

WebSphere® 멀티플랫폼용

IBM WebSphere Process Server

버전 7.0.0

WebSphere Process Server
모니터링

IBM®

WebSphere® 멀티플랫폼용

IBM WebSphere Process Server

버전 7.0.0

WebSphere Process Server
모니터링



2010년 4월

이 개정판은 새 개정판에 별도로 명시하지 않는 한, 멀티플랫폼용 WebSphere Process Server의 버전 7, 릴리스 0, 수정 0(제품 번호 5724-L01) 및 모든 후속 릴리스와 수정에 적용됩니다.

이 문서에 대한 사용자 의견을 보내시려면 ibmkspoe@kr.ibm.com으로 전자 우편 메시지를 보내십시오. 사용자의 의견을 기다리고 있습니다.

IBM에 정보를 보내는 경우, IBM은 귀하의 권리를 침해하지 않는 범위 내에서 적절하다고 생각하는 방식으로 귀하가 제공한 정보를 사용하거나 배포할 수 있습니다.

© Copyright IBM Corporation 2006, 2010.

목차

제 1 장 서비스 컴포넌트 모니터링 개요	1	제 3 장 모니터링한 이벤트 보기	43
모니터링 목적	1	Tivoli Performance Viewer로 성능 매트릭스 보기	43
모니터 대상	2	서비스 컴포넌트 이벤트 로그 파일 보기 및 해석	45
모니터링 사용 방법	3		
제 2 장 서비스 컴포넌트 모니터링 사용 및 구성	7	제 4 장 이벤트 카탈로그	49
성능 모니터링	7	공통 기본 이벤트 표준 요소	49
PMI(Performance Monitoring Infrastructure) 통		이벤트의 비즈니스 오브젝트	51
계	8	Business Process Choreographer 이벤트	51
Service Component Architecture에 대한 응용프		WebSphere Process Server 이벤트	51
로그램 응답 조치 통계	14	자원 어댑터 이벤트	52
서비스 컴포넌트 이벤트 모니터링	28	비즈니스 규칙 이벤트	53
비즈니스 프로세스 및 휴먼 타스크 이벤트의 모니		비즈니스 상태 머신 이벤트	54
터링 사용	28	맵 이벤트	55
서비스 컴포넌트 이벤트에 대한 로깅 구성	30	중개 이벤트	56
Common Event Infrastructure 서버로 서비스 컴		복구 이벤트	57
포넌트 모니터링	36	Service Component Architecture 이벤트	57
세션 모니터링	41	선택기 이벤트	58

제 1 장 서비스 컴포넌트 모니터링 개요

프로세스 서버에서 서비스 컴포넌트를 모니터링하는 이유에 대한 개념 개요, 모니터링하기 위해 선택하는 서비스 컴포넌트 내의 이벤트 지점 및 시스템에 모니터링을 구성하는 방법을 설명합니다.

WebSphere® Process Server에서는 성능 조정 및 문제점 판별과 같은 시스템 관리 기능을 보강하는 서비스 컴포넌트 모니터링을 위한 기능을 제공합니다. 시스템에 전개되는 응용프로그램 내에서 서비스 컴포넌트의 처리를 계속해서 모니터링하기 위해 정보 기술 전문가가 아닌 사용자를 위한 기능도 제공해서 전형적인 기능 이상의 범위를 다룹니다. 상호 연결된 컴포넌트의 전반적인 처리 플로우를 감독해서 사용자의 예상대로 시스템이 생성하도록 할 수 있습니다.

WebSphere Process Server는 WebSphere Application Server 설치의 맨 위에서 작동하며 이로 인해 시스템 성능 및 문제점 해결을 모니터링하기 위해 Application Server 인프라의 많은 기능을 사용합니다. 프로세스 서버 서비스 컴포넌트를 모니터링하기 위해 설계된 일부 추가 기능도 포함됩니다. 이 절에서는 서버 특정 서비스 컴포넌트를 모니터링하는 방법에 대해 중점을 둡니다. 이 절은 WebSphere Application Server Information Center에 있는 모니터링 및 문제점 해결 주제를 보완합니다. 따라서 해당 문서에서 결함된 제품의 기타 모니터링 기능에 대한 세부사항을 참조하십시오.

모니터링 목적

WebSphere Process Server 내의 서비스 컴포넌트를 모니터링해서 성능에 액세스하고 문제점을 해결하며 시스템에 전개된 응용프로그램을 구성하는 서비스 컴포넌트의 전반적인 처리 진행 상태를 평가합니다.

서비스 컴포넌트는 WebSphere Process Server에 통합된 필수 불가결한 기능으로, 이를 통해 엔터프라이즈에 사용되는 프로세스를 미러링하는 시스템에서 응용프로그램을 작성하고 전개할 수 있습니다. 따라서 서버가 수행하려는 작업을 관리하기 위해서는 이러한 서비스 컴포넌트를 효과적으로 모니터링해야 합니다. 서버의 서비스 컴포넌트를 모니터링해야 하는 주요 이유 세 가지는 다음과 같습니다.

문제점 해결

WebSphere Process Server 아래에 있는, WebSphere Application Server에서 제공하는 로깅 및 추적 기능을 사용하여 특정 오류를 진단할 수 있습니다. 예를 들어, 특정 응용프로그램이 예상되는 결과를 생성하지 않은 경우 로거(logger)를 설정해서 해당 응용프로그램을 구성하는 서비스 컴포넌트의 처리를 모니터링할 수 있습니다. 파일에 공개된 로그 출력이 있을 수 있으며 그러면 이를

검토해서 문제점의 원인을 정확히 지적할 수 있습니다. 문제점 해결은 시스템 관리자 및 시스템 하드웨어와 소프트웨어의 유지보수에 관련된 기타 사용자에게 중요한 TASK입니다.

성능 조정

대부분의 프로세스 서버 특정 서비스 컴포넌트가 생성하는 특정 성능 통계를 모니터링할 수 있습니다. 이 정보를 사용하여 시스템 성능 상태를 유지보수 및 조정하고 응용프로그램이 최적 상태에서 효율적으로 조정되게 하십시오. 하나 이상의 서비스가 빈약한 레벨에서 수행되고 있는 상황(기타 프로그램이 시스템에 있음을 나타내는 것일 수 있음)을 분별할 수도 있습니다. 문제점 판별과 마찬가지로 성능 조정은 정보 기술 전문가가 일반적으로 수행하는 TASK입니다.

서비스 컴포넌트의 처리 평가

문제점 판별 및 성능 조정은 특정 문제를 해결하기 위해 단기간에 수행하는 TASK입니다. 시스템에 전개된 응용프로그램으로 통합된 서비스 컴포넌트를 계속해서 모니터링하도록 프로세스 서버를 설정할 수도 있습니다. 이 유형의 서비스 컴포넌트 모니터링은 프로세스가 설계 목표에 도달하도록 하며 설계와 구현을 담당하는 사용자에게 중요하며 반드시 정보 기술 전문가가 아닌 사용자도 이를 수행할 수 있습니다.

모니터 대상

서비스 컴포넌트 이벤트가 처리 중 도달하는 특정 지점을 선택해서 WebSphere Process Server의 서비스 컴포넌트 이벤트를 모니터링할 수 있습니다. 각 서비스 컴포넌트는 응용 프로그램이 지정된 해당 지점에서 처리할 때 이벤트를 생성(또는 『가동』)하는 이벤트 지점을 정의합니다. 서비스 컴포넌트 이벤트에 대한 성능 통계를 모니터링할 수도 있습니다.

서비스 컴포넌트에서 수행하려는 모니터링 유형(문제점 판별, 성능 조정 또는 프로세스 모니터링)에 관계없이 처리 중 도달하는 특정 지점을 모니터링합니다. 이 지점을 이벤트 지점이라 하며 모니터 시 이 지점을 선택합니다. 각 이벤트 지점은 서비스 컴포넌트 종류 태그, 선택적 요소 종류(서비스 컴포넌트 유형의 특정 함수) 및 이벤트의 네이처를 캡슐화합니다. 이러한 모든 요소로 모니터링에서 생성된 이벤트 유형을 판별합니다.

이벤트 네이처는 서비스 컴포넌트 처리 중에 이벤트를 생성하는 데 필요한 상황에 대해 설명합니다. 이러한 네이처는 모니터링하도록 선택하는 서비스 컴포넌트의 논리 구조에서 핵심 요소입니다. 서비스 컴포넌트 이벤트의 가장 일반적인 네이처는 ENTRY, EXIT 및 FAILURE이지만 특정 컴포넌트 및 요소에 따라 여러 다른 네이처가 있습니다. 지정된 서비스 컴포넌트가 포함된 응용프로그램이 나중에 호출될 때마다 서비스 컴포넌트 처리가 이벤트 네이처에 해당하는 지점을 지나가면 언제나 이벤트가 가동됩니다.

서비스 컴포넌트 종류에 대해 이벤트가 정의되는 방식의 예를 들면 MAP 서비스 컴포넌트 종류는 ENTRY, EXIT 및 FAILURE 네이처가 있는 이벤트를 직접 가동합니다. MAP 컴포넌트 종류 내에서 특정 유형의 기능을 정의하는 Transformation이라는 요소

종류도 포함합니다. 이 요소도 ENTRY, EXIT 및 FAILURE 네이처가 있는 이벤트를 가동합니다. 결과적으로 MAP 서비스 컴포넌트 종류는 지정된 네이처와 요소의 조합에 따라 최대 6개의 다른 이벤트를 가동할 수 있습니다. 모든 서비스 컴포넌트, 요소 및 이벤트 네이처의 목록이 이벤트 카탈로그에 포함되어 있습니다.

모니터링은 응용프로그램 처리의 맨 위에 위치하며 서비스 컴포넌트의 처리와 인터페이스하지 않는 별도의 기능 레이어입니다. 모니터링은 지정된 이벤트 지점에서 활동을 감지하는 경우에만 서비스 컴포넌트 처리에 관여합니다. 활동이 발견되면 수행 중인 모니터링의 유형을 기반으로 이벤트가 전송된 위치 및 해당 이벤트에 포함된 데이터를 판별하는 이벤트가 모니터링에 의해 가동됩니다.

성능 매트릭스

성능 매트릭스를 집계하기 위해 서비스 컴포넌트를 모니터 중인 경우 PMI(Performance Monitoring Infrastructure)로 경량의 이벤트가 가동됩니다. 서버 특정 서버 컴포넌트에 대해 생성된 세 가지 성능 통계 중 하나 이상을 모니터링하도록 선택할 수 있습니다.

- 각 EXIT 이벤트 네이처에 대한 카운터 - 성공한 수치를 계산합니다.
- 각 FAILURE 이벤트 네이처에 대한 카운터 - 실패한 수치를 계산합니다.
- 해당 ENTRY 및 EXIT 이벤트 간에 계산된 처리 지속 기간(동기 수치만).

ARM(Application Response Measurement) 통계를 사용하여 SCA(Service Component Architecture) 레벨에서 응용프로그램의 성능을 모니터링할 수도 있습니다. 이를 통해 기타 서비스 컴포넌트 이벤트에서 다르게 사용할 수 있는 것보다 응용프로그램 내의 훨씬 더 정교한 레벨에서 응용프로그램의 모니터가 가능합니다. 이 통계를 사용하여 초기 응용프로그램 호출 및 서비스 응답(SCA를 사용할 때) 사이의 여러 다른 지점을 모니터링할 수 있습니다.

비즈니스 오브젝트가 있는 서비스 컴포넌트 이벤트

서비스 컴포넌트의 지정된 이벤트 지점에서 모니터링에 의해 생성된 이벤트의 데이터를 캡처하려는 경우 이벤트 및 공통 기본 이벤트 형식으로 인코딩할 데이터를 생성하도록 서버를 구성합니다. 각 서비스 컴포넌트 이벤트에서 캡처할 비즈니스 오브젝트 데이터의 세부사항 레벨을 지정할 수 있습니다. 특별히 구성된 CEI 서버 데이터베이스에 출력을 지정하는 Common Event Infrastructure 버스 또는 로거로 이벤트를 공개할 수 있습니다.

모니터링 사용 방법

수행할 계획 중인 모니터링의 유형에 따라 모니터링할 서비스 컴포넌트 이벤트 지점을 지정하는 데 사용할 수 있는 여러 메소드가 있습니다.

성능 통계

PMI(Performance Monitoring Infrastructure) 통계의 경우 관리 콘솔을 사용하여 모니터링하려는 특정 이벤트 지점 및 연관된 성능 조치를 지정하십시오. 모

니터링 서비스 컴포넌트 성능을 시작하면 생성된 통계는 일정한 간격으로 Tivoli® Performance Viewer에 공개됩니다. 이 표시기를 사용하여 시스템에 발생하는 결과를 보고 선택적으로 동일한 표시기 내에서 나중에 보고 분석할 수 있는 파일에 결과를 로깅할 수 있습니다.

ARM(Application Response Measurement) 통계의 경우는 관리 콘솔 요청 매트릭스 섹션을 사용하여 모니터링하려는 통계를 지정하십시오.

문제점 판별 및 비즈니스 프로세스 모니터링을 위한 공통 기본 이벤트

응용프로그램을 작성할 때 실행 중인 서버에 응용프로그램이 전개된 후 이벤트의 특정 세부사항 레벨과 함께 서비스 컴포넌트 이벤트 지점을 모니터링하도록 지정할 수 있습니다. 응용프로그램이 전개되고 이벤트를 최소 한 번은 호출한 후 모니터링할 이벤트 지점을 선택할 수도 있습니다. 두 경우 모두 모니터링으로 생성된 이벤트는 Common Event Infrastructure 버스에서 가동됩니다. 이 이벤트를 로그 파일에 공개하거나 구성된 CEI 서버 데이터베이스에 공개할 수 있습니다. WebSphere Process Server는 문제점 판별 및 비즈니스 프로세스 모니터링에 대한 두 가지 유형의 공통 기본 이벤트 사용을 지원합니다.

정적 응용프로그램 내의 특정 이벤트 지점 및 세부사항 레벨에 WebSphere Integration Developer 도구를 사용하여 모니터링을 위한 태그를 붙일 수 있습니다. 선택사항은 계속해서 모니터링하고 응용프로그램과 함께 분배 및 전개되는 확장자가 .mon인 파일에 저장될 이벤트 지점을 나타냅니다. WebSphere Process Server가 CEI 서버를 사용하도록 구성되면 모니터링 함수는 지정된 서비스가 호출될 때마다 CEI 서버에 서비스 컴포넌트 이벤트를 가동하기 시작합니다. 응용프로그램이 WebSphere Process Server에 전개되는 한, .mon 파일에 지정된 서비스 컴포넌트 이벤트 지점은 응용프로그램이 중지될 때까지 지속적으로 모니터링됩니다. 실행 중인 응용프로그램에서 추가 이벤트를 모니터링하도록 지정하고 이미 모니터링한 이벤트 지점의 세부사항 레벨을 증가시킬 수 있습니다. 그러나 이 응용프로그램이 활성화된 동안에는 전개된 응용프로그램의 .mon에 지정된 모니터링 이벤트 지점을 중지하거나 해당 세부사항 레벨을 낮출 수 없습니다.

동적 응용프로그램의 처리 중 서버를 종료하지 않고 추가 이벤트 지점을 모니터링할 필요가 있으면 동적 모니터링을 사용할 수 있습니다. 관리 콘솔을 사용하여 모니터링할 서비스 컴포넌트 이벤트 지점을 지정하고 공통 기본 이벤트에 포함될 페이로드에 대한 세부사항 레벨을 설정하십시오. 서버가 시작된 후 처리된 서비스 컴포넌트가 도달한 이벤트 지점으로 목록이 컴파일됩니다. 이 목록에서 모니터링할 개별 이벤트 지점 또는 이벤트 지점 그룹을 선택하고 서비스 컴포넌트 이벤트를 로거 또는 CEI 서버 데이터베이스로 지정하십시오.

동적 인에이블먼트의 기본 용도는 로그에 공개되는 상관된 서비스 컴포넌트 이벤트를 작성해서 서비스에 대한 문제점 판별을 수행하는 것입니다. 서비스 컴포넌트 이벤트는 요청되고 있는 데이터의 용량에 따라 커질 수 있으며 CEI 서버에 이벤트를 전송하도록 선택하는 경우 많은 데이터베이스 자원을 사용할 수 있습니다. 결과적으로 이벤트의 비즈니스 데이터를 읽어야 하는 경우 또는 그렇지 않으면 이벤트의 데이터베이스 레코드를 보존해야 하는 경우에만 모니터링 이벤트를 동적으로 CEI 서버에 공개해야 합니다. 그러나 특정 세션을 모니터링하고 있으면 CEI 서버 데이터베이스를 사용하여 이 세션에 관련된 서비스 컴포넌트 이벤트에 액세스합니다.

관련 개념

7 페이지의 『성능 모니터링』

성능 조치는 서비스 컴포넌트 이벤트 지점에 사용 가능하며 PMI(Performance Monitoring Infrastructure)를 통해 처리됩니다. 서비스 컴포넌트 이벤트 지점에서 성능 메트릭을 수집하도록 프로세스 서버를 구성합니다. 응용프로그램의 서비스 호출에서 직접 Service Component Architecture의 성능 통계를 수집할 수도 있습니다.

41 페이지의 『세션 모니터링』

공통 기본 이벤트 브라우저를 사용하여 동일한 세션 ID 속성을 포함하는 Common Event Infrastructure 데이터베이스에서 모든 이벤트를 찾아서 동일한 세션의 일부인 여러 이벤트를 모니터링할 수 있습니다.

관련 태스크

7 페이지의 제 2 장 『서비스 컴포넌트 모니터링 사용 및 구성』

서비스 컴포넌트를 모니터링하려면 먼저 모니터링 기능을 사용 가능으로 설정해야 합니다. 그런 다음, 모니터링하려는 이벤트, 이벤트에서 캡처하려는 정보 및 결과를 공개하는 데 사용되는 메소드를 지정해야 합니다.



Common Event Infrastructure 관리



관리 콘솔을 사용하여 공통 기본 이벤트 및 추적 감사 사용 가능



요청 매트릭스에서 성능 데이터 가져오기

제 2 장 서비스 컴포넌트 모니터링 사용 및 구성

서비스 컴포넌트를 모니터링하려면 먼저 모니터링 기능을 사용 가능으로 설정해야 합니다. 그런 다음, 모니터링하려는 이벤트, 이벤트에서 캡처하려는 정보 및 결과를 공개하는 데 사용되는 메소드를 지정해야 합니다.

관련 개념

3 페이지의 『모니터링 사용 방법』

수행할 계획 중인 모니터링의 유형에 따라 모니터링할 서비스 컴포넌트 이벤트 지점을 지정하는 데 사용할 수 있는 여러 메소드가 있습니다.

성능 모니터링

성능 조치는 서비스 컴포넌트 이벤트 지점에 사용 가능하며 PMI(Performance Monitoring Infrastructure)를 통해 처리됩니다. 서비스 컴포넌트 이벤트 지점에서 성능 메트릭을 수집하도록 프로세스 서버를 구성합니다. 응용프로그램의 서비스 호출에서 직접 Service Component Architecture의 성능 통계를 수집할 수도 있습니다.

저조한 성능을 진단하는지 아니면 최적의 성능을 위해 서비스 컴포넌트를 조정하는지에 관계없이 성능 Perspective에서 얼마나 다양한 런타임 및 응용프로그램 자원이 작동하고 있는지 이해하는 것이 중요합니다. PMI(Performance Monitoring Infrastructure)는 런타임 및 응용프로그램 자원 작동을 설명하는 포괄적인 데이터 세트를 제공합니다. PMI 데이터를 사용하여 Application Server의 성능 병목 현상을 식별하고 수정할 수 있습니다. Application Server의 성능 상태를 모니터링하는 데에도 PMI 데이터를 사용할 수 있습니다.

PMI는 기본 WebSphere Application Server 설치에 포함됩니다. 이 절은 WebSphere Process Server에 특정한 서비스 컴포넌트에 관련되어 있어서 성능 모니터링에 대한 보충 정보만을 제공합니다. 따라서 전체 제품의 기타 파트에 관한 PMI 사용에 대해서는 WebSphere Application Server 문서의 정보를 참조하십시오.

PMI에서 모니터링할 수 있는 WebSphere Process Server에 특정한 서비스 컴포넌트 이벤트 지점은 ENTRY, EXIT 및 FAILURE 이벤트 네이처를 포함하는 이벤트입니다. 이 패턴에 따라 정의되지 않은 이벤트 소스는 지원되지 않습니다. 지원되는 이벤트에는 측정할 수 있는 세 가지 유형의 성능 통계가 있습니다.

- 성공적 호출
- 실패한 호출
- 이벤트 완료 경과 시간

ARM(Application Response Measurement) 통계를 사용하여 응용프로그램의 서비스 호출에서 파생된 성능 통계를 모니터링할 수도 있습니다. 이러한 통계는 엔터프라이즈 응용프로그램을 구성하는 프로세스 서비스 컴포넌트 이벤트 아래에 있는 실제 런타임 프로세스를 측정합니다. 이러한 통계를 사용하여 응용프로그램의 처리를 위한 다양한 성능 조치를 이끌어낼 수 있습니다.

관련 개념

3 페이지의 『모니터링 사용 방법』

수행할 계획 중인 모니터링의 유형에 따라 모니터링 서비스 컴포넌트 이벤트 지점을 지정하는 데 사용할 수 있는 여러 메소드가 있습니다.

PMI(Performance Monitoring Infrastructure) 통계

PMI(Performance Monitoring Infrastructure)를 사용하여 세 가지 유형의 성능 통계 즉, 성공한 호출 수, 장애 수 및 이벤트 완료 경과 시간을 모니터링할 수 있습니다. 이 통계는 ENTRY, EXIT 및 FAILURE 이벤트 네이처가 있는 이벤트에만 사용할 수 있습니다.

관리 콘솔을 통한 PMI 사용

성능 데이터를 모니터링하려면 먼저 서버에서 PMI(Performance Monitoring Infrastructure)를 사용 가능하게 해야 합니다.

이 태스크 정보

관리 콘솔을 통해 PMI를 사용 가능하게 할 수 있습니다.

프로시저

1. 관리 콘솔을 여십시오.
2. 콘솔 탐색 트리에서 서버 > 서버 유형 > **WebSphere Application Server**를 클릭하십시오.
3. *server_name*을 클릭하십시오.

주: 관리 콘솔에서 모니터링 및 조정 > PMI > *server_name*을 클릭하여 동일한 패널을 여십시오.

4. 구성 탭을 클릭하십시오.
5. **PMI(Performance Monitoring Infrastructure)** 사용 선택란을 선택하십시오.
6. 옵션: 순차 카운터 갱신 사용 선택란을 선택해서 세밀한 통계 갱신을 사용 가능하게 하십시오.
7. 서버 이름 링크를 클릭해서 서버 PMI 구성 페이지로 돌아가십시오.
8. 적용 또는 확인을 클릭하십시오.
9. 저장을 클릭하십시오.

10. 서버를 다시 시작하십시오.

다음에 수행할 작업

변경사항은 서버를 다시 시작해야 적용됩니다.

이벤트 성능 통계

성능 모니터링 통계는 대부분의 서버 이벤트에 대해 사용 가능합니다. 성능 모니터링 통계를 사용하여 성공한 호출 요청 및 실패한 호출 요청의 계수와 이벤트를 완료하는데 걸린 시간을 모니터링할 수 있습니다.

PMI(Performance Monitoring Infrastructure)를 사용하여 다음 표에 표시된 대로 특정 서버 이벤트에서 생성된 세 가지 성능 통계를 모니터링할 수 있습니다.

표 1. 이벤트의 PMI 통계

통계 이름	유형	설명
BadRequests	카운터	실패 이벤트 호출 수
GoodRequests	카운터	성공한 이벤트 호출 수
ResponseTime	타이머	이벤트 완료 경과 시간

이 통계는 ENTRY, EXIT 및 FAILURE 네이처가 있는 요소의 서비스 컴포넌트 이벤트로 제한됩니다. 응용프로그램에서 제공된 서버 이벤트 유형의 단일 이벤트에 대해 각 통계가 작성됩니다. 모든 성능 조치는 카운터(제공된 이벤트 지점의 누적 가동 수) 또는 타이머(두 이벤트 지점의 가동 간 밀리초 단위 지속 기간)입니다. 모니터링할 수 있는 각 이벤트 종류(및 관련 요소)가 아래에 나옵니다.

표 2. 이벤트 성능 통계를 생성할 수 있는 이벤트 유형 및 요소

이벤트 유형	요소
비즈니스 프로세스	Process Invoke Staff Receive Wait Compensate Pick Scope
휴먼 타스크	Task
비즈니스 규칙	Operation
비즈니스 상태 머신	Transition Guard Action EntryAction ExitAction
선택기	Operation

표2. 이벤트 성능 통계를 생성할 수 있는 이벤트 유형 및 요소 (계속)

이벤트 유형	요소
맵	Map Transformation
중개	OperationBinding ParameterMediation
자원 어댑터	InboundEventRetrieval InboundEventDelivery Outbound

관련 참조

14 페이지의 『Service Component Architecture에 대한 응용프로그램 응답 조치 통계』

SCA(Service Component Architecture) 레벨에서 모니터링할 수 있는 25개의 성능 통계가 있습니다. 카운터 또는 타이머인 이러한 ARM(Application Response Measurement)을 사용해서 다양한 패턴의 서비스로의 호출 및 서비스에서의 응답을 측정할 수 있습니다.

모니터할 성능 통계 지정

관리 콘솔을 사용하여 PMI(Performance Monitoring Infrastructure)를 통해 단일 통계, 복수 통계 또는 관련된 통계 그룹을 모니터링하도록 지정할 수 있습니다.

시작하기 전에

이 작업을 수행하기 전에 성능 모니터링을 사용 가능하게 했는지 및 모니터링하려는 이벤트를 적어도 한번은 호출했는지 확인하십시오.

프로시저

1. 관리 콘솔을 여십시오.
2. 모니터링 및 조정 → **PMI(Performance Monitoring Infrastructure)**를 선택하십시오.
3. 모니터링하려는 이벤트 지점을 포함하는 Node Agent 또는 서버를 선택하십시오.

주: 클러스터의 통계를 모니터링하도록 선택할 수는 없습니다. 특정 서버나 노드의 통계만을 모니터링할 수 있습니다.

4. WBISStats.RootGroup 또는 Enterprise Bean과 같은 일부 그룹을 펼치십시오. 모니터링할 수 있는 모든 통계는 나열된 그룹 내에 있습니다. 일부 통계는 서버가 마지막으로 시작된 이후 호출되지 않았기 때문에 표시되지 않을 수 있습니다.
5. 패널의 왼쪽에 있는 트리에서 모니터링하려는 통계를 선택한 다음 오른쪽에서 수집하려는 통계를 선택하고 **사용**을 클릭하십시오. 모니터링하려는 모든 통계에 대해 반복하십시오.
6. 서버 이름 링크를 클릭해서 서버 PMI 구성 페이지로 돌아가십시오.

7. 적용 또는 확인을 클릭하십시오.

8. 저장을 클릭하십시오.

결과

이제 Tivoli Performance Viewer에서 선택한 통계의 성능 모니터링을 시작할 수 있습니다.

주: 이러한 통계를 볼 때 카운터 유형 통계와 지속 기간 유형 통계를 혼합하지 마십시오. 카운터는 누적되며 그래프로 표시되는 비율이 응용프로그램에 따라 빠르게 증가할 수 있습니다. 이와 반대로 지속 기간 통계는 시스템이 각 이벤트를 처리하는 데 걸리는 평균 시간을 나타내기 때문에 일정 범위 내에서 유지됩니다. 결과적으로 통계 및 상대 배율 간의 불일치로 인해 하나 또는 기타 유형의 통계가 표시기 그래프에서 부정확하게 표시될 수 있습니다.

학습서: 서비스 컴포넌트 성능 모니터링

이 학습서는 성능 모니터링 설정 예제 및 결과 통계를 보는 방법을 단계적으로 안내합니다.

모니터하는 서비스 컴포넌트 이벤트 지점의 경우 PMI(Performance Monitoring Infrastructure)에 공개하고 TPV(Tivoli Performance Viewer)에서 결과 성능 통계를 볼 수 있습니다. 이 연습은 서비스 컴포넌트 이벤트 지점의 성능 모니터링이 Common Event Infrastructure 서버 및 로거를 사용하는 모니터링과 어떻게 다른지 설명합니다. 주목할 주요 차이점은 특정의 네이처를 가지고 있는 개별 이벤트 대신, 성능 모니터링에 대한 전체 서비스 컴포넌트 요소를 선택하는 것입니다. WebSphere Process Server가 ENTRY, EXIT 및 FAILURE 네이처가 있는 이벤트를 갖는 서비스 컴포넌트 요소에 대해서만 성능을 모니터링할 수 있기 때문에 이러한 유형의 서비스 컴포넌트 유형만 모니터링하도록 선택할 수 있습니다.

서비스 컴포넌트 이벤트 지점 ENTRY, EXIT 및 FAILURE가 모든 모니터링 유형에 대해 동일한 반면, 서버의 성능 모니터링 기능은 CEI 이벤트에 포함된 모든 정보를 포함하지는 않는 "최소화된" 이벤트를 발생합니다. 이 이벤트는 해당되는 이벤트 세트로부터 성능 통계를 계산하는 PMI로 전송됩니다.

- 성공적인 호출 - 해당되는 ENTRY 이벤트 다음에 EXIT 네이처 유형의 이벤트 발생.
- 실패한 호출 - 해당되는 ENTRY 이벤트 다음에 FAILURE 네이처의 이벤트 발생.
- 성공적 완료에 대한 시간 - ENTRY 이벤트 발생 시간과 해당되는 EXIT 이벤트 지점 발생 시간 사이의 경과 시간.

PMI는 통계를 TPV에 공개합니다. 이 TPV는 성공 및 실패 호출 수에 대한 누적 카운터와 완료 응답 시간의 현재 평균을 표시합니다.

이 학습서의 목표

이 학습서를 완료하면 다음을 수행할 수 있습니다.

- 모니터하려고 하는 서비스 컴포넌트 요소의 성능 통계를 선택합니다.
- 결과 성능 통계를 보고 해석합니다.

이 학습을 완료하는 데 필요한 시간

이 학습서는 완료하는 데 대략 15 - 20분이 소요됩니다.

전제조건

이 학습서를 수행하려면 이미 다음을 수행한 상태여야 합니다.

- 서버 구성 및 시작
- 서버에서 PMI 사용
- 서버에서 샘플 갤러리 응용프로그램 설치 및 시작
- 서버에서 비즈니스 규칙 샘플 응용프로그램 설치 및 시작. 비즈니스 규칙 샘플 응용 프로그램을 설정하고 실행하려면 샘플 갤러리 페이지에 있는 지시사항을 따르십시오.

이와 같은 모든 전제조건이 완료되고 난 후에는 학습서를 진행하기 전에 최소 한 번 샘플 갤러리에서 비즈니스 규칙 샘플 응용프로그램을 실행하십시오.

예제: 서비스 컴포넌트 성능 모니터링:

모니터링 성능을 위해 관리 콘솔을 사용하여 모니터링할 서비스 컴포넌트를 선택하고 성능 조치를 볼 수 있습니다. 이 예제는 성능 통계를 모니터링하기 위한 콘솔 사용을 보여줍니다.

이 태스크 정보

이 시나리오의 비즈니스 규칙 샘플 응용프로그램을 사용하여 세 가지 성능 통계 즉, 성공, 실패 및 응답 시간을 모두 모니터링합니다. 이 응용프로그램을 포함하는 웹 페이지가 이미 열려 있어야 합니다. 모니터링을 시작한 후 샘플을 여러 번 실행할 예정이므로 응용프로그램을 계속 열어 놓으십시오. 샘플을 최소 한 번은 이미 실행했는지 확인하십시오. 실행한 경우 모니터하려고 선택할 수 있는 함수 목록에 샘플이 나타납니다.

프로시저

1. 관리 콘솔을 여십시오.
2. 모니터할 클러스터 또는 서버를 선택하십시오.
 - 클러스터를 모니터하려면 서버 → 클러스터 → **WebSphere Application Server** 클러스터 → *cluster_name*을 클릭하십시오.
 - 단일 서버를 모니터하려면 서버 → 서버 유형 → **WebSphere Application Server** → *server_name*을 클릭하십시오.

3. 런타임 탭을 클릭하십시오.
4. 성능에서 **PMI**를 클릭하십시오.
5. 사용자 정의를 선택하십시오.
6. **WBISStats.RootGroup** → **BR** → **brsample_module.DiscountRuleGroup** → 조작을 펼치십시오.
7. **_calculateDiscount**를 선택하십시오.
8. BadRequests, GoodRequests 및 ResponseTime 옆의 선택란을 선택하십시오.
9. 사용을 클릭하십시오.
10. 탐색 분할창에서 모니터링 및 조정 → 성능 표시기 → 현재 활동을 클릭하십시오.
11. *server_name* 옆의 선택란을 선택한 후 모니터링 시작을 클릭하십시오.
12. *server_name*을 클릭하십시오.
13. **WBISStats.RootGroup** → **BR** → **brsample_module.DiscountRuleGroup** → 조작을 펼치십시오.
14. **_calculateDiscount** 옆의 선택란을 선택하십시오.

결과

이제 빈 그래프와 이 그래프 아래에 세 가지 통계의 이름 및 값이 표시되어야 합니다. 통계 이름 옆의 선택란이 아직 선택되지 않은 경우 선택하십시오. 이제 PMI는 선택된 이벤트의 성능 데이터를 공개할 준비가 되었으며, Tivoli Performance Viewer가 결과를 표시할 준비가 되었습니다.

비즈니스 규칙 샘플 응용프로그램을 여러 번 실행한 다음 주기적으로 새로 고쳐지는 성능 표시기를 지켜 보십시오. 이제 그래프에 성공한 요청의 누적 수와 성공한 각 요청에 대한 평균 응답 시간을 나타내는 행이 있어야 함에 유의하십시오. 그래프 아래에 각 통계의 이름 옆에서 값도 볼 수 있습니다. 성공 수를 나타내는 행은 추가로 샘플을 호출함에 따라 계속해서 올라가야 하는 반면 응답 시간 행은 몇 번의 새로 고치기 후 낮아져야 합니다.

이 예제를 완료한 후 WebSphere Process Server가 서비스 컴포넌트의 성능 모니터링을 구현하는 방법을 이해해야 합니다. 모니터링 서비스 컴포넌트를 선택하는 방법 및 성능 통계를 계산하는 방법을 알아야 합니다. 성능 모니터를 시작하고 사용 중인 응용프로그램에 대한 성능 조치를 볼 수도 있습니다.

다음에 수행할 작업

성능 모니터링은 많은 시스템 자원을 소모할 수 있습니다. 따라서 이 작업을 완료한 후에는 모니터를 중지해야 합니다. 이를 수행하려면 Tivoli 성능 표시기 링크를 클릭하고 노드 및 서버 모두를 선택한 후 모니터링 중지를 누르십시오.

Service Component Architecture에 대한 응용프로그램 응답 조치 통계

SCA(Service Component Architecture) 레벨에서 모니터링할 수 있는 25개의 성능 통계가 있습니다. 카운터 또는 타이머인 이러한 ARM(Application Response Measurement)을 사용해서 다양한 패턴의 서비스로의 호출 및 서비스에서의 응답을 측정할 수 있습니다.

다음 테이블에 표시된 ARM(Application Response Measurement) 통계는 단순화된 방식의 SCA(Service Component Architecture) 레이어에 대한 호출자 호출의 시간 및 계수 조치와 서비스에서 리턴된 결과입니다. 실제로 지연된 응답의 동기 및 비동기 구현 사이의 다양한 여러 서비스 호출 패턴, 결과 검색, 콜백 및 단방향 호출이 있습니다. 그러나 모든 패턴은 호출자 호출과 서비스, 서비스로부터의 응답 또는 일부 경우 SCA 레이어가 사이에 있는 데이터 소스 사이에 있습니다.

관리 콘솔에서 **모니터링 및 조정 > 요청 매트릭스** 패널을 열어서 모니터링하려는 ARM 통계를 지정할 수 있습니다. 요청 매트릭스 정보는 나중에 저장 및 분석할 수 있도록 로그 파일에 저장되거나 ARM 에이전트에 송신되거나 또는 두 가지 모두입니다. WebSphere Process Server에서는 ARM 에이전트를 제공하지 않습니다. 그러나 ARM 4.0에 따라 에이전트 사용을 지원합니다. 자신만의 ARM 구현 프로바이더를 선택해서 ARM 구현 라이브러리를 얻을 수 있습니다. ARM 프로바이더의 지시사항에 따르면 ARM 프로바이더에 있는 ARM API Java™ 아카이브(JAR) 파일이 클래스 경로에 있는지 확인하여 WebSphere Process Server가 필요한 클래스를 로드할 수 있도록 하십시오. 그런 다음, 서버를 다시 시작하기 전에 관리 콘솔에서 **Application Server > server_name > 프로세스 정의 > JVM(Java Virtual Machine) > 사용자 정의 특성**을 선택해서 각 서버에 대한 시스템 특성에 다음 항목을 추가해야 합니다.

- Arm40.ArmMetricFactory — ARM 구현 프로바이더 매트릭스 팩토리의 전체 Java 클래스 이름입니다.
- Arm40.ArmTranReportFactory — ARM 구현 프로바이더 트랜잭션 보고서 팩토리의 전체 Java 클래스 이름입니다.
- Arm40.ArmTransactionFactory — ARM 구현 프로바이더 트랜잭션 팩토리의 전체 Java 클래스 이름입니다.

ARM 통계를 수집하도록 서버를 구성하는 방법에 대한 자세한 내용은 WebSphere Application Server 문서를 참조하십시오.

표 3. ARM 통계를 생성할 수 있는 이벤트 유형 및 요소

이벤트 유형	요소
비즈니스 프로세스	Process
휴먼 태스크	Task
비즈니스 규칙	Operation

표 3. ARM 통계를 생성할 수 있는 이벤트 유형 및 요소 (계속)

이벤트 유형	요소
비즈니스 상태 머신	Transition Guard Action EntryAction ExitAction
선택기	Operation
맵	Map Transformation
중개	OperationBinding ParameterMediation
자원 어댑터	InboundEventRetrieval InboundEventDelivery Outbound

표 4. 공통. 다음 통계는 모든 서비스 호출 패턴에 공통됩니다.

통계 이름	유형	설명
GoodRequests	카운터	예외가 발생하지 않은 서버 호출 수
BadRequests	카운터	예외가 발생한 서버 호출 수
ResponseTime	타이머	요청 수신과 결과 계산 사이의 서버측에서 측정된 지속 기간
TotalResponseTime	타이머	호출자가 서비스를 요청한 시간부터 호출자에 대한 결과가 사용 가능한 시간까지 호출자측에서 측정된 지속 기간. 호출자의 결과 처리는 포함하지 않습니다.
RequestDeliveryTime	타이머	호출자가 서비스를 요청한 시간부터 서버측 구현으로 요청이 인계되는 시간까지 호출자측에서 측정된 지속 기간. 분배 환경에서 이 조치의 품질은 시스템 시계의 동기화 품질에 따라 다릅니다.
ResponseDeliveryTime	타이머	결과가 클라이언트에 사용 가능하도록 하는 데 필요한 시간. 지연된 응답의 경우 이 시간에는 결과 검색 시간이 포함되지 않습니다. 분배 환경에서 이 조치의 품질은 시스템 시계의 동기화 품질에 따라 다릅니다.

표 5. 참조. 이러한 통계는 호출자가 서비스로부터의 응답 없이 SCA 레이어 또는 데이터 소스에 호출할 때 발생합니다.

통계 이름	유형	설명
GoodRefRequests	카운터	예외가 발생하지 않은 SCA 레이어에 대한 호출자 호출 수
BadRefRequests	카운터	예외가 발생한 SCA 레이어에 대한 호출자 호출 수
RefResponseTime	타이머	호출자가 SCA 레이어를 요청한 시간부터 이 호출의 결과가 호출자에 리턴된 시간까지 호출자측에서 측정된 지속 기간
BadRetrieveResult	카운터	예외가 발생한 데이터 소스에 대한 호출자 호출 수
GoodRetrieveResult	카운터	예외가 발생하지 않은 데이터 소스에 대한 호출자 호출 수

표 5. 참조 (계속). 이러한 통계는 호출자가 서비스로부터의 응답 없이 SCA 레이어 또는 데이터 소스에 호출할 때 발생합니다.

통계 이름	유형	설명
RetrieveResultResponseTime	타이머	호출자가 SCA 레이어를 요청한 시간부터 데이터 소스 응답이 호출자에 리턴된 시간까지 호출자측에서 측정한 지속 기간
RetrieveResultWaitTime	타이머	제한시간 초과가 발생하는 경우 호출자측에서 측정한 지속 기간

표 6. 대상. 이 통계는 서비스와 SCA 또는 데이터 소스 간에 나온 요청이 있을 때 발생합니다.

통계 이름	유형	설명
GoodTargetSubmit	카운터	예외가 발생하지 않은 서비스에 대한 SCA 호출 수
BadTargetSubmit	카운터	예외가 발생한 서비스에 대한 SCA 호출 수
TargetSubmitTime	타이머	SCA가 서비스를 요청한 시간부터 이 호출의 결과가 SCA에 리턴된 시간까지 서버측에서 측정한 지속 기간
GoodResultSubmit	카운터	예외가 발생하지 않은 데이터 소스에 대한 서비스 호출 수
BadResultSubmit	카운터	예외가 발생한 데이터 소스에 대한 서비스 호출 수
ResultSubmitTime	타이머	서비스가 데이터 소스를 요청한 시간부터 결과가 서비스에 리턴된 시간까지 서버측에서 측정한 지속 기간

표 7. 콜백. 이 통계는 호출자에 콜백(원래 호출의 "동위")이 있을 때 발생합니다.

통계 이름	유형	설명
GoodCB	카운터	예외가 발생하지 않은 콜백에 대한 SCA 호출 수
BadCB	카운터	예외가 발생한 콜백에 대한 SCA 호출 수
CBTime	타이머	SCA가 콜백을 요청한 시간부터 콜백의 결과가 SCA에 리턴된 시간까지의 지속 기간
GoodCBSSubmit	카운터	서비스에서 콜백을 핸들하는 SCA로의 예외가 발생하지 않은 호출 수
BadCBSSubmit	카운터	서비스에서 콜백을 핸들하는 SCA로의 예외가 발생한 호출 수
CBSSubmitTime	타이머	서비스가 콜백을 핸들하는 SCA를 요청한 시간부터 SCA의 결과가 서비스에 리턴된 시간까지의 지속 기간

관련 참조

8 페이지의 『PMI(Performance Monitoring Infrastructure) 통계』

PMI(Performance Monitoring Infrastructure)를 사용하여 세 가지 유형의 성능 통계 즉, 성공한 호출 수, 장애 수 및 이벤트 완료 경과 시간을 모니터링할 수 있습니다. 이 통계는 ENTRY, EXIT 및 FAILURE 이벤트 네이처가 있는 이벤트에만 사용할 수 있습니다.


관련 정보


 WebSphere Application Server 문서 Network Deployment

동기 호출

서비스의 응답 및 서비스에 대한 단순 SCA(Service Component Architecture) 호출에서 ARM(Application Response Measurement) 성능 통계를 가져올 수 있습니다.

매개변수

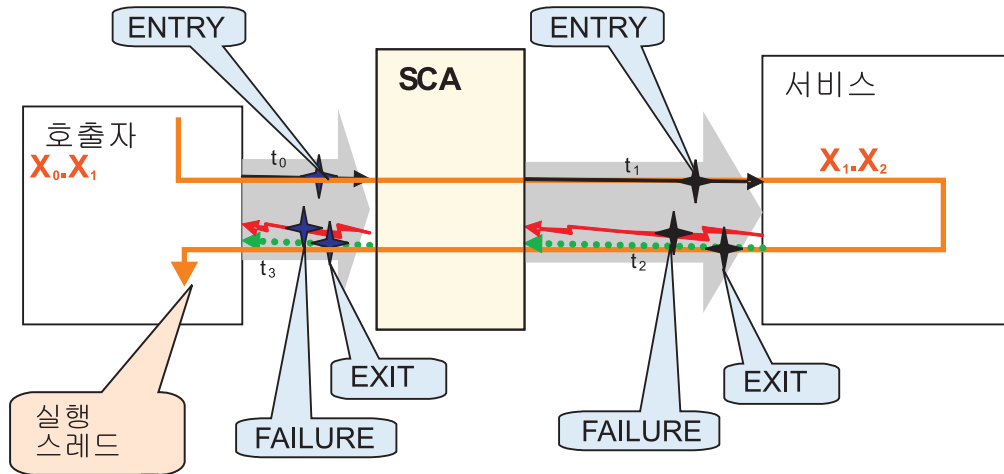
SCA 컴포넌트에 대한 이벤트 모니터링은 검은색  로 표시되는 이벤트 지점을 포

함하는 반면 파란색  로 표시되는 이벤트 지점은 파일 PMI/ARM 통계를 계산하는 데에만 사용됩니다.

아래의 표와 다이어그램에서 "현재" ARM 트랜잭션(X_1 로 표시됨)은 호출 서비스 컴포넌트가 처음으로 호출되었을 때 작성됩니다. 호출자가 서비스 컴포넌트가 아닌 경우에는 현재 ARM 트랜잭션이 사용되거나 새 ARM 트랜잭션이 작성됩니다. 시작 트랜잭션이 아닌 경우 다음 표 및 다이어그램에서 X_n, X_{n+1} 표기법으로 표시된 대로 상위를 갖습니다. 표기법은 트랜잭션 연계를 문서화하는 데 사용됩니다. 모든 SCA 호출은 새 트랜잭션을 시작하며 호출자의 현재 트랜잭션이 이 트랜잭션의 상위이 됩니다. 새 트랜잭션을 작성할 수 있으며 현재 트랜잭션에 액세스할 수 있지만 SCA 트랜잭션 연계를 수정하지 않습니다.

표 8. SCA의 동기 호출에 대한 ARM 통계

통계	공식	ARM 트랜잭션
TotalResponseTime	$t_3 - t_0$	$X_0 \cdot X_1$
RequestDeliveryTime	$t_1 - t_0$	$X_1 \cdot X_2$
ResponseDeliveryTime	$t_3 - t_2$	
GoodRequests	Count _{EXIT}	
BadRequests	Count _{FAILURE}	
ProcessTime	$t_2 - t_1$	



동기 구현의 지연된 응답

요청의 동기 호출로 ARM(Application Response Measurement) 통계를 가져올 수 있습니다. 리턴된 결과가 출력으로 동기 구현에 대한 데이터 저장소로 전송됩니다.

매개변수

SCA(Service Component Architecture) 컴포넌트에 대한 이벤트 모니터링은 검은색



으로 표시되는 이벤트 지점을 포함하는 반면 파란색



으로 표시되는 이벤트 지점은 PMI/ARM 통계를 계산하고 시작하는 데에만 사용됩니다.

아래의 표와 다이어그램에서 "현재" ARM 트랜잭션(X_1 로 표시됨)은 호출 서비스 컴포넌트가 처음으로 호출되었을 때 작성됩니다. 호출자가 서비스 컴포넌트가 아닌 경우에는 현재 ARM 트랜잭션이 사용되거나 새 ARM 트랜잭션이 작성됩니다. 시작 트랜잭션이 아닌 경우 다음 표 및 다이어그램에서 X_n, X_{n+1} 표기법으로 표시된 대로 상위를 갖습니다. 표기법은 트랜잭션 연계를 표시하는 데 사용됩니다. 모든 SCA 호출은 새 트랜잭션을 시작하며 호출자의 현재 트랜잭션이 이 트랜잭션의 상위이 됩니다. 새 트랜잭션을 작성할 수 있으며 현재 트랜잭션에 액세스할 수 있지만 SCA 트랜잭션 연계를 수정할 수 없습니다.

표 9. 요청 및 리턴 결과 호출

유형	통계	공식	ARM 트랜잭션
공통	TotalResponseTime	$t_3 - t_0$	X_0, X_1
	RequestDeliveryTime	$t'_0 - t_0$	X_1, X_2
	ResponseDeliveryTime	해당 없음	해당 없음
	GoodRequests	$Count_{EXIT}$	X_1, X_2
	BadRequests	$Count_{FAILURE}$	
	ResponseTime	$t'_1 - t'_0$	

표 9. 요청 및 리턴 결과 호출 (계속)

유형	통계	공식	ARM 트랜잭션
참조 A	GoodRefRequest	$Count_{EXIT}$	X_1, X_2
	BadRefRequests	$Count_{FAILURE}$	
	RefResponseTime	$t_1 - t_0$	

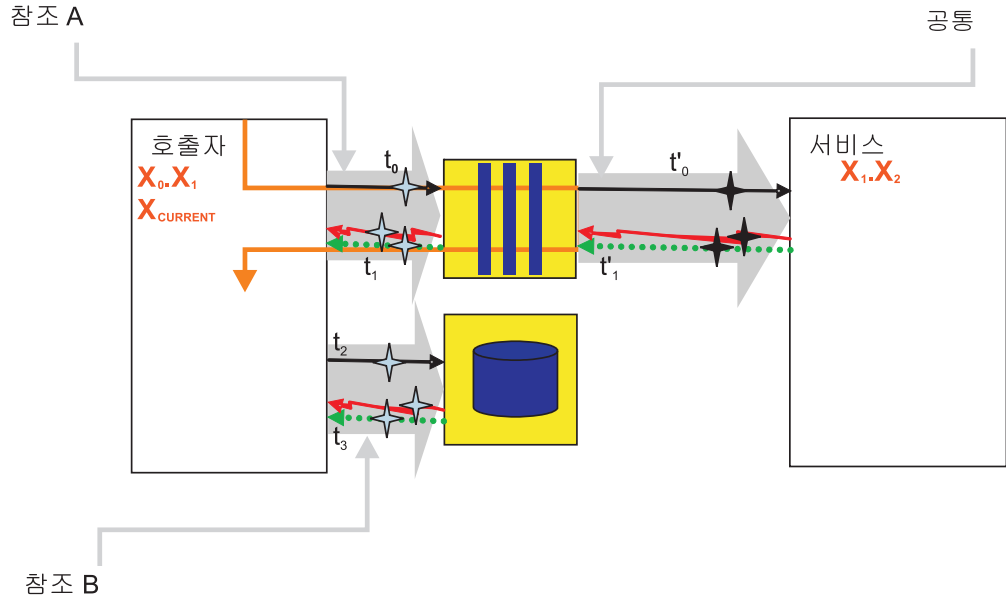


표 10. 데이터 소스에 대한 출력 호출

유형	통계	공식	ARM 트랜잭션
참조 B	GoodRetrieveResult	$Count_{EXIT}$	X_1, X_2
	BadRetrieveResult	$Count_{FAILURE}$	
	ResultRetrieveResponseTime	$\sum t_3 - t_2$	
	ResultRetrieveWaitTime	$\sum timeout$	

비동기 구현의 지연된 응답

비동기 구현에서 ARM(Application Response Measurement) 통계를 가져올 수 있습니다. 서비스 및 리턴 결과에 대한 호출이 호출되지만 결과 출력이 서비스 대상에서 데이터 저장소로 전송됩니다.

매개변수

SCA(Service Component Architecture) 컴포넌트에 대한 이벤트 모니터링은 검은색



으로 표시되는 이벤트 지점을 포함하는 반면 파란색



으로 표시되는 이벤트 지점은 PMI/ARM 통계를 계산하고 시작하는 데에만 사용됩니다.

아래의 표와 다이어그램에서 "현재" ARM 트랜잭션(X_1 로 표시됨)은 호출 서비스 컴포넌트가 처음으로 호출되었을 때 작성됩니다. 호출자가 서비스 컴포넌트가 아닌 경우에는 현재 ARM 트랜잭션이 사용되거나 새 ARM 트랜잭션이 작성됩니다. 시작 트랜잭션이 아닌 경우 다음 표 및 다이어그램에서 X_n, X_{n+1} 표기법으로 표시된 대로 상위를 갖습니다. 표기법은 트랜잭션 연계를 표시하는 데 사용됩니다. 모든 SCA 호출은 새 트랜잭션을 시작하며 호출자의 현재 트랜잭션이 이 트랜잭션의 상위가 됩니다. 새 트랜잭션을 작성할 수 있으며 현재 트랜잭션에 액세스할 수 있지만 SCA 트랜잭션 연계를 수정할 수 없습니다.

표 11. 요청 및 리턴 결과 호출

유형	통계	공식	ARM 트랜잭션
공통	TotalResponseTime	$t_3 - t_0$	$X_0 \cdot X_1$
	RequestDeliveryTime	$t'_0 - t_0$	$X_1 \cdot X_2$
	ResponseDeliveryTime	$t'_{03} - t'_2$	
	GoodRequests	$Count_{EXIT}$	
	BadRequests	$Count_{FAILURE}$	
	ResponseTime	$t'_3 - t'_0$	
참조 A	GoodRefRequest	$Count_{EXIT}$	$X_0 \cdot X_1$
	BadRefRequests	$Count_{FAILURE}$	
	RefResponseTime	$t_1 - t_0$	
대상 A	GoodTargetSubmit	$Count_{EXIT}$	$X_1 \cdot X_2$
	BadTargetSubmit	$Count_{FAILURE}$	
	TargetSubmitTime	$t'_1 - t'_0$	

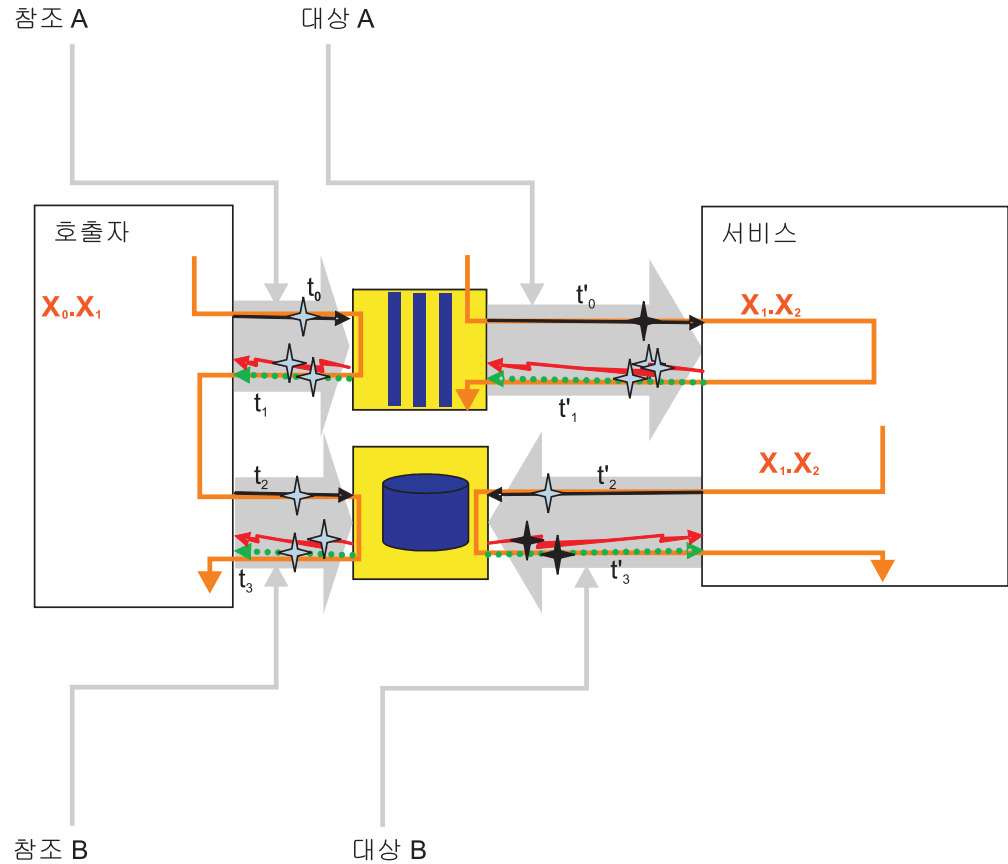


표 12. 데이터 저장소로의 리턴 결과 호출

유형	통계	공식	ARM 트랜잭션
참조 B	GoodResultSubmit	$Count_{EXIT}$	$X_0 \cdot X_1$
	BadResultSubmit	$Count_{FAILURE}$	
	ResultResponseTime	$t'_3 - t'_2$	
대상 B	GoodResultRetrieve	$Count_{EXIT}$	$X_1 \cdot X_2$
	BadResultRetrieve	$Count_{FAILURE}$	
	ResultRetrieveResponseTime	$\sum t_3 - t_2$	
	ResultRetrieveWaitTime	$\sum timeout$	

비동기 결과 검색의 지연된 응답

$X_{PARENT-1}$ 및 $X_{PARENT-2}$ 에 공통 상위 트랜잭션이 있는 경우에만 ARM 트랜잭션을 사용하여 ResultRetrieve ARM(Application Response Measurement) 통계를 일부 원래 요청과 상관시킬 수 있습니다. 요청 호출 및 결과 검색은 다른 스레드에서 발생합니다.

매개변수

SCA(Service Component Architecture) 컴포넌트에 대한 이벤트 모니터링은 검은색



으로 표시되는 이벤트 지점을 포함하는 반면 파란색



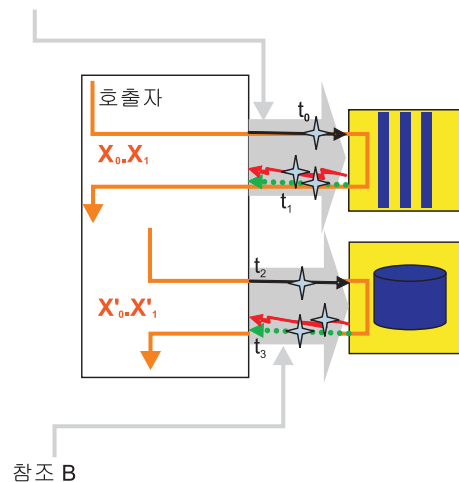
으로 표시되는 이벤트 지점은 PMI/ARM 통계를 계산하고 시작하는 데에만 사용됩니다.

아래의 표와 다이어그램에서 "현재" ARM 트랜잭션(X_1 로 표시됨)은 호출 서비스 컴포넌트가 처음으로 호출되었을 때 작성됩니다. 호출자가 서비스 컴포넌트가 아닌 경우에는 현재 ARM 트랜잭션이 사용되거나 새 ARM 트랜잭션이 작성됩니다. 시작 트랜잭션이 아니면 상위가 있습니다. 이 내용은 다음 표와 다이어그램에서 X_n, X_{n+1} 로 표시됩니다. 이는 트랜잭션 연계를 보여주는 데 사용됩니다. 모든 SCA 호출은 새 트랜잭션을 시작하며 호출자의 현재 트랜잭션이 이 트랜잭션의 상위가 됩니다. 새 트랜잭션을 작성할 수 있으며 현재 트랜잭션에 액세스할 수 있지만 SCA 트랜잭션 연계가 수정되는 않습니다.

표 13. 요청 및 리턴 결과 호출

유형	통계	공식	ARM 트랜잭션
공통	TotalResponseTime	$t_3 - t_0$	X_0, X_1
	RequestDeliveryTime	$t'_0 - t_0$	X_1, X_2
	ResponseDeliveryTime	해당 없음	해당 없음
	GoodRequests	Count _{EXIT}	X_1, X_2
	BadRequests	Count _{FAILURE}	
	ResponseTime	아래 다이어그램 참조	
참조 A	GoodReferenceRequest	Count _{EXIT}	X_1, X_2
	BadReferenceRequests	Count _{FAILURE}	
	ReferenceResponseTime	$t_1 - t_0$	

참조 A



다이어그램의 오른쪽의 경우 "동기/비동기 구현에서의 지연된 응답" 주제의 다이어그램을 참조하십시오.

요청 수신 시간 t_0 이 두 경우 모두에서 동일합니다.

참조 B

표 14. 요청 및 리턴 결과 호출

유형	통계	공식	ARM 트랜잭션
참조 B	GoodRetrieveResult	Count _{EXIT}	X' ₀ .X' ₁
	BadRetrieveResult	Count _{FAILURE}	
	RetrieveResultResponseTime	$\sum t_3 - t_2$	
	RetrieveResultWaitTime	$\sum \text{timeout}$	

동기 구현의 비동기 콜백

콜백 요청 및 콜백 실행이 동기 구현에서 다른 스레드를 사용할 때 ARM(Application Response Measurement) 통계를 가져올 수 있습니다.

매개변수

SCA(Service Component Architecture) 컴포넌트에 대한 이벤트 모니터링은 검은색



으로 표시되는 이벤트 지점을 포함하는 반면 파란색



으로 표시되는 이벤트 지점은 PMI/ARM 통계를 계산하고 시작하는 데에만 사용됩니다.

아래의 표와 다이어그램에서 "현재" ARM 트랜잭션(X₁로 표시됨)은 호출 서비스 컴포넌트가 처음으로 호출되었을 때 작성됩니다. 호출자가 서비스 컴포넌트가 아닌 경우에는 현재 ARM 트랜잭션이 사용되거나 새 ARM 트랜잭션이 작성됩니다. 시작 트랜잭션이 아닌 경우 다음 표 및 다이어그램에서 X_n.X_{n+1} 표기법으로 표시된 대로 상위를 갖습니다. 표기법은 트랜잭션 연계를 표시하는 데 사용됩니다. 모든 SCA 호출은 새 트랜잭션을 시작하며 호출자의 현재 트랜잭션이 이 트랜잭션의 상위가 됩니다. 새 트랜잭션을 작성할 수 있으며 현재 트랜잭션에 액세스할 수 있지만 SCA 트랜잭션 연계를 수정할 수 없습니다.

표 15. 요청 및 리턴 결과 호출

유형	통계	공식	ARM 트랜잭션
공통	TotalResponseTime	t ₂ - t ₀	X ₀ .X ₁
	RequestDeliveryTime	t' ₀ - t ₀	
	ResponseDeliveryTime	t ₂ - t' ₁	
	GoodRequests	Count _{EXIT}	
	BadRequests	Count _{FAILURE}	
	ResponseTime	t ₃ - t ₂	
참조	GoodRefRequest	Count _{EXIT}	X ₁ .X ₂
	BadRefRequests	Count _{FAILURE}	
	RefResponseTime	t' ₁ - t' ₀	

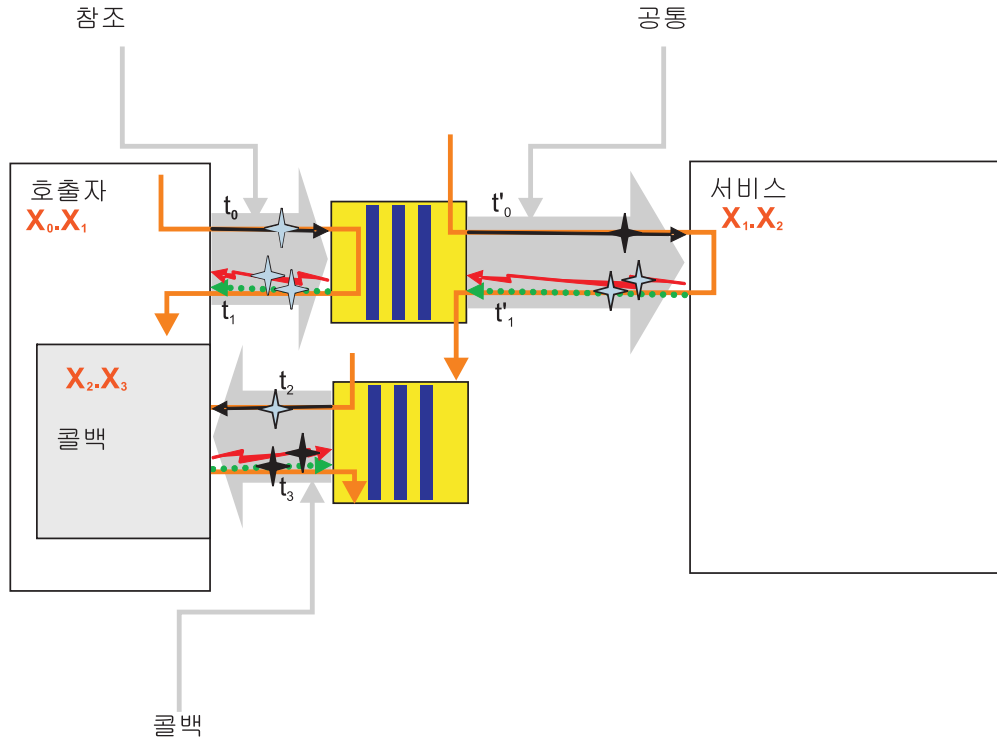


표 16. 콜백의 호출

유형	통계	공식	ARM 트랜잭션
콜백	GoodCB	$Count_{EXIT}$	X_1, X_3
	BadCB	$Count_{FAILURE}$	
	CBTime	$t_3 - t_2$	

비동기 구현의 비동기 콜백

ARM(Application Response Measurement) 통계는 비동기 구현의 다른 스레드를 사용하는 콜백 실행 및 콜백 요청에 사용 가능합니다.

매개변수

SCA(Service Component Architecture) 컴포넌트에 대한 이벤트 모니터링은 검은색



으로 표시되는 이벤트 지점을 포함하는 반면 파란색



으로 표시되는 이벤트 지점은 PMI/ARM 통계를 계산하고 시작하는 데에만 사용됩니다.

아래의 표와 다이어그램에서 "현재" ARM 트랜잭션(X_1 로 표시됨)은 호출 서비스 컴포넌트가 처음으로 호출되었을 때 작성됩니다. 호출자가 서비스 컴포넌트가 아닌 경우에는 현재 ARM 트랜잭션이 사용되거나 새 ARM 트랜잭션이 작성됩니다. 시작 트랜잭션이 아닌 경우 다음 표 및 다이어그램에서 X_n, X_{n+1} 표기법으로 표시된 대로 상위를 갖습니다. 표기법은 트랜잭션 연계를 표시하는 데 사용됩니다. 모든 SCA 호출은 새 트랜

작업을 시작하며 호출자의 현재 트랜잭션이 이 트랜잭션의 상위가 됩니다. 새 트랜잭션을 작성할 수 있으며 현재 트랜잭션에 액세스할 수 있지만 SCA 트랜잭션 연계를 수정할 수 없습니다.

표 17. 요청 및 리턴 결과 호출

유형	통계	공식	ARM 트랜잭션
공통	TotalResponseTime	$t_2 - t_0$	$X_0 \cdot X_1$
	RequestDeliveryTime	$t'_0 - t_0$	$X_1 \cdot X_2$
	ResponseDeliveryTime	$t_2 - t'_2$	
	GoodRequests	$\text{Count}_{\text{EXIT}}$	
	BadRequests	$\text{Count}_{\text{FAILURE}}$	
	ResponseTime	$t'_3 - t'_0$	
참조 A	GoodRefRequest	$\text{Count}_{\text{EXIT}}$	
	BadRefRequests	$\text{Count}_{\text{FAILURE}}$	
	RefResponseTime	$t_1 - t_0$	
대상 A	GoodTargetSubmit	$\text{Count}_{\text{EXIT}}$	$X_1 \cdot X_2$
	BadTargetSubmit	$\text{Count}_{\text{FAILURE}}$	
	TargetSubmitTime	$t'_1 - t'_0$	

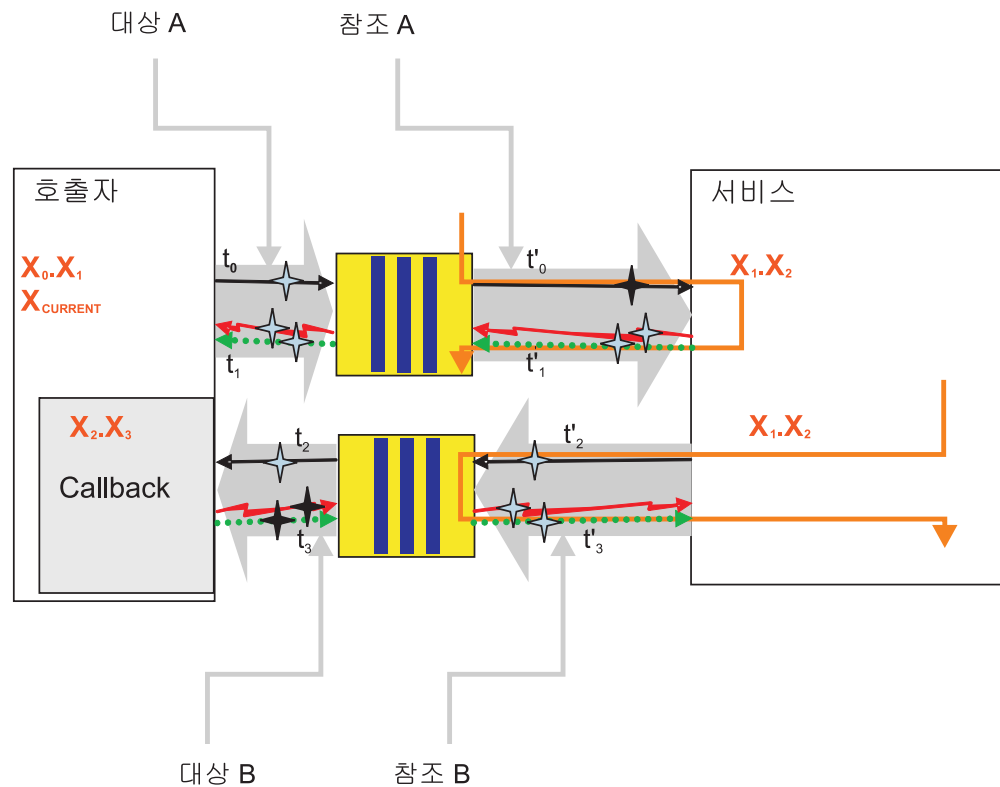


표 18. 콜백의 호출

유형	통계	공식	ARM 트랜잭션
참조 B	GoodCBSubmit	Count _{EXIT}	$X_1 \cdot X_2$
	BadCBSubmit	Count _{FAILURE}	
	CBSubmitTime	$t'_3 - t'_2$	
대상 B	GoodCB	Count _{EXIT}	$X_0 \cdot X_1$
	BadCB	Count _{FAILURE}	
	CBTime	$t_3 - t_2$	

동기 구현의 비동기 단방향

호출이 동기 구현으로 제출(가동 후 일임)될 때 이러한 ARM(Application Response Measurement) 통계를 가져올 수 있습니다.

매개변수

SCA(Service Component Architecture) 컴포넌트에 대한 이벤트 모니터링은 검은색



으로 표시되는 이벤트 지점을 포함하는 반면 파란색

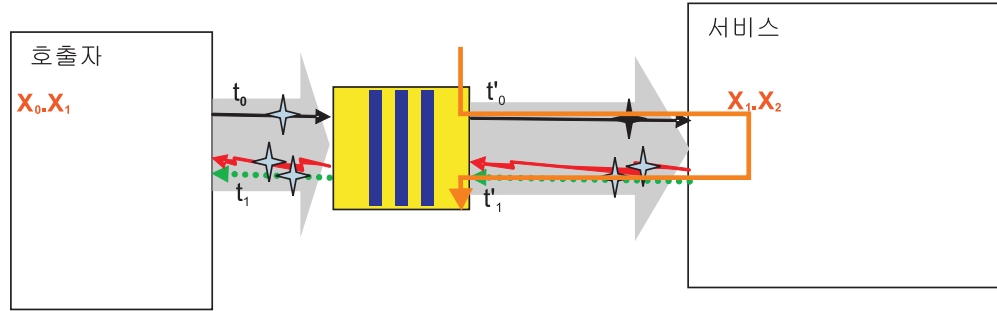


으로 표시되는 이벤트 지점은 PMI/ARM 통계를 계산하고 시작하는 데에만 사용됩니다.

아래의 표와 다이어그램에서 "현재" ARM 트랜잭션(X_1 로 표시됨)은 호출 서비스 컴포넌트가 처음으로 호출되었을 때 작성됩니다. 호출자가 서비스 컴포넌트가 아닌 경우에는 현재 ARM 트랜잭션이 사용되거나 새 ARM 트랜잭션이 작성됩니다. 시작 트랜잭션이 아닌 경우 다음 표 및 다이어그램에서 $X_n \cdot X_{n+1}$ 표기법으로 표시된 대로 상위를 갖습니다. 표기법은 트랜잭션 연계를 표시하는 데 사용됩니다. 모든 SCA 호출은 새 트랜잭션을 시작하며 호출자의 현재 트랜잭션이 이 트랜잭션의 상위가 됩니다. 새 트랜잭션을 작성할 수 있으며 현재 트랜잭션에 액세스할 수 있지만 SCA 트랜잭션 연계를 수정할 수 없습니다.

표 19. 요청 및 리턴 결과 호출

유형	통계	공식	ARM 트랜잭션
공통	TotalResponseTime	$t_1 - t_0$	$X_0 \cdot X_1$
	RequestDeliveryTime	$t'_0 - t_0$	$X_1 \cdot X_2$
	ResponseDeliveryTime	해당 없음	해당 없음
	GoodRequests	Count _{EXIT}	$X_1 \cdot X_2$
	BadRequests	Count _{FAILURE}	
	ResponseTime	$t'_1 - t'_0$	



비동기 구현의 비동기 단방향

비동기 구현으로 호출이 제출(가동 후 일임)된 경우 ARM(Application Response Measurement) 통계

매개변수

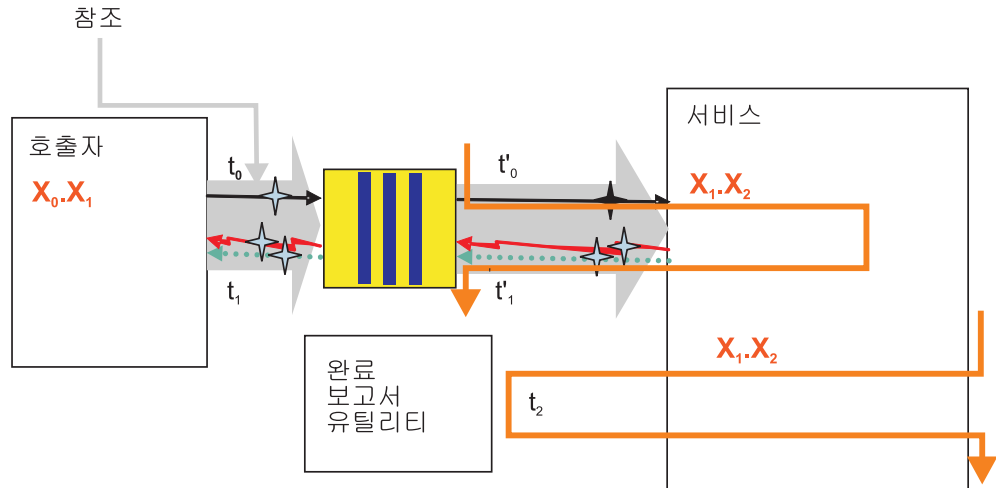
SCA(Service Component Architecture) 컴포넌트에 대한 이벤트 모니터링은 검은색

✦ 으로 표시되는 이벤트 지점을 포함하는 반면 파란색 ✦ 으로 표시되는 이벤트 지점은 PMI/ARM 통계를 계산하고 시작하는 데에만 사용됩니다.

아래의 표와 다이어그램에서 "현재" ARM 트랜잭션(X_1 로 표시됨)은 호출 서비스 컴포넌트가 처음으로 호출되었을 때 작성됩니다. 호출자가 서비스 컴포넌트가 아닌 경우에는 현재 ARM 트랜잭션이 사용되거나 새 ARM 트랜잭션이 작성됩니다. 시작 트랜잭션이 아닌 경우 상위가 있습니다. 이 관계는 다음 테이블과 다이어그램에서 $X_n \cdot X_{n+1}$ 로 표시됩니다. 표기법은 트랜잭션 연계를 표시하는 데 사용됩니다. 모든 SCA 호출은 새 트랜잭션을 시작하며 호출자의 현재 트랜잭션이 이 트랜잭션의 상위가 됩니다. 새 트랜잭션을 작성할 수 있으며 현재 트랜잭션에 액세스할 수 있지만 SCA 트랜잭션 연계를 수정할 수 없습니다.

표 20. 요청 및 리턴 결과 호출

유형	통계	공식	ARM 트랜잭션
공통	TotalResponseTime	$t_1 - t_0$	$X_0 \cdot X_1$
	RequestDeliveryTime	$t'_0 - t_0$	$X_1 \cdot X_2$
	ResponseDeliveryTime	해당 없음	해당 없음
	GoodRequests	$Count_{EXIT}$	$X_1 \cdot X_2$
	BadRequests	$Count_{FAILURE}$	
	ResponseTime	$t_2 - t_0$	
참조	GoodRefRequest	$Count_{EXIT}$	$X_0 \cdot X_1$
	BadRefRequest	$Count_{FAILURE}$	
	RefResponseDuration	$t_1 - t_0$	



서비스 컴포넌트 이벤트 모니터링

WebSphere Process Server 모니터링은 특정 이벤트 지점에서 서비스 컴포넌트에 있는 데이터를 캡처합니다. 로그 파일에서 각 이벤트를 보거나, Common Event Infrastructure 서버의 한층 기능이 많은 모니터링 성능을 사용할 수 있습니다.

프로세스 서버에서 전개된 응용프로그램은 응용프로그램이 실행되는 경우 모니터링 서비스 컴포넌트 이벤트의 스펙을 포함할 수 있습니다. WebSphere Integration Developer를 사용하여 응용프로그램을 개발한 경우, 서비스 컴포넌트 이벤트를 지정하여 계속 모니터링할 수 있습니다. 이 스펙은 응용프로그램의 일부로 포함되며, 응용프로그램이 전개될 때 프로세스 서버가 읽은 .mon 확장자의 파일 양식으로 제공됩니다. 응용프로그램이 시작되면 .mon 파일에 지정된 서비스 컴포넌트의 모니터링을 끌 수 없습니다. WebSphere Process Server에 대한 문서에는 이와 같은 유형의 연속 모니터링에 대해 설명이 없습니다. 이 주제에 대한 자세한 정보는 WebSphere Integration Developer 문서를 참조하십시오.

응용프로그램의 .mon 파일에 아직 지정되지 않은 서비스 컴포넌트 이벤트를 모니터링하려면 WebSphere Process Server를 사용하면 됩니다. 이벤트 모니터 출력을 로그 파일로 보내거나 Common Event Infrastructure 서버 데이터베이스로 보내도록 프로세스 서버를 구성할 수 있습니다. 모니터 대상 이벤트는 공통 기본 이벤트 표준을 사용하여 형식화되지만, 각각의 이벤트에 포함된 정보의 양을 조정할 수 있습니다. 문제점을 진단하거나 응용프로그램의 프로세스 플로우를 분석하거나 응용프로그램 사용 방법을 감사하려면 WebSphere Process Server에서 모니터링 기능을 사용하십시오.

비즈니스 프로세스 및 휴먼 타스크 이벤트의 모니터링 사용

비즈니스 프로세스 및 휴먼 타스크 서비스 컴포넌트의 실제 모니터링을 수행하기 전에 이러한 종류의 서비스 컴포넌트 모니터링을 지원하도록 WebSphere Process Server를 구성해야 합니다.

시작하기 전에

비즈니스 프로세스 컨테이너 및 휴먼 태스크 컨테이너를 프로세스 서버에 이전에 작성했어야 합니다.

이 태스크 정보

이 태스크를 수행하여 WebSphere Process Server의 Common Event Infrastructure 모니터링 지원을 사용 가능하게 하십시오.

프로시저

1. 관리 콘솔을 여십시오.
2. Business Process Choreographer가 단일 서버에 구성된 경우 다음 단계를 완료하여 서버에서 비즈니스 프로세스 이벤트를 생성할 수 있게 하십시오.
 - a. 휴먼 태스크 관리자에 대한 비즈니스 프로세스 이벤트를 사용하려면 서버 → 서버 유형 → **WebSphere Application Server** → *server_name*을 클릭한 후 비즈니스 통합 아래의 구성 탭에서 **Business Process Choreographer**를 펼치고 휴먼 태스크 관리자를 클릭하십시오. 상태 옵저버 절에서 **Common Event Infrastructure** 로깅 사용, 감사 로깅 사용 및 태스크 히스토리 사용에 대한 상자가 선택되었는지 확인하십시오. 선택란이 선택되지 않은 경우 선택란을 선택하고 서버를 다시 시작하십시오.
 - b. 비즈니스 플로우 관리자에 대한 비즈니스 프로세스 이벤트를 사용하려면 서버 → 서버 유형 → **WebSphere Application Server** → *server_name*을 클릭한 후 비즈니스 통합 아래의 구성 탭에서 **Business Process Choreographer**를 펼치고 비즈니스 플로우 관리자를 클릭하십시오. 상태 옵저버 절에서 **Common Event Infrastructure** 로깅 사용 및 감사 로깅 사용에 대한 상자가 선택되었는지 확인하십시오. 선택란이 선택되지 않은 경우 선택란을 선택하고 서버를 다시 시작하십시오.
3. Business Process Choreographer가 클러스터에 구성된 경우 다음 단계를 완료하여 클러스터에서 비즈니스 프로세스 이벤트를 생성할 수 있게 하십시오.
 - a. 휴먼 태스크 관리자에 대한 비즈니스 프로세스 이벤트를 사용하려면 서버 → 클러스터 → **WebSphere Application Server** 클러스터 → *cluster_name*을 클릭한 다음 비즈니스 통합 아래의 구성에서 **Business Process Choreographer**를 펼치고 **Common Event Infrastructure** 로깅 사용, 감사 로깅 사용 및 태스크 히스토리 사용 상자가 선택되었는지 확인하십시오. 선택란이 선택되지 않은 경우 선택란을 선택하고 서버를 다시 시작하십시오.
 - b. 비즈니스 플로우 관리자에 대한 비즈니스 프로세스 이벤트를 사용하려면 서버 → 클러스터 → **WebSphere Application Server** 클러스터 → *cluster_name*을 클릭한 후 비즈니스 통합 아래의 구성 탭에서 **Business Process Choreographer**를 펼치고 비즈니스 플로우 관리자를 클릭하십시오. 상태 옵저버 절에서

Common Event Infrastructure 로깅 사용 및 감사 로깅 사용에 대한 상자가 선택되었는지 확인하십시오. 선택란이 선택되지 않은 경우 선택란을 선택하고 서버를 다시 시작하십시오.

다음에 수행할 작업

선택란을 선택해야 한 경우에는 변경사항이 적용되도록 서버나 클러스터를 다시 시작해야 합니다.

서비스 컴포넌트 이벤트에 대한 로깅 구성

WebSphere Application Server의 로깅 기능을 사용하도록 선택해서 프로세스 서버 모니터링이 가능한 서비스 컴포넌트 이벤트를 캡처할 수 있습니다. 로거를 사용하여 응용 프로그램 처리에 대한 문제점을 진단할 때 이벤트의 데이터를 보십시오.

WebSphere Process Server는 기본 WebSphere Application Server의 광범위한 로깅 기능을 사용하여 서비스 컴포넌트 이벤트 지점에서 서버 모니터링에서 발생한 이벤트를 캡처할 수 있습니다. 관리 콘솔을 사용하여 모니터링하려는 특정 서비스 컴포넌트 이벤트 지점, 결과 서비스 컴포넌트 이벤트에 포함된 페이로드 세부사항의 양 및 결과를 공개하는 데 사용된 메소드(예: 특정 형식의 파일로 또는 콘솔에 직접)를 지정할 수 있습니다. 모니터 로그는 공통 기본 이벤트 형식으로 인코딩된 이벤트를 포함하며 이벤트 요소에 포함된 정보를 사용하여 서비스 컴포넌트 처리에 대한 문제점을 추적할 수 있습니다.

로깅 및 추적이 전체 제품에서 사용되는 방법에 대한 완전한 세부사항과 함께 WebSphere Application Server 로깅 및 추적 기능의 기능이 WebSphere Application Server 문서에 비교적 상세하게 문서화됩니다. 이 절은 WebSphere Process Server에 특정한 서비스 컴포넌트에 관련되어 있어서 로깅에 대한 보충 정보만을 제공합니다. 전체 제품의 기타 컴포넌트에 관한 로깅 및 추적 사용에 대해서는 WebSphere Application Server 문서의 정보를 참조하십시오.

진단 추적 서비스 사용

이 타스크를 사용하여 서비스 컴포넌트 이벤트에 포함된 세부사항의 양을 관리할 수 있는 로깅 서비스인 진단 추적 서비스를 사용 가능하게 하십시오.

시작하기 전에

Common Event Infrastructure 로깅 및 감사 로깅을 허용하도록 비즈니스 프로세스 및 휴먼 타스크 컨테이너를 구성해야 합니다.

이 태스크 정보

진단 추적 서비스는 서비스 컴포넌트 이벤트의 요소에 포함된 세부사항을 캡처하는 데 필요한 세부사항 레벨을 제공할 수 있는 유일한 로거 유형입니다. 이벤트를 로깅하려면

프로세스 서버를 시작하기 전에 진단 추적 서비스를 사용 가능하게 해야 합니다. 관리 콘솔을 통해 CEI 서버를 사용하여 모니터링하기 위한 서비스 컴포넌트 이벤트 지점을 선택하는 경우에도 서비스가 사용 가능해야 합니다.

프로시저

1. 탐색 분할창에서 서버 → 서버 유형 → **WebSphere Application Server**를 클릭하십시오.
2. 작업하려는 서버 이름을 클릭하십시오.
3. 문제점 해결 아래에서 진단 추적 서비스를 클릭하십시오.
4. 구성 탭에서 로그 사용을 선택하십시오.
5. 적용을 클릭한 다음 저장을 클릭하십시오.
6. 확인을 클릭하십시오.

다음에 수행할 작업

서버가 이미 시작되었으면 변경사항이 적용되도록 서버를 다시 시작해야 합니다.

관리 콘솔을 사용한 로깅 특성 구성

이 작업을 사용하여 모니터링 함수가 서비스 컴포넌트 이벤트를 로거 파일에 공개하도록 지정하십시오.

이 태스크 정보

응용프로그램이 모니터링 이벤트를 로그하려면 먼저 사용자가 모니터링하려는 서비스 컴포넌트 이벤트 지점, 각 이벤트에 필요한 세부사항 레벨 및 로그에 이벤트를 공개하는데 사용되는 출력 형식을 지정해야 합니다. 관리 콘솔을 사용하여 다음을 수행할 수 있습니다.

- 특정 이벤트 로그를 사용 가능 또는 불가능으로 설정합니다.
- 로그의 세부사항 레벨을 지정합니다.
- 로그 파일이 저장되는 위치, 보관되는 로그 파일 수 및 로그 출력의 형식을 지정합니다.

로그 구성을 정적 또는 동적으로 변경할 수 있습니다. 정적 구성 변경은 Application Server를 시작 또는 다시 시작할 때 응용프로그램에 영향을 줍니다. 동적 또는 런타임 구성 변경은 즉시 적용됩니다.

로그가 작성되면 이 로그의 레벨 값이 구성 데이터로부터 설정됩니다. 특정 로그 이름으로 사용할 구성 데이터가 없으면 로그의 상위에서 이 로그의 레벨을 얻습니다. 상위 로그에 대한 구성 데이터가 존재하지 않는 경우에는 레벨 값이 널(null)이 아닌 로그를 찾을 때까지 트리의 위쪽으로 이 로그의 상위를 검토합니다. 로그 레벨을 변경하면 로

그의 하위에 변경사항이 전파되고 필요에 따라 반복적으로 하위에 변경사항을 전파합니다.

프로시저

1. 로깅을 사용 가능하게 하고 로그의 출력 특성을 설정하십시오.
2. 탐색 분할창에서 서버 > 서버 유형 > **WebSphere Application Server**를 클릭하십시오.
3. 작업하려는 서버 이름을 클릭하십시오.
4. 문제점 해결 아래에서 로깅 및 추적을 클릭하십시오.
5. 로그 세부사항 레벨 변경을 클릭하십시오.
6. 컴포넌트, 패키지 및 그룹의 목록에는 실행 중인 서버에 현재 등록된 모든 컴포넌트가 표시됩니다. 한 번 이상 호출된 서버 이벤트만 이 목록에 나타납니다. 로그 될 수 있는 이벤트 지점이 있는 서버 컴포넌트는 모두 **WBILocationMonitor.LOG** 라는 이름으로 시작하는 컴포넌트 중 한 컴포넌트 아래에 표시됩니다.
 - 구성을 정적으로 변경할 이벤트를 선택하려면 구성 탭을 클릭하십시오.
 - 구성을 동적으로 변경할 이벤트를 선택하려면 런타임 탭을 클릭하십시오.
7. 로깅하려는 이벤트 또는 이벤트 그룹을 선택하십시오.
8. 각 이벤트 또는 이벤트 그룹에 대한 로깅 레벨을 설정하십시오.

주: FINE, FINER 및 FINEST 레벨만이 CEI 이벤트에 유효합니다.
9. 적용을 클릭하십시오.
10. 확인을 클릭하십시오.
11. 정적 구성 변경사항을 적용하려면 서버를 중지한 다음 다시 시작하십시오.

결과

기본적으로 로거(logger)는 `install_root/profiles/profile_name/logs/server_name` 폴더에 있는 `trace.log` 파일에 출력을 공개합니다.

학습서: 서비스 컴포넌트 이벤트 로깅

사용자가 모니터링하는 서비스 컴포넌트 이벤트 지점의 경우 기본 WebSphere Application Server의 로깅 기능에 이벤트를 공개할 수 있습니다. 이 학습서는 로깅을 포함한 모니터링 설정 예제 및 로그 파일에 저장된 이벤트를 보는 방법을 단계적으로 안내합니다.

이 예제에 대해 사용자가 수행할 시나리오는 서버에서 이미 전개되어 실행 중인 응용 프로그램에서 모니터링에 대한 서비스 컴포넌트 이벤트 지점을 선택하는 방법을 보여줍니다. 응용프로그램 처리가 해당 이벤트 지점 중 하나에 도달할 때마다 모니터링 기능이 이벤트를 발생하는 방법을 보여줍니다. 발생한 이벤트 각각은 직접 로그 파일에 XML 문자열로 공개되는 표준화된 공통 기본 이벤트 양식을 사용합니다.

이 학습서의 목표

이 학습서를 완료한 후에는 다음을 수행할 수 있습니다.

- 서버 로그에 공개된 출력으로, 모니터링 서비스 컴포넌트 이벤트 지점을 선택합니다.
- 로그 파일에서 저장된 이벤트를 봅니다.

이 학습을 완료하는 데 필요한 시간

이 학습서는 완료하는 데 대략 15 - 20분이 소요됩니다.

전제조건

이 학습서를 수행하려면 이미 다음을 수행한 상태여야 합니다.

- 서버 구성 및 시작
- Common Event Infrastructure 구성
- 서버에서 진단 추적 서비스 사용
- 서버에서 샘플 갤러리 응용프로그램 설치 및 시작
- 서버에서 비즈니스 규칙 샘플 응용프로그램 설치 및 시작. 비즈니스 규칙 샘플 응용 프로그램을 설정하고 실행하려면 샘플 갤러리 페이지에 있는 지시사항을 따르십시오.

이와 같은 모든 전제조건이 완료되고 난 후에는 학습서를 진행하기 전에 최소 한 번 샘플 갤러리에서 비즈니스 규칙 샘플 응용프로그램을 실행하십시오.

예제: 로그의 이벤트 모니터링:

로그를 사용한 모니터링을 위해 관리 콘솔을 사용하여 이벤트 유형의 세부사항을 관리할 수 있습니다. 이 예제는 일부 이벤트 유형에 대해 기록되는 세부사항의 레벨을 변경하고 개별 이벤트에 대한 정보를 보기 위해 문서 편집기를 사용하여 `trace.log` 파일을 열기 위한 콘솔 사용을 보여줍니다.

이 태스크 정보

이 시나리오의 비즈니스 규칙 샘플 응용프로그램을 사용하므로 웹 페이지에 이 응용프로그램이 이미 열려 있어야 합니다. 모니터링 매개변수를 지정한 후 샘플을 실행해야 하므로 응용프로그램을 계속 열어 놓으십시오. 샘플을 최소 한번은 이미 실행했는지 확인하십시오. 실행한 경우 모니터링하려고 선택할 수 있는 함수 목록에 샘플이 나타납니다.

프로시저

1. 관리 콘솔을 여십시오.
2. 탐색 분할창에서 서버 → **Application Server**를 클릭하십시오.
3. `server_name`을 클릭하십시오.
4. 문제점 해결 아래에서 로깅 및 추적을 클릭하십시오.

5. 로그 세부사항 레벨 변경을 클릭하십시오.
6. 런타임 탭을 선택하십시오.
7. **WBILocationMonitor.LOG.BR**에 대한 트리를 펼치십시오. 그러면 **WBILocationMonitor.LOG.BR.brsample.*** 요소 아래에 7개의 이벤트 유형이 나타납니다. 첫 번째 이벤트는 **WBILocationMonitor.LOG.BR.brsample_module.DiscountRuleGroup**이며 다음 네이처가 있는 **Operation._calculateDiscount**라는 단일 함수를 포함합니다.
 - ENTRY
 - EXIT
 - FAILURE
 - SelectionKeyExtracted
 - TargetFound
8. 각 이벤트를 클릭하고 **finest**를 선택하십시오.
9. 확인을 클릭하십시오.
10. 비즈니스 규칙 샘플 응용프로그램 페이지를 전환하고 응용프로그램을 한번 실행하십시오.
11. 문서 편집기를 사용하여 시스템의 *profile_root/logs/server_name* 폴더에 있는 *trace.log* 파일을 여십시오.

결과

샘플 응용프로그램을 실행했을 때 모니터가 가동한 비즈니스 규칙 이벤트가 포함된 로그의 행이 표시되어야 합니다. 유의해야 할 주요 사항은 공통 기본 이벤트 표준에 따른 구문 분석되지 않은 긴 XML 문자열로 출력이 이루어진다는 사실입니다. ENTRY 및 EXIT 이벤트를 검토하십시오. 그러면 **finest** 세부사항 레벨을 선택했기 때문에 포함된 비즈니스 오브젝트가 16진 형식으로 인코딩된 것을 알 수 있습니다. XML을 읽기 쉬운 테이블로 구문 분석하고 비즈니스 오브젝트 데이터를 읽기 쉬운 형식으로 해독하는, Common Event Infrastructure 서버에 공개된 이벤트와 이 출력을 비교해보십시오. 이 학습을 거슬러 올라가서 세부사항 레벨을 **finest**에서 **fine** 또는 **finer**로 변경한 후 이벤트 간의 차이점을 비교해 볼 수 있습니다.

학습을 완료한 후에는 로그에 모니터할 서비스 컴포넌트 이벤트 지점을 선택하는 방법을 이해해야 합니다. 이 유형의 모니터링에 가동된 이벤트가 표준 형식이며 결과는 로그 파일에 직접 원시 XML 형식의 문자열로 공개되는 것을 보았습니다. 공개된 이벤트를 보려면 문서 편집기에서 로그 파일을 열고 개별 이벤트의 콘텐츠를 해독하십시오.

다음에 수행할 작업

비즈니스 규칙 샘플 응용프로그램을 더 이상 모니터링하지 않으려면 여기에 설명된 단계를 거슬러 올라가서 샘플 이벤트의 세부사항 레벨을 **info**로 재설정할 수 있습니다.

비즈니스 규칙 및 선택기에 대한 감사 로깅

비즈니스 규칙 및 선택기에 대한 변경사항을 자동으로 로깅하고 변경하도록 WebSphere Process Server을 설정할 수 있습니다.

비즈니스 규칙 및 선택기가 변경되는 시기를 자동으로 감지해서 로그 파일에 변경사항이 자세히 설명된 항목을 작성하도록 서버를 구성할 수 있습니다.

표준 JVM SystemOut.log 파일이나 선택한 사용자 정의 감사 로그 파일에 로그 항목을 쓰도록 선택할 수 있습니다. 변경이 작성된 방식에 따라 각 비즈니스 규칙 또는 선택기 변경이 작성되는 프로세스 서버가 다음을 로깅합니다.

- 변경을 수행한 사용자의 이름
- 변경 요청이 발생한 위치
- 이전 비즈니스 규칙 또는 선택기 오브젝트
- 이전 오브젝트를 바꾸는 새 비즈니스 규칙 또는 선택기

비즈니스 규칙 및 선택기 오브젝트는 완전한 비즈니스 규칙 세트, 의사결정 테이블, 비즈니스 규칙 그룹 또는 교체되는 비즈니스 규칙 및 선택기와 이를 바꾼 새 버전 모두에 대한 선택기입니다. 로그를 확인해서(감사 출력은 Common Event Infrastructure 데이터베이스에 지정할 수 없음) 이전 및 새 비즈니스 규칙 또는 선택기를 비교하여 작성된 변경사항을 판별할 수 있습니다. 다음 시나리오는 로깅이 발생하는(구성된 경우) 상황 및 로그 항목의 콘텐츠를 설명합니다.

시나리오	결과	로그 항목 콘텐츠
비즈니스 규칙 관리자를 사용하여 비즈니스 규칙을 공개함	요청	사용자 ID, 서버 이름(적용 가능한 경우 셀과 노드 포함), 이전 비즈니스 규칙 규칙 세트, 새 규칙 세트
	장애	사용자 ID, 서버 이름(적용 가능한 경우 셀과 노드 포함), 이전 비즈니스 규칙 규칙 세트, 새 규칙 세트
저장소 데이터베이스 갱신 및 커밋(비즈니스 규칙 관리자를 사용한 공개 시도에서)	성공	사용자 ID, 이전 규칙 세트, 새 규칙 세트
	장애	사용자 ID, 새 규칙 세트
선택기 또는 비즈니스 규칙 그룹 내보내기	요청	사용자 ID, 선택기 또는 비즈니스 규칙 그룹 이름
	성공	사용자 ID, 서버 이름(적용 가능한 경우 셀과 노드 포함), 내보낸 선택기 또는 비즈니스 규칙 그룹의 사본
	장애	사용자 ID, 서버 이름(적용 가능한 경우 셀과 노드 포함), 선택기 또는 비즈니스 규칙 그룹 이름

시나리오	결과	로그 항목 콘텐츠
선택기 또는 비즈니스 규칙 그룹 가져오기	요청	사용자 ID, 새 선택기 또는 비즈니스 규칙 그룹의 사본
	성공	사용자 ID, 서버 이름(적용 가능한 경우 셀과 노드 포함), 가져온 선택기 또는 비즈니스 규칙 그룹의 사본, 가져온 버전으로 바꾼 선택기 또는 비즈니스 규칙 그룹의 사본
	장애	사용자 ID, 서버 이름(적용 가능한 경우 셀과 노드 포함), 가져오려 한 선택기 또는 비즈니스 규칙 그룹의 사본
응용프로그램 설치	성공	사용자 ID, 서버 이름(적용 가능한 경우 셀과 노드 포함), 선택기 또는 비즈니스 규칙 그룹 이름
	장애	사용자 ID, 서버 이름(적용 가능한 경우 셀과 노드 포함), 선택기 또는 비즈니스 규칙 그룹 이름
응용프로그램 갱신(관리 콘솔 또는 wsadmin 명령을 통해)	성공	사용자 ID, 서버 이름(적용 가능한 경우 셀과 노드 포함), 새 선택기 또는 비즈니스 규칙 그룹의 사본, 이전 선택기 또는 비즈니스 규칙 그룹의 사본
	장애	사용자 ID, 서버 이름(적용 가능한 경우 셀과 노드 포함), 새 선택기 또는 비즈니스 규칙 그룹의 사본
기존 비즈니스 규칙, 선택기 또는 둘 모두가 있는 이전에 전개된 응용프로그램이 시작됨	성공	서버 이름(적용 가능한 경우 셀과 노드 포함), 선택기 또는 비즈니스 규칙 그룹의 사본
	장애	서버 이름(적용 가능한 경우 셀과 노드 포함), 선택기 또는 비즈니스 규칙 그룹의 사본

Common Event Infrastructure 서버로 서비스 컴포넌트 모니터링

서비스 컴포넌트 모니터링 결과를 CEI(Common Event Infrastructure) 서버에 공개하도록 선택할 수 있습니다. 서비스 컴포넌트 이벤트 지점은 응용프로그램 플로우를 보고 관리하기 위해 영구적으로 또는 문제점을 해결하기 위해 임시로 CEI 서버로 모니터링 하도록 지정될 수 있습니다.

모니터링을 사용하여 CEI 버스에 발생된 서비스 컴포넌트 이벤트 내 서비스 컴포넌트 이벤트 지점의 데이터를 공개할 수 있습니다. 이 모니터링 접근법은 시스템의 서비스 컴포넌트 활동을 보다 유연하게 분석하도록 합니다. 공통 기본 이벤트 브라우저처럼 CEI 이벤트에 최적화된 브라우저를 사용할 수도 있습니다.

이벤트는 로거에 전송된 이벤트와 동일하게 구조화되지만 서비스 컴포넌트 이벤트를 분석하기 위해 특별히 정의된 표시기로 액세스할 수 있는 데이터베이스에 저장됩니다. 응용프로그램이 서버에 전개되어 실행된 이후 항상 계속해서 모니터링하도록(『정적』 모니터링으로 알려진 메소드) 응용프로그램이 작성될 때 응용프로그램 내에 서비스 컴포넌트 이벤트 지점을 지정할 수 있습니다. 시스템에서 컴포넌트 처리의 적절한 플로우에서 특별히 중요한 서비스 컴포넌트 이벤트 지점에 대한 정적 모니터링을 수행합니다. 이 정보로 시스템에서 실행하는 서비스 컴포넌트 프로세스 간의 상호작용 및 조치를 쉽게 감독할 수 있습니다. 서비스 컴포넌트가 제대로 작동 중이 아님을 나타낼 수 있는 이러한 프로세스의 정상 플로우 이탈을 빠르게 발견할 수도 있습니다.

서비스 컴포넌트의 정적 모니터링을 구성하려면 WebSphere Integration Developer를 사용하여 응용프로그램에서 서비스 컴포넌트 이벤트 지점을 선택합니다. 선택사항은 응용프로그램과 함께 전개될 확장자가 .mon인 XML 파일의 양식으로 지정됩니다. 실행 중인 서버에 전개된 후에는 응용프로그램의 .mon 파일에 지정된 이벤트에 대한 모니터링의 세부사항 레벨을 끄거나 낮출 수 없습니다. 이 모니터링을 중지하려면 서버를 중지하고 응용프로그램 전개를 취소해야 합니다.

실행 중인 서버에 이미 전개된 응용프로그램에서 사용 가능 및 불가능으로 설정할 수 있는 『동적』 모니터링에 대한 서비스 컴포넌트 이벤트 지점을 선택할 수도 있습니다. CEI 서버를 사용한 동적 모니터링 수행 이론은 본질적으로 시스템의 문제점을 진단해서 해결하는 로깅에 대한 이론과 동일합니다. CEI 버스를 통해 가동된 각 이벤트에 대한 구조를 구성하는 공통 기본 이벤트 요소와 함께 출력은 본래 로거(logger)에 공개된 출력과 동일합니다. 또한 로깅 데이터와 마찬가지로 세부사항 레벨의 차이는 페이로드가 이벤트 내에 인코딩되는 정도에만 영향을 줍니다.

관리 콘솔을 사용한 서비스 컴포넌트 이벤트 모니터링 구성

관리 콘솔을 사용하여 Common Event Infrastructure 서버에 서비스 컴포넌트 이벤트를 공개할 모니터링 함수를 동적으로 지정하십시오.

시작하기 전에

로거의 경우처럼 진단 추적 서비스를 사용 가능하게 해야 합니다. 서버를 다시 시작한 후 모니터링하려는 이벤트를 한번 호출합니다. 그러면 모니터링할 수 있는 이벤트의 목록이 나타납니다.

이 태스크 정보

모니터할 이벤트를 선택하는 이 메소드는 프로세스 서버에 이미 전개된 응용프로그램에 사용됩니다. 프로세스 서버의 응용프로그램으로 전개되는 .mon 파일에 지정된 이벤트는 여기에서 수행한 변경사항과 무관하게 Common Event Infrastructure 데이터베이스에서 모니터링됩니다. 이러한 이벤트의 경우 캡처하여 CEI 데이터베이스에 공개할 보다

높은 레벨의 세부사항만을 지정할 수 있습니다. CEI 데이터베이스에 공개되는 출력은 로거가 공개하는 출력과 매우 유사합니다.

프로시저

1. 관리 콘솔에서 문제점 해결 > 로깅 및 추적을 클릭하십시오.
2. 로그 세부사항 레벨 변경을 클릭하십시오.
3. 컴포넌트, 패키지 및 그룹의 목록은 실행 중인 서버에 현재 등록된 모든 컴포넌트를 표시합니다. 적어도 한번은 호출된 프로세스 서버 이벤트만이 이 목록에 나타납니다. 로그될 수 있는 모드 프로세스 서버 이벤트는 **WBILocationMonitor.CEI** 이름으로 시작하는 컴포넌트 중 한 컴포넌트 아래에 표시됩니다.
 - 구성에 대한 정적 변경을 수행하려면 구성 탭을 클릭하십시오.
 - 구성을 동적으로 변경하려면 런타임 탭을 클릭하십시오.
4. 모니터링할 이벤트 또는 이벤트 그룹을 선택하십시오.
5. 각 이벤트에 대해 캡처하려는 세부사항 레벨을 클릭하십시오.

주: FINE, FINER 및 FINEST 레벨만이 CEI 이벤트에 유효합니다.
6. 적용을 클릭한 다음 저장을 클릭하십시오.
7. 확인을 클릭하십시오.
8. 구성을 정적으로 변경한 경우 변경사항을 적용하려면 프로세스 서버를 다시 시작해야 합니다.

결과

공통 기본 이벤트 브라우저에서 모니터링한 이벤트 결과를 볼 수 있습니다.

학습서: 이벤트 모니터링에 Common Event Infrastructure 서버 사용

이 학습서는 CEI 서버를 사용한 모니터링 설정 예제 및 데이터에 저장된 이벤트를 보는 방법을 단계적으로 안내합니다.

사용자가 모니터링하는 서비스 컴포넌트 이벤트 지점의 경우 이벤트를 Common Event Infrastructure 서버에 공개하고 CEI 서버 데이터베이스에 저장할 수 있습니다. 이벤트가 캡처되면 공통 기본 이벤트를 사용하여 저장된 해당 이벤트를 보십시오. 이 시나리오에서 사용하는 예제는 .mon 파일로 전개된 응용프로그램이 계속 특정 서비스 컴포넌트 이벤트 지점을 모니터링하는 정적 모니터링이 포함되지 않습니다. 정적 모니터링을 수행하는 방법에 대한 정보는 IBM® WebSphere Integration Developer Information Center를 참조하십시오.

대신, 이 예제에 대해 사용자가 수행할 시나리오는 서버에서 이미 전개되어 실행 중인 응용프로그램에 있는 서비스 컴포넌트에 대한 이벤트 지점 모니터링을 위한 선택 방법을 보여줍니다. 응용프로그램 처리가 해당 이벤트 지점 중 하나에 도달할 때마다 모니

터링 기능이 이벤트를 발생하는 방법을 볼 수 있습니다. 발생한 이벤트 각각은 해당 데이터베이스에 대한 이벤트 정보를 저장할 CEI 서버에 공개됩니다. 그런 다음 공통 기본 이벤트를 사용해서 이벤트를 보십시오.

이 학습서의 목표

이 학습서를 완료한 후에는 다음을 수행할 수 있습니다.

- CEI 서버에 공개된 이벤트로, 모니터링 서비스 컴포넌트 이벤트 지점을 선택합니다.
- 공통 기본 이벤트 브라우저에서 저장된 이벤트를 봅니다.

이 학습을 완료하는 데 필요한 시간

이 학습서는 완료하는 데 대략 15 - 20분이 소요됩니다.

전제조건

이 학습서를 수행하려면 이미 다음을 수행한 상태여야 합니다.

- 서버 구성 및 시작
- CEI 및 해당 데이터베이스 구성
- 서버에서 진단 추적 서비스 사용
- 서버에서 샘플 갤러리 응용프로그램 설치 및 시작
- 서버에서 비즈니스 규칙 샘플 응용프로그램 설치 및 시작. 비즈니스 규칙 샘플 응용 프로그램을 설정하고 실행하려면 샘플 갤러리 페이지에 있는 지시사항을 따르십시오.

이와 같은 모든 전제조건이 완료되고 난 후에는 학습서를 진행하기 전에 최소 한 번 샘플 갤러리에서 비즈니스 규칙 샘플 응용프로그램을 실행하십시오.

예제: Common Event Infrastructure 서버로 모니터링:

CEI 서버를 사용한 모니터링을 위해 관리 콘솔을 사용하여 이벤트 유형에 대한 세부사항을 관리하고 공통 기본 이벤트 브라우저에 기록된 이벤트를 표시할 수 있습니다. 이 예제는 일부 이벤트 유형에 대해 기록된 세부사항의 레벨을 변경하고 공통 기본 이벤트 브라우저를 사용하여 개별 이벤트의 정보를 보기 위한 콘솔 사용을 보여줍니다.

이 태스크 정보

이 시나리오의 비즈니스 규칙 샘플 응용프로그램을 사용하므로 이 응용프로그램을 포함하는 웹 페이지가 이미 열려 있어야 합니다. 모니터링 매개변수를 지정한 후 샘플을 실행해야 하므로 응용프로그램을 계속 열어 놓으십시오. 샘플을 최소 한번은 이미 실행했는지 확인하십시오. 실행한 경우 모니터링하려고 선택할 수 있는 함수 목록에 샘플이 나타납니다.

프로시저

1. 관리 콘솔을 여십시오.
2. 탐색 분할창에서 서버 → 서버 유형 → **WebSphere Application Server**를 클릭하십시오.
3. *server_name*을 클릭하십시오.
4. 문제점 해결 아래에서 로깅 및 추적을 클릭하십시오.
5. 로그 세부사항 레벨 변경을 클릭하십시오.
6. 런타임 탭을 선택하십시오.
7. **WBILocationMonitor.CEI.BR**에 대한 트리를 펼치십시오. 그러면 **WBILocationMonitor.CEI.BR.brsample.*** 요소 아래에 7가지 이벤트 유형이 나타납니다. 각 이벤트 유형은 **Operation._calculateDiscount** 함수로 추가된 **WBILocationMonitor.CEI.BR.brsample_module.DiscountRuleGroup** 이름 및 다음 네이처를 포함합니다.
 - ENTRY
 - EXIT
 - FAILURE
 - SelectionKeyExtracted
 - TargetFound
8. 각 이벤트를 클릭하고 **finest**를 선택하십시오.
9. 확인을 클릭하십시오.
10. 비즈니스 규칙 샘플 응용프로그램 페이지를 전환하고 응용프로그램을 한번 실행하십시오.
11. 관리 콘솔로 돌아가서 탐색 분할창에서 통합 응용프로그램 → 공통 기본 이벤트 브라우저를 선택하십시오.
12. Network Deployment 환경 내의 노드에서 서버를 실행 중이면 서버 및 노드의 이름을 포함하도록 **Event Data Store** 필드를 수정해야 할 수 있습니다. 'cell/nodes/node_name/servers/server_name/ejb/com/ibm/events/access/EventAccess' 양식으로 문자열을 입력하십시오.
13. 이벤트 가져오기를 누르십시오.

결과

이제 샘플 응용프로그램을 실행했을 때 CEI 서버에 공개된 네 가지 비즈니스 규칙 이벤트의 공통 기본 이벤트 브라우저의 위쪽 분할창에 목록이 표시되어야 합니다. 이벤트 중 하나를 선택하십시오. 그러면 아래쪽 분할창에 이벤트의 콘텐츠가 표시됩니다. 로그에 공개된 이벤트와 이 이벤트를 비교해보십시오. 브라우저는 CEI 서버에 공개된 원래 XML 문자열을 구문 분석했으며 ENTRY 및 EXIT 이벤트의 비즈니스 오브젝트 코드

가 원래의 16진 형식에서 읽기 쉬운 XML로 변환되었음에 유의하십시오. 이 학습을 거슬러 올라가서 세부사항 레벨을 **finest**에서 **fine** 또는 **finer**로 변경한 후 이벤트 간의 차이점을 비교해 볼 수 있습니다.

학습을 완료한 후에는 CEI 서버를 사용하여 모니터링 서비스 컴포넌트 이벤트 지점을 선택하는 방법을 이해해야 합니다. 이 유형의 모니터링에 가동된 이벤트가 표준 형식이며 데이터베이스에 결과가 공개되는 것을 보았습니다. 공통 기본 이벤트 브라우저를 사용하여 데이터베이스에서 이벤트를 검색하고 개별 이벤트에 대한 정보를 브라우저의 구분 분석된 테이블 형식으로 볼 수 있어야 합니다.

다음에 수행할 작업

비즈니스 규칙 샘플 응용프로그램을 더 이상 모니터링하지 않으려면 여기에 설명된 단계를 거슬러 올라가서 샘플 이벤트의 세부사항 레벨을 **info**로 재설정할 수 있습니다.

세션 모니터링

공통 기본 이벤트 브라우저를 사용하여 동일한 세션 ID 속성을 포함하는 Common Event Infrastructure 데이터베이스에서 모든 이벤트를 찾아서 동일한 세션의 일부인 여러 이벤트를 모니터링할 수 있습니다.

WebSphere Process Server에는 단일 세션의 파트인 모든 서비스 컴포넌트 이벤트를 식별할 수 있는 향상된 기능이 있습니다. 공통 기본 이벤트의 표준 요소는 contextDataElement 요소 아래에 WBISESSIONID라는 속성을 포함합니다. 세션의 일부였던 모든 서비스 컴포넌트에 대해 개별 세션의 고유 ID는 이 속성에 저장됩니다. 공통 기본 이벤트 브라우저의 **SessionID** 필드를 사용하여 Common Event Infrastructure 데이터베이스에 저장된 이벤트 중 지정한 세션 ID에 일치하는 이벤트를 찾을 수 있습니다. 이 기능을 통해 모든 서비스 컴포넌트 이벤트의 콘텐츠 및 프로세스 플로우를 쉽게 검토할 수 있습니다. 이 정보를 사용하여 응용프로그램의 효율성을 평가하고 특정 상황에서만 발생하는 문제점을 진단할 수 있습니다.

공통 기본 이벤트 브라우저에서 이벤트 및 연관된 콘텐츠의 리턴된 목록을 볼 수 있습니다. 모든 이벤트 보기를 클릭하면 링크 열에서 이벤트에 대한 자세한 정보를 확인할 수 있습니다. 특정 이벤트에 실패 열의 링크가 있는 경우 이 링크를 클릭하여 실패 이벤트에 대한 내용을 보다 자세히 볼 수 있습니다. 이와 유사하게 특정 이벤트와 연관된 **Business Process**에 링크가 있는 경우 해당 링크를 클릭해서 Business Process Choreographer Explorer를 열고 Business Process 또는 휴먼 태스크 이벤트에 대한 자세한 정보를 볼 수 있습니다.

관련 개념

3 페이지의 『모니터링 사용 방법』

수행할 계획 중인 모니터링의 유형에 따라 모니터링 서비스 컴포넌트 이벤트 지점을 지정하는 데 사용할 수 있는 여러 메소드가 있습니다.

제 3 장 모니터한 이벤트 보기

사용 중인 모니터링의 유형에 따라 모니터한 이벤트의 공개된 결과를 볼 수 있는 여러 방법이 있습니다. 이 절에서는 Common Event Infrastructure 데이터베이스에 저장된 성능 데이터, 이벤트 로그 및 서비스 컴포넌트 이벤트를 보기 위해 사용할 수 있는 메소드를 제시합니다.

Tivoli Performance Viewer로 성능 매트릭스 보기

Tivoli Performance Viewer를 사용하여 성능 모니터링을 시작 및 중지하고 PMI 데이터가 시스템에 발생할 때 PMI를 도표나 표 형식으로 보며 선택적으로 나중에 동일한 표시기에서 검토할 수 있는 파일에 데이터를 로깅할 수 있습니다.

시작하기 전에

Tivoli Performance Viewer를 사용해서 성능 매트릭스를 보려면 다음 조건에 해당되어야 합니다.

- 모니터할 서버가 노드에서 실행 중이어야 함
- PMI가 사용 가능으로 설정됨
- 모니터할 서비스 컴포넌트 이벤트 지점이 한 번 이상 호출되어 해당 지점을 표시기에서 선택할 수 있음.

이 태스크 정보

TPV(Tivoli Performance Viewer)는 서버의 성능에 대한 다양한 세부사항을 볼 수 있도록 하는 강력한 응용프로그램입니다. WebSphere Application Server Information Center의 『Tivoli Performance Viewer로 성능 모니터링』이란 제목의 절에는 이 프로그램 사용에 대한 완전한 지시사항 자원을 포함해서 다양한 목적을 위한 이 도구 사용 방법에 대한 세부사항이 포함됩니다. 이 절에서는 WebSphere Process Server 특정 이벤트에 대한 성능 데이터를 보는 방법에 대해서만 설명합니다.

성능 표시기를 사용하면 관리자와 프로그래머가 WebSphere Process Server의 현재 성능 상태를 모니터할 수 있습니다. 데이터를 수집하고 보는 것은 프로세스 서버에서 발생하기 때문에 성능에 영향이 미칩니다. 성능 영향을 최소화하려면 활동을 모니터하려는 서버만을 모니터하십시오.

주: 이러한 통계를 볼 때 카운터 유형 통계와 지속 기간 유형 통계를 혼합하지 마십시오. 카운터는 누적되며 그래프로 표시되는 비율이 응용프로그램에 따라 빠르게 증가할 수 있습니다. 이와 반대로 지속 기간 통계는 시스템이 각 이벤트를 처리하는 데 걸리는

평균 시간을 나타내기 때문에 일정 범위 내에서 유지됩니다. 결과적으로 통계 및 상대 배율 간의 불일치로 인해 하나 또는 기타 유형의 통계가 표시기 그래프에서 부정확하게 표시될 수 있습니다.

프로시저

• 현재 성능 활동 보기

1. 관리 콘솔 탐색 트리에서 **모니터링 및 조정** → **성능 표시기** → **현재 활동을 클릭** 하십시오.
2. 서버를 선택한 후 활동을 모니터링할 서버의 이름을 클릭하십시오. 대신 활동을 모니터링할 서버의 선택란을 체크한 후 **모니터링 시작**을 클릭하십시오. 동시에 여러 서버에 대한 모니터링을 시작하려면 서버를 선택한 후 **모니터링 시작**을 클릭하십시오.
3. **성능 모듈**을 선택하십시오.
4. 보려는 각 성능 모듈의 이름 옆에 있는 선택란을 선택하십시오. 성능 통계를 생성하며 한 번 이상 호출된 WebSphere Process Server 이벤트가 **WBISStats.RootGroup** 계층 구조 아래에 표시됩니다. 노드 옆의 **+**를 클릭해서 트리를 펼치고 노드 옆의 **-**를 클릭해서 노드를 접으십시오.
5. **모듈 보기**를 클릭하십시오. 요청한 데이터가 제공된 도표나 테이블이 페이지의 오른쪽에 표시됩니다. 도표는 기본적으로 표시됩니다.

각 모듈에는 모듈과 연관된 여러 카운터가 있습니다. 이 카운터는 데이터 도표 또는 테이블 아래의 테이블에 표시됩니다. 선택한 카운터는 도표나 테이블에 표시됩니다. 도표나 테이블 옆의 선택란을 선택하거나 지워서 도표 또는 테이블에서 카운터를 추가하거나 제거할 수 있습니다. 기본적으로 각 모듈의 처음 세 개의 카운터가 표시됩니다.

최대 20개의 카운터를 선택해서 TPV에 현재 활동 모드로 표시할 수 있습니다.

6. 옵션: 도표나 표에서 모듈을 제거하려면 모듈 옆의 선택란을 선택 해제한 후 **모듈 보기**를 다시 클릭하십시오.
 7. 옵션: 테이블의 데이터를 보려면 카운터 선택 테이블에서 **테이블 보기**를 클릭하십시오. 도표로 다시 토글하려면 **그래프 보기**를 클릭하십시오.
 8. 옵션: 차트의 범례를 보려면 **범례 표시**를 클릭하십시오. 범례를 숨기려면 **범례 숨기기**를 클릭하십시오.
 9. 이벤트의 성능 모니터링이 완료되면 **Tivoli Performance Viewer**를 클릭하고 모니터 중인 서버를 선택한 다음 **모니터링 중지**를 클릭하십시오.
- **성능 통계 로깅**

모니터링이 서버에서 활성화된 동안 현재 사용 가능한 모든 PMI 카운터의 데이터를 로깅하고 결과를 TPV 로그 파일에 기록할 수 있습니다. 최대 20가지 카운터의 여러

조합을 매번 다르게 선택해서 특정 기간에 대한 TPV 로그 파일을 여러 번 볼 수 있습니다. 특정 기간 동안 서버의 여러 다른 성능 측정 간 관계를 유연하게 관찰합니다.

1. 성능 모듈 또는 요약 보고서를 볼 때 로깅 시작을 클릭하십시오.
2. 완료되면 로깅 중지를 클릭하십시오. 기본적으로 로그 파일은 서버가 실행 중인 노드의 `profile_root/logs/tpv` 디렉토리에 저장됩니다. TPV는 공간을 보존하기 위해 로그 파일에 쓰기가 완료되면 로그 파일을 자동으로 압축합니다. 압축된 각 파일에는 단일 로그 파일만 있어야 하며 압축된 파일과 이름이 동일해야 합니다.
3. 관리 콘솔 탐색 트리에서 모니터링 및 조정 → 성능 표시기 → 로그 보기를 클릭해서 로그를 보십시오.

서비스 컴포넌트 이벤트 로그 파일 보기 및 해석

이 주제는 서비스 컴포넌트 모니터링으로 생성된 로그 파일의 정보를 해석하는 방법을 논의합니다. 관리 콘솔의 로그 표시기에서 또는 선택한 별도의 텍스트 파일 편집기에서 로그 파일을 볼 수 있습니다.

서비스 컴포넌트 모니터링으로 로거에 가동된 이벤트는 공통 기본 이벤트 형식으로 인코딩됩니다. 로그 파일에 공개될 때 이벤트는 XML 태그가 붙은 형식의 긴 단일 텍스트 행으로 포함되며 여러 로거 특정 필드도 포함됩니다. 로깅된 이벤트의 공통 기본 이벤트 코딩 해독에 대한 자세한 사항은 이 문서의 이벤트 카탈로그 섹션을 참조하십시오. 이 섹션을 사용하여 로그 파일의 각 항목에 포함된 다른 필드 및 로거를 구성할 때 로그 파일에 대해 선택한 형식이 구조화된 방식을 이해하십시오.

기본 및 고급 형식 필드

로깅 출력을 파일 또는 인메모리 순환 버퍼로 지정할 수 있습니다. 추적 출력이 인메모리 순환 버퍼로 지정된 경우 이 출력을 보려면 파일에 출력을 덤프해야 합니다. 출력은 사용자가 지정한 기본, 고급 또는 로그 분석기 형식의 일반 텍스트로 생성됩니다. 출력의 기본 및 고급 형식은 메시지 로그에 사용할 수 있는 기본 및 고급 형식과 유사합니다. 기본 및 고급 형식은 여러 동일한 필드 및 형식화 기법을 사용합니다. 이 형식으로 사용할 수 있는 필드는 다음과 같습니다.

TimeStamp

시간소인은 형식화되는 프로세스의 로케일을 사용하여 형식화됩니다. 밀리초 단위의 정밀도와 시간대가 있는 완전한 날짜(YMMDD)를 포함합니다.

ThreadId

추적 이벤트를 발행한 스레드의 해시 코드에서 생성된 8자의 16진 값

ThreadName

메시지 또는 추적 이벤트를 발행한 Java 스레드의 이름

ShortName

추적 이벤트를 발행한 로깅 컴포넌트의 축약 이름. 이는 일반적으로 WebSphere Process Server 내부 컴포넌트의 클래스 이름이지만 사용자 응용프로그램의 일부 기타 ID일 수 있습니다.

LongName

추적 이벤트를 발행한 로깅 컴포넌트의 전체 이름. 이는 일반적으로 WebSphere Process Server 내부 컴포넌트의 완전한 클래스 이름이지만 사용자 응용프로그램의 일부 기타 ID일 수 있습니다.

EventType

추적 이벤트의 유형을 나타내는 한 글자 필드. 추적 유형은 소문자입니다. 다음과 같은 값이 가능합니다.

- 1 이벤트 또는 fine 유형의 추적 항목
- 2 finer 유형의 추적 항목
- 3 덤프, 디버그 또는 finest 유형의 추적 항목
- Z 추적 유형이 인식되지 않았음을 나타내는 플레이스홀더

ClassName

메시지나 추적 이벤트를 발행한 클래스

MethodName

메시지나 추적 이벤트를 발행한 메소드

Organization

메시지나 추적 이벤트를 발행한 응용프로그램을 소유하는 조직
 제품 메시지나 추적 이벤트를 발행한 제품
 컴포넌트

메시지나 추적 이벤트를 발행한 제품 내의 컴포넌트

기본 형식

기본 형식으로 표시되는 추적 이벤트는 다음 형식을 사용합니다.

```
<timestamp><threadId><shortName><eventType>
[className] [methodName] <textmessage>
      [parameter 1]
      [parameter 2]
```

고급 형식

고급 형식으로 표시되는 추적 이벤트는 다음 형식을 사용합니다.

```
<timestamp><threadId><eventType><UOW><source=longName>
[className] [methodName]
<Organization><Product><Component>[thread=threadName]
<textMessage>[parameter 1=parameterValue] [parameter 2=parameterValue]
```

로그 분석기 형식

로그 분석기 형식을 지정하면 WebSphere Application Server에 포함된 응용프로그램인 로그 분석기 도구를 사용하여 추적 출력을 열 수 있습니다. 이 도구를 통해 로그 분석기의 병합 기능을 사용할 수 있으므로 두 가지 다른 서버 프로세스의 추적을 상관시키려는 경우에 유용합니다.

제 4 장 이벤트 카탈로그

이벤트 카탈로그에는 각 이벤트가 생성하는 연관된 공통 기본 이벤트 확장 데이터 요소와 각 서비스 컴포넌트 유형에 대해 모니터링할 수 있는 모든 이벤트의 스펙이 포함되어 있습니다.

이 절에 제공된 정보를 참조 자료로 사용하여 개별 이벤트가 구조화되는 방식을 이해하십시오. 이러한 지식은 각 이벤트가 생성하는 상대적으로 용량이 큰 데이터에서 필요한 정보를 빠르게 식별할 수 있도록 각 이벤트에 포함된 정보를 해독하는 데 도움이 됩니다.

이 절에 포함된 정보에서는 다음 항목을 다룹니다.

- 공통 기본 이벤트의 구조 및 표준 요소
- Business Process Choreographer 서비스 컴포넌트의 이벤트 목록
- WebSphere Process Server 특정 서비스 컴포넌트 목록
- 각 이벤트 유형에 고유한 공통 기본 이벤트의 확장

서비스 컴포넌트가 처리할 수 있는 비즈니스 오브젝트가 서비스 컴포넌트 이벤트에서 캡처되는 방식에 대해서도 설명합니다.

제공된 유형의 이벤트는 Common Event Infrastructure 버스에서 CEI 서버 또는 로거로 가동될 때 본래 이벤트 카탈로그 스펙에 따라 작성된 이벤트 요소의 XML 캡슐화인 공통 기본 이벤트의 양식을 사용합니다. 공통 기본 이벤트에는 각 이벤트 유형에 고유한 추가 요소, 이벤트 상관 스피어 ID, 서버 컴포넌트 식별 요소 및 표준 요소 세트가 포함되어 있습니다. 이 모든 요소는 한 가지 경우를 제외하면 서비스 컴포넌트 모니터에서 이벤트가 가동될 때마다 CEI 서버 또는 로거에 전달됩니다. 이벤트가 페이로드 내에 비즈니스 오브젝트 코드를 포함하는 경우에는 이벤트에 포함하려는 비즈니스 오브젝트 데이터의 용량을 지정할 수 있습니다.

공통 기본 이벤트 표준 요소

서비스 컴포넌트 모니터링에서 가동된 모든 이벤트에 포함되는 공통 기본 이벤트의 요소가 여기에 표시됩니다.

속성	설명
version	1.0.1로 설정됩니다.
creationTime	이벤트가 UTC로 작성된 시간
globalInstanceId	공통 기본 이벤트 인스턴스의 ID. 이 ID는 자동으로 생성됩니다.
localInstanceId	이 ID는 자동으로 생성됩니다(공백일 수 있음).

속성	설명
severity	이벤트가 비즈니스 프로세스 또는 휴먼 task에 미치는 영향. 이 속성은 10(정보)으로 설정됩니다. 그렇지 않으면 사용되지 않습니다.
priority	사용되지 않음
reporterComponentId	사용되지 않음
locationType	Hostname으로 설정하십시오.
location	실행 중인 서버의 호스트 이름으로 설정됩니다.
application	사용되지 않음
executionEnvironment	운영 체제를 식별하는 문자열
컴포넌트	프로세스 서버 버전. 비즈니스 프로세스 및 휴먼 task의 경우: WPS# 다음에 SCA 버전, 현재 플랫폼의 ID 및 기본 소프트웨어 스택의 버전 ID를 설정하십시오.
componentType	Apache QName 형식에 기반한 컴포넌트 QName 비즈니스 프로세스: www.ibm.com/namespaces/autonomic/Workflow_Engine 휴먼 task: www.ibm.com/xmlns/prod/websphere/scdl/human-task
subComponent	관찰 가능한 요소 이름 비즈니스 프로세스의 경우 BFM으로 설정되며, 휴먼 task의 경우 HTM으로 설정됩니다.
componentIdType	ProductName으로 설정하십시오.
instanceId	서버 ID. 이 ID의 형식은 cell_name/node_name/server_name입니다. 분리문자는 운영 체제에 따라 다릅니다.
processId	운영 체제의 프로세스 ID
threadId	JVM(Java Virtual Machine)의 스레드 ID.
Situation Type	이벤트의 보고를 유발한 상황의 유형. 특정 컴포넌트의 경우 ReportSituation으로 설정됩니다.
Situation Category	이벤트의 보고를 유발한 상황 유형의 카테고리. 특정 컴포넌트의 경우 STATUS로 설정됩니다.
Situation Reasoning Scope	보고된 상황의 영향 범위. 특정 컴포넌트의 경우 EXTERNAL로 설정됩니다.
ECSCurrentID	현재 이벤트 상관 스피어 ID의 값
ECSParentID	상위 이벤트 상관 스피어 ID의 값
WBISessionID	현재 세션 ID의 값
extensionName	이벤트 이름으로 설정하십시오.

이벤트의 비즈니스 오브젝트

비즈니스 오브젝트 데이터는 버전 6.1에서부터 이벤트 내에서 XML 형식으로 전달됩니다. 공통 기본 이벤트 형식은 비즈니스 오브젝트 페이로드를 XML 요소에 캡슐화하는 `xs:any` 스키마를 포함합니다.

서비스 컴포넌트 이벤트에 캡처할 비즈니스 오브젝트 세부사항의 레벨을 지정합니다. 이 세부사항 레벨은 이벤트에 전달될 비즈니스 오브젝트 코드의 양에만 영향을 미칩니다. 다른 모든 공통 기본 이벤트 요소(표준 및 이벤트 특정 모두)는 이벤트에 공개됩니다. 서비스 컴포넌트 이벤트에 적용 가능한 세부사항 레벨의 이름은 WebSphere Integration Developer를 사용하여 정적 모니터를 작성했는지 또는 관리 콘솔에서 동적 모니터를 작성했는지 여부에 따라 다르며 아래 테이블에 표시됩니다.

관리 콘솔 세부사항 레벨	공통 기본 이벤트/WebSphere Integration Developer 세부사항 레벨	공개된 페이로드 정보
FINE	EMPTY	없음
FINER	DIGEST	페이로드 설명만
FINEST	FULL	모든 페이로드

세부사항 레벨은 이벤트 인스턴스 데이터의 일부인 `PayloadType` 요소로 지정됩니다. 실제 비즈니스 오브젝트 데이터는 모니터가 레코드 FULL/FINEST 세부사항으로 설정된 경우에만 이벤트에 포함됩니다. 비즈니스 오브젝트 데이터 자체는 `xsd:any` 스키마 아래의 공통 기본 이벤트에 포함됩니다. 루트 요소 `wbi:event`가 있는 프로세스 서버 비즈니스 오브젝트 페이로드를 확인할 수 있습니다. 이벤트 출력을 로거에 공개 중이면 로그 파일을 볼 때 출력이 표시됩니다. 이벤트가 CEI 서버에 공개된 경우에는 공통 기본 이벤트 브라우저를 사용하여 이벤트를 볼 수 있습니다. 그런 다음, `wbi:event` 링크를 클릭해서 비즈니스 오브젝트 데이터를 볼 수 있습니다.

Business Process Choreographer 이벤트

WebSphere Process Server는 비즈니스 프로세스 및 휴먼 태스크에 대한 Business Process Choreographer 서비스 컴포넌트를 통합합니다. 이 컴포넌트에서 모니터링할 수 있는 이벤트 지점을 이 절에서 설명합니다.

WebSphere Process Server 이벤트

WebSphere Process Server에는 자체 서비스 컴포넌트가 있으며 이러한 컴포넌트 각각에는 모니터링할 수 있는 자체 이벤트 지점 세트가 있습니다.

서비스 컴포넌트는 하나 이상의 요소를 포함하며 이들 요소는 각 서비스 컴포넌트에서 처리되는 다른 단계 세트입니다. 각 요소에는 서비스 컴포넌트 요소를 처리할 때 도달

하는 핵심 지점인 자체 이벤트 네이처 세트가 있습니다. 모든 서비스 컴포넌트, 요소 및 연관된 이벤트 네이처와 각 이벤트에 고유한 확장 데이터 요소가 표시됩니다.

자원 어댑터 이벤트

자원 어댑터 컴포넌트에 사용 가능한 이벤트 유형이 표시됩니다.

각 이벤트에 고유한 확장 데이터 요소, 이름 및 연관된 이벤트 네이처와 함께 모니터링할 수 있는 자원 어댑터 컴포넌트의 요소(기본 이름은 `eis:WBI.JCAAdapter`)가 여기에 표시됩니다.

이벤트 이름	이벤트 네이처	이벤트 콘텐츠	유형
InboundEventRetrieval 요소			
<code>eis:WBI.JCAAdapter.InboundEventRetrieval.ENTRY</code>	ENTRY	pollQuantity	정수
		status	정수
		eventTypeFilters	문자열
<code>eis:WBI.JCAAdapter.InboundEventRetrieval.EXIT</code>	EXIT	해당 없음	
<code>eis:WBI.JCAAdapter.InboundEventRetrieval.FAILURE</code>	FAILURE	FailureReason	예외
InboundEventDelivery 요소			
<code>eis:WBI.JCAAdapter.InboundEventDelivery.ENTRY</code>	ENTRY	해당 없음	
<code>eis:WBI.JCAAdapter.InboundEventDelivery.EXIT</code>	EXIT	해당 없음	
<code>eis:WBI.JCAAdapter.InboundEventDelivery.FAILURE</code>	FAILURE	FailureReason	예외
Outbound 요소			
<code>eis:WBI.JCAAdapter.Outbound.ENTRY</code>	ENTRY	해당 없음	
<code>eis:WBI.JCAAdapter.Outbound.EXIT</code>	EXIT	해당 없음	
<code>eis:WBI.JCAAdapter.Outbound.FAILURE</code>	FAILURE	FailureReason	예외
InboundCallbackAsyncDeliverEvent 요소			
<code>eis:WBI.JCAAdapter.InboundCallbackAsyncDeliverEvent.ENTRY</code>	ENTRY	해당 없음	
<code>eis:WBI.JCAAdapter.InboundCallbackAsyncDeliverEvent.EXIT</code>	EXIT	해당 없음	
<code>eis:WBI.JCAAdapter.InboundCallbackAsyncDeliverEvent.FAILURE</code>	FAILURE	FailureReason	예외
InboundCallbackSyncDeliverEvent 요소			
<code>eis:WBI.JCAAdapter.InboundCallbackSyncDeliverEvent.ENTRY</code>	ENTRY	해당 없음	
<code>eis:WBI.JCAAdapter.InboundCallbackSyncDeliverEvent.EXIT</code>	EXIT	해당 없음	

이벤트 이름	이벤트 네이처	이벤트 콘텐츠	유형
eis:WBI.JCAAdapter. InboundCallbackSyncDeliverEvent. FAILURE	FAILURE	FailureReason	예외
Polling 요소			
eis:WBI.JCAAdapter. Polling.STARTED	STARTED	PollFrequency	정수
		PollQuantity	정수
eis:WBI.JCAAdapter. Polling.STOPPED	STOPPED	해당 없음	
Delivery 요소			
eis:WBI.JCAAdapter. Delivery.EXIT	EXIT	해당 없음	
eis:WBI.JCAAdapter. Delivery.FAILURE	FAILURE	EventID	문자열
		FailureReason	예외
Retrieval 요소			
eis:WBI.JCAAdapter. Retrieval.FAILURE	FAILURE	EventID	문자열
		FailureReason	예외
Endpoint 요소			
eis:WBI.JCAAdapter. Endpoint.FAILURE	FAILURE	FailureReason	예외
Recovery 요소			
eis:WBI.JCAAdapter. Recovery.EXIT	EXIT	해당 없음	
eis:WBI.JCAAdapter. Recovery.FAILURE	FAILURE	FailureReason	예외
EventFailure 요소			
eis:WBI.JCAAdapter. EventFailure.FAILURE	FAILURE	FailureReason	예외
Connection 요소			
eis:WBI.JCAAdapter. Connection.FAILURE	FAILURE	FailureReason	예외

비즈니스 규칙 이벤트

비즈니스 규칙 컴포넌트에 사용 가능한 이벤트 유형이 표시됩니다.

비즈니스 규칙 컴포넌트(기본 이름 br:WBI.BR)는 모니터링할 수 있는 단일 요소를 포함합니다. 각 이벤트에 고유한 확장 데이터 요소, 이벤트 이름 및 연관된 이벤트 네이처와 함께 이 요소에 대한 모든 이벤트 유형이 여기에 표시됩니다.

이벤트 이름	이벤트 네이처	이벤트 콘텐츠	유형
br:WBI.BR.ENTRY	ENTRY	operationName	문자열
br:WBI.BR.EXIT	EXIT	operationName	문자열
br:WBI.BR.FAILURE	FAILURE	ErrorReport	예외
		operationName	문자열
WBI.BR. br:SelectionKeyExtracted	SelectionKeyExtracted	operationName	문자열

이벤트 이름	이벤트 네이처	이벤트 콘텐츠	유형
br:WBI.BR.TargetFound	TargetFound	operationName	문자열
		target	문자열

비즈니스 상태 머신 이벤트

비즈니스 상태 머신 컴포넌트에 사용 가능한 이벤트 유형이 표시됩니다.

다음은 모니터링할 수 있는 비즈니스 상태 머신 컴포넌트(기본 이름 bsm:WBI.BSM)의 요소 목록입니다. 각각의 이벤트에 고유한 연관된 이벤트 네이처, 이벤트 이름 및 모든 확장된 데이터 요소가 같이 표시되어 있습니다.

이벤트 이름	이벤트 네이처	이벤트 콘텐츠	유형
StateMachineDefinition 요소			
bsm:WBI.BSM.StateMachineDefinition.ALLOCATED	ALLOCATED	instanceID	문자열
bsm:WBI.BSM.StateMachineDefinition.RELEASED	RELEASED	instanceID	문자열
Transition 요소			
bsm:WBI.BSM.Transition.ENTRY	ENTRY	instanceID	문자열
		name	문자열
bsm:WBI.BSM.Transition.EXIT	EXIT	instanceID	문자열
		name	문자열
bsm:WBI.BSM.Transition.FAILURE	FAILURE	ErrorReport	예외
		instanceID	문자열
		name	문자열
State 요소			
bsm:WBI.BSM.State.ENTRY	ENTRY	instanceID	문자열
		name	문자열
bsm:WBI.BSM.State.EXIT	EXIT	instanceID	문자열
		name	문자열
bsm:WBI.BSM.State.FAILURE	FAILURE	ErrorReport	예외
		instanceID	문자열
		name	문자열
Guard 요소			
bsm:WBI.BSM.Guard.ENTRY	ENTRY	instanceID	문자열
		name	문자열
bsm:WBI.BSM.Guard.EXIT	EXIT	instanceID	문자열
		name	문자열
		result	부울
bsm:WBI.BSM.Guard.FAILURE	FAILURE	ErrorReport	예외
		instanceID	문자열
		name	문자열

이벤트 이름	이벤트 네이처	이벤트 콘텐츠	유형
Action 요소			
bsm:WBI.BSM.Action.ENTRY	ENTRY	instanceID	문자열
		name	문자열
bsm:WBI.BSM.Action.EXIT	EXIT	instanceID	문자열
		name	문자열
bsm:WBI.BSM.Action.FAILURE	FAILURE	ErrorReport	예외
		instanceID	문자열
		name	문자열
EntryAction 요소			
bsm:WBI.BSM.EntryAction. ENTRY	ENTRY	instanceID	문자열
		name	문자열
bsm:WBI.BSM.EntryAction. EXIT	EXIT	instanceID	문자열
		name	문자열
bsm:WBI.BSM.EntryAction. FAILURE	FAILURE	ErrorReport	예외
		instanceID	문자열
		name	문자열
ExitAction 요소			
bsm:WBI.BSM.ExitAction.ENTRY	ENTRY	instanceID	문자열
		name	문자열
bsm:WBI.BSM.ExitAction.EXIT	EXIT	instanceID	문자열
		name	문자열
bsm:WBI.BSM.ExitAction. FAILURE	FAILURE	ErrorReport	예외
		instanceID	문자열
		name	문자열
Timer 요소			
bsm:WBI.BSM.Timer.START	START	instanceID	문자열
		name	문자열
		duration	문자열
bsm:WBI.BSM.Timer.STOPPED	STOPPED	instanceID	문자열
		name	문자열
		duration	문자열

맵 이벤트

맵 컴포넌트에 사용 가능한 이벤트 유형이 표시됩니다.

각 이벤트에 고유한 모든 확장 데이터 요소, 이벤트 이름 및 이벤트 네이처와 함께 모니터링할 수 있는 맵 컴포넌트의 요소(기본 이름은 map:WBI.MAP)가 여기에 표시됩니다.

표 21. 기본 요소

이벤트 이름	이벤트 네이처	이벤트 콘텐츠	유형
map:WBI.MAP.ENTRY	ENTRY	해당 없음	해당 없음

표 21. 기본 요소 (계속)

이벤트 이름	이벤트 네이처	이벤트 콘텐츠	유형
map:WBI.MAP.EXIT	EXIT	해당 없음	해당 없음
map:WBI.MAP.FAILURE	FAILURE	FailureReason	예외
Transformation 요소			
map:WBI.MAP.Transformation. ENTRY	ENTRY	해당 없음	해당 없음
map:WBI.MAP.Transformation. EXIT	EXIT	해당 없음	해당 없음
map:WBI.MAP.Transformation. FAILURE	FAILURE	FailureReason	예외

중개 이벤트

중개 컴포넌트에 사용 가능한 이벤트 유형이 표시됩니다.

각 이벤트에 고유한 모든 확장 데이터 요소, 이름 및 연관된 이벤트 네이처와 함께 모니터링할 수 있는 중개 컴포넌트의 요소(기본 이름은 ifm:WBI.MEDIATION)가 여기에 표시됩니다.

이벤트 이름	이벤트 네이처	이벤트 콘텐츠	유형
OperationBinding 요소			
ifm:WBI.MEDIATION. OperationBinding.ENTRY	ENTRY	InteractionType	문자열
		TicketID	문자열
		Source	문자열
		Target	문자열
ifm:WBI.MEDIATION. OperationBinding.EXIT	EXIT	InteractionType	문자열
		TicketID	문자열
		Source	문자열
		Target	문자열
ifm:WBI.MEDIATION. OperationBinding.FAILURE	FAILURE	InteractionType	문자열
		TicketID	문자열
		Source	문자열
		Target	문자열
		ErrorReport	예외
ParameterMediation 요소			
ifm:WBI.MEDIATION. ParameterMediation. ENTRY	ENTRY	유형	문자열
		TransformName	문자열
WBI.MEDIATION. ParameterMediation. EXIT	EXIT	유형	문자열
		TransformName	문자열
ifm:WBI.MEDIATION. ParameterMediation. FAILURE	FAILURE	유형	문자열
		TransformName	문자열
		ErrorReport	예외

복구 이벤트

복구 컴포넌트에 사용 가능한 이벤트 유형이 표시됩니다.

복구 컴포넌트(기본 이름 `recovery:WBI.Recovery`)는 모니터링할 수 있는 단일 요소를 포함합니다. 각 이벤트에 고유한 확장 데이터 요소, 이벤트 이름 및 연관된 이벤트 네이처와 함께 이 요소에 대한 모든 이벤트 유형이 여기에 표시됩니다.

이벤트 이름	이벤트 네이처	이벤트 콘텐츠	유형
recovery:WBI.Recovery.FAILURE	FAILURE	MsgId	문자열
		DestModuleName	문자열
		DestComponentName	문자열
		DestMethodName	문자열
		SourceModuleName	문자열
		SourceComponentName	문자열
		ResubmitDestination	문자열
		ExceptionDetails	문자열
		SessionId	문자열
		FailureTime	dateTime
		ExpirationTime	dateTime
		Status	정수
		MessageBody	byteArray
Deliverable	부울		
recovery:WBI.Recovery.DEADLOOP	DEADLOOP	DeadloopMsgId	문자열
		SIBusName	문자열
		QueueName	문자열
		Reason	문자열
recovery:WBI.Recovery.RESUBMIT	RESUBMIT	MsgId	문자열
		OriginalMesId	문자열
		ResubmitCount	정수
		Description	문자열
recovery:WBI.Recovery.DELETE	DELETE	MsgId	문자열
		deleteTime	dateTime
		Description	문자열

Service Component Architecture 이벤트

Service Component Architecture에 사용 가능한 이벤트 유형이 표시됩니다.

SCA(Service Component Architecture)는 기본 이름이 `sca:WBI.SCA.MethodInvocation`인 단일 요소를 포함합니다. 각 이벤트에 고유한 모든 확장 데이터 요소와 함께 이 요소의 모든 이벤트 및 연관된 네이처가 여기에 표시됩니다.

주: 이 이벤트를 SCA 특정 ARM(Application Response Measurement) 성능 통계와 혼동하지 마십시오.

이벤트 이름	이벤트 네이처	이벤트 콘텐츠	유형
WBI.SCA. MethodInvocation. ENTRY	ENTRY	SOURCE COMPONENT	문자열
		SOURCE INTERFACE	문자열
		SOURCE METHOD	문자열
		SOURCE MODULE	문자열
		SOURCE REFERENCE	문자열
		TARGET COMPONENT	문자열
		TARGET INTERFACE	문자열
		TARGET METHOD	문자열
		TARGET MODULE	문자열
WBI.SCA. MethodInvocation. EXIT	EXIT	SOURCE COMPONENT	문자열
		SOURCE INTERFACE	문자열
		SOURCE METHOD	문자열
		SOURCE MODULE	문자열
		SOURCE REFERENCE	문자열
		TARGET COMPONENT	문자열
		TARGET INTERFACE	문자열
		TARGET METHOD	문자열
		TARGET MODULE	문자열
WBI.SCA. MethodInvocation. FAILURE	FAILURE	SOURCE COMPONENT	문자열
		SOURCE INTERFACE	문자열
		SOURCE METHOD	문자열
		SOURCE MODULE	문자열
		SOURCE REFERENCE	문자열
		TARGET COMPONENT	문자열
		TARGET INTERFACE	문자열
		TARGET METHOD	문자열
		TARGET MODULE	문자열
		예외	문자열

선택기 이벤트

선택기 컴포넌트에 사용 가능한 이벤트 유형이 표시됩니다.

선택기 컴포넌트는 모니터링할 수 있는 단일 요소를 포함합니다. 각 이벤트에 고유한 확장 데이터 요소, 이벤트 이름 및 연관된 이벤트 네이처와 함께 이 요소에 대한 모든 이벤트 유형이 여기에 표시됩니다. 모든 선택기 이벤트에는 기본 이름, sel:WBI.SEL이 있습니다.

이벤트 이름	이벤트 네이치	이벤트 콘텐츠	유형
sel:WBI.SEL.ENTRY	ENTRY	operationName	문자열
sel:WBI.SEL.EXIT	EXIT	operationName	문자열
sel:WBI.SEL.FAILURE	FAILURE	ErrorReport	예외
		operationName	문자열
sel:WBI.SEL.SelectionKeyExtracted	SelectionKeyExtracted	operationName	문자열
sel:WBI.SEL.TargetFound	TargetFound	operationName	문자열
		target	문자열

