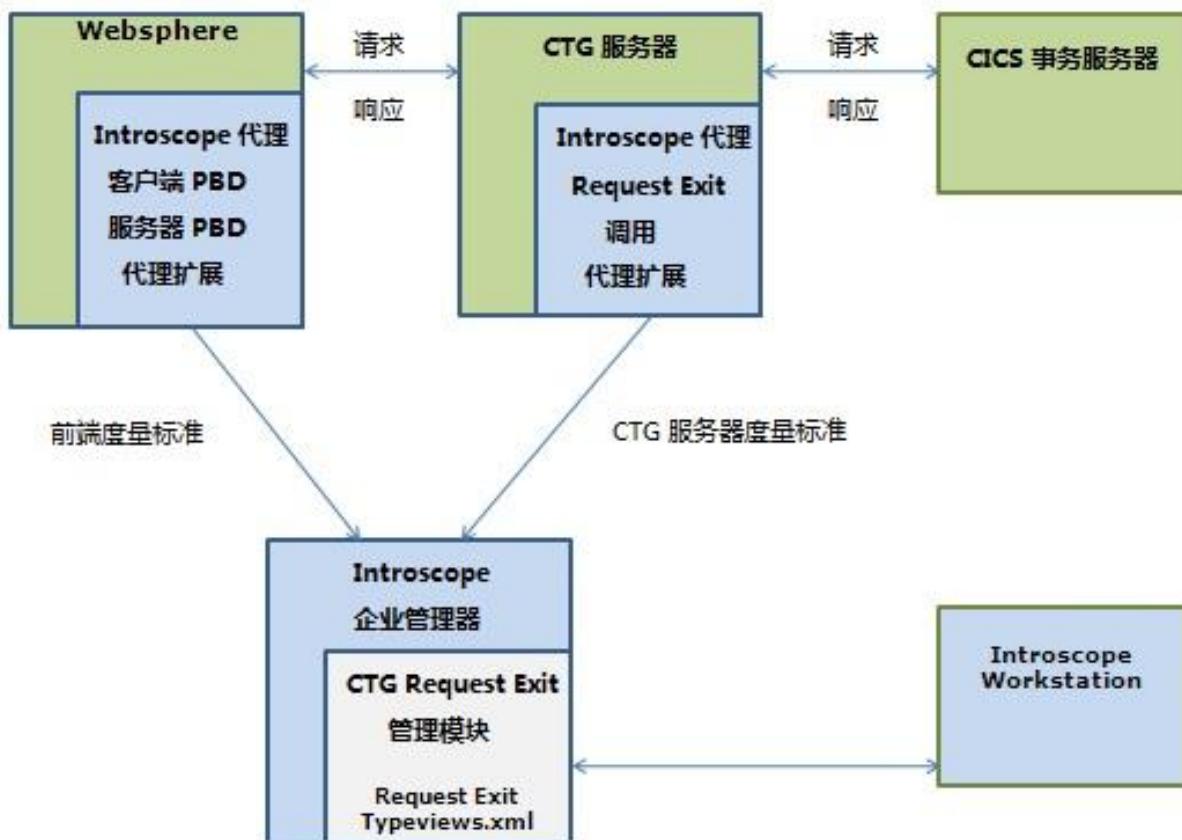
下图显示了分布式系统上的 CTG 组件之间的交互：



## Request Exit 概览

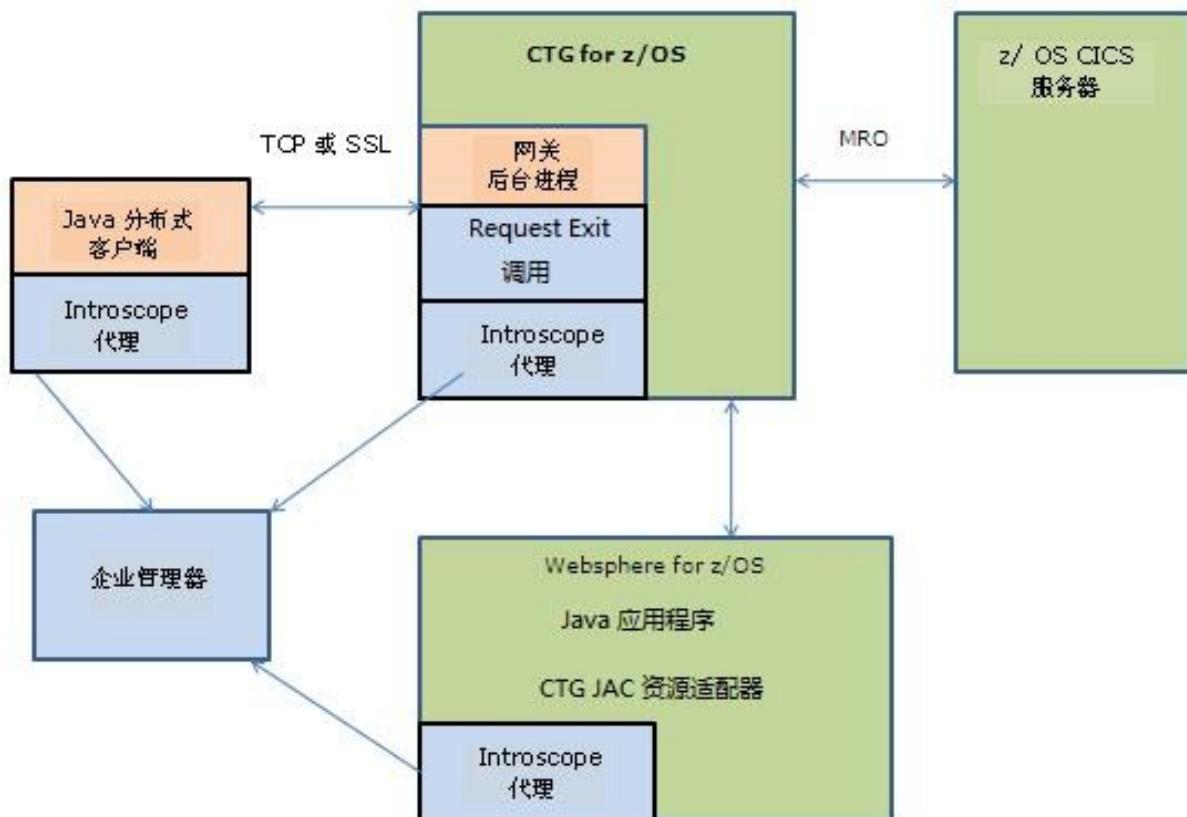
IBM 提供 Request Exit 点，可用于允许第三方插件提取由 CTG 处理的请求的统计信息。该退出仅支持 ECI 和 IPIC 事务。不支持 EPI。

Request Exit 部署使用以下组件和交互：



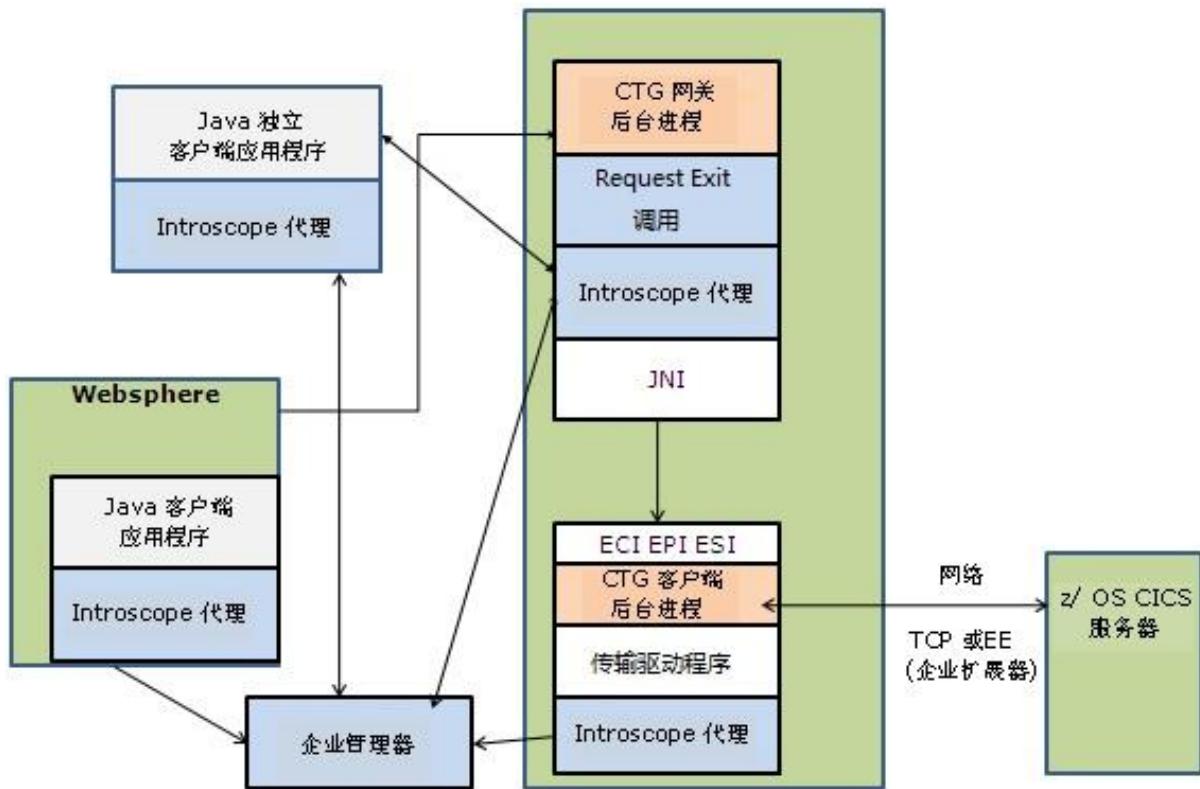
## CTG 组件的 Request Exit

下图显示了 z/OS 上的 CTG Request Exit 组件之间的交互：



## 分布式系统上的 Request Exit 组件

下图显示了分布式系统上的 CTG Request Exit 组件之间的交互：



## 混合分布式和 z/OS CTG

CTG 扩展也可以部署在混合分布式和 z/OS 环境中。例如，分布式 WebSphere 实例可以同与 z/OS 上运行的 CTG 服务器进行通信的 CTG 资源适配器一起运行。

此扩展能够监控两个环境。因此，单个工具 (Introscope) 和单个扩展套件 (CTG) 可以根据需要集成多个环境中的测量数据。

**注意：** WebSphere 不直接与 CICS 交互。所有通信通过 CTG 完成。



# 第 2 章： 安装 CTG 扩展

---

本部分介绍了如何安装和配置 CTG 扩展，以及如何在 Introscope 代理配置文件中定义 CTG 属性。

此部分包含以下主题：

[准备安装 \(p. 19\)](#)

[解压缩安装存档 \(p. 20\)](#)

[针对 PBD 安装管理模块和类型视图 \(p. 21\)](#)

[针对 Request Exit 安装管理模块和类型视图 \(p. 21\)](#)

[安装 Introscope 代理文件以监控 CTG 服务器 \(p. 22\)](#)

[使用 AutoProbe 启用 CTG 代码 \(p. 22\)](#)

[使用 Request Exit 启用 CTG 代码 \(p. 22\)](#)

[在分布式 CTG 平台上使用手工 ProbeBuilder 启用 CTG 代码 \(p. 22\)](#)

[在 WebSphere 上启用 CTG 客户端代码 \(p. 25\)](#)

[修改 z/OS CTG 启动脚本 \(p. 26\)](#)

[为分布式系统创建已检测的 CTG 启动脚本 \(p. 27\)](#)

[运行已检测代码 \(p. 28\)](#)

[恢复到原始代码 \(p. 28\)](#)

[在分布式 CTG 平台上启用 ChangeDetector \(p. 29\)](#)

[\(可选\) 配置 CTG 服务器 RequestExit 和全局统计信息探测 \(p. 31\)](#)

[配置 CTG 客户端支持 \(p. 33\)](#)

[验证安装和配置 \(p. 35\)](#)

## 准备安装

要准备进行安装，请确保已完成下列步骤：

**遵循这些步骤：**

1. 确保您的系统满足[要求 \(p. 10\)](#)。

**注意：**如果您还没有安装 Introscope，请按照《CA APM 安装和升级指南》中的说明进行操作。

2. 在您的 Introscope 环境中确定以下目录位置：

- *Introscope* 企业管理器目录(<EM\_Home>)—您的企业管理器的安装目录。
- *Introscope* 代理主目录(<代理主目录>)—监控 CTG 服务器的 Introscope 代理的安装目录。确定收集 CTG 服务器数据的每个代理的目录。

- *Introscope* 代理配置文件目录—在您计划安装 CTG 扩展的每个代理上找到 *IntroscopeAgent.profile* 文件的目录。

**注意：**标识 *com.wily.introscope.agentProfile* 的系统属性目录。代理配置文件通常位于代理安装的 *wily\core\config* 目录中。

3. 停止计划安装 CTG 扩展的企业管理器。

如果您具有群集环境：

1. 停止充当管理器的管理器的企业管理器。
2. 停止连接到启用了 CTG 扩展的代理的每个收集器企业管理器。

4. 停止计划安装 CTG 扩展的所有组件。这些可能包括：

- Java 应用程序
- WebSphere 应用程序服务器
- CTG 服务器软件

**注意：**您不需要同时停止多个组件即可安装 CTG 扩展。停止一个组件并安装，然后再转到另一个组件。例如，停止企业管理器并安装 CTG 扩展，然后停止每个 *Introscope* 代理并安装。如果您具有群集环境，请停止管理器的管理器并安装 CTG 扩展，然后停止每个收集器企业管理器并安装，再停止每个代理并安装。

## 解压缩安装存档

解压缩适用于您的系统的 CTG 扩展存档。

CTG 扩展管理模块和服务器扩展可扩展企业管理器，使 *Introscope* 能够监控 CTG 使用情况和性能。

**重要信息！** 如果您使用的是 z/OS 平台，请确保使用 FTP 二进制模式传输整个存档；ASCII 转换会损坏文件。

## 针对 PBD 安装管理模块和类型视图

以下过程适用于针对 PBD 检测安装管理模块和类型视图。

### 遵循这些步骤:

1. 停止计划安装 CTG 扩展的企业管理器。

如果您具有群集环境:

1. 停止充当管理器的管理器的企业管理器。
2. 停止每个附加的收集器企业管理器。

2. 将以下文件复制到 <企业管理器主目录>/config/modules 中:

- PPCTGClient\_ManagementModule.jar
- PPCTGServer\_ManagementModule.jar

如果要在群集 Introscope 环境中安装 CTG 扩展, 请在管理器的管理器上以及连接的所有企业管理器上安装它。

3. 将 ctg.typeviewers.xml 复制到 <企业管理器主目录>/ext/xmltv 中。

如果要在群集 Introscope 环境中安装 CTG 扩展, 请在充当管理器的管理器的企业管理器上以及连接到管理器的管理器的所有收集器企业管理器上, 安装 CTG 扩展服务器扩展。

4. 通过将解压缩的安装存档文件 *ctg.typeviewers.xml* 复制到 <*Introscope\_Home*>/ext/xmltv, 安装该扩展。

## 针对 Request Exit 安装管理模块和类型视图

以下过程适用于针对 Request Exit 检测安装管理模块和类型视图。

### 遵循这些步骤:

1. 停止计划安装 CTG 扩展的企业管理器。

如果您具有群集环境:

1. 停止充当管理器的管理器的企业管理器。
2. 停止每个附加的收集器企业管理器。

2. 将 PPCTGRequestExit\_ManagementModule.jar 复制到 <企业管理器主目录>/config/modules 中。

如果要在群集 Introscope 环境中安装 CTG 扩展, 请在管理器的管理器上以及与其连接的所有企业管理器上安装它。

3. 将 ctg.requestexit.typeviewers.xml 复制到 <企业管理器主目录>/ext/xmltv 中。

## 安装 Introscope 代理文件以监控 CTG 服务器

通过 CTG 扩展，可以监控 CTG 服务器的使用和性能。以下内容介绍了如何安装代理或 Request Exit 以监控 CTG 服务器。

要安装代理以监控 CTG 服务器，请执行以下步骤：

**遵循这些步骤：**

1. 停止计划安装扩展的 CTG 服务器。
2. 将 PPCTGAgent.jar 从 <代理主目录>/examples/PowerPackForIBMCTG/ext 复制到 <代理主目录>/wily/core/ext。

要安装 Request Exit 以监控 CTG 服务器，请执行以下步骤：

**遵循这些步骤：**

1. 停止计划安装扩展的 CTG 服务器。
2. 将 PPCTGRequestExit.jar 从 <代理主目录>/examples/PowerPackForIBMCTG/ext 复制到 CTG 的 classes 目录。
3. 确保 PPCTGRequestExit.jar 位于 CTG 的类路径列表中。
4. 在 CTG 的“配置”菜单中配置您的 CTG Request Exit 的名称。

## 使用 AutoProbe 启用 CTG 代码

对于 PBD 部署，AutoProbe 将在启动时自动创建检测探测器。

## 使用 Request Exit 启用 CTG 代码

您可以使用 Request Exit 动态探测 CTG 服务器。对于 CTG 客户端代码，您必须改为使用 JVM AutoProbe。

## 在分布式 CTG 平台上使用手工 ProbeBuilder 启用 CTG 代码

手工 ProbeBuilder 是 Introscope 启用 CTG Gateway Server 类的非动态方法。在您手工运行 ProbeBuilder 时，它可在 CTG 服务器运行前在磁盘上通过 Introscope 启用 CTG Java 类。

CTG 检测的首选方法是使用 JVM 自动探测或使用 Request Exit。但是，如果您喜欢手工探测 CTG 应用程序，请遵循以下步骤。

## 准备启用 CTG 服务器代码

本节中的说明假定您已执行以下安装和配置任务：

- 已安装 Introscope 代理。
- 已完成以下 ProbeBuilder 指令检测：
  - 已将 PPCTGClient-full.pbd 和 PPCTGServer-full.pbd 复制到 EM 文件夹下的 custompbds 中。
  - 已将 PPCTGAgent.jar 从 examples 复制到代理扩展文件夹中。
  - 已包括 PBD 文件。有关详细信息，请参阅“在 WebSphere 中启用 CTG 客户端代码”（请参阅页面）。
  - 已配置代理名称。
  - 已配置所有需要的 ProbeBuilder 选项。
  - 已将 errors.pbd 复制到 custompbds 中（如果您使用的是 ErrorDetector）。
- 已完成 Request Exit 检测：
  - 已安装针对 Request Exit 检测的 PPCTGRequestExit.jar。
  - 已将 PPCTGAgent.jar 从 examples 复制到代理扩展文件夹中。
  - 已包括 PBD 文件。
  - 已配置代理名称。
  - 已将 PPCTGRequestExit.jar 复制到 CTG 的 classes 文件夹。
  - 已确保 PPCTGRequestExit.jar 文件位于 CTG 的类路径列表中。
  - 已在 ctg.ini 文件中或在 Windows 的 CTG “配置” 菜单中配置 CTG Request Exit 的名称。

## 指定适用于 Windows 的 CTG 服务器 RequestExit 检测

CTG 的 Windows 版本仅作为 Windows 服务运行，并且需要使用 ctgservice 命令。此命令用于配置 Introscope 属性，该属性在服务启动时传递到 CTG 服务器中。此命令的典型调用方法是：

```
ctgservice -R -A-j-Dcom.wily.introscope.agentProfile=C:\<代理主目录>\wily\core\config\IntroscopeAgent.profile  
-A-j-javaagent:C:\<代理主目录>\wily\Agent.jar
```

## 输入 CTG 服务器的手工 ProbeBuilding 选项

在针对 CTG 服务器文件进行探测构建前, 请执行以下操作作为手工探测构建做准备。

### 遵循这些步骤:

1. 将当前 CTG classes/\*.jar 文件保存到备份文件夹中。
2. 针对 CTG 的 classes 目录中的以下 jar 文件运行 ProbeBuilder:

- *ctgclient.jar*
- *ctgserver.jar*
- *cicsj2ee.jar*

通过此操作可在 CTG 的 \classes 目录中创建一组 \*.isc.jar 文件, 例如 *ctgclient.isc.jar*、*ctgserver.isc.jar* 和 *cicsj2ee.isc.jar*。

3. 重命名下列 \*.isc.jar 文件:

- Windows:

```
rename ctgserver.isc.jar ctgserver.jar  
rename ctgclient.isc.jar ctgclient.jar  
rename cicsj2ee.isc.jar cicsj2ee.jar
```

- UNIX:

```
mv ctgserver.isc.jar ctgserver.jar  
mv ctgclient.isc.jar ctgclient.jar  
mv cicsj2ee.isc.jar cicsj2ee.jar
```

4. 如果在其他目录中创建了探测的 jar 文件的 .isc 版本, 请将这些文件直接复制到 CTG 的 classes 目录中。

5. 手工检测 jar 文件。检测方法有两种:

- ProbeBuilder 向导—提供一个用于运行 ProbeBuilder 的 GUI 对话框。
- 命令行 ProbeBuilder—一个针对 ProbeBuilder 的命令行界面, 适用于不存在窗口化系统的环境。

**注意:** 有关检测字节码的详细信息, 请参阅《CA APM Java 代理实施指南》。

## 针对 CTG 服务器使用 ProbeBuilder 向导

如果您的计算机具有 Windows 环境, 您可以使用基于 GUI 的 ProbeBuilder 向导。

**遵循这些步骤:**

1. 确保 ProbeBuilder 向导可访问 CTG 扩展 .pbd 文件（应安装在 <Agent\_Home> 目录中）。
2. 针对 *ctgserver.jar*、*ctgclient.jar* 和 *cicsj2ee.jar* 运行 ProbeBuilder 向导。
3. 开始之前，为确保安全，请先将原始 CTG jar 文件复制到备份区域。

## 针对 CTG 服务器使用命令行 ProbeBuilder

如果您的计算机不具有窗口化环境，您可以使用命令行 ProbeBuilder 手工探测 CTG jar 文件。

**遵循这些步骤:**

1. 在运行 ProbeBuilder 前，为确保安全，请先将原始 CTG jar 文件复制到备份区域。
2. 使用 Probe builder 命令和指令。
3. 如果您正在探测位于 CTG classes 目录中的分布式 CTG 文件，请确保手工探测下列 jar 文件：
  - *ctgserver.jar*
  - *ctgclient.jar*
  - *cicsj2ee.jar*

**注意：**有关 Probe builder 命令和指令的详细信息，请参阅《CA APM Java 代理实施指南》。

## 在 WebSphere 上启用 CTG 客户端代码

在 z/OS 上的 WebSphere 上，可以使用 JVM AutoProbe 自动监控 CTG 客户端和服务器。

遵循这些步骤:

- 编辑文件 *IntroscopeAgent.profile* 并修改属性 *introscope.autoprobe.directivesFile*, 将下列文件包含在其定义中:
  - *PPCTGClient-full.pbd*
  - *PPCTGServer-full.pbd*
  - *required.pbd*

例如:

*introscope.autoprobe.directivesFile=<现有 PBD 列表>,  
PPCTGClient-full.pbd,PPCTGServer-full.pbd,required.pbd*

注意: *introscope.autoprobe.directivesFile* 的整个定义必须包含在一行中。

## 修改 z/OS CTG 启动脚本

如果在 z/OS 环境中进行安装, 请修改 CTG 的 *ctgstart* 脚本以在启动时将启用 Introscope 的 JVM AutoProbe 参数传递给 CTG JVM。

遵循这些步骤:

1. 创建原始 *ctgstart* 脚本的备份副本。
2. 打开 CTG 的 *ctgstart* 脚本进行编辑。*ctgstart* 脚本位于 CTG 的 bin 目录中。
3. 搜索以下语句块 (位于 *ctgstart* 脚本底部附近) :

```
jvmoptions=$(echo $alloptions | $awk '{  
    for (i = 1; i<=NF; i++) {  
        if (substr($i,1,2) == "-j") jvmoptions = jvmoptions " "  
        substr($i,3)  
    }  
    print jvmoptions  
'})
```
4. 要使用 AutoProbeConnector 启用 AutoProbe, 请直接在步骤 2 中的语句后添加下列行:  
*wilyoptions="-Xbootclasspath/p:./wily/connectors/AutoProbeConnector.jar:./wily/Agent.jar -Dcom.wily.introscope.agentProfile=./wily/IntroscopeAgent.profile -Xverify:none"*  
*jvmoptions="\$jvmoptions \$wilyoptions"*
5. 要使用 javaagent 选项启用 AutoProbe, 请直接在步骤 2 中的语句后添加下列行:  
*wilyoptions="-javaagent:./wily/Agent.jar -Dcom.wily.introscope.agentProfile=./wily/IntroscopeAgent.profile"*  
*jvmoptions="\$jvmoptions \$wilyoptions"*

## 为分布式系统创建已检测的 CTG 启动脚本

在 Windows 和 UNIX 下，分布式 CTG 启动脚本是一个二进制文件。此文件必须使用启动脚本进行调用以传递所需 Introscope 参数。

### 遵循这些步骤：

1. 在使用手工 ProbeBuilder 将探测器添加至 CTG 类后，为分布式 CTG 服务器创建一个启动脚本。在 Windows 中，脚本是 .bat 文件；在 UNIX 中，脚本是 .sh shell 文件。
2. 在启动脚本中，指定 Introscope 类和代理配置文件的位置。

以下是针对 Windows 的示例启动脚本 (.bat 文件)。（假定这些 IBM 产品的已安装文件夹位于 CTG 和 WebSphere 上。）

```
set
CLASSPATH=c:\CTG<ccc>\wily\Agent.jar;c:\CTG<ccc>\classes\cicsj2ee.jar;c:\CTG<ccc>\classes\ctgserver.jar;c:\CTG<ccc>\classes\ctgclient.jar;
c:\CTG<ccc>\classes\ccf2.jar;c:\CTG<ccc>\classes\ctgsamples.jar;
C:\WebSphere<www>\AppServer\java\jre\bin;%CLASSPATH%
set JAVA_HOME=C:\IBM_JVM<jjj>\java\jre
set PATH=C:\WebSphere<www>\AppServer\java\jre\bin;.\bin;%PATH%
ctgstart
-j-Dcom.wily.introscope.agentProfile=C:\CTG<ccc>\wily\core\config\IntroscopeAgent.profile
```

其中

<ccc> = CTG 版本号  
<jjj> = Java 版本号  
<www> = WebSphere 版本号

3. 编辑应用程序启动脚本的类路径，添加包含已检测代码（通过 ProbeBuilder 创建）的目录的位置。

**注意：**确保这些条目继续类路径中的原始条目。请参阅[运行已检测代码 \(p. 28\)](#)。

4. 编辑应用程序启动脚本中的类路径，以将该路径添加至：  
<代理主目录>/Agent.jar。

例如，您可编辑以下类路径：

```
java -classpath
/<your-applicationpath>/classes:/<your-applicationpath>/lib/app.jar
MainClass
```

使其与以下类路径相似：

```
java -classpath /<您的应用程序路径>.isc/classes:/<您的应用程序路
径>.isc/lib/app.jar:<代理主目录>/Agent.jar MainClass
```

5. 完成手工检测后，请更新 CTG 服务器启动脚本的类路径，以反映已检测代码和代理的位置。
6. 使用新的启动脚本启动应用程序。

## 运行已检测代码

指向启用 Introscope 的代码而非原始代码的方式有三种：

- 在类路径中，将原始类路径替换为已检测代码的路径。本章中的说明已指导您在首次检测应用程序时执行此步骤。

**注意：**在 Windows 或 UNIX 中进行安装时，如果安装未完成，则可能需要更新您的类路径以使用 `Java\bin` 文件夹。此文件夹由 WebSphere 应用程序服务器 `Java\bin` 目录提供，与 Java 由 Java 社区提供不同。此类路径更新对于确保 JAVA-CTG 兼容性十分有必要。

- 在类路径前面追加路径。如果在类路径中只有一部分应用程序代码被检测，请将已检测代码路径置于原始路径前。

**注意：**如果您执行此操作，已检测代码就会加载和报告性能数据。未检测代码仍可正常加载和运行，但不会报告性能数据。

- 将已检测代码置于原始类路径中。

当类路径设置在多个位置或要评估系统时可以使用此方法。在生产环境中使用此方法时要谨慎。您可能会容易忘记正在使用原始代码还是已检测代码。

- 将原始代码移至新位置。保留类路径不变，接着将已检测代码移至原始位置。
- 在 UNIX 计算机上，您也可以创建指向原始路径位置中的已检测代码的符号链接。

## 恢复到原始代码

要恢复到原始的未检测代码，请按照如下所述撤消检测：

- 如果您已将已检测代码的路径放置到 Java 类路径中，请使用原始值替换已检测代码的路径。

- 如果您已在原始代码的路径前添加已检测代码的路径，请删除类路径中已在前面附加的部分，只保留原始类路径。
- 如果您已删除原始代码并已将已检测代码放置在原始类路径中，请从原始路径中删除已检测代码。将原始代码放入原始类路径中。

**注意：**如果您在 UNIX 系统上使用了符号链接，请将该符号链接指向原始目录或删除该链接，然后将代码移到原始类路径中。

## 在分布式 CTG 平台上启用 ChangeDetector

Windows、Linux 和 UNIX 等分布式（非 z/OS）系统支持 ChangeDetector。ChangeDetector 设计用于监控 CTG 配置文件（如 *ctg.ini* 和 *ctgenvar*）以及相关 JAR 文件，以检测是否对系统进行了任何更改。通过此设计可跟踪 CTG 环境中会导致中断或问题产生的所有配置更改。

CTG 扩展包括一个您必须自定义的默认 ChangeDetector 配置文件，以便正确启用 ChangeDetector 来监控 CTG 安装。此外，您还可以添加要由 ChangeDetector 监控的其他文件。

**注意：**有关添加用于监控的文件的详细信息，请参阅《CA APM ChangeDetector 用户指南》。

本节明确地重点介绍与 CTG 相关的 ChangeDetector 监控。

*ChangeDetector-config.xml* 文件包含一组用于监控 CTG 及相关 Introscope 代理文件实体的默认 ChangeDetector 配置设置。要正确使用此配置，请更新 *ChangeDetector-config.xml* 文件中的两个条目。

**注意：**如果您要在分布式 CTG 系统上使用 ChangeDetector，请注意 ChangeDetector 无法使用 Java 类监视器检测 Java 类的更改。此限制是因为 CTG 代理已被手工探测到且其无法检测 Java 类的动态更改。

（ChangeDetector 可监控和检测文件系统和配置属性文件的更改，完全符合您的预期。）

遵循这些步骤：

1. 要更新 ChangeDetector-config.xml 文件，在 CTG 目录注释块下：

```
<!-- ===== -->
<!-- change the name= property below to point to your CTG directory -->
<!-- ===== -->
<scan-directory recursive="true" name="your CTG directory"
fileset="default" enabled="true" />
```

更改指向特定 CTG 目录的 name= 参数。

例如，如果您的 CTG 安装位于目录 ctg<ccc>/usr/lpp/ctg/ctg<ccc> 下，您需要按如下方式设置 scan-directory 条目中的 name 参数：

```
<scan-directory recursive="true" name="/usr/lpp/ctg/ctg<ccc>"
fileset="default" enabled="true" />
```

2. 在 CA APM 目录注释块下：

```
<!-- ===== -->
<!-- change the name= property below to point to your CA APM directory -->
<!-- ===== -->
<scan-directory recursive="true" name="your CA APM directory"
fileset="default" enabled="true" />
```

更改指向特定 CA APM 安装目录的 name= 参数。

例如，如果您的 CA APM 安装位于以下目录下：

/usr/vendor/ca\_apm

请按如下方式更改 scan-directory 条目中的 name 参数：

```
<scan-directory recursive="true" name="/usr/vendor/ca_apm"
fileset="default" enabled="true" />
```

3. 完成上述修改后，请将已更新的 *ChangeDetector-config.xml* 文件放入 Introscope 代理目录中。

4. 确保 *ChangeDetectorAgent.jar* 文件已安装在代理的 .ext 目录中。

5. 修改代理配置文件的 *introscope.changeDetector.profile=config* 条目以指定 ChangeDetector 配置文件的路径。

6. 确保代理配置文件的 *introscope.changeDetector.agentID=config* 条目中包含您希望 ChangeDetector 代理使用的名称。

例如，如果您的 CTG 安装位于目录 ctg<ccc>/usr/lpp/ctg/ctg<ccc> 下，您需要按如下方式设置 scan-directory 条目中的 name 参数：

```
<scan-directory recursive="true" name="/usr/lpp/ctg/ctg<ccc>"
fileset="default" enabled="true" />
```

在执行上述步骤后，您可以使用 ChangeDetector 监控 CTG 服务器。

有关查看和解析 ChangeDetector 数据的详细信息，请参阅《CA APM ChangeDetector 指南》。

## (可选) 配置 CTG 服务器 RequestExit 和全局统计信息探测

要配置 CTG 服务器以获得 IBM RequestExit 度量标准, 请执行以下操作。

遵循这些步骤:

1. 将 PPCTGRequestExit.jar 放入 CTG 的 classes 目录中。

例如:

```
/u/usr/lpp/cicstg/ctg800/classes/PPCTGRequestExit.jar
```

2. 将 PPCTGRequestExit.jar 添加到 CTG 的类路径列表中。

例如:

```
export CLASSPATH=${CTG_CLASSES}/PPCTGRequestExit.jar:${CLASSPATH}
```

3. 在 IntroscopeAgent.profile 中自定义以下值

```
ppctg.statistics.host=localhost  
ppctg.statistics.port=2980  
ppctg.statistics.sleep=30  
ppctg.statistics.enable=true
```

**注意:** 在 CTG\_Global\_Statistics\_StatsExit 下的概览类型视图故意保留为空并且在此选项卡下不包含任何度量标准数据。如果您在 CTG\_Global\_Statistics\_StatsExit 的类型视图中没有看到任何数据, 则表示 CTG\_Global\_Statistics\_StatsExit 节点中不包含任何度量标准数据。

## (可选) 通过配置文件 ctg.ini 在 z/OS 上配置 CTG 服务器 RequestExit 和全局统计信息探测

要通过配置文件 ctg.ini 在 z/OS 平台上进行配置以获取 z/OS 上的 RequestExit 详细信息和全局状态探测, 请执行以下操作:

遵循这些步骤:

1. 将 APM/Wily PPCTG Request Exit 的名称添加到 ctg.ini 中。
2. 根据 CTG 安装环境添加其他参数。

例如:

```
requestexits = com.ibm.ctg.server.APM_RequestExit_Monitor  
protocol@statsapi.handler=com.ibm.ctg.server.RestrictedTCPHandler  
protocol@statsapi.parameters=port=2980;bind=;connecttimeout=2000;maxconn=5;
```

## (可选) 在 Windows 平台上进行配置以分别获取 RequestExit 详细信息和全局统计信息

要在 Windows 平台上分别获取 RequestExit 详细信息和全局统计信息，请执行以下操作：

### 遵循这些步骤：

1. 运行 ctgservice 命令以设置 Request Exit 的类路径。在将 PPCTGRequestExit.jar 从 Agent\wily\examples\ext 文件夹复制到 IBM CTG\Classes 文件夹之后。

例如：

```
ctgservice -R -A-classpath=C:\<IBM CTG 主目录>\classes\PPCTGRequestExit.jar  
-A-j-Dcom.wily.introscope.agentProfile=C:\<代理主目  
录>\wily\core\config\IntroscopeAgent.profile  
-A-j-javaagent:C:\<代理主目录>\wily\Agent.jar
```

2. 通过调用 IBM CTG 配置工具并选择网关后台进程节点，将 RequestExit\_Monitor 名称配置到 CTG 程序中。
3. 选择“监控”选项卡，通过 Tab 键移动到“使用这些请求监视器”字段，并将以下 APM 退出值填入框中：  
`com.ibm.ctg.server.APM_RequestExit_Monitor`
4. 选择“添加”以添加上述条目。

要为 RequestExit 详细状态配置参数，请执行以下操作：

### 遵循这些步骤：

1. 展开并选择节点“CICS Transaction Gateway”、“网关后台进程”、“统计信息 API”选项。
2. 选中“启用协议处理程序”复选框。
3. (可选) 要覆盖默认值，请更改 TCP 端口号、超时和其他选项。
4. 如果将字段 BinAddress 留空，则默认情况下 CTG 使用本地主机。

执行上一步骤后，您将可以收集 IBM Request Exit 的详细度量标准。

要为 RequestExit 全局统计信息配置参数，请执行以下操作：

**遵循这些步骤：**

**注意：**由于 IBM 支持的限制，无法自动启用 Request Exit 全局统计信息。

1. 要通过 PP 调用全局统计信息收集，请执行 PPrunGlobalStats.bat 文件，并自定义以下条目。
2. 指向 CTG 类和 Introscope 代理目录：

```
set CTG_CLASSES= <point to the CTG class files>
set ISCOPE_AGENT= <point to Agent extension folder>
```

## 配置 CTG 客户端支持

CTG 扩展可提供针对 CTG 服务器和 CTG 客户端应用程序的度量标准。本节介绍了如何将 CTG 客户端应用程序定义为向 CTG 服务器发出请求的应用程序。此外，CTG 服务器请求运行一个特定的 CICS ECI 或 EPI 应用程序。相应结果会返回给客户端。

CTG 扩展提供的客户端度量标准可将测量范围延伸至正在启动的客户端。通过延伸测量范围可实现对响应时间、分类等的全面管理。

CTG 扩展可通过配置为与独立 CTG 客户端应用程序或与 WebSphere 下运行的客户端应用程序进行协作。

## 配置独立 CTG 客户端应用程序

对于独立 CTG 客户端应用程序，Introscope 支持的配置方式与使用 Introscope 测量的任何其他独立 Java 应用程序一样。此外，您必须将 *PPCTGClient-full.pbd* 和 *required.pbd* 文件添加至 *IntroscopeAgent.profile* 的 *introscope.autoprobe.directivesFile= <parameter>* 中。如果您要监控本地模式 (EXCI) 交互，您还必须将 *PPCTGServer-full.pbd* 添加至上述列表中。

**注意：**确保 CTG 扩展的 *PPCTGAgent.jar* 文件位于代理的 *ext* 目录中。

## 配置通道和容器度量标准

为了最小化针对通道和容器的度量标准收集和 CPU 开销，在 PBD 文件中提供一个选项，用于切换通道和容器度量标准。也通过在 PPCTGServer-full.pbd 和 PPCTGServer-typical.pbd 文件中注释或取消注释指令，来完成相同操作。

```
# To disable IPIC Channel|Container metrics use this directive  
#           Turnoff: IPICChannelContainerTracing  
# To enable use this directive  
#           Turnon: IPICChannelContainerTracing
```

必须重新启动代理，才能使更改生效。

## 配置 WebSphere 下的客户端应用程序

对于在 WebSphere 下运行的客户端应用程序，在配置 CTG 客户端支持时存在两种方式。

遵循这些步骤：

- 将 CTG 的 *ctgclient.jar*、*cicsj2ee.jar* 和 *ccf2.jar* 等文件直接添加在 WebSphere JVM 的类路径配置菜单中。  
**注意：**从 CTG 8.0 开始，*cicsj2ee.jar* 已被重命名为 *cicsjee.jar*。
- 通过资源适配器屏幕在 WebSphere 下安装 CTG jar 文件，并安装相关 *cicseci.rar* 和/或 *cicsepi.rar* 文件。

在将 CTG 客户端支持添加至 WebSphere 后，*PPCTGClient-full.pbd* 和 *required.pbd* 文件必须添加至 *IntroscopeAgent.profile introscope.autoprobe.directivesFile=<parameter>* 中。如果您要监控本地模式 (EXCI) 交互，请将 *PPCTGServer-full.pbd* 添加至上述列表中。

**注意：**确保 CTG 扩展的 *PPCTGAgent.jar* 文件位于代理的 *ext* 目录中。有关包括的文件列表，请参阅 “[CTG 扩展组件 \(p. 10\)](#)”。

## 升级 CTG 扩展

您不能从 CTG 扩展的早期版本进行升级。相反，您必须首先卸载早期版本，然后安装当前版本。

## 验证安装和配置

确认 CTG 扩展安装已正确安装和配置非常重要。

### 遵循这些步骤：

1. 重新启动受监控的 WebSphere 应用程序服务器、Java 应用程序、CTG 服务器软件和 Introscope 企业管理器。
2. 在成功重新启动这些组件后，监控 CTG 服务器和 WebSphere 应用程序服务器的代理应开始向企业管理器报告数据。
3. 启动 Workstation 并连接到企业管理器。
4. 确认您可以在调查器中[查看来自 CTG 服务器和应用程序服务器的数据 \(p. 37\)](#)。



# 第3章： 使用 CTG 扩展

---

本节介绍了以下内容：

- 如何使用 Introscope 监控 CTG 环境
- CTG 扩展可用的各种数据概览
- 如何在 Workstation 和控制台中查看该数据

本节中的说明假定您已完成安装步骤，启用 Introscope 的 CTG 应用程序正在正常运行并会将数据报告给企业管理器。

此部分包含以下主题：

- [在 Introscope 中查看 CTG 扩展数据 \(p. 37\)](#)
- [在树视图中查看度量标准数据 \(p. 38\)](#)
- [服务器度量标准 \(p. 39\)](#)
- [使用 Introscope 调查器选项卡视图 \(p. 47\)](#)
- [在显示板中查看 CTG 扩展数据 \(p. 48\)](#)
- [修改 Introscope 警告/危险报警阈值 \(p. 56\)](#)
- [CTG 事务跟踪器 \(p. 56\)](#)

## 在 Introscope 中查看 CTG 扩展数据

在 Introscope 中查看 CTG 扩展数据的方式有几种：

- 调查器中的原始度量标准—为技术用户提供一个关于 CTG 的所有资源和组件的基本性能的详细视图。
- 调查器中的选项卡视图—为技术用户提供关于 CTG 系统的性能、资源和组件的聚合视图。
- 控制台中的显示板—为不熟悉 CTG 体系结构详情的用户提供易于使用的界面。
- 调查器中的报警—显示报警（诸如由 CTG 扩展显示板生成的报警和您创建的报警）。

## 在树视图中查看度量标准数据

要在调查器中查看 CTG 扩展度量标准，请执行以下步骤：

### 遵循这些步骤：

1. 启动托管应用程序。
2. 启动企业管理器。
3. 启动并登录 Workstation。
4. 打开调查器窗口。

所有 CTG 扩展特定的度量标准会出现在树中的几个节点下。

**注意：**可用度量标准取决于您的应用程序使用的 CTG 和 WebSphere 资源。仅显示您的托管 Java 应用程序使用的度量标准。

## 前端

提供以下前端 CTG 度量标准：

- **Apps**—各个应用程序的度量标准会显示在此节点下(按照应用程序名称显示)
- **Client\_to\_CTG\_Aggregates**—此节点具有一个“客户端聚合图形摘要”选项卡。树的此节点下是几个聚合度量标准和以下子节点：
  - **BASE\_ECI\_EPI**—在使用不支持 Request Exit 检测的旧版 CTG 且不使用 CTG 8.x 时可能会出现。
  - **JCA\_ECI**
  - **JCA\_EPI**—在使用不支持 Request Exit 检测的旧版 CTG 且不使用 CTG 8.x 时可能会出现。
    - 屏幕
    - 终端
- **Client\_to\_CTG\_Details**—此节点具有一个“客户端聚合图形摘要”选项卡。树的此节点下是表示各个客户端的子节点。
- **Client\_to\_CTG\_JSSE\_Session**—此节点具有一个“JSSE 到 CTG 的网关”选项卡。树的此节点下是几个聚合度量标准。

## 后端

提供以下后端 CTG 度量标准：

- CTG\_Global\_Statistics—此节点可显示统计信息：
  - CICS 聚合
  - 连接管理器
  - 连接管理器线程
  - 网关后台进程
  - HTTP 请求
  - 客户端的传入连接请求数
  - 各个服务器
  - 会话
  - 工作线程
- CTG\_to\_CICS\_ECI\_IPIC—此节点具有一个“ECI 所有网关图形摘要”选项卡。树的此节点下是具有每个客户端的度量标准的子节点。
- CTG\_to\_CICS\_EPI—此节点具有一个“EPI 所有网关”选项卡。树的此节点下是具有每个客户端的度量标准的子节点。
- JSSE\_to\_CTG—此节点具有一个“JSSE 图形摘要”选项卡。树的此节点下是一个显示“传入 JSSE 会话摘要”选项卡的子节点（显示两个度量标准）：
  - 传入的 SSL 合计握手
  - 每个时间间隔的传入 SSL 握手数

**注意：**在执行 IPIC 事务时，会立即显示“HTTP 请求”节点。

## 分析特定度量标准的历史数据

要分析特定度量标准的历史性能数据，请设置永久性集合。

**注意：**有关设置永久性集合的信息，请参阅《CA APM 安装和升级指南》。

## 服务器度量标准

本节包括有关 CTG 扩展中可用的后端和前端度量标准的信息。

## 后端 | CTG\_Global\_Statistics

选择“CTG\_Global\_Statistics”节点时，Introscope 调查器将显示“全局统计信息图形摘要”选项卡，其中显示以下几个度量标准：

- 每个时间间隔的客户端请求合计调用数
- 处理客户端请求响应时间
- 发送客户端回复响应时间
- 使用中的连接管理器线程
- 使用中的工作线程

通常，连接线程池用于处理客户端应用程序和 CTG 服务器之间的请求。而工作线程池用于在大型机上处理 CTG 服务器和 CICS 之间的请求。

“CTG\_Global\_Statistics”节点还包含“全局统计信息表格摘要”选项卡。此选项卡提供了一组运行的聚合统计信息，显示自 CTG 服务器启动以来流经该服务器的工作量。该页面显示以下统计信息：

- 接收的数据包总数
- 发送的数据包总数
- 使用中的连接管理器线程
- 使用中的工作线程

Introscope 调查器树中的“CTG\_Global\_Statistics”节点下提供了一整组有关流经 CTG 服务器的工作的全局统计信息。调查器树按分层形式排列，层次结构中的每个子节点显示有关 CTG 服务器中一组特定功能的关键统计信息。

“CICS 聚合”子节点显示有关在 CTG 服务器和上游 CICS 系统之间流动的流量的关键统计信息。该子节点包括如下统计信息：

- 每个时间间隔的读取数据包数和读取数据包响应时间（接收的 SNA 或 TCP 数据包）
- 每个时间间隔的写入数据包数和写入数据包响应时间（发送的 SNA 或 TCP 数据包）

**注意：**对于 JCA，“CommArea 聚合请求数据”度量标准会以不同的数据值显示。对于 Base，不显示度量标准。

“CICS WLM”子节点显示内部 WLM 如何排定要处理的请求，以及是否正在驱动任何用户编写的出口。

“连接管理器”子节点显示关于 CTG 服务器和请求的 CTG 客户端之间的流量处理的统计信息。

“连接管理器线程”子节点显示关于使用线程池（用于处理发送给客户端的连接管理器请求/回复）的统计信息。这些统计信息指示客户端是否发生任何线程瓶颈。

“网关后台进程”子节点显示流动请求的整体总数，以及正在执行的 CICS 事务提交或回滚（SYNCONRETURN 等）的相关总数。

“客户端的传入连接请求数”子节点显示每个时间间隔请求从客户端传入 CTG 服务器的到达率。

“工作线程”子节点显示关于使用线程池（用于将工作请求分发给 CICS）的统计信息。此统计信息可以指示是否发生任何上溯到 CICS 的线程瓶颈。

## “后端 | [CTG\\_to\\_CICS\\_ECI\\_IPIC](#)” 节点

选择 “CTG\_to\_CICS\_ECI\_IPIC” 节点时，Introscope 调查器将显示“ECI 所有网关图形摘要”选项卡，其中显示以下几个度量标准：

- 调用的程序合计数
- 每个时间间隔的程序调用数
- 程序平均响应时间 (毫秒)
- 程序合计错误

## “[CTG to CICS ECI 主机 ID](#)” 子节点

每个 CICS ECI 主机在 “CTG\_to\_CICS\_ECI\_IPIC” 父节点下都有自己的子节点。选择 “CTG to CICS ECI 主机 ID” 子节点时，Introscope 调查器查看器窗格将在 “ECI 网关表格摘要” 选项卡中显示以下度量标准：

- 程序合计计数
- 平均响应时间 (毫秒)
- 响应数/时间间隔
- 错误数/时间间隔
- 停顿

对于每个主机，树中将显示与在主机上运行的每个 CICS 程序相对应的一个更深入的子节点。

## “后端 | CTG\_to\_CICS\_EPI” 节点

选择 “CTG\_to\_CICS\_EPI” 节点时，Introscope 调查器将显示 “EPI 所有网关” 选项卡，其中显示以下几个度量标准：

- 调用的事务合计数
- 事务平均响应时间 (毫秒)
- 每个时间间隔的事务调用数
- 事务合计错误
- 调用的服务请求合计数
- 服务请求合计错误

## “CTG to CICS EPI 主机 ID” 子节点

每个 CICS EPI 主机在 “CTG\_to\_CICS\_EPI” 父节点下都有自己的子节点。

选择 “CTG to CICS EPI 主机 ID” 子节点时，Introscope 调查器查看器窗格将在 “EPI 聚合摘要” 选项卡中显示以下度量标准：

- 调用的事务合计数
- 事务调用/秒
- 事务平均响应时间 (毫秒)
- 事务合计错误

对于每个主机，树中将显示与在主机上运行的每个 CICS 程序相对应的一个更深入的子节点。

## “后端 | JSSE to CTG” 节点

选择 “JSSE\_to\_CTG” 节点时，Introscope 调查器将显示 “JSSE 图形摘要” 选项卡，其中显示以下两个度量标准：

- 传入的 SSL 合计握手
- 每个时间间隔的传入 SSL 握手数

在 “JSSE to CTG” 节点下，存在 “JSSE ServerSocket” 的子节点。

## “JSSE ServerSocket” 子节点

“JSSE ServerSocket” 子节点显示 “传入 JSSE 会话摘要” 选项卡，其中显示以下两个度量标准：

- 传入的 JSSE SSL 握手的合计数
- 每个时间间隔的传入 JSSE SSL 握手调用数

## “前端 | Client\_to\_CTG\_Aggregates” 节点

选择 “Client\_to\_CTG\_Aggregates” 节点时，Introscope 调查器将显示 “客户端聚合图形摘要” 选项卡，其中显示以下度量标准：

- TCP 合计打开套接字
- TCP 流聚合计数
- TCP 流响应时间 (毫秒)
- SSL 合计打开套接字
- SSL 流聚合计数
- SSL 流响应时间 (毫秒)

在 Introscope 调查器树的 “Client\_to\_CTG\_Aggregates” 节点下，存在以下度量标准（一些显示在选项卡视图中），以及若干子节点：

- 聚合 EPI 错误
- 事务合计错误
- 每个时间间隔的 CICS 资源异常
- 每个时间间隔的 CICS Txn Abend 异常
- 每个时间间隔的 EPI 网关异常
- 每个时间间隔的 EPI 请求异常
- 每个时间间隔的 资源异常
- SSL 聚合流
- SSL 流聚合计数
- SSL 流响应时间 (毫秒)
- TCP 合计打开套接字

- TCP 流聚合计数
- TCP 流响应时间 (毫秒)

以下各节讨论 “Client\_to\_CTG\_Aggregates” 节点下的子节点。

#### [“前端 | Client\\_to\\_CTG\\_Aggregates | BASE\\_ECI\\_EPI” 子节点](#)

选择此节点时, Introscope 调查器将显示 “客户端聚合摘要” 选项卡。此选项卡显示以下度量标准的图表:

- 调用的程序合计数
- 程序平均响应时间 (毫秒)
- 每个时间间隔的程序调用数
- 连接打开合计

#### [“前端 | Client\\_to\\_CTG\\_Aggregates | CCF\\_ECI” 子节点](#)

选择此节点时, Introscope 调查器将显示 “客户端聚合摘要” 选项卡。此选项卡显示以下度量标准的图表:

- 调用的程序合计数
- 程序平均响应时间 (毫秒)
- 每个时间间隔的程序调用数
- 连接打开合计

此选项卡类似于为 “BASE\_ECI\_EPI” 节点显示的选项卡。

#### [“前端 | Client\\_to\\_CTG\\_Aggregates | CCF\\_EPI” 子节点](#)

选择此节点时, Introscope 调查器将显示 “客户端聚合摘要” 选项卡。此选项卡显示以下度量标准的图表:

- 调用的事务合计数
- 事务平均响应时间 (毫秒)
- 每个时间间隔的事务调用数
- 连接打开合计

### “前端 | Client\_to\_CTG\_Aggregates | JCA\_ECI” 子节点

选择此节点时，Introscope 调查器将显示“客户端聚合摘要”选项卡。此选项卡显示以下度量标准的图表：

- 调用的 CICS 程序合计数
- 每个时间间隔的程序调用数
- 程序平均响应时间 (毫秒)
- 连接打开合计

### “前端 | Client\_to\_CTG\_Aggregates | JCA\_EPI” 子节点

选择此节点时，Introscope 调查器将显示“客户端聚合摘要”选项卡。此选项卡显示以下度量标准的图表：

- 调用的事务合计数
- 事务平均响应时间 (毫秒)
- 每个时间间隔的事务调用数
- 连接打开合计

### “前端 | Client\_to\_CTG\_Details” 节点

“Client\_to\_CTG\_Details” 节点跨所有网关提供端到端的聚合响应时间。选择此节点时，Introscope 调查器将显示“所有客户端网关表格摘要”选项卡。此选项卡显示以下列：

- 服务器
- 程序名称
- 聚合计数
- 平均响应时间 (毫秒)
- 响应数/时间间隔
- 错误数/时间间隔
- 停顿

## ECI 主机详细信息子节点

在“Client\_to\_CTG\_Details”节点下，包含与承载 CICS 程序的服务器相对应的子节点。选择这些服务器子节点之一时，Introscope 调查器查看器窗格将显示“客户端网关表格摘要”选项卡，其中带有以下度量标准的表显示：

- 聚合计数
- 平均响应时间 (毫秒)
- 响应数/时间间隔
- 错误数/时间间隔
- 停顿

## ECI 程序详细信息子节点

展开 ECI 主机子节点之一时，树中将为服务器上的每个程序显示一个更深入的子节点。选择这些程序节点将显示“程序详细信息视图”选项卡。此选项卡显示以下度量标准的图表：

- 请求聚合计数
- 每个时间间隔的调用数
- 平均响应时间 (毫秒)
- 聚合错误

## “前端 | Client\_to\_CTG\_Details” 事务视图节点

在 Introscope 调查器树中的“前端 | Client\_to\_CTG\_Details”节点下，每个子节点都与一个 EPI 主机相对应。

选择主机 ID 子节点时，Introscope 调查器查看器窗格将显示“客户端网关表格摘要”选项卡。此选项卡将为运行在主机上的每个程序显示以下度量标准：

- 聚合计数
- 平均响应时间 (毫秒)
- 响应数/时间间隔
- 错误数/时间间隔
- 停顿

在每个主机子节点下针对每个运行在主机上的程序包含一个更深入的子节点。选择这些程序子节点之一时，Introscope 调查器查看器窗格将显示“程序详细信息视图”选项卡，其中带有以下度量标准的图表：

- 请求聚合计数
- 每个时间间隔的调用数
- 平均响应时间 (毫秒)
- 聚合错误

## “前端 | Client\_to\_CTG\_JSSE 会话”节点

“Client\_to\_CTG\_JSSE 会话”节点跨所有网关提供端到端的聚合响应时间。选择此节点时，Introscope 调查器将显示“JSSE 到 CTG 的网关”选项卡。此选项卡显示以下度量标准的图表：

- 创建的 SSL 会话合计数
- SSL 握手平均响应时间 (毫秒)
- 每个时间间隔的 SSL 会话调用数
- 每个时间间隔的 SSL 加密数据请求数

在 Introscope 调查器树中，以下度量标准显示在“Client\_to\_CTG\_JSSE 会话”节点下：

- 聚合 SSL 数据流
- 聚合 SSL 握手
- 聚合 SSL 打开套接字
- SSL 数据流响应时间 (毫秒)
- 每个时间间隔的 SSL 数据流
- SSL 握手分配响应时间 (毫秒)
- 停顿计数

## 使用 Introscope 调查器选项卡视图

您可以使用 Introscope 调查器查看器窗格顶部的选项卡更改显示在该窗格中的视图。树上许多特定于 CTG 的节点具有特殊的开箱即用型摘要视图。

不管您使用的扩展为何，以下为 Introscope Workstation 中提供的标准选项卡。

- 常规—当选择 Introscope 调查器树中某个代理下的任何项时，“常规”选项卡为默认选项卡。当选择某个度量标准时，“常规”选项卡是该度量标准的可视化表示—不论其是实时数据，还是选定的历史时段。对于树中的节点，“常规”选项卡显示指向 Introscope 调查器层次结构中相应节点对象的路径。
- 概览—在 Introscope 调查器树中选择代理时会显示“概览”选项卡，可启用应用程序监控。此信息提供高级别运行状况指示符、相关事件日志和历史度量标准信息。
- 搜索—在包含度量标准的 Introscope 调查器树中选择某节点时会显示“搜索”选项卡，利用此选项卡可快速查找度量标准。
- 跟踪—在 Introscope 调查器树中选择某资源或组件时会显示“跟踪”选项卡，其类似于跟踪查看器。此信息列出了当前选定资源或组件参与的事务跟踪。
- 错误—在 Introscope 调查器树中选择某资源或组件时会显示“错误”选项卡，该选项卡上列出了选定项的错误和错误详细信息。

**注意：**有关标准 Introscope 选项卡的详细信息，请参阅《CA APM Workstation 用户指南》。

## 在显示板中查看 CTG 扩展数据

CTG 扩展提供了大量预配置的 Introscope 显示板和报警。

CTG 客户端或服务器的所有显示板都以 CTGClient 或 CTGServer 开头，以便与安装在 Introscope 中的其他管理模块区分开来。

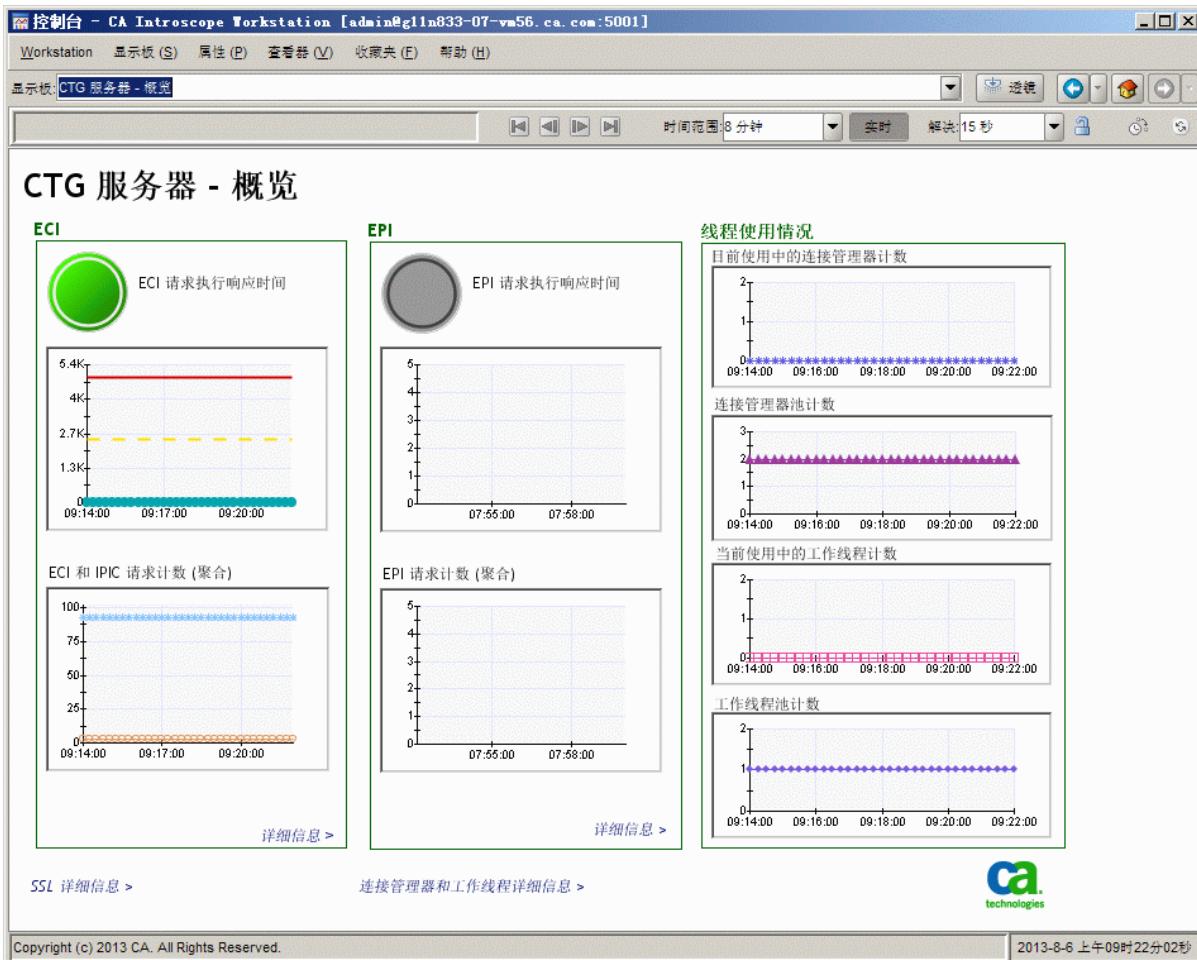
要查看显示板，请启动 Introscope Workstation，并打开控制台。CTG 扩展也为许多显示在这些显示板中的度量标准附带默认的警告/危险报警阈值。

**注意：**由于您的 CTG 应用程序的唯一模式，CTG 扩展附带的一些面板和显示板可以留空。

## “CTG 服务器 - 概览” 显示板

“CTG 服务器 - 概览” 显示板显示了三个由 Introscope 监控的主要 CTG 服务器区域的摘要。此显示板还包括可提供这些区域详细信息的显示板以及以下显示板的链接：

- CTG 客户端 - Java 网关及 SSL
- CTG 服务器 - 连接管理器和工作线程



交通灯以可视方式表示是否有任何基本度量标准已超过其定义的阈值。

“线程使用情况” 部分显示了 CTG 当前使用连接和工作线程的情况。通常，当客户端连接到 CTG 时分配连接线程，而当客户端与 CTG 断开连接时释放连接线程。对于在 CTG 和 CICS 之间流动的每个活动请求，分配工作线程。

显示在“CTG 服务器 - 概览”显示板上的度量标准包括：

- ECI
  - ECI 请求执行响应时间
  - ECI 请求计数
- EPI
  - EPI 请求执行响应时间
  - EPI 请求计数

“当前使用中”度量标准反映通过 CTG 服务器代码处理请求（或回复）过程中的线程数量。“池计数”度量标准反映为该实体（连接管理器或工作线程）分配的线程数量。池计数最初是基于 *initconnect* 参数（用于连接管理器线程）和 *initworker* 参数（用于工作线程）建立的，如在 *ctg.ini* 文件中所定义。随着工作负荷增加，各自的池计数将增长，直到由 *ctg.ini* 的 *maxconnect* 和 *maxworker* 参数指定的最大值。您可以使用“池计数”中记录的峰值来微调 *ctg.ini* 文件中相应的 *maxconnect* 和 *maxworker* 参数。

## “CTG 摘要 - 服务器 ECI 活动”显示板

“CTG 摘要 - 服务器 ECI 活动”显示板提供了可能导致问题的执行和对象的简要快照。

显示在“CTG 摘要 - 服务器 ECI 活动”显示板上的度量标准包括：

- ECI 所有网关 - 聚合摘要
  - 调用的程序数
  - 每个时间间隔的程序调用数
  - 程序平均响应时间
- 网络聚合
  - 每个时间间隔新接受的 TCP 套接字
  - 每个时间间隔的 JSSE/SSL 调用数
  - TCP 套接字输入带宽
  - TCP 套接字输出带宽
  - JSSE/SSL 新合计计数

**注意：**要显示“TCP 套接字输入带宽”和“TCP 套接字输出带宽”，请在 *IntroscopeAgent.profile* 中设置属性 *introscope.agent.sockets.reportRateMetrics=true*。

## “CTG 服务器 - 全局统计信息” 显示板

“CTG 服务器 - 全局统计信息” 显示板从 CTG 服务器的角度提供了整体 CTG 统计信息的简要快照。使用上部角落中的链接，您可以导航到 “CTG 服务器 - 概览” 显示板。

显示在 “CTG Request Exit - 全局统计信息” 显示板上的度量标准包括：

- 已处理的 I/O 请求 (聚合)
- 每个时间间隔处理的 I/O 请求数
- 客户端的传入连接请求数 (每个时间间隔)
- 对 CICS 的 ECI/IPIC/EPI 总请求 (每个时间间隔)
- 当前使用中的连接管理器线程
- 连接管理器线程池计数
- 当前使用中的工作线程
- 工作线程池计数

## “CTG 服务器 - ECI/IPIC 请求” 显示板

“CTG 服务器 - ECI 请求” 显示板提供了可能导致问题的执行和对象的简要快照。使用上部角落中的链接，您可以导航到 “CTG 服务器 - 概览” 显示板。

显示在此显示板上的度量标准包括：

- 执行性能
  - 请求执行时间
  - 执行列表时间
  - 请求执行计数
  - 执行列表计数
- 请求初始化/终止和写入/读取对象
  - 写入/读取对象时间
  - 写入/读取对象计数

“写入/读取对象” 是 IBM CTG 术语，用于表示在 CTG 客户端和 CTG 服务器之间交换的服务消息（数据包）。这组测量值显示发送的此类消息总数及其平均响应时间。如果 CTG 服务器变得超载（如工作线程或连接线程太少），这些消息会排队等待可用的线程，而响应时间将增加。

## “CTG 服务器 - EPI 请求” 显示板

“CTG 服务器 - EPI 请求” 显示板提供了服务器请求的性能摘要。使用上部角落中的链接，您可以导航到“CTG 服务器 - 概览”显示板。

显示在“CTG 服务器 - EPI 请求”显示板上的度量标准包括：

- 执行性能
  - 事务执行时间
  - 聚合事务开始计数
- 读取/写入对象
  - 写入/读取消耗时间
  - 写入/读取消耗计数
- CICS CP 请求
  - CP 请求执行时间
  - CP 请求执行计数

如果 CTG 无法建立到 CICS 的 EPI 会话，则不会填充以下度量标准：

- 事务执行时间
- 写入/读取消耗时间
- 聚合事务开始计数
- 写入/读取消耗计数

而是会记录错误消息（位于“错误”选项卡下），以指示导致连接失败的错误。

## “CTG 服务器 - 连接管理器和工作线程” 显示板

“CTG 服务器 - 连接管理器和工作线程” 显示板提供了可能导致问题的连接管理器和工作线程的性能的简要快照。使用上部角落中的链接，您可以导航到“CTG 服务器 - 概览” 显示板。

该显示板显示以下度量标准：

- 连接管理器性能
  - 分发响应时间
  - 发送回复响应时间

- 分发计数
- 发送回复计数
- 工作线程性能
  - 分发响应时间
  - 运行/关闭响应时间
  - 工作线程分发计数
  - 运行/关闭计数

## “CTG 服务器 - SSL” 显示板

“CTG 服务器 - SSL” 显示板提供了可能导致问题的 SystemSSL 和 JSSE SSL 的性能的简要快照。使用上部角落中的链接，您可以导航到“CTG 服务器 - 概览”显示板。

显示在“CTG 服务器 - SSL” 显示板上的度量标准包括：

- JSSE/SSL 服务器套接字
  - 套接字接受/关闭响应时间
  - 套接字接受/关闭计数
- SSL 套接字 (已有)
  - 套接字关闭响应时间
  - 套接字关闭计数
- JSSE 套接字
  - JSSE 套接字关闭响应时间
  - JSSE 套接字关闭计数

## “CTG 客户端 - 概览” 显示板

“CTG 客户端 - 概览” 显示板显示了三个由 Introscope 监控的主要 CTG 服务器区域的摘要。该显示板还提供指向某些显示板的链接，这些显示板提供了这三个区域的更多详细信息。交通灯以可视方式表示是否有任何基本度量标准已超过其定义的阈值。

显示在“CTG 客户端 - 概览”显示板上的度量标准包括：

- Java 网关及 SSL 会话
  - 网关流响应时间
  - JSSE SSL 会话分配响应时间
- EPI
  - EPI 连接操作响应时间
  - EPI 终端事务操作响应时间
- 终端及终端请求
  - 终端及终端请求发送响应时间
  - 终端连接及断开响应时间

## “CTG 客户端 - Java 网关及 SSL 会话” 显示板

“CTG 客户端 - Java 网关及 SSL 会话” 显示板提供了可能导致问题的网关和会话的简要快照。使用上部角落中的链接，您可以导航到“CTG 客户端 - 概览”显示板。

显示在“CTG 客户端 - Java 网关及 SSL 会话” 显示板上的度量标准包括：

- Java 网关性能
  - 流响应时间
  - 当前活动会话计数
  - TCP 会话调用合计计数
  - 打开响应时间
  - 关闭响应时间
- JSSE SSL 会话性能
  - JSSE 聚合 SSL 会话计数
  - JSSE SSL 会话分配时间
  - 每个时间间隔的 JSSE SSL 会话调用数

## “CTG 客户端 - EPI 请求” 显示板

“CTG 客户端 - EPI 请求” 显示板显示了网关连接的快照。

该显示板显示以下度量标准:

- 网关响应时间度量标准:
  - EPI 网关连接操作响应时间
  - EPI 网关连接合计计数
- 终端及基本屏幕处理程序度量标准:
  - EPI 终端连接操作响应时间
  - EPI 终端事务操作响应时间
  - EPI 基本屏幕处理程序连接计数

## “CTG 客户端 - EPI 终端请求” 显示板

“CTG 客户端 - EPI 终端请求” 显示板提供了可能导致问题的终端和终端请求活动的简要快照。使用上部角落中的链接，您可以导航到“CTG 客户端 - EPI 终端请求” 显示板。

显示在此显示板上的度量标准包括:

- 终端活动及性能
- 发送响应时间
- 连接及断开时间
- 发送计数
- 连接计数

## “Introscope ChangeDetector” 显示板

“Introscope ChangeDetector” 显示板提供了对可能导致问题的 CTG 配置文件和 .JAR 文件所做更改的摘要。

显示在“Introscope ChangeDetector” 显示板上的度量标准包括:

- 每个时间间隔检测到的更改
- 每个时间间隔检测到的多数变更
- 自代理上次重新启动以来的更改总数
- 自代理上次重新启动以来已完成的扫描

## 修改 Introscope 警告/危险报警阈值

许多 CTG 扩展显示板都包括报警。

### 修改 Introscope 警告/危险报警阈值:

1. 在 Introscope 调查器内，转至“管理模块”>“CTGCLIENT”>“报警”，选择报警。  
报警定义位于底部窗格中。
2. 在报警定义窗格中，修改“时期”字段，以指定您要将性能度量标准与警告阈值和危险阈值进行比较的频率。
3. 指定适用于您组织的服务级别的警告阈值和危险阈值。
4. 单击“应用”应用更改。

## CTG 事务跟踪器

CTG 事务跟踪器提供按服务器名称和程序名称进行跟踪的功能。此功能主要用于在 CTG 客户端（如 WebSphere）和 CTG 服务器之间进行事务跟踪。不需要结合 CICS 大型机跨产品事务跟踪器 SYSVIEW 功能。该跟踪器还支持在 CommArea 中提取和包括客户指定的“用户关联”字段，执行响应时间阈值保留，并跟踪特定程序名称。借助此功能，您可以使用代理属性条目进行自定义，并基于响应时间阈值、按程序名称或通过优先编写自己的自定义扩展，来创建增值 CTG 跟踪。

### PP CTG 事务跟踪器对特殊字符的处理

Introscope Workstation 上运行的 Introscope 事务跟踪关联逻辑严格限制在用户生成的关联 ID 中使用特殊字符。Introscope Workstation 关联支持不允许使用的任何特殊字符都将自动转换为下划线。

以下特殊字符将始终转换为下划线：

: - + ( ) [ ] \* " ~ ^ ?

以下附加的一组特殊字符在默认情况下也转换为下划线：

\ / { } | , . ; = ' < >

但是，如果您要保留这些特殊字符（并允许使用所有特殊字符和跟踪），请将以下属性添加到 IntroscopeAgent.profile 中：

`introscope.ctg.trantracer.corrid.specialcharstrip=false`

该值的默认值为“true”，该值会将所有特殊字符隐藏为下划线，并关联前端和后端跟踪。

如果要删除关联 ID 中的重复下划线（将几个下划线减少为一个），则必须在 `IntroscopeAgent.profile` 中设置以下属性：

```
introscope.ctg.trantracer.corrid.underscorestrip=true
```

## CTG 事务跟踪器属性的详细列表

下表显示了可在 `IntroscopeAgent.profile` 文件中配置的 CTG 事务跟踪器属性的详细列表：

属性	用法
<code>introscope.ctg.trantracer.publish.metrics</code>	布尔值，指示是否应启用任何简单度量标准（响应时间和调用计数）。请注意，这意味着它是“轻量型”跟踪器。 默认值 = <code>true</code>
<code>introscope.ctg.trantracer.publish.trantrace</code>	布尔值，指示是否应启用事务跟踪。 默认值 = <code>true</code>
<code>introscope.ctg.trantracer.corridscan.offset</code>	整数值，指示用户关联 ID 与 CommArea 起点之间的偏移量。 默认值 = <code>147</code>
<code>introscope.ctg.trantracer.corridscan.length</code>	整数值，指示将提取的用户关联 ID 的长度。 默认值 = <code>100</code>
<code>introscope.ctg.trantracer.corrid.format.id</code>	布尔值，指示是否应在用户关联 ID 中扫描 EM TranTracer 字符串条目中不允许使用的非法字符 [例如，“.”（冒号）和“-”（破折号）]。
<code>introscope.ctg.trantracer.corridscan.starttrace</code>	布尔值，指示是在开始跟踪还是完成跟踪时提取用户关联 ID。 默认值 = <code>false</code>
<code>introscope.ctg.trantracer.rspthreshhold.value</code>	整数值，指示此跟踪器允许的最大响应时间阈值。超过该值的响应时间将事务跟踪到 EM。低于该值的值不进行跟踪。 默认值 = <code>0</code> = 不执行 RSP 阈值检查

属性	用法
introscope.ctg.trantracer.program.name	字符串，指示要匹配的 CICS 程序的完整或部分名称。匹配的值将事务跟踪到 EM。不匹配的值不进行跟踪。 默认值 = <b>null</b> = 不进行程序匹配
introscope.ctg.trantracer.program.match.criteria	整数值，定义应执行的程序名称匹配类型。  1 = 全名匹配 2 = 程序名称以上面的名称字符串开头 3 = 程序名称以上面的名称字符串结尾 0 = 不进行匹配。 默认值 = <b>0</b> (不进行匹配)。
introscope.ctg.trantracer.metric.HLQname	字符串，指示调查器树的高级名称。 默认为：“后端   CTG_to_CICS_ECI_IPIC_Trace” 和“前端  Client_to_CTG_Trace”
introscope.ctg.trantracer.debug	布尔值，指示是否应将调试消息跟踪到代理日志。 默认值 = <b>false</b>

# 附录 A: CTG 性能度量标准

---

本附录介绍了 Introscope 调查器中显示的 CTG 类和方法，CTG 扩展会为这些类和方法收集性能度量标准。

此部分包含以下主题：

- [前端度量标准 \(p. 59\)](#)
- [“后端”度量标准 \(p. 61\)](#)
- [CTG 显示板度量标准 \(p. 64\)](#)
- [Request Exit 度量标准 \(p. 69\)](#)

## 前端度量标准

前端度量标准被 CTG 客户端检测捕获。数据从应用程序服务器（如 WebSphere）或已检测的用户编写的 CTG 客户端应用程序（正在 WebSphere 外部独立运行）内被捕获。前端度量标准用于捕获和测量正在运行的 CTG 应用程序类型：JCA、CCF 或者 ECI 或 EPI 之一的基类。

度量标准在聚合级别（例如，Websphere 应用程序服务器调用的所有 ECI 程序）捕获。详细的度量标准在单个程序或事务级别（例如，调用的单个 ECI 程序的统计信息）捕获。许多度量标准（特别是详细的度量标准）使用 BlamePoint 跟踪器，因此提供了一种通用格式：

度量标准	定义
请求聚合计数	启动对程序或事务的请求的总次数。如果打开连接失败（见下文），此度量标准可能不出现。 <b>注意：</b> 请参阅“聚合服务计数”中的“注意”。
聚合服务计数	EPI 使用不执行任何实际 CICS 事务的多个设置和拆卸流（AddTerminal、PurgeTerminal、DeleteTerminal）。它们仅仅是设置与 CICS 的活动 3270 连接的服务请求。此度量标准保留所有这些服务类型请求的聚合计数。 <b>注意：</b> “聚合请求计数”记录实际的 3270 事务调用和响应（也就是通过 EPI 请求发送 3270 缓冲区和来自 CICS 事务的关联 3270 回复缓冲区）。
聚合错误计数	对打开连接或者程序或事务启动的请求由于错误而失败的总次数。此度量标准仅在发生实际错误时才会显示。

度量标准	定义
平均响应时间	从客户端处理和测量请求的平均响应时间（也就是往返总时间）。
并发调用	程序或事务请求有多少并发调用处于运行状态。
每个时间间隔的错误	在最近的测量时间间隔（通常为 7-15 秒）期间报告的错误总数。
每个时间间隔的响应数	在最近的测量时间间隔期间报告的 CTG 回复总数。
停顿计数	在已启动对程序的请求，但 CICS 在停顿检测时间限制内未响应该请求时，指示是否检测到任何停顿。停顿检测限制由 <i>IntroscopeAgent.profile</i> 中的 <i>introscope.agent.stalls.thresholdseconds=nn</i> 属性设置。 <i>nn</i> 值以秒为单位指定停顿检测时间。

对于 Client\_to\_CTG\_Aggregates，跟踪以下附加度量标准：

度量标准	定义
聚合打开	发出的从 CTG 客户端到 CTG 服务器的 TCP 或 SSL 连接打开的总数。只有执行 CTG 打开连接后，才能启动 CTG 请求。
打开响应时间	从客户端处理和测量连接打开请求的平均响应时间。也就是打开 CTG 客户端到 CTG 服务器的连接花费的往返总时间。

## 前端客户端到 CTG 聚合

这些度量标准提供发出的从 CTG 客户端到 CTG 服务器的所有 ECI 和 EPI 调用的聚合计数。为了方便起见，调用按照 CTG API 类型进行分类。下表中列出的所有度量标准都使用 BlamePoint 度量标准。

如果任何打开连接请求失败，可以在显示在 Client\_to\_CTG\_Aggregates 树下的“打开错误”选项卡下找到更多详细信息。

度量标准	定义
BASE_ECI_EPI	使用基础 ECI 类 API 的所有 ECI 调用，外加使用基础 EPI 类 API 的所有 EPI 调用。
JCA_ECI	使用更新的 Java 连接器体系结构 (JCA) ECI API 的所有 ECI 调用。
JCA_EPI	使用更新的 Java 连接器体系结构 (JCA) EPI API 的所有 EPI 调用。
停顿计数	在已启动对程序的请求，但 CICS 在停顿检测时间限制内未响应该请求时，指示是否检测到任何停顿。停顿检测限制由 <code>IntroscopeAgent.profile</code> 中的 <code>introscope.agent.stalls.thresholdseconds=nn</code> 属性设置。 <code>nn</code> 值以秒为单位指定停顿检测时间。

## 前端客户端到 CTG 详细信息

这些度量标准提供 CTG 客户端调用的每个程序或事务的详细计数。为了方便起见，CLI（网关）服务器对调用进行分类。在每个网关服务器的下面是调用的每个 ECI 或 EPI 程序和相应程序的详细度量标准列表。

## “后端”度量标准

后端度量标准在由 Introscope 检测的 CTG 服务器内捕获。“后端”度量标准用于捕获和测量客户端和 CICS 系统之间的 ECI 和 EPI 请求流。

度量标准在聚合级别（例如，客户端调用的所有 ECI 程序）捕获。详细的度量标准在单个程序级别（例如，调用的单个 ECI 程序统计信息）捕获。

在每个网关服务器的下面是调用的每个 ECI 或 EPI 程序和相应程序的详细度量标准列表。详细的度量标准以通用格式使用 **BlamePoint** 跟踪器：

度量标准	定义
聚合计数	启动对程序或事务的请求的总次数。如果打开连接失败（见下文），此度量标准可能不出现。
请求聚合计数	
聚合错误计数	对打开连接或者程序或事务启动的请求由于错误而失败的总次数。此度量标准仅在发生实际错误时才会显示。
平均响应时间	处理请求的平均总往返时间。
每个时间间隔的错误	在最近的测量时间间隔期间报告的错误总数。
每个时间间隔的响应数	在最近的测量时间间隔期间报告的 CTG 回复总数。
停顿计数	检测到的停顿总数。在已启动对程序的请求，但 CICS 在停顿检测时间限制内未响应该请求时，停顿就会发生。停顿检测限制由 IntroscopeAgent.profile 中的 introscope.agent.stalls.thresholdseconds=nn 属性设置，其中 nn 是以秒为单位的停顿检测时间。

## 后端 CTG\_to\_CICS\_ECI\_IPIC 聚合

这些度量标准提供在 CTG 客户端和相应的 CICS 服务器之间发出的所有 ECI/IPIC 调用的聚合计数。如表中所述，在每个网关服务器的下面是使用 BlamePoint 跟踪器度量标准调用的每个 ECI/IPIC 程序的列表。以下聚合度量标准基于网关服务器提供：

度量标准	定义
程序聚合计数	启动对 ECI/IPIC 程序的请求的总聚合次数。
程序平均响应时间	从 CTG 服务器到 CICS 处理和测量 ECI/IPIC 请求的平均聚合响应时间。也就是 CTG 服务器和 CICS 之间的往返总时间。
每个时间间隔的程序调用数	最近的测量时间间隔期间报告的对 ECI/IPIC 程序的调用总数。

## 后端 CTG 全局统计信息

度量标准	定义
处理客户端请求响应时间	在客户端请求中读取并转发给 CICS 所需的总时间。
发送客户端回复响应时间	在来自 CICS 的回复中读取并转发回请求客户端所需的总时间。
使用中的连接管理器线程	当前用于处理客户端连接的线程数。
使用中的工作线程	当前用于处理 CICS 请求的线程数。

## 后端 CTG\_to\_CICS\_EPI 聚合度量标准

CTG\_to\_CICS\_EPI 度量标准仅在 CTG 的分布式（非 z/OS）版本上可用。这些度量标准提供在 CTG 客户端和相应的 CICS 服务器之间发出的 EPI 调用的聚合计数。

度量标准	定义
事务聚合计数	启动对 EPI 事务的请求的总聚合次数。
事务平均响应时间	从 CTG 服务器到 CICS 处理和测量 EPI 请求的平均聚合响应时间。也就是 CTG 服务器和 CICS 之间的往返总时间。
每个时间间隔的事务调用数	最近的测量时间间隔期间报告的对 EPI 事务的调用总数。

**注意：**如表中所述，在每个网关服务器的下面是使用 BlamePoint 跟踪器度量标准调用的每个 EPI 事务的列表。

## 后端 CTG\_to\_CICS 线程

这些度量标准显示 CTG 服务器对连接和工作线程的当前使用情况。

## CTG 显示板度量标准

以下各表显示在扩展显示板中显示的标签与特定的 CTG 性能度量标准之间的对应关系。

右边的列描述了度量标准在 Introscope 调查器中的位置。

## “CTG 客户端 - 概览” 显示板

标签	CTG 性能度量标准
网关流响应时间	CTG 客户端   {. +} Java 网关:打开响应时间 (毫秒)
JSSE SSL 会话分配响应时间	Client_to_CTG   JSSE 会话:SSL 握手分配响应时间 (毫秒)
EPI 连接操作响应时间	CTG 客户端   EPI   网关:打开响应时间 (毫秒) CTG 客户端   EPI   终端:{连接 断开 进程 连接 进程断开 终止} 响应时间 (毫秒)
EPI 终端事务响应时间	CTG 客户端   EPI   终端:{发送 设置事务数据  开始事务 终止} 响应时间 (毫秒)

标签	CTG 性能度量标准
终端及终端请求发送响应时间	CTG 客户端   {终端及终端请求}:发送响应时间 (毫秒)
终端连接及断开响应时间	CTG 客户端   终端{连接 断开} 响应时间 (毫秒)

## “CTG 客户端 - EPI” 显示板

标签	CTG 性能度量标准
EPI 网关连接操作响应时间	CTG 客户端   EPI   网关:{.*} 响应时间 (毫秒)
EPI 网关连接合计计数	CTG 客户端   EPI   网关:{.*} 计数
EPI 终端连接操作响应时间	CTG 客户端   EPI 终端:{连接 断开 进程连接 进程断开 终止} 响应时间 (毫秒)
EPI 终端事务操作响应时间	CTG 客户端   EPI 终端{发送 设置事务数据 开始事务 终止} 响应时间 (毫秒)
EPI 监视器及终端连接计数	CTG 客户端   EPI 终端:计数 CTG 客户端   EPI 监视器:终端连接计数
EPI 基本屏幕处理程序连接计数	CTG 客户端   EPI 基本屏幕处理程序:终端连接计数

## “CTG 客户端 - Java 网关及 SSL” 显示板

标签	CTG 性能度量标准
Java 网关: 流响应时间	CTG 客户端   {.+} Java 网关:流响应时间 (毫秒)
Java 网关: 总活动会话计数	CTG 客户端   {.+} Java 网关:计数
Java 网关: 打开响应时间	CTG 客户端   {.+} Java 网关:打开响应时间 (毫秒)
Java 网关: 关闭响应时间	CTG 客户端   {.+} Java 网关:关闭响应时间 (毫秒)
JSSE 聚合 SSL 会话计数	Client_to_CTG   JSSE 会话:SSL 合计打开套接字数

标签	CTG 性能度量标准
JSSE SSL 会话分配时间 (毫秒)	Client_to_CTG   JSSE 会话:SSL 握手分配响应时间
每个时间间隔的 JSSE SSL 会话调用数	Client_to_CTG   JSSE 会话:每个时间间隔的 SSL 合计 套接字

## “**CTG 客户端 - 终端及终端请求**” 显示板

标签	CTG 性能度量标准
终端发送计数	CTG 客户端   EPI 网关:{.*} 响应时间 (毫秒)
终端连接及断开计数	CTG 客户端   EPI 网关:{.*} 计数
终端请求发送响应时 间	CTG 客户端   终端请求:发送响应时间 (毫秒)
连接及断开响应时间	CTG 客户端   终端请求:{释放   分配} 响应时间 (毫秒)

## “**CTG 服务器 - 概览**” 显示板

标签	CTGServer 性能度量标准
ECI 请求执行响应时 间	CTG 服务器   服务器 ECI 请求   执行响应时间 (毫秒)
ECI 请求计数	CTG 服务器   服务器 ECI 请求   执行计数
EPI 请求执行响应时 间	CTG 服务器   服务器 EPI 请求   执行响应时间 (毫秒)
EPI 请求计数	CTG 服务器   服务器 EPI 请求   执行计数

## “**CTG 服务器 - 连接管理器和工作线程**” 显示板

标签	WebSphere 性能度量标准
连接管理器: 分发响 应时间	CTG 服务器   连接管理器: 踢除响应时间 (毫秒)
连接管理器: 分发计 数	CTG 服务器   连接管理器: 踢除计数

标签	WebSphere 性能度量标准
连接管理器: 发送回复响应时间	CTG 服务器   连接管理器:发送回复响应时间 (毫秒)
连接管理器: 发送回复计数	CTG 服务器   连接管理器 发送回复计数
工作线程: 分发响应时间	CTG 服务器   工作线程:踢除响应时间 (毫秒)
工作线程: 分发计数	CTG 服务器   工作线程:踢除计数
工作线程: 运行/关闭响应时间	CTG 服务器   工作线程:{关闭 运行} 响应时间 (毫秒)
工作线程: 运行/关闭计数	CTG 服务器   工作线程:{关闭 运行} 计数

## “CTG 服务器 - ECI 请求”显示板

标签	WebSphere 性能度量标准
请求执行时间	CTG 服务器   服务器 ECI 请求:执行响应时间 (毫秒)
执行列表时间	CTG 服务器   服务器 ECI 请求:执行列表响应时间 (毫秒)
请求执行计数	CTG 服务器   服务器 ECI 请求:执行计数
执行列表计数	CTG 服务器   服务器 ECI 请求:执行列表计数
请求初始化/终止时间	CTG 服务器   服务器 ECI 请求:{初始化 终止} 响应时间 (毫秒)
写入/读取对象时间	CTG 服务器   服务器 ECI 请求:{读取 写入} 对象响应时间 (毫秒)
请求初始化/终止计数	CTG 服务器   服务器 ECI 请求:{初始化 终止} 计数
写入/读取对象计数	CTG 服务器   服务器 ECI 请求:{读取 写入} 对象计数

## “CTG 服务器 - EPI 请求” 显示板

标签	WebSphere 性能度量标准
请求执行时间	CTG 服务器 服务器 EPI 请求:执行响应时间 (毫秒)
执行列表时间	CTG 服务器 服务器 EPI 请求:执行列表响应时间 (毫秒)
请求执行计数	CTG 服务器 服务器 EPI 请求:执行计数
执行列表计数	CTG 服务器 服务器 EPI 请求:执行列表计数
请求初始化/终止时间	CTG 服务器 服务器 EPI 请求:{初始化 终止} 响应时间 (毫秒)
写入/读取对象时间	CTG 服务器 服务器 EPI 请求:{读取 写入} 对象响应时间 (毫秒)
CicsCp 请求执行响应时间	CTG 服务器 服务器 CicsCp 请求 执行响应时间 (毫秒)
CicsCp 请求计数	CTG 服务器 服务器 CicsCp 请求 执行计数
请求初始化/终止计数	CTG 服务器 服务器 EPI 请求:{初始化 终止} 计数
写入/读取对象计数	CTG 服务器 服务器 EPI 请求:{读取 写入} 对象计数

## “CTG 服务器 - 全局统计信息” 显示板

标签	CTGServer 性能度量标准
已处理的 I/O 请求 (聚合)	已处理的 TCP I/O 请求的总数。
每个时间间隔处理的 I/O 请求数	在上一个 15 秒时间间隔中处理的 TCP I/O 请求的数目。
客户端的传入连接请求数 (每个时间间隔)	从客户端接收的在上一个 15 秒时间间隔中发生的新请求的数目。
对 CICS 的 ECI/IPIC/EPI 总请求 (每个时间间隔)	在上一个 15 秒时间间隔中处理的请求总数 (ECI、IPIC 和 EPI)。
当前使用中的连接管理器线程	当前用于处理客户端连接的线程数。

标签	CTGServer 性能度量标准
连接管理器线程池计数	当前分配用于处理客户端连接的线程数。
当前使用中的工作线程	当前用于处理 CICS 请求的线程数。
工作线程池计数	当前分配用于处理 CICS 请求的线程数。

## “CTG 服务器 - SSL” 显示板

标签	WebSphere 性能度量标准
JSSE SSL 聚合传入 SSL 握手	传入的 SSL 握手的总数
每个时间间隔的 JSSE SSL 传入 SSL 握手	每个时间间隔的 SSL 握手数

## Request Exit 度量标准

下表显示了 Request Exit 功能的度量标准。

### 后端 | CTG\_to\_CTG\_ECI\_IPIC\_RequestExit 度量标准

下表详细介绍了调查器节点下显示的“后端 | CTG\_to\_CICS\_ECI\_IPIC\_RequestExit”的度量标准。

详细的度量标准名称	说明
程序平均响应时间 (微秒)	所有程序的平均响应时间
每个时间间隔的程序调用数	每个时间间隔所有程序的总调用数
程序聚合计数 ECI_IPIC	调用程序的总次数
程序聚合错误数	所有程序的合计错误数
{服务器名称 程序名称}: 平均响应时间 (微秒)	程序的平均响应时间

详细的度量标准名称	说明
{服务器名称 程序名称}: 每个时间间隔的响应数	程序每个时间间隔的响应数（调用数）
{服务器名称 程序名称}: 并发调用数	程序在给定时间间隔（15 秒）内的并发调用数
{服务器名称 程序名称}: 停顿/挂起数	在给定时间间隔（15 秒）内为程序记录的停顿/挂起数
{服务器名称 程序名称}: 错误数	在给定时间间隔（15 秒）内为程序记录的错误数
{服务器名称 程序名称}: 程序合计计数 ECI_IPIC	调用程序的总合计数
{服务器名称 程序名称}: 合计错误	调用程序时报告错误的总合计数
{服务器名称 程序名 称}:CommArea 聚合请求数据	该程序发送给 CICS 的 COMMAREA 数据总字节数
{服务器名称 程序名 称}:CommArea	该程序从 CICS 接收的 COMMAREA 数据总字节数

## “CTG\_Global\_Statistics\_RequestExit” 度量标准

此表详细介绍了“CTG\_Global\_Statistics\_RequestExit”的度量标准。度量标准的详细名称与 IBM 分配给度量标准的名称相对应。此表有助于最大程度减少 Introscope 所用度量标准名称与 IBM 所用度量标准定义之间的冲突。

详细的度量标准名称	说明
CICS 服务器:CICS 请求数	发送给所连接的 CICS 服务器的请求数据量（字节）。此数量包括应用程序和 CICS 协议数据。
CICS 服务器:CICS 响应数	从所连接的 CICS 服务器接收的响应数据量（字节）。此数量包括应用程序和 CICS 协议数据。
CICS 服务器:当前已安装终端数	当前已建立并安装的 (EPI) 终端会话数。
CICS 服务器:当前孤立的 CICS 请求数	当前等待 CICS 响应而其所属应用程序已超时或结束的请求数。
CICS 服务器:最大活动请求数	配置文件中定义的最大活动请求数。

详细的度量标准名称	说明
CICS 服务器:CICS 请求数 据量	发送给所连接的 CICS 服务器的请求数据量（字节）。此数量包括应用程序和 CICS 协议数据。
CICS 服务器:CICS 服务 器数	网关后台进程已尝试向其发送请求的 CICS 服务器的数目。
CICS 服务器:IPIC 会话失 败数	针对 CICS 服务器的 IPIC 会话的失败数。
CICS 服务器:使用中的 IPIC 会话数	正由 CICS 服务器使用的 IPIC 会话数。
CICS 服务器:活动请求数	客户端后台进程中当前运行的活动请求数。
CICS 服务器:连接失败数	尝试连接到 CICS 服务器的失败次数。
CICS 服务器:定义的 CICS 服务器数	配置文件中定义的 CICS 服务器数。
CICS 服务器:丢失连接数	与 CICS 服务器建立的连接丢失的次数。
CICS 服务器:协商的 IPIC 会话数	与 CICS 服务器协商的 IPIC 会话数。
CICS 服务器:孤立的 CICS 请求数	等待响应、已超时或已结束的 CICS 服务器请求 请求数。
CICS 服务器:已处理请 求数	已处理的 CICS 服务器请求数。
CICS 服务器:等待响应的 请求数	当前等待 CICS 服务器响应的请求数。
CICS 服务器:终端安装请 求数	发送给 CICS 服务器的终端安装数。
CICS 服务器:终端卸载事 件数	由网关处理的终端卸载事件的数目。
CICS 服务器:超时连接数	与 CICS 服务器的连接超时的次数。
CICS 后台进程:客户后 台进程运行时间	自客户端后台进程成功初始化以来的时间长度 (秒)。
CICS 后台进程:已连接的 客户端应用程序数	当前连接到客户端后台进程的客户端应用程 序进程数。
CICS 后台进程:已处理请 求数	已处理的 API 调用请求数。
连接管理器:池计数	并发分配给客户端的连接管理器线程的峰值数 目。

详细的度量标准名称	说明
CICS 服务器:CICS 请求数 据量	发送给所连接的 CICS 服务器的请求数据量（字节）。此数量包括应用程序和 CICS 协议数据。
连接管理器:线程分发计 数	连接管理器线程的分配数，代表已从客户端建立的连接数。
连接管理器:分配给客户 端的线程数	当前分配给客户端的连接管理器线程数。
网关后台进程:客户端请 求数据量	从客户端应用程序接收的请求数据量（字节）。
网关后台进程:客户端响 应数据量	发送到客户端应用程序的响应数据量（字节）。
网关后台进程:平均网关 后台进程响应时间	网关后台进程响应客户端 API 请求的平均时间（毫秒）。
网关后台进程:时间间隔 结束时间 HHMMSS	下一个排定时间间隔的统计信息重置事件的当地时间。此时，所有时间间隔统计信息将重置为零。
网关后台进程:已提交的 扩展 LUW 事务数	已提交的基于扩展 LUW 的 1 阶段事务的数目。
网关后台进程:已回滚的 扩展 LUW 事务	已回滚的基于扩展 LUW 的 1 阶段事务的数目。
网关后台进程:失败的 SYNCONRETURN 事务	当前时间间隔内失败的 SYNCONRETURN 事务的数目。
网关后台进程:网关后台 进程运行时间	自网关后台进程成功初始化以来的时间长度（秒）。
网关后台进程:时间间隔 运行时间	自上次时间间隔重置事件以来的时间长度（秒）（当前时间间隔的已过时间）。
网关后台进程:统计时间 间隔的长度 HHMMSS	网关后台进程使用的统计时间间隔的持续时间。
网关后台进程:一天的逻 辑结束时间 HHMMSS	网关后台进程指定为一天的逻辑结束时间的当地时间。此时，所有时间间隔统计信息将重置为零。
网关后台进程:CICS Request Exit 调用数	由网关后台进程调用的 Request Exit 调用数。
网关后台进程:扩展 LUW 事务数	在最新时间间隔内处理的扩展 LUW 事务数。

详细的度量标准名称	说明
CICS 服务器:CICS 请求数 据量	发送给所连接的 CICS 服务器的请求数据量（字节）。此数量包括应用程序和 CICS 协议数据。
网关后台进 程:SYNCONRETURN 事 务数	在最新时间间隔内处理的 SYNCONRETURN 事 务数。
网关后台进程:已处理请 求数	在最新时间间隔内处理的请求数。
网关后台进程:成功的 SYNCONRETURN 事 务	在网关后台进程期间成功的 SYNCONRETURN 事 务的数目。
网关后台进程:已处理请 求总数/秒	已处理的 API 调用的合计数。
协议处理程序:SSL 协议 处理程序端口号	用于 SSL 通信的 TCP 端口。
协议处理程序:TCP 协议 处理程序端口号	用于非 SSL 通信的 TCP 端口。
系统环境:JVM GC 计数	垃圾回收事件数。
系统环境:JVM GC 时间	JVM 进行垃圾回收的时间（毫秒）。
系统环境:GC 后的 JVM 堆大小	最近一次垃圾回收后的 JVM 堆大小。
系统环境:JVM 初始堆大 小	JVM 堆的初始大小。
系统环境:JVM 最大堆大 小	JVM 堆的最大大小。
工作线程:当前工作线程 数	当前已创建工作线程数。
工作线程:当前已分配的 工作线程数	当前由连接管理器使用的工作线程数。
工作线程:初始工作线程 数	网关后台进程创建工作线程的初始配置数目。
工作线程:最大工作线程 数	网关可处理的平行工作线程的最大配置数目。
工作线程:工作线程超时 次数	网关后台进程无法在配置的超时时间内将工作 线程分配给连接管理器的次数。
工作线程:已分配工作线 程的峰值数目	并发分配给连接管理器线程的工作线程的峰值 数目。

