

# CA Application Performance Management

for IBM CICS Transaction Gateway 안내서

릴리스 9.5



포함된 도움말 시스템 및 전자적으로 배포된 매체를 포함하는 이 문서(이하 "문서")는 정보 제공의 목적으로만 제공되며 CA 에 의해 언제든지 변경 또는 취소될 수 있습니다.

CA 의 사전 서면 동의 없이 본건 문서의 전체 혹은 일부를 복사, 전송, 재생, 공개, 수정 또는 복제할 수 없습니다. 이 문서는 CA 의 기밀 및 독점 정보이며, 귀하는 이 문서를 공개하거나 다음에 의해 허용된 경우를 제외한 다른 용도로 사용할 수 없습니다: (i) 귀하가 이 문서와 관련된 CA 소프트웨어를 사용함에 있어 귀하와 CA 사이에 별도 동의가 있는 경우, 또는 (ii) 귀하와 CA 사이에 별도 기밀 유지 동의가 있는 경우.

상기 사항에도 불구하고, 본건 문서에 기술된 라이선스가 있는 사용자는 귀하 및 귀하 직원들의 해당 소프트웨어와 관련된 내부적인 사용을 위해 합당한 수의 문서 복사본을 인쇄 또는 제작할 수 있습니다. 단, 이 경우 각 복사본에는 전체 CA 저작권 정보와 범례가 첨부되어야 합니다.

본건 문서의 사본 인쇄 또는 제작 권한은 해당 소프트웨어의 라이선스가 전체 효력을 가지고 유효한 상태를 유지하는 기간으로 제한됩니다. 어떤 사유로 인해 라이선스가 종료되는 경우, 귀하는 서면으로 문서의 전체 또는 일부 복사본이 CA 에 반환되거나 파괴되었음을 입증할 책임이 있습니다.

CA 는 관련법의 허용 범위 내에서, 상품성에 대한 묵시적 보증, 특정 목적에 대한 적합성 또는 권리 위반 보호를 비롯하여(이에 제한되지 않음) 어떤 종류의 보증 없이 본 문서를 "있는 그대로" 제공합니다. CA 는 본 시스템의 사용으로 인해 발생하는 직, 간접 손실이나 손해(수익의 손실, 사업 중단, 영업권 또는 데이터 손실 포함)에 대해서는 (상기 손실이나 손해에 대해 사전에 명시적으로 통지를 받은 경우라 하더라도) 귀하나 제 3 자에게 책임을 지지 않습니다.

본건 문서에 언급된 모든 소프트웨어 제품의 사용 조건은 해당 라이선스 계약을 따르며 어떠한 경우에도 이 문서에서 언급된 조건에 의해 라이선스 계약이 수정되지 않습니다.

본 문서는 CA 에서 제작되었습니다.

본 시스템은 "제한적 권리"와 함께 제공됩니다. 미합중국 정부에 의한 사용, 복제 또는 공개는 연방조달규정(FAR) 제 12.212 조, 제 52.227-14 조, 제 52.227-19(c)(1)호 - 제(2)호 및 국방연방구매규정(DFARS) 제 252.227-7014(b)(3)호 또는 해당하는 경우 후속 조항에 명시된 제한 사항을 따릅니다.

Copyright © 2013 CA. All rights reserved. 본 시스템에서 언급된 모든 상표, 상호, 서비스 표시 및 로고는 각 해당 회사의 소유입니다.

## CA Technologies 제품 참조

이 문서에서는 다음과 같은 CA Technologies 제품과 기능을 참조합니다.

- 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)

## CA 에 문의

### 기술 지원팀에 문의

온라인 기술 지원 및 지사 목록, 기본 서비스 시간, 전화 번호에 대해서는 <http://www.ca.com/worldwide>에서 기술 지원팀에 문의하십시오.



# 목차

---

<b>제 1 장: CTG 용 확장 개요</b>	<b>11</b>
CTG 용 확장 정보 .....	11
CTG 용 확장의 시스템 요구 사항 .....	12
CTG 용 확장의 구성 요소 .....	12
CTG 용 확장 및 사용자의 CA Introscope® 환경 .....	14
ProbeBuilder 지시문 배포 .....	15
요청 종료 개요 .....	18
분산 및 z/OS 혼합 CTG .....	20
<b>제 2 장: CTG 용 확장 설치</b>	<b>21</b>
설치 준비 .....	21
설치 아카이브 추출 .....	23
PBD 용 관리 모듈 및 Typeview 설치 .....	23
요청 종료용 관리 모듈 및 Typeview 설치 .....	24
CTG 서버를 모니터링할 Introscope Agent 파일 설치 .....	24
AutoProbe 를 사용하여 CTG 코드가 사용되도록 설정 .....	25
요청 종료를 사용하여 CTG 코드가 사용되도록 설정 .....	25
분산 CTG 플랫폼에서 수동 ProbeBuilder 를 사용하여 CTG 코드가 사용되도록 설정 .....	25
CTG 서버 코드 사용 준비 .....	26
Windows 의 CTG 서버 RequestExit 계측 지정 .....	26
CTG 서버의 수동 ProbeBuilding 옵션 입력 .....	27
CTG 서버를 위해 ProbeBuilder 마법사 사용 .....	28
CTG 서버를 위한 명령줄 ProbeBuilder 사용 .....	28
WebSphere 에서 CTG 클라이언트 코드가 사용되도록 설정 .....	28
z/OS CTG 시작 스크립트 수정 .....	29
분산 시스템을 위한 계측된 CTG 시작 스크립트 만들기 .....	30
계측된 코드 실행 .....	31
원래 코드로 되돌리기 .....	32
분산 CTG 플랫폼에서 ChangeDetector 가 사용되도록 설정 .....	32
CTG 서버 RequestExit 및 전체 통계 프로빙 구성(선택 사항) .....	35
구성 파일 ctg.ini 에서 z/OS 의 CTG 서버 RequestExit 및 전체 통계 프로빙 구성(선택 사항) .....	35
Windows 플랫폼에서 RequestExit 세부 정보 및 전체 통계를 개별적으로 가져오도록 구성(선택 사항) .....	36
CTG 클라이언트 지원 구성 .....	37

독립 실행형 CTG 클라이언트 응용 프로그램 구성 .....	38
채널 및 컨테이너 메트릭 구성 .....	38
WebSphere 에서 클라이언트 응용 프로그램 구성 .....	38
CTG 용 확장 업그레이드 .....	39
설치 및 구성 확인 .....	39

### 제 3 장: CTG 용 확장 사용 41

Introscope 에서 CTG 용 확장 데이터 보기 .....	41
트리 뷰에서 메트릭 데이터 보기 .....	42
프런트엔드 .....	42
백엔드 .....	43
특정 메트릭의 기록 데이터 분석 .....	44
서버 메트릭 .....	44
백엔드   CTG_Global_Statistics .....	44
백엔드   CTG_to_CICS_ECI_IPIC 노드 .....	46
백엔드   CTG_to_CICS_EPI 노드 .....	46
백엔드   JSSE - CTG 노드 .....	47
프런트엔드   Client_to_CTG_Aggregates 노드 .....	48
프런트엔드   Client_to_CTG_Details 노드 .....	50
프런트엔드   Client_to_CTG_Details 트랜잭션 뷰 노드 .....	51
프런트엔드   Client_to_CTG_JSSE 세션 노드 .....	52
Introscope Investigator 탭 뷰 사용 .....	52
대시보드에서 CTG 용 확장 데이터 보기 .....	53
CTG 서버 - 개요 대시보드 .....	54
CTG 요약 - 서버 ECI 동작 대시보드 .....	55
CTG 서버 - 전체 통계 대시보드 .....	56
CTG 서버 - EPI 요청 대시보드 .....	56
CTG 서버 - EPI 요청 대시보드 .....	57
CTG 서버 - 연결 관리자 및 작업자 대시보드 .....	58
CTG Server - SSL(CTG 서버 - SSL) 대시보드 .....	59
CTG 클라이언트 - 개요 대시보드 .....	59
CTG 클라이언트 - Java 게이트웨이 및 SSL 세션 대시보드 .....	60
CTG 클라이언트 - EPI 요청 대시보드 .....	60
CTG 클라이언트 - EPI 터미널 요청 대시보드 .....	61
Introscope ChangeDetector 대시보드 .....	61
Introscope 주의/위험 경고 임계값 수정 .....	62
CTG 트랜잭션 추적 프로그램 .....	62
PP CTG Transaction Tracer(PP CTG 트랜잭션 추적 프로그램)의 특수 문자 처리 .....	62
CTG 트랜잭션 추적 프로그램 속성의 자세한 목록 .....	63

---

## 부록 A: CTG 성능 메트릭

67

프런트엔드 메트릭 .....	67
프런트엔드 클라이언트 - CTG 집계 .....	69
프런트엔드 클라이언트 - CTG 세부 정보 .....	70
백엔드 메트릭 .....	70
백엔드 CTG_to_CICS_ECI_IPIC 집계 .....	72
백엔드 CTG 전체 통계 .....	72
백엔드 CTG_to_CICS_EPI 집계 메트릭 .....	73
백엔드 CTG_to_CICS 스레드 .....	73
CTG 대시보드 메트릭 .....	73
CTG 클라이언트 - 개요 대시보드 .....	74
CTG 클라이언트 - EPI 대시보드 .....	74
CTG 클라이언트 - Java 게이트웨이 및 SSL 대시보드 .....	75
CTG 클라이언트 - 터미널 및 터미널 요청 대시보드 .....	76
CTG 서버 - 개요 대시보드 .....	76
CTG 서버 - 연결 관리자 및 작업자 대시보드 .....	76
CTG 서버 - EPI 요청 대시보드 .....	77
CTG 서버 - EPI 요청 대시보드 .....	78
CTG 서버 - 전체 통계 대시보드 .....	78
CTG 서버 - SSL 대시보드 .....	79
요청 종료 메트릭 .....	79
백엔드   CTG_to_CTG_ECI_IPIC_RequestExit 메트릭 .....	80
CTG_Global_Statistics_RequestExit 메트릭 .....	81



# 제 1 장: CTG 용 확장 개요

---

IBM® CICS® Transaction Gateway 용 확장을 사용하면 CA Introscope 관리자가 CTG 클라이언트 인터페이스를 모니터링할 수 있습니다. 이러한 인터페이스는 Java, J2EE 및 CTG 서버의 응용 프로그램에서 사용됩니다.

이 섹션은 다음 항목을 포함하고 있습니다.

[CTG 용 확장 정보 \(페이지 11\)](#)

[CTG 용 확장 및 사용자의 CA Introscope® 환경 \(페이지 14\)](#)

[분산 및 z/OS 혼합 CTG \(페이지 20\)](#)

## CTG 용 확장 정보

CTG 확장은 CTG(CICS Transaction Gateway) 서버 제품과 CTG 클라이언트 Java 에 대한 실시간 모니터링 기능을 제공합니다. CTG 확장은 CTG 를 통해 CICS(Customer Information Control System)를 호출하는 WebSphere CTG 클라이언트 응용 프로그램도 모니터링합니다. 이 확장은 CTG 서버 자체와 CTG 서버를 호출하는 계층되는 클라이언트를 모두 모니터링합니다. CTG 서버는 Java 및 J2EE 프론트엔드와 CICS Transaction Server 백엔드 사이에서 중개자 역할을 합니다. 클라이언트에 대한 모니터링에는 JCA 및 기본 API 클라이언트 응용 프로그램 모두에 대한 모니터링이 포함됩니다.

**참고:** CTG 서버를 CTG 데몬이라고도 합니다. 서버라고 표현하는 쪽이 더 이해가 쉽기 때문에 이 안내서에서는 CTG 서버라는 표현을 사용하지만, 다른 기술 문서에서는 CTG 데몬이라고 나올 수도 있습니다.

CTG 용 확장을 사용하면 Introscope 사용자가 Java 응용 프로그램, WebSphere Application Server 및 CTG 자체에서 발생할 수 있는 병목 지점을 상호 연관시키고 격리할 수 있습니다. CTG 용 확장은 트랜잭션을 모니터링하여 Java 또는 J2EE 프론트엔드, CTG 서버 미들웨어 또는 CICS 백엔드에 병목 지점이 존재하는지 여부를 결정할 수 있는 상세한 메트릭을 제공합니다.

CTG 용 확장이 설치되어 있으면 다음을 볼 수 있습니다.

- CTG 를 사용하는 응용 프로그램의 성능 및 동작 그래프. 응용 프로그램에서 CTG 호출에 CTG 클라이언트 기본 클래스, CCF 인터페이스 또는 JCA 인터페이스를 사용하는지에 관계없이 이러한 항목을 사용할 수 있습니다.
- 기본 제공되는 대시보드를 통해 표시되는 CTG 의 건전성에 대한 간략한 개요
- 성능의 계층적 뷰와 기록 뷰

### CTG 용 확장의 시스템 요구 사항

CTG 시스템 요구 사항의 전체 목록은 CA APM 북셀프의 "Application Performance Management Compatibility Guide"(Application Performance Management 호환성 안내서)를 참조하십시오. 제품 호환성 매트릭스에는 지원되는 모든 운영 환경의 목록이 나와 있습니다.

참고: CTG 용 확장의 일본어 버전은 Introscope 9.0.5 에서만 작동합니다.

### CTG 용 확장의 구성 요소

다음과 같은 CTG 용 확장 구성 요소를 통해 Introscope 와 상호 작용할 수 있습니다.

구성 요소	설명
<i>PPCTGServer_ManagementModule.jar</i> <i>PPCTGClient_ManagementModule.jar</i>	CTG 용 확장 관리 모듈은 메트릭에 적용되는 모니터링 및 보고 논리를 정의합니다. 이러한 메트릭에는 Workstation 에 표시되는 대시보드와 메트릭 임계값을 정의하는 경고가 포함됩니다.
<i>PPCTGRequestExit_ManagementModule.jar</i>	CTG 용 확장에서는 요청 종료 계측에 이 모듈을 사용합니다.
<i>PPCTGClient-full.pbd</i> <i>PPCTGClient-typical.pbd</i>	Introscope 지원 CTG 구성 요소가 에이전트에 보고하는 메트릭을 제어하는 ProbeBuilder 지시문 파일입니다. PPCTGClient-full.pbd 파일을 사용하면 Introscope 는 모든 메트릭을 표시합니다. 모든 메트릭을 표시함으로써 인해 발생하는 오버헤드를 방지하려면 PPCTGClient-typical.pbd 파일을 사용하십시오.

<p><i>PPCTGServer-full.pbd</i>  <i>PPCTGServer-minimal.pbd</i>  <i>PPCTGServer-typical.pbd</i></p>	<p>Introscope 지원 CTG 구성 요소가 에이전트에 보고하는 메트릭을 제어하는 ProbeBuilder 지시문 파일입니다.</p> <p>PPCTGServer-full.pbd 파일을 사용하면 Introscope 는 모든 메트릭을 표시합니다. 모든 메트릭을 표시함으로 인해 발생하는 오버헤드를 방지하려면 PPCTGServer-minimal.pbd 또는 PPCTGServer-typical.pbd 파일을 사용하십시오.</p>
<p><i>PPCTGTranTrace.pbd</i></p>	<p>PPCTGTranTrace.pbd 지시문 파일은 CTG 트랜잭션을 추적하고 프런트엔드와 백엔드 간에 트랜잭션을 상호 관련시키는 기능을 제공합니다.</p>
<p><i>PPCTGAgent.jar</i></p>	<p>에이전트가 CTG 서버 메트릭을 Enterprise Manager 에 보고할 수 있도록 하는 확장입니다. CTG 버전 6.x 및 7.x 모두에 적용됩니다.</p>
<p><i>PPCTGRequestExit.jar</i>  <i>PPCTGbmonitor.jar</i></p>	<p>에이전트가 CTG 8.x 서버 메트릭을 Enterprise Manager 에 보고할 수 있도록 하는 확장입니다. 이 확장을 사용하려면 PPrunGlobalStats 배치 작업을 시작해야 합니다. CTG 버전 7.x 및 8.x 에 적용됩니다.</p>
<p><i>ctg.typeviewers.xml</i></p>	<p>트리에서 CTG 구성 요소를 선택한 경우 Investigator 의 탭에서 선택할 수 있는 CTG 관련 뷰를 정의하는 Enterprise Manager 확장입니다.</p>
<p><i>ctg.requestexit.typeviewers.xml</i></p>	<p>CTG-Request 확장 관련 뷰를 정의하는 Enterprise Manager 확장입니다. 트리에서 CTG 요청 종료 구성 요소를 선택한 경우 이러한 뷰를 Investigator 의 탭에서 선택할 수 있습니다.</p>
<p><i>ChangeDetector-config.xml</i></p>	<p>CTG 를 위한 기본 ChangeDetector 모니터링 구성 항목이 들어 있는 XML 파일입니다.</p>
<p><i>CTG_Trان_Trace_Template.profile</i></p>	<p>이 파일에는 CTG 추적 관련 메트릭을 수집하기 위해 IntroscopeAgent.profile 에 포함될 수 있는 CTG 트랜잭션 추적 속성이 들어 있습니다. 자세한 내용은 <a href="#">CTG 트랜잭션 추적 프로그램</a> (페이지 62)을 참조하십시오.</p>

*PPrunGlobalStats.bat*

*PPrunGlobalStats.sh*

이 필수 구성 요소는 Windows 플랫폼을 위한 것입니다. PPrunGlobalStats.bat 파일 내에서 CTG\_CLASSES 및 ISCOPE\_AGENT 변수를 지정하고 PPCTGbmonitor.jar 에 포함된 전체 통계 모니터를 실행할 수 있습니다. Windows 에서는 CTG 가 Windows 의 서비스로 실행되므로 이 파일이 필요합니다. Windows 에서 실행하는 경우 전체 통계 정보를 추출하기 위해 별도의 프로세스를 사용해야 합니다. Unix 플랫폼에도 동등한 파일이 존재하며, PPrunGlobalStatus.sh 는 위 .bat 파일의 Unix/Linux 버전입니다.

---

## CTG 용 확장 및 사용자의 CA Introscope® 환경

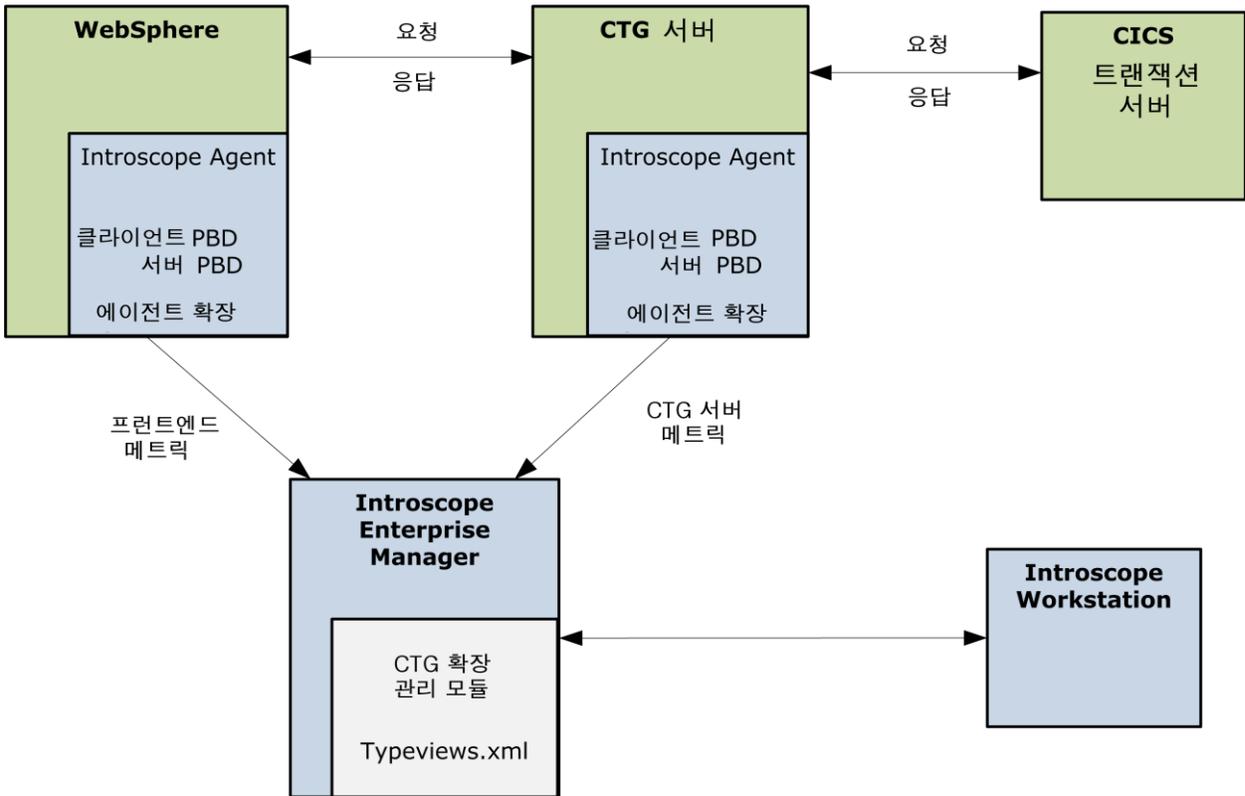
환경에 따라, CTG 용 확장을 배포하려면 하나 이상의 컴퓨터가 필요할 수 있습니다. 배포는 PBD 또는 요청 종료를 사용하여 수행됩니다. 배포에 포함된 CA Introscope 구성 요소 ProbeBuilder 지시문은 다음과 같습니다.

- CTG 클라이언트 라이브러리가 사용되도록 설정하는 데 사용되는 ProbeBuilder 지시문. 응용 프로그램을 호스팅하는 컴퓨터와 CTG 서버를 호스팅하는 컴퓨터 모두에 배포됩니다.
- CTG 서버 라이브러리가 사용되도록 설정하는 데 사용되는 CTG 용 확장의 에이전트 확장 및 ProbeBuilder 지시문. CTG 서버를 호스팅하는 컴퓨터와 응용 프로그램을 호스팅하는 컴퓨터 모두에 배포됩니다.
- 관리 모듈 Enterprise Manager(대개 다른 컴퓨터에 있음)에 배포됩니다.

**중요!** CA Technologies 에서는 ProbeBuilder 와 요청 종료를 동시에 사용하여 CTG 가 사용되도록 설정하는 것을 권장하지 않습니다. 이렇게 하면 메트릭이 중복되고 CPU 사용률이 높아질 수 있습니다.

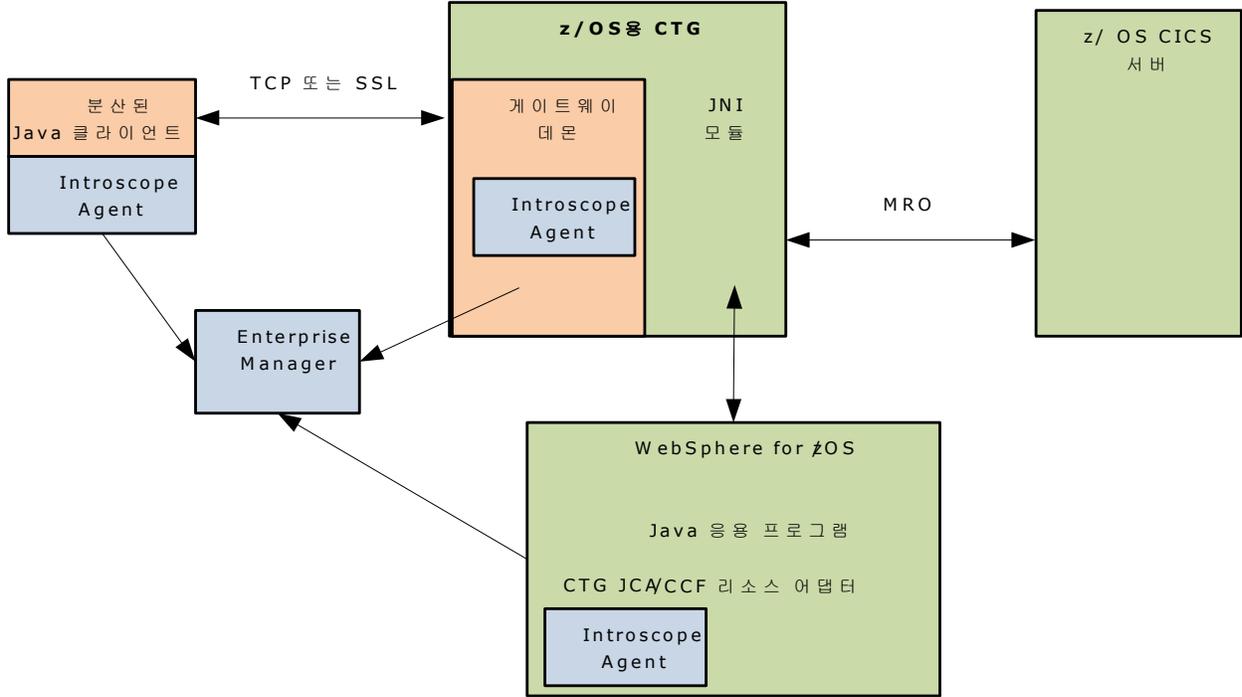
## ProbeBuilder 지시문 배포

다음 그림에서는 ProbeBuilder 지시문을 배포에 사용할 때의 전반적인 상호 작용을 보여 줍니다. CTG 용 확장은 z/OS 환경과 분산(Windows/Linux/UNIX) 환경 모두에서 작동합니다.



### z/OS 의 CTG 구성 요소

다음 그림에서는 z/OS 에 있는 CTG 구성 요소 간의 상호 작용을 보여 줍니다.



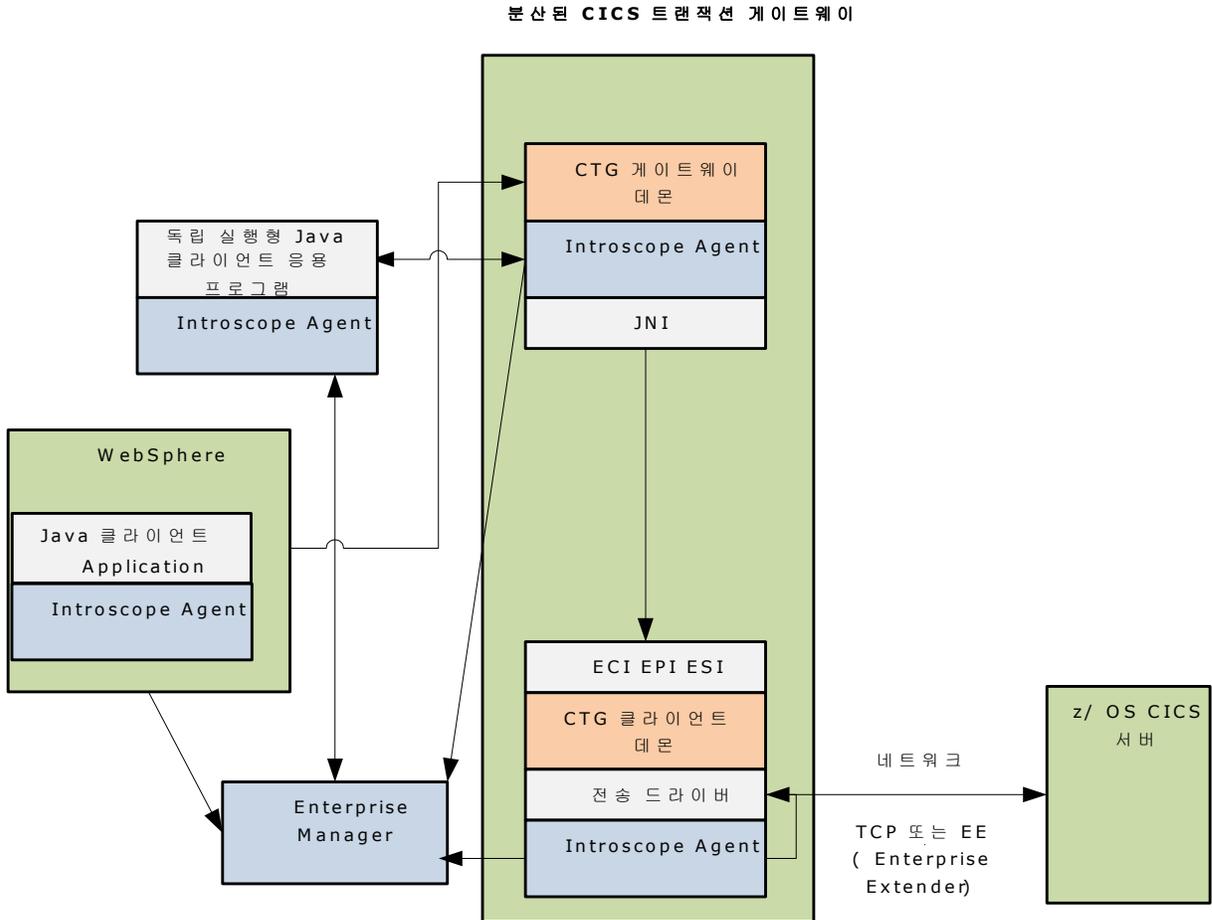
z/OS 용 CTG 는 다음과 같은 두 가지 기본 런타임 구성 요소로 구성되어 있습니다.

- 들어오는 작업을 수신 대기하고 IBM EXCI 기능을 사용하여 해당 작업을 로컬 CICS 백엔드 서버로 전달하는 CTG 게이트웨이 데몬
- WebSphere 런타임 환경에 배포되는 JCA(또는 CCF) 리소스 어댑터

CTG 용 확장은 이러한 구성 요소 두 개를 모두 모니터링합니다. Introscope Investigator 에서 모든 클라이언트 메트릭은 Frontends|Client\_to\_CTG\_Aggregates 및 Frontends|Client\_to\_CTG\_Details 노드 아래에 나타납니다. 모든 서버 메트릭은 백엔드|CTG\_to\_CICS\_xxx 및 백엔드|JSSE\_to\_CTG 노드 아래에 나타납니다.

### 분산 시스템의 CTG 구성 요소

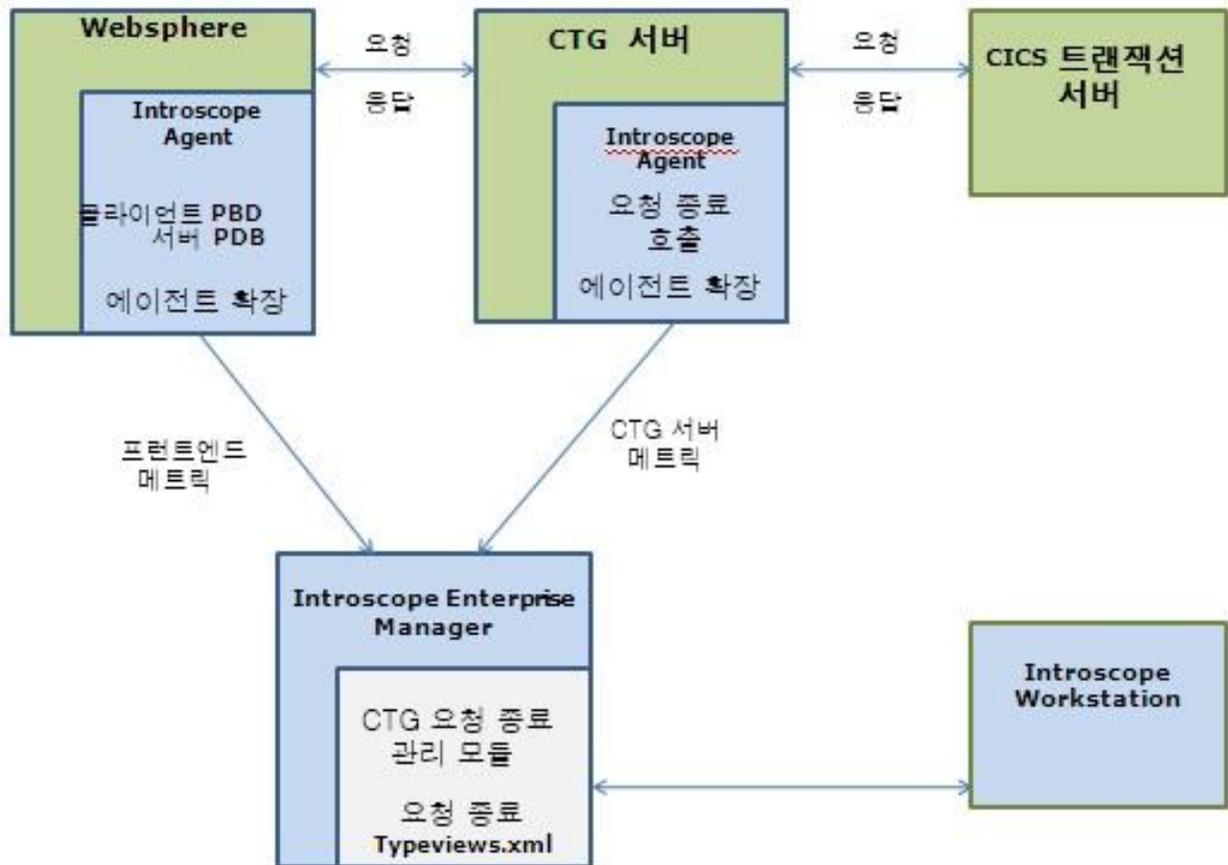
다음 그림에서는 분산 시스템에 있는 CTG 구성 요소 간의 상호 작용을 보여 줍니다.



## 요청 종료 개요

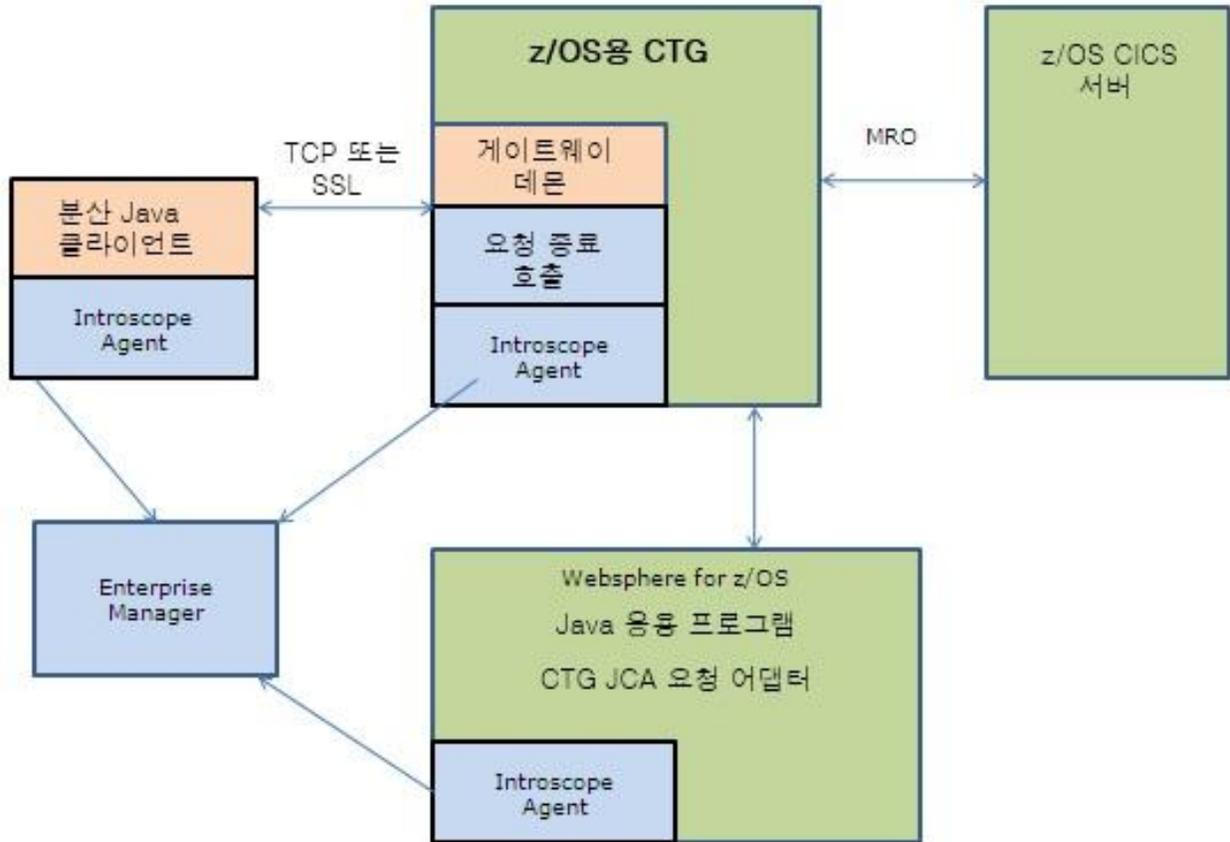
IBM 이 제공하는 요청 종료 지점을 사용하면 타사 플러그 인이 CTG 로 처리되는 요청에 대한 통계를 추출할 수 있습니다. 종료는 ECI 및 IPIC 트랜잭션만 지원됩니다. EPI 는 지원되지 않습니다.

요청 종료 배포에서는 다음과 같은 구성 요소와 상호 작용을 사용합니다.



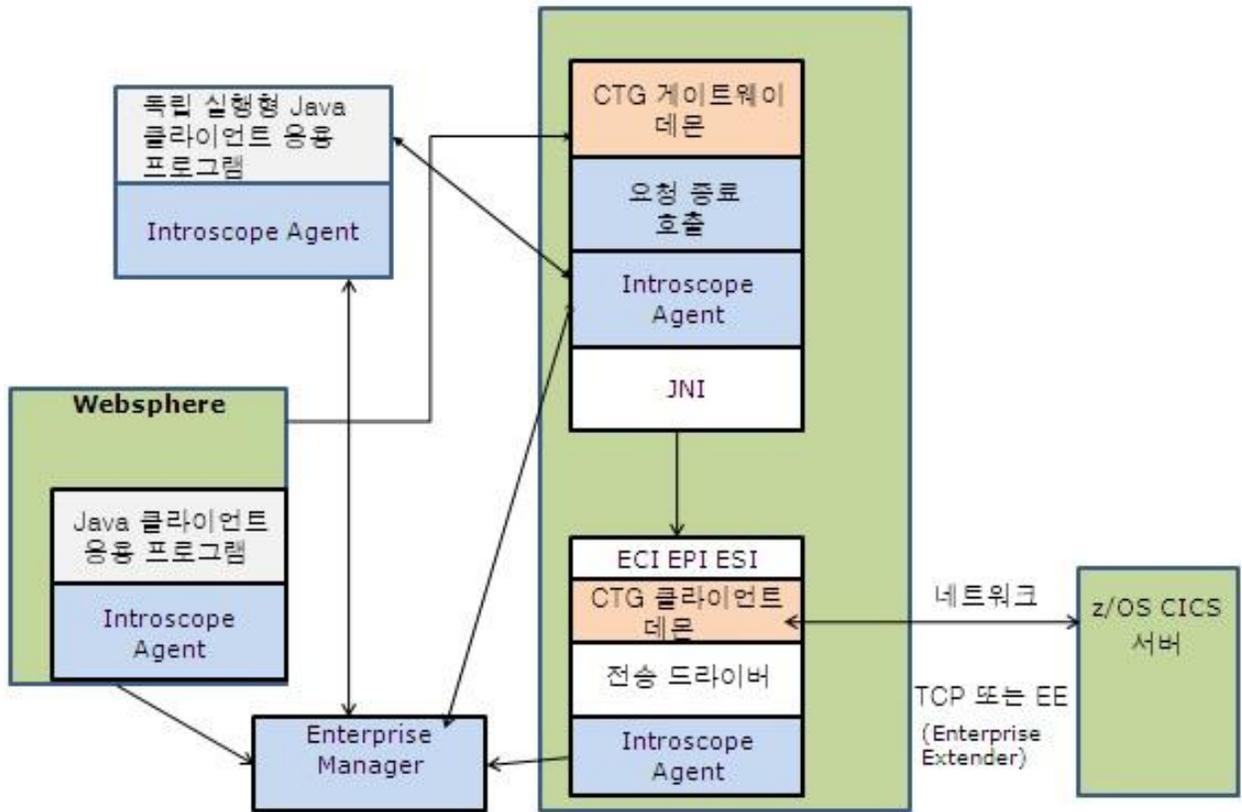
요청 종료와 CTG 구성 요소

다음 그림에서는 z/OS 에 있는 CTG 요청 종료 구성 요소 간의 상호 작용을 보여 줍니다.



### 분산 시스템의 요청 종료 구성 요소

다음 그림에서는 분산 시스템에 있는 CTG 요청 종료 구성 요소 간의 상호 작용을 보여 줍니다.



### 분산 및 z/OS 혼합 CTG

CTG 용 확장을 분산 및 z/OS 가 혼합된 환경에 배포할 수도 있습니다. 예를 들어 z/OS 에서 실행되는 CTG 서버와 통신하는 CTG 리소스 어댑터를 사용하여 분산 WebSphere 인스턴스가 실행되고 있을 수 있습니다.

이 확장은 두 환경을 모두 모니터링할 수 있습니다. 따라서 단일 도구(Introscope)와 단일 확장 제품군(CTG)으로 여러 환경에서 필요한 측정 데이터를 통합할 수 있습니다.

**참고:** WebSphere 는 CICS 와 직접 상호 작용하지 않습니다. 모든 통신은 CTG 를 통해 이루어집니다.

## 제 2 장: CTG 용 확장 설치

---

이 단원에서는 CTG 용 확장을 설치 및 구성하는 방법과 Introscope Agent 프로파일 파일에서 CTG 속성을 정의하는 방법을 설명합니다.

이 섹션은 다음 항목을 포함하고 있습니다.

[설치 준비](#) (페이지 21)

[설치 아카이브 추출](#) (페이지 23)

[PBD 용 관리 모듈 및 Typeview 설치](#) (페이지 23)

[요청 종료용 관리 모듈 및 Typeview 설치](#) (페이지 24)

[CTG 서버를 모니터링할 Introscope Agent 파일 설치](#) (페이지 24)

[AutoProbe 를 사용하여 CTG 코드가 사용되도록 설정](#) (페이지 25)

[요청 종료를 사용하여 CTG 코드가 사용되도록 설정](#) (페이지 25)

[분산 CTG 플랫폼에서 수동 ProbeBuilder 를 사용하여 CTG 코드가 사용되도록 설정](#) (페이지 25)

[WebSphere 에서 CTG 클라이언트 코드가 사용되도록 설정](#) (페이지 28)

[z/OS CTG 시작 스크립트 수정](#) (페이지 29)

[분산 시스템을 위한 계측된 CTG 시작 스크립트 만들기](#) (페이지 30)

[계측된 코드 실행](#) (페이지 31)

[원래 코드로 되돌리기](#) (페이지 32)

[분산 CTG 플랫폼에서 ChangeDetector 가 사용되도록 설정](#) (페이지 32)

[CTG 서버 RequestExit 및 전체 통계 프로빙 구성\(선택 사항\)](#) (페이지 35)

[CTG 클라이언트 지원 구성](#) (페이지 37)

[설치 및 구성 확인](#) (페이지 39)

### 설치 준비

설치를 준비하려면 다음 단계가 완료되었는지 확인하십시오.

다음 단계를 수행하십시오.

1. 시스템이 [요구 사항](#) (페이지 12)을 충족하는지 확인합니다.

**참고:** 아직 Introscope 를 설치하지 않은 경우 *CA APM 설치 및 업그레이드 안내서*의 지침을 따르십시오.

2. Introscope 환경에서 다음 디렉터리 위치를 확인합니다.

- *Introscope Enterprise Manager 디렉터리(<EM\_Home>)* - Enterprise Manager 의 설치 디렉터리
- *Introscope Agent 홈 디렉터리(<Agent\_Home>)* - CTG 서버를 모니터링하는 Introscope Agent 의 설치 디렉터리. CTG 서버 데이터를 수집하는 각 에이전트에 대해 이 디렉터리를 확인합니다.
- *Introscope Agent 프로파일 디렉터리* - CTG 용 확장을 설치할 예정인 각 에이전트에서 *IntroscopeAgent.profile* 이 위치한 디렉터리

**참고:** *com.wily.introscope.agentProfile* 의 시스템 속성 디렉터리를 식별하십시오. 에이전트 프로파일은 대개 에이전트 설치의 *wily\core\config* 디렉터리에 있습니다.

3. CTG 용 확장을 설치할 예정인 Enterprise Manager 를 중지합니다.

클러스터 환경인 경우:

1. MOM(Manager of Managers) 역할을 하는 Enterprise Manager 를 중지합니다.
  2. CTG 용 확장 지원 에이전트에 연결되는 각 수집기 Enterprise Manager 를 중지합니다.
4. CTG 용 확장을 설치할 예정인 구성 요소를 모두 중지합니다. 이러한 구성 요소는 다음과 같을 수 있습니다.
- Java 응용 프로그램
  - WebSphere Application Server
  - CTG 서버 소프트웨어

**참고:** CTG 용 확장을 설치하기 위해 한 번에 여러 개의 구성 요소를 중지할 필요가 없습니다. 구성 요소 하나를 중지하고 관련 설치를 수행한 후 다른 구성 요소 설치를 진행하십시오. 예를 들어 Enterprise Manager 를 중지하고 CTG 용 확장을 설치한 다음 각 Introscope Agent 를 중지하고 관련 설치를 수행할 수 있습니다. 클러스터 환경인 경우 MOM(Manager of Managers)을 중지하고 CTG 용 확장을 설치하고, 각 수집기 Enterprise Manager 를 중지하고 관련 설치를 수행한 다음, 각 에이전트를 중지하고 관련 설치를 수행하십시오.

## 설치 아카이브 추출

시스템에 적합한 CTG 용 확장 아카이브를 추출하십시오.

CTG 용 확장 관리 모듈 및 서버 확장은 Introscope 에서 CTG 사용량과 성능을 모니터링할 수 있도록 Enterprise Manager 를 확장합니다.

**중요!** z/OS 플랫폼을 사용하는 경우 FTP 바이너리 모드를 사용하여 전체 아카이브를 전송해야 합니다. 그렇지 않으면 ASCII 변환 시 파일이 손상됩니다.

## PBD 용 관리 모듈 및 Typeview 설치

다음 절차는 PBD 계측을 위한 관리 모듈 및 TypeViewer 설치에 적용됩니다.

다음 단계를 수행하십시오.

1. CTG 용 확장을 설치할 예정인 Enterprise Manager 를 중지합니다.

클러스터 환경인 경우:

1. MOM(Manager of Managers) 역할을 하는 Enterprise Manager 를 중지합니다.
2. 연결된 수집기 Enterprise Manager 를 각각 중지합니다.

2. 다음 파일을 <EM\_Home>/config/modules 에 복사합니다.

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

클러스터된 Introscope 환경에서 CTG 용 확장을 설치하는 경우 MOM(Manager of Managers)과, 이 MOM 에 연결된 모든 Enterprise Manager 에 CTG 용 확장을 설치합니다.

3. ctg.typeviewers.xml 을 <EM\_Home>/ext/xmltv 에 복사합니다.

클러스터된 Introscope 환경에서 CTG 용 확장을 설치하는 경우 MOM(Manager of Managers) 역할을 하는 Enterprise Manager 와 MOM(Manager of Managers)에 연결된 모든 수집기 Enterprise Manager 에 CTG 용 확장의 서버 확장을 설치합니다.

4. 추출된 설치 아카이브 파일 ctg.typeviewers.xml 을 <Introscope\_Home>/ext/xmltv 로 복사하여 확장을 설치합니다.

## 요청 종료용 관리 모듈 및 Typeview 설치

다음 절차는 요청 종료 계측을 위한 관리 모듈 및 TypeViewer 설치에 적용됩니다.

다음 단계를 수행하십시오.

1. CTG 용 확장을 설치할 예정인 Enterprise Manager 를 중지합니다.  
클러스터 환경인 경우:
  1. MOM(Manager of Managers) 역할을 하는 Enterprise Manager 를 중지합니다.
  2. 연결된 수집기 Enterprise Manager 를 각각 중지합니다.
2. PPCTGRequestExit\_ManagementModule.jar 를 <EM\_Home>/config/modules 에 복사합니다.  
클러스터된 Introscope 환경에서 CTG 용 확장을 설치하는 경우 MOM(Manager of Managers)과, 이 MOM 에 연결된 모든 Enterprise Manager 에 CTG 용 확장을 설치합니다.
3. ctg.requestexit.typeviewers.xml 을 <EM\_Home>/ext/xmltv 에 복사합니다.

## CTG 서버를 모니터링할 Introscope Agent 파일 설치

CTG 용 확장을 사용하면 CTG 서버 사용량과 성능을 모니터링할 수 있습니다. 다음은 CTG 서버를 모니터링할 에이전트 또는 요청 종료를 설치하는 방법에 대한 설명입니다.

CTG 서버를 모니터링할 에이전트를 설치하려면 다음 단계를 수행하십시오.

다음 단계를 수행하십시오.

1. 확장을 설치할 예정인 CTG 서버를 중지합니다.
2. <Agent\_Home>/examples/PowerPackForIBMCTG/ext 에서 <Agent\_Home>/wily/core/ext 로 PPCTGAgent.jar 를 복사합니다.

CTG 서버를 모니터링할 요청 종료를 설치하려면 다음 단계를 수행하십시오.

다음 단계를 수행하십시오.

1. 확장을 설치할 예정인 CTG 서버를 중지합니다.
2. <Agent\_Home>/examples/PowerPackForIBMCTG/ext 에서 CTG 의 classes 디렉터리로 PPCTGRequestExit.jar 를 복사합니다.
3. CTG 의 CLASSPATH 목록에 PPCTGRequestExit.jar 가 있는지 확인합니다.
4. CTG "구성" 메뉴에서 CTG 요청 종료를 이름을 구성합니다.

## AutoProbe 를 사용하여 CTG 코드가 사용되도록 설정

PBD 배포의 경우 AutoProbe 는 시작 시 자동으로 계측 프로브를 만듭니다.

## 요청 종료를 사용하여 CTG 코드가 사용되도록 설정

요청 종료를 사용하여 CTG 서버를 동적으로 프로빙할 수 있습니다. CTG 클라이언트 코드의 경우에는 JVM AutoProbe 를 대신 사용해야 합니다.

## 분산 CTG 플랫폼에서 수동 ProbeBuilder 를 사용하여 CTG 코드가 사용되도록 설정

수동 ProbeBuilder 는 Introscope 에서 CTG 게이트웨이 서버 클래스를 사용할 수 있도록 설정하는 방법 중에 동적이지 않은 방법입니다. 수동으로 ProbeBuilder 를 실행하면 CTG 서버가 실행되기 전에 Introscope 에서 디스크의 CTG Java 클래스를 사용할 수 있도록 설정됩니다.

CTG 계측에 대한 기본 설정 방법은 JVM AutoProbe 를 사용하거나 요청 종료를 사용하는 것입니다. 하지만 CTG 응용 프로그램을 수동으로 프로빙하려면 아래 단계를 따르십시오.

## CTG 서버 코드 사용 준비

이 단원의 지침에서는 다음 설치 및 구성 작업을 이미 수행했다고 가정합니다.

- Introscope Agent 설치
- 다음 ProbeBuilder 지시문 계측 완료
  - EM 폴더 아래의 custompbd 에 PPCTGClient-full.pbd 및 PPCTGServer-full.pbd 복사
  - examples 에서 에이전트 확장 폴더로 PPCTGAgent.jar 복사
  - PBD 파일 포함. 자세한 내용은 "WebSphere 에서 CTG 클라이언트 코드가 사용되도록 설정"( 페이지 참조)을 참조하십시오.
  - 에이전트 이름 구성
  - 필요한 모든 ProbeBuilder 옵션 구성
  - ErrorDetector 를 사용하는 경우 custompbd 에 errors.pbd 복사
- 요청 종료 계측 완료
  - 요청 종료 계측용 PPCTGRequestExit.jar 설치
  - examples 에서 에이전트 확장 폴더로 PPCTGAgent.jar 복사
  - PBD 파일 포함.
  - 에이전트 이름 구성
  - CTG 의 classes 폴더에 PPCTGRequestExit.jar 복사
  - CTG 클래스 경로 목록에 PPCTGRequestExit.jar 파일이 있는지 확인
  - Windows 의 경우 ctg.ini 파일 또는 CTG "구성" 메뉴에서 CTG 요청 종료를 이름 구성

## Windows 의 CTG 서버 RequestExit 계측 지정

Windows 버전의 CTG 는 Windows 서비스 형태로만 실행되기 때문에 ctgservice 명령을 사용해야 합니다. 이 명령은 서비스가 시작될 때 CTG 서버로 전달되는 Introscope 속성을 구성하는 데 사용됩니다. 이 명령의 일반적인 호출 방법은 다음과 같습니다.

```
ctgservice -R
-A-j-Dcom.wily.introscope.agentProfile=C:\<Agent_HOME>\wily\core\config\Introsco
peAgent.profile
-A-j-javaagent:C:\<Agent_Home>\wily\Agent.jar
```

## CTG 서버의 수동 ProbeBuilding 옵션 입력

CTG 서버 파일의 ProbeBuilding 을 하기 전에 다음을 수행하여 수동 ProbeBuilding 을 준비하십시오.

다음 단계를 수행하십시오.

1. 현재 CTG classes/\*.jar 파일을 백업 폴더에 저장합니다.
2. 그런 다음 CTG 의 classes 디렉터리에 있는 다음 jar 파일에 대해 ProbeBuilder 를 실행합니다.

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

이 작업은 CTG 의 \classes 디렉터리에 *ctgclient.isc.jar*, *ctgserver.isc.jar*, *cicsj2ee.isc.jar* 등과 같은 일련의 \*.isc.jar 파일을 생성합니다.

3. 다음 \*.isc.jar 파일의 이름을 변경합니다.

- 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. 프로빙되는 jar 파일의 .isc 버전이 다른 디렉터리에 생성된 경우 해당 파일을 CTG 의 classes 디렉터리에 직접 복사합니다.

5. jar 파일을 수동으로 계측합니다. 이 작업을 수행하는 두 가지 방법이 있습니다.

- ProbeBuilder 마법사 - ProbeBuilder 를 실행할 수 있는 GUI 대화 상자를 제공합니다.
- 명령줄 ProbeBuilder - 창 시스템이 없는 환경을 위한 ProbeBuilder 의 명령줄 인터페이스입니다.

**참고:** 바이트 코드 계측에 대한 자세한 내용은 *CA APM Java Agent 구현 안내서*를 참조하십시오.

## CTG 서버를 위해 ProbeBuilder 마법사 사용

컴퓨터에 Windows 환경이 있는 경우 GUI 기반 ProbeBuilder 마법사를 사용할 수 있습니다.

다음 단계를 수행하십시오.

1. CTG 확장 .pbd 파일이 <Agent\_Home> 디렉터리에 설치되어 있으며, ProbeBuilder 마법사가 CTG 확장 .pbd 파일에 액세스할 수 있는지 확인합니다.
2. *ctgserver.jar*, *ctgclient.jar* 및 *cicsj2ee.jar* 에 대해 ProbeBuilder 마법사를 실행합니다.
3. 안전을 위해서 시작하기 전에 원래 CTG jar 파일을 백업 영역에 복사합니다.

## CTG 서버를 위한 명령줄 ProbeBuilder 사용

컴퓨터에 창 환경이 없는 경우 명령줄 ProbeBuilder 를 사용하여 수동으로 CTG jar 파일을 프로빙할 수 있습니다.

다음 단계를 수행하십시오.

1. 안전을 위해서 ProbeBuilder 를 실행하기 전에 원래 CTG jar 파일을 백업 영역에 복사합니다.
2. Probe builder 명령 및 지시문을 사용합니다.
3. CTG 클래스 디렉터리에 있는 분산 CTG 파일을 프로빙하는 경우 다음 jar 파일을 수동으로 프로빙해야 합니다.
  - *ctgserver.jar*
  - *ctgclient.jar*
  - *cicsj2ee.jar*

**참고:** Probe builder 명령 및 지시문에 대한 자세한 내용은 *CA APM Java Agent 구현 안내서*를 참조하십시오.

## WebSphere 에서 CTG 클라이언트 코드가 사용되도록 설정

z/OS 의 WebSphere 에서 JVM AutoProbe 를 사용하여 CTG 클라이언트와 서버를 자동으로 모니터링할 수 있습니다.

다음 단계를 수행하십시오.

- *IntroscopeAgent.profile* 파일을 편집하여 속성 정의의 일부로 다음 파일을 포함하도록 *introscope.autoprobe.directivesFile* 속성을 수정합니다.
  - *PPCTGClient-full.pbd*
  - *PPCTGServer-full.pbd*
  - *required.pbd*

예:

```
introscope.autoprobe.directivesFile=<list of existing PBDs>,
PPCTGClient-full.pbd,PPCTGServer-full.pbd,required.pbd
```

**참고:** *introscope.autoprobe.directivesFile* 의 전체 정의가 한 줄에 포함되어야 합니다.

## z/OS CTG 시작 스크립트 수정

z/OS 환경에서 설치하는 경우 시작 시에 Introscope 지원 JVM AutoProbe 매개 변수를 CTG JVM 으로 전달하도록 CTG 의 *ctgstart* 스크립트를 수정합니다.

다음 단계를 수행하십시오.

1. 원래 *ctgstart* 스크립트의 백업 복사본을 만듭니다.
2. CTG 의 *ctgstart* 스크립트를 편집할 수 있도록 엽니다. *ctgstart* 스크립트는 CTG 의 bin 디렉터리에 있습니다.
3. *ctgstart* 스크립트의 아래쪽에 있는 다음 문 블록을 검색합니다.

```
jvmoptions=$(echo $aloptions | $awk '{
  for (i = 1; i<=NF; i++) {
    if (substr($i,1,2) == "-j") jvmoptions = jvmoptions " "
    substr($i,3)
  }
  print jvmoptions
}')
```

4. **AutoProbeConnector** 를 사용하여 **Auto-Probe** 가 사용되도록 설정하려면 2 단계의 문 바로 아래에 다음 줄을 추가합니다.

```
wilyoptions="-Xbootclasspath/p:./wily/connectors/AutoProbeConnector.jar:./wily/Agent.jar -Dcom.wily.introscope.agentProfile=./wily/IntroscopeAgent.profile -Xverify:none"
jvmoptions="$jvmoptions $wilyoptions"
```

5. **javaagent** 옵션을 사용하여 **Auto-Probe** 가 사용되도록 설정하려면 2 단계의 문 바로 아래에 다음 줄을 추가합니다.

```
wilyoptions="-javaagent:./wily/Agent.jar -Dcom.wily.introscope.agentProfile=./wily/IntroscopeAgent.profile"
jvmoptions="$jvmoptions $wilyoptions"
```

## 분산 시스템을 위한 계측된 CTG 시작 스크립트 만들기

Windows 및 UNIX 에서 분산 CTG 시작 파일은 바이너리 파일입니다. 시작 스크립트를 통해 이 파일을 호출하여 필요한 Introscope 매개 변수를 전달해야 합니다.

다음 단계를 수행하십시오.

1. 수동 ProbeBuilder 를 사용하여 CTG 클래스에 프로브를 추가한 후 분산 CTG 서버를 위한 시작 스크립트를 만듭니다. Windows 에서는 스크립트가 **.bat** 파일이고, UNIX 에서는 **.sh** 셸 파일입니다.
2. 시작 스크립트에서 Introscope 클래스와 에이전트 프로필의 위치를 지정합니다.

다음은 Windows 용 샘플 시작 스크립트(.bat 파일)입니다. 여기에서는 CTG 및 WebSphere 가 해당 IBM 제품의 설치 폴더 위치라고 가정합니다.

```
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
```

설명:

<ccc> = CTG 버전 번호

<jjj> = Java 버전 번호

<www> = WebSphere 버전 번호

- 응용 프로그램 시작 스크립트의 클래스 경로를 편집하여 ProbeBuilder 로 생성된 계측된 코드가 들어 있는 디렉터리의 위치를 포함하게 만듭니다.

**참고:** 클래스 경로에서 이러한 항목을 원래 항목 앞에 배치해야 합니다. [계측된 코드 실행](#) (페이지 31)을 참조하십시오.

- 응용 프로그램 시작 스크립트에서 클래스 경로를 편집하여 다음 경로를 포함하도록 만듭니다.

```
<Agent_Home>/ Agent.jar
```

예를 들어, 클래스 경로가 다음과 같다고 가정합니다.

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

이 경우 다음과 같이 수정할 수 있습니다.

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

- 수동 계측을 완료한 후 계측된 코드 및 에이전트의 위치를 반영하도록 CTG 서버 시작 스크립트의 클래스 경로를 업데이트합니다.
- 새 시작 스크립트를 사용하여 응용 프로그램을 시작합니다.

## 계측된 코드 실행

원래 코드 대신 Introscope 지원 코드를 가리키는 방법은 다음 세 가지가 있습니다.

- 클래스 경로에서 원래 클래스 경로를 계측된 코드 경로로 대체합니다. 이 장의 지침에서는 응용 프로그램을 처음 계측할 때 이 과정을 수행하도록 안내했습니다.

**참고:** Windows 또는 UNIX 에 설치하고 이 과정을 수행하지 않은 경우 *Java\bin* 폴더를 사용하도록 클래스 경로를 업데이트해야 할 수 있습니다. 이 폴더는 Java 커뮤니티에서 제공한 Java 가 아닌 WebSphere Application Server 의 *Java\bin* 디렉터리에서 제공됩니다. 이 클래스 경로 업데이트는 JAVA-CTG 호환성을 위해 필요합니다.

- 클래스 경로의 앞부분에 경로를 추가합니다. 응용 프로그램 코드의 일부만 계측된 경우 클래스 경로에서 원래 경로 앞에 계측된 코드 경로를 배치합니다.

**참고:** 이 작업을 수행하면 계측된 코드가 로드되고 성능 데이터를 보고합니다. 계측되지 않은 코드는 계속 로드되고 정상적으로 작동하지만 성능 데이터를 보고하지 않습니다.

- 원래 클래스 경로에 계측된 코드를 배치합니다.

클래스 경로가 여러 위치에 설정되어 있거나 시스템을 평가하는 경우 이 방법을 사용합니다. 프로덕션 환경에서 이 방법을 사용할 경우 주의하십시오. 원래 코드를 사용하는지, 아니면 계측된 코드를 사용하는지 혼동하기 쉽습니다.

- 원래 코드를 새 위치로 이동합니다. 클래스 경로는 그대로 두고 계측된 코드를 원래 위치로 이동합니다.
- UNIX 컴퓨터의 경우 원래 경로 위치에서 계측된 코드를 가리키는 기호로 된 링크를 만들 수도 있습니다.

## 원래 코드로 되돌리기

계측되지 않은 원래 코드로 되돌리려면 다음과 같이 계측을 실행 취소하십시오.

- 계측된 코드의 경로를 Java 클래스 경로에 배치한 경우 계측된 코드의 경로를 원래 값으로 대체하십시오.
- 원래 코드의 경로 앞에 계측된 코드의 경로를 추가한 경우 클래스 경로에서 앞부분에 추가한 경로를 제거하여 원래 클래스 경로만 남기십시오.
- 원래 클래스 경로에서 원래 코드를 제거하고 계측된 코드를 배치한 경우 원래 경로에서 계측된 코드를 제거한 다음 원래 클래스 경로에 원래 코드를 배치하십시오.

**참고:** UNIX 시스템에서 기호로 된 링크를 사용한 경우 기호로 된 링크로 원래 디렉터리를 가리키도록 하거나 링크를 제거한 다음 코드를 원래 클래스 경로로 이동하십시오.

## 분산 CTG 플랫폼에서 ChangeDetector 가 사용되도록 설정

ChangeDetector 는 Windows, Linux 및 UNIX 같은 분산(비 z/OS) 시스템에서 지원됩니다. ChangeDetector 는 CTG 구성 파일(*ctg.ini*, *ctgenvvar* 등)과 관련 JAR 파일을 모두 모니터링하여 시스템에 변경 사항이 있는지 여부를 감지하도록 설계되어 있습니다. 따라서 CTG 환경과 관련된 중단이나 문제를 일으킬 수 있는 구성 변경 사항을 쉽게 추적할 수 있습니다.

CTG 용 확장에는 기본 ChangeDetector 구성 파일이 포함되어 있으며, ChangeDetector 가 CTG 설치를 올바르게 모니터링하도록 설정하려면 이 구성 파일을 사용자 지정해야 합니다. 또한 ChangeDetector 에서 모니터링할 다른 파일을 추가할 수 있습니다.

**참고:** 모니터링할 파일을 추가하는 것에 대한 자세한 내용은 *CA APM ChangeDetector 사용자 안내서*를 참조하십시오.

이 단원에서는 ChangeDetector 의 CTG 관련 모니터링에 대해 중점적으로 다룹니다. *ChangeDetector-config.xml* 파일에는 CTG 및 관련 Introscope Agent 파일 엔터티를 모니터링하기 위한 기본 ChangeDetector 구성 설정 집합이 들어 있습니다. 이 구성을 적절하게 사용하려면 *ChangeDetector-config.xml* 파일에서 두 가지 항목을 업데이트하십시오.

**참고:** 분산 CTG 시스템에서 ChangeDetector 를 사용하는 경우 ChangeDetector 가 Java 클래스 모니터를 사용하는 Java 클래스의 변경 사항을 감지할 수 없다는 것에 주의하십시오. 이 제한은 CTG 에이전트가 수동으로 프로브되고 Java 클래스의 동적 변경 사항을 감지할 수 없기 때문에 발생합니다.

(ChangeDetector 는 파일 시스템과 구성 속성 파일의 변경 사항은 예상대로 모니터링하고 감지할 수 있습니다.)

다음 단계를 수행하십시오.

1. ChangeDetector-config.xml 파일을 업데이트하려면 다음과 같은 CTG 디렉터리 주석 블록을 찾습니다.

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

*name= parameter* 를 사용자의 특정 CTG 디렉터리를 가리키도록 변경합니다.

예를 들어 디렉터리 `ctg<ccc>`

`/usr/lpp/ctg/ctg<ccc>` 아래에 CTG 를 설치한 경우 `scan-directory` 항목의 `name` 매개 변수를 다음과 같이 설정할 수 있습니다.

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

- 다음과 같은 CA APM 디렉터리 주석 블록을 찾습니다.

```
<!-- ===== -->
<!-- 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" />
```

name= parameter 를 사용자의 특정 CA APM 설치 디렉터리를 가리키도록 변경합니다.

예를 들어, CA APM 이 다음 디렉터리에 설치되어 있다고 가정합니다.

`/usr/vendor/ca apm`

이 경우 scan-directory 항목의 name 매개 변수를 다음과 같이 변경할 수 있습니다.

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

- 이러한 수정을 마쳤으면 업데이트된 `ChangeDetector-config.xml` 파일을 Introscope Agent 디렉터리에 배치합니다.
- `ChangeDetectorAgent.jar` 파일이 에이전트의 `.ext` 디렉터리에 설치되어 있는지 확인합니다.
- 에이전트 프로필의 `introscope.changeDetector.profile=config` 항목을 수정하여 ChangeDetector 구성 파일의 경로를 지정합니다.
- 에이전트 프로필의 `introscope.changeDetector.agentID= config` 항목에 ChangeDetector 에이전트용으로 사용할 이름이 들어 있는지 확인합니다.

예를 들어 디렉터리 `ctg<ccc>`

`/usr/lpp/ctg/ctg<ccc>` 아래에 CTG 를 설치한 경우 scan-directory 항목의 name 매개 변수를 다음과 같이 설정할 수 있습니다.

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

위 단계를 수행했으면 ChangeDetector 를 사용하여 CTG 서버를 모니터링할 수 있습니다.

ChangeDetector 데이터의 보기 및 해석에 대한 자세한 내용은 *CA APM ChangeDetector 사용자 안내서*를 참조하십시오.

## CTG 서버 RequestExit 및 전체 통계 프로빙 구성(선택 사항)

IBM RequestExit 메트릭을 가져오도록 CTG 서버를 구성하려면 다음을 수행하십시오.

다음 단계를 수행하십시오.

1. CTG 의 클래스 디렉터리에 PPCTGRequestExit.jar 를 배치합니다.

예:

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

2. CTG 의 CLASSPATH 목록에 PPCTGRequestExit.jar 를 추가합니다.

예:

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

3. IntroscopeAgent.profile 에서 다음 값을 사용자 지정합니다.

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

**참고:** CTG\_Global\_Statistics\_StatsExit 아래의 개요 typeview 는 의도적으로 비워 둔 것이며 이 탭 아래에 어떠한 메트릭 데이터도 포함되지 않습니다. CTG\_Global\_Statistics\_StatsExit 에 대한 typeview 에 데이터가 표시되지 않으면 CTG\_Global\_Statistics\_StatsExit 노드에 포함된 메트릭 데이터가 없는 것입니다.

## 구성 파일 ctg.ini 에서 z/OS 의 CTG 서버 RequestExit 및 전체 통계 프로빙 구성(선택 사항)

z/OS 플랫폼의 경우 z/OS 에 대한 RequestExit 세부 정보와 전역 상태 프로빙을 가져오도록 구성 파일 ctg.ini 를 구성하려면 다음을 수행하십시오.

다음 단계를 수행하십시오.

1. ctg.ini 에 APM/Wily PPCTG 요청 종료의 이름을 추가합니다.
2. CTG 설치 환경에 따라 다른 매개 변수를 추가합니다.

예:

```
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;
```

## Windows 플랫폼에서 RequestExit 세부 정보 및 전체 통계를 개별적으로 가져오도록 구성(선택 사항)

Windows 플랫폼에서 RequestExit 세부 정보 및 전체 통계를 개별적으로 가져오려면 다음을 수행하십시오.

다음 단계를 수행하십시오.

1. ctgservice 명령을 실행하여 요청 종료에 대한 클래스 경로를 설정합니다. 이렇게 하려면 먼저 Agent\wily\examples\ext 폴더에서 IBM CTG\Classes 폴더로 PPCTGRequestExit.jar 를 복사해야 합니다.

예:

```
ctgservice -R -A-classpath=C:\<IBM CTG Home>\classes\PPCTGRequestExit.jar  
-A-j-Dcom.ibm.wily.introscope.agentProfile=C:\<Agent-Home>\wily\core\config\Intro  
scopeAgent.profile  
-A-j-javaagent:C:\<Agent-Home>\wily\Agent.jar
```

2. "IBM CTG Config Tool"(IBM CTG 구성 도구)을 호출하고 "Gateway Daemon"(게이트웨이 데몬) 노드를 선택하여 RequestExit\_Monitor 이름을 CTG 프로그램으로 구성합니다.
3. "Monitoring"(모니터링) 탭을 선택하고 "Use These Request Monitors"(다음 요청 모니터 사용) 필드로 이동한 후 상자에 다음 APM 종료 값을 입력합니다.  
com.ibm.ctg.server.APM\_RequestExit\_Monitor
4. "Add"(추가)를 선택하여 위 항목을 추가합니다.

RequestExit 세부 정보 상태에 대한 매개 변수를 구성하려면 다음을 수행하십시오.

다음 단계를 수행하십시오.

1. 노드 "CICS Transaction Gateway"(CICS 트랜잭션 게이트웨이), "Gateway Daemon"(게이트웨이 데몬), "Statistics API"(통계 API) 옵션을 확장하고 선택합니다.
2. "Enable Protocol Handler"(프로토콜 처리기 사용) 확인란을 선택합니다.
3. (선택 사항) 기본값을 재정의하려면 TCP 포트 번호, 시간 만료 및 기타 옵션을 변경합니다.
4. "BinAddress" 필드를 비워 두는 경우 기본적으로 CTG 는 "localhost"를 사용합니다.

이전 단계를 사용하면 IBM 요청 종료 세부 정보 메트릭을 수집할 수 있습니다.

RequestExit 전체 통계에 대한 매개 변수를 구성하려면 다음을 수행하십시오.

다음 단계를 수행하십시오.

**참고:** IBM 의 제한된 지원으로 인해 요청 종료 전체 통계가 자동으로 사용되도록 설정할 수 없습니다.

1. PP 에 의한 전체 통계 수집을 호출하려면 PPrunGlobalStats.bat 파일을 실행하고 후속 항목을 사용자 지정합니다.
2. 다음과 같이 CTG 클래스 및 Introscope Agent 디렉터리의 위치를 지정합니다.

```
set CTG_CLASSES= <CTG 클래스 파일의 위치>
set ISCOPE_AGENT= <에이전트 확장 폴더의 위치>
```

## CTG 클라이언트 지원 구성

CTG 용 확장은 CTG 서버 및 CTG 클라이언트 응용 프로그램 모두에 대한 메트릭을 제공합니다. 이 단원에서는 CTG 클라이언트 응용 프로그램을 CTG 서버로 요청을 보내는 응용 프로그램으로 정의하는 방법에 대해 설명합니다. 또한 CTG 서버는 특정 CICS ECI 또는 EPI 응용 프로그램을 실행하도록 요청하며, 결과를 클라이언트로 반환합니다.

CTG 용 확장에 의해 제공되는 클라이언트 측 메트릭은 측정 범위를 시작 클라이언트까지 확장합니다. 이 범위를 사용하면 응답 시간, 심사 등의 종합적인 관리가 가능합니다.

CTG 용 확장을 독립 실행형 CTG 클라이언트 응용 프로그램이나 WebSphere 에서 실행되는 클라이언트 응용 프로그램과 함께 작동하도록 구성할 수 있습니다.

## 독립 실행형 CTG 클라이언트 응용 프로그램 구성

독립 실행형 CTG 클라이언트 응용 프로그램의 경우 Introscope 를 사용하여 측정하는 다른 모든 독립 실행형 Java 응용 프로그램과 동일한 방식으로 Introscope 지원을 구성합니다. 또한 *PPCTGClient-full.pbd* 및 *required.pbd* 파일을 *IntroscopeAgent.profile* 의 *introscope.autoprobe.directivesFile=<parameter>* 에 추가해야 합니다. 로컬 모드(EXCI) 상호 작용을 모니터링하려면 *PPCTGServer-full.pbd* 도 목록에 추가해야 합니다.

**참고:** CTG 확장의 *PPCTGAgent.jar* 파일이 에이전트의 *ext* 디렉터리에 있는지 확인하십시오.

## 채널 및 컨테이너 메트릭 구성

채널 및 컨테이너 메트릭을 전환하여 채널 및 컨테이너와 관련된 메트릭 수집 및 CPU 오버헤드를 최소화하는 옵션이 PBD 파일에서 제공됩니다. 즉, *PPCTGServer-full.pbd* 및 *PPCTGServer-typical.pbd* 파일에서 지시문을 주석 처리하거나 지시문의 주석 처리를 제거하면 됩니다.

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

변경 사항을 적용하려면 에이전트를 다시 시작해야 합니다.

## WebSphere 에서 클라이언트 응용 프로그램 구성

WebSphere 에서 실행되는 클라이언트 응용 프로그램의 경우 CTG 클라이언트 지원을 구성하는 방법은 두 가지가 있습니다.

다음 단계를 수행하십시오.

- CTG 의 *ctgclient.jar*, *cicsj2ee.jar*, *ccf2.jar* 등과 같은 파일을 WebSphere JVM 의 클래스 경로 구성 메뉴에 직접 추가합니다.

**참고:** CTG 8.0 부터 *cicsj2ee.jar* 의 이름이 *cicsjee.jar* 로 변경되었습니다.

- WebSphere 의 리소스 어댑터 화면에서 CTG jar 파일을 설치하고 관련 *cicseci.rar* 및/또는 *cicsepi.rar* 파일을 설치합니다.

WebSphere 에 CTG 클라이언트 지원을 추가한 후에는 *PPCTGClient-full.pbd* 및 *required.pbd* 파일을 *IntroscopeAgent.profile* 의 *introscope.autoprobe.directivesFile=<parameter>*에 추가해야 합니다. 로컬 모드(EXCI) 상호 작용을 모니터링하려면 *PPCTGServer-full.pbd* 를 목록에 추가합니다.

**참고:** CTG 확장의 *PPCTGAgent.jar* 파일이 에이전트의 *ext* 디렉터리에 있는지 확인하십시오. 포함되는 파일 목록은 [CTG 용 확장의 구성 요소](#) (페이지 12)를 참조하십시오.

## CTG 용 확장 업그레이드

이전 버전의 CTG 용 확장에서 업그레이드할 수 없습니다. 먼저 이전 버전을 제거한 다음 최신 버전을 설치하십시오.

## 설치 및 구성 확인

CTG 용 확장이 올바르게 설치 및 구성되었는지 확인해야 합니다.

다음 단계를 수행하십시오.

1. 모니터링되는 WebSphere Application Server, Java 응용 프로그램, CTG 서버 소프트웨어 및 Introscope Enterprise Manager 를 다시 시작합니다.
2. 이러한 구성 요소를 성공적으로 다시 시작하면 CTG 서버 및 WebSphere Application Server 를 모니터링하는 에이전트가 Enterprise Manager 로 데이터를 보고하기 시작합니다.
3. Workstation 을 시작하고 Enterprise Manager 에 연결합니다.
4. [CTG 서버 및 응용 프로그램 서버의 데이터](#) (페이지 41)를 Investigator 에서 볼 수 있는지 확인하십시오.



## 제 3 장: CTG 용 확장 사용

---

이 단원에서는 다음과 같은 항목에 대해 설명합니다.

- Introscope 를 사용하여 CTG 환경을 모니터링하는 방법
- CTG 용 확장을 통해 사용할 수 있는 다양한 종류의 데이터에 대한 개요
- Workstation 및 콘솔에서 데이터를 볼 수 있는 방법

이 단원의 지침에서는 설치 단계를 완료했으며 Introscope 지원 CTG 응용 프로그램이 시작되고 실행되어 Enterprise Manager 로 데이터를 보고하고 있다고 가정합니다.

이 섹션은 다음 항목을 포함하고 있습니다.

[Introscope 에서 CTG 용 확장 데이터 보기](#) (페이지 41)

[트리 뷰에서 메트릭 데이터 보기](#) (페이지 42)

[서버 메트릭](#) (페이지 44)

[Introscope Investigator 탭 뷰 사용](#) (페이지 52)

[대시보드에서 CTG 용 확장 데이터 보기](#) (페이지 53)

[Introscope 주의/위험 경고 임계값 수정](#) (페이지 62)

[CTG 트랜잭션 추적 프로그램](#) (페이지 62)

### Introscope 에서 CTG 용 확장 데이터 보기

Introscope 에서 CTG 용 확장의 데이터를 보는 여러 가지 방법이 있습니다.

- Investigator 의 원시 메트릭 - 기술 사용자에게 모든 CTG 리소스와 구성 요소의 기반 성능에 대한 자세한 뷰를 제공합니다.
- Investigator 의 탭 뷰 - 기술 사용자에게 CTG 시스템의 성능, 리소스 및 구성 요소에 대한 집계 뷰를 제공합니다.
- 콘솔의 대시보드 - CTG 아키텍처를 자세히 알지 못하는 사용자도 쉽게 사용할 수 있는 인터페이스를 제공합니다.
- Investigator 의 경고 - CTG 용 확장 대시보드에서 생성된 경고와 사용자 생성한 경고를 표시합니다.

## 트리 뷰에서 메트릭 데이터 보기

Investigator 에서 CTG 용 확장 메트릭을 보려면 다음 단계를 따르십시오.

다음 단계를 수행하십시오.

1. 관리되는 응용 프로그램을 시작합니다.
2. Enterprise Manager 를 시작합니다.
3. Workstation 을 시작하고 로그인합니다.
4. Investigator 창을 엽니다.

트리의 여러 노드 아래에 모든 CTG 용 확장 관련 메트릭이 나타납니다.

**참고:** 사용할 수 있는 메트릭은 응용 프로그램이 사용하는 CTG 및 WebSphere 리소스에 따라 달라집니다. 관리되는 Java 응용 프로그램에서 사용하는 메트릭만 나타납니다.

### 프런트엔드

다음 프런트엔드 CTG 메트릭을 사용할 수 있습니다.

- **Apps** - 개별 응용 프로그램에 대한 메트릭은 이 노드 아래에 응용 프로그램 이름별로 나타납니다.
- **Client\_to\_CTG\_Aggregates** - 이 노드에는 "클라이언트 집계 그래픽 요약" 탭이 있습니다. 트리에서 이 노드 아래에는 여러 개의 집계 메트릭과 다음에 대한 하위 노드가 있습니다.
  - **BASE\_ECI\_EPI** - CTG 8.x 가 아니며 요청 종료 계측이 아닌 계측을 포함하는 이전 버전의 CTG 를 사용할 경우 이 노드가 표시될 수 있습니다.
  - **JCA\_ECI**
  - **JCA\_EPI** - CTG 8.x 가 아니며 요청 종료 계측이 아닌 계측을 포함하는 이전 버전의 CTG 를 사용할 경우 이 노드가 표시될 수 있습니다.
    - 화면
    - 터미널

- **Client\_to\_CTG\_Details** - 이 노드에는 "클라이언트 집계 그래픽 요약" 탭이 있습니다. 트리에서 이 노드 아래에는 개별 클라이언트를 나타내는 하위 노드가 있습니다.
- **Client\_to\_CTG\_JSSE\_Session** - 이 노드에는 "JSSE - CTG 게이트웨이" 탭이 있습니다. 트리에서 이 노드 아래에는 여러 개의 집계 메트릭이 있습니다.

## 백엔드

다음 백엔드 CTG 메트릭을 사용할 수 있습니다.

- **CTG\_Global\_Statistics** - 이 노드에는 통계가 표시됩니다.
  - CICS 집계
  - 연결 관리자
  - 연결 관리자 스레드
  - 게이트웨이 데몬
  - HTTPRequest
  - 클라이언트에서 들어오는 연결 요청
  - 개별 서버
  - 세션
  - 작업자 스레드
- **CTG\_to\_CICS\_ECI\_IPIC** - 이 노드에는 "ECI 모든 게이트웨이 그래픽 요약" 탭이 있습니다. 트리에서 이 노드 아래에는 각 클라이언트의 메트릭을 포함하는 하위 노드가 있습니다.
- **CTG\_to\_CICS\_EPI** - 이 노드에는 "EPI 모든 게이트웨이" 탭이 있습니다. 트리에서 이 노드 아래에는 각 클라이언트의 메트릭을 포함하는 하위 노드가 있습니다.
- **JSSE\_to\_CTG** - 이 노드에는 "JSSE 그래픽 요약" 탭이 있습니다. 트리에서 이 노드 아래에는 다음 두 메트릭을 표시하는 "들어오는 JSSE 세션 요약" 탭이 있는 하위 노드가 있습니다.
  - 들어오는 SSL 핸드셰이크 집계
  - 간격당 들어오는 SSL 핸드셰이크

**참고:** IPIC 트랜잭션이 실행되면 HTTPRequest 노드가 표시됩니다.

## 특정 메트릭의 기록 데이터 분석

특정 메트릭의 기록 성능 데이터를 분석하려면 *영구 수집*을 설정하십시오.

**참고:** 영구 수집 설정에 대한 자세한 내용은 *CA APM 설치 및 업그레이드 안내서*를 참조하십시오.

## 서버 메트릭

이 단원에서는 CTG 용 확장에서 사용할 수 있는 백엔드 및 프론트엔드 메트릭에 대한 정보를 제공합니다.

### 백엔드 | CTG\_Global\_Statistics

CTG\_Global\_Statistics 노드를 선택하면 Introscope Investigator 에 다음과 같은 몇 가지 메트릭이 있는 전체 통계 그래픽 요약 탭이 표시됩니다.

- 간격당 클라이언트 요청 호출 집계
- 클라이언트 요청 처리 응답 시간
- 클라이언트 회신 보내기 응답 시간
- 사용 중인 연결 관리자 스레드
- 사용 중인 작업자 스레드

일반적으로 연결 스레드 풀은 클라이언트 응용 프로그램과 CTG 서버 사이의 요청을 처리하는 데 사용됩니다. 또한 작업자 스레드 풀은 메인프레임에서 CICS 와 CTG 서버 사이의 요청을 처리하는 데 사용됩니다.

CTG\_Global\_Statistics 노드에는 전체 통계 표 요약 탭도 포함되어 있습니다. 이 탭은 CTG 서버가 시작된 이후로 얼마나 많은 작업이 해당 CTG 서버를 통해 이동했는지를 보여 주는 일련의 실행 집계 통계를 제공합니다. 이 페이지에는 다음과 같은 통계가 표시됩니다.

- 받은 총 패킷
- 보낸 총 패킷 수
- 사용 중인 연결 관리자 스레드
- 사용 중인 작업자 스레드

Introscope Investigator 트리에서 CTG\_Global\_Statistics 노드 아래에는 CTG 서버를 통해 이동한 작업에 대한 전체 통계 집합이 표시됩니다. Investigator 트리는 계층적으로 배치되어 있습니다. 즉, 계층 구조의 각 하위 노드가 CTG 서버 내의 특정 기능 집합에 대한 주요 통계를 보여 줍니다.

CICS 집계 하위 노드에는 CTG 서버와 업스트림 CICS 시스템 사이를 이동하는 트래픽에 대한 주요 통계가 표시됩니다. 이 하위 노드에는 다음과 같은 통계가 포함되어 있습니다.

- Read packets per interval and response time (SNA or TCP packets received)(간격당 읽은 패킷 및 응답 시간(받은 SNA 또는 TCP 패킷))
- Write packets per interval and response time (SNA or TCP packets sent)(간격당 쓴 패킷 및 응답 시간(보낸 SNA 또는 TCP 패킷))

**참고:** JCA 의 경우 다양한 데이터 값의 "공용 영역 집계 요청 데이터" 메트릭이 표시됩니다. "기본"의 경우 메트릭이 표시되지 않습니다.

CICS WLM 하위 노드에는 내부 WLM 이 처리할 요청을 예약하는 방식과 사용자 작성 종료가 작동 중인지 여부가 표시됩니다.

연결 관리자 하위 노드에는 CTG 서버와 요청하는 CTG 클라이언트 사이의 트래픽 처리와 관련된 통계가 표시됩니다.

"연결 관리자 스레드" 하위 노드에는 클라이언트에 대한 연결 관리자 요청/회신을 처리하는 데 사용되는 스레드 풀의 사용량과 관련된 통계가 표시됩니다. 이러한 통계는 클라이언트와 관련된 스레드 병목 현상이 발생하는지 여부를 나타냅니다.

"Gateway Daemon"(게이트웨이 데몬) 하위 노드에는 이동하는 요청의 총 수와 수행 중인 관련 CICS 트랜잭션 커밋 또는 롤백(SYNCONRETURN 등)의 총 수가 표시됩니다.

"클라이언트에서 들어오는 연결 요청" 하위 노드에는 클라이언트에서 CTG 서버로 들어오는 요청의 간격당 도착 비율이 표시됩니다.

"작업자 스레드" 하위 노드에는 작업 요청을 CICS 로 발송하는 데 사용되는 스레드 풀의 사용량과 관련된 통계가 표시됩니다. 이 통계는 CICS 로 업스트림되는 스레드 병목 현상이 발생하는지 여부를 나타낼 수 있습니다.

## 백엔드 | CTG\_to\_CICS\_ECI\_IPIC 노드

CTG\_to\_CICS\_ECI\_IPIC 노드를 선택하면 Introscope Investigator 에 다음과 같은 몇 가지 메트릭이 있는 ECI 모든 게이트웨이 그래픽 요약 탭이 표시됩니다.

- 호출된 프로그램 수 집계
- 간격당 프로그램 호출 수
- 프로그램 평균 응답 시간(ms)
- 프로그램 오류 집계

## CTG to CICS ECI Host ID(CTG - CICS EPI 호스트 ID) 하위 노드

각 CICS ECI 호스트마다 부모 CTG\_to\_CICS\_ECI\_IPIC 노드 아래에 자체 하위 노드가 있습니다. CTG to CICS ECI Host ID(CTG - CICS ECI 호스트 ID) 하위 노드를 선택하면 Introscope Investigator 뷰어 창의 ECI 게이트웨이 표 요약 탭에 다음과 같은 메트릭이 표시됩니다.

- Aggregate Program Count(프로그램 수 집계)
- 평균 응답 시간(ms)
- 응답/간격
- 오류/간격
- 중단

트리에는 각 호스트에서 실행되는 각 CICS 프로그램에 해당하는 상세 하위 노드가 표시됩니다.

## 백엔드 | CTG\_to\_CICS\_EPI 노드

CTG\_to\_CICS\_EPI 노드를 선택하면 Introscope Investigator 에 다음과 같은 몇 가지 메트릭이 있는 EPI 모든 게이트웨이 탭이 표시됩니다.

- 호출된 트랜잭션 수 집계
- 트랜잭션 평균 응답 시간(ms)
- 간격당 트랜잭션 호출 수
- 트랜잭션 오류 집계

- 호출된 서비스 요청 수 집계
- 서비스 요청 오류 집계

### CTG to CICS EPI Host ID(CTG - CICS EPI 호스트 ID) 하위 노드

각 CICS EPI 호스트마다 부모 CTG\_to\_CICS\_EPI 노드 아래에 자체 하위 노드가 있습니다. CTG to CICS EPI Host ID(CTG - CICS EPI 호스트 ID) 하위 노드를 선택하면 Introscope Investigator 뷰어 창의 EPI 집계 요약 탭에 다음과 같은 메트릭이 표시됩니다.

- 호출된 트랜잭션 수 집계
- 초당 트랜잭션 호출 수
- 트랜잭션 평균 응답 시간(ms)
- 트랜잭션 오류 집계

트리에는 각 호스트에서 실행되는 각 CICS 프로그램에 해당하는 상세 하위 노드가 표시됩니다.

### 백엔드 | JSSE - CTG 노드

JSSE - CTG 노드를 선택하면 Introscope Investigator 에 다음과 같은 두 가지 메트릭이 있는 JSSE 그래픽 요약 탭이 표시됩니다.

- 들어오는 SSL 핸드셰이크 집계
- 간격당 들어오는 SSL 핸드셰이크

"JSSE - CTG" 노드 아래에는 JSSE ServerSocket 에 대한 하위 노드가 있습니다.

### JSSE ServerSocket 하위 노드

JSSE ServerSocket 하위 노드에는 다음과 같은 두 가지 메트릭이 있는 들어오는 JSSE 세션 요약 탭이 표시됩니다.

- 들어오는 JSSE SSL 핸드셰이크 수 집계
- 간격당 들어오는 JSSE SSL 핸드셰이크 호출

## 프런트엔드 | Client\_to\_CTG\_Aggregates 노드

Client\_to\_CTG\_Aggregates 노드를 선택하면 Introscope Investigator 에 다음과 같은 메트릭이 있는 클라이언트 집계 그래픽 요약 탭이 표시됩니다.

- TCP 열기 집계
- TCP 흐름 집계 수
- TCP 흐름 응답 시간(ms)
- SSL 열기 집계
- SSL 흐름 집계 수
- SSL 흐름 응답 시간(ms)

Introscope Investigator 트리의 Client\_to\_CTG\_Aggregates 노드 아래에는 다음과 같은 개별 메트릭(일부는 탭 뷰에 표시됨)과 여러 하위 노드가 있습니다.

- Aggregate EPI Errors(EPI 오류 집계)
- 트랜잭션 오류 집계
- CICS Resource Exceptions Per Interval(간격당 CICS 리소스 예외)
- CICS Txn Abend Exceptions Per Interval(간격당 CICS Txn 이상 종료 예외)
- EPI Gateway Exceptions Per Interval(간격당 EPI 게이트웨이 예외)
- EPI Request Exceptions Per Interval(간격당 EPI 요청 예외)
- Resource Exceptions Per Interval(간격당 리소스 예외)
- SSL 흐름 집계
- SSL 흐름 집계 수
- SSL 흐름 응답 시간(ms)
- TCP 열기 집계
- TCP 흐름 집계 수
- TCP 흐름 응답 시간(ms)

다음 단원에서는 Client\_to\_CTG\_Aggregates 노드 아래에 있는 하위 노드에 대해 설명합니다.

### 프런트엔드 | Client\_to\_CTG\_Aggregates | BASE\_ECI\_EPI 하위 노드

이 노드를 선택하면 Introscope Investigator 에 클라이언트 집계 요약 탭이 표시됩니다. 이 탭에는 다음 메트릭에 대한 차트가 표시됩니다.

- 호출된 프로그램 수 집계
- 프로그램 평균 응답 시간(ms)
- 간격당 프로그램 호출 수
- 열려 있는 연결 집계

### 프런트엔드 | Client\_to\_CTG\_Aggregates | CCF\_ECI 하위 노드

이 노드를 선택하면 Introscope Investigator 에 클라이언트 집계 요약 탭이 표시됩니다. 이 탭에는 다음 메트릭에 대한 차트가 표시됩니다.

- 호출된 프로그램 수 집계
- 프로그램 평균 응답 시간(ms)
- 간격당 프로그램 호출 수
- 열려 있는 연결 집계

이 탭은 BASE\_ECI\_EPI 노드에 대해 표시되는 탭과 유사합니다.

### 프런트엔드 | Client\_to\_CTG\_Aggregates | CCF\_EPI 하위 노드

이 노드를 선택하면 Introscope Investigator 에 클라이언트 집계 요약 탭이 표시됩니다. 이 탭에는 다음 메트릭에 대한 차트가 표시됩니다.

- 호출된 트랜잭션 수 집계
- 트랜잭션 평균 응답 시간(ms)
- 간격당 트랜잭션 호출 수
- 열려 있는 연결 집계

### 프런트엔드 | Client\_to\_CTG\_Aggregates | JCA\_ECI 하위 노드

이 노드를 선택하면 Introscope Investigator 에 클라이언트 집계 요약 탭이 표시됩니다. 이 탭에는 다음 메트릭에 대한 차트가 표시됩니다.

- 호출된 CICS 프로그램 수 집계
- 간격당 프로그램 호출 수

- 프로그램 평균 응답 시간(ms)
- 열려 있는 연결 집계

### 프런트엔드 | Client\_to\_CTG\_Aggregates | JCA\_EPI 하위 노드

이 노드를 선택하면 Introscope Investigator 에 클라이언트 집계 요약 탭이 표시됩니다. 이 탭에는 다음 메트릭에 대한 차트가 표시됩니다.

- 호출된 트랜잭션 수 집계
- 트랜잭션 평균 응답 시간(ms)
- 간격당 트랜잭션 호출 수
- 열려 있는 연결 집계

### 프런트엔드 | Client\_to\_CTG\_Details 노드

Client\_to\_CTG\_Details 노드는 모든 게이트웨이에 대한 중단 간 응답 시간 집계를 제공합니다. 이 노드를 선택하면 Introscope Investigator 에 모든 클라이언트 게이트웨이 표 요약 탭이 표시됩니다. 이 탭에는 다음과 같은 열이 표시됩니다.

- 서버
- 프로그램 이름
- 집계 수
- 평균 응답 시간(ms)
- 응답/간격
- 오류/간격
- 중단

### ECI 호스트 세부 정보 하위 노드

Client\_to\_CTG\_Details 노드 아래에는 CICS 프로그램을 호스팅하는 서버에 해당하는 하위 노드가 있습니다. 이러한 서버 하위 노드 중 하나를 선택하면 Introscope Investigator 뷰어 창에 다음과 같은 메트릭이 표 형태로 표시되는 클라이언트 게이트웨이 표 요약 탭이 나타납니다.

- 집계 수
- 평균 응답 시간(ms)

- 응답/간격
- 오류/간격
- 중단

### ECI 프로그램 세부 정보 하위 노드

ECI 호스트 하위 노드 중 하나를 확장하면 트리에 서버의 각 프로그램에 대한 추가 하위 노드가 표시됩니다. 이러한 프로그램 노드를 선택하면 프로그램 세부 정보 뷰 탭이 표시됩니다. 이 탭에는 다음과 같은 메트릭에 대한 차트가 표시됩니다.

- 요청 수 집계
- 간격당 호출 수
- 평균 응답 시간(ms)
- 오류 집계

### 프런트엔드 | Client\_to\_CTG\_Details 트랜잭션 뷰 노드

Introscope Investigator 트리의 프런트엔드 | Client\_to\_CTG\_Details 노드 아래에 있는 각 하위 노드는 EPI 호스트에 해당합니다.

호스트 ID 하위 노드를 선택하면 Introscope Investigator 뷰어 창에 클라이언트 게이트웨이 표 요약 탭이 표시됩니다. 이 탭에는 호스트에서 실행 중인 각 프로그램에 대한 다음과 같은 메트릭이 표시됩니다.

- 집계 수
- 평균 응답 시간(ms)
- 응답/간격
- 오류/간격
- 중단

각 호스트 하위 노드 아래에는 호스트에서 실행 중인 각 프로그램에 대한 추가 하위 노드가 있습니다. 이러한 프로그램 하위 노드 중 하나를 선택하면 Introscope Investigator 뷰어 창에 다음과 같은 메트릭에 대한 차트가 있는 프로그램 세부 정보 뷰 탭이 표시됩니다.

- 요청 수 집계
- 간격당 호출 수

- 평균 응답 시간(ms)
- 오류 집계

## 프런트엔드 | Client\_to\_CTG\_JSSE 세션 노드

Client\_to\_CTG\_JSSE 세션 노드는 모든 게이트웨이에 대한 중단 간 응답 시간 집계를 제공합니다. 이 노드를 선택하면 Introscope Investigator 에 JSSE - CTG 게이트웨이 탭이 표시됩니다. 이 탭에는 다음 메트릭에 대한 차트가 표시됩니다.

- 생성된 SSL 세션 수 집계
- SSL 핸드셰이크 평균 응답 시간(ms)
- 간격당 SSL 세션 호출 수
- 간격당 SSL 암호화된 데이터 요청 수

Introscope Investigator 트리의 Client\_to\_CTG\_JSSE Session 세션 노드 아래에는 다음과 같은 메트릭이 표시됩니다.

- Aggregate SSL Data Flows(SSL 데이터 흐름 집계)
- Aggregate SSL Handshakes(SSL 핸드셰이크 집계)
- Aggregate SSL Opens(SSL 열기 집계)
- SSL Data Flow Response Time (ms)(SSL 데이터 흐름 응답 시간(ms))
- SSL Data Flows Per Interval(간격당 SSL 데이터 흐름)
- SSL Handshake Allocate Response Time (ms)(SSL 핸드셰이크 할당 응답 시간(ms))
- 중단 수

## Introscope Investigator 탭 뷰 사용

Introscope Investigator 뷰어 창의 맨 위에 있는 탭을 사용하여 창에 표시되는 뷰를 변경할 수 있습니다. 트리에 있는 많은 CTG 관련 노드에는 특수한 요약 뷰가 기본적으로 제공됩니다.

다음 탭은 표준으로 제공되는 탭이며 사용하는 확장에 관계없이 Introscope Workstation 에서 사용할 수 있습니다.

- 일반 - "일반" 탭은 Introscope Investigator 트리에서 에이전트 아래에 있는 항목을 선택할 경우의 기본 탭입니다. 메트릭을 선택하면 "일반" 탭에 라이브 데이터 또는 선택한 기록 기간에 대한 메트릭이 시각화되어 나타납니다. 트리에 있는 노드의 경우 "일반" 탭에 Introscope Investigator 계층의 해당 노드 개체에 대한 경로가 표시됩니다.
- 개요 - Introscope Investigator 트리에서 에이전트를 선택하면 사용할 수 있는 "개요" 탭에서는 응용 프로그램을 모니터링할 수 있습니다. 여기에는 상위 수준 건전성 표시기와 관련 이벤트 및 기록 메트릭 정보의 로그가 표시됩니다.
- 검색 - Introscope Investigator 트리에서 메트릭을 포함하는 노드를 선택하면 사용할 수 있는 "검색" 탭에서는 메트릭을 빠르게 찾을 수 있습니다.
- 추적 - Introscope Investigator 트리에서 리소스나 구성 요소를 선택하면 사용할 수 있는 "추적" 탭은 추적 뷰어와 유사합니다. 여기에는 현재 선택한 리소스 또는 구성 요소가 참여한 트랜잭션 추적이 나열됩니다.
- 오류 - Introscope Investigator 트리에서 리소스나 구성 요소를 선택하면 사용할 수 있는 "오류" 탭에는 선택한 항목에 대한 오류와 오류 세부 정보가 나열됩니다.

**참고:** 표준 Introscope 탭에 대한 자세한 내용은 *CA APM Workstation 사용자 안내서*를 참조하십시오.

## 대시보드에서 CTG 용 확장 데이터 보기

CTG 용 확장에는 여러 개의 미리 구성된 Introscope 대시보드와 경고가 있습니다.

CTG 클라이언트 또는 서버의 모든 대시보드는 Introscope 에 설치되어 있는 다른 관리 모듈과 구별하기 쉽게 CTGClient 또는 CTGServer 로 시작합니다.

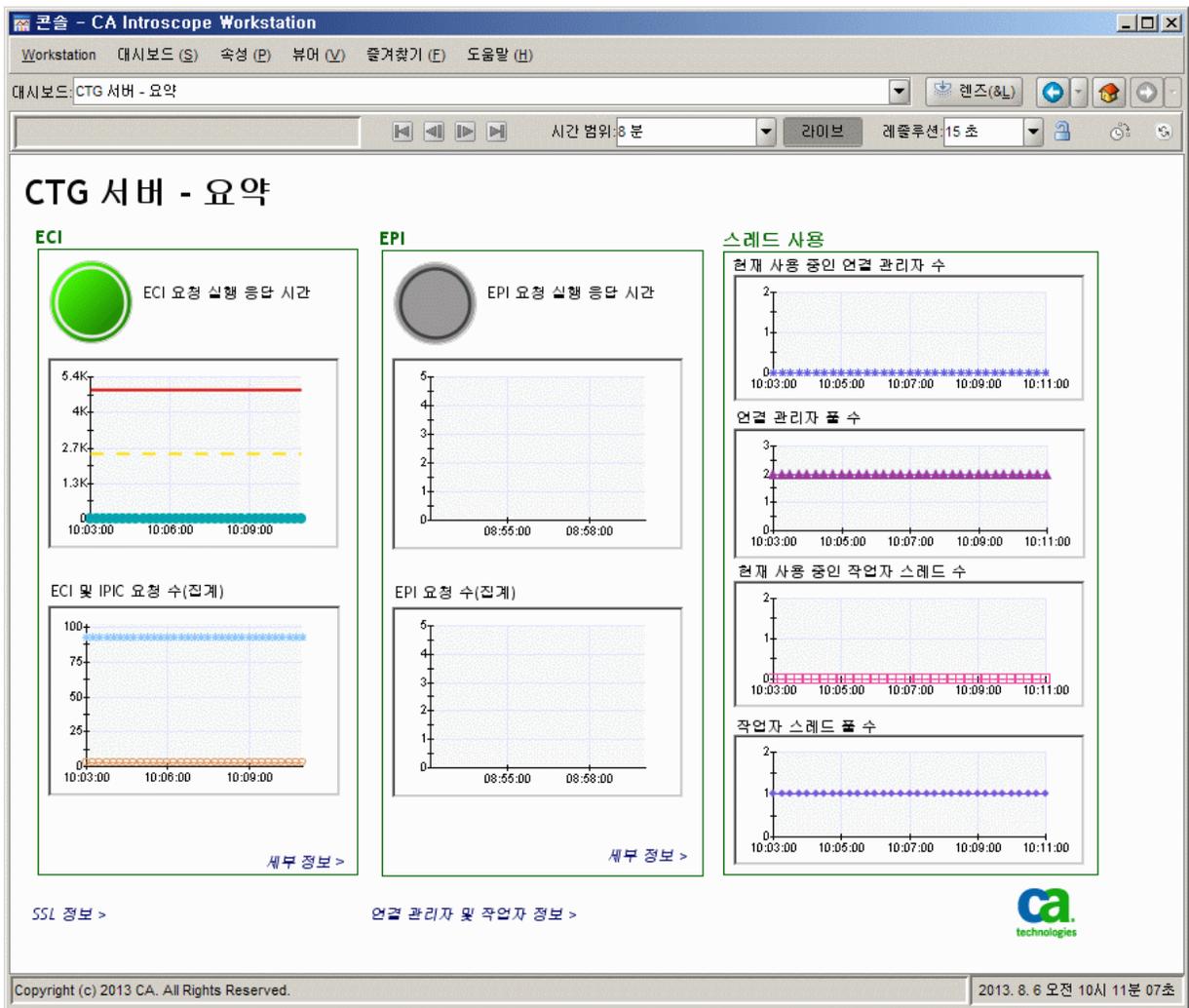
대시보드를 보려면 Introscope Workstation 을 시작하고 콘솔을 엽니다. 또한 CTG 용 확장에는 이러한 대시보드에 표시되는 여러 메트릭에 대한 기본 주의/위험 경고 임계값이 설정되어 있습니다.

**참고:** CTG 응용 프로그램의 고유한 형태 때문에 CTG 용 확장에 포함된 일부 패널 및 대시보드가 비어 있을 수 있습니다.

## CTG 서버 - 개요 대시보드

CTG 서버 - 개요 대시보드에는 Introscope 로 모니터링되는 세 가지 주요 CTG 서버 영역에 대한 요약이 표시됩니다. 또한 이 대시보드에는 이러한 영역의 세부 정보를 제공하는 대시보드와 함께 다음 대시보드로 연결되는 링크가 표시됩니다.

- CTG 클라이언트 - Java 게이트웨이 및 SSL
- CTG 서버 - 연결 관리자 및 작업자



신호등은 기반 메트릭 중에 정의된 임계값을 초과한 메트릭이 있는지 여부를 시각적으로 나타냅니다.

"스레드 사용" 섹션에는 CTG 의 현재 연결 및 작업자 스레드 사용량이 표시됩니다. 일반적으로 연결 스레드는 클라이언트가 CTG 에 연결할 때 할당되고 클라이언트와 CTG 의 연결이 끊길 때 해제됩니다. CTG 와 CICS 간에 이동하는 활성 요청 각각에 대해 *작업자 스레드*가 할당됩니다.

CTG 서버 - 개요 대시보드에는 다음과 같은 메트릭이 표시됩니다.

- ECI
  - ECI 요청 실행 응답 시간
  - ECI 요청 수
- EPI
  - EPI 요청 실행 응답 시간
  - EPI 요청 수

현재 사용 중 메트릭은 CTG 서버 코드로 요청(또는 회신)을 처리 중인 스레드 수를 나타냅니다. 풀 수 메트릭은 해당 엔터티(연결 관리자 또는 작업자)에 할당된 스레드 수를 나타냅니다. 풀 수는 처음에 *ctg.ini* 파일에 정의되어 있는 *initconnect* 매개 변수(연결 관리자 스레드인 경우) 및 *initworker* 매개 변수(작업자 스레드인 경우)에 따라 설정되어 있습니다. 작업 부하가 증가함에 따라 각각의 풀 수가 *ctg.ini* 의 *maxconnect* 및 *maxworker* 매개 변수로 지정된 최대값까지 증가합니다. 풀 수에 기록된 최고 값을 사용하여 *ctg.ini* 파일의 관련 *maxconnect* 및 *maxworker* 매개 변수를 세부적으로 조정할 수 있습니다.

## CTG 요약 - 서버 ECI 동작 대시보드

CTG 요약 - 서버 ECI 동작 대시보드는 문제를 일으킬 수 있는 개체 및 실행에 대한 간략한 스냅샷을 제공합니다.

CTG 요약 - 서버 ECI 동작 대시보드에는 다음과 같은 메트릭이 표시됩니다.

- ECI 모든 게이트웨이 - 집계 요약
  - 호출된 프로그램 수
  - 간격당 프로그램 호출 수
  - 프로그램 평균 응답 시간

- 네트워크 집계
  - 간격당 TCP 소켓 신규 수락 수
  - 간격당 JSSE/SSL 호출 수
  - TCP 소켓 입력 대역폭
  - TCP 소켓 출력 대역폭
  - JSSE/SSL 새 집계 수

**참고:** "TCP 소켓 입력 대역폭" 및 "TCP 소켓 출력 대역폭"을 표시하려면 `IntroscopeAgent.profile` 에서 `introscope.agent.sockets.reportRateMetrics=true` 속성을 설정하십시오.

## CTG 서버 - 전체 통계 대시보드

CTG 서버 - 전체 통계 대시보드는 CTG 서버의 관점에서 전체 CTG 통계에 대한 간략한 스냅샷을 제공합니다. 위쪽 구석에 있는 링크를 사용하면 CTG 서버 - 개요 대시보드를 탐색할 수 있습니다.

CTG 요청 종료 - 전체 통계 대시보드에는 다음과 같은 메트릭이 표시됩니다.

- 처리된 I/O 요청(집계)
- 간격당 처리된 I/O 요청
- 클라이언트에서 들어오는 연결 요청(간격당)
- CICS 에 대한 ECI/IPIC/EPI 요청 합계(간격당)
- 현재 사용 중인 연결 관리자 스레드
- 연결 관리자 스레드 풀 수
- 현재 사용 중인 작업자 스레드
- 작업자 스레드 풀 수

## CTG 서버 - EPI 요청 대시보드

CTG 서버 - ECI 요청 대시보드는 문제를 일으킬 수 있는 개체 및 실행에 대한 간략한 스냅샷을 제공합니다. 위쪽 구석에 있는 링크를 사용하면 CTG 서버 - 개요 대시보드를 탐색할 수 있습니다.

이 대시보드에는 다음과 같은 메트릭이 표시됩니다.

- 실행 성능
  - 요청 실행 시간
  - 실행 목록 시간
  - 요청 실행 수
  - 실행 목록 수
- Request Initialize/Terminate and Write/Read Objects(요청 초기화/종료 및 읽기/쓰기 개체)
  - 쓰기/읽기 개체 시간
  - 쓰기/읽기 개체 수

쓰기/읽기 개체는 CTG 클라이언트와 CTG 서버 간에 교환되는 서비스 메시지(패킷)를 나타내는 데 사용되는 IBM CTG 용어입니다. 이 측정 그룹은 보낸 메시지의 합계, 평균 응답 시간 등을 표시합니다. CTG 서버가 오버로드되는 경우(예: 작업자 또는 연결 스레드가 부족해지는 경우) 메시지가 사용할 수 있는 스레드를 기다리기 위해 큐에 저장되고 응답 시간이 증가합니다.

## CTG 서버 - EPI 요청 대시보드

CTG 서버 - EPI 요청 대시보드는 서버 요청의 성능 요약을 제공합니다. 위쪽 구석에 있는 링크를 사용하면 CTG 서버 - 개요 대시보드를 탐색할 수 있습니다.

"CTG 서버 - EPI 요청" 대시보드에는 다음과 같은 메트릭이 표시됩니다.

- 실행 성능
  - 트랜잭션 실행 시간
  - 트랜잭션 시작 수 집계
- 읽기/쓰기 개체
  - 쓰기/읽기 개체 오버헤드 시간
  - 쓰기/읽기 개체 오버헤드 수

- CICS CP 요청
  - CP 요청 실행 시간
  - CP 요청 실행 수

CTG 가 CICS 에 대한 EPI 세션을 설정할 수 없는 경우 다음 메트릭이 채워지지 않습니다.

- 트랜잭션 실행 시간
- 쓰기/읽기 개체 오버헤드 시간
- 트랜잭션 시작 수 집계
- 쓰기/읽기 개체 오버헤드 수

대신 연결 실패를 일으킬 수 있는 오류 메시지가 기록되며 이 메시지는 "오류" 탭에 표시됩니다.

## CTG 서버 - 연결 관리자 및 작업자 대시보드

CTG 서버 - 연결 관리자 및 작업자 대시보드는 문제를 일으킬 수 있는 연결 관리자 및 작업자의 성능에 대한 간략한 스냅샷을 제공합니다. 위쪽 구석에 있는 링크를 사용하면 CTG 서버 - 개요 대시보드를 탐색할 수 있습니다.

이 대시보드에는 다음과 같은 메트릭이 표시됩니다.

- 연결 관리자 성능
  - 발송 응답 시간
  - 회신 전송 응답 시간
  - 발송 수
  - 회신 전송 수
- 작업자 스레드 성능
  - 발송 응답 시간
  - 실행/종료 응답 시간
  - 작업자 발송 수
  - 실행/종료 수

## CTG Server - SSL(CTG 서버 - SSL) 대시보드

CTG Server - SSL(CTG 서버 - SSL) 대시보드는 문제를 일으킬 수 있는 시스템 SSL 및 JSSE SSL 의 성능에 대한 간략한 스냅샷을 제공합니다. 위쪽 구석에 있는 링크를 사용하면 CTG 서버 - 개요 대시보드를 탐색할 수 있습니다.

CTG Server - SSL(CTG 서버 - SSL) 대시보드에는 다음과 같은 메트릭이 표시됩니다.

- JSSE/SSL Server Sockets(JSSE/SSL 서버 소켓)
  - Socket Accept/Close Response Time(소켓 수락/종료 응답 시간)
  - Socket Accept/Close Count(소켓 수락/종료 수)
- SSL Sockets (Legacy)(SSL 소켓(레거시))
  - Socket Close Response Time(소켓 종료 응답 시간)
  - Socket Close Count(소켓 종료 수)
- JSSE Socket(JSSE 소켓)
  - JSSE Socket Close Response Time(JSSE 소켓 종료 응답 시간)
  - JSSE Socket Close Count(JSSE 소켓 종료 수)

## CTG 클라이언트 - 개요 대시보드

CTG 클라이언트 - 개요 대시보드에는 Introscope 로 모니터링되는 세 가지 주요 CTG 서버 영역에 대한 요약이 표시됩니다. 또한 이러한 세 가지 영역에 대한 세부 정보를 제공하는 대시보드로 연결되는 링크도 표시됩니다. 신호등은 기반 메트릭 중에 정의된 임계값을 초과한 메트릭이 있는지 여부를 시각적으로 나타냅니다.

CTG 클라이언트 - 개요 대시보드에는 다음과 같은 메트릭이 표시됩니다.

- Java 게이트웨이 및 SSL 세션
  - 게이트웨이 흐름 응답 시간
  - JSSE SSL Session Allocate Response Time(JSSE SSL 세션 할당 응답 시간)
- EPI
  - EPI 연결 작업 응답 시간
  - EPI 터미널 트랜잭션 작업 응답 시간

- 터미널 및 터미널 요청
  - Terminal & Terminal Request Send Response Time(터미널 및 터미널 요청 보내기 응답 시간)
  - 터미널 연결 및 연결 끊기 응답 시간

## CTG 클라이언트 - Java 게이트웨이 및 SSL 세션 대시보드

CTG 클라이언트 - Java 게이트웨이 및 SSL 세션 대시보드는 문제를 일으킬 수 있는 게이트웨이 및 세션에 대한 간략한 스냅샷을 제공합니다. 위쪽 구석에 있는 링크를 사용하면 CTG 클라이언트 - 개요 대시보드를 탐색할 수 있습니다.

CTG 클라이언트 - Java 게이트웨이 및 SSL 세션 대시보드에는 다음과 같은 메트릭이 표시됩니다.

- Java 게이트웨이 성능
  - 흐름 응답 시간
  - Currently Active Sessions Count(현재 활성 세션 수)
  - TCP 집계 세션 호출 수
  - 열기 응답 시간
  - 종료 응답 시간
- JSSE SSL 세션 성능
  - JSSE 집계 SSL 세션 수
  - JSSE SSL 세션 할당 시간
  - 간격당 JSSE SSL 세션 호출 수

## CTG 클라이언트 - EPI 요청 대시보드

CTG 클라이언트 - EPI 요청 대시보드는 게이트웨이 연결에 대한 스냅샷을 표시합니다.

이 대시보드에는 다음과 같은 메트릭이 표시됩니다.

- 게이트웨이 응답 시간 메트릭:
  - EPI 게이트웨이 연결 작업 응답 시간
  - EPI 게이트웨이 집계 연결 수
- 터미널 및 기본 화면 처리기 메트릭:
  - EPI 터미널 연결 작업 응답 시간
  - EPI 터미널 트랜잭션 작업 응답 시간
  - EPI Basic Screen Handler Connection Count(EPI 기본 화면 처리기 연결 수)

## CTG 클라이언트 - EPI 터미널 요청 대시보드

CTG 클라이언트 - EPI 터미널 요청 대시보드는 문제를 일으킬 수 있는 터미널 및 터미널 요청 동작에 대한 간략한 스냅샷을 제공합니다. 위쪽 구석에 있는 링크를 사용하면 CTG 클라이언트 - EPI 터미널 요청 대시보드를 탐색할 수 있습니다.

이 대시보드에는 다음과 같은 메트릭이 표시됩니다.

- 터미널 작업 및 성능
- 보내기 응답 시간
- 연결 및 연결 끊기 시간
- 전송 수
- 연결 수

## Introscope ChangeDetector 대시보드

Introscope ChangeDetector 대시보드는 CTG 구성 파일 및 .JAR 파일의 문제를 일으킬 수 있는 변경 사항에 대한 요약を提供합니다.

Introscope ChangeDetector 대시보드에는 다음과 같은 메트릭이 표시됩니다.

- 간격당 발견된 변경 수
- 간격당 발견된 가장 많은 변경 수

- 마지막 에이전트 재시작 후 총 변경 수
- 마지막 에이전트 재시작 후 완료된 검색 수

## Introscope 주의/위험 경고 임계값 수정

많은 CTG 용 확장 대시보드에 경고가 포함되어 있습니다.

### Introscope 주의/위험 경고 임계값을 수정하려면

1. Introscope Investigator 에서 관리 모듈 > CTG 클라이언트 > 경고로 이동한 다음 경고를 선택합니다.  
경고 정의는 아래쪽 창에 있습니다.
2. 경고 정의 창에서 기간 필드를 수정하여 주의 임계값 및 위험 임계값과 성능 메트릭을 비교할 빈도를 지정합니다.
3. 조직의 서비스 수준에 적합한 주의 임계값과 위험 임계값을 지정합니다.
4. "적용"을 클릭하여 변경 사항을 적용합니다.

## CTG 트랜잭션 추적 프로그램

CTG 트랜잭션 추적 프로그램은 서버 이름 및 프로그램 이름을 기준으로 추적하는 기능을 제공합니다. 이 기능은 기본적으로 CTG 클라이언트(예: Websphere)와 CTG 서버 간의 트랜잭션을 추적하기 위한 것입니다. CICS 메인프레임의 제품 간 TranTracer SYSVIEW 기능과 연결할 필요는 없습니다. 또한 이 추적 프로그램은 공용 영역에서 고객이 지정한 사용자 상관 관계 필드를 추출하여 포함하고, 응답 시간 임계값 조정을 수행하고, 특정 프로그램 이름을 추적하는 기능을 지원합니다. 이 기능을 사용하여 에이전트 속성 항목의 사용 방법을 사용자 지정하고, 프로그램 이름을 사용하거나 상위 수준에 사용자 고유의 사용자 지정 확장을 작성하여 응답 시간 임계값을 기반으로 하는 고급 CTG 추적 기능을 생성할 수 있습니다.

## PP CTG Transaction Tracer(PP CTG 트랜잭션 추적 프로그램)의 특수 문자 처리

Introscope Workstation 에서 실행되는 Introscope 트랜잭션 추적 상관 관계 논리는 사용자가 생성한 상관 관계 ID 에서 특수 문자의 사용을 엄격하게 제한합니다. Introscope Workstation 의 상관 관계 지원에서 허용되지 않는 모든 특수 문자는 자동으로 밑줄로 변환됩니다.

다음 특수 문자는 항상 밑줄로 변환됩니다.

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

다음 추가 특수 문자 집합도 기본적으로 밑줄로 변환됩니다.

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

이러한 특수 문자를 그대로 전달하려면(그리고 모든 특수 문자와 추적을 허용하려면) `IntroscopeAgent.profile` 에 다음 속성을 추가하십시오.

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

이 값의 기본값은 'true'이며, 이 값은 모든 특수 문자를 밑줄로 변환하고 프런트엔드 및 백엔드 추적을 서로 연관시킵니다.

상관 관계 ID 에서 중복되는 밑줄을 제거하려면(여러 개의 밑줄을 밑줄 한 개로 줄이려면) `IntroscopeAgent.profile` 에서 다음 속성을 설정해야 합니다.

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

## CTG 트랜잭션 추적 프로그램 속성의 자세한 목록

`IntroscopeAgent.profile` 파일에서 구성할 수 있는 CTG 트랜잭션 추적 프로그램 속성의 자세한 목록이 표에 나와 있습니다.

속성	용도
<code>introscope.ctg.trantracer.publish.metrics</code>	단순 메트릭("Response Time"(응답 시간) 및 "Invocation Count"(호출 수))을 사용할지 여부를 나타내는 부울입니다. '경량' 추적 프로그램이라는 점에 주의하십시오. 기본값 = true
<code>introscope.ctg.trantracer.publish.trantrace</code>	트랜잭션 추적이 사용되는지 여부를 나타내는 부울입니다. 기본값 = true
<code>introscope.ctg.trantracer.corridscan.offset</code>	공용 영역 시작 위치를 기준으로 사용자 상관 관계 ID 가 시작되는 오프셋을 나타내는 정수입니다. 기본값 = 147

속성	용도
introscope.ctg.trantracer.corridscan.length	추출할 사용자 상관 관계 ID 의 길이를 나타내는 정수입니다. 기본값 = 100
introscope.ctg.trantracer.corrid.format.id	사용자 상관 관계 ID 에서 ':'(콜론), '-'(대시)와 같이 EM TranTracer 문자열 항목에 허용되지 않는 잘못된 문자를 검색할지 여부를 나타내는 부울입니다.
introscope.ctg.trantracer.corridscan.starttrace	사용자 상관 관계 ID 를 "Finish Trace"(추적 종료) 대신 "Start Trace"(추적 시작)에서 추출할지 여부를 나타내는 부울입니다. 기본값 = false
introscope.ctg.trantracer.rspthreshhold.value	이 추적 프로그램에 허용되는 응답 시간 임계값의 최대값을 나타내는 정수입니다. 이 값을 초과하는 응답 시간은 EM 으로 추적됩니다. 이 값보다 작은 값은 추적되지 않습니다. 기본값 = 0 = RSP 임계값을 확인하지 않음
introscope.ctg.trantracer.program.name	일치하는지 확인할 CICS 프로그램 이름의 전체 또는 일부를 나타내는 문자열입니다. 일치하는 값은 EM 으로 추적됩니다. 일치하지 않는 값은 추적되지 않습니다. 기본값 = null = 프로그램 일치 사용 안 함

속성	용도
introscope.ctg.trantracer.program.match.criteria	<p>어떠한 종류의 프로그램 이름 일치를 수행할지 정의하는 정수입니다.</p> <p>1 = 전체 이름 일치  2 = 위의 name(이름) 문자열로 시작하는 프로그램 이름  3 = 위의 name(이름) 문자열로 끝나는 프로그램 이름  0 = 일치 사용 안 함  기본값 = 0(일치 사용 안 함)</p>
introscope.ctg.trantracer.metric.HLQname	<p>Investigator 트리의 상위 수준 이름이 무엇인지 나타내는 문자열입니다.</p> <p>기본값:  백엔드 CTG_to_CICS_ECI_IPIC_Trace  및 Frontends Client_to_CTG_Trace</p>
introscope.ctg.trantracer.debug	<p>에이전트 로그에 대한 디버그 메시지를 추적할지 여부를 나타내는 부울입니다.</p> <p>기본값 = false</p>



# 부록 A: CTG 성능 메트릭

---

이 부록에서는 CTG 용 확장이 성능 메트릭을 수집하는 대상이고 Introscope Investigator 에 나타나는 CTG 클래스 및 메서드에 대해 설명합니다.

이 섹션은 다음 항목을 포함하고 있습니다.

[프런트엔드 메트릭](#) (페이지 67)

[백엔드 메트릭](#) (페이지 70)

[CTG 대시보드 메트릭](#) (페이지 73)

[요청 종료 메트릭](#) (페이지 79)

## 프런트엔드 메트릭

프런트엔드 메트릭은 CTG 클라이언트 계측으로 캡처됩니다. 데이터는 WebSphere 와 같은 응용 프로그램 서버 내에서 캡처되거나, WebSphere 외부에서 독립형으로 실행되고 있는 계측된 사용자 작성 CTG 클라이언트 응용 프로그램 내에서 캡처됩니다. 프런트엔드 메트릭은 실행 중인 CTG 응용 프로그램 유형(JCA, CCF) 또는 ECI 나 EPI 의 기본 클래스를 캡처하고 측정하는데 사용됩니다.

메트릭은 집계 수준(예: Websphere 응용 프로그램 서버에 의해 호출되는 모든 ECI 프로그램)에서 캡처됩니다. 세부 메트릭은 개별 프로그램 또는 트랜잭션 수준(예: 호출된 개별 ECI 프로그램에 대한 통계)에서 캡처됩니다. 많은 메트릭(특히 세부 메트릭)이 BlamePoint 추적 프로그램을 사용하므로 공통 형식이 제공됩니다.

메트릭	정의
요청 수 집계	<p>프로그램 또는 트랜잭션에 대한 요청이 시작된 횟수의 합계입니다. 연결 열기가 실패하는 경우(아래 참조) 이 메트릭이 나타나지 않을 수 있습니다.</p> <p><b>참고:</b> "Aggregate Service Count(서비스 수 집계)"의 "참고"를 참조하십시오.</p>
Aggregate Service Count(서비스 수 집계)	<p>EPI에서는 실제 CICS 트랜잭션을 실행하지 않는 몇 가지 설정 및 해제 흐름(AddTerminal, PurgeTerminal, DeleteTerminal)을 사용합니다. 이러한 흐름은 CICS에 대한 활성 3270 연결을 설정하는 서비스 요청일 뿐입니다. 이 메트릭은 이러한 서비스 유형의 모든 요청에 대한 집계 수를 유지합니다.</p> <p><b>참고:</b> "요청 수 집계"는 실제 3270 트랜잭션 호출 및 응답(EPI 요청을 통한 3270 버퍼의 전송 및 CICS 트랜잭션의 관련 3270 회신 버퍼)을 전송합니다.</p>
Aggregate Errors Count(오류 수 집계)	<p>연결 열기에 대한 요청 또는 프로그램이나 트랜잭션 시작에 대한 요청이 오류로 인해 실패한 횟수의 합계입니다. 이 메트릭은 실제 오류가 발생한 경우에만 나타납니다.</p>
평균 응답 시간	<p>요청이 처리되는 평균 응답 시간이며 클라이언트 쪽에서 측정합니다(전체 왕복 시간).</p>
동시 호출	<p>동시에 호출된 프로그램 또는 트랜잭션 요청의 수를 나타냅니다.</p>
간격당 오류 수	<p>가장 최근의 측정 간격(일반적으로 7-15 초) 중에 보고된 오류의 합계입니다.</p>

메트릭	정의
간격당 응답 수	가장 최근의 측정 간격 중에 보고된 CTG 로부터의 회신의 합계입니다.
중단 수	프로그램에 대한 요청이 시작되었을 때 중단이 감지되었지만 중단 감지 시간 제한 내에 CICS 에서 응답하지 않은 경우를 나타냅니다. 중단 감지 제한은 <i>IntroscopeAgent.profile</i> 의 <i>introscope.agent.stalls.thresholdseconds=nn</i> 속성을 사용하여 설정합니다. <i>nn</i> 값은 중단 감지 시간을 초 단위로 지정합니다.

Client\_to\_CTG\_Aggregates 의 경우 다음과 같은 추가 메트릭이 추적됩니다.

메트릭	정의
Aggregate Opens(열기 집계)	CTG 클라이언트에서 CTG 서버에 대해 실행된 TCP 또는 SSL 연결 열기의 합계입니다. CTG 연결 열기가 수행되기 전에는 CTG 요청을 시작할 수 없습니다.
열기 응답 시간	클라이언트 측에서 연결 열기 요청이 처리되고 측정되는 평균 응답 시간입니다. 다시 말하면 CTG 클라이언트에서 CTG 서버로의 연결이 열리는 데 걸린 총 왕복 시간입니다.

## 프런트엔드 클라이언트 - CTG 집계

이러한 메트릭은 CTG 클라이언트에서 CTG 서버에 대해 실행된 모든 ECI 및 EPI 호출의 집계 수를 제공합니다. 편의를 위해 호출은 CTG API 유형별로 분류됩니다. 다음 표에 있는 모든 메트릭은 BlamePoint 메트릭을 사용합니다.

연결 열기 요청이 실패하는 경우 Client\_to\_CTG\_Aggregates 트리 아래에 나타나는 "Open Errors"(열기 오류) 탭에서 세부 정보를 찾을 수 있습니다.

메트릭	정의
BASE_ECI_EPI	기본 ECI 클래스 API 를 사용하는 모든 ECI 호출과 기본 EPI 클래스 API 를 사용하는 모든 EPI 호출
JCA_ECI	최신 JCA(Java 커넥터 아키텍처) ECI API 를 사용하는 모든 ECI 호출
JCA_EPI	최신 JCA(Java 커넥터 아키텍처) EPI API 를 사용하는 모든 EPI 호출
중단 수	프로그램에 대한 요청이 시작되었을 때 중단이 감지되었지만 중단 감지 시간 제한 내에 CICS 에서 응답하지 않은 경우를 나타냅니다. 중단 감지 제한은 IntroscopeAgent.profile 의 <i>introscope.agent.stalls.thresholdseconds=nn</i> 속성을 사용하여 설정합니다. nn 값은 중단 감지 시간을 초 단위로 지정합니다.

## 프런트엔드 클라이언트 - CTG 세부 정보

이러한 메트릭은 CTG 클라이언트에 의해 호출된 각 프로그램 또는 트랜잭션의 수에 대한 세부 정보를 제공합니다. 편의를 위해 호출은 CLI(게이트웨이) 서버별로 분류됩니다. 각 게이트웨이 서버 아래에는 호출된 각 개별 ECI 또는 EPI 프로그램의 목록과 해당 프로그램의 세부 메트릭이 나와 있습니다.

## 백엔드 메트릭

백엔드 메트릭은 Introscope 로 계측된 CTG 서버 내에서 캡처됩니다. 백엔드 메트릭은 클라이언트와 CICS 시스템 사이에서 ECI 및 EPI 요청 흐름을 캡처하는 데 사용됩니다.

메트릭은 집계 수준(예: 클라이언트가 호출하는 모든 ECI 프로그램)에서 캡처됩니다. 세부 메트릭은 개별 프로그램 수준(예: 호출된 개별 ECI 프로그램 통계)에서 캡처됩니다.

각 게이트웨이 서버 아래에는 호출된 각 개별 ECI 또는 EPI 프로그램의 목록과 해당 프로그램의 세부 메트릭이 표시됩니다. 세부 메트릭에서는 공통 형식의 BlamePoint 추적 프로그램을 사용합니다.

메트릭	정의
집계 수 요청 수 집계	프로그램 또는 트랜잭션에 대한 요청이 시작된 횟수의 합계입니다. 연결 열기가 실패하는 경우(아래 참조) 이 메트릭이 나타나지 않을 수 있습니다.
Aggregate Errors Count(오류 수 집계)	연결 열기에 대한 요청 또는 프로그램이나 트랜잭션 시작에 대한 요청이 오류로 인해 실패한 횟수의 합계입니다. 이 메트릭은 실제 오류가 발생한 경우에만 나타납니다.
평균 응답 시간	요청이 처리되는 평균 총 왕복 시간입니다.
간격당 오류 수	가장 최근의 측정 간격 중에 보고된 오류의 합계입니다.
간격당 응답 수	가장 최근의 측정 간격 중에 보고된 CTG 로부터의 회신의 합계입니다.
중단 수	감지된 중단 수의 합계입니다. 프로그램에 대한 요청이 시작되었지만 CICS 가 중단 감지 시간 제한 내에 응답하지 않은 경우 중단이 발생합니다. 중단 시간 제한은 IntroscopeAgent.profile 의 introscope.agent.stalls.thresholdseconds=nn 속성으로 설정됩니다. 여기서 nn 은 중단 감지 시간(초)입니다.

## 백엔드 CTG\_to\_CICS\_ECI\_IPIC 집계

이러한 메트릭은 CTG 클라이언트와 해당하는 CICS 서버 사이에서 실행된 모든 ECI/IPIC 호출의 집계 수를 제공합니다. 각 게이트웨이 서버 아래에는 표에 설명된 대로 호출된 각 개별 ECI/IPIC 프로그램의 목록이 BlamePoint 추적 프로그램 메트릭을 사용하여 표시됩니다. 다음 집계 메트릭은 게이트웨이 서버에서 게이트웨이 서버별로 제공됩니다.

메트릭	정의
Program Aggregate Count(프로그램 집계 수)	ECI/IPIC 프로그램에 대한 요청을 시작한 횟수의 총 집계입니다.
프로그램 평균 응답 시간	CTG 서버에서 CICS 까지 ECI/IPIC 요청이 처리되고 측정되는 평균 응답 시간 집계입니다. 즉, CTG 서버와 CICS 사이의 총 왕복 시간입니다.
간격당 프로그램 호출 수	가장 최근의 측정 간격 중에 보고된 ECI/IPIC 프로그램에 대한 호출의 합계입니다.

## 백엔드 CTG 전체 통계

메트릭	정의
클라이언트 요청 처리 응답 시간	클라이언트 요청을 읽고 CICS 까지 요청을 전달하는 데 필요한 전체 시간입니다.
클라이언트 회신 보내기 응답 시간	CICS 의 회신을 읽고 요청한 클라이언트로 다시 전달하는 데 필요한 전체 시간입니다.
사용 중인 연결 관리자 스레드	클라이언트 연결을 처리하기 위해 현재 사용 중인 스레드 수입니다.
사용 중인 작업자 스레드	CICS 요청을 처리하기 위해 현재 사용 중인 스레드 수입니다.

## 백엔드 CTG\_to\_CICS\_EPI 집계 메트릭

CTG\_to\_CICS\_EPI 메트릭은 z/OS 가 아닌 분산 버전의 CTG 에서만 사용할 수 있습니다. 이러한 메트릭은 CTG 클라이언트와 해당 CICS 서버 간에 실행된 EPI 호출의 집계 수를 제공합니다.

메트릭	정의
Transaction Aggregate Count(트랜잭션 집계 수)	EPI 트랜잭션에 대한 요청을 시작한 횟수의 집계입니다.
트랜잭션 평균 응답 시간	CTG 서버에서 CICS 까지 EPI 요청이 처리되고 측정되는 평균 응답 시간 집계입니다. 즉, CTG 서버와 CICS 사이의 총 왕복 시간입니다.
간격당 트랜잭션 호출 수	최근 측정 간격 중에 보고된 EPI 트랜잭션에 대한 호출의 합계입니다.

**참고:** 각 게이트웨이 서버 아래에는 표에 설명된 대로 호출된 각 개별 EPI 트랜잭션의 목록이 BlamePoint 추적 프로그램 메트릭을 사용하여 표시됩니다.

## 백엔드 CTG\_to\_CICS 스레드

이러한 메트릭은 CTG 서버의 연결 및 작업자 스레드에 대한 현재 사용량을 보여 줍니다.

## CTG 대시보드 메트릭

다음 표에는 확장 대시보드에 표시되는 레이블과 특정 CTG 성능 메트릭 간의 대응 관계가 나와 있습니다.

오른쪽 열에는 Introscope Investigator 에서의 메트릭 위치가 표시되어 있습니다.

## CTG 클라이언트 - 개요 대시보드

레이블	CTG 성능 메트릭
게이트웨이 흐름 응답 시간	CTGCLIENT {.+}JavaGateway:open Response Time (ms)
JSSE SSL 세션 할당 응답 시간(ms)	Client_to_CTG JSSE Session:SSL Handshake Allocate Response Time
EPI 연결 작업 응답 시간	CTGCLIENT EPI Gateway: open Response Time (ms) CTGCLIENT EPI Terminal:{connect disconnect process Connect processDisconnect terminate} Response Time (ms)
EPI 터미널 트랜잭션 응답 시간	CTGCLIENT EPI Terminal:{send setTransactionData startTran terminate} Response Time (ms)
Terminal & Terminal Request Send Response Time(터미널 및 터미널 요청 보내기 응답 시간)	CTGCLIENT {TerminalTerminalRequest}:send Response Time (ms)
터미널 연결 및 연결 끊기 응답 시간	CTGCLIENT Terminal{connect disconnect} Response Time (ms)

## CTG 클라이언트 - EPI 대시보드

레이블	CTG 성능 메트릭
EPI 게이트웨이 연결 작업 응답 시간	CTGCLIENT EPI Gateway:{.*} Response Time (ms)
EPI 게이트웨이 집계 연결 수	CTGCLIENT EPI Gateway:{.*} Count
EPI 터미널 연결 작업 응답 시간	CTGCLIENT EPI Terminal:{connect diswconnect processConnect processDisconnect terminate} Response Time (ms)

레이블	CTG 성능 메트릭
EPI 터미널 트랜잭션 작업 응답 시간	CTGCLIENT EPITerminal{send setTransactionData startTran terminate} Response Time (ms)
EPIMonitor & Terminal Connection Count(EPI 모니터 및 터미널 연결 수)	CTGCLIENT EPITerminal:Count CTGCLIENT EPIMonitor:terminalConnectedCount
EPI BasicScreenHandler Connection Count(EPI 기본 화면 처리기 연결 수)	CTGCLIENT EPIBasicScreenHandler: terminalConnectedCount

## CTG 클라이언트 - Java 게이트웨이 및 SSL 대시보드

레이블	CTG 성능 메트릭
Java 게이트웨이: 흐름 응답 시간	CTGCLIENT {.+}JavaGateway:flow Response Time (ms)
JavaGateways: Total Active Sessions Count(Java 게이트웨이: 활성 세션 수 합계)	CTGCLIENT {.+}JavaGateway:Count
Java 게이트웨이: 열기 응답 시간	CTGCLIENT {.+}JavaGateway:open Response Time (ms)
Java 게이트웨이: 종료 응답 시간	CTGCLIENT {.+}JavaGateway:close Response Time (ms)
JSSE 집계 SSL 세션 수	Client_to_CTG JSSE Session:Aggregate SSL Opens
JSSE SSL 세션 할당 시간(ms)	Client_to_CTG JSSE Session:SSL Handshake Allocate Response Time
간격당 JSSE SSL 세션 호출 수	Client_to_CTG JSSE Session:SSL Opens Per Interval

## CTG 클라이언트 - 터미널 및 터미널 요청 대시보드

레이블	CTG 성능 메트릭
Terminal Send Count(터미널 보내기 수)	CTGCLIENT EPIGateway:{.*}Response Time (ms)
터미널 연결 및 연결 끊기 수	CTGCLIENT EPIGateway:{.*} Count
터미널 요청 보내기 응답 시간	CTGCLIENT TerminalRequest:send Response Time (ms)
연결 및 연결 끊기 응답 시간	CTGCLIENT TerminalRequest:{release allocate} Response time (ms)

## CTG 서버 - 개요 대시보드

레이블	CTG 서버 성능 메트릭
ECI 요청 실행 응답 시간	CTGSERVER ServerECIRequest execute Response Time (ms)
ECI 요청 수	CTGSERVER ServerECIRequest execute Count
EPI 요청 실행 응답 시간	CTGSERVER ServerEPIRequest execute Response Time (ms)
EPI 요청 수	CTGSERVER ServerEPIRequest execute Count

## CTG 서버 - 연결 관리자 및 작업자 대시보드

레이블	WebSphere 성능 메트릭
연결 관리자: 발송 응답 시간	CTGSERVER ConnectionManager:kick Response Time (ms)
연결 관리자: 발송 수	CTGSERVER ConnectionManager:kick Count
연결 관리자: 회신 전송 응답 시간	CTGSERVER ConnectionManager:sendReply Response Time (ms)

레이블	WebSphere 성능 메트릭
연결 관리자: 회신 전송 수	CTGSERVER ConnectionManager sendReply Count
작업자: 발송 응답 시간	CTGSERVER Worker:kick Response Time (ms)
작업자: 발송 수	CTGSERVER Worker:kick Count
작업자: 실행/종료 응답 시간	CTGSERVER Worker:{close run} Response Time (ms)
작업자: 실행/종료 수	CTGSERVER Worker:{close run} Count

## CTG 서버 - EPI 요청 대시보드

레이블	WebSphere 성능 메트릭
요청 실행 시간	CTGSERVER ServerECIRequest:execute Response Time (ms)
실행 목록 시간	CTGSERVER ServerECIRequest:executeList Response Time (ms)
요청 실행 수	CTGSERVER ServerECIRequest:execute Count
실행 목록 수	CTGSERVER ServerECIRequest:executeList Count
Request Initialize/Terminate time(요청 초기화/종료 시간)	CTGSERVER ServerECIRequest:{initialize terminate} Response Time (ms)
쓰기/읽기 개체 시간	CTGSERVER ServerECIRequest:{read write}Object Response Time (ms)
Request Initialize/Terminate Count(요청 초기화/종료 수)	CTGSERVER ServerECIRequest:{initialize terminate} Count
쓰기/읽기 개체 수	CTGSERVER ServerECIRequest:{read write}Object Count

## CTG 서버 - EPI 요청 대시보드

레이블	WebSphere 성능 메트릭
요청 실행 시간	CTGSERVER ServerEPIRequest:execute Response Time (ms)
실행 목록 시간	CTGSERVER ServerEPIRequest:executeList Response Time (ms)
요청 실행 수	CTGSERVER ServerEPIRequest:execute Count
실행 목록 수	CTGSERVER ServerEPIRequest:executeList Count
Request Initialize/Terminate time(요청 초기화/종료 시간)	CTGSERVER ServerEPIRequest:{initialize terminate} Response Time (ms)
쓰기/읽기 개체 시간	CTGSERVER ServerEPIRequest:{read write}Object Response Time (ms)
CicsCp 요청 실행 응답 시간	CTGSERVER ServerCicsCpRequest execute Response Time (ms)
CicsCp 요청 수	CTGSERVER ServerCicsCpRequest execute Count
Request Initialize/Terminate Count(요청 초기화/종료 수)	CTGSERVER ServerEPIRequest:{initialize terminate} Count
쓰기/읽기 개체 수	CTGSERVER ServerEPIRequest:{read write}Object Count

## CTG 서버 - 전체 통계 대시보드

레이블	CTG 서버 성능 메트릭
처리된 I/O 요청(집계)	처리된 TCP I/O 요청 수의 합계입니다.
간격당 처리된 I/O 요청	지난 15 초 간격 동안 처리된 TCP I/O 요청의 수입니다.

레이블	CTG 서버 성능 메트릭
클라이언트에서 들어오는 연결 요청(간격당)	지난 15 초 간격 동안 발생하여 클라이언트로부터 수신된 새 요청의 수입입니다.
CICS 에 대한 ECI/IPIC/EPI 요청 합계(간격당)	지난 15 초 간격 동안 처리된 총 요청(ECI, IPIC 및 EPI) 수입입니다.
현재 사용 중인 연결 관리자 스레드	클라이언트 연결을 처리하기 위해 현재 사용 중인 스레드 수입입니다.
연결 관리자 스레드 풀 수	클라이언트 연결을 처리하기 위해 할당된 현재 스레드 수입입니다.
현재 사용 중인 작업자 스레드	CICS 요청을 처리하기 위해 현재 사용 중인 스레드 수입입니다.
작업자 스레드 풀 수	CICS 요청을 처리하기 위해 할당된 현재 스레드 수입입니다.

## CTG 서버 - SSL 대시보드

레이블	WebSphere 성능 메트릭
JSSE SSL 집계 들어오는 SSL 핸드셰이크	들어오는 SSL 핸드셰이크 수의 집계입니다.
간격당 JSSE SSL 들어오는 SSL 핸드셰이크	간격당 SSL 핸드셰이크 수입입니다.

## 요청 종료 메트릭

다음 표에는 요청 종료 기능에 대한 메트릭이 나와 있습니다.

## 백엔드 | CTG\_to\_CTG\_ECI\_IPIC\_RequestExit 메트릭

이 표에는 Investigator 노드 아래에 나타나는 백엔드 | CTG\_to\_CICS\_ECI\_IPIC\_RequestExit 에 대한 메트릭이 자세히 나와 있습니다.

자세한 메트릭 이름	설명
프로그램 평균 응답 시간(usec)	Average response time for all programs(모든 프로그램의 평균 응답 시간)
간격당 프로그램 호출 수	모든 프로그램의 간격당 호출 수 합계
Program Aggregate Count ECI_IPIC(프로그램 집계 수 ECI_IPIC)	프로그램이 호출된 횟수의 집계
Program Aggregate Errors(프로그램 오류 집계)	모든 프로그램의 오류 집계
{server_name program_name}:평균 응답 시간(usec)	프로그램의 평균 응답 시간
{server_name program_name}:간격당 응답 수	프로그램의 간격당 응답(호출) 수
{server_name program_name}:동시 호출	지정된(15 초) 간격 동안 프로그램의 동시 호출 수
{server_name program_name}:중단 수	지정된(15 초) 간격 동안 프로그램에 대해 기록된 중단/멈춤 수
{server_name program_name}:간격당 오류 수	지정된(15 초) 간격 동안 프로그램에 대해 기록된 오류 수
{server_name program_name}:Aggregate Program Count ECI_IPIC(프로그램 수 집계 ECI_IPIC)	프로그램이 호출된 횟수의 집계
{server_name program_name}:오류 집계	프로그램이 호출될 때 오류가 보고된 횟수의 집계

자세한 메트릭 이름	설명
{server_name program_name}:공용 영역 집계 요청 데이터	이 프로그램에 대해 CICS 로 전송된 공용 영역 데이터 바이트 양의 집계
{server_name program_name}:공용 영역	이 프로그램에 대해 CICS 로부터 수신된 공용 영역 데이터 바이트 양의 집계

## CTG\_Global\_Statistics\_RequestExit 메트릭

이 표에는 CTG\_Global\_Statistics\_RequestExit 에 대한 메트릭이 자세히 나와 있습니다. 메트릭의 세부 정보 이름은 IBM 에서 메트릭에 할당한 이름에 해당합니다. 이 표를 참조하면 Introscope 의 메트릭 이름과 IBM 의 메트릭 정의 간의 혼동을 최소화할 수 있습니다.

자세한 메트릭 이름	설명
CICS Server:Amount of CICS request data(CICS 서버:CICS 요청 데이터의 양)	연결된 CICS 서버로 전송된 요청 데이터의 양(바이트). 이 양에는 응용 프로그램 및 CICS 프로토콜 데이터가 모두 포함됩니다.
CICS Server:Amount of CICS response data(CICS 서버:CICS 응답 데이터의 양)	연결된 CICS 서버로부터 수신된 응답 데이터의 양(바이트). 이 양에는 응용 프로그램 및 CICS 프로토콜 데이터가 모두 포함됩니다.
CICS Server:Current number of installed terminals(CICS 서버:현재 설치된 터미널 수)	현재 설정되고 설치된(EPI) 터미널 세션의 수
CICS Server:Current number of Orphaned CICS requests(CICS 서버:현재 고립된 CICS 요청의 수)	소유 응용 프로그램이 시간 초과되거나 종료된 CICS 로부터의 응답을 기다리는 현재 요청의 수
CICS Server:Maximum number of active requests(CICS 서버:최대 활성 요청 수)	구성 파일에 정의된 활성 요청에 대하여 정의된 최대 개수

자세한 메트릭 이름	설명
CICS Server:Amount of CICS request data(CICS 서버:CICS 요청 데이터의 양)	연결된 CICS 서버로 전송된 요청 데이터의 양(바이트). 이 양에는 응용 프로그램 및 CICS 프로토콜 데이터가 모두 포함됩니다.
CICS Server:Number of CICS servers(CICS 서버:CICS 서버 수)	게이트웨이 데몬이 요청 전송을 시도한 CICS 서버 수
CICS Server:Number of IPIC session failures(CICS 서버:IPIC 세션 실패 수)	CICS 서버에 대한 IPIC 세션의 실패 수
CICS Server:Number of IPIC session in use(CICS 서버:사용 중인 IPIC 세션 수)	CICS 서버에서 사용 중인 IPIC 세션 수
CICS Server:Number of active requests(CICS 서버:활성 요청 수)	클라이언트 데몬에서 실행 중인 활성 요청의 현재 수
CICS Server:Number of connect failures(CICS 서버:연결 실패 수)	CICS 서버에 연결하려는 시도가 실패한 횟수
CICS Server:Number of defined CICS servers(CICS 서버:정의된 CICS 서버 수)	구성 파일에 정의되어 있는 CICS 서버 수
CICS Server:Number of lost connections(CICS 서버:끊어진 연결 수)	CICS 서버와 설정된 연결이 끊어진 횟수
CICS Server:Number of negotiated IPIC sessions(CICS 서버:협상된 IPIC 세션 수)	CICS 서버와 협상된 IPIC 세션 수
CICS Server:Number of orphaned CICS requests(CICS 서버:고립된 CICS 요청 수)	응답 대기 중이거나 시간 초과되거나 종료된 CICS 서버 요청 수

자세한 메트릭 이름	설명
CICS Server:Amount of CICS request data(CICS 서버:CICS 요청 데이터의 양)	연결된 CICS 서버로 전송된 요청 데이터의 양(바이트). 이 양에는 응용 프로그램 및 CICS 프로토콜 데이터가 모두 포함됩니다.
CICS Server:Number of requests processed(CICS 서버:처리된 요청 수)	처리된 CICS 서버 요청 수
CICS Server:Number of requests waiting on a response(CICS 서버:응답 대기 중인 요청 수)	CICS 서버의 응답을 현재 대기 중인 요청 수
CICS Server:Number of terminal install requests(CICS 서버:터미널 설치 요청 수)	CICS 서버로 전송된 터미널 설치 요청 수
CICS Server:Number of terminal uninstall events(CICS 서버:터미널 제거 이벤트 수)	게이트웨이에서 처리된 터미널 제거 이벤트 수
CICS Server:Number of timed out connections(CICS 서버:시간 초과된 연결 수)	CICS 서버에 대한 연결이 시간 초과된 횟수
CICS Daemon:Client daemon running Time(CICS 데몬:클라이언트 데몬 실행 시간)	클라이언트 데몬이 성공적으로 초기화된 이후 경과한 시간(초)
CICS Daemon:Number of connected Client applications(CICS 데몬:연결된 클라이언트 응용 프로그램 수)	클라이언트 데몬에 연결된 클라이언트 응용 프로그램 프로세스의 현재 수

자세한 메트릭 이름	설명
CICS Server:Amount of CICS request data(CICS 서버:CICS 요청 데이터의 양)	연결된 CICS 서버로 전송된 요청 데이터의 양(바이트). 이 양에는 응용 프로그램 및 CICS 프로토콜 데이터가 모두 포함됩니다.
CICS Daemon:Number of requests Processed(CICS 데몬:처리된 요청 수)	처리된 API 호출 요청 수
Connection Manager:Pool Count(연결 관리자:풀 수)	현재 클라이언트에 할당된 연결 관리자 스레드의 최대 수
Connection Manager:Thread dispatch count(연결 관리자:스레드 발송 수)	클라이언트에서 설정된 연결 수를 나타내는 연결 관리자 스레드에 대한 할당 수
Connection Manager:Threads allocated to clients(연결 관리자:클라이언트에 할당된 스레드 수)	클라이언트에 할당된 연결 관리자 스레드의 현재 수
Gateway Daemon:Amount of client request data(게이트웨이 데몬:클라이언트 요청 데이터의 양)	클라이언트 응용 프로그램으로부터 수신된 요청 데이터의 양(바이트)
Gateway Daemon:Amount of client response data(게이트웨이 데몬:클라이언트 응답 데이터의 양)	클라이언트 응용 프로그램으로 전송된 응답 데이터의 양(바이트)
Gateway Daemon:Average Gateway daemon response time(게이트웨이 데몬:게이트웨이 데몬 평균 응답 시간)	게이트웨이 데몬이 클라이언트의 API 요청에 응답하는 데 걸린 평균 시간(밀리초)

자세한 메트릭 이름	설명
CICS Server:Amount of CICS request data(CICS 서버:CICS 요청 데이터의 양)	연결된 CICS 서버로 전송된 요청 데이터의 양(바이트). 이 양에는 응용 프로그램 및 CICS 프로토콜 데이터가 모두 포함됩니다.
Gateway Daemon:End of interval time HHMMSS(게이트웨이 데몬:간격 끝 시간 HHMMSS)	다음 번 예약된 간격 통계 재설정 이벤트의 로컬 시간. 이 시간이 되면 모든 간격 통계가 0 으로 재설정됩니다.
Gateway Daemon:Extended LUW transactions committed(게이트웨이 데몬:커밋된 확장 LUW 트랜잭션)	커밋된 확장 LUW 기반 1 단계 트랜잭션의 수
Gateway Daemon:Extended LUW transactions rolled back(게이트웨이 데몬:롤백된 확장 LUW 트랜잭션)	롤백된 확장 LUW 기반 1 단계 트랜잭션의 수
Gateway Daemon:Failed SYNCONRETURN Transactions(게이트웨이 데몬:실패한 SYNCONRETURN 트랜잭션)	현재 간격에서 실패한 SYNCONRETURN 트랜잭션의 수
Gateway Daemon:Gateway daemon running time(게이트웨이 데몬:게이트웨이 데몬 실행 시간)	게이트웨이 데몬이 성공적으로 초기화된 이후 경과한 시간(초)
Gateway Daemon:Interval running Time(게이트웨이 데몬:간격 실행 시간)	마지막 간격 재설정 이벤트 이후에 경과한 초 단위의 시간(현재 간격의 경과 시간)

자세한 메트릭 이름	설명
CICS Server:Amount of CICS request data(CICS 서버:CICS 요청 데이터의 양)	연결된 CICS 서버로 전송된 요청 데이터의 양(바이트). 이 양에는 응용 프로그램 및 CICS 프로토콜 데이터가 모두 포함됩니다.
Gateway Daemon:Length of the statistics interval HHMMSS(게이트웨이 데몬:통계 간격 길이 HHMMSS)	게이트웨이 데몬에서 사용 중인 통계 간격의 기간
Gateway Daemon:Logical End of Day time HHMMSS(게이트웨이 데몬:논리적 하루 끝 시간 HHMMSS)	게이트웨이 데몬에서 논리적 하루 끝으로 지정된 로컬 시간. 이 시간이 되면 모든 간격 통계가 0으로 재설정됩니다.
Gateway Daemon:Number of CICS request exit calls(게이트웨이 데몬:CICS 요청 종료 호출 수)	게이트웨이 데몬에서 호출된 요청 종료 호출 수
Gateway Daemon:Number of Extended LUW transactions(게이트웨이 데몬:확장 LUW 트랜잭션 수)	최근 간격에서 처리된 확장 LUW 트랜잭션 수
Gateway Daemon:Number of SYNCONRETURN transactions(게이트웨이 데몬:SYNCONRETURN 트랜잭션의 수)	최근 간격에서 처리된 SYNCONRETURN 트랜잭션 수
Gateway Daemon:Number of requests processed(게이트웨이 데몬:처리된 요청 수)	마지막 간격에서 처리된 요청 수

자세한 메트릭 이름	설명
CICS Server:Amount of CICS request data(CICS 서버:CICS 요청 데이터의 양)	연결된 CICS 서버로 전송된 요청 데이터의 양(바이트). 이 양에는 응용 프로그램 및 CICS 프로토콜 데이터가 모두 포함됩니다.
Gateway Daemon:Successful SYNCONRETURN transactions(게이트웨이 데몬:성공한 SYNCONRETURN 트랜잭션)	게이트웨이 데몬 프로세스 중에 성공한 SYNCONRETURN 트랜잭션 수
Gateway Daemon:Total Requests Processed/Second(게이트웨이 데몬:처리된 총 요청/초)	처리된 API 호출 수의 집계
Protocol Handler:SSL protocol handler port number(프로토콜 처리기:TCP 프로토콜 처리기 포트 번호)	SSL 트래픽에 사용되고 있는 TCP 포트
Protocol Handler:TCP protocol handler port number(프로토콜 처리기:TCP 프로토콜 처리기 포트 번호)	SSL 이 아닌 트래픽에 사용되고 있는 TCP 포트
System Environment:JVM GC count(시스템 환경:JVM GC 수)	가비지 수집 이벤트의 수
System Environment:JVM GC time(시스템 환경:JVM GC 시간)	JVM 에서 가비지 수집에 걸린 시간(밀리초)
System Environment:JVM heap size after GC(시스템 환경:GC 후 JVM 힙 크기)	최근 가비지 수집 후의 JVM 힙 크기

자세한 메트릭 이름	설명
CICS Server:Amount of CICS request data(CICS 서버:CICS 요청 데이터의 양)	연결된 CICS 서버로 전송된 요청 데이터의 양(바이트). 이 양에는 응용 프로그램 및 CICS 프로토콜 데이터가 모두 포함됩니다.
System Environment:JVM initial heap size(시스템 환경:JVM 초기 힙 크기)	JVM 힙의 초기 크기
System Environment:JVM maximum heap size(시스템 환경:JVM 최대 힙 크기)	JVM 힙의 최대 크기
Worker Threads:Current number of worker threads(작업자 스레드:현재 작업자 스레드 수)	생성된 현재 작업자 스레드 수
Worker Threads:Currently allocated worker threads(작업자 스레드:현재 할당된 작업자 스레드)	연결 관리자에서 사용되고 있는 현재 작업자 스레드 수
Worker Threads:Initial number of worker threads(작업자 스레드:초기 작업자 스레드 수)	게이트웨이 데몬에 의해 생성되도록 구성된 초기 작업자 스레드의 수
Worker Threads:Maximum number of worker threads(작업자 스레드:최대 작업자 스레드 수)	게이트웨이에서 처리할 수 있도록 구성된 병렬 작업자 스레드의 최대 수

자세한 메트릭 이름	설명
CICS Server:Amount of CICS request data(CICS 서버:CICS 요청 데이터의 양)	연결된 CICS 서버로 전송된 요청 데이터의 양(바이트). 이 양에는 응용 프로그램 및 CICS 프로토콜 데이터가 모두 포함됩니다.
Worker Threads:Number of times worker timeout reached(작업자 스레드:작업자 시간 초과 도달 횟수)	게이트웨이 데몬이 구성된 시간 초과 시간 내에 연결 관리자에 작업자 스레드를 할당하지 못한 횟수
Worker Threads:Peak number of allocated worker threads(작업자 스레드:할당된 작업자 스레드의 최대 수)	연결 관리자 스레드에 현재 할당된 작업자 스레드의 최대 수