

CA Application Performance Management

for SOA 구현 안내서

릴리스 9.5



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

CA의 사전 서면 동의 없이 본건 문서의 전체 혹은 일부를 복사, 전송, 재생, 공개, 수정 또는 복제할 수 없습니다. 이 문서는 CA의 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 CA에 반환되거나 폐기되었음을 입증할 책임이 있습니다.

CA는 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. CA는 본 시스템의 사용으로 인해 발생되는 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 3자에게 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 CA에서 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

제 1 장: CA Application Performance Management for SOA 정보 17

서비스 지향 아키텍처란?	17
SOA 인프라의 일반적인 구성 요소	19
CA Introscope®를 사용하여 SOA 성능을 모니터링하는 방법	20
웹 서비스 클라이언트 및 서버 모니터링	21
사전 대응적 관리를 위한 SOA 특정 대시보드 사용	22
종속성 보기 및 심사와 진단을 위한 메트릭 사용	22
SOA 구성 요소가 관련된 트랜잭션 추적	23
여러 플랫폼 및 전송 모니터링	23
SOA 모니터링 아키텍처 이해	24
추가 SOA 플랫폼 지원	26

제 2 장: CA APM for SOA 설치 및 구성 29

설치 및 구성	29
CA Introscope® 구성 요소 및 버전	30
에이전트의 기본 시스템 요구 사항	30
에이전트에 대한 SOAP 및 응용 프로그램 서버 요구 사항	31
디렉터리 및 파일 명명 규칙 이해	32
이전 버전에서 업그레이드하기 전에 백업	33
CA APM for SOA 를 에이전트에 추가하는 방법	33
Standalone Agent 설치 관리자에서 CA APM for SOA 선택	34
자동 모드를 사용하여 에이전트에 대해 CA APM for SOA 를 사용하도록 설정	35
CA APM for SOA 를 수동으로 에이전트에 추가	36
설치 후 에이전트 속성 구성	38
SOA 플랫폼 확장을 사용하도록 설정하는 방법	38
Enterprise Manager 에서 확장을 사용하도록 설정	41
CA APM for SOA 배포 확인	44
CA APM for SOA 제거	45

제 3 장: SOA(Service-Oriented Architecture) 모니터링 47

CA Introscope®에서 SOA 특정 정보 보기	47
클라이언트와 서버 웹 서비스 및 작업 정보	48
서비스 네임스페이스 및 작업 이름 정보	49

식별되지 않은 서비스 및 작업 정보	49
가상 에이전트 정보	50
SOA 성능 대시보드 사용	51
SOA 성능 개요 대시보드	55
SOA 성능 클라이언트 건전성 대시보드	55
SOA 성능 서버 건전성 대시보드	56
SOA 성능 - 가장 중요한 작업 대시보드	56
SOA 성능 - 가장 종속된 작업 대시보드	57
SOA 성능 - 가장 사용량이 많은 작업 대시보드	57
Investigator 를 사용하여 SOA 성능 메트릭 보기	58
사용할 수 있는 메트릭	58
"개요" 탭에서 요약된 메트릭 보기	59
"종속성" 탭에서 종속성 메트릭 보기	61
"편차" 탭에서 편차 메트릭 보기	66
"가장 중요한" 탭에서 중요한 작업 메트릭 보기	67
"가장 종속된" 탭에서 종속된 작업 보기	67
"오류" 탭에서 SOAP 결합 및 오류 메트릭 보기	68
Investigator 에서 경계 blame 보기	69
기본 SOA 특정 메트릭 그룹화 보기	69
기본 CA APM for SOA 경고 보기	70

제 4 장: SOA 종속성 맵 사용 73

SOA 종속성 맵의 사용 이해	73
SOA 환경의 문제점 이해	74
SOA 종속성 맵이 제공하는 정보 이해	74
종속성 및 SOA 용어 이해	75
SOA 종속성 맵을 통해 수행할 수 있는 작업 이해	76
SOA 종속성 맵이 데이터를 가져오는 방법	76
SOA 종속성 맵의 지속 데이터 정보	77
새 서비스와 작업으로 종속성 맵 업데이트	79
사용되지 않는 맵 노드의 만료 처리 및 제거	80
종속성 맵에 불완전한 데이터가 표시되는 경우에 수행할 작업	80
논리적 동등성 규칙 정보	82
SOA 종속성 맵의 컨텍스트 이해	82
콘텐츠 유형이 표시되는 내용에 미치는 영향 이해	82
Investigator 노드가 표시되는 내용에 미치는 영향 이해	83
SOA 종속성 맵 표시	84
Investigator 트리 노드와 맵 노드 정보	85
독립 실행형 맵 노드 및 종속 맵 노드 정보	85

작업 맵 노드 정보	86
물리적 또는 논리적 뷔 선택	86
콘텐츠 유형 선택	87
"종속성 맵" 노드의 "Primary Metric"(기본 메트릭) 설정	88
종속성 맵 노드의 도구 설명 메트릭 선택	89
표시되는 종속성 수준 변경	90
맵 노드의 추가적인 종속성 확인	91
맵 노드의 종속성 숨기기	91
맵에 있는 모든 항목에 대한 추가적인 종속성 확인	92
맵의 모든 항목에 대해 종속성 숨기기	92
SOA 종속성 맵 내에서 탐색	93
SOA 종속성 맵의 이동, 확대/축소 및 맞춤	93
맵 노드에서 관련 Investigator 트리 노드로 이동	94
서비스의 모든 작업 표시	95
에이전트의 모든 서비스 표시	95
SOA 종속성 맵 이미지 저장	96
SOA 종속성 맵 공유	96
SOA 종속성 맵을 스냅숏으로 저장	96
맵 노드에서 트랜잭션 추적 시작	98
클러스터에 대한 SOA 종속성 맵	99

제 5 장: SOA 환경에서 트랜잭션 추적 사용 101

크로스 프로세스 트랜잭션 추적 정보	101
트랜잭션의 세그먼트가 연결되는 방식 이해	102
크로스 프로세스 트랜잭션 추적의 컨텍스트 이해	103
트랜잭션 추적을 사용하여 문제 해결	104
트랜잭션 추적 시작 및 보기	105
요약 뷔 사용	106
추적 뷔 사용	107
트리 뷔 사용	107
시퀀스 뷔 사용	108
트랜잭션 추적에 대한 필터 설정	112
필터 추가 및 저장	114
필터 제거	114
복합 필터 사용	115
이벤트 데이터베이스에서 SOA 서비스 쿼리	115
오류 이벤트 정보 보기	116
상호 관련된 이벤트 정보 보기	116

제 6 장: SOA 특정 속성 구성 119

웹 서비스의 표시 이름 구성	119
상호 관련된 추적 구성	121
클라이언트와 서버의 속성 구성	123
상호 관련된 추적 사용 안 함	123
SOAP 핸들러의 삽입 지점 구성.....	124
SOAP 핸들러 삽입을 위한 클라이언트 및 서버 속성	124
SOAP 핸들러의 기본 순서 변경	125
SOA 종속성 맵의 제한 구성.....	125

제 7 장: Oracle Service Bus 모니터링 127

OSB(Oracle Service Bus) 정보	127
Oracle Service Bus 모니터링을 사용하도록 설정하는 방법	131
Oracle Service Bus 모니터링을 사용하도록 에이전트 설정	131
Oracle Service Bus Enterprise Manager 확장을 사용하도록 설정	134
대시보드를 사용하여 Oracle Service Bus 모니터링.....	136
Oracle Service Bus에 대한 메트릭 이해 및 보기	139
비즈니스 서비스에 대한 메트릭	141
파이프라인에 대한 메트릭	141
프록시 서비스에 대한 메트릭	142
전송에 대한 메트릭	143
UDDI에 대한 메트릭	144
XQuery에 대한 메트릭	144
기본 Oracle Service Bus 메트릭 그룹화 보기	145
기본 Oracle Service Bus 경고 보기	146
Oracle Service Bus 종속성 보기	147
Oracle Service Bus에 대한 트랜잭션 추적	148
크로스 프로세스 트랜잭션 추적의 가치 이해.....	148
샘플 트랜잭션 추적 시작 및 보기	148

제 8 장: TIBCO BusinessWorks 모니터링 151

TIBCO BusinessWorks 정보	151
TIBCO BusinessWorks 모니터링을 사용하도록 설정하는 방법.....	153
TIBCO BusinessWorks 모니터링을 위하여 에이전트를 사용하도록 설정.....	154
에이전트를 사용하도록 TIBCO BusinessWorks를 구성합니다.....	158
자동 에이전트 이름 지정 구성	159
Enterprise Manager 확장을 사용하도록 설정	161
대시보드를 사용하여 TIBCO BusinessWorks 모니터링	161

TIBCO BusinessWorks에 대한 메트릭 이해 및 보기	164
동작에 대한 메트릭	166
Group Actions(그룹 작업)에 대한 메트릭	166
Hawk에 대한 메트릭	167
작업 및 작업 풀에 대한 메트릭	174
프로세스에 대한 메트릭	177
전송에 대한 메트릭	180
WebServices에 대한 메트릭	183
기본 TIBCO BusinessWorks 메트릭 그룹화 보기	185
기본 TIBCO BusinessWorks 경고 보기	185
TIBCO BusinessWorks 종속성 보기	186
TIBCO BusinessWorks에 대한 트랜잭션 추적	189
Tibco에서 트랜잭션 추적 데이터를 사용하는 방법	190
크로스 프로세스 트랜잭션 추적의 가치 이해	190
샘플 트랜잭션 추적 시작 및 보기	191
BusinessWorks의 프런트엔드 및 백엔드 정보	192
생성된 프로세스에 대한 제한 사항 이해	193
한 프런트엔드에 여러 백엔드를 연결하지 않도록 설정	194
사용자 지정 TIBCO BusinessWorks 백엔드 추가	194
메트릭 만료 처리 및 제거 사용자 지정	195
TIBCO BusinessWorks를 위한 상호 관련된 추적 구성	196

제 9 장: TIBCO Enterprise Message Service 모니터링 197

TIBCO Enterprise Message Service 정보	198
SOA Extension for TIBCO EMS 설치 방법	198
Standalone Agent 설치 관리자 실행	199
자동 설치를 위해 응답 파일 사용	200
설치 아카이브 수동 추출	201
TIBCO EMS 서버 모니터링 준비	201
TIBCO EMSMonitor 에이전트 구성	203
기본 연결 속성 구성	204
서버의 폴링 간격 구성	205
선택적 모니터링을 위한 필터 구성	206
큐 및 항목에 대한 필터 구성	206
포함할 고급 구성 요소 정의	207
고급 구성 요소에 대한 정규식 필터 정의	208
EMS 이름의 특수 문자 바꾸기	209
모니터링 수준 정의	209
기본 모니터링 수준 정의 사용	210

기본 모니터링 수준 정의 설정	212
암호화된 암호 만들기	214
SSL 을 사용하여 TIBCO EMS 서버 인스턴스에 연결	215
EMSMonitor 에이전트 시작.....	217
시작 스크립트를 사용하여 EMSMonitor 시작.....	217
EMSMonitor 에이전트를 Windows 서비스로 실행.....	218
Enterprise Manager 확장을 사용하도록 설정	219
대시보드를 사용하여 TIBCO EMS 모니터링	220
TIBCO EMS 메트릭 이해 및 보기	222
마지막 검사에 대한 메트릭	227
큐에 대한 메트릭	228
서버에 대한 메트릭	233
항목에 대한 메트릭	244
라우트에 대한 메트릭	249
채널에 대한 메트릭	251
브리지에 대한 메트릭	253
기본 TIBCO EMS 메트릭 그룹화 보기	253
기본 TIBCO EMS 경고 보기	254
에이전트 구성 속성 요약	255

제 10 장: **webMethods Broker** 모니터링 263

webMethods Broker 정보.....	263
SOA Extension for webMethods Broker 설치 방법	265
사전 요구 사항 확인	265
Standalone Agent 설치 관리자 실행	266
webMethods Broker 를 모니터링을 위해 준비	268
webMethods Broker 에 대한 에이전트 구성	268
Enterprise Manager Extension for webMethods Broker 를 사용하도록 설정	275
대시보드를 사용하여 webMethods Broker 모니터링	275
브로커에 대한 메트릭 이해 및 보기.....	278
브로커에 대한 메트릭	281
클라이언트 그룹에 대한 메트릭	282
클라이언트에 대한 메트릭	282
문서 유형에 대한 메트릭	284
다시 시도 큐에 대한 메트릭	285
영역 통계에 대한 메트릭	286
추적 큐에 대한 메트릭	287
사용률에 대한 메트릭	287
기본 브로커 메트릭 그룹화 보기	288

기본 브로커 경고 보기	289
--------------------	-----

제 11 장: **webMethods Integration Server** 모니터링 291

webMethods Integration Server 정보	291
webMethods Integration Server 모니터링을 사용하도록 설정하는 방법	295
webMethods Integration Server 모니터링을 사용하도록 수동으로 에이전트 설정	295
webMethods Integration Server에 대한 지시문 파일 정보	297
Enterprise Manager 확장을 사용하도록 설정	298
대시보드를 사용하여 webMethods Broker 모니터링	299
모니터링 및 표시되는 서비스 필터링	304
기본 구성 파일 정보	305
정규식을 사용하여 서비스 포함 및 제외	305
다른 구성 파일 위치 선택	306
webMethods에 대한 메트릭 보기 및 탐색	306
어댑터에 대한 메트릭	307
권한 부여에 대한 메트릭	310
비즈니스 프로세스에 대한 메트릭	311
흐름 서비스에 대한 메트릭	314
Java 서비스에 대한 메트릭	315
JDBC 풀에 대한 메트릭	316
스레드 풀에 대한 메트릭	316
트레이딩 네트워크에 대한 메트릭	317
트리거에 대한 메트릭	320
WebServices에 대한 메트릭	321
XSLT 서비스에 대한 메트릭	322
기본 webMethods 메트릭 그룹화 보기	323
기본 webMethods 경고 보기	324
webMethods 종속성 보기	324
webMethods에 대한 트랜잭션 추적	325
크로스 프로세스 트랜잭션 추적 구성 정보	326
크로스 프로세스 트랜잭션 추적의 가치 이해	326
샘플 트랜잭션 추적 시작 및 보기	327
webMethods 프로세스에 대한 시퀀스 뷰 사용	327

제 12 장: **WebSphere Process Server** 및 **WESB** 모니터링 329

WebSphere Process Server 및 WESB 정보	329
WebSphere Process Server 구성 요소 모니터링	330
독립 실행형 WebSphere Enterprise Service Bus 모니터링	333

WebSphere Process Server 또는 WESB 모니터링을 사용하도록 설정하는 방법	334
WPS 또는 WESB 모니터링을 위하여 에이전트를 사용하도록 설정	335
Enterprise Manager Extension for WPS 또는 WESB 를 사용하도록 설정	339
WPS 또는 WESB 모니터링에 대시보드 사용	342
WebSphere Process Server 대시보드 정보	343
WESB 대시보드 정보	344
WebSphere Process Server 또는 WESB 대시보드 보기	346
WPS/WESB 에 대한 메트릭 보기 및 탐색	347
비즈니스 개체 맵에 대한 메트릭	348
비즈니스 프로세스에 대한 메트릭	348
비즈니스 규칙에 대한 메트릭	348
비즈니스 상태 컴퓨터에 대한 메트릭	349
데이터 바인딩에 대한 메트릭	349
수동 작업에 대한 메트릭	349
인터페이스 맵에 대한 메트릭	350
J2CA 어댑터에 대한 메트릭	350
Java 구성 요소에 대한 메트릭	350
조정 흐름 및 조정 기본에 대한 메트릭	351
관계에 대한 메트릭	354
선택기에 대한 메트릭	355
SIB(Service Integration Bus) 통신에 대한 메트릭	355
WebSphere Process Server 결합에 대한 메트릭	355
기본 WPS 및 WESB 메트릭 그룹화 보기	355
기본 WPS 및 WESB 경고 보기	356
WPS 또는 WESB 종속성 보기	357
WPS 또는 WESB 에 대한 트랜잭션 추적	358
크로스 프로세스 트랜잭션 추적의 가치 이해	359
샘플 트랜잭션 추적 시작 및 보기	359
WebSphere 프로세스에 대한 시퀀스 뷰 사용	360
비즈니스 프로세스/비즈니스 상태 컴퓨터 동작이 포함되도록 추적 구성	360

부록 A: SOA 특정 에이전트 구성 속성 363

에이전트 속성의 위치 이해	363
에이전트 속성 구성	363
SOA 특정 에이전트 속성 정보	364
agent.httpheaderinsertion.enabled	366
agent.httpheaderread.enabled	367
agent.soapheaderinsertion.enabled	368
agent.soapheaderread.enabled	369

agent.soa.metricNameFormatting.....	369
agent.transactiontrace.boundaryTracing.cacheFlushFrequency.....	371
agent.transactiontrace.boundaryTracing.enable.....	372
soa.client.prependhandler	373
soa.server.appendhandler	374

부록 B: SOA 특정 Enterprise Manager 구성 속성 375

Enterprise Manager 속성 구성	375
SOA 특정 Enterprise Manager 속성 정보	377
soa.dependencymap.aging.refresh.interval	380
soa.dependencymap.aging.expire.interval	381
soa.dependencymap.heuristics.clientside.enable	382
soa.dependencymap.heuristics.namematch.enable	383
soa.dependencymap.heuristics.serverside.enable	384
soa.dependencymap.log.suppress	385
soa.dependencymap.max.edge.ratio	386
soa.dependencymap.max.vertices	387
soa.deviation.enable	388
soa.deviation.art.enable	389
soa.deviation.dependencymetric.enable	389
soa.deviation.count.per.metric	390
soa.deviation.dependency.refreshrate	391
soa.deviation.errors.enable	392
soa.deviation.max.metric.count	393
soa.deviation.mean.days	393
soa.deviation.metric.expressionlist	394
soa.deviation.metric.calledbackends	395
soa.deviation.usage.enable	396
soa.dashboard.typeviewer.average.enable	397
soa.dashboard.typeviewer.errors.enable	398
soa.dashboard.typeviewer.response.enable	399
soa.dashboard.typeviewer.mostcritical.enable	400
soa.dashboard.typeviewer.mostcritical.count	401
soa.dashboard.typeviewer.mostdependent.enable	402
soa.dashboard.typeviewer.mostdependent.count	403

부록 C: SOA 특정 Workstation 구성 속성 405

Workstation 속성 구성	406
SOA 특정 Workstation 속성 정보	407
soa.dependencymap.ui.view.nodecount	408

com.wily.introscope.soa.dependencymap.ui.view.edgecount.....	409
soa.dashboard.typeviewer.average.enable	411
soa.dashboard.typeviewer.errors.enable	412
soa.dashboard.typeviewer.response.enable	413
soa.dashboard.typeviewer.mostcritical.enable	414
soa.dashboard.typeviewer.mostcritical.count.....	415
soa.dashboard.typeviewer.mostdependent.enable	416
soa.dashboard.typeviewer.mostdependent.count.....	417
workstation.soa.dependencymap.fetchmetrics	418
workstation.traceview.crossprocess.duration.full.....	419
workstation.traceview.crossprocess.duration.net.....	420

부록 D: SOA 특정 WebView 구성 속성 421

WebView 속성 구성	421
SOA 특정 WebView 속성 정보	422
soa.dependencymap.ui.view.nodecount.....	423
com.wily.introscope.soa.dependencymap.ui.view.edgecount.....	424

제 1 장: CA Application Performance Management for SOA 정보

SOA(서비스 지향 아키텍처)는 조직에서 비즈니스 목표 달성을 위해 느슨하게 결합된 서비스를 공유하고 다시 사용할 수 있게 해주는 응용 프로그램 플랫폼입니다.

서비스 지향 아키텍처를 사용하여 응용 프로그램을 배포하면 몇 가지 이점이 있지만 SOA 환경을 모니터링하는 데 따르는 어려움도 있습니다. 이 단원에서는 CA Application Performance Management for SOA(CA APM for SOA)를 사용하여 이러한 어려움을 해결하는 주요 방법을 설명합니다.

이 섹션은 다음 항목을 포함하고 있습니다.

[서비스 지향 아키텍처란?](#) (페이지 17)

[SOA 인프라의 일반적인 구성 요소](#) (페이지 19)

[CA Introscope®를 사용하여 SOA 성능을 모니터링하는 방법](#) (페이지 20)

[SOA 모니터링 아키텍처 이해](#) (페이지 24)

[추가 SOA 플랫폼 지원](#) (페이지 26)

서비스 지향 아키텍처란?

SOA(서비스 지향 아키텍처)는 느슨하게 결합된 서비스를 통해 비즈니스 목표를 달성할 수 있도록 표준화된 통신 프로토콜을 사용하는 응용 프로그램 플랫폼입니다.

SOA를 사용하면 비즈니스의 민첩성과 유연성을 높이고, 고객 서비스와 효율성을 개선하며, 개발 비용을 절감할 수 있다는 장점이 있습니다. 그러나 복잡한 SOA 환경의 모니터링 및 관리는 기존의 클라이언트-서버 환경보다 훨씬 어려울 수 있습니다.

기존의 클라이언트-서버 환경에서는 클라이언트와 제한된 개수의 서버 사이에 직접 통신이 사용됩니다. 문제가 발생하면 개별 비즈니스 트랜잭션에 관련된 시스템이 몇 개뿐이므로 오류의 원인을 찾기가 대체로 쉽습니다. 트랜잭션에 직접 관련된 특정 시스템을 검사하여 문제의 원인을 격리할 수 있습니다.

그러나 웹 응용 프로그램 서버가 여러 클라이언트 서버 시스템 간에 응용 프로그램에 대한 액세스를 배포하는 중심 역할을 하는 경우에는 문제의 원인을 식별하기가 더 어렵습니다. 성능 저하, 오류 또는 작업 실패는 웹 서버가 연결된 인프라에 속한 사실상 모든 구성 요소 또는 컴퓨터로 인해 발생할 수 있습니다.

서비스 지향 아키텍처를 사용하면 응용 프로그램 성능과 가용성의 모니터링이 좀 더 복잡해집니다. SOA의 경우 느슨하게 연결된 서비스는 표준화된 통신을 사용하여 서로 다른 플랫폼에서 실행되는 응용 프로그램을 통합하고 확장합니다. 이러한 서비스는 비즈니스 논리를 기본 운영 체제나 플랫폼과 분리하므로 조직에서는 민첩성을 높이고 시장이나 제품의 변화에 보다 빠르게 대처할 수 있습니다. 각 서비스가 복잡한 또는 여러 단계로 이루어진 비즈니스 프로세스의 특정 부분을 처리하도록 설계할 수 있어 이기종 환경에서 종속성 체인을 만들 수 있습니다.

서비스 지향 아키텍처를 사용하면 조직에서 서비스를 다시 사용하거나, 수정하거나, 독립된 구성 요소로서 대체할 수 있으므로 응용 프로그램을 보다 빠르고 비용 효율적인 방법으로 개발 및 배포할 수 있습니다. 그러나 이와 같이 효율적인 모듈 방식 응용 프로그램 아키텍처도 응용 프로그램 관리 측면에서는 문제점을 가지고 있습니다.

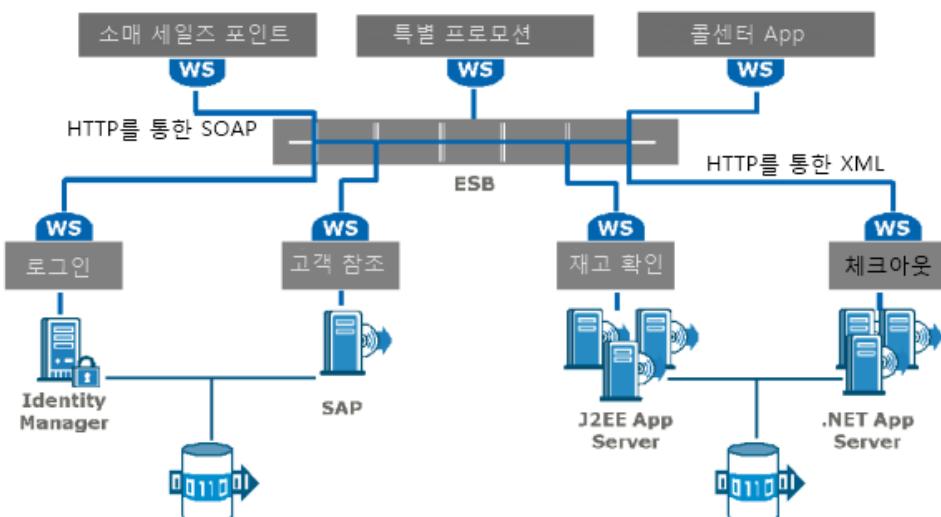
SOA 인프라의 일반적인 구성 요소

SOA를 사용하면 비즈니스 프로세스의 구현과 통합이 간단해지지만 기본 SOA 인프라는 일반적으로 여러 구성 요소의 복잡한 상호 작용을 필요로 합니다. 예를 들어 단순한 비즈니스 트랜잭션을 완료하는 데는 일반적으로 서로 다른 프로토콜을 사용하여 메시지를 교환하는 여러 구성 요소에서 실행되는 여러 개의 서비스가 관련됩니다. 트랜잭션의 주요 구성 요소를 모니터링하려면 서비스 간의 통신 방법, 서비스 간에 요청과 응답이 라우트되는 방법, 서비스 간의 종속성, 심각한 병목 현상이 발생하는 지점을 파악할 수 있어야 합니다.

구체적인 모니터링 요구 사항은 SOA 구현에 따라 다르지만 일반적인 SOA 인프라는 다음을 포함합니다.

- 표준 인터페이스를 사용하여 클라이언트 측 및 서버 측 구성 요소가 서로 메시지를 교환할 수 있도록 하는 여러 가지 서비스
- 구성 요소 간에 요청과 응답을 라우트, 변환 및 전달할 수 있도록 하는 메시지 처리 시스템
- 외부 또는 레거시 시스템이 배포된 서비스를 연결하고 사용할 수 있도록 하는 어댑터
- 사용 가능한 서비스에 대한 정보를 기록하고 게시하는 레지스트리

다음 그림은 메시지가 올바른 대상으로 전달되도록 제어하는 *ESB(Enterprise Service Bus)*를 통해 WS(웹 서비스)가 서로 통신할 수 있도록 하는 SOAP(Simple Object Access Protocol) 및 XML(eXtensible Markup Language)을 사용하는 단순한 SOA 인프라를 보여 줍니다.



SOA 인프라는 재사용 및 수정이 가능한 식으로 제공되는 구성 요소로 이루어지므로 이러한 환경의 트랜잭션에는 일반적으로 더 많은 구성 요소가 관련됩니다. 트랜잭션에 관련된 구성 요소의 수가 증가할수록 트랜잭션 흐름의 추적이 더 어려워집니다. 프로세스에서 프로세스로, 플랫폼에서 플랫폼으로, 그리고 종종 서로 다른 전송 프로토콜 간에 전달되는 메시지의 수가 많아짐에 따라 잠재적인 실패 지점도 증가합니다. 또한 조직에서 SOA 환경에 응용 프로그램을 배포하는 경우에는 점점 더 많은 수의 중요 서비스를 이 SOA 환경으로 이동하는 경향이 있기 때문에 비즈니스의 건전성을 위해 이 인프라의 모니터링이 갖는 중요성이 더욱 커집니다.

CA APM for SOA 는 SOA 인프라의 건전성과 SOA 트랜잭션에 참여하는 구성 요소의 실시간 성능에 대한 가시성을 제공하여 이러한 문제를 해결합니다.

CA Introscope®를 사용하여 SOA 성능을 모니터링하는 방법

CA APM for SOA 는 SOA 환경의 응용 프로그램에 대한 실시간 및 시간별 뷔를 제공합니다. CA Introscope 의 20 핵심 기능을 확장하여 사용자가 서비스를 요청하는 프런트엔드에서 요청을 수행하는 데 사용되는 백엔드 시스템에 이르기까지 웹 서비스 트랜잭션을 모니터링하고 추적할 수 있게 해줍니다. CA Introscope 를 20 CA APM for SOA 와 함께 사용하면 응용 프로그램 소스 코드나 SOA 아키텍처에 액세스할 필요 없이 SOA 클라이언트 및 서버 성능을 사전 대응적으로 모니터링하고, 인시던트를 심사하고, 서비스 관련 문제를 자세히 분석 할 수 있습니다.

CA APM for SOA 를 CA Introscope 에 20 추가하면 다음을 제공하여 SOA 환경을 모니터링하는 데 도움이 됩니다.

- 환경의 사전 대응적 관리를 위한 SOA 클라이언트/서버 건전성 및 성능에 대한 대시보드 가시성
- 문제를 웹 서비스, 응용 프로그램 또는 백엔드로 격리하는 데 도움이 되는 개요 및 세부 메트릭
- 한 구성 요소의 문제가 다른 구성에 미칠 수 있는 영향을 평가하는 데 도움이 되는 에이전트, 서비스 및 작업 간의 종속성에 대한 시각적 표현
- 플랫폼, 전송 프로토콜 및 응용 프로그램 서버 간의 상호 관련된 트랜잭션 추적
- 웹 서비스 구성 요소가 제공하는 응용 프로그램 오류 또는 SOAP 결함에 대한 가시성 및 컨텍스트

SOA 클라이언트와 서버의 모니터링 및 응용 프로그램 관리를 지원하는 이러한 기본 기능 이외에도 CA Introscope 와 21 CA APM for SOA 를 필요에 맞게 사용자 지정할 수 있는 기능이 제공됩니다. 예를 들어 다음을 수행할 수 있습니다.

- 웹 서비스 호출 수신 및 완료를 확인하여 웹 서비스 감사 추적을 구성
- 웹 서비스의 응용 프로그램 그룹을 생성하고 그룹이 응답 시간, 트랜잭션 양 또는 오류 임계값을 초과하는지 확인
- 웹 서비스 클라이언트 및 서비스에 대한 집계 메트릭을 수집하는 가상 에이전트 구성

웹 서비스 클라이언트 및 서버 모니터링

대부분의 경우 웹 서비스 트랜잭션은 웹 서비스에 요청을 하는 클라이언트와 이 요청에 응답하는 서버로 구성됩니다. 예를 들어 사용자나 프로그램이 주가를 요청할 수 있도록 하는 Java 클래스는 웹 서비스 클라이언트로 간주할 수 있습니다. 이 요청에 응하여 가격을 제공하는 Stock Quote 웹 서비스는 웹 서비스 서버로 간주됩니다.

에이전트는 클라이언트 및 서버 메트릭을 따로 추적하고 표시하여 클라이언트 측 성능과 서버 측 성능을 따로 모니터링하고 각각 임계값을 설정할 수 있도록 합니다. 또한 이렇게 클라이언트 측과 서버 측을 분리하면 적절한 컨텍스트에서 종속성을 확인할 수 있습니다.

대개의 경우 웹 서비스를 위한 클라이언트와 서버는 별도의 JVM(Java Virtual Machine) 또는 Microsoft CLR(Common Language Runtime) 환경에서 실행됩니다. 따라서 클라이언트 측 및 서버 측 작업에 대한 메트릭은 일반적으로 두 개 이상의 에이전트를 통해 보고됩니다. 클라이언트 및 서버에 대한 메트릭을 함께 집계하여 보려면 [가상 에이전트](#) (페이지 50)를 구성하십시오.

사전 대응적 관리를 위한 SOA 특정 대시보드 사용

CA APM for SOA를 사용하면 중요 서비스를 프로덕션 부하가 있는 상태에서도 24시간 실시간으로 모니터링 할 수 있습니다. SOA 특정 대시보드는 핵심 성능 표시기를 요약해서 보여 주며 응답 시간이 증가하거나 부하 증가 또는 처리량 감소와 같은 비정상적인 동작이 있을 때 초기에 알려주는 기본 임계값 및 경고 표시기를 제공합니다. 대시보드 및 경고 표시기를 통해 잠재적 문제를 빠르게 사전에 식별할 수 있으며 그런 다음 문제를 자세히 분석하여 문제의 본질과 해결 책임자를 결정할 수 있습니다.

CA Introscope 및 CA APM for SOA 대시보드를 사용하면 웹 서비스 클라이언트 및 서버 성능을 지속적으로 모니터링하여 잠재적 문제가 나타나는 경우 한눈에 파악할 수 있습니다. 대개의 경우 이러한 사전 대응적 조치를 취하면 문제가 발생한 구성 요소가 고객이나 최종 사용자에게 영향을 미치기 전에 이를 파악하고 수정할 수 있습니다.

종속성 보기 및 심사와 진단을 위한 메트릭 사용

복잡한 SOA 환경에서 문제가 발생하면 어디서부터 원인을 찾아야 할지 파악하기가 어렵습니다. 응용 프로그램 아키텍처에서 모듈식으로 구성된 부분이 많을수록 각 트랜잭션에 관련되는 구성 요소도 많고 잠재적 실패 지점도 증가합니다.

대시보드 및 경고 표시기는 잠재적 문제를 초기에 알 수 있도록 합니다. SOA 종속성 맵은 모듈형 구성 요소의 상호 관계와 이러한 관계가 성능에 미치는 영향에 대한 가시성을 제공합니다. 종속성을 통해 잠재적 병목 지점 또는 중요 작업의 위치를 분석하고 서로 종속되어 있는 서비스 간의 평균 응답 시간 또는 기타 메트릭을 비교할 수 있습니다.

상위 수준 뷰는 전반적인 응용 프로그램 성능 개요를 표시하고 잠재적 문제를 경고하며 구성 요소 간의 종속성 관계를 보여 줍니다. CA Introscope를 22 CA APM for SOA와 함께 사용하면 조직의 구축자와 개발자가 문제를 검토하고, 격리하고, 해결하는 데 사용할 수 있는 상위 수준의 요약 및 상세한 작동 수준의 메트릭이 모두 제공됩니다. 메트릭을 사용하여 웹 서비스 성능 및 작동을 평가하는 방법에 대한 자세한 내용은 [Investigator를 사용하여 SOA 성능 메트릭 보기](#) (페이지 58)를 참조하십시오.

SOA 구성 요소가 관련된 트랜잭션 추적

SOA 환경에서는 서로 느슨하게 연결된 구성 요소 간에 여러 프로토콜을 사용하여 메시지가 전달되는 경우가 많으므로 트랜잭션 흐름을 추적하기가 더 어렵습니다. 트랜잭션 추적은 웹 서비스가 관련된 개별 트랜잭션, 웹 서비스 결합의 수와 특징 및 구성 요소 상호 작용에 대한 자세한 정보를 제공합니다. 예를 들어 트랜잭션 추적은 트랜잭션의 각 세그먼트가 발생한 장소와 각 세그먼트의 실시간 성능을 식별합니다. 응용 프로그램 전문가는 이러한 정보를 사용하여 오류나 성능 저하의 근본적인 원인을 파악할 수 있습니다.

특정 웹 서비스의 응답 시간이 길어지는 경우와 같이 문제가 발생하면 SLA(서비스 수준 계약)가 위반되거나 최종 사용자가 영향을 받기 전에 경고를 통해 이해당사자에게 문제를 알릴 수 있습니다. 그러면 응용 프로그램 지원 담당자가 개별 트랜잭션을 모니터링하여, 트랜잭션에서 가장 많은 시간이 소요되는 부분 및 캡처된 각 트랜잭션을 완료하는 데 사용된 동기식 또는 비동기식 호출의 시퀀스를 시각적으로 확인함으로써 문제의 원인에 대해 보다 자세한 정보를 수집할 수 있습니다.

여러 플랫폼 및 전송 모니터링

CA Introscope 를 23 CA APM for SOA 와 함께 사용하면 여러 운영 체제, 응용 프로그램 플랫폼 또는 전송 프로토콜이 관련된 트랜잭션을 포함하여 전체 인프라를 모니터링하고 관리할 수 있습니다. 일반적인 SOA 환경에서는 트랜잭션에서 SOAP, XML, HTTP, HTTPS 및 JMS 전송 프로토콜을 조합하여 사용하고 여러 응용 프로그램 서버 환경에서 처리가 이루어지는 경우가 많습니다.

CA APM for SOA 를 사용하면 지원되는 모든 J2EE 또는 .NET 서버의 조합에서 트랜잭션을 추적할 수 있습니다. 예를 들어 SAP NetWeaver 또는 .NET 응용 프로그램 서버에서 실행 중인 서비스에 요청을 전송하는 WebLogic 에서 실행되는 서비스를 포함하는 트랜잭션의 전체 경로를 추적할 수 있습니다. 사용할 수 있게 설정된 추가 SOA 확장에 따라 Oracle Service Bus, IBM WebSphere Process Server, IBM WebSphere Enterprise Service Bus, TIBCO BusinessWorks, webMethods Integration Server 또는 IBM WebSphere MQ 구성 요소의 모든 조합에서 트랜잭션을 모니터링 할 수도 있습니다.

또한 SOA 환경에는 어댑터를 통해 SOA 인프라에 연결된 복합 응용 프로그램과 SOAP 및 XML 처럼 표준화된 프로토콜을 사용하는 새로운 응용 프로그램이 모두 포함된 경우가 많습니다. 이러한 환경에는 레거시 응용 프로그램과 새로운 응용 프로그램이 모두 포함되므로 다양한 전송 및 페이로드 유형을 사용하는 응용 프로그램을 모니터링해야 할 수 있습니다. 예를 들어 CA APM for SOA 를 사용하면 SOAP over HTTP, SOAP over JMS, XML over HTTP 또는 다른 메시징 전송 및 프로토콜을 사용하는 응용 프로그램을 모니터링할 수 있습니다.

SOA 모니터링 아키텍처 이해

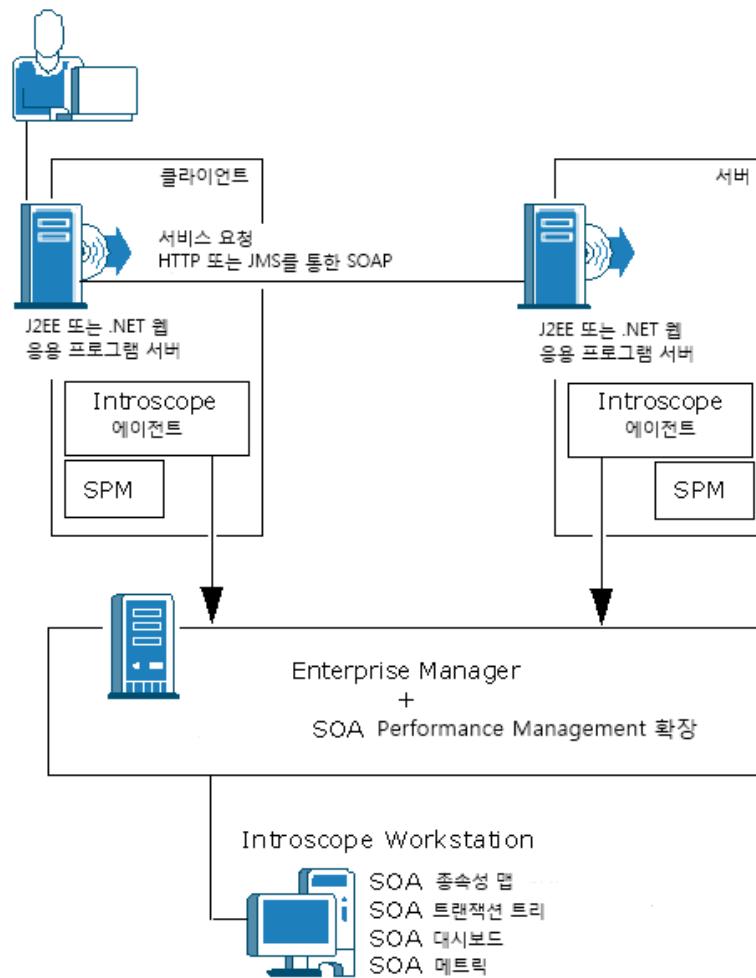
CA APM for SOA 에서는 CA Introscope 를 24 위하여 다음과 같은 경량 구성 요소를 제공합니다.

- Java 또는 .NET 에이전트를 위한 SOA 특정 확장
- Enterprise Manager 를 위한 SOA 특정 관리 모듈

에이전트는 에이전트 확장을 통해 지원되는 응용 프로그램 서버에서 Java 및 .NET 기반 웹 서비스를 모니터링하고 이러한 웹 서비스의 성능에 대한 정보를 Enterprise Manager 로 보낼 수 있습니다.

Enterprise Manager 에 대한 SOA 특정 확장을 사용하면 에이전트에서 수집한 SOA 특정 정보를 CA Introscope?Workstation 을 사용하여 SOA 특정 종속성 맵, 대시보드, Investigator 노드 및 탭에 표시할 수 있습니다.

다음 그림에서는 두 대의 응용 프로그램 서버를 위한 에이전트 및 CA APM의 기본 아키텍처를 보여 줍니다. 응용 프로그램 서버는 웹 서비스 요청을 보내고 받으며 CA APM for SOA 가 사용하도록 설정된 단일 Enterprise Manager 에 웹 서비스 데이터를 보고합니다.

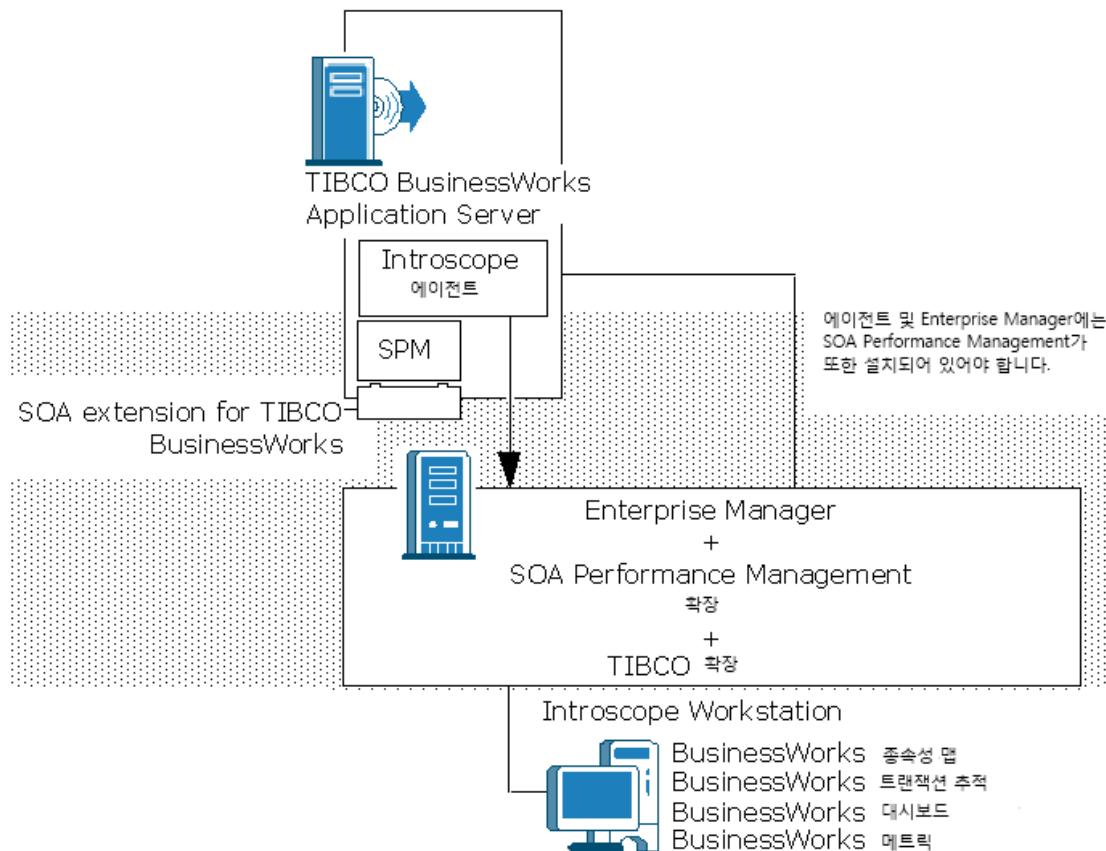


추가 SOA 플랫폼 지원

CA APM for SOA에서 제공하는 기능 외에도 몇 개의 추가적인 SOA 플랫폼을 모니터링할 수 있습니다. 환경에 따라 다음과 같은 추가 SOA 플랫폼을 모니터링할 수 있습니다.

- OSB(Oracle Service Bus)는 Oracle WebLogic에 대한 메시지 배포 및 변환을 처리하는 Enterprise Service Bus입니다.
- TIBCO BusinessWorks는 비즈니스 서비스 개발 및 처리 엔진입니다.
- TIBCO Enterprise Message Service는 큐, 항목 및 브리지, 멀티캐스트 채널, 경로와 같은 고급 구성 요소를 사용하여 엔터프라이즈 전체에서 시스템 간의 동기 및 비동기 통신을 가능하게 합니다.
- webMethods Broker는 문서를 게시하고 제공하기 위한 비동기 처리 및 메시지 처리 서비스를 제공합니다.
- webMethods Integration Server는 비즈니스 프로세스와 웹 서비스의 작성, 조정 및 통합을 가능하게 하는 인프라 구성 요소를 제공합니다.
- IBM WebSphere Process Server(WPS) with IBM WebSphere Enterprise Service Bus(WESB)는 WebSphere Enterprise Service Bus를 통해 메시지를 처리하는 기능이 포함된 통합 플랫폼입니다.
- 독립 실행형 제품으로서의 IBM WESB(WebSphere Enterprise Service Bus). WebSphere Enterprise Service Bus는 서비스와 클라이언트 간의 메시지 흐름, 데이터 변환 및 라우팅을 조정 흐름 및 조정 기본을 통해 관리합니다.

대부분의 SOA 플랫폼 확장은 에이전트와 Enterprise Manager에 파일을 추가합니다. 예를 들어 다음 그림은 응용 프로그램 서버와 Enterprise Manager에 SOA extension for TIBCO BusinessWorks를 추가하는 경우를 보여줍니다.



일부 SOA 확장에는 에이전트를 사용할 수 없습니다. 예를 들어 SOA extension for TIBCO Enterprise Message Service는 Standalone Agent를 사용합니다. SOA 확장에 대한 자세한 내용은 해당 확장에 대한 장을 참조하십시오.

제 2 장: CA APM for SOA 설치 및 구성

이 섹션은 다음 항목을 포함하고 있습니다.

[설치 및 구성](#) (페이지 29)

[CA Introscope® 구성 요소 및 버전](#) (페이지 30)

[CA APM for SOA 를 에이전트에 추가하는 방법](#) (페이지 33)

[Enterprise Manager](#)에서 확장을 사용하도록 설정 (페이지 41)

[CA APM for SOA 배포 확인](#) (페이지 44)

[CA APM for SOA 제거](#) (페이지 45)

설치 및 구성

에이전트 및 Enterprise Manager에서 다음과 같은 방법으로 CA APM for SOA를 설치하고 구성할 수 있습니다.

에이전트 설치 여부	CA APM for SOA 사용 여부	다음 단계
예	예	Enterprise Manager 에서 확장을 사용하도록 설정 (페이지 41)
예	아니요	CA APM for SOA 를 수동으로 에이전트에 추가 (페이지 36)
아니요	아니요	CA APM for SOA 사전 요구 사항 검토 (페이지 30) CA APM for SOA 를 에이전트에 추가 (페이지 33) 참고: 이 정보와 CA APM Java Agent 구현 안내서 또는 CA APM .NET 에이전트 구현 안내서의 정보를 함께 사용하여 설치를 완료하십시오.

CA Introscope® 구성 요소 및 버전

CA APM for SOA 는 서비스 지향 아키텍처 및 웹 서비스의 모니터링을 가능하게 하는 CA Introscope 확장을 제공합니다. CA APM for SOA 는 핵심 CA Introscope 구성 요소를 설치할 때 함께 설치하거나 핵심 CA Introscope 구성 요소를 설치한 후에 별도로 설치하거나 CA APM for SOA 를 CA Introscope®와 함께 설치하거나 CA APM for SOA 를 별도로 설치할 때 모두 다음의 CA Introscope 구성 요소 및 버전을 설치해야 합니다.

- Java Agent 또는 .NET 에이전트의 버전은 CA Introscope®와 동일해야 합니다.
- Enterprise Manager 의 버전은 에이전트의 버전과 동일하거나 에이전트보다 높아야 합니다.

에이전트의 기본 시스템 요구 사항

CA APM for SOA 는 Enterprise Manager 와 에이전트에 파일을 추가합니다. CA APM for SOA 는 에이전트를 확장하므로 처리, 메모리 및 사용 가능한 디스크 공간에 대한 최소 시스템 요구 사항은 에이전트의 요구 사항과 동일합니다.

CA APM 시스템 요구 사항에 대한 구체적인 정보는 다음 안내서를 참조하십시오.

- 기본 메모리 및 디스크 공간 요구 사항은 *CA APM 설치 및 업그레이드 안내서*를 참조하십시오.
- 처리 부하 조정 및 CPU 등의 시스템 리소스 관리에 대한 지침은 *CA APM Sizing and Performance Guide*(CA APM 크기 조정 및 성능 안내서)를 참조하십시오.
- 최소 JVM 및 JRE 버전과 관련 JVM 메모리 요구 사항은 *CA APM Java Agent 구현 안내서*를 참조하십시오.

CA APM for SOA 사용 에이전트는 기본 에이전트 요구 사항에 더하여 지원되는 SOAP 엔진을 로컬 운영 환경에서 사용할 수 있어야 합니다. 지원되는 SOAP 엔진을 보려면 *CA APM 호환성 안내서*를 참조하십시오. 사용하는 SOAP 엔진에 따라 에이전트가 지원할 수 있는 SOAP 스택 구현 및 웹 서비스 메시지 교환 유형이 결정됩니다.

에이전트에 대한 SOAP 및 응용 프로그램 서버 요구 사항

사용하는 운영 체제와 응용 프로그램 서버에 따라 웹 서비스 메시지를 처리하는 데 사용할 수 있는 SOAP 엔진 및 SOAP 스택 구현 방법이 결정됩니다. 예를 들어 Apache Axis 2.0 을 사용하는 컴퓨터의 응용 프로그램은 JAX-RPC 나 JAX-WS 를 사용하여 SOAP 메시지를 처리할 수 있습니다.

Java Agent에 대한 기본 SOAP 엔진 지원

CA APM for SOA 를 Java Agent 에 추가하면 에이전트는 다양한 SOAP 엔진 및 API(응용 프로그램 프로그래밍 인터페이스)를 지원할 수 있습니다. 응용 프로그램 서버 및 버전에 따라 에이전트는 네이티브, JAX_RPC 또는 JAX-WS 표준을 준수하는 SOAP 스택 구현을 지원합니다.

참고: 에이전트가 현재 버전에서 지원하는 응용 프로그램 서버 및 SOAP 스택 구현의 전체 목록은 *Compatibility Guide*(호환성 안내서)의 CA APM for SOA 단원을 참조하십시오.

.NET 에이전트에 대한 기본 SOAP 엔진 지원

기본적으로 .NET 에이전트는 ASP.NET 웹 서비스의 모니터링을 지원합니다. CA APM for SOA 를 .NET 에이전트에 추가하면 에이전트는 표준 웹 서비스 이름 지정 및 추가적인 종속성 관련 메트릭과 현재 사용 중인 SOAP 통신 서비스에 대한 종속성 기반 매핑을 제공합니다. .NET Framework 를 사용하는 .NET 에이전트에서 웹 서비스 모니터링을 지원하려면 다음의 웹 서비스를 사용할 수 있습니다.

- 표준 ASP.NET 웹 서비스
- WCF(Windows Communication Foundation) 웹 서비스

.NET Framework 외에도 ASP.NET 웹 서비스를 모니터링하려면 다음의 특성이 필요합니다.

- 서버 측 웹 서비스에 WebMethod 특성이 적용되어 있어야 합니다.
- 클라이언트 측 웹 서비스에는 SOAPDocumentMethod, HttpMethod 또는 SoapRpcMethod 특성이 적용되어 있어야 합니다.

참고: SOAP 지원에 대한 자세한 내용은 *Compatibility Guide*(호환성 안내서)의 "SOA Performance Management"(SOA 성능 관리) 단원을 참조하십시오. 이러한 특성 또는 이 특성을 메서드나 클래스에 적용하는 방법에 대한 자세한 내용은 MSDN(Microsoft Developer Network)을 참조하십시오.

디렉터리 및 파일 명명 규칙 이해

이 안내서에서는 파일 이름과 디렉터리 경로에 다음과 같은 규칙을 사용합니다.

표시	의미
<code><Agent_Home></code>	에이전트가 설치된 최상위 디렉터리입니다. 일반적으로 이 디렉터리는 이름이 <i>wily</i> 입니다.
<code><EM_Home></code>	Enterprise Manager 가 설치된 최상위 디렉터리입니다.
<code><version></code>	파일 이름에 포함되거나 사용자 인터페이스에 표시된 버전별 식별자입니다. 예를 들어 다음 파일 이름은 <i>com.wily.introscope.soa.dependencymap_<version>.jar</i> 다음과 같은 버전별 파일 이름을 나타냅니다. <i>com.wily.introscope.soa.dependencymap_v9.1.0.0.jar</i>
슬래시(/) 경로 구분 기호	사용 중인 플랫폼에서 디렉터리 이름에 사용되는 경로 구분 기호입니다. UNIX 플랫폼 및 이 안내서 전체의 예제에서는 슬래시(/)가 사용되지만 사용 중인 플랫폼에 적절한 구분 기호를 사용해야 합니다.
달러 기호(\$) 환경 변수	플랫폼에서 사용되는 환경 변수 표기법입니다. UNIX 플랫폼 및 이 안내서 전체의 예제에서는 달러 기호(\$)가 사용되지만 사용 중인 플랫폼에 적절한 문자를 사용해야 합니다.

이전 버전에서 업그레이드하기 전에 백업

이전 버전의 Web Services Manager 또는 CA APM for SOA 에서 업그레이드하는 경우에는 현재 환경을 백업하고 기존 파일을 백업 디렉터리에 저장한 후 CA APM for SOA 에이전트 또는 Enterprise Manager 파일을 추가하십시오.

이전 버전에서 업그레이드하는 경우에는 다음과 같이 하십시오.

- 기존 <Agent_Home>/IntroscopeAgent.profile 파일을 백업 디렉터리에 복사합니다.
- 기존 <Agent_Home>/webservices.pbd 파일을 백업 디렉터리에 복사합니다.
- 기존 <EM_Home>/modules/SPM_ManagementModule.jar 파일을 백업 디렉터리에 복사합니다.

이전 버전에서 업그레이드하는 경우에 에이전트가 .NET 에이전트이면 다음과 같이 하십시오.

- 새 버전의 .NET 에이전트를 설치하기 전에 이전 버전의 .NET 에이전트를 제거하십시오.
- 업그레이드하려는 .NET 에이전트 및 CA APM for SOA 파일의 버전 수준이 같은지 확인하십시오. 버전 정보가 일치하지 않으면 오류가 발생하여 응용 프로그램을 모니터링하지 못할 수 있습니다.

CA APM for SOA 를 에이전트에 추가하는 방법

에이전트를 대화식으로 설치하거나 응답 파일을 사용하여 자동으로 설치할 때 CA APM for SOA 및 SOA 관련 확장을 에이전트에 자동으로 추가할 수 있습니다. 또한 에이전트 설치 후 CA APM for SOA 및 기타 SOA 관련 확장을 에이전트에 수동으로 추가할 수도 있습니다.

선택한 설치 유형에 따라 적절한 작업을 수행하십시오.

- [Standalone Agent 설치 관리자에서 CA APM for SOA 선택](#) (페이지 34)
- [자동 설치 중 CA APM for SOA 추가](#) (페이지 35)
- [CA APM for SOA 를 수동으로 에이전트에 추가](#) (페이지 36)

에이전트를 설치하거나 CA APM for SOA 가 포함되도록 업데이트한 후에는 [설치 후 에이전트 속성 구성](#) (페이지 38)에서 응용 프로그램 환경 구성에 대한 정보를 참조하십시오. Oracle Service Bus 또는 TIBCO BusinessWorks 와 같은 다른 SOA 플랫폼을 모니터링하려면 해당 파일을 응답 파일을 사용하여 대화식으로 설치하거나, 에이전트를 설치하고 CA APM for SOA 를 선택한 후 수동으로 설치할 수도 있습니다. 자세한 내용은 [SOA 플랫폼 확장을 사용하도록 설정하는 방법](#) (페이지 38)을 참조하십시오.

Standalone Agent 설치 관리자에서 CA APM for SOA 선택

에이전트를 대화식으로 설치하는 경우 설치할 때 CA APM for SOA 및 기타 모니터링 확장을 추가하도록 선택할 수 있습니다. Standalone Agent 에 표시되는 옵션은 선택하는 에이전트 및 응용 프로그램 서버에 따라 다릅니다. 예:

- CA APM for SOA 를 사용하도록 설정하려면 "기본값", "JBoss", "Tomcat", "WebLogic" 또는 "WebSphere"를 응용 프로그램 서버로 선택합니다.
- CA APM for Oracle Service Bus 를 사용하도록 설정하려면 "WebLogic"을 응용 프로그램 서버로 선택합니다.
- CA APM for TIBCO BusinessWorks 또는 CA APM for webMethods Integration Server 를 사용하도록 설정하려면 "기본값"을 응용 프로그램 서버로 선택합니다.
- CA APM for WebSphere Process Server with WESB 또는 CA APM for WebSphere Enterprise Service Bus(WESB)를 독립 실행형 제품으로 사용하도록 설정하려면 "WebSphere"를 응용 프로그램 서버로 선택합니다.

예를 들어 Windows 에서 Standalone Agent 설치 관리자를 대화식으로 실행하고 기본값을 응용 프로그램 서버로 선택하면 CA APM for SOA 및 CA APM for TIBCO BusinessWorks 또는 CA APM for webMethods Integration Server 를 모니터링 옵션으로 선택할 수 있습니다.

Standalone Agent 설치 관리자에서 CA APM for SOA 옵션을 선택하면 에이전트에 대한 관련 파일이 자동으로 에이전트의 홈 디렉터리에 있는 적절한 디렉터리로 복사되고 에이전트의 프로필이 SOA 환경을 모니터링하기 위한 적절한 ProbeBuilder 지시문 파일을 포함하도록 자동으로 업데이트됩니다. 예를 들어 *appmap-soa.pbd* 및 *webservices.pbd* 가 자동으로 *introscope.autoprobe.directivesFile*에 적절하게 추가됩니다.

Standalone Agent 설치 관리자를 사용하여 CA APM for SOA 를 설치하는 경우 Java 및 .NET 에이전트를 [구성](#) (페이지 38)하기 위하여 다음 단계를 수행하십시오.

자동 모드를 사용하여 에이전트에 대해 CA APM for SOA 를 사용하도록 설정

에이전트를 자동 모드로 설치하는 경우에는 응답 파일의 설정을 사용할 수 있습니다. 이 설정을 통해 에이전트에 대해 CA APM for SOA 를 사용하도록 설정할지 여부를 지정할 수 있습니다. 응답 파일을 사용하면 에이전트 속성을 사용자 개입 없이 원격으로 설치하고 구성할 수 있습니다. 이 방법을 사용하면 스크립트 또는 기타 소프트웨어 제공 옵션을 통해 프로세스를 자동화할 수 있습니다.

참고: 자동 모드에서 에이전트를 설치하는 방법에 대한 자세한 내용은 *CA APM Java Agent* 구현 안내서 또는 *CA APM .NET* 에이전트 구현 안내서를 참조하십시오.

다음 단계를 수행하십시오.

1. Standalone Agent 설치 관리자와 같은 디렉터리에 있는 *SampleResponseFile.Agent.txt* 파일을 엽니다.
2. 원하는 설치 설정에 맞게 *SampleResponseFile.Agent.txt* 파일을 편집합니다.
3. *shouldEnableSPM* 속성을 *true*로 설정하여 CA APM for SOA 를 에이전트에 추가합니다. 예:
`shouldEnableSPM=true`

이 속성을 *false*로 설정하면 SPM 특정 파일이 자동으로 *<Agent_Home>/examples* 디렉터리로 복사되지만 에이전트 프로필은 업데이트되지 않습니다.

응답 파일의 속성을 사용하여 특정 SOA 플랫폼의 모니터링을 활성화할 수도 있습니다. 추가 SOA 플랫폼에 대한 모니터링을 사용하도록 설정하려면 *shouldEnableSPM* 속성을 *true*로 설정하십시오.

4. SampleResponseFile.Agent.txt 파일을 저장합니다.
 5. 명령줄에 적절한 명령을 입력하여 설치 관리자를 호출합니다.
- 응답 파일을 사용하여 CA APM for SOA 를 자동으로 설치하는 경우 다음 단계는 [설치 후 에이전트 속성 구성](#) (페이지 38)입니다.

추가 정보:

[CA Introscope® 구성 요소 및 버전](#) (페이지 30)
[SOA 플랫폼 확장을 사용하도록 설정하는 방법](#) (페이지 38)

CA APM for SOA 를 수동으로 에이전트에 추가

Standalone Agent 설치 관리자 또는 자동 설치용 응답 파일에서 CA APM for SOA 를 사용하도록 설정하지 않은 경우에는 설치 후 에이전트를 수동으로 업데이트할 수 있습니다. CA APM for SOA 가 포함되도록 에이전트를 수동으로 업데이트하려면 에이전트 프로필을 수정해야 합니다. 수동으로 업데이트해야 하는 에이전트의 유형에 따라 적절한 단원을 참조하십시오.

- [Java Agent 를 수동으로 업데이트](#) (페이지 36)
- [.NET 에이전트를 수동으로 업데이트](#) (페이지 37)

CA APM for SOA 를 사용하도록 Java Agent 를 수동으로 업데이트

설치 후 CA APM for SOA 를 사용하도록 Java Agent 를 수동으로 업데이트할 수 있습니다. 파일을 <Agent_Home>/examples 디렉터리에서 <Agent_Home>/core/ext 디렉터리로 이동합니다. 그런 다음 CA APM for SOA 에 대한 적절한 ProbeBuilder 지시문 파일이 포함되도록 에이전트 프로필을 편집합니다.

다음 단계를 수행하십시오.

1. <Agent_Home>/examples/SOAPerformanceManagement/ext 디렉터리로 이동합니다.
2. 디렉터리의 파일을 복사하여 <Agent_Home>/core/ext 디렉터리에 붙여 넣습니다.
3. 응용 프로그램 서버를 중지합니다.
4. <Agent_Home>/core/config/IntroscopeAgent.profile 파일을 텍스트 편집기에서 엽니다.

5. IntroscopeAgent.profile 파일의 introscope.autoprobe.directivesFile 속성에 spm.pbl 값을 추가합니다.
예:
introscope.autoprobe.directivesFile=websphere-full.pbl,hotdeploy,spm.pbl
6. 기본 설정을 변경하려면 IntroscopeAgent.profile 파일에서 추가 에이전트 속성을 수정합니다. 그런 다음 파일을 저장하고 텍스트 편집기를 종료합니다.
7. 응용 프로그램 서버를 다시 시작합니다.
응용 프로그램 서버는 에이전트를 다시 시작하고 응용 프로그램에 SOA 성능 모니터링을 사용하도록 지시합니다.
Java Agent 프로필을 수동으로 구성한 후에는 [설치 후 에이전트 속성 구성](#) (페이지 38)을 수행합니다.

CA APM for SOA 를 사용하도록 .NET 에이전트를 수동으로 업데이트

설치 후 CA APM for SOA 를 사용하도록 .NET 에이전트를 수동으로 업데이트할 수 있습니다. 파일을 <Agent_Home>/examples 디렉터리에서 <Agent_Home>/ext 디렉터리로 이동하고 CA APM for SOA 파일이 포함되도록 기본 ProbeBuilder 목록을 편집합니다. 에이전트 프로필을 업데이트한 후에는 [설치 후 에이전트 속성 구성](#) (페이지 38)을 수행합니다.

다음 단계를 수행하십시오.

1. <Agent_Home>/examples/SOAPerformanceManagement 폴더로 이동합니다.
2. 파일을 <Agent_Home>/examples/SOAPerformanceManagement/ext 폴더에서 <Agent_Home>/ext 폴더로 복사합니다.
3. wily.WCFServicesAgent.ext.dll 파일을 <Agent_Home>/examples/SOAPerformanceManagement/wcf/ext 폴더에서 <Agent_Home>/ext 폴더로 이동합니다.
4. IIS 응용 프로그램 서버를 중지합니다.
5. <Agent_Home>/IntroscopeAgent.profile 파일을 텍스트 편집기에서 열고 introscope.autoprobe.directivesFile 속성 설정을 확인합니다.
예:
introscope.autoprobe.directivesFile=default-full.pbl,hotdeploy
6. 변경 내용을 저장하고 IntroscopeAgent.profile 파일을 닫습니다.

7. introscope.autoprobe.directivesFile 속성에 대해 지정한 default-full.pbl 또는 default-typical.pbl 을 텍스트 편집기에서 엽니다.
8. default*.pbl 파일을 저장하고 닫습니다.
9. 구성을 마친 후 IIS 서버를 다시 시작합니다.

그러면 .NET 에이전트가 CA APM for SOA 를 사용하도록 업데이트됩니다.

설치 후 에이전트 속성 구성

설치 후 필요에 따라 다른 에이전트 속성을 구성하고 응용 프로그램 서버를 업데이트할 수 있습니다. 예를 들어 에이전트 프로필을 사용자 지정하여 모니터링되는 구성 요소 또는 상관 관계 정보의 처리를 조정할 수 있습니다. 응용 프로그램 환경에 따라 응용 프로그램 서버에서 설정을 구성하거나 서버 시작 스크립트를 수정해야 할 수도 있습니다.

참고: Java Agent 를 설치한 경우에는 *CA APM Java Agent 구현 안내서*에서 응용 프로그램 환경 구성에 대한 자세한 내용을 확인하십시오. .NET 에이전트 설치와 구성 및 .NET 환경에 대한 속성 설정에 대한 자세한 내용은 *CA APM .NET 에이전트 구현 안내서*를 참조하십시오. Java 또는 .NET 에이전트를 CA APM for SOA 와 함께 작동하도록 하기 위해 필요한 별도의 변경 작업은 없습니다.

SOA 플랫폼 확장을 사용하도록 설정하는 방법

CA APM for SOA 를 사용하도록 설정한 에이전트가 있는 경우 모니터링을 다음 SOA 플랫폼의 구성 요소로 확장할 수 있습니다.

- AquaLogic Service Bus(ALSB) 및 Oracle Service Bus(OSB)
- IBM WebSphere Process Server(WPS) 및 WebSphere Enterprise Service Bus(WESB) 또는 WebSphere Enterprise Service Bus(WESB) 독립 실행형 제품
- TIBCO BusinessWorks 또는 TIBCO ActiveMatrix BusinessWorks(BW)
- Software AG webMethods Integration Server(IS)

응용 프로그램 서버 및 에이전트 환경에 따라 Standalone Agent 설치 관리자를 대화식으로 실행할 때, 응답 파일을 사용하여 에이전트를 자동으로 설치할 때 또는 에이전트 설치 후 수동으로 설치할 때 이 SOA 플랫폼에 대해 CA APM 을 사용하도록 설정할 수 있습니다.

에이전트와 CA APM for SOA 를 사용하는 SOA 관련 확장 이외에도 다음의 SOA 플랫폼을 모니터링하기 위한 독립 실행형 확장이 있습니다.

- TIBCO EMS(Enterprise Message Service)
- webMethods Broker

이러한 독립 실행형 확장은 에이전트와 별도로 설치되고 구성됩니다.

참고: 독립 실행형 확장의 설치와 구성에 대한 자세한 내용은 해당 단원을 참조하십시오.

SOA 관련 확장을 대화식으로 추가

Standalone Agent 설치 관리자를 대화식으로 실행할 때는 SOA 관련 확장을 선택하여 설치 관리자를 활성화할 수 있습니다. 선택 가능한 구체적인 모니터링 옵션은 선택한 응용 프로그램 서버에 따라 다릅니다. 예를 들어 WebLogic 을 응용 프로그램 서버로 선택한 경우에는 CA APM for SOA 및 CA APM for Oracle Service Bus 를 선택할 수 있습니다. 응용 프로그램 서버로 "기본값"을 선택한 경우에는 CA APM for SOA 및 CA APM for TIBCO BusinessWorks 또는 CA APM for webMethods Integration server 를 선택 할 수 있습니다.

설치 관리자에서 선택한 사항에 따라 적절한 파일이 자동으로 에이전트 홈 디렉터리의 적절한 디렉터리에 복사되고 적절한 ProbeBuilder 지시문이 자동으로 에이전트의 프로필에 추가됩니다.

자동 설치 응답 파일을 사용하여 SOA 관련 확장 추가

에이전트를 자동 모드로 설치하는 경우에는 응답 파일의 설정을 사용하여 에이전트에 대해 SOA 확장을 사용하도록 설정할지 여부를 지정할 수 있습니다.

자동 모드에서 에이전트에 대해 SOA 플랫폼 확장을 사용하도록 설정하려면

1. Standalone Agent 설치 관리자와 같은 디렉터리에 있는 *SampleResponseFile.Agent.txt* 파일을 엽니다.
2. 원하는 설치 설정에 맞게 *SampleResponseFile.Agent.txt* 파일을 편집합니다.

3. *shouldEnableSPM* 속성을 *true* 로 설정하여 CA APM for SOA 를 에이전트에 추가합니다. 예:

```
shouldEnableSPM=true
```

CA APM for Oracle Service Bus(OSB), CA APM for IBM WebSphere Process Server(WPS), CA APM for WebSphere Enterprise Service Bus(WESB), CA APM for TIBCO BusinessWorks(BW) 또는 CA APM for webMethods Integration Server 를 사용하도록 설정하려면 이 속성을 *true* 로 설정해야 합니다.

4. SOA 확장을 에이전트에 추가하기 위해 적절한 *shouldEnable** 속성을 *true* 로 설정합니다. 예:

- *shouldEnableSOAExtForOSB=true* 를 지정하면 CA APM for Oracle Service Bus(OSB)가 사용하도록 설정됩니다.
- *shouldEnableSOAExtForWPSandWESB=true* 를 지정하면 CA APM for IBM WebSphere Process Server(WPS) with WebSphere Enterprise Service Bus(WESB) 또는 CA APM for WebSphere Enterprise Service Bus 독립 실행형 제품이 사용하도록 설정됩니다.
- *shouldEnableSOAExtForTibcoBW=true* 를 지정하면 CA APM for TIBCO BusinessWorks 가 사용하도록 설정됩니다.
- *shouldEnableSOAExtForWebMethodsIS=true* 를 지정하면 CA APM for webMethods Integration Server 가 사용하도록 설정됩니다.

속성을 *false* 로 설정하면 확장별 파일이 *<Agent_Home>/examples* 디렉터리에 복사되지만 에이전트 프로필은 업데이트되지 않습니다. 에이전트 설치 후 확장을 사용하도록 설정하는 방법에 대해서는 [에이전트 설치 후 SOA 확장을 수동으로 추가](#) (페이지 41)를 참조하십시오.

5. *SampleResponseFile.Agent.txt* 파일을 저장합니다.

6. 명령줄에 적절한 명령을 입력하여 설치 관리자를 호출합니다.

에이전트 설치 후 SOA 관련 확장을 수동으로 추가

Standalone Agent 설치 관리자 또는 응답 파일에서 SOA 관련 확장을 사용하도록 설정하지 않은 경우에는 설치 후 에이전트를 수동으로 업데이트할 수 있습니다. 에이전트를 수동으로 업데이트하려면 일반적으로 다음과 같은 단계가 필요합니다.

- 파일을 적절한 `<Agent_Home>/examples` 디렉터리에서 `<Agent_Home>/core/ext` 디렉터리로 복사합니다.
- 플랫폼별 에이전트 확장에 대한 적절한 ProbeBuilder 지시문(.pbd 또는 .pbl)이 포함되도록 에이전트 프로필을 구성합니다.
- 에이전트를 호출하도록 서버를 구성하거나 서버를 다시 시작합니다.

하지만 에이전트를 수동으로 업데이트하려면 SOA 플랫폼에 따라 추가적인 플랫폼별 단계가 필요할 수 있습니다.

참고: 플랫폼별 단계에 대한 자세한 내용은 사용하도록 설정하고자 하는 SOA 확장에 해당하는 단원을 참조하십시오.

Enterprise Manager에서 확장을 사용하도록 설정

Management Module 및 CA APM for SOA 와 SOA 관련 확장에 대한 기타 파일은 Enterprise Manager를 설치할 때 기본적으로 `<EM_Home>/examples` 디렉터리에 설치됩니다. 예를 들어 CA APM for SOA에 대한 파일은 기본적으로 `<EM_Home>/examples/SOAPerformanceManagement` 디렉터리에 설치됩니다.

확장을 사용하도록 설정하려면 확장의 `<EM_Home>/examples` 하위 디렉터리에 있는 파일을 적절한 `<EM_Home>` 디렉터리로 이동하고 확장의 관리 모듈을 배포하십시오.

CA APM for SOA를 Enterprise Manager에 추가하려면

1. Enterprise Manager를 중지합니다.
2. `<EM_Home>/examples/SOAPerformanceManagement` 디렉터리로 이동합니다.

3. <EM_Home>/examples/SOAPerformanceManagement/config/modules 디렉터리의 내용을 복사하여 <EM_Home>/config/modules 디렉터리에 붙여 넣습니다.

<EM_Home>/examples/SOAPerformanceManagement/config/modules 디렉터리에는 CA APM for SOA 관리 모듈(SPM_ManagementModule.jar)이 있습니다. 클러스터 환경에 여러 개의 Enterprise Manager 가 있는 경우 이 파일만 MOM 컴퓨터로 사용 중인 Enterprise Manager 의 <EM_Home>/config/modules 디렉터리에 복사하십시오.
4. <EM_Home>/examples/SOAPerformanceManagement/ext 디렉터리의 내용을 복사하여 <EM_Home>/ext 디렉터리에 붙여 넣습니다.

<EM_Home>/examples/SOAPerformanceManagement/ext/xmltv 디렉터리에는 Workstation에서 SOA 개요, 편차 및 종속성 메트릭을 표시할 수 있도록 하는 CA APM for SOA 탭 정의가 있습니다.
5. <EM_Home>/examples/SOAPerformanceManagement/product/enterprisemanager/plugins 디렉터리를 복사하여 <EM_Home>/product/enterprisemanager/plugins 디렉터리에 붙여 넣습니다.

<EM_Home>/examples/SOAPerformanceManagement/product/enterprise manager/plugins 디렉터리에는 트랜잭션 추적, 편차 메트릭 및 SOA 종속성 맵을 지원하는 파일이 있습니다. Enterprise Manager 를 클러스터로 배포한 경우에는 <EM_Home>/examples/SOAPerformanceManagement/product/enterprise manager/plugins 의 파일을 클러스터의 모든 Enterprise Manager 에 있는 <EM_Home>/product/enterprisemanager/plugins 로 복사하십시오. Collector Enterprise Manager 에 파일이 없으면 종속성 맵이 제대로 표시되지 않습니다.
6. <EM_Home>/examples/SOAPerformanceManagement/ws-plugins 디렉터리를 복사하여 <EM_Home>/ws-plugins 디렉터리에 붙여 넣습니다.

<EM_Home>/examples/SOAPerformanceManagement/ws-plugins 디렉터리에는 SOA 종속성 맵, 종속성 메트릭 및 트랜잭션 추적을 지원하는 추가 파일이 있습니다.
7. Enterprise Manager 를 다시 시작합니다.

기본적으로 CA APM for SOA에는 종속성 맵의 관계를 새로 고치는 빈도 또는 보고할 편차 메트릭 등과 같은 작업을 제어하는 Enterprise Manager 속성 (페이지 363)이 포함되어 있습니다. 대부분의 조직에서는 기본 설정이 적절하며 이를 변경할 필요가 없습니다. 하지만 필요한 경우 속성을 수정할 수 있습니다.

추가적인 SOA 플랫폼에 대한 모니터링을 사용하도록 설정하려면 먼저 Enterprise Manager에서 CA APM for SOA를 사용하도록 설정하십시오.

플랫폼 특정 SOA 확장을 Enterprise Manager에 추가하려면

1. Enterprise Manager를 중지합니다.
2. 적절한 <EM_Home>/examples 디렉터리로 이동합니다. 예를 들어 다음 디렉터리 중 하나로 이동합니다.
 - SOAExtensionForOSB
 - SOAExtensionForTibcoBW
 - SOAExtensionForTibcoEMS
 - SOAExtensionForWebMethodsBroker
 - SOAExtensionForWebMethodsIS
 - SOAExtensionForWPSandWESB
3. <EM_Home>/examples/<extension>/config/modules 디렉터리의 내용을 복사하여 <EM_Home>/config/modules 디렉터리에 붙여 넣습니다. <EM_Home>/examples/<extension>/config/modules 디렉터리에는 SOA 확장용 관리 모듈이 있습니다. 클러스터 환경에 여러 개의 Enterprise Manager가 있는 경우에는 관리 모듈만 현재 MOM 컴퓨터로 사용 중인 Enterprise Manager의 modules 디렉터리로 복사해야 합니다. 또는 Enterprise Manager가 실행 중인 경우에는 관리 모듈 파일을 <EM_Home>/deploy 디렉터리로 복사할 수 있습니다.
4. <EM_Home>/examples/<extension>/ext/xmltv 디렉터리의 내용을 복사하여 <EM_Home>/ext/xmltv 디렉터리에 붙여 넣습니다. <EM_Home>/examples/<extension>/ext/xmltv 디렉터리에는 확장과 관련한 메트릭을 Workstation에 표시하는 데 사용할 수 있는 탭 정의가 있습니다.

5. 추가 디렉터리의 내용을 복사하여 <EM_Home> 디렉터리 아래의 해당 디렉터리에 붙여 넣습니다.

예를 들어

<EM_Home>/examples/<extension>/product/enterprisemanager/plugins 디렉터리의 내용을 <EM_Home>/product/enterprisemanager/plugins 디렉터리로 복사합니다.

참고: Enterprise Manager 를 클러스터로 배포한 경우에는 <EM_Home>/examples/SOAPerformanceManagement/product/enterprise manager/plugins 의 파일을 클러스터의 모든 Enterprise Manager 와 MOM 에 있는 <EM_Home>/product/enterprisemanager/plugins 로 복사하십시오.

추가 디렉터리는 사용하도록 설정하는 SOA 확장에 따라 다릅니다.

6. Enterprise Manager 를 다시 시작합니다.

참고: 특정 SOA 플랫폼의 구성 및 작업에 대한 자세한 내용은 해당 플랫폼에 해당하는 단원을 참조하십시오.

CA APM for SOA 배포 확인

웹 서비스 및 기타 SOA 구성 요소가 포함된 응용 프로그램을 호출하고 WebServices 노드가 Investigator 트리에 있는지 확인하여 CA APM for SOA 및 SOA 관련 확장의 배포를 확인할 수 있습니다.

다음 단계를 수행하십시오.

1. Enterprise Manager 가 실행되고 있지 않으면 시작합니다.
2. 응용 프로그램 서버가 실행되고 있지 않으면 시작합니다.

응용 프로그램 서버를 다시 시작하면 에이전트가 시작되고 SOA 성능 모니터링을 사용하도록 응용 프로그램에 지시됩니다.

3. 웹 서비스를 사용하는 응용 프로그램을 시작하고 하나 이상의 트랜잭션을 실행합니다.
4. Workstation 을 시작하고 Investigator 를 엽니다.
5. CA APM for SOA 가 설치된 에이전트를 확장합니다.

6. 에이전트 아래에서 "WebServices" > "클라이언트" 또는 "WebServices" > "서버" 노드에 액세스할 수 있는지 확인합니다.

트랜잭션에 같은 에이전트의 클라이언트 측 작업과 서버 측 작업이 모두 포함된 경우 에이전트에서 "WebServices" > "클라이언트" 및 "WebServices" > "서버" 노드를 모두 볼 수 있어야 합니다.

검색된 웹 서비스를 트랜잭션 실행에 사용한 경우 서비스 네임스페이스 노드를 확장하여 트랜잭션의 클라이언트 측 또는 서버 측과 연관된 메트릭을 볼 수 있습니다.

경우에 따라 웹 서비스 작동을 보이기 위해 트랜잭션을 시뮬레이션하는 샘플 응용 프로그램에서는 메트릭을 생성하지 않을 수 있습니다. 예를 들어 WebLogic 테스트 클라이언트는 CA APM for SOA 가 감지할 수 없는 WebLogic에서 호스팅하는 서비스에 대하여 실행됩니다.

7. 에이전트 노드를 선택한 다음 "개요" 탭을 클릭하여 Enterprise Manager `ws.overview.tv.xml` 파일이 제대로 로드되는지 확인합니다.

CA APM for SOA 제거

CA APM for SOA 및 SOA 관련 확장을 환경에서 제거하기로 결정한 경우 다음을 수행하십시오.

- 설치 및 구성된 모든 에이전트 확장을 제거합니다.
- 배포된 Enterprise Manager 확장을 제거합니다.

Java Agent에서 CA APM for SOA를 제거하려면

1. 응용 프로그램 서버를 중지합니다.
2. `<Agent_Home>/core/config` 디렉터리로 이동한 다음 `IntroscopeAgent.profile` 파일을 텍스트 편집기에서 엽니다.
3. `introscope.autoprobe.directivesFile` 속성에서 `appmap-soa.pbd` 및 `webservices.pbd` 값을 제거합니다. 파일을 저장하고 닫습니다.
4. `<Agent_Home>` 디렉터리에서 `webservices.pbd`를 제거합니다.
5. `<Agent_Home>/core/ext` 디렉터리로 이동하고 다음 파일을 제거합니다.
`WebServicesAgent.jar`
`BoundaryOnlyTrace.jar`
6. 응용 프로그램 서버를 시작합니다.

CA APM for SOA 가 Java Agent에서 제거됩니다.

.NET 에이전트에서 CA APM for SOA 를 제거하려면

1. IIS 응용 프로그램 서버를 중지합니다.
2. 명령 프롬프트를 열고 <Agent_Home>/ext 디렉터리로 이동합니다. 다음 파일을 제거합니다.
 - wily.WebServicesAgent.ext.dll
 - wilyHttpCorrelationTracers.ext.dll
 - wilyBoundaryOnlyTrace.ext.dll
3. IIS 응용 프로그램 서버를 시작합니다.

CA APM for SOA 가 .NET 에이전트에서 제거됩니다.

Enterprise Manager에서 CA APM for SOA 를 제거하려면

1. Enterprise Manager 를 중지합니다.
2. <EM_Home>/config/modules 디렉터리로 이동하고 다음 파일을 삭제합니다.
SPM_ManagementModule.jar.
3. <EM_Home>/ext/xmltv 디렉터리로 이동하고 다음 파일을 삭제합니다.
ws.overview.tv.xml
ws.wsdependency.tv.xml
ws.wsdeviation.tv.xml
4. <EM_Home>/product/enterprisemanager/plugins 디렉터리로 이동하고 다음 파일을 삭제합니다.
com.wily.introscope.soa.tracefilters.common_<version>.jar
com.wily.introscope.soa.tracefilters.em_<version>.jar
com.wily.introscope.soa.dependencymap.common_<version>.jar
com.wily.introscope.soa.dependencymap_<version>.jar
com.wily.introscope.soa.deviationmetrics_<version>.jar
5. <EM_Home>/ws-plugins 디렉터리로 이동하고 다음 파일을 삭제합니다.
com.wily.introscope.soa.crossprocessviewer.workstation_<version>.jar
com.wily.introscope.soa.dependencymap.common_<version>.jar
com.wily.introscope.soa.dependencymap.ui_<version>.jar
com.wily.introscope.soa.dependencymetrics.typeviewer_<version>.jar
com.wily.introscope.soa.tracefilters.common_<version>.jar
com.wily.introscope.soa.tracefilters.workstation_<version>.jar
com.wily.introscope.ui.tomsawyer_<version>.jar
6. Enterprise Manager 를 시작합니다.

CA APM for SOA 가 Enterprise Manager에서 제거됩니다.

제 3 장: SOA(Service-Oriented Architecture) 모니터링

CA APM for SOA 를 사용하면 웹 서비스 클라이언트 및 공급자와 프로세스 종속성, 중요 작업, 프로세스 흐름 및 SOAP 결합을 포함한 기본 작업에 대한 정보를 확인하여 SOA 인프라를 모니터링 할 수 있습니다.

이 단원에서는 SOA 인프라를 모니터링하기 위한 기본 대시보드와 메트릭 및 클라이언트 측과 서버 측 작업에 대한 SOA 특정 정보에 액세스하는 방법을 설명합니다.

이 섹션은 다음 항목을 포함하고 있습니다.

[CA Introscope®에서 SOA 특정 정보 보기](#) (페이지 47)

[SOA 성능 대시보드 사용](#) (페이지 51)

[Investigator 를 사용하여 SOA 성능 메트릭 보기](#) (페이지 58)

[Investigator 에서 경계 blame 보기](#) (페이지 69)

[기본 SOA 특정 메트릭 그룹화 보기](#) (페이지 69)

[기본 CA APM for SOA 경고 보기](#) (페이지 70)

CA Introscope®에서 SOA 특정 정보 보기

CA APM for SOA 를 설치한 후에는 Workstation 및 SOA 특정 대시보드, 탭, 메트릭을 사용하여 웹 서비스 클라이언트와 웹 서비스 공급자에 대한 정보를 확인함으로써 SOA 인프라를 모니터링 할 수 있습니다.

다음 목록은 이 제품이 SOA 환경을 모니터링하기 위해 Workstation 에 추가하는 항목에 대한 요약입니다.

- "SOA 성능" 대시보드는 배포된 에이전트에서 SOA 환경의 전반적인 건전성을 한눈에 모니터링 할 수 있게 해 줍니다.
- "SOA 종속성 맵"은 SOA 구성 요소 간의 종속성을 시각적으로 표현하며 웹 서비스 및 작업이 서로 관련된 방식을 이해하는 데 도움이 됩니다.
- "종속성" 탭에서는 특정 서비스가 사용하는 직접 또는 간접 중요 작업의 수와 같은 종속성 메트릭을 볼 수 있습니다.
- "편차" 탭에서는 특정 네임스페이스 또는 작업의 평균 응답 시간에 대한 편차와 같은 편차 메트릭을 볼 수 있습니다.

- "가장 중요한" 탭에서는 종속된 작업 수가 가장 많은 작업을 볼 수 있습니다.
- "가장 종속된" 탭에서는 다른 작업에 대한 종속 수가 가장 많은 작업을 볼 수 있습니다.
- "오류" 탭에서는 웹 서비스 작업에 의해 생성된 SOAP 결함에 대한 추가 정보를 제공합니다.

또한 **Investigator** 를 사용하여 표준 및 SOA 특정 메트릭을 드릴다운하고 "트랜잭션 추적 뷰어"를 사용하여 JVM 또는 CLR 경계를 가로지르는 SOA 관련 트랜잭션을 모니터링함으로써 SOA 특정 구성 요소에 대한 세부 정보를 볼 수 있습니다.

클라이언트와 서버 웹 서비스 및 작업 정보

CA APM for SOA 를 설치한 후 응용 프로그램 서버를 다시 시작하면 에이전트는 지시문 파일 목록에 정의된 지시문을 바탕으로 발견된 응용 프로그램을 계측합니다. 이 프로세스 중에 에이전트는 원격 프로시저 호출 또는 작업 호출에 사용된 SOAP 메시지에서 메서드 호출에 해당하는 웹 서비스 작업을 식별합니다. 각 웹 서비스 내의 개별 작업은 클라이언트(또는 아웃바운드) 요청이나 서버응답으로 식별되고 에이전트 응용 프로그램 서버의 "WebServices|클라이언트" 또는 "WebServices|서버" 노드에 배치됩니다.

대개의 경우 웹 서비스를 위한 클라이언트와 서버는 별도의 JVM(Java Virtual Machine) 또는 Microsoft CLR(Common Language Runtime) 환경에서 실행됩니다. 따라서 "클라이언트" 및 "서버" 노드는 별도의 에이전트 아래에 나열되는 경우가 많습니다. 같은 JVM 또는 CLR 이 요청과 응답을 모두 처리하는 경우에는 "클라이언트" 및 "서버" 노드 모두가 해당 에이전트의 WebServices 노드 아래에 나열됩니다. 예를 들어 클라이언트 및 서버가 같은 JVM 을 사용하는 경우에는 "클라이언트"와 "서버" 노드가 둘 다 **Investigator** 트리의 <hostname>|<process_name>|<agent_name>|WebServices 노드 아래에 표시됩니다.

서비스 네임스페이스 및 작업 이름 정보

Investigator에서 클라이언트 및 서버 노드를 확장하면 웹 서비스 네임스페이스를 사용하여 개별 웹 서비스가 표시됩니다. 웹 서비스 네임스페이스는 웹 서비스 URI(Uniform Resource Identifier)의 수정된 버전입니다. URI에는 두 가지 유형이 있습니다.

- URL(Uniform Resource Locator)
- URN(Uniform Resource Name)

Investigator에 표시되는 네임스페이스는 URL의 콜론(:)과 같이 예약된 의미를 가진 문자를 바꾸도록 수정되는 점을 제외하고는 웹 서비스 URL이나 URN과 같습니다. 예를 들어 일반적인 네임스페이스는 다음과 유사합니다.

`http://CreditManagement.demobank.ca.com`

웹 서비스 네임스페이스를 확장하면 해당 작업을 볼 수 있습니다. 웹 서비스에서 작업 이름은 메서드 이름과 마찬가지이므로 성능 저하 문제를 분석할 경우 응용 프로그램의 가장 낮은 수준까지 드릴다운할 수 있습니다. 예를 들어 `getSmallBusinessAcctBalance` 또는 `isAcctNumberValid`와 같이 일반적인 작업 이름은 수행될 세부 동작을 설명할 수 있습니다.

식별되지 않은 서비스 및 작업 정보

경우에 따라서는 CA APM for SOA에 이전트 확장이 웹 서비스 네임스페이스 또는 작업을 식별하지 못할 수 있습니다. 웹 서비스 이름 또는 작업을 식별하지 못하면 보고되는 메트릭 및 데이터 집계에 영향을 미칠 수 있습니다. 예를 들어 Investigator 트리에 네임스페이스 대신 `UnknownService`가 표시되거나 작업 이름 대신 `UnknownOperationName`이 표시될 수 있습니다.

WebServices 클라이언트 또는 서버에 대해 `UnknownService` 또는 `UnknownOperationName`이 표시되는 경우에는 다음을 검토하십시오.

- 사용 중인 프로토콜, API 및 응용 프로그램 서버 버전에 대한 웹 서비스의 구현
- 들어오는 메시지와 나가는 메시지의 구조

가상 에이전트 정보

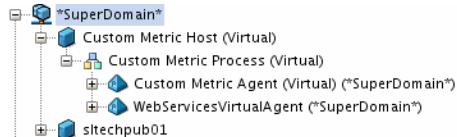
기본적으로 메트릭은 각 에이전트별로 독립적으로 표시됩니다. 여러 에이전트나 클라이언트 및 서버 네임스페이스에 대해 집계된 웹 서비스 메트릭을 보려면 가상 에이전트를 정의해야 합니다. 예를 들어 웹 서비스 클라이언트와 웹 서비스 서버가 모두 같은 JVM을 사용하는 경우, 웹 서비스 메트릭을 클라이언트와 서버가 모두 포함된 집계된 데이터로 보는 가상 에이전트를 정의하고자 할 수 있습니다.

에이전트가 보고를 하는 Enterprise Manager의 `<EM_Home>/config/agentclusters.xml` 파일을 수정하여 가상 에이전트를 정의할 수 있습니다. `agentclusters.xml` 파일 내에서 집계할 에이전트와 메트릭을 지정합니다. 예를 들어 가상 에이전트가 모든 서버 측 웹 서비스에서 "평균 응답 시간"에 대한 데이터를 수집하고 집계하거나 서버 측 및 클라이언트 웹 서비스의 "평균 응답 시간"을 결합하도록 구성할 수 있습니다.

다음 예제는 사용자 지정 에이전트가 아닌 모든 에이전트에서 서버 메트릭을 집계하는 `WebServicesVirtualAgent`라는 가상 에이전트를 정의하는 방법을 보여 줍니다.

```
<agent-cluster name="WebServicesVirtualAgent" domain="SuperDomain" >
  <agent-specifier>(.*)\|[^Custom.*](.*)\|(.*)</agent-specifier>
  <metric-specifier>WebServices\|Server\|(.*):Average Response Time
  \|(ms\|)</metric-specifier>
  <metric-specifier>WebServices\|Server\|(.*):Concurrent
  Invocations</metric-specifier>
  <metric-specifier>WebServices\|Server\|(.*):Errors Per
  Interval</metric-specifier>
  <metric-specifier>WebServices\|Server\|(.*):Responses Per
  Interval</metric-specifier>
  <metric-specifier>WebServices\|Server\|(.*):Stall Count</metric-specifier>
  <metric-specifier>WebServices\|Server\|(.*):SOAP Faults Per
  Interval</metric-specifier>
</agent-cluster>
```

그런 다음 **WebServicesVirtualAgent** 노드를 확장하여 집계된 서버 메트릭을 볼 수 있습니다. 다음 그림은 **Investigator** 트리에서 노드로 표시되는 **agentclusters.xml** 파일에 정의된 가상 에이전트를 보여 줍니다.



참고: 가상 에이전트 구성에 대한 자세한 내용은 **CA APM 구성 및 관리 안내서**를 참조하십시오.

SOA 성능 대시보드 사용

CA APM for SOA에는 SOA 환경의 전반적인 건전성을 모니터링하는 데 사용할 수 있는 사전 구성된 대시보드가 포함됩니다. 대시보드는 에이전트의 데이터를 집계하여 성능 및 가용성 정보를 요약하며 잠재적 문제에 대하여 상위 수준의 가시성을 제공합니다.

일반적으로 대시보드는 다음과 같은 작업을 가능하게 하므로 환경을 모니터링하는 시작점으로 사용됩니다.

- SOA 인프라의 주요 구성 요소에 대한 전반적인 건전성, 성능, 가용성 및 현재 상태를 한눈에 모니터링
- 하위 수준 메트릭이 주의 또는 위험 임계값이 초과되었음을 알리는 경우 프로덕션 응용 프로그램 환경의 잠재적 문제를 조기에 식별
- 성능 정보를 드릴다운하여 지역되고 있거나 오류가 발생 중인 웹 서비스 클라이언트, 공급자 또는 작업을 격리 및 식별

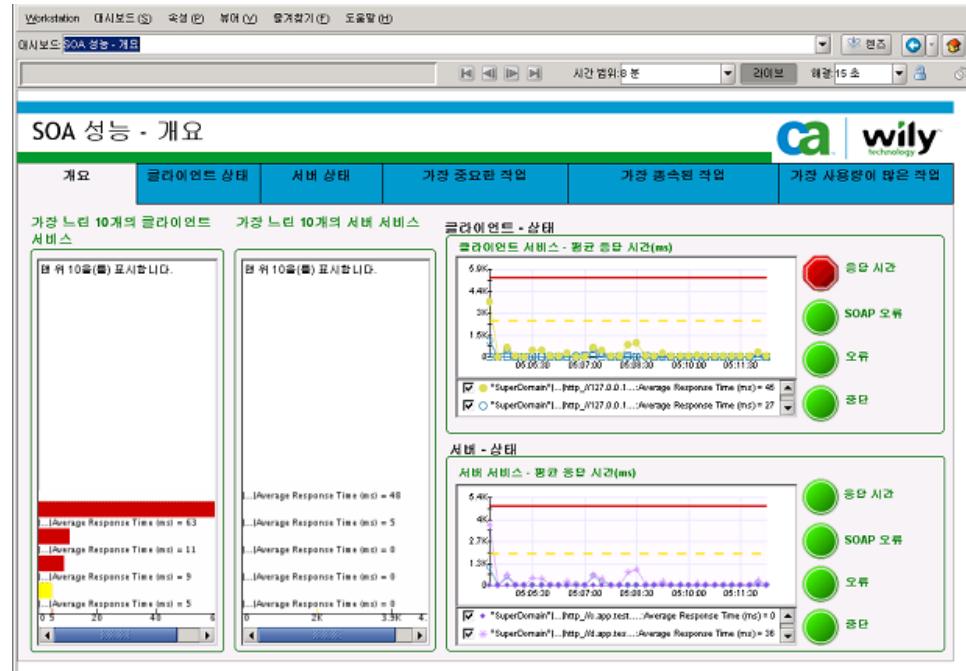
CA APM for SOA 용 Enterprise Manager 확장에는 **SPM_ManagementModule.jar** 파일에 패키지된 사전 구성된 SOA 특정 대시보드가 포함되어 있습니다.

다음 단계를 수행하십시오.

1. Enterprise Manager가 실행되고 있지 않으면 시작합니다.
2. Workstation을 시작하고 SOA가 설치된 Enterprise Manager에 로그인합니다.
3. "Workstation" > "새 콘솔"을 클릭합니다.

4. "대시보드" 드롭다운 목록에서 "SOA 성능" 대시보드 중 하나를 선택합니다.

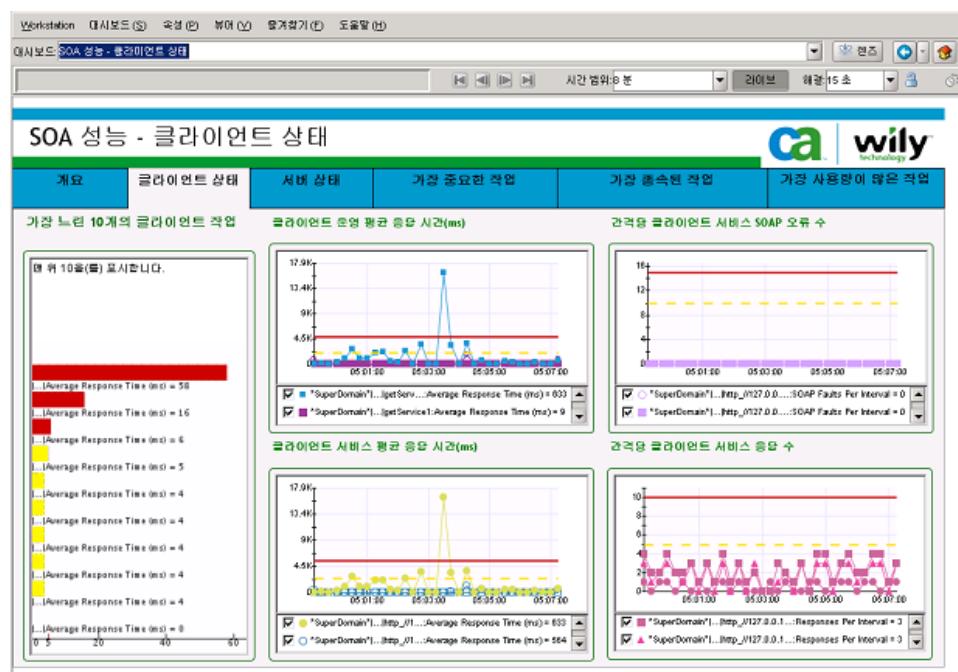
예를 들어 웹 서비스 모니터링을 위한 시작점으로 "SOA 성능 개요" 대시보드를 선택합니다. "SOA 성능 개요" 대시보드에는 배포된 서비스의 전반적인 건전성에 대한 정보가 표시됩니다.



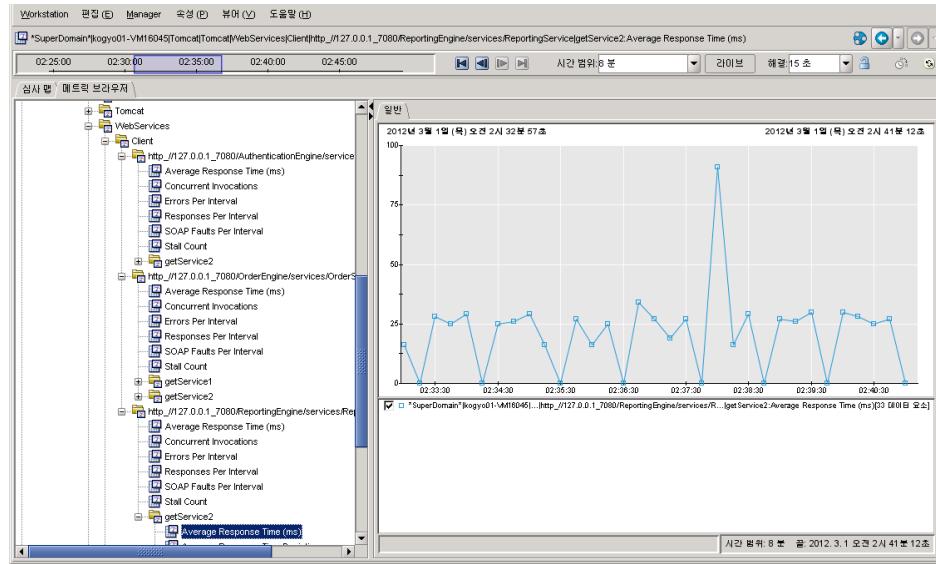
"SOA 성능 개요" 대시보드에서는 다음 대시보드를 탐색하여 클라이언트 또는 서버 작업에 대한 더 자세한 정보를 볼 수 있습니다.

- 클라이언트 건전성
- 서버 건전성
- 가장 중요한 작업
- 가장 종속된 작업
- 가장 사용량이 많은 작업

예를 들어 클라이언트 측의 평균 응답 시간이 문제인 것 같으면 "클라이언트 건전성" 탭을 클릭하여 가장 느린 클라이언트 작업, 평균 응답 시간, 간격당 응답 수 및 간격당 SOAP 결함 수에 대한 세부 정보가 있는 해당 대시보드를 확인할 수 있습니다.



- 모든 대시보드에서 서비스 또는 작업 이름을 두 번 클릭하면 Workstation Investigator 를 열고 더 자세히 분석할 수 있습니다. 예를 들어 "클라이언트 건전성" 대시보드에서 느린 서비스를 선택하면 Investigator 가 열리고 해당 클라이언트 서비스에 대한 관련 메트릭이 표시됩니다.



참고: Workstation 시작과 사용, 대시보드 액세스 또는 Investigator 열기와 탐색에 대한 자세한 내용은 **CA APM Workstation 사용자 안내서**를 참조하십시오.

추가 정보:

[Investigator 를 사용하여 SOA 성능 메트릭 보기](#) (페이지 58)

SOA 성능 개요 대시보드

"SOA 성능 - 개요" 대시보드는 모니터링되는 웹 서비스의 개요를 클라이언트 측 및 서버 측 서비스 작업으로 구분하여 제공합니다.

"개요" 대시보드에는 다음이 포함됩니다.

- 가장 느린 클라이언트 및 서버 서비스의 목록
- 클라이언트 및 서버 서비스의 평균 응답 시간 그래프
- 클라이언트 및 서버 서비스의 응답 시간, SOAP 결함, 오류 및 중단에 대한 경고 표시기

이 대시보드를 사용하면 현재 모니터링 중인 웹 서비스에 대한 요약된 정보를 볼 수 있습니다.

SOA 성능 클라이언트 건전성 대시보드

"SOA 성능 - 클라이언트 건전성" 대시보드는 클라이언트 측 관점에서 웹 서비스 작업에 대한 자세한 정보를 제공합니다.

"클라이언트 건전성" 대시보드에는 다음이 포함됩니다.

- 가장 느린 10 개의 클라이언트 작업 목록
- 각 클라이언트 측 작업에 대해 집계된 평균 응답 시간
- 각 클라이언트 측 웹 서비스에서 모든 작업에 대해 집계된 평균 응답 시간
- 간격당 각 클라이언트 측 웹 서비스에 대해 집계된 총 SOAP 결함 수
- 간격당 각 클라이언트 측 웹 서비스에 대해 집계된 총 응답 수

모니터링 중인 클라이언트 측 웹 서비스의 문제가 경고된 경우 이 대시보드를 사용하여 클라이언트 측 작업에 대한 더 자세한 정보를 볼 수 있습니다.

SOA 성능 서버 건전성 대시보드

"SOA 성능 - 서버 건전성" 대시보드는 서버 측 관점에서 웹 서비스 작업에 대한 자세한 정보를 제공합니다.

"서버 건전성" 대시보드에는 다음이 포함됩니다.

- 가장 느린 10 개의 웹 서비스 작업 목록
- 각 서버 측 작업의 평균 응답 시간
- 각 서버 측 웹 서비스에서 모든 메서드 또는 작업에 대해 집계된 평균 응답 시간
- 간격당 각 서버 측 웹 서비스에 대해 집계된 총 SOAP 결함 수
- 간격당 각 서버 측 웹 서비스에 대해 집계된 총 응답 수

모니터링 중인 서버 측 웹 서비스의 문제가 경고된 경우 이 대시보드를 사용하여 서버 측 작업에 대한 더 자세한 정보를 볼 수 있습니다.

SOA 성능 - 가장 중요한 작업 대시보드

"SOA 성능 - 가장 중요한 작업" 대시보드는 종속된 작업 수가 가장 많은 작업의 편차 메트릭을 제공합니다. 중요한 작업은 다른 작업의 성공 또는 실패 여부의 영향을 받을 가능성이 가장 크기 때문에 이 작업을 면밀히 모니터링해야 합니다.

"가장 중요한 작업" 대시보드에는 종속된 작업 수가 가장 많은 작업의 목록과 다음에 대한 그래프가 포함됩니다.

- 평균 응답 시간 편차
- 간격당 오류 편차
- 간격당 응답 편차

편차 값이 높으면 중요 작업이 안정적으로 수행되고 있지 않거나 다운스트림 작업이 제대로 실행 중이지 않음을 나타낼 수 있습니다.

CA APM for TIBCO BusinessWorks 와 같은 추가 SOA 플랫폼에 대한 모니터링을 사용하도록 설정한 경우 이 대시보드에는 확장의 가장 중요한 프로세스에 대한 정보가 함께 포함됩니다.

SOA 성능 - 가장 종속된 작업 대시보드

"SOA 성능 - 가장 종속된 작업" 대시보드는 다른 작업에 대한 종속 수가 가장 많은 작업의 편차 메트릭을 제공합니다.

"가장 종속된 작업" 대시보드에는 다른 작업에 대한 종속이 가장 많은 작업의 목록과 다음에 대한 그래프가 포함됩니다.

- 평균 응답 시간 편차
- 간격당 오류 편차
- 간격당 응답 편차

편차 값이 높으면 업스트림 작업이 안정적으로 수행되고 있지 않다는 의미일 수 있습니다.

CA APM for TIBCO BusinessWorks 와 같은 추가 SOA 플랫폼에 대한 모니터링을 사용하도록 설정한 경우 이 대시보드에는 확장의 가장 종속된 프로세스에 대한 정보도 포함됩니다.

SOA 성능 - 가장 사용량이 많은 작업 대시보드

"SOA 성능 - 가장 사용량이 많은 작업" 대시보드는 클라이언트 측 및 서버 측에서 경험하는 간격당 최고 웹 서비스 호출 비율에 대한 세부 정보를 제공합니다.

"가장 사용량이 많은 작업" 대시보드에는 다음이 포함됩니다.

- 가장 사용량이 많은 클라이언트 및 서비스 작업의 목록
- 모든 클라이언트 측 작업에 대한 간격당 총 응답 수
- 모든 서버 측 작업에 대한 간격당 총 응답 수

이 대시보드를 사용하여 현재 모니터링 중인 클라이언트 및 서버 웹 서비스의 현재 처리량에 대한 더 상세한 정보를 볼 수 있습니다.

Investigator 를 사용하여 SOA 성능 메트릭 보기

대시보드는 클라이언트 및 서버 상태에 대한 개요와 가장 중요한 작업 또는 가장 사용량이 많은 작업 같이 오류 발생 가능성이 높은 지점을 제공하는데 비해, **Investigator** 는 SOA 환경의 세부 사항으로 드릴다운하여 문제의 근본적인 원인을 파악할 수 있도록 합니다.

Investigator 를 사용하면 응용 프로그램의 최상위 수준부터 개별 서비스 내의 개별 작업에 대한 세부적인 메트릭에 이르기까지 집계된 메트릭을 분석하여 성능 저하나 오류가 발생한 동작의 원인을 찾을 수 있습니다.

사용할 수 있는 메트릭

CA APM for SOA 는 모니터링되는 각 작업 또는 서비스 네임스페이스에 대한 표준 CA Introscope 메트릭 외에도 몇 가지 SOA 특정 메트릭을 제공합니다. 개별 작업에 대한 메트릭은 개별 웹 서비스에 대한 메트릭으로 집계되며 응용 프로그램의 클라이언트 측 및 서버 측 성능을 평가하는 데 사용할 수 있습니다.

웹 서비스로 구현되는 응용 프로그램에는 일반적으로 다음과 같은 표준 및 SOA 특정 메트릭을 사용할 수 있습니다.

- **평균 응답 시간(ms)** - 호출된 메서드, 작업 또는 서비스가 15 초의 기간 내에 완료되는 데 걸리는 평균 시간(밀리초)입니다.
- **평균 응답 시간 편차** - 간격당 평균 응답 시간의 평균에 대한 편차입니다.
- **동시 호출** - 진행 중이고 15 초의 기간이 끝날 때 완료되지 않은 동시 요청 수입니다.
- **간격당 오류 수** - 15 초의 기간 동안 생성된 예외, HTTP 오류 또는 SOAP 결함의 수입니다.
- **간격당 오류 편차** - 간격 내에서 발생한 예외, HTTP 오류 또는 SOAP 결함 수의 평균에 대한 편차입니다.
- **간격당 응답 수** - 15 초의 기간 동안 완료된 작업 요청 또는 서비스 요청의 수입니다.
- **간격당 응답 편차** - 간격 내에 완료된 호출의 평균에 대한 편차입니다.
- **중단 수** - 30 초의 기간이 끝날 때 완료되지 않은 요청 수입니다.
- **간격당 SOAP 결함 수** - 15 초의 기간 동안 SOAP 엔진에 의해 생성되거나 사용된 SOAP 메시지 오류의 수입니다.

- **중요 직접** - 성공적으로 실행 중인 현재 작업에 직접적으로 의존하는 다운스트림 작업의 수입니다.
- **중요 간접** - 성공적으로 실행 중인 현재 작업에 직접적으로 또는 간접적으로 의존하는 다운스트림 작업의 수입니다.
- **종속 직접** - 현재 작업이 직접적으로 종속된 업스트림 작업의 수입니다.
- **종속 간접** - 현재 작업이 직접적으로 또는 간접적으로 종속된 업스트림 작업의 수입니다.

"개요" 템에서 요약된 메트릭 보기

일반적으로 Investigator 를 사용하여 문제의 근본 원인을 찾는 데 가장 좋은 시작점은 "개요" 템입니다. "개요" 템에 표시되는 정보는 선택한 Investigator 노드에 따라 다릅니다. 예를 들어 에이전트 노드를 선택한 경우 "개요" 템에는 에이전트에서 계측된 모든 응용 프로그램의 상위 수준 건전성 표시기가 표시됩니다. 이 "개요" 템은 Workstation에서 기본적으로 제공됩니다. CA APM for SOA에서는 다음에 대하여 추가 "개요" 템을 제공합니다.

- **WebServices 노드** - WebServices 노드를 선택하면 "개요" 템에는 선택된 에이전트 응용 프로그램과 연관된 클라이언트 또는 서버 웹 서비스 네임스페이스가 모두 표시됩니다. 에이전트에 클라이언트 및 서버 웹 서비스가 모두 있는 경우 "개요" 템에는 클라이언트 및 서버 네임스페이스가 별도의 템에 표시됩니다.
- **클라이언트 노드** - "클라이언트" 노드를 선택하면 "개요" 템에는 선택된 에이전트 응용 프로그램의 클라이언트 측 작업과 연관된 웹 서비스 네임스페이스가 나열됩니다.
- **서버 노드** - "서버" 노드를 선택하면 "개요" 템에는 선택된 에이전트 응용 프로그램의 서버 측 작업과 연관된 웹 서비스 네임스페이스가 나열됩니다.
- **<web_service_namespace> 노드** - 개별 웹 서비스 네임스페이스를 선택하면 "개요" 템에는 선택된 클라이언트 또는 선택된 <web_service_namespace>로 식별되는 서버 웹 서비스에 속하는 작업이 나열됩니다.
- **<operation_name> 노드** - 개별 작업을 선택하면 "개요" 템에는 선택된 <operation_name>으로 식별된 작업에 대한 "간격당 SOAP 결합 수"와 표준 CA Introscope® 메트릭을 사용하는 현재 동작 수준을 보여 주는 그래프가 표시됩니다.

다음 단계를 수행하십시오.

1. Investigator 를 열고 에이전트 노드를 클릭한 다음 "뷰어" 창에서 "개요" 탭을 선택합니다.
"개요"에는 선택된 에이전트에서 계측된 각 웹 서비스에 대한 경고 표시기가 표시됩니다.
2. 에이전트 노드를 확장하고 Investigator 트리에서 WebServices 를 선택합니다. "개요" 탭을 클릭합니다.
클라이언트 및 서버 웹 서비스 메트릭에 대한 별도의 테이블이 나타납니다. 다음 옵션을 사용할 수 있습니다.
 - "클라이언트" 또는 "서버" 테이블에서 특정 서비스 네임스페이스를 선택합니다.
해당 서비스에 대한 "평균 응답 시간" 그래프가 나타납니다.
 - 표시되는 그래프를 변경하려면 표시할 메트릭에 대한 메트릭 열 머리글을 클릭합니다. 예를 들어 "간격당 오류 수"에 대한 그래프를 표시하려면 "간격당 오류 수" 메트릭 열 머리글을 클릭하십시오.
3. WebServices 노드를 확장하고 Investigator 트리에서 "클라이언트" 또는 "서버"를 선택합니다. "개요" 탭을 클릭합니다.
클라이언트 네임스페이스 목록 또는 서버 네임스페이스 목록만 나타납니다.
4. Investigator 트리에서 "클라이언트" 또는 "서버" 노드를 확장하고 특정 <web_service_namespace>를 선택합니다. "개요" 탭을 클릭합니다.
선택한 웹 서비스 네임스페이스에 대한 작업 및 관련 메트릭이 나타납니다.
5. <web_service_namespace> 노드를 확장하고 Investigator 트리에서 <operation_name>을 선택합니다. "개요" 탭을 클릭합니다.
해당 작업에 대한 메트릭의 그래프가 나타납니다.

"종속성" 탭에서 종속성 메트릭 보기

CA APM for SOA 는 표준 CA Introscope 메트릭 이외에도 웹 서비스 트랜잭션에서 가장 중요한 작업과 가장 종속된 작업을 결정하는 메트릭을 제공합니다. 이 메트릭은 각 작업에 대한 직접 및 간접 종속성의 수를 기반으로 합니다.

이러한 종속성 기반 메트릭을 "종속성" 탭에서 보려면 적절한 Investigator 노드를 선택하십시오.

"종속성" 탭에 표시되는 정보는 선택한 Investigator 노드에 따라 달라집니다. 예를 들면 다음과 같습니다.

- **WebServices** 노드를 선택하면 "종속성" 탭에는 선택된 에이전트 응용 프로그램과 연관된 클라이언트 및 서버 웹 서비스 네임스페이스, 작업 및 종속성 메트릭이 표시됩니다.
- **Client** 노드를 선택하면 "종속성" 탭에는 선택된 에이전트 응용 프로그램의 클라이언트 측 작업과 연관된 웹 서비스 네임스페이스, 작업 및 종속성 메트릭이 표시됩니다.
- **Server** 노드를 선택하면 "종속성" 탭에는 선택된 에이전트 응용 프로그램의 서버 측 작업과 연관된 웹 서비스 네임스페이스, 작업 및 종속성 메트릭이 표시됩니다.
- 개별 **<web_service_namespace>** 노드를 선택하면 "종속성" 탭에는 선택된 클라이언트 또는 선택된 **<web_service_namespace>**로 식별되는 서버 웹 서비스와 연관된 작업 및 종속성 메트릭이 표시됩니다.

직접 및 간접 작업 종속성 정보

웹 서비스 트랜잭션에서 특정 작업이 다른 작업을 호출하면 종속성 체인이 생성되며, 호출된 작업은 호출한 작업에 종속됩니다. 호출된 작업으로부터의 추가 호출이나 응답에 의해 추가적인 종속성이 생성됩니다. 작업 간의 종속성은 종속성 체인에서 작업의 위치에 따라 직접종속성 또는 간접종속성이 됩니다.

예를 들어 작업 A 가 작업 B 를 호출하고, 작업 B 가 작업 C 를 호출할 경우의 트랜잭션은 다음과 같습니다.

A --> B --> C

이 트랜잭션의 종속성 관계는 다음과 같습니다.

- 작업 A 와 B 는 직접종속성 관계
- 작업 B 와 C 는 직접종속성 관계
- 작업 A 와 C 는 간접종속성 관계

작업은 종속성 체인에서의 위치에 따라 다른 작업의 업스트림 또는 다운스트림으로 구분됩니다. 여러 작업으로 구성된 트랜잭션에서 직접 종속성은 서로 직접적인 업스트림 또는 다운스트림인 경우입니다. 위의 예제에서 작업 A 는 작업 B 와 C 모두의 업스트림이지만 작업 B 에서만 직접적인 업스트림입니다.

마찬가지로, 간접 종속성은 직접 호출되지 않은 작업의 업스트림 또는 다운스트림으로 발생할 수 있습니다. 예를 들어 작업 A 가 작업 B 를 호출한 후 작업 B 가 작업 C 와 D 를 호출할 경우, 작업 C 와 D 는 모두 작업 A 에 대해 다운스트림 간접 종속성 관계를 형성합니다.

중요한 작업과 메트릭 정보

웹 서비스의 작업은 다른 작업과의 상호 작용에 따라 중요한 작업이나 종속된 작업 또는 중요하고 종속된 작업으로 간주됩니다.

중요한 작업은 다운스트림 작업이 의존하는 작업입니다. 다운스트림 작업은 중요한 작업에 대해 직접 또는 간접 종속성을 가질 수 있지만, 중요한 작업이 성공적으로 실행되어야만 다운스트림 작업이 실행될 수 있습니다.

중요한 작업에는 다음과 같은 두 가지 메트릭이 관련됩니다.

- "중요 직접" 메트릭은 특정 작업의 직접 업스트림인 작업의 수만 추적합니다. 작업이 업스트림 작업에 종속되지 않은 경우, 즉 작업이 다른 작업의 다운스트림이 아닌 경우에는 작업의 "중요 직접" 메트릭이 0 입니다.
- "중요 간접" 메트릭은 작업의 직접 및 간접 업스트림 작업의 수를 추적합니다.

예를 들어 *getAccountNum* 작업의 "중요 간접" 메트릭 값이 5 라면 *getAccountNum* 작업이 먼저 실행되어야만 실행될 수 있는 직접 또는 간접 작업이 5 개 있는 것입니다.

종속 작업 및 메트릭 정보

종속 작업은 업스트림 작업의 실행에 의존하는 작업입니다. 종속 작업은 업스트림 작업에 대해 직접 또는 간접 종속성을 가질 수 있지만, 종속 작업이 실행되려면 업스트림 작업이 성공적으로 실행되어야 합니다.

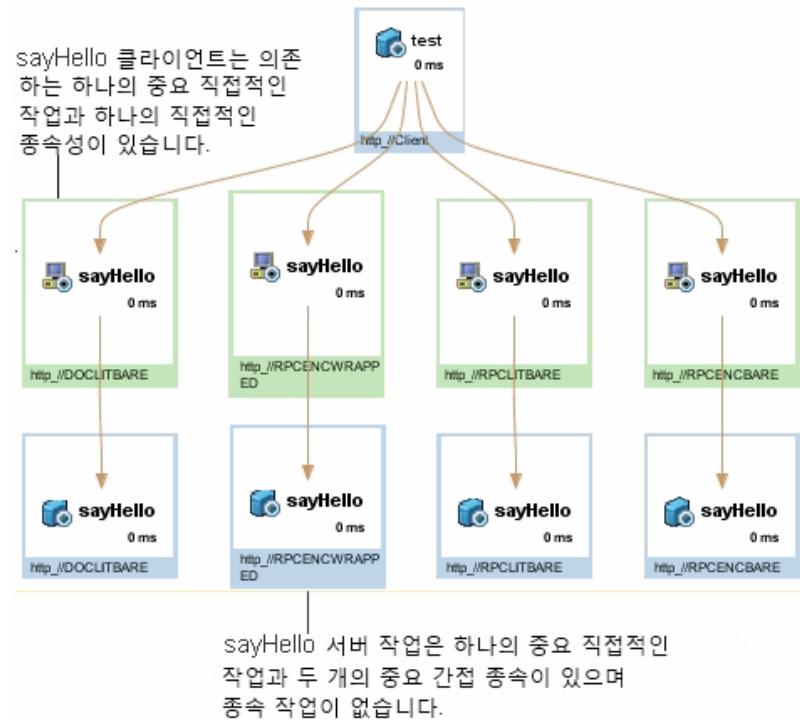
중요한 작업에는 다음과 같은 두 가지 메트릭이 관련됩니다.

- "종속성 직접" 메트릭은 작업의 직접 다운스트림 작업 수를 추적합니다. 작업에 종속된 작업이 없는 경우, 즉 이 작업이 다른 작업의 업스트림이 아닌 경우에는 "종속성 직접" 메트릭이 0 입니다.
- "종속성 간접" 메트릭은 작업의 직접 및 간접 다운스트림인 작업의 수를 추적합니다.

예를 들어 *sendConfirmation* 작업의 "종속성 간접" 메트릭 값이 8 이라면 *sendConfirmation* 작업을 실행하기 위해 먼저 실행되어야 하는 직접 또는 간접 작업이 8 개 있는 것입니다.

중요 메트릭 및 종속성 메트릭 자세히 살펴보기

"중요 직접/간접" 및 '종속성 직접/간접" 메트릭이 어떻게 작동하는지 이해하기 위해 *sayHello* 클라이언트 작업 4 개를 호출하는 *Test* 서버 작업을 예로 들어 보겠습니다. 이 경우 각 클라이언트 작업은 아래의 샘플 종속성 맵에서 볼 수 있듯이 다운스트림 *sayHello* 서버 작업을 하나씩 호출합니다.



이 예제의 경우 다음과 같은 중요 메트릭 값과 종속성 메트릭 값이 적용됩니다.

Test 서버

업스트림 작업이 없기 때문에 중요한 작업의 메트릭 값은 "중요 직접" = 0 이고 "중요 간접" = 0 입니다.

직접 다운스트림 작업 4 개와 추가적인 간접 다운스트림 작업 4 개가 있으므로 종속 작업의 메트릭 값은 "종속성 직접" = 4 이고 "종속성 간접" = 8 입니다.

sayHello 클라이언트

직접 업스트림 작업이 하나 있으므로 중요한 작업의 메트릭 값은 "중요 직접" = 1 이고 "중요 간접" = 1 입니다.

직접 다운스트림 작업이 하나 있으므로 종속 작업의 메트릭 값은 "종속성 직접" = 1 이고 "종속성 간접" = 1 입니다.

sayHello 서버

sayHello 에 대한 직접 업스트림 종속성과 Test 서버에 대한 간접 종속성이 존재하므로 중요한 작업의 메트릭 값은 "중요 직접" = 1 이고 "중요 간접" = 2 입니다.

직접 또는 간접 다운스트림 작업이 없으므로 종속 작업의 메트릭 값은 "종속성 직접" = 0 이고 "종속성 간접" = 0 입니다.

종속성 탭에서 종속성 메트릭 보기

1. 에이전트 노드를 확장하고 Investigator 트리에서 WebServices 를 선택한 다음 "뷰어" 창에서 "종속성" 탭을 클릭하여 클라이언트 및 서버 종속성 메트릭에 대한 개별 테이블을 봅니다.
각 작업마다 "종속성" 탭에 "중요 직접", "중요 간접", "종속성 직접" 및 "종속성 간접" 메트릭 값이 표시됩니다.
2. WebServices 노드를 확장하고 Investigator 트리에서 클라이언트 또는 서버 노드를 선택한 다음 "뷰어" 창에서 "종속성" 탭을 클릭하면 클라이언트 네임스페이스 또는 서버 네임스페이스 및 관련 종속성 메트릭만 표시됩니다.
3. 클라이언트 또는 서버 노드를 확장하고 Investigator 트리에서 특정 `<web_service_namespace>`를 선택한 후 "뷰어" 창에서 "종속성" 탭을 클릭하여 작업 목록 및 선택한 웹 서비스 네임스페이스와 관련된 종속성 메트릭을 표시합니다.

"편차" 탭에서 편차 메트릭 보기

CA APM for SOA 는 "평균 응답 시간", "간격당 오류 수" 및 "간격당 응답 수"의 평균 값에 대한 편차를 모니터링하는 메트릭을 제공합니다. 편차 값이 낮으면 "평균 응답 시간"과 같은 메트릭 데이터가 평균 주위에 밀집되어 있다는 의미입니다. 편차 값이 높으면 메트릭 데이터가 평균보다 높거나 낮은 값으로 넓게 분산되어 있다는 의미입니다.

참고: 간격 내에 호출되지 않은 작업의 "평균 응답 시간" 메트릭 값은 0입니다. 값이 0인 메트릭은 평균 응답 시간 계산에 포함되지 않습니다. 따라서 가끔 "편차" 탭에 모니터링이 중지된 것으로 나타나는 경우가 있습니다. 이와 같이 보이는 일시 중지는 정상 동작입니다.

편차 메트릭을 보려면 **Investigator** 노드를 선택하고 "편차" 탭을 클릭하십시오. 예:

- "WebServices" 노드에는 선택한 애이전트 응용 프로그램과 연관된 클라이언트 및 서버 웹 서비스 네임스페이스, 작업 및 편차 메트릭이 표시됩니다.
- "클라이언트" 노드에는 클라이언트와 연관된 웹 서비스 네임스페이스, 작업 및 편차 메트릭이 표시됩니다.
- "서버" 노드에는 서버와 연관된 웹 서비스 네임스페이스, 작업 및 편차 메트릭이 표시됩니다.
- 개별 <web_service_namespace> 노드에는 선택한 클라이언트 또는 서버 웹 서비스와 연관된 작업 및 편차 메트릭이 표시됩니다.

참고: 기본적으로 편차는 네임스페이스 수준에 대해서만 계산됩니다. "WebServices", "클라이언트" 및 "서버" 노드에 대해 편차가 사용되도록 설정하려면 추가 메트릭을 계산하도록 [com.wily.introscope.soa.deviation.metric.expressionlist](#) (페이지 394) 속성을 구성하십시오.

편차 메트릭 계산

편차 메트릭을 사용하려면 CA APM for SOA 가 평균 값을 계산해야 합니다. 이 제품은 "n"일 동안의 작업 메트릭에 기반하여 평균 값을 결정합니다. 연속 평균이 계산되는 기본 기간(일)은 7일입니다. 이 기간(일)을 조정하려면 `com.wily.introscope.soa.deviation.mean.days` 속성 값을 설정하십시오.

CA APM for SOA 는 Enterprise Manager 에 미치는 성능 영향을 최소화하기 위해 편차 메트릭을 계산할 수 있는 작업의 수를 제한합니다. 기본적으로 편차 메트릭은 각 종속성 메트릭 유형에서 상위 25 개 작업에 대해 계산됩니다. 편차가 계산되는 최대 작업 수를 변경하려면 Enterprise Manager 속성 파일의 `com.wily.introscope.soa.deviation.count.per.metric` 속성을 수정하십시오.

편차 메트릭 생성에 대한 성능 오버헤드를 줄이기 위해 특정 편차 메트릭 유형을 선택적으로 사용하거나 사용하지 않도록 설정할 수 있습니다. 예를 들어 "평균 응답 시간 편차" 메트릭만 생성하도록 구성 속성을 설정할 수 있습니다. 또는 `com.wily.introscope.soa.deviation.enable` 속성을 `false` 로 설정하면 편차 메트릭이 계산되지 않습니다.

추가 정보:

[Enterprise Manager 속성 구성](#) (페이지 375)

"가장 중요한" 탭에서 중요 작업 메트릭 보기

SOA 환경에서 가장 중요한 작업은 종속된 작업의 수가 가장 많은 작업입니다. 이러한 작업은 가장 중요한 잠재적 실패 지점이며 Investigator 의 "가장 중요한 작업" 대시보드 및 "가장 중요한" 탭에서 세밀하게 모니터링해야 합니다. 가장 중요한 작업 및 관련 편차 메트릭의 목록은 Investigator 에서 WebServices 노드를 선택했을 때 "가장 중요한" 탭에서 볼 수 있습니다.

다음 단계를 수행하십시오.

1. 에이전트 노드를 확장하고 Investigator 트리에서 WebServices 를 선택한 다음 "뷰어" 창에서 "가장 중요한" 탭을 클릭합니다.
2. 중요한 작업의 목록에서 개별 작업을 선택하여 해당 작업에 대한 편차 메트릭의 그래프를 봅니다.

"가장 종속된" 탭에서 종속된 작업 보기

SOA 환경에서 가장 종속된 작업은 다른 작업에 대한 종속 수가 가장 많은 작업입니다. 가장 종속된 작업 및 관련 편차 메트릭의 목록은 Investigator 에서 WebServices 노드를 선택했을 때 "가장 종속된" 탭에서 볼 수 있습니다.

다음 단계를 수행하십시오.

1. 에이전트 노드를 확장하고 Investigator 트리에서 WebServices 를 선택한 다음 "뷰어" 창에서 "가장 종속된" 탭을 클릭합니다.
2. 종속된 작업의 목록에서 개별 작업을 선택하여 해당 작업에 대한 편차 메트릭의 그래프를 봅니다.

"오류" 탭에서 SOAP 결함 및 오류 메트릭 보기

Investigator 는 Investigator 트리에서 선택하는 노드에 대한 오류 메시지 및 세부 정보가 표시되는 기본 "오류" 탭을 제공합니다. CA APM for SOA 확장이 설치된 경우에는 "오류" 탭에 클라이언트 측 및 서버 측 SOAP 결함에 대한 정보도 표시됩니다.

"간격당 오류 수" 메트릭은 에이전트에 의해 생성되는 표준 CA Introscope 메트릭입니다. "간격당 SOAP 결함 수" 메트릭은 CA APM for SOA 에이전트 확장에 의해 생성되는 SOA 특정 메트릭입니다. 이들 메트릭 사이에는 직접적인 상관 관계가 없습니다.

SOAP 결함은 웹 서비스가 정상 응답 대신 결함 응답을 반환할 때 SOAP 엔진에서 생성하거나 사용하는 메시지입니다. SOAP 결함은 조건에 대한 응답으로 자동으로 생성되거나 응용 프로그램의 논리에 명시적으로 코딩될 수 있습니다. 결함을 캡처하고 처리하는 방법은 사용 중인 SOAP 엔진에 따라 다릅니다.

"간격당 오류 수" 메트릭은 `throw` 된 예외 및 HTTP 오류를 추적합니다. 웹 서비스의 비즈니스 논리와 응용 프로그램 서버의 동작에 따라 SOAP 결함에 해당하는 오류가 "간격당 오류 수" 메트릭에 존재할 수 있습니다.

응용 프로그램 서버가 문제를 발견하여 SOAP 결함을 생성하면 클라이언트 및 서버 "간격당 SOAP 결함 수" 메트릭이 모두 결함을 기록합니다. 응용 프로그램 서버가 SOAP 결함에 관련된 예외를 `throw` 하면 해당 "간격당 오류 수"도 업데이트됩니다. 클라이언트에 SOAP 결함을 예외로 `catch` 할 코드가 없는 경우 클라이언트의 "간격당 오류 수" 메트릭이 업데이트되지 않습니다.

"간격당 오류 수" 메트릭은 CA APM for SOA 계측된 응용 프로그램 서버 또는 SOAP 스택 메서드에서 `catch` 된 예외 및 오류 코드만 추적하므로 메트릭에 SOAP 결함이 포함되지 않을 수 있습니다.

다음 단계를 수행하십시오.

1. 에이전트 노드를 확장하고 Investigator 트리에서 WebServices 를 선택하거나 WebServices 노드를 확장한 다음 "클라이언트" 또는 "서버" 노드를 선택합니다.
2. "뷰어" 창에서 "오류" 탭을 마우스 오른쪽 단추로 클릭합니다.
"오류" 탭의 위쪽 창에 각 오류의 시간, 설명 및 오류 메시지가 나열됩니다. SOAP 엔진에서 생성된 오류 메시지의 경우 오류 메시지 시작 부분에 SOAP 결함이 표시됩니다.
3. 개별 오류를 선택하면 해당 오류에 대한 세부 정보가 표시됩니다.
그런 후 SOAP 결함에 대한 세부 정보가 "스택 뷰" 탭에 표시되며 여기에서 메시지 텍스트는 빨간색으로 강조 표시됩니다.

Investigator에서 경계 blame 보기

경계 blame 은 응답 시간 문제가 느린 서블릿 속도와 같은 응용 프로그램 서버 내부의 문제인지 아니면 데이터베이스 서버와 같은 백엔드 시스템으로의 외부 호출 때문인지 확인할 수 있게 Investigator 트리에서 프런트엔드 및 백엔드 구성 요소만 포함하도록 메트릭을 격리합니다. CA APM for SOA 는 백엔드 시스템에 대한 웹 서비스 호출을 자동으로 감지하여 이 기능을 SOA 구성 요소로 확장합니다.

프런트엔드 및 백엔드 구성 요소에 대한 메트릭을 격리하기 위한 이 확장 기능은 기본적으로 활성화됩니다.

참고: 응용 프로그램에 대한 경계 blame 의 구성, 비활성화 또는 사용자 지정에 대한 자세한 내용은 *CA APM Java Agent* 구현 안내서, *CA APM .NET* 에이전트 구현 안내서 및 *CA APM Workstation* 사용자 안내서를 참조하십시오.

기본 SOA 특정 메트릭 그룹화 보기

CA APM for SOA 는 기본 대시보드와 경고를 정의하는 데 사용되는 기본 메트릭 그룹화를 포함합니다. 기본 메트릭 그룹화를 사용자 지정 대시보드 또는 경고에서도 사용할 수 있습니다. 기본 메트릭 그룹화는 CA APM for SOA 용 Enterprise Manager 확장에 포함되며 SPM_ManagementModule.jar 파일에 패키지됩니다. Workstation 관리 모듈 편집기를 사용하여 기본 메트릭 그룹화를 볼 수 있습니다.

다음 단계를 수행하십시오.

1. Investigator에서 Workstation > 새 관리 모듈 편집기를 클릭합니다.
2. "*SuperDomain*" > "관리 모듈" > "SOA 성능 관리(*SuperDomain*)"를 확장합니다.
3. "메트릭 그룹화" 노드를 확장하여 SOA 성능 관리 모듈에 대해 정의된 모든 메트릭 그룹화를 표시합니다.
4. 특정 메트릭 그룹화를 클릭하여 "뷰어" 창에서 해당 정의를 봅니다.

필요에 따라 모든 메트릭 그룹화의 기본 설정을 수정하거나 사용자 지정 메트릭 그룹화를 직접 생성할 수 있습니다.

참고: 메트릭 그룹화 생성과 사용에 대한 자세한 내용은 *CA APM 구성 및 관리 안내서*를 참조하십시오.

기본 CA APM for SOA 경고 보기

CA APM for SOA에는 사전 구성된 대시보드에서 사용되는 기본 경고 정의가 포함됩니다. 기본 경고 정의는 CA APM for SOA Enterprise Manager 확장에 포함되며 SPM_ManagementModule.jar 파일에 패키지됩니다.

Workstation 관리 모듈 편집기를 사용하여 기본 경고 정의를 볼 수 있습니다. 또한 CA APM for SOA 관리 모듈을 확장하여 사용자 지정 경고 정의와 알림 유형을 포함하거나 사용자 지정 대시보드에서 기본 경고 정의를 사용할 수도 있습니다.

다음 단계를 수행하십시오.

1. Investigator에서 Workstation > 새 관리 모듈 편집기를 클릭합니다.
2. "*SuperDomain*" > "관리 모듈" > "SOA 성능 관리(*SuperDomain*)"를 확장합니다.
3. "경고" 노드를 확장하여 CA APM for SOA에 대해 정의된 모든 경고를 표시합니다.
4. 특정 경고를 클릭하여 "뷰어" 창에서 해당 정의를 봅니다.

특히 "주의" 및 "위험" 임계값의 기본 설정을 검토하고 필요한 경우 값을 조정합니다. 알림 또는 수정 조치를 추가할 수도 있습니다.

다음 표에서는 사전 구성된 CA APM for SOA 경고 정의와 연관된 기본 "주의" 및 "위험" 임계값을 요약합니다.

경고	주의	위험
클라이언트 작업 평균 응답 시간	2000 ms	5000 ms
서버 작업 평균 응답 시간		
서버 서비스 평균 응답 시간		
클라이언트 서비스 평균 응답 시간	2500 ms	5500 ms
간격당 클라이언트 작업 오류 수	10 개 오류	15 개 오류
간격당 클라이언트 서비스 오류 수		
간격당 서버 작업 오류 수		
간격당 서버 서비스 오류 수		
간격당 클라이언트 작업 응답 수	5 개 응답	10 개 응답
간격당 클라이언트 서비스 응답 수		
간격당 서버 작업 응답 수		
간격당 서버 서비스 응답 수		
간격당 클라이언트 작업 SOAP 결함 수	10 개 결함	15 개 결함
간격당 클라이언트 서비스 SOAP 결함 수		
간격당 서버 작업 SOAP 결함 수		
간격당 서버 서비스 SOAP 결함 수		
클라이언트 작업 중단 수	1 개 중단	2 개 중단
클라이언트 서비스 중단 수		
서버 작업 중단 수		
서버 서비스 중단 수		

CA APM for SOA 로 에이전트에 대한 사용자 지정 경고를 생성하려면

1. 관리 모듈 편집기에서 "요소" > "경고 새로 만들기"를 클릭합니다.
2. "이름" 필드에 경고의 이름을 입력합니다.
3. "관리 콘솔" 드롭다운 목록에서 SOA Performance Management 를 선택하고 "확인"을 클릭합니다.

참고: 경고 생성과 경고 알림 사용자 지정에 대한 자세한 내용은 *CA APM 구성 및 관리 안내서*를 참조하십시오.

제 4 장: SOA 종속성 맵 사용

SOA 종속성 맵은 SOA 환경에 배포되어 있는 웹 서비스나 비즈니스 프로세스 작업 흐름을 시각적으로 보여 줍니다. 이 맵을 사용하여 구성 요소 간의 실시간 종속성을 보거나 탐색하고, 관련된 구성 요소 간의 주요 메트릭을 모니터링하고, 문제가 발생한 서비스에 대해 트랜잭션 추적을 시작하거나 종속성 관계를 드릴다운할 수 있습니다.

이 섹션은 다음 항목을 포함하고 있습니다.

- [SOA 종속성 맵의 사용 이해 \(페이지 73\)](#)
- [SOA 종속성 맵이 데이터를 가져오는 방법 \(페이지 76\)](#)
- [SOA 종속성 맵의 컨텍스트 이해 \(페이지 82\)](#)
- [SOA 종속성 맵 표시 \(페이지 84\)](#)
- [물리적 또는 논리적 뷰 선택 \(페이지 86\)](#)
- [콘텐츠 유형 선택 \(페이지 87\)](#)
- ["종속성 맵" 노드의 "Primary Metric"\(기본 메트릭\) 설정 \(페이지 88\)](#)
- [종속성 맵 노드의 도구 설명 메트릭 선택 \(페이지 89\)](#)
- [표시되는 종속성 수준 변경 \(페이지 90\)](#)
- [SOA 종속성 맵 내에서 탐색 \(페이지 93\)](#)
- [SOA 종속성 맵 이미지 저장 \(페이지 96\)](#)
- [맵 노드에서 트랜잭션 추적 시작 \(페이지 98\)](#)
- [클러스터에 대한 SOA 종속성 맵 \(페이지 99\)](#)

SOA 종속성 맵의 사용 이해

일반적인 SOA 환경에는 웹 서비스나 비즈니스 프로세스 사이에 복잡한 관계가 존재합니다. 그러나 이러한 서비스는 서로 느슨하게 연결되어 있기 때문에 대체로 구성 요소 사이의 관계를 추적하거나 이해하기가 어렵습니다. SOA 종속성 맵은 배포되어 있는 서비스를 시각적으로 보여 주기 때문에 다양한 구성 요소가 서로 어떻게 연결되어 있는지 모니터링하고 이해하는 데 도움이 됩니다.

SOA 환경의 문제점 이해

대부분의 조직에서는 서비스를 개발하고 배포하는 데 여러 그룹이 관련되며, 일반적으로 서비스가 어떻게 사용되는지를 하나의 아키텍처 그림으로 요약할 수 없습니다. 배포된 서비스를 기록하는 레지스트리가 존재하는 경우도 있지만 그 목록이 최신 상태가 아닐 수도 있습니다. 서비스 레지스트리가 비교적 잘 관리되는 환경에서도 레지스트리가 올바른 사용자에게 바로 사용할 수 있도록 제공되지 못하거나, 서비스 사이의 복잡한 관계를 정확하게 설명하지 못할 수 있습니다.

예를 들어 일반적으로 응용 프로그램은 서로 다른 그룹에서 개발한 다양한 서비스를 포함합니다. 이러한 추가적인 서비스 계층은 쉽게 드러나지 않으며 응용 프로그램 개발자가 전혀 알지 못할 수도 있지만 이러한 서비스의 성능은 서비스에 의존하는 응용 프로그램에 있어서 매우 중요한 문제가 됩니다. 이와 같이 "숨겨진" 서비스를 노출하면 문제를 찾고 격리하고 해결하는 데 필요한 추가적인 수준의 가시성을 얻을 수 있습니다.

SOA 종속성 맵이 제공하는 정보 이해

SOA 환경의 문제점을 해결하려면 SOA 환경에 포함된 모든 서비스를 자동으로 적시에 볼 수 있어야 합니다. 뿐만 아니라 특정 서비스가 다른 서비스에서 언제 어떻게 사용되는지 또는 잠재적인 병목 현상이 나타나는 위치를 이해할 수 있도록 구성 요소 사이의 관계를 탐색할 수 있어야 합니다. 이 정보는 운영 환경에 포함된 모든 서비스의 상태와 가용성을 실시간으로 모니터링하는 데 매우 중요합니다.

SOA 종속성 맵은 바로 이러한 가시성을 제공합니다. 종속성 맵을 사용하면 다음과 같은 정보를 한눈에 실시간으로 볼 수 있습니다.

- 실제로 배포되어 있는 서비스
- 새 서비스가 배포되거나 종속성이 변경되는 시점
- 서비스 사이의 관계
- 클라이언트 측 요청과 서버 측 응답의 상태

종속성 및 SOA 용어 이해

SOA 종속성 맵의 컨텍스트에서 서비스라는 용어는 일반적으로 특정한 결과를 얻기 위해 설계된 일련의 작업이라는 의미로 사용됩니다. 서비스 구성 요소에 대한 세부 정보는 환경마다 달라지거나 다르게 정의될 수 있습니다. 예를 들어 종속성 맵에서 "서비스" 뷔를 선택하면 웹 서비스, 비즈니스 프로세스, 어댑터 또는 메시징 서비스에 대한 정보가 표시될 수 있습니다.

하나의 작업이 후속 작업을 호출할 때 종속성이 존재합니다. 후속 작업은 완전히 별개의 서비스에 속하고 다른 응용 프로그램 서버에서 실행될 수 있습니다. 예를 들어 사용자가 여행 웹 사이트에 방문하여 항공편을 예약하는 트랜잭션을 시작하는 경우를 살펴 보겠습니다. 이 경우 여행 웹 사이트의 예약 서비스는 좌석 예약을 위한 항공사 예약 서비스를 호출한 후 티켓 구매를 위한 신용 카드 지불 서비스를 호출합니다. 이 예제에서 트랜잭션을 시작한 여행 웹 사이트의 서비스는 트랜잭션을 완료하기 위해 항공 예약 서비스와 신용 카드 처리 서비스에 종속됩니다.

대부분의 경우 종속성은 높은 순위의 서비스를 구성하기 위해 낮은 수준의 서비스를 결합하여 발생한 결과입니다. 예를 들어 여행 웹 사이트의 예약 서비스는 항공 예약 서비스와 신용 카드 처리 서비스가 성공적으로 처리되어야만 완료될 수 있습니다. 이러한 두 서비스 중 하나라도 실패하거나 성능 문제가 발생하면 여행 웹 사이트의 예약 서비스가 직접적인 영향을 받습니다.

사용자가 선택하는 컨텍스트에 따라 SOA 종속성 맵에는 배포된 모든 상위 수준 서비스가 표시되거나, 개별 서비스의 자세한 구조와 관련 종속성 및 종속 작업에 대한 성능 정보가 표시될 수 있습니다. 표시 범위와 세부 수준을 사용자가 변경할 수 있기 때문에 SOA 종속성 맵은 SOA 환경에서 발생한 문제를 분석하는 데 뛰어난 유연성을 제공합니다. SOA 종속성 맵을 사용하면 프로세스 흐름에서 서비스와 작업 사이의 종속성을 추적하여 느린 응답 시간, SOAP 결함 또는 기타 문제의 근본적인 원인을 파악할 수 있습니다.

SOA 종속성 맵을 통해 수행할 수 있는 작업 이해

SOA 종속성 맵을 사용하면 다음과 같은 중요한 작업을 완료할 수 있습니다.

- 서비스에 대한 정확한 최신 정보를 설계된 대로가 아니라 배포된 대로 봅니다.
- 서비스 종속성을 보고, **Investigator** 트리에서 맵 노드를 클릭하여 구성 요소의 세부 정보로 드릴다운함으로써 서비스 문제를 조사하고 해결합니다.
- 응용 프로그램 종속성에 대한 물리적 뷰와 논리적 뷰를 모두 제공하여 IT 구축자와 함께 기업 전반의 SOA 아키텍처를 계획하고 실현합니다.
- 맵의 스냅숏 이미지를 만들어 서비스 종속성 및 서비스 문제에 대한 정보를 조직의 다른 이해당사자와 공유합니다.
- 응용 프로그램 전문가의 도움 없이 모니터링된 에이전트, 서비스 및 작업의 종속성 관계를 이해하고, 이러한 종속성이 서비스 성능에 미치는 영향을 이해합니다.
- SOA 종속성 맵에서 바로 문제 서비스에 대한 트랜잭션 추적을 시작하여 트랜잭션 추적을 수동으로 설정하지 않고 문제 진단을 진행합니다.

SOA 종속성 맵이 데이터를 가져오는 방법

CA APM for SOA 를 처음 설치하고 응용 프로그램 서버를 시작하면 에이전트는 발견된 모든 웹 서비스 또는 비즈니스 프로세스를 자동으로 검색하여 이 정보를 **Enterprise Manager** 로 보냅니다.

이러한 서비스는 호출되어 실행되었기 때문에 에이전트는 프로세스에서 호출 순서를 추적하고 이 정보를 **Enterprise Manager** 로 보냅니다. **Enterprise Manager** 는 호출 순서 정보를 사용하여 서비스 간의 종속성을 식별합니다. **Enterprise Manager** 는 검색된 종속성을 파일에 저장하고 맵 데이터를 표시하기 위해 **Workstation** 으로 보냅니다.

초기 검색 후 에이전트는 주기적으로 새 응용 프로그램 및 이전에 검색된 종속성에 대한 변경 사항을 검사합니다. 새 서비스가 발견되면 에이전트는 서비스 호출에 대한 세부 정보를 **Enterprise Manager** 로 보내고, **Enterprise Manager** 는 새 맵 데이터를 **Workstation** 으로 보내서 SOA 종속성 맵 표시를 이전에 알려지지 않은 서비스 또는 이미 알려진 서비스에 대해 새로 검색된 종속성으로 업데이트합니다.

새 SOA 종속성이 검색되면 Enterprise Manager 는 논리적 동등성 규칙이 활성화되었는지 확인하여 활성화된 경우에는 새 종속성이 기존 종속성과 논리적으로 동등한지 여부를 확인합니다. 새 종속성이 고유한 경우에는 Enterprise Manager 에 저장된 종속성 데이터에 추가됩니다. Workstation 은 데이터 저장소를 15 초마다 검사하여 SOA 종속성 맵에 대한 데이터가 변경되었는지 확인한 다음 필요에 따라 SOA 종속성 맵 표시를 업데이트합니다.

SOA 종속성 맵의 지속 데이터 정보

Enterprise Manager 는 검색된 SOA 종속성을 로컬 데이터 저장소에 저장하여 재검색을 요청하지 않고도 정보를 사용할 수 있도록 합니다. SOA 종속성에 대한 데이터 저장소는 <EM_Home>/data/dependencymap 디렉터리에 있는 다음 파일로 구성됩니다.

- dependencymap.sav 에는 최근에 저장된 종속성 데이터가 포함됩니다.
- dependencymap.bak 는 이전에 저장된 SOA 종속성의 백업 사본이 포함됩니다.

Enterprise Manager 는 1 시간마다 그리고 수동으로 종료될 때마다 검색된 모든 종속성을 dependencymap.sav 파일에 저장합니다. 또한 dependencymap.sav 파일 저장 간격 사이에는 SOA 종속성 맵의 현재 검색된 종속성을 메모리에 보관합니다.

Enterprise Manager 는 시작될 때 dependencymap.sav 파일에서 종속성을 로드하므로 CA APM for SOA 가 이전에 검색된 종속성을 다시 검색할 필요가 없습니다. dependencymap.sav 파일의 새 사본을 저장하지 않고 Enterprise Manager 가 갑자기 중지된 경우에는 dependencymap.sav 파일의 데이터가 최대 1 시간 오래된 데이터일 수 있습니다. dependencymap.sav 파일이 없으면 Enterprise Manager 는 에이전트가 서비스 종속성을 다시 검색할 때까지 기다린 후에 파일을 다시 만듭니다.

Enterprise Manager에서 종속성을 강제로 다시 검색

대부분의 경우 *dependencymap.sav* 파일에 저장되어 있는 데이터를 로드하면 다시 검색하지 않고 최신 정보를 볼 수 있습니다. 그러나 다음과 같이 Enterprise Manager를 시작할 때 이전에 저장된 종속성을 무시하고 모든 종속성을 강제로 다시 검색해야 하는 경우도 있습니다. 예:

- 논리적 동등성 추론의 사용 여부를 변경하려는 경우 Enterprise Manager에서 모든 종속성을 강제로 다시 검색하여 종속성에 새 규칙을 적용해야 할 수 있습니다. 논리적 동등성 추론에 대한 자세한 내용은 [논리적 동등성 규칙 정보](#) (페이지 82)를 참조하십시오. 논리적 동등성 추론의 속성을 설정하는 데 대한 자세한 내용은 [Enterprise Manager 속성 구성](#) (페이지 363)을 참조하십시오.
- Enterprise Manager가 중지된 동안 에이전트 연결을 끊으면 해당 에이전트의 맵 노드와 종속성은 만료될 때까지(기본적으로 60 일) 계속 표시됩니다. 이 경우에는 Enterprise Manager에서 강제로 종속성을 다시 검색하여 만료될 때까지 기다릴 필요 없이 연결이 끊어진 에이전트의 데이터를 종속성 맵에서 제거할 수 있습니다.

저장된 종속성 데이터 무시 또는 제거

Enterprise Manager가 저장된 종속성 데이터를 무시하고 시작 시 종속성을 다시 검색하도록 하려면 *dependencymap.sav* 파일과 *dependencymap.bak* 파일을 삭제하거나, 이동하거나 이름을 변경해야 합니다.

저장된 SOA 종속성 맵 파일을 삭제하거나, 이동하거나, 이름을 바꾸려면

1. Enterprise Manager를 중지합니다.
2. *<EM_Home>/data/dependencymap* 디렉터리로 이동합니다.
3. *dependencymap.sav* 파일을 삭제하거나, 이동하거나, 이름을 바꿉니다. 예를 들어 Enterprise Manager에서 파일의 데이터를 사용하여 종속성 맵을 표시하지 않게 하려면 파일 이름을 *old_dependencymap.savedcopy*로 변경합니다.
4. *dependencymap.bak* 파일을 삭제하거나, 이동하거나, 이름을 바꿉니다. 예를 들어 Enterprise Manager에서 파일을 사용하여 종속성 맵을 표시하지 않게 하려면 파일 이름을 *old_dependencymap.backupcopy*로 변경합니다.
5. Enterprise Manager를 시작하고 현재 종속성을 다시 검색한 다음 종속성 맵과 *dependencymap.sav* 파일을 새로 만듭니다.

에이전트 연결 끊기 및 다시 마운트

Enterprise Manager 가 실행 중일 때 에이전트와 Enterprise Manager 사이의 연결이 끊어지면 해당 에이전트의 맵 노드와 종속성이 비활성화됩니다. 에이전트를 Workstation에서 마운트 해제하면 마운트 해제 이벤트로 인해 제거된 서비스에서 시작하는 또는 끝나는 모든 종속성을 포함하여 해당 에이전트의 모든 맵 노드와 종속성이 맵에서 제거됩니다.

에이전트를 다시 마운트하면 Enterprise Manager 는 에이전트의 모든 종속성을 다시 검색한 후 맵에 다시 추가합니다.

새 서비스와 작업으로 종속성 맵 업데이트

에이전트는 새로운 서비스나 작업이 검색될 때마다 업데이트된 정보를 Enterprise Manager 에 보냅니다. 그러면 Enterprise Manager 는 종속성 맵 저장소를 업데이트하고 SOA 종속성 맵에 새 데이터를 표시할 준비가 되었음을 Workstation 에 알립니다. 이 알림을 받으면 Workstation 에 "사용 가능한 데이터가 변경되었습니다."라는 메시지가 표시됩니다. 그러면 Workstation 의 새로운 서비스나 작업을 표시하도록 종속성 맵을 다시 로드할 수 있습니다.

SOA 종속성 맵을 다시 로드하려면

1. SOA 종속성 맵 도구 모음 아래에 "사용 가능한 데이터가 변경되었습니다." 메시지가 표시되는지 확인합니다.
2. "여기를 클릭하십시오."를 사용하여 종속성을 다시 로드한 후 새 서비스 또는 작업과 관련 메트릭을 봅니다. 예를 들어 다시 로드한 후에는 "사용 가능한 데이터가 변경되었습니다." 메시지가 더 이상 표시되지 않고 새로 검색된 서비스와 종속성이 맵에 포함됩니다.

종속성 맵을 다시 로드하면 마지막으로 클릭했던 Investigator 트리 노드와 관련된 기본 뷰가 표시됩니다. 예를 들어 WebServices 노드를 선택한 상태에서 맵을 다시 로드한 경우에는 서비스에 대한 "물리적" 뷰가 표시됩니다.

콘텐츠 유형을 변경한 경우, "사용 가능한 데이터가 변경되었습니다."가 표시된 후 맵을 다시 로드하면 WebServices 트리 노드의 기본 뷰인 서비스에 대한 "물리적" 뷰가 다시 로드됩니다. Investigator 트리에서 에이전트 노드와 같이 다른 노드를 클릭한 후 맵을 다시 로드하면 에이전트에 대한 "물리적" 뷰가 표시됩니다.

사용되지 않는 맵 노드의 만료 처리 및 제거

Enterprise Manager 는 검색된 각 종속성의 경과 시간을 추적하고 주기적으로 종속성을 다시 검색하여 여전히 존재하는지 확인합니다. 특정 에이전트, 서비스 또는 작업의 맵 노드가 일정 기간 이후 다시 검색되지 않으면 해당 항목이 만료 처리되고 종속성 맵에서 제거됩니다. 제거된 에이전트, 서비스 또는 작업이 나중에 다시 호출되면 Enterprise Manager 는 해당 항목을 종속성 맵에 다시 추가합니다.

기본적으로 Enterprise Manager 는 최근에 검색된 종속성을 6 시간마다 확인하고, 60 일 동안 검색되지 않은 종속성을 제거합니다.

Enterprise Manager 가 검색된 종속성의 기간을 검사하는 빈도 및 다시 검색되지 않은 종속성을 맵에 유지할 기간은 *IntroscopeEnterpriseManager.properties* 파일의 속성 값을 설정하여 구성할 수 있습니다. Enterprise Manager 속성 구성에 대한 자세한 내용은 [Enterprise Manager 속성 구성](#) (페이지 363)을 참조하십시오.

종속성 맵에 불완전한 데이터가 표시되는 경우에 수행할 작업

대개의 경우 서비스 및 종속성 관계의 초기 검색에는 에이전트가 정보를 수집한 다음 Enterprise Manager 에 제공해야 하므로 완료되는 데 시간이 필요합니다. 초기 검색 후에는 서비스 및 종속성에 대한 변경 사항이 항상 즉시 표시되지는 않습니다. 일반적으로 독립 실행형 Enterprise Manager 를 실행 중인 경우에는 변경 사항이 표시되는 데 몇 분밖에 걸리지 않습니다. 클러스터에서 Enterprise Manager 를 실행 중인 경우에는 시간이 조금 더 걸릴 수 있습니다.

하지만 에이전트가 모든 노드 및 종속 관계에 대하여 데이터를 수집하려면 서비스가 실행되거나 능동적으로 실행 중이어야 합니다. 호출되지 않은 서비스 및 작업은 호출되기 전에는 맵에 표시되지 않습니다.

종속성 맵에 데이터가 표시되지 않거나 불완전한 데이터가 표시되는 경우에는 다음과 같은 원인 때문일 수 있습니다.

- 모니터링 중인 응용 프로그램 서버 또는 서비스가 동적으로 실행 중이지 않거나 아직 호출되지 않았습니다. 서버가 실행 중이고 서비스가 호출될 수 있는지 확인하십시오.
- 실행 중인 에이전트와 클러스터되지 않은(MOM 이 아닌) 환경의 Enterprise Manager 사이의 연결이 끊겼습니다. 에이전트 캐시를 플러시하여 에이전트가 이전에 검색된 종속성을 다시 검색하도록 함으로써 정보를 새로운 또는 다시 시작된 Enterprise Manager로 전송합니다.
- 에이전트 캐시 플러시로 인해 모든 서비스 및 종속성 데이터가 제거되었습니다. 캐시 플러시 후 에이전트가 정보를 수집하고 Enterprise Manager에 전송하도록 기다리십시오.
- 종속성 맵에 필요한 매핑 추적 프로그램이 에이전트의 프로필에서 사용하지 않도록 설정되었습니다. *IntroscopeAgent.profile* 파일의 `com.wily.introscope.agent.transactiontrace.boundaryTracing.enable` 속성이 `False`로 설정되어 있지 않음을 확인하십시오.
- 에이전트의 이름을 변경했으므로 연관된 맵 노드에 대한 데이터가 없습니다. 에이전트가 새 에이전트 맵 노드에 대해 표시할 데이터를 수집할 때까지 기다리십시오.
- 누락된 서비스가 에이전트에 의해 모니터링되지 않는 응용 프로그램 서버에서 실행 중입니다.
- 누락된 웹 서비스가 지원되지 않는 플랫폼 또는 지원되지 않는 SOAP 엔진에서 구현되었습니다.

논리적 동등성 규칙 정보

CA APM for SOA 는 서비스 및 작업에 대한 논리적 동등성을 확인하기 위하여 세 가지 추론을 제공합니다. 이 추론은 환경의 필요에 따라 사용하거나 사용하지 않도록 설정할 수 있습니다. 기본적으로 CA APM for SOA 에서는 다음 규칙을 사용하여 클라이언트 측 및 서버 측 동등성을 확인합니다.

- 클라이언트 측 서비스의 경우에는 서비스가 둘 이상의 서버 측 서비스에 종속되면 모든 서버 측 서비스가 논리적으로 동등해야 합니다.
- 서버 측 서비스의 경우에는 서비스가 둘 이상의 클라이언트 측 웹 서비스에 종속되면 모든 클라이언트 측 웹 서비스가 논리적으로 동등해야 합니다.
- 이름 일치의 경우에는 에이전트 지정자를 삭제한 후 두 물리적 서비스 작업의 메트릭 경로가 같으면 두 작업이 논리적으로 동등한 것으로 간주됩니다.

논리적 동등성 추론의 구성에 사용되는 속성에 대한 자세한 내용은 [Enterprise Manager 속성 구성](#) (페이지 363)을 참조하십시오.

SOA 종속성 맵의 컨텍스트 이해

SOA 종속성 맵에 표시되는 정보는 다음과 같은 순서로 두 가지 요소에 의해 결정됩니다.

- 선택한 SOA 종속성 맵 콘텐츠 유형
- 정보를 볼 *Investigator* 트리 노드

콘텐츠 유형과 *Investigator* 트리 노드에 따라 SOA 종속성 맵 컨텍스트가 결정됩니다. 이 컨텍스트는 다른 데이터를 선택하면 바뀝니다.

콘텐츠 유형이 표시되는 내용에 미치는 영향 이해

SOA 종속성 맵에 표시되는 내용을 결정하는 첫 번째 요소는 SOA 종속성 맵 콘텐츠 유형입니다. 다음과 같은 콘텐츠 유형 중에서 선택할 수 있습니다.

- 에이전트
- 서비스
- 작업

예를 들어 "에이전트"를 선택하면 에이전트 사이의 상위 수준 종속성을 볼 수 있고, "작업"을 선택하면 작업 사이의 하위 수준 종속성을 볼 수 있습니다. 적절한 콘텐츠 유형을 선택하면 전체 SOA 토플로지에서의 위치를 기억하면서 특정 하위 집합으로 드릴다운할 수 있습니다.

선택한 노드를 변경하지 않고 콘텐츠 유형 변경

"서비스"를 선택한 후 **Investigator** 트리에서 에이전트 노드를 클릭하면 모든 서비스 및 해당 에이전트와 관련된 서비스 수준 종속성이 SOA 종속성 맵에 표시됩니다.

같은 **Investigator** 트리 노드에서 다른 콘텐츠 유형, 예를 들어 "에이전트"를 선택하면 모든 에이전트 및 선택한 에이전트 노드와 관련된 모든 종속성이 SOA 종속성 맵에 표시됩니다. 예를 들어 **Tomcat01**이라는 에이전트가 에이전트 **WebLogic02**에서 실행되는 서비스에 종속된 경우, SOA 종속성 맵에는 **Tomcat01** 에이전트 아이콘 및 **WebLogic02** 에이전트를 가리키는 종속성 화살표가 표시됩니다. 이 화살표는 에이전트 수준 종속성을 나타냅니다.

맵의 모든 개체에 콘텐츠 유형 변경 적용

콘텐츠 유형으로 "에이전트"를 선택했고 두 에이전트(예: **Tomcat01** 에이전트와 **WebLogic02** 에이전트) 사이의 종속성이 종속성 맵에 표시된 경우, 콘텐츠 유형을 "서비스"로 변경하면 **Tomcat01** 에이전트와 **WebLogic02** 에이전트의 서비스 수준 종속성이 모두 표시됩니다. 콘텐츠 유형을 변경할 때마다 변경 사항이 SOA 종속성 맵에 현재 표시되어 있는 모든 정보에 적용됩니다. 이 경우에는 "에이전트"에서 "서비스"로 변경했기 때문에 맵에 현재 표시되어 있는 에이전트의 모든 서비스를 표시하도록 컨텍스트가 설정됩니다.

Investigator 노드가 표시되는 내용에 미치는 영향 이해

SOA 종속성 맵에 표시되는 내용을 결정하는 두 번째 요소는 사용자가 선택한 **Investigator** 트리 노드입니다. 표시되는 종속성 유형은 콘텐츠 유형에 따라 결정되지만, 맵에 포함되는 정보의 범위는 **Investigator** 노드에 따라 결정됩니다. 상위 맵 노드와 그 아래의 종속성은 항상 현재 선택된 **Investigator** 트리 노드를 기반으로 합니다. 다른 **Investigator** 트리 노드를 선택하면 관련된 SOA 종속성 맵이 아주 다르게 표시될 수 있습니다.

그러나 현재 선택된 **Investigator** 트리 노드에 대해 표시되는 맵 노드를 사용하여 새 **Investigator** 트리 노드로 이동할 수도 있습니다. 예를 들어 SOA 종속성 맵을 사용하여 광범위한 서비스 네트워크를 확장한 다음, 맵 노드를 클릭하여 새 **Investigator** 트리 노드로 이동하고 이 노드부터 맵을 다시 그릴 수 있습니다. 이렇게 하면 한 번의 클릭으로 관심 있는 특정 **Investigator** 또는 SOA 종속성 맵 노드에 초점을 맞추고 관심이 없는 네트워크 분기를 제거할 수 있습니다.

SOA 종속성 맵 표시

Investigator 트리 노드를 선택한 다음 "SOA 종속성 맵" 탭을 클릭하여 SOA 종속성 맵을 표시할 수 있습니다. 에이전트에 대해 CA APM for SOA 를 활성화한 경우 **Investigator**에서 에이전트 또는 **WebServices** 노드를 선택하여 SOA 종속성 맵을 표시할 수 있습니다. CA APM for Oracle Service Bus 또는 TIBCO BusinessWorks 와 같은 추가 SOA 플랫폼을 모니터링하는 경우에는 확장과 연관된 **Investigator** 트리 노드를 선택할 때 SOA 종속성 맵을 표시할 수 있습니다. 예를 들어 Oracle Service Bus 를 실행 중일 때는 **Investigator** 트리에서 "프록시 서비스" 노드 또는 특정 프록시 서비스 이름을 선택하여 모든 프록시 서비스 또는 특정 프록시 서비스에 대한 종속성 맵을 표시할 수 있습니다.

SOA 종속성 맵을 표시하려면

1. **Investigator** 트리에서 적절한 노드를 선택합니다.
2. "뷰어" 창에서 "SOA 종속성 맵" 탭을 클릭합니다.

"SOA 종속성 맵"이 앞서 선택한 **Investigator** 노드에 기반한 기본 뷰에서 표시됩니다. 예:

- 에이전트 노드의 경우 SOA 종속성 맵에는 에이전트 수준 종속성의 "물리적" 뷰가 표시됩니다.
- "WebServices", "클라이언트" 또는 "서버" 노드의 경우 SOA 종속성 맵에는 서비스 수준 종속성의 "물리적" 뷰가 표시됩니다.

- <web_service_namespace> 노드의 경우 SOA 종속성 맵에는 서비스 수준 종속성의 "물리적" 뷰가 표시됩니다.
- <operation_name> 노드의 경우 SOA 종속성 맵에는 작업 수준 종속성의 "물리적" 뷰가 표시됩니다.

예를 들어 **Investigator** 트리에서 "WebServices" > "서버"를 선택한 다음 "SOA 종속성 맵" 탭을 클릭하면 맵에 모든 서버 측 서비스와 다른 서비스에 대한 다운스트림 종속성의 한 수준이 표시됩니다.

Investigator 트리 노드와 맵 노드 정보

Investigator 트리에서 노드를 선택하면 해당 노드가 SOA 종속성 맵의 시작 지점이 됩니다. 이때 선택한 노드에서 시작되는 종속성만 SOA 종속성 맵에 표시됩니다.

SOA 종속성 맵의 각 항목은 맵 노드로 간주됩니다. 표시되는 맵 노드는 선택한 콘텐츠 유형에 따라 달라집니다. 예를 들어 맵 노드는 에이전트, 클라이언트 측 서비스나 서버 측 서비스 또는 개별 작업을 나타낼 수 있습니다.

독립 실행형 맵 노드 및 종속 맵 노드 정보

종속성이 전혀 없는 에이전트, 서비스 또는 작업은 독립 실행형 맵 노드로 표시됩니다. 설정한 컨텍스트에 따라 SOA 종속성 맵에 독립 실행형 맵 노드만 표시되거나, 독립 실행형 맵 노드와 종속 맵 노드가 함께 표시되거나, 종속 맵 노드만 표시될 수 있습니다. 예를 들어 콘텐츠 유형으로 "서비스"를 선택하고 **Investigator** 트리에서 "클라이언트" 또는 "서버"를 선택하면 "SOA 종속성 맵"에 선택한 노드와 관련된 독립 실행형 서비스와 종속 서비스가 모두 표시될 수 있습니다.

Investigator 노드를 변경하지 않은 채로 콘텐츠 유형만 "에이전트"로 변경할 경우, 에이전트 수준 종속성이 없으면 독립 실행형 에이전트 맵 노드 하나만 표시될 수 있습니다.

그러나 콘텐츠 유형을 "작업"으로 변경하면 선택한 **Investigator** 트리 노드와 관련된 모든 종속 작업에 대해 하나의 종속성 수준만 맵에 표시됩니다. 종속 작업에는 클라이언트 측 작업 또는 서버 측 작업이 포함될 수 있습니다.

작업 맵 노드 정보

콘텐츠 유형을 "작업"으로 설정하거나, 서비스를 마우스 오른쪽 단추를 클릭한 후 "모든 작업 표시"를 선택하면 다음과 같은 규칙을 사용하여 "SOA 종속성 맵"에 작업 수준 종속성이 표시됩니다.

- 클라이언트 측 서비스 네임스페이스는 초록색 상자 맵 노드로 표시됩니다.
- 서버 측 서비스 네임스페이스는 파란색 상자 맵 노드로 표시됩니다.
- 에이전트 이름은 회색 상자 맵 노드로 표시됩니다.

이러한 색상 상자 사용 규칙은 특정 작업이 속해 있는 서비스 및 에이전트를 쉽게 알 수 있도록 하기 위하여 사용됩니다.

물리적 또는 논리적 뷰 선택

SOA 종속성 맵에서는 환경의 SOA 구성 요소를 표시하는 두 가지 방법을 제공합니다.

- 물리적 뷰는 에이전트가 발견된 서비스 및 작업의 물리적 위치에 기반한 표현 방식입니다.
- 논리적 뷰는 물리적 위치에 관계없이 비슷하게 이름이 지정된 서비스 및 작업의 그룹화에 기반한 표현 방식입니다.

물리적 뷰와 논리적 뷰의 차이 및 문제 해결을 위해 두 가지 뷰를 전환하는 방법을 설명하기 위해 다음 예를 들어보겠습니다.

서비스 문제를 분석하는 중에 특정 서비스가 근본 원인으로 확인되었지만 이 서비스가 분산 서비스라는 점을 발견했습니다. 즉 부하 분산기가 이 서비스에 대한 요청을 다섯 개의 응용 프로그램 서버로 분산합니다.

이 시나리오에서 문제의 원인을 찾으려면 다음을 이해해야 합니다.

- 한 서비스가 다른 서비스를 호출하는 시점을 확인하기 위한 논리적 인스턴스 간의 관계
- 실제 호출된 물리적 인스턴스를 확인하기 위한 물리적 인스턴스 간의 관계(서비스 A의 어느 인스턴스가 서비스 B의 어느 인스턴스를 호출하는지)
- 물리적 인스턴스 간의 성능 차이
- 서비스에 대한 트래픽이 인스턴스로 분산되는 방식

예를 들어 논리 서비스에서 분당 100 개의 호출이 발생하고 이 서비스가 5 개의 인스턴스로 구성된 경우, 100 개의 호출이 각각의 개별 인스턴스로 어떻게 분산되는지를 이해해야 합니다. 100 개의 호출 중 90 개가 단일 인스턴스로 가는 경우에는 문제를 이 인스턴스로 격리하고 문제를 해결한 다음, 인스턴스의 부하 증가에 대한 원인을 찾기 위한 추가 분석을 수행할 수 있습니다.

"SOA 종속성 맵"에서는 "표시" 드롭다운 목록을 사용하여 물리적 뷰와 논리적 뷰 간을 전환할 수 있습니다. 이렇게 하면 서비스의 논리적 인스턴스와 물리적 인스턴스를 구분하여 특정 물리적 인스턴스의 문제가 논리적 서비스의 전체 성능에 어떤 영향을 미치는지 확인할 수 있습니다.

SOA 종속성 맵 물리적 뷰 또는 논리적 뷰를 선택하려면

- SOA 종속성 맵 도구 모음의 "표시" 드롭다운 목록을 클릭한 다음 "물리적" 또는 "논리적"을 선택합니다.

SOA 종속성 맵이 새로 고쳐지고 선택한 뷰에 기반하여 SOA 환경이 표시됩니다. 예를 들어 "물리적" 뷰에서 "작업"을 보다가 "논리적" 뷰를 선택하면 SOA 종속성 맵이 새로 고쳐지고 SOA 종속성 맵의 논리적 표현에 기반하여 작업이 표시됩니다.

Investigator 트리에서 "가상 에이전트" 노드를 선택한 경우를 제외하고 SOA 종속성 맵에서는 기본적으로 "물리적" 뷰가 표시됩니다. "가상 에이전트" 노드 및 그 하위 노드의 경우에는 기본적으로 "논리적" 뷰가 표시됩니다.

콘텐츠 유형 선택

콘텐츠 유형에 따라 화면에 표시되는 내용이 달라지므로 "작업", "서비스" 또는 "에이전트"에 대한 종속성을 사용하여 맵을 표시할 수 있습니다. 기본 콘텐츠 유형은 Investigator 트리에서 선택한 노드에 따라 다릅니다. 콘텐츠 유형을 변경하여 다른 관점에서 종속성 관계를 드릴다운 할 수 있습니다. 뷰 간을 전환하면서 문제 서비스를 진단할 수 있습니다.

다음 단계를 수행하십시오.

1. "SOA 종속성 맵" 도구 모음에서 콘텐츠 유형 드롭다운 목록을 클릭합니다.
2. "에이전트", "서비스" 또는 "작업"을 선택합니다.

참고: "논리적" 뷔를 표시하는 경우 또는 "사용자 지정 가상 에이전트"를 선택한 경우에는 "작업" 또는 "서비스"만 선택할 수 있습니다.

"SOA 종속성 맵"이 새로 고쳐져서 콘텐츠 유형에 대한 종속성이 표시됩니다.

추가 정보:

콘텐츠 유형이 표시되는 내용에 미치는 영향 이해 (페이지 82)

"종속성 맵" 노드의 "Primary Metric"(기본 메트릭) 설정

SOA 종속성 맵에서는 표준 CA Introscope 및 SOA 특정 메트릭을 모두 볼 수 있습니다. 각 SOA 종속성 맵 서비스 및 작업 노드 아래에서 기본 메트릭은 사용자에게 중요한 표준 CA Introscope 메트릭을 표시합니다. 또한 SOA 종속성 맵 에이전트, 서비스 또는 작업 노드 위에 마우스를 놓으면 표시하도록 선택한 메트릭이 도구 모음에 표시됩니다.

에이전트 및 서비스에 대해 SOA 종속성 맵에 표시되는 메트릭은 현재 선택되거나 SOA 종속성 맵에 표시된 맵 노드뿐만 아니라 해당 에이전트나 서비스 아래의 모든 작업에 대해 집계된 값입니다. 예를 들어 서버 측 서비스 맵 노드를 마우스로 가리키면 표시하도록 선택한 메트릭이 도구 설명에 표시됩니다. 하지만 맵 노드에 대한 메트릭에는 모니터링되지 않는 클라이언트와 종속 클라이언트 노드에 대한 클라이언트 호출, 또는 C++ 클라이언트처럼 SOA 종속성 맵에 나타나지 않는 클라이언트가 포함될 수 있습니다. 모든 클라이언트 호출은 서버 측 메트릭 계산에 반영됩니다.

다음 단계를 수행하십시오.

1. SOA 종속성 맵 도구 모음에서 "Primary Metric"(기본 메트릭) 드롭다운 목록을 클릭합니다.
2. 모든 종속성 맵 노드에 표시할 표준 CA Introscope® 메트릭 중 하나를 선택합니다.
 - 평균 응답 시간
 - 간격당 응답 수
 - 간격당 오류 수
 - 동시 호출
 - 중단 수

SOA 종속성 맵의 기본 메트릭에 대한 기본값은 "평균 응답 시간"입니다.

기본 메트릭을 설정하면 종속성 맵의 모든 맵 노드에 해당 메트릭의 현재 메트릭 값이 표시됩니다.

맵 노드를 마우스로 가리키면 수집된 데이터 포인트 수의 최소값, 최대값 및 현재 개수와 같은 추가 정보가 도구 설명에 표시됩니다.

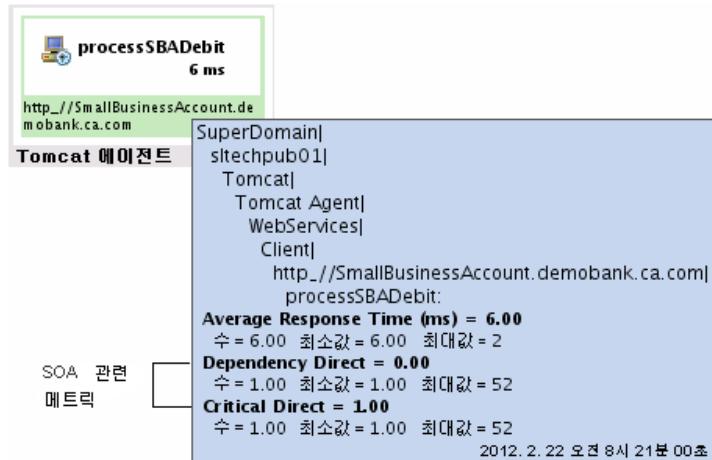
종속성 맵 노드의 도구 설명 메트릭 선택

언제든지 "SOA 종속성 맵"에서 맵 노드를 커서로 가리키면 도구 설명이 표시됩니다. 도구 설명에는 기본적으로 기본 메트릭 데이터가 표시됩니다. 필요한 경우 도구 설명에 SOA 관련 메트릭을 표시할 수도 있습니다. 예를 들어 도구 설명에 직접 및 간접 종속성 메트릭 또는 편차 메트릭을 표시하도록 선택할 수 있습니다.

SOA 종속성 맵 도구 설명 메트릭을 설정하려면

1. "SOA 종속성 맵" 도구 모음에서 "도구 설명" 단추를 클릭하여 "도구 설명 메트릭 선택" 대화 상자를 표시합니다.
2. SOA 종속성 맵 도구 설명으로 표시할 메트릭을 선택합니다. 예를 들어 "종속성 직접" 및 "중요 직접" 메트릭 같은 SOA 관련 메트릭을 선택할 수 있습니다.
3. "확인"을 클릭합니다.

SOA 종속성 맵을 마우스로 가리키면 현재 선택된 **Investigator** 노드를 기준으로, **Enterprise Manager** 가 해당 **Investigator** 트리 노드에 대해 보고하는 메트릭에 따라 선택한 메트릭의 전체 또는 일부가 SOA 종속성 맵에 표시됩니다. 예를 들어 관련된 "종속성 직접" 도구 설명 메트릭을 표시하도록 SOA 종속성 맵을 설정한 후 작업 노드를 마우스로 가리키면 도구 설명에 이 메트릭에 대한 데이터가 표시됩니다.



그러나 서비스는 "종속성 직접" 메트릭을 계산하는 데이터를 생성하지 않으므로 서비스 노드를 마우스로 가리키면 "종속성 직접" 메트릭이 표시되지 않습니다.

표시되는 종속성 수준 변경

Investigator 트리에서 노드를 선택하고 "SOA 종속성 맵" 탭을 클릭하면 선택한 노드의 종속성과 추가적인 종속성 수준 하나가 SOA 종속성 맵에 기본적으로 표시됩니다. 표시되는 종속성을 수정하는 몇 가지 방법이 있습니다.

- 맵에서 개별 항목에 대한 추가적인 종속성을 확인할 수 있습니다.
- 맵에서 개별 항목에 대한 하위 수준 종속성을 숨길 수 있습니다.
- 맵의 모든 항목에 대해 다른 전체 수준 종속성을 확장하거나 펼칠 수 있습니다.
- 맵의 모든 항목에 대해 가장 하위 수준의 종속성을 숨기거나 롤업할 수 있습니다.

맵 노드의 추가적인 종속성 확인

기본적으로 SOA 종속성 맵에는 종속성 수준이 하나만 표시됩니다. 특정 맵 노드에 관심이 있는 경우 다른 맵 노드의 하위 수준 종속성을 확장할 필요 없이 해당 에이전트, 서비스 또는 작업에 대한 하위 수준 종속성을 추가로 확인할 수 있습니다.

맵의 항목에 대한 추가적인 종속성을 확인할 때는 한 번에 종속성 수준 하나씩 펼칠 수 있습니다. 추가적인 종속성을 확인할 때마다 그 다음으로 낮은 수준의 종속성이 맵에 추가됩니다. 다음 수준을 확인했을 때 더 이상 추가적인 종속성이 없으면 "추가 종속성이 없습니다."라는 메시지가 표시됩니다.

특정 맵 노드에서 추가적인 종속성을 확인하려면

- 맵에서 개별 항목을 마우스 오른쪽 단추로 클릭한 다음 메뉴에서 "다음 종속성 표시"를 선택합니다.

SOA 종속성 맵에서 선택한 맵 노드에 대해서만 다음 종속성 수준이 펼쳐집니다. 예를 들어 서버 측 작업을 펼치면 관련된 클라이언트 측 작업이 다음 종속성 수준으로 펼쳐집니다.

맵 노드의 종속성 숨기기

특정 에이전트, 서비스 또는 작업에 대해 종속성 수준을 여러 개 펼쳐 놓은 경우 해당 항목의 종속성 수준 일부를 숨길 수 있습니다. 맵에서 항목에 대한 종속성을 숨길 때는 종속성 수준이 한 번에 하나씩 롤업됩니다. 종속성을 숨길 때마다 가장 낮은 다음 종속성 수준이 맵에서 제거됩니다.

특정 맵 노드에서 종속성을 제거하려면

- 맵에서 개별 항목을 마우스 오른쪽 단추로 클릭한 다음 메뉴에서 "종속성 숨기기"를 선택합니다.

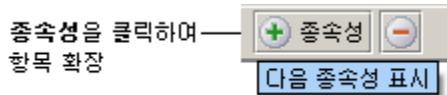
선택한 맵 노드의 모든 종속성 수준이 SOA 종속성 맵에서 롤업됩니다.

맵에 있는 모든 항목에 대한 추가적인 종속성 확인

기본적으로 SOA 종속성 맵에는 종속성 수준이 하나만 표시됩니다. 현재 맵에서 모든 항목에 대한 추가적인 하위 수준 종속성을 확인하려면 전체 맵에 대하여 종속성 수준을 한 번에 하나씩 펼치면 됩니다. 추가적인 종속성을 확인할 때마다 그 다음으로 낮은 수준의 종속성이 맵에 추가됩니다. 다음 수준을 확인했을 때 더 이상 추가적인 종속성이 없으면 "추가 종속성이 없습니다."라는 메시지가 표시됩니다.

맵의 모든 항목에 대해 종속성을 펼치려면

- SOA 종속성 맵 도구 모음에서 "다음 종속성 표시" 단추를 클릭합니다. 예:



SOA 종속성 맵에서 맵의 모든 하위 수준 항목에 대한 종속성이 펼쳐집니다. 초기 맵 노드에 종속성이 3 개 표시되어 있는 경우 이 단추를 클릭하면 세 가지 항목 각각에 대해 새 종속성 수준이 추가됩니다.

맵의 모든 항목에 대해 종속성 숨기기

모든 에이전트, 서비스 또는 작업에 대해 종속성 수준을 여러 개 펼쳐놓은 경우 전체 맵에서 일부 종속성 수준을 제거할 수 있습니다. 맵에서 모든 항목에 대한 종속성을 숨길 때는 종속성 수준이 한 번에 하나씩 롤업됩니다. 종속성을 숨길 때마다 가장 낮은 다음 종속성 수준이 맵에서 제거됩니다.

맵의 모든 항목에 대해 종속성을 숨기려면

- SOA 종속성 맵 도구 모음에서 "마지막 종속성 숨기기" 단추를 클릭합니다. 예:



SOA 종속성 맵에서 맵의 가장 하위 수준 종속성이 롤업됩니다.

SOA 종속성 맵 내에서 탐색

SOA 종속성 맵을 확장하여 여러 개의 종속성 수준을 표시하는 경우 또는 조직의 SOA 인프라가 유난히 복잡한 경우에는 SOA 종속성 맵을 탐색하거나 원하는 정보를 찾기가 어려울 수 있습니다. SOA 종속성 맵에는 맵 내에서 탐색하거나, **Investigator** 와 맵 사이를 이동하거나, 정보를 신속하게 표시하거나 숨길 수 있게 도와주는 몇 가지 도구가 있습니다.

SOA 종속성 맵의 이동, 확대/축소 및 맞춤

현재 보고 있는 종속성 맵이 큰 경우 내용을 더 잘 파악하기 위해 특정 섹션을 탐색하거나 일부분의 크기를 조정해야 할 수 있습니다. SOA 종속성 맵에는 템 내에서 SOA 종속성 맵을 이동, 확대 및 다시 맞추게 하는 도구가 있습니다.

- 이동
- Magnify a selection(선택 영역에 돋보기 사용)
- 확대/축소
- 템에 맞춤

템 뷰에서 SOA 종속성 맵을 이동하려면

1. SOA 종속성 맵 도구 모음에서 "Pan"(이동) 단추를 클릭합니다. "Pan"(이동) 단추는 손 모양 아이콘으로 표시됩니다.
2. SOA 종속성 맵 내의 위치를 클릭합니다.
3. 템 내에서 맵을 가운데로 끌어 원하는 영역을 확인합니다.

SOA 종속성 맵의 영역에 돋보기를 사용하려면

1. SOA 종속성 맵 도구 모음에서 "Magnify a selection"(선택 영역에 돋보기 사용) 단추를 클릭합니다. "Magnify a selection"(선택 영역에 돋보기 사용) 단추는 선택 영역 내의 돋보기로 표시됩니다.
2. SOA 종속성 맵 내의 위치를 클릭합니다.
3. 돋보기로 확대할 사각형 영역을 마우스로 끌어 선택합니다.

SOA 종속성 맵을 확대하거나 축소하려면

1. SOA 종속성 맵 도구 모음에서 "Zoom in and out"(확대/축소) 단추를 클릭합니다. "Zoom in and out"(확대/축소) 단추는 위쪽 화살표와 아래쪽 화살표가 표시된 돋보기로 표시됩니다.
2. SOA 종속성 맵 내에서 마우스 단추를 클릭한 채로 누르고 있습니다.
3. SOA 종속성 맵에서 마우스 휠을 앞쪽으로 돌려 확대하거나 뒤쪽으로 돌려 축소합니다.

SOA 종속성 맵 크기를 탭에 맞게 조정하려면

1. SOA 종속성 맵 도구 모음에서 "Fit to Tab"(탭에 맞춤) 단추를 클릭합니다. 이 단추는 눈금선이 있는 돋보기로 표시됩니다.
SOA 종속성 맵의 크기가 탭 크기에 맞게 조정됩니다.
2. 맵을 잘 볼 수 있도록 사용자 지정하려면 "Magnify a selection"(선택 영역에 돋보기 사용) 또는 "확대/축소" 도구를 사용합니다.

맵 노드에서 관련 **Investigator** 트리 노드로 이동

SOA 종속성 맵 노드를 "물리적" 뷰에서 볼 때는 원하는 맵 노드를 클릭하여 **Investigator** 트리에서 해당 구성 요소로 이동할 수 있습니다. 새 **Investigator** 트리 노드로 이동하면 해당 **Investigator** 트리 노드를 시작 지점으로 하여 SOA 종속성 맵이 다시 그려집니다. 이렇게 하면 선택한 맵 노드 위치를 기준으로 방향을 다시 설정할 수 있습니다.

맵 노드에서 새로운 **Investigator** 트리 노드로 이동하고 맵을 다시 그리려면

1. SOA 종속성 맵이 "물리적" 뷰로 표시되었는지 확인합니다.
2. SOA 종속성 맵에서 맵 노드를 선택합니다.
3. 마우스 오른쪽 단추를 클릭하고 "트리에서 이 위치로 이동"을 선택합니다.
새 **Investigator** 트리 노드가 선택되어 SOA 종속성 맵의 새 초점이 됩니다.

서비스의 모든 작업 표시

"서비스" 콘텐츠 유형을 선택하면 SOA 종속성 맵을 확장하여 특정 서비스의 모든 작업을 표시할 수 있습니다.

서비스의 모든 작업을 표시하려면

1. Investigator 트리의 "클라이언트" 또는 "서버" 노드 아래에서 서비스에 대한 개별 작업을 선택합니다.

2. "SOA 종속성 맵" 탭을 클릭합니다.

SOA 종속성 맵에서 해당 서비스와 에이전트를 나타내는 상자 안에 해당 클라이언트 또는 서버 작업과 그 첫 번째 수준의 종속성이 표시됩니다.

3. 서비스 맵 노드를 선택하고 마우스 오른쪽 단추를 클릭한 다음 "모든 작업 표시"를 선택합니다.

해당 클라이언트 측 서비스 또는 서버 측 서비스의 모든 작업이 SOA 종속성 맵에 표시됩니다.

에이전트의 모든 서비스 표시

"에이전트" 콘텐츠 유형을 표시하도록 선택한 경우, SOA 종속성 맵을 확장하여 지정한 에이전트의 모든 서비스를 표시할 수 있습니다.

에이전트의 모든 서비스를 표시하려면

1. Investigator 트리의 "클라이언트" 또는 "서버" 노드 아래에서 개별 서비스를 선택합니다.

2. "SOA 종속성 맵" 탭을 클릭합니다.

SOA 종속성 맵에서 에이전트를 나타내는 하나 이상의 상자 안에 해당 클라이언트 또는 서버 서비스와 그 첫 번째 수준의 종속성이 표시됩니다.

3. SOA 종속성 맵에서 관련 에이전트를 선택하고 마우스 오른쪽 단추를 클릭한 다음 "모든 서비스 표시"를 클릭합니다.

SOA 종속성 맵에 해당 에이전트의 모든 서비스가 표시됩니다.

SOA 종속성 맵 이미지 저장

다음과 같은 목적을 위하여 SOA 종속성 맵 전체 또는 일부를 다양한 형식으로 저장할 수 있습니다.

- 문제 해결을 위해 동료와 정보를 공유합니다.
- 나중에 다시 돌아올 수 있도록 현재 맵을 환경의 스냅숏으로 저장합니다.

SOA 종속성 맵 공유

서비스 문제를 해결하기 위해 CA Introscope 또는 SOA 환경에 익숙하지 않은 동료와 협업해야 하는 경우가 있습니다. 예를 들어 경고에 응답 중일 때 응용 프로그램 문제를 해결하기 위해 다른 서비스를 담당하는 개발자와 함께 작업해야 할 수 있습니다. 종속성 맵을 이미지로 저장하면 서비스와 현재 메트릭 간의 관계에 대한 중요한 정보를 동료들과 공유할 수 있습니다.

또한 에이전트, 서비스 또는 작업 수준 종속성을 보여 주는 종속성 맵을 종속성의 물리적 또는 논리적 뷰를 포함하여 저장할 수도 있습니다. 서비스 종속성에 대한 그래픽 뷰를 제공하면 의사 소통이 수월해지고 조직에서 문제를 더 효율적으로 해결하는 데 도움이 됩니다.

SOA 종속성 맵을 스냅숏으로 저장

SOA 종속성 맵에는 항상 모니터링되는 서비스에서 가장 최근에 검색된 종속성이 반영됩니다. 데이터가 주기적으로 검사되고 새로 고쳐지지만 종속성 정보는 저장되지 않습니다.

대부분의 조직에서 프로덕션 환경의 응용 프로그램을 가끔씩만 배포, 수정 또는 제거하기 때문에 SOA 종속성 맵 내의 종속성 관계는 대개 상당히 안정적입니다. 그러나 경우에 따라서는 현재 환경의 이미지를 나중에 사용할 수 있도록 종속성 데이터의 "스냅숏"으로 저장하면 유용합니다.

SOA 종속성 맵 전체 또는 일부를 이미지로 저장하려면

1. SOA 종속성 맵 도구 모음에서 "이미지로 저장" 단추를 클릭하여 "이미지 내보내기" 대화 상자를 표시합니다. "이미지로 저장" 단추는 디스크 모양 아이콘으로 표시됩니다.
2. 출력 파일에 대해 다음과 같은 설정을 선택합니다.

유형

이미지의 파일 형식을 선택합니다. 종속성 맵은 JPEG, GIF, PDF, PNG 또는 SVG 파일로 저장할 수 있습니다.

파일 이름

출력 이미지 파일의 대상 경로와 파일 이름을 입력합니다.

이미지 내용

SOA 종속성 맵의 특정 부분을 이미지 파일에 저장하는 옵션을 선택합니다.

표시된 창만

이 옵션은 "현재 확대/축소 수준" 크기 옵션을 선택했을 때만 사용할 수 있습니다. 이 옵션은 현재 창에 보이는 맵 부분을 저장하려는 경우에만 선택하십시오. 예를 들어 SOA 종속성 맵에 노드가 100개 있지만 노드 10개만 보이도록 뷔를 확대한 경우 이 옵션을 선택하면 해당 노드 10개의 이미지만 만들어집니다.

Draw Grid(그리드 그리기)

이 옵션은 사용할 수 없습니다.

선택한 개체만

이 옵션은 맵 노드를 선택했을 때만 사용할 수 있습니다. 맵에서 선택한 개체만 저장하려면 이 옵션을 선택합니다. 예를 들어 클라이언트 웹 서비스 맵 노드를 선택하면 이 옵션은 해당 클라이언트 웹 서비스의 이미지만 만들합니다.

Image Characteristics(이미지 특성)

슬라이더 막대를 끌거나 1부터 100 사이의 숫자를 선택하여 이미지 품질과 파일 크기를 선택합니다. 100은 이미지 출력 시 가장 좋은 이미지 품질을 나타냅니다. 이미지 품질이 좋을수록 저장된 이미지의 파일 크기가 커집니다.

크기

다음 옵션 중 하나를 선택하여 이미지의 크기를 제어할 수 있습니다.

현재 확대/축소 수준

맵의 영역을 확대한 경우 이 옵션을 선택합니다. 확대된 영역만 이미지에 표시하려면 이 옵션을 "표시된 창만" 옵션과 함께 사용합니다.

실제 크기

전체 맵을 실제 크기로 저장하려면 이 옵션을 선택합니다. 이 옵션을 사용하면 맵이 축소되지 않습니다.

캔버스에 맞춤

전체 맵을 창에 맞춰서 저장하려면 이 옵션을 선택합니다. 이 옵션은 필요한 경우 창에 맞게 맵을 축소합니다.

Custom

이미지 너비와 높이를 픽셀 단위로 선택하려면 이 옵션을 선택합니다.

3. "확인"을 클릭하여 파일을 만듭니다.

지정한 위치에 파일이 저장됩니다.

맵 노드에서 트랜잭션 추적 시작

맵 노드에서 직접 "트랜잭션 추적" 세션을 시작할 수 있습니다. 이렇게 하면 서비스에서 관계의 상위 수준 뷰부터 해당 서비스를 거치는 실제 트랜잭션의 세부 정보 뷰까지 빠르게 탐색할 수 있습니다.

예를 들어 "SOA 성능 - 개요" 대시보드에서 빨간색 경고를 확인한 후 특정 서비스에서 문제 해결을 시도 중인 경우를 가정합니다. 조사 중인 서비스를 중요한 비즈니스 트랜잭션에서 사용 중이므로 이 서비스와 연관된 트랜잭션을 가능한 한 빠르게 보는 것이 중요합니다.

수동으로 트랜잭션 추적 세션을 시작하고 필터 조건을 입력할 필요 없이 SOA 종속성 맵에 있는 서비스에 대한 맵 노드에서 직접 새 트랜잭션 추적 세션을 시작하면 시간을 절약할 수 있습니다.

맵 노드에서 트랜잭션 추적 세션을 시작하려면

1. 맵 노드를 마우스 오른쪽 단추로 클릭하고 "트랜잭션 추적 시작"을 선택하여 트랜잭션 추적 세션을 시작합니다.

종속성 맵 노드에서 트랜잭션 추적을 시작하면 "새 트랜잭션 추적 세션" 대화 상자가 맵 노드에 기반하여 기본 필터가 자동으로 선택된 상태로 열립니다.

2. 필요에 따라 "추가"를 클릭하여 필터 목록에 기본 필터를 추가하거나 필터 조건을 수정한 다음 "추가"를 클릭합니다.
3. "필터 설정"을 클릭한 다음 "확인"을 클릭하여 "트랜잭션 추적 뷰어"를 엽니다. 트랜잭션 추적 뷰어는 선택된 맵 노드에 영향을 미치는 트랜잭션과 연관된 모든 추적을 수집하고 표시합니다.

참고: SOA 환경에서 프로세스 간 트랜잭션 추적을 사용하는 방법과 필터를 사용하는 방법에 대한 자세한 내용은 [SOA 환경에서 트랜잭션 추적 사용](#) (페이지 101)을 참조하십시오. "새 트랜잭션 추적 세션" 및 "트랜잭션 추적 뷰어" 사용에 대한 자세한 내용은 [CA APM Workstation 사용자 안내서](#)를 참조하십시오.

클러스터에 대한 SOA 종속성 맵

클러스터된 CA Introscope 환경이 있는 경우 SOA 종속성 맵에는 MOM에 보고하는 Collectors(수집기)에 대한 서비스 및 검색된 종속성이 표시됩니다. 예를 들어 Collector Enterprise Manager 가 세 개인 환경에서는 별도의 Collectors(수집기)에서 실행 중인 에이전트 또는 서비스 간의 종속성을 볼 수 있습니다. 종속성은 콘텐츠 유형 및 Investigator 트리 노드에 기반하여 응용 프로그램 서버에 대해 표시되는 방식과 동일하게 표시됩니다.

SOA 종속성 맵 데이터는 Collectors(수집기)에서만 저장됩니다. MOM은 데이터를 Collectors(수집기)에서 수신합니다. MOM은 [SOA 종속성 맵 데이터](#) (페이지 76)를 저장하지 않습니다.

SOA 종속성 맵을 표시하려면 MOM Enterprise Manager 와 클러스터의 각 Collector Enterprise Manager 모두에 다음 파일이 설치되어 있어야 합니다.

com.wily.introscope.soa.dependencymap.common_<version>.jar
com.wily.introscope.soa.dependencymap_<version>.jar

이 파일은 클러스터의 모든 Enterprise Manager 에 있는
`<EM_Home>/product/enterprisemanager/plugins` 디렉터리에 설치됩니다.
파일을 찾을 수 없으면 SOA 종속성 맵이 제대로 표시되지 않습니다.

제 5 장: SOA 환경에서 트랜잭션 추적 사용

CA APM for SOA에서는 성능, 작업 및 서비스 관련 구성 요소의 전반적인 건전성에 대한 메트릭을 수집하여 서비스 및 비즈니스 프로세스를 모니터링할 수 있습니다. 대시보드 또는 특정 메트릭이 서비스에 문제가 있을 가능성을 제시하는 경우 트랜잭션 추적을 사용하여 실제 비즈니스 트랜잭션에 대한 세부 정보를 확인하면 문제의 원인을 찾고 해결하는 데 도움이 됩니다.

이 섹션은 다음 항목을 포함하고 있습니다.

- [크로스 프로세스 트랜잭션 추적 정보](#) (페이지 101)
- [트랜잭션 추적을 사용하여 문제 해결](#) (페이지 104)
- [트랜잭션 추적 시작 및 보기](#) (페이지 105)
- [트랜잭션 추적에 대한 필터 설정](#) (페이지 112)
- [이벤트 데이터베이스에서 SOA 서비스 쿼리](#) (페이지 115)

크로스 프로세스 트랜잭션 추적 정보

SOA 환경에서는 트랜잭션이 여러 JVM(Java Virtual Machine) 또는 CLR(Common Language Runtime) 인스턴스에서 실행되는 경우가 일반적입니다. 처리는 한 서비스에서 다른 서비스로 이동합니다. 따라서 트랜잭션의 전체 경로를 보려면 JVM 및 CLR 경계를 지나 동기적 및 비동기적 호출을 추적할 수 있어야 합니다. 또한 Java 또는 .NET 에이전트를 실행 중인 여러 플랫폼에서 트랜잭션을 추적할 수 있어야 할 수도 있습니다.

CA APM for SOA 를 활성화하면 HTTP, HTTPS 또는 JMS 전송 프로토콜을 사용하고 지원되는 플랫폼에서 실행되는 모든 비즈니스 트랜잭션의 전체 경로를 추적할 수 있습니다. 전체 트랜잭션은 많은 다른 플랫폼에서 처리되는 세그먼트를 포함할 수 있습니다. 예를 들어 WebLogic에서 실행되는 서비스를 사용한 다음 SAP NetWeaver 또는 .NET 서버에서 실행되는 서비스로 전달되는 트랜잭션을 추적할 수 있습니다. 활성화하는 추가 SOA 확장에 따라 트랜잭션은 다음 구성 요소와 관련된 세그먼트를 포함할 수도 있습니다.

- Oracle Service Bus
- WebSphere Process Server
- WebSphere Enterprise Service Bus
- TIBCO BusinessWorks
- webMethods Integration Server
- WebSphere MQ
- Spring 웹 서비스

에이전트의 Spring 웹 서비스 지원은 클라이언트 및 서버 모두를 위한 웹 서비스 내부 메트릭 및 웹 서비스 전송 메트릭을 제공합니다.

추적되는 모든 노드에 CA APM for SOA 가 사용하도록 설정되어 있다면 비즈니스 트랜잭션에 플랫폼의 모든 조합이 포함될 수 있습니다. 에이전트에 대해 CA APM for SOA 를 사용하도록 설정하면 여러 노드에서 실행 중인 여러 JVM 또는 CLR 인스턴스에서 서비스를 호출하는 트랜잭션의 세부 정보를 드릴다운 할 수 있습니다.

CA APM for SOA 를 사용하도록 설정한 후에는 추가 구성이 필요 없습니다.

참고: 지원에 대한 자세한 내용은 *Compatibility Guide*(호환성 안내서)의 "SOA Performance Management"(SOA 성능 관리) 단원을 참조하십시오.

트랜잭션의 세그먼트가 연결되는 방식 이해

일반적으로 트랜잭션은 프로세스 간에 전달되는 일련의 호출과 응답으로 이루어집니다. 트랜잭션의 각 세그먼트는 서로 다른 논리적 서버나 물리적 서버에서 실행되거나, 서로 다른 구성요소나 백엔드 시스템에 분산되어 있는 경우가 많습니다.

따라서 전체 트랜잭션을 조합하려면 어떤 처리 세그먼트들이 같은 트랜잭션에 속하는지, 그리고 트랜잭션의 특정 프로세스가 다른 프로세스를 호출하는 시기가 언제인지 파악해야 합니다. 에이전트는 서로 다른 JVM이나 CRL 인스턴스에서 실행되는 프로세스를 호출하는 트랜잭션의 전체 경로를 추적하기 위해 트랜잭션에 상관 관계 식별자를 추가합니다. 상관 관계 식별자는 같은 트랜잭션에 속하는 세그먼트를 식별하기 위해 한 프로세스에서 다른 프로세스로 전달될 수 있습니다.

상관 관계 식별자는 같은 트랜잭션에 속하는 구성 요소를 식별하는 것 외에도 트랜잭션의 서로 다른 부분이 호출되는 순서를 추적하기 위한 시퀀스 정보를 제공합니다. 시퀀스 정보를 사용하면 트랜잭션 세그먼트가 호출되는 순서를 알 수 있습니다. 동기식 트랜잭션의 경우에는 이 순서를 통해 트랜잭션 세그먼트 간의 호출자/피호출자 관계를 파악할 수 있습니다. 비동기식 트랜잭션의 경우에는 이 순서를 통해 복잡한 클라이언트 및 서버 트랜잭션 세그먼트의 여러 프로세스 간에 처리 작업 흐름을 식별할 수 있습니다.

상관 관계 식별자 데이터 집합은 에이전트에 의해 자동으로 관리되고 Enterprise Manager에 전달됩니다. 그런 다음 이 정보는 "트랜잭션 추적 뷰어"에 표시되는 선택한 트랜잭션의 그래픽 표현을 형성하는 데 사용됩니다.

크로스 프로세스 트랜잭션 추적의 컨텍스트 이해

트랜잭션 추적 세션은 서비스 네임스페이스 또는 작업 이름을 기반으로 시작할 수 있습니다. 대부분의 경우 네임스페이스는 문제가 발생한 서비스를 식별하는 데 도움이 되는 적절한 정보를 제공합니다. 문제의 원인일 수 있는 특정 작업을 식별했다면 그 작업에 대해 트랜잭션 추적을 시작하고자 할 수 있습니다.

그러나 작업 이름을 사용하는 경우 서비스 네임스페이스가 작업 컨텍스트를 제공한다는 점을 기억하십시오. 예를 들어 항공편 좌석을 예약하거나, 식당의 테이블을 요청하거나, 진료 예약을 하는 응용 프로그램에서 *makeReservation*이라는 작업 이름을 사용할 수 있습니다. *makeReservation* 작업에 대한 트랜잭션 추적을 시작하려면 해당 작업이 사용된 네임스페이스를 알아야 합니다.

작업의 네임스페이스 정보는 특정 추적을 선택하면 "추적 뷰" 탭의 "구성 요소 정보"에 표시됩니다. 예를 들어 "추적 뷰" 탭에서 *makeReservation* 작업을 선택하면 "구성 요소 정보"를 확인하여 해당 작업과 관련된 네임스페이스를 봄으로써 작업이 여행사 웹 서비스의 일부인지 아니면 호텔 예약 서비스의 일부인지 알 수 있습니다.

트랜잭션 추적을 사용하여 문제 해결

다음 예제를 보면 크로스 프로세스 트랜잭션 추적이 문제를 빠르고 효과적으로 식별하고 평가하는 데 어떻게 도움이 되는지 알 수 있습니다. 트랜잭션 추적 세션을 실행한 후 응용 프로그램 지원 담당자가 실행 시간이 6 초(6000 밀리초)인 트랜잭션을 발견합니다.

해당 트랜잭션의 "추적 뷰"를 검토한 결과 트랜잭션에 클라이언트 측 웹 서비스인 dataservice.yourcompany.net/Invoke에서 서버 측 웹 서비스인 cics.mycompany.net/Invoke로의 호출이 포함되어 있고, 서버 측 웹 서비스가 CICS 메인프레임을 여러 번 호출한다는 사실이 확인되었습니다.

CICS 처리 시간은 트랜잭션의 해당 부분이 측정되지 않기 때문에 추적에 명시적으로 표시되지 않지만, "추적 뷰"에는 CICS 백엔드가 반복적인 요청을 연속적으로 빠르게 처리하는 것으로 표시됩니다. 이러한 트랜잭션 추적을 통해 응용 프로그램 지원 전문가는 서버 측 서비스에서 중첩된 루프 같은 프로그래밍 논리로 인하여 이와 같이 반복적인 호출이 발생했고 해당 서비스의 호출작업이 트랜잭션의 전체 실행 시간 대부분을 차지한다는 것을 알 수 있습니다. 이러한 정보를 바탕으로 응용 프로그램 지원 전문가는 바로 서버 측 웹 서비스의 개발자나 소유자에게 CICS 백엔드를 호출하는 응용 프로그램 논리에 대한 자세한 조사와 요청할 수 있습니다.

트랜잭션 추적 시작 및 보기

로컬 또는 원격 컴퓨터의 여러 JVM 또는 CLR 인스턴스에 대한 호출이 포함된 비즈니스 트랜잭션이 있는 경우 프로세스 간 트랜잭션 추적을 사용하여 웹 서비스의 문제를 식별하고 해결할 수 있습니다. 트랜잭션에 참여하는 CLR 또는 JVM의 에이전트는 클러스터의 일부가 아닐 경우 동일한 Enterprise Manager에 정보를 보고해야 하며 그렇지 않으면 상호 관련 추적을 볼 수 없습니다. 에이전트가 동일한 Enterprise Manager로 정보를 보고하는 경우 트랜잭션 추적 세션을 시작하여 전체 비즈니스 프로세스에 대한 정보를 캡처할 수 있습니다. 클러스터 환경에서는 Workstation이 MOM에 연결되어 있으면 프로세스 간 트랜잭션에 참여하는 에이전트는 상호 관련된 모든 수집기 및 추적에 연결할 수 있습니다.

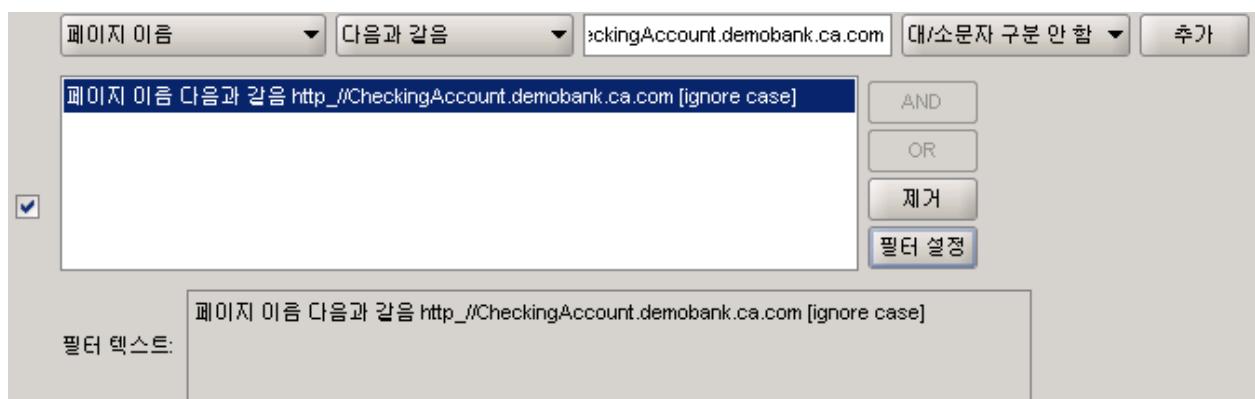
트랜잭션 추적 세션은 다음과 같이 시작할 수 있습니다.

- SOA 종속성 맵의 맵 노드에서 바로 시작합니다.
- "Workstation" > "새 트랜잭션 추적 세션"을 클릭하여 Workstation에서 수동으로 시작합니다.

다음 단계를 수행하십시오.

1. SOA 종속성 맵에서 맵 노드를 마우스 오른쪽 단추로 클릭하거나 "Workstation" > "새 트랜잭션 추적 세션"을 클릭하여 "새 트랜잭션 추적 세션" 대화 상자를 표시합니다.
2. 네 번째 확인란을 클릭하여 SOA 관련 필터 옵션을 활성화합니다.

다음 예제에서는 SOA 필터 및 네임스페이스 필터 정의를 활성화하기 위해 선택할 확인란을 보여 줍니다.



3. SOA 관련 필터링 옵션 목록에서 사용할 트랜잭션 추적 필터의 유형을 선택한 다음 "추가"를 클릭합니다.

CA APM for SOA 에는 서비스 네임스페이스 또는 특정 작업 이름에 기반하는 트랜잭션을 선택할 수 있는 SOA 특정 트랜잭션 추적 필터가 포함되어 있습니다. 새 트랜잭션 추적을 시작하기 전에 선택한 노드에 따라 기본 필터 및 값이 자동으로 설정됩니다.

지정한 기준과 일치하는 모든 웹 서비스 또는 웹 서비스 작업에 관여된 트랜잭션을 추적하기 위한 네임스페이스 또는 작업 이름 필터를 선택할 수 있습니다. 이 기준에는 TIBCO BusinessWorks 또는 webMethods Integration Server 등의 SOA 플랫폼에서 제공하는 웹 서비스가 포함될 수 있습니다.

4. "필터 설정"을 클릭합니다.
5. "Run session for N minutes"(N 분 동안 세션 실행) 필드에 트랜잭션 추적 세션을 실행할 시간을 분 단위로 입력합니다.
6. "에이전트 추적" 섹션에서 트랜잭션을 추적할 모든 에이전트 또는 특정 에이전트를 선택합니다.
7. "확인"을 클릭합니다.

트랜잭션 추적 세션이 시작됩니다.

참고: Workstation에서 트랜잭션 추적을 시작하는 방법에 대한 자세한 내용은 CA APM Workstation 사용자 안내서를 참조하십시오.

추가 정보:

[맵 노드에서 트랜잭션 추적 시작 \(페이지 98\)](#)

요약 뷰 사용

트랜잭션 추적 세션을 시작하면 트랜잭션 추적 결과가 "트랜잭션 추적 뷰어"의 위쪽에 표시됩니다. 목록에는 현재 추적 중인 네임스페이스 또는 작업에 대한 모든 트랜잭션이 표시됩니다. "기간" 열 머리글을 클릭하면 트랜잭션을 기간별로 정렬할 수 있습니다.

뷰어 위쪽에서 트랜잭션을 선택하면 다른 JVM 또는 CLR에 대한 호출을 포함하여 트랜잭션을 구성하는 모든 호출이 "요약 뷰" 탭에 표시됩니다.

트랜잭션을 선택하면 "요약 뷰"에는 호출 수, 밀리초 단위 호출 길이, 최소, 평균 및 최대 호출 시간을 포함하여 선택된 트랜잭션에서 호출된 서비스와 작업이 나열됩니다.

참고: 트랜잭션 추적에 "요약 뷰"를 사용하는 방법에 대한 자세한 내용은 *CA APM Workstation 사용자 안내서*를 참조하십시오.

추적 뷰 사용

트랜잭션에 대한 세부 정보를 그래픽 형식으로 보려면 "추적 뷰" 탭을 클릭하십시오.

"트랜잭션 추적 뷰"에는 트랜잭션의 각 구성 요소가 막대로 표시되며 총 트랜잭션 실행 시간 중 각 구성 요소가 차지하는 시간이 표시됩니다. 동기 트랜잭션의 경우 "추적 뷰" 탭에는 현재 보고 있는 트랜잭션의 세그먼트에 관여한 각 노드의 구성 요소 간의 호출 관계도 표시됩니다.

트랜잭션에 둘 이상 JVM의 처리가 포함된 경우에는 각 JVM에 대한 추적이 별도 행에 표시됩니다. SOA 트랜잭션의 경우 여러 JVM에서 처리되는 것이 일반적입니다.

그래프에서 트랜잭션의 구성 요소를 선택하면 구성 요소에 대한 추가적인 세부 정보가 그래프 아래의 "구성 요소 세부 정보"에 표시됩니다.

참고: "추적 뷰" 사용에 대한 자세한 내용이나 트랜잭션 추적에서 "구성 요소 세부 정보"에 표시되는 정보에 대한 자세한 내용은 *CA APM Workstation 사용자 안내서*를 참조하십시오.

트리 뷰 사용

선택한 트랜잭션에 대한 구성 요소의 계층적 뷰를 보려면 "트리 뷰" 탭을 클릭하십시오.

"트리 뷰"에는 트랜잭션의 구성 요소가 계층 구조로 표시되며, 트랜잭션의 전체 실행 시간 중 구성 요소가 차지하는 시간에 따라 빨간색, 노란색 또는 초록색 표시기가 표시됩니다.

구성 요소를 선택하고 확장하여 세부 정보를 볼 수 있습니다. 구성 요소를 선택하면 탭의 "구성 요소 세부 정보" 창에 추가 정보가 표시됩니다.

참고: "트리 뷰" 사용에 대한 자세한 내용이나 트랜잭션 추적에서 "구성 요소 세부 정보"에 표시되는 정보에 대한 자세한 내용은 *CA APM Workstation 사용자 안내서*를 참조하십시오.

시퀀스 뷰 사용

SOA 환경의 분산 응용 프로그램은 복잡한 성향이 있기 때문에 사용자 트랜잭션 하나가 개별 에이전트 JVM 또는 CLR에서 실행되는 여러 스레드와 관련되고 동기식 호출과 비동기식 호출을 모두 포함하는 경우가 많습니다. 트랜잭션의 전체 경로를 논리적 단위로 나타내려면 이러한 개별 트랜잭션 세그먼트를 고려해야 합니다. "시퀀스 뷰" 탭에는 호출의 시퀀스를 시각적으로 볼 수 있도록 세그먼트 사이의 호출자-피호출자 관계가 표시됩니다.

"시퀀스 뷰"는 특히 다음과 같은 특징을 가진 트랜잭션에 유용합니다.

- 비동기식 호출을 포함
- 서로 시간이 동기화되지 않은 에이전트에서 실행되는 프로세스에 대한 호출을 포함
- 여러 JVM 또는 CLR 사이의 복잡한 동기식 호출을 포함
- 다른 에이전트의 트랜잭션 추적 세션에서 호출된 다운스트림 에이전트에 대해 자동으로 시작된 추적을 포함

선택한 트랜잭션의 프로세스 호출 순서를 보려면 "시퀀스 뷰" 탭을 클릭하십시오. 이 탭에서 화살표는 호출된 프로세스를 가리키고 가로 막대형 차트에는 기간을 기준으로 정렬된 프로세스가 나열됩니다. 추적이 트랜잭션의 일부로 확인되었지만 올바른 위치를 확인할 수 없는 경우, 추적이 점선으로 표시되어 시퀀스에 추가됩니다. 그래프나 가로 막대형 차트에서 프로세스를 선택하면 해당 프로세스의 세부 정보가 오른쪽 패널에 표시됩니다.

속도가 가장 느린 프로세스는 가로 막대형 차트에서의 순위와 느린 프로세스 아이콘으로 구분됩니다. 프로세스 실행 시 오류가 발생한 경우 해당 프로세스에 오류 아이콘이 표시됩니다.

프로세스 노드 레이블 지정 방법 선택

"시퀀스 뷰"에 나열되는 각 프로세스는 작업을 완료하는 데 관련되는 프로세스의 구성 요소 추적을 나타냅니다. 추적이 같은 서버나 에이전트에서 이루어지면 해당 프로세스에 대해 표시되는 레이블을 식별하기가 어렵거나 아예 불가능할 수 있습니다.

기본적으로 프로세스 레이블은 작업 이름을 포함하는 구성 요소, EJB 문자열을 포함하는 구성 요소 또는 에이전트 이름 앞에 붙는 첫 번째 고유한 구성 요소 리소스 이름을 사용하여 자동으로 선택됩니다. 기본적으로 선택되는 레이블을 사용하지 않으려면 첫 번째 구성 요소 이름, 마지막 구성 요소 이름, 스퍼드 이름 또는 설명을 사용하도록 프로세스 레이블을 변경할 수 있습니다.

마지막 구성 요소 이름을 프로세스 레이블로 선택하면 호출 스택에서 모호하지 않은 마지막 이름이 사용됩니다. 이 이름은 호출 시퀀스 및 동기식 호출 또는 비동기식 호출의 존재 여부에 따라 프로세스의 마지막 구성 요소일 수도 있고 그렇지 않을 수도 있습니다.

다른 프로세스 레이블 지정 체계를 선택하려면

- "시퀀스 뷰" 탭에서 "프로세스 레이블" 옵션을 선택합니다.

기간 계산 방법 선택

"시퀀스 뷰" 탭에서 추적 기간을 계산하는 방법에는 두 가지가 있습니다.

- 전체 기간**은 프로세스의 시작 시간과 종료 시간을 사용하여 계산됩니다. 이 방법은 "추적 뷰" 탭에 기간이 표시되는 방법과 유사합니다. 이 방법을 사용하면 동기 트랜잭션에서 루트 추적이 항상 가장 느립니다. 이 방법은 기본 계산 방법입니다.
- 순 기간**은 추적의 전체 기간에서 추적의 비동기 자식의 기간 시간을 제하여 계산합니다. 이 방법의 주요한 장점은 동기식 트랜잭션에서 가장 느린 프로세스를 손쉽게 확인할 수 있다는 점입니다. 전체 기간에서 순 기간으로 전환하면 기간 등급 및 기간에 따라 나열되는 프로세스의 순서가 실행 시간이 계산되는 방법에 따라 달라집니다.

기간 계산 방법을 변경하려면

- "시퀀스 뷰" 탭에서 "기간 유형"으로 "전체" 또는 "순"을 선택합니다.

시퀀스 뷰와 추적 뷰 간의 탐색

"시퀀스 뷰"와 "추적 뷔" 간을 이동하면서 탐색하면 편리한 경우가 많습니다. 마우스 오른쪽 단추 메뉴 옵션을 사용하면 특정 프로세스나 구성 요소의 컨텍스트를 유지하면서 이러한 뷔 간을 더 빠르게 이동할 수 있습니다.

"시퀀스 뷔"의 프로세스에서 "추적 뷔"로 이동하려면

1. 그래프에서 프로세스를 선택합니다.
2. 마우스 오른쪽 단추를 클릭하고 "추적 뷔에서 스레드 선택"을 클릭합니다.

"추적 뷔"의 구성 요소에서 "시퀀스 뷔"로 이동하려면

1. 그래프에서 구성 요소를 선택합니다.
2. 마우스 오른쪽 단추를 클릭하고 "시퀀스 뷔에서 스레드 선택"을 클릭합니다.

트랜잭션의 프로세스에 대한 세부 정보 보기

그래프나 가로 막대형 차트에서 프로세스를 선택하면 오른쪽 패널에서 해당 프로세스에 대한 세부 정보를 볼 수 있습니다. 선택한 프로세스에 따라 다음과 같은 정보의 전체 또는 일부가 표시됩니다.

스레드 이름

호출 스택에서 선택한 스레드의 이름입니다.

Thread Group Name(스레드 그룹 이름)

스레드 그룹의 이름입니다.

응용 프로그램 이름

응용 프로그램의 표시 이름입니다.

클래스

계측된 클래스입니다.

Method

계측된 메서드입니다.

Method Descriptor(메서드 설명자)

계측된 메서드 인수 또는 서명입니다.

리소스 이름

응용 프로그램 리소스의 유형입니다. 예를 들어 리소스 유형이 서블릿 또는 EJB 를 나타낼 수 있습니다.

URL

응용 프로그램 프런트엔드를 식별하는 데 사용되는 전체 URL(Uniform Resource Locator)입니다.

Normalized URL(정규화된 URL)

에이전트 프로필에 응용 프로그램 프런트엔드를 식별하는 방법으로 정의된 URL 의 정규화된 부분입니다.

Referring URL(참조 URL)

참조하는 페이지의 URL입니다.

서버 이름

프로세스가 실행되는 서버의 이름입니다.

서버 포트

프로세스가 실행되는 포트 번호입니다.

스키마

사용되는 URI(Uniform Resource Identifier)의 유형입니다.

URI 스키마는 URI 명명 구조의 최상위 수준이며, 스키마 이름 다음에 콜론(:) 문자가 오는 형식입니다.

예를 들어 HTTP(Hypertext Transfer Protocol)을 사용하여 액세스하는 웹 서비스의 URI 는 http 가 됩니다.

HTTP 메서드

사용되는 HTTP 메서드 유형입니다. 예를 들어 가장 일반적인 HTTP 메서드는 POST 와 GET 입니다.

CrossProcessData

같은 트랜잭션에 속하는 프로세스를 식별하는 데 사용되는 고유한 상관 관계 식별자입니다.

Trace Type(추적 유형)

프로세스의 트랜잭션 추적 유형입니다. 추적 유형은 프로세스가 트랜잭션 추적 세션의 일부인지 아니면 샘플 트랜잭션에 수집되는지를 나타냅니다.

트랜잭션 추적에 대한 필터 설정

CA APM for SOA 를 사용하도록 설정하면 관심 있는 트랜잭션을 빠르게
격리하는 데 도움이 되는 네임스페이스 필터 및 작업 이름 필터가
Workstation 에 추가됩니다. 예를 들어 특정 서비스가 중요한 비즈니스
트랜잭션에 사용되는 경우 네임스페이스 필터를 사용하면 트랜잭션에 대한
모든 들어오는 트랜잭션을 해당 특정 네임스페이스로 필터링 할 수
있습니다. 문제가 발생한 네임스페이스를 식별할 때 작업 이름 필터를
사용해서 하나 이상의 특정 작업에 대한 트랜잭션을 필터링하여 문제를
심사할 수 있습니다.

또한 네임스페이스 및 작업 이름 필터를 다른 필터와 함께 사용하여
반환되는 트랜잭션 결과를 더 세밀하게 제어할 수 있습니다. 예를 들어
네임스페이스 필터를 "사용자 ID" 또는 "오류" 필터와 함께 사용하여 두
조건을 모두 충족하는 트랜잭션만 캡처할 수 있습니다.

다음 단계를 수행하십시오.

1. "Workstation" > "새 트랜잭션 추적 세션"을 클릭합니다.
사용할 수 있는 필터는 설치한 구성 요소에 따라 달라집니다.
 2. 특정 필터를 활성화하려면 해당 확인란을 클릭하십시오.
예를 들어 "Lasting longer than"(다음보다 오래 지속됨) 확인란을
클릭하여 지정된 시간보다 오래 지속된 트랜잭션을 찾습니다. "사용자
ID"나 "세션 ID"와 같은 특정 매개 변수 필터를 사용하여 지정한
조건으로 트랜잭션을 제한할 수 있습니다.
 3. CA APM for SOA 및 기타 SOA 플랫폼에 대한 필터를 활성화하려면 해당
확인란을 클릭합니다.
 4. 필터 유형과 필터 조건을 선택하고, 값을 입력하고, 대/소문자 구분
여부를 선택합니다. "추가"를 클릭하여 이 필터를 사용 가능한 필터
목록에 추가합니다.
- 네임스페이스 및 작업 이름 옵션은 웹 서비스에 대한 표준 필터입니다.
이 필터는 Oracle Service Bus, TIBCO BusinessWorks 또는 webMethods
Integration Server 와 같은 SOA 플랫폼을 통해 제공되는 웹 서비스에도
적용됩니다.

하지만 다른 SOA 플랫폼을 모니터링하는 경우에는 다음의 추가 필터 유형을 사용하여 트랜잭션을 캡처할 수 있습니다.

어댑터 노드

WebSphere Process Server 아웃바운드 어댑터

비즈니스 프로세스

TIBCO BusinessWorks 프로세스 인스턴스

webMethods Integration Server 비즈니스 서비스

WebSphere Process Server 비즈니스 프로세스

비즈니스 서비스

Oracle Service Bus 비즈니스 서비스

비즈니스 상태 컴퓨터

WebSphere Process Server 비즈니스 상태 컴퓨터

조정 흐름

WebSphere Enterprise Service Bus 조정 흐름

조정 흐름 작업

WebSphere Enterprise Service Bus 조정 흐름 작업

메시지 서버

IBM WebSphere MQ 메시지 서버

프록시 서비스

Oracle Service Bus 프록시 서비스

네임스페이스 또는 작업 이름을 수동으로 입력할 때는 이름의 철자를 정확하게 입력해야 필터가 일치하는 트랜잭션을 찾을 수 있습니다.

5. 필요한 경우 필터 목록에 필터를 추가하고 필터를 조합하여 [복합 필터를 구성](#) (페이지 115)합니다. 예를 들어 네임스페이스에 대한 필터를 추가한 다음 작업 이름에 대한 다른 필터를 추가할 수 있습니다. 정의된 필터 목록에서 두 필터를 모두 선택한 다음 "AND"를 클릭하면 지정한 네임스페이스와 작업 이름을 모두 사용하여 트랜잭션이 필터링됩니다.
6. 이 세션에 사용할 필터를 선택한 다음 "필터 설정"을 클릭합니다. 이전에 설정된 필터는 모두 교체됩니다. 한 번에 하나의 필터만 적용됩니다.

7. "Run session for N minutes"(N 분 동안 세션 실행) 필드에 트랜잭션 추적 세션을 실행할 시간을 분 단위로 입력합니다.
8. "에이전트 추적" 섹션에서 트랜잭션을 추적할 모든 에이전트 또는 특정 에이전트를 선택합니다.
9. "확인"을 클릭합니다.
"트랜잭션 추적" 세션이 시작됩니다.

필터 추가 및 저장

필터를 정의하고 "추가"를 클릭하여 필터를 원하는 만큼 추가할 수 있습니다. "추가"를 클릭하면 사용할 수 있는 필터 목록에 새 필터가 추가됩니다. 중복되는 필터는 사용할 수 없습니다.

모든 트랜잭션 추적 세션에서 사용할 필터를 지정하려면 먼저 사용 가능한 필터 목록에서 필터 정의를 선택한 후 "필터 설정"을 클릭해야 합니다.

"확인"을 클릭하여 트랜잭션 추적 세션을 시작하면 정의한 필터가 다음에 새 트랜잭션 추적 세션을 시작할 때 사용할 수 있도록 저장됩니다. "확인"을 클릭하면 사용자별로 필터가 자동 저장됩니다. 다음에 **Workstation**에 로그온하면 이전에 정의한 사용자 고유의 필터를 사용하여 추적에 사용할 필터를 선택할 수 있습니다.

필터 제거

사용 가능한 필터 목록에서 필터를 선택한 후 "제거"를 클릭하여 필터를 제거할 수 있습니다. "제거"를 클릭하면 선택한 필터가 "새 트랜잭션 추적 세션" 대화 상자의 사용 가능한 필터 목록에서 제거됩니다. 여기서 "확인"을 클릭해야만 선택한 필터가 영구적으로 제거됩니다. "확인"을 클릭하지 않으면 제거한 필터가 다음에 "새 트랜잭션 추적 세션"을 시작할 때 사용 가능한 필터 목록에 복원됩니다.

다른 필터의 구성 요소로 사용되지 않는 필터만 제거할 수 있습니다. 예를 들어 *namespace filter1* 과 *namespace filter2* 를 결합하여 *namespace filter1 AND namespace filter2* 라는 필터를 만든 경우, *namespace filter1* 을 삭제하려면 먼저 AND 필터를 삭제해야만 합니다.

복합 필터 사용

AND 또는 OR 연산자를 사용하여 필터 목록에서 하나 이상의 필터를 결합하는 방법으로 복합 필터를 만들 수 있습니다. 정의된 필터 목록에 이미 있는 필터만 결합할 수 있습니다. AND 또는 OR 연산자를 사용하여 간단한 필터 문 두 개를 조인하면 새 필터가 만들어지는데, 이 필터는 단독 필터로 사용하거나 다른 필터를 만들기 위한 구성 단위로 사용할 수 있습니다.

AND 연산자를 사용하면 추적이 필터의 모든 구성 요소를 충족해야 함을 의미합니다. 예를 들어 *namespace contains demobank AND operationname starts with process*라는 필터의 경우 들어오는 추적이 두 필터 조건을 모두 충족해야 합니다.

OR 연산자를 사용하면 추적이 필터 구성 요소 중 적어도 하나를 충족해야 함을 의미합니다. 예를 들어 *namespace contains demobank OR namespace contains creditcheck*라는 필터의 경우 들어오는 추적이 첫 번째 필터 조건, 두 번째 필터 조건 또는 두 필터 조건 모두를 충족해야 합니다.

이벤트 데이터베이스에서 SOA 서비스 쿼리

"트랜잭션 추적 뷰어"는 주로 트랜잭션 추적 세션의 라이브 데이터 또는 최신 데이터를 보는 데 사용됩니다. 트랜잭션 추적 세션의 결과는 트랜잭션 이벤트 데이터베이스에 자동으로 저장되며 "기록 쿼리 뷰어"를 사용하여 볼 수 있습니다. 나중에 정보에 액세스해야 할 경우에는 트랜잭션 이벤트 데이터베이스를 쿼리하여 SOA 관련 서비스와 연관된 모든 비즈니스 트랜잭션이나 오류에 대한 정보를 표시할 수 있습니다.

웹 서비스와 관련된 모든 트랜잭션을 데이터베이스에서 쿼리하려면

1. Workstation 을 열어 둔 상태에서 "Workstation" > "기록 이벤트 쿼리"를 선택합니다.
2. "쿼리" 필드에 적절한 정보를 검색하기 위한 쿼리 정보를 입력합니다.
예:
 - 웹 서비스가 포함된 모든 이벤트를 쿼리하려면 *webservices* 를 입력합니다.
 - 특정 웹 서비스 끝점과 관련된 모든 이벤트를 쿼리하려면 원하는 URL 을 입력합니다.
3. 필요한 경우 쿼리 시간 범위를 선택합니다.

4. "실행"을 클릭하여 웹 서비스 정보가 포함된 모든 비즈니스 트랜잭션이나 오류를 표시합니다.
5. 테이블에서 원하는 행을 클릭하여 아래쪽 창에서 해당 세부 정보를 봅니다.
예를 들어 선택한 행이 트랜잭션 추적(T)인 경우 "요약 뷰", "추적 뷰", "트리 뷰" 및 "시퀀스 뷰" 탭을 클릭하여 "트랜잭션 추적 뷰어"에서와 마찬가지로 추가 정보를 볼 수 있습니다.

오류 이벤트 정보 보기

기록 이벤트가 오류(E)인 경우 아래쪽 창에 "스택 뷰" 탭이 표시됩니다. 이벤트가 트랜잭션인 경우에는 아랫쪽 창에 "요약 뷰", "추적 뷰", "트리 뷰" 및 "시퀀스 뷰" 탭이 표시됩니다. 표시되는 정보를 탐색하여 웹 서비스의 어느 구성 요소에서 문제가 발생했는지 확인할 수 있습니다.

웹 서비스와 관련된 모든 오류를 데이터베이스에서 쿼리하려면

1. "Workstation" > "기록 이벤트 쿼리"를 선택합니다.
2. "쿼리" 필드에 다음을 입력합니다.
`type:errorsnapshot AND webservices`
3. 테이블에서 행을 클릭하여 "스택 뷰" 탭에서 오류 세부 정보를 봅니다.
 - 서버 측 오류에 대해서는 빨간색 오류 메시지에서 오류의 근본 원인을 볼 수 있습니다.
 - 클라이언트 측과 서버 측에서는 SOAT 오류 예외와 함께 근본 원인을 볼 수 있습니다.
 - SOAP 오류 메시지에는 빨간색 오류 메시지 앞에 **SOAP** 오류 접두사가 붙습니다.

상호 관련된 이벤트 정보 보기

트랜잭션 세부 정보를 검토하는 경우 같은 트랜잭션에 속하지만 트랜잭션 중에 호출되는 다른 프로세스나 다른 노드에서 실행되는 이벤트를 확인할 수 있습니다.

트랜잭션 추적의 상호 관련된 이벤트를 보려면

1. "트랜잭션 추적 뷰어" 또는 "기록 쿼리 뷰어"에 표시된 목록에서 트랜잭션 추적을 선택합니다.
2. "추적" > "상호 관련된 이벤트"를 선택하여 선택한 트랜잭션 추적의 일부이지만 다른 프로세스 또는 노드에 있는 추적 목록을 표시합니다.
그런 다음 원하는 상호 관련 추적을 선택하여 해당 추적에 대한 "요약 뷰", "추적 뷰", "트리 뷰" 또는 "시퀀스 뷰"를 표시할 수 있습니다.

제 6 장: SOA 특정 속성 구성

이 단원에서는 구성 속성을 사용하여 CA APM for SOA 를 사용자 지정하는 방법을 설명합니다.

이 섹션은 다음 항목을 포함하고 있습니다.

[웹 서비스의 표시 이름 구성](#) (페이지 119)

[상호 관련된 추적 구성](#) (페이지 121)

[SOAP 핸들러의 삽입 지점 구성](#) (페이지 124)

[SOA 종속성 맵의 제한 구성](#) (페이지 125)

웹 서비스의 표시 이름 구성

기본적으로 웹 서비스 노드는 웹 서비스 네임스페이스를 사용하여 이름이 지정됩니다 웹 서비스 네임스페이스는 해당 URL 과 유사합니다. 예를 들어 웹 서비스가 다음 URL 을 사용하는 경우

`http://ClearingHouse.demobank.ca.com`

해당 노드는 기본적으로 다음과 같이 표시됩니다.

`http://ClearingHouse.demobank.ca.com`

선택적으로 웹 서비스 끝점을 노드 이름으로 사용하도록 에이전트를 구성할 수 있습니다. 웹 서비스 끝점에는 서비스의 포트 번호와 서버 이름 같은 추가 정보가 포함됩니다. 예를 들어 `ClearingHouse.demobank.ca.com` 웹 서비스의 웹 서비스 끝점을 표시하도록 선택하면 Investigator 에서 해당 노드가 다음과 같이 표시될 수 있습니다.

`http://localhost:8383/demobank_services/``ClearingHouseService`

에이전트의 `webservices.pbd` 파일을 편집하고 `{namespace}` 또는 `{servicename}`을 사용할지 여부를 지정하여 표시되는 이름을 변경할 수 있습니다. 대부분의 경우 네임스페이스는 Investigator 와 SOA 종속성 맵에 표시되는 가장 알기 쉬운 이름입니다. 그러나 `webservices.pbd` 파일과 `IntroscopeAgent.profile` 파일을 편집하여 서비스 끝점을 사용할 수도 있습니다.

Investigator 및 종속성 맵에서 서비스 끝점을 노드 이름으로 사용하려면

1. <Agent_Home> 디렉터리에서 *webservices.pbd* 파일을 엽니다.
2. *{namespace}*의 모든 인스턴스를 찾아 *{servicename}*으로 바꿉니다.
{namespace} 문자열은 추적 프로그램의 정규화된 메트릭 이름의 일부로 포함되기 때문에 파일에서 여러 번 사용됩니다.
3. *webservices.pbd* 파일을 저장합니다.
4. <Agent_Home> 디렉터리에서 *appmap-soa.pbd* 파일을 엽니다.
5. *{namespace}*의 모든 인스턴스를 찾아 *{servicename}*으로 바꿉니다.
6. *appmap-soa.pbd* 파일을 저장합니다.
7. 응용 프로그램 서버를 다시 시작합니다.

응용 프로그램 서버를 다시 시작하면 **Investigator** 트리와 SOA 종속성 맵 네임스페이스 대신 웹 서비스 끝점 이름이 표시됩니다.

네임스페이스 대신 서비스 끝점 이름을 사용하는 경우 클라이언트 및 서버 "개요" 탭에 표시되는 네임스페이스 레이블도 변경할 수 있습니다.

클라이언트 및 서버 "개요" 탭에 사용되는 레이블을 변경하려면

1. <EM_Home>/ext/xmltv/ 디렉터리에서 *ws.overview.tv.xml* 파일을 엽니다.
2. *Namespaces*의 모든 인스턴스를 찾아 *Services*로 바꿉니다.
3. *ws.overview.tv.xml* 파일을 저장합니다.
4. **Enterprise Manager**를 다시 시작합니다.

Enterprise Manager를 다시 시작한 후 **Investigator** 트리에서 "WebServices", "클라이언트" 또는 "서버" 노드를 선택하면 "개요" 탭에 "서비스" 레이블이 표시됩니다.

Web Services Manager 7.0.x 의 메트릭 명명 규칙을 사용하려면

1. <Agent_Home> 디렉터리에서 *IntroscopeAgent.profile* 파일을 엽니다.
2. 파일에 *com.wily.introscope.agent.soa.metricNameFormatting* 속성을 추가합니다.

3. 다음과 같이 *com.wily.introscope.agent.soa.metricNameFormatting* 속성을 설정합니다.

`com.wily.introscope.agent.soa.metricNameFormatting=/:`

이 설정은 메트릭 이름에서 슬래시(/)와 콜론(:) 문자를 밑줄(_) 문자로 바꿉니다. 이 설정을 사용하면
`http://CheckingAccount/demobank.ca.com` ◎
`http_CheckingAccount_demobank.ca.com` 으로 표시됩니다.

4. *IntroscopeAgent.profile* 파일을 저장합니다.

5. 응용 프로그램 서버를 다시 시작합니다.

응용 프로그램 서버를 다시 시작하면 **Investigator** 트리와 SOA 종속성 맵 네임스페이스 대신 웹 서비스 끝점 이름이 표시됩니다.

상호 관련된 추적 구성

크로스 프로세스 트랜잭션을 추적하려면 에이전트가 한 프로세스에서 다른 프로세스로 전달될 수 있는 상관 관계 식별자를 삽입해야 합니다.

에이전트는 이 상관 관계 식별자를 SOAP 또는 HTTP 헤더에 삽입할 수 있습니다.

대부분의 서비스가 SOAP 메시징을 사용하기 때문에 상관 관계 식별자는 기본적으로 SOAP 헤더에서 삽입되고 읽힙니다. 그러나 경우에 따라 HTTP 헤더를 사용하여 상관 관계 식별자를 전달할 수도 있습니다. 예를 들어 드문 경우이지만, SOAP 헤더에 상관 관계 식별자를 추가하면 상관 관계 식별자 때문에 SOAP 페이로드가 변경되어 메시지가 거부될 수 있습니다.

응용 프로그램에서 SOAP 메시지를 처리하는 방법과 보안이 구현된 방식에 따라 *IntroscopeAgent.profile* 파일을 수정하여 상관 관계 식별자를 HTTP 헤더에 전달할지 아니면 SOAP 헤더에 전달할지 선택할 수 있습니다.

클라이언트와 서버에서 상관 관계 식별자의 처리를 제어하는 에이전트 구성 속성에는 네 가지가 있습니다. 이러한 속성을 사용하면 ID를 SOAP 헤더에 삽입할지, HTTP 헤더에 삽입할지, 두 가지 헤더 프로토콜에 모두 삽입할지 또는 아예 삽입하지 않을지 지정할 수 있습니다. 모든 속성은 표준 에이전트 접두사인 *com.wily.introscope.agent*로 시작합니다.

soapheaderinsertion.enabled

클라이언트가 상관 관계 식별자를 SOAP 헤더에 삽입하도록 합니다.

- 클라이언트가 SOAP 헤더를 사용하도록 허용하려면 속성을 **true**로 설정합니다.
- 클라이언트가 상관 관계 식별자를 SOAP 헤더에 삽입하지 못하게 하려면 속성을 **false**로 설정합니다.

이 속성은 기본적으로 **true**로 설정됩니다.

httpheaderinsertion.enabled

클라이언트가 상관 관계 식별자를 HTTP 헤더에 삽입하도록 합니다.

- 클라이언트가 HTTP 헤더를 사용하도록 허용하려면 이 속성을 **true**로 설정합니다.
- 클라이언트가 상관 관계 식별자를 HTTP 헤더에 삽입하지 못하게 하려면 이 속성을 **false**로 설정합니다.

이 속성은 기본적으로 **false**로 설정됩니다.

soapheaderread.enabled

서버가 SOAP 헤더에서 상관 관계 식별자를 읽도록 합니다.

- 서버가 SOAP 헤더를 사용하도록 허용하려면 이 속성을 **true**로 설정합니다.
- 서버가 SOAP 헤더에서 읽지 못하게 하려면 이 속성을 **false**로 설정합니다.

이 속성은 기본적으로 **true**로 설정됩니다.

httpheaderread.enabled

서버가 HTTP 헤더에서 상관 관계 식별자를 읽도록 합니다.

- 서버가 HTTP 헤더를 사용하도록 허용하려면 이 속성을 **true**로 설정합니다.
- 서버가 HTTP 헤더에서 읽지 못하게 하려면 이 속성을 **false**로 설정합니다.

이 속성은 기본적으로 **false**로 설정됩니다.

클라이언트와 서버의 속성 구성

에이전트별로 특정 동작을 사용하거나 사용하지 않도록 속성을 설정할 수 있습니다. 예를 들어 서버가 SOAP 기반 서비스와 HTTP 기반 서비스 양쪽에서 메시지를 처리하는 경우, 이 서버의 에이전트가 두 가지 헤더 모두에서 상관 관계 식별자를 읽되 SOAP 헤더에 ID를 삽입하지는 못하도록 구성할 수 있습니다.

```
com.wily.introscope.agent.soapheaderread.enabled=true
com.wily.introscope.agent.httpheaderread.enabled=true
com.wily.introscope.agent.soapheaderinsertion.enabled=false
com.wily.introscope.agent.httpheaderinsertion.enabled=true
```

이러한 설정을 사용하면 로컬 컴퓨터는 다른 에이전트가 보낸 상관 관계 식별자를 SOAP 헤더에서 읽을 수 있지만 상관 관계 식별자를 HTTP 헤더에만 삽입할 수 있습니다. 그러나 추적 기능을 사용하려면 웹 서비스 클라이언트와 서버가 상관 관계 식별자를 삽입하고 읽는 데 같은 유형의 헤더를 사용해야 합니다. 예를 들어 상관 관계 식별자를 HTTP 헤더에 삽입하도록 클라이언트를 구성한 경우 반드시 HTTP 헤더에서 상관 관계 식별자를 읽도록 서버를 구성해야 합니다.

상호 관련된 추적 사용 안 함

SOAP 및 HTTP 둘 다에 상관 관계 식별자를 삽입하지 못하게 설정하고 크로스 프로세스 트랜잭션 추적 기능을 사용하지 않으려면 네 가지 속성 모두를 false로 설정하면 됩니다. 예:

```
com.wily.introscope.agent.soapheaderread.enabled=false
com.wily.introscope.agent.httpheaderread.enabled=false
com.wily.introscope.agent.soapheaderinsertion.enabled=false
com.wily.introscope.agent.httpheaderinsertion.enabled=false
```

이러한 속성을 수정하기 전에 SOA 종속성 맵과 크로스 프로세스 트랜잭션 추적 기능을 사용하려면 프로세스 간에 상관 관계 식별자가 전달되어야 한다는 점에 유의하십시오.

이러한 속성을 false로 설정하여 상호 관련된 추적을 사용하지 않도록 지정하면 SOA 종속성 맵에 종속성이 올바르게 표시되지 않고 종속성 메트릭을 정확하게 수집할 수 없으며 트랜잭션 추적이 완료되지 않을 수 있습니다.

SOAP 핸들러의 삽입 지점 구성

IntroscopeAgent.profile 파일의 구성 속성을 사용하여 SOAP 핸들러 체인에서 SOAP 핸들러가 삽입되는 위치를 제어할 수 있습니다. 이 속성은 암호화가 포함되거나 원본 SOAP 메시지에 대한 서명 확인이 필요한 응용 프로그램에 유연성을 제공하므로 SOAP 헤더를 삽입해도 응용 프로그램의 SOAP 메시지를 확인하는 데 방해가 되지 않습니다.

SOAP 핸들러 삽입을 위한 클라이언트 및 서버 속성

SOAP 핸들러를 사용하는 응용 프로그램의 경우 다음의 구성 속성을 사용하여 CA APM for SOA SOAP 헤더를 제어할 수 있습니다.

soa.client.prependhandler

SOAP 헤더를 클라이언트의 SOAP 핸들러 체인에서 처음 또는 마지막에 삽입할 수 있습니다.

- SOAP 헤더가 SOAP 핸들러 체인의 첫 번째 핸들러에 의해 삽입되도록 하려면 속성을 **true**로 설정하십시오.
- SOAP 헤더가 SOAP 핸들러 체인의 마지막 핸들러에 의해 추가되도록 하려면 속성을 **false**로 설정하십시오.

이 속성은 기본적으로 **true**로 설정됩니다.

soa.server.appendhandler

SOAP 헤더를 서버의 SOAP 핸들러 체인에서 처음으로 또는 마지막으로 읽도록 합니다.

- SOAP 헤더를 SOAP 핸들러 체인의 마지막 핸들러가 읽도록 하려면 이 속성을 **true**로 설정하십시오.
- SOAP 헤더를 SOAP 핸들러 체인의 첫 번째 핸들러가 읽도록 하려면 이 속성을 **false**로 설정하십시오.

이 속성은 기본적으로 **true**로 설정됩니다.

SOAP 핸들러의 기본 순서 변경

기본적으로 클라이언트의 핸들러 체인 중 첫 번째 핸들러가 SOAP 헤더를 삽입하며 서버의 체인 중 마지막 핸들러가 헤더를 읽습니다. 일부 응용 프로그램의 경우 CA APM for SOA SOAP 헤더 및 원본 SOAP 메시지를 적절하게 전달하기 위해 기본 동작을 수정해야 할 수 있습니다. 예를 들어 응용 프로그램이 SOAP 메시지의 서명을 확인해야 하는 경우에는 기본 동작을 수정해야 할 수 있습니다.

기본 속성 설정을 변경해야 하는 이유를 설명하기 위해, 체인의 첫 번째 핸들러가 SOAP 헤더를 삽입하는 응용 프로그램을 예로 들겠습니다. 이 경우 SOAP 메시지는 두 번째 핸들러에서 서명됩니다. 서버에 메시지가 수신되면 첫 번째 핸들러가 서명 확인을 시도합니다. CA APM for SOA SOAP 헤더가 삽입되었지만 아직 이를 읽고 제거하지 않았으므로 서명이 실패하고 메시지가 거부됩니다.

구성 속성을 사용하면 체인의 첫 번째 핸들러가 CA APM for SOA SOAP 헤더를 읽은 다음 이를 제거하도록 기본 동작을 변경할 수 있습니다. 이렇게 설정하면 체인의 다음 핸들러가 원본 SOAP 메시지를 기준으로 서명을 확인하도록 할 수 있습니다. 예:

```
com.wily.introscope.agent.soa.client.prependhandler=true  
com.wily.introscope.agent.soa.server.appendhandler=false
```

기본 속성 설정은 대부분의 환경에 적합하지만 이러한 구성 속성은 CA APM for SOA 를 여러 응용 프로그램 시나리오에 맞게 조정할 수 있는 유연성을 제공합니다.

SOA 종속성 맵의 제한 구성

Enterprise Manager에 저장되는 종속성 맵 데이터는 대개 검색된 응용 프로그램 간의 모든 종속성을 나타내며, 배포한 SOA(Service-Oriented Architecture)의 완전한 모델을 제공합니다. 하지만 매우 크거나 복잡한 SOA 환경에서는 모든 SOA 구성 요소와 해당 종속성의 전체 표현이 Enterprise Manager 자체의 성능과 작업에 영향을 미칠 수 있습니다.

종속성 맵이 Enterprise Manager의 성능에 영향을 주지 않도록 맵의 크기와 복잡성을 제한하는 몇 가지 구성 속성이 있습니다. 기본적으로 이러한 구성 속성은 종속성 맵의 크기를 최대 5000 개의 노드와 최대 25000 개의 종속성(노드 당 종속성 관계 5 개의 비율)으로 제한합니다.

이러한 제한을 제어하기 위해 *IntroscopeEnterpriseManager.properties* 파일에서 다음과 같은 Enterprise Manager 구성 속성을 설정할 수 있습니다.

dependencymap.max.vertices

독립 실행형 또는 Collector Enterprise Manager에 종속성 데이터를 저장할 수 있는 노드의 최대 수를 설정합니다.

기본적으로 이 속성은 5000으로 설정됩니다.

dependencymap.max.edge.ratio

독립 실행형 또는 Collector Enterprise Manager에 저장할 수 있는 종속성 데이터의 전체적인 복잡성을 제어하기 위해 노드에 대한 종속성 관계의 최대 비율을 설정합니다.

기본적으로 이 속성은 5로 설정됩니다.

대부분의 조직에서 SOA 네트워크에는 5000개 미만의 구성 요소가 필요하며 일반적인 종속성 비율은 구성 요소당 종속성 1개 또는 2개입니다. 따라서 기본 설정은 대부분의 조직에 필요한 것보다 더 크고 복잡한 환경을 지원하기에 충분합니다. 드물지만 이러한 기본값을 수정해야 하는 경우도 있습니다. 예를 들어 다음과 같은 경우에 속성을 수정해야 할 수 있습니다.

- 기본값이 SOA 환경을 지원하기에 충분하지 않아 더 크고 복잡한 종속성 맵을 지원하도록 값을 늘리려는 경우. 이 경우에는 값을 변경했을 때 Enterprise Manager의 성능에 어떤 영향을 줄지 고려해야 합니다.
- SOA 환경에 필요한 노드와 종속성 수가 기본값의 수보다 적어서 종속성 맵의 크기와 복잡성을 의도적으로 제한하려는 경우. 이 경우에는 변경으로 인해 SOA 종속성 맵 모델이 완료되지 않을 가능성이 있는지 고려해야 합니다.

예를 들어 데이터를 저장할 노드 수는 줄이고 노드당 종속성 관계 수는 늘리려면 속성을 다음과 같이 수정해야 합니다.

```
com.wily.introscope.soa.dependencymap.max.vertices=2000  
com.wily.introscope.soa.dependencymap.max.edge.ratio=8
```

제 7 장: Oracle Service Bus 모니터링

OSB(Oracle Service Bus)는 느슨하게 연결된 서비스 소비자와 서비스 공급자 사이에 메시지 처리 및 통신을 제공합니다. SOA extension for Oracle Service Bus 를 사용하면 파이프라인, 프록시 서비스 및 전송 프로토콜 같은 주요 구성 요소를 통해 Oracle Service Bus 작업을 모니터링 할 수 있습니다.

이 단원에서는 Oracle Service Bus 의 성능, 가용성 및 전반적인 상태를 모니터링하고 분석하는 데 사용할 수 있는 OSB 관련 대시보드, 메트릭 및 경고에 대해 설명합니다.

이 섹션은 다음 항목을 포함하고 있습니다.

[OSB\(Oracle Service Bus\) 정보](#) (페이지 127)

[Oracle Service Bus 모니터링을 사용하도록 설정하는 방법](#) (페이지 131)

[대시보드를 사용하여 Oracle Service Bus 모니터링](#) (페이지 136)

[Oracle Service Bus 에 대한 메트릭 이해 및 보기](#) (페이지 139)

[기본 Oracle Service Bus 메트릭 그룹화 보기](#) (페이지 145)

[기본 Oracle Service Bus 경고 보기](#) (페이지 146)

[Oracle Service Bus 종속성 보기](#) (페이지 147)

[Oracle Service Bus 에 대한 트랜잭션 추적](#) (페이지 148)

OSB(Oracle Service Bus) 정보

SOA 환경에서 Enterprise Service Bus 는 대개 서비스 소비자와 서비스 공급자 사이의 메시징 계층을 제공합니다. Enterprise Service Bus 는 분산된 이기종 환경에서 데이터와 메시지의 검사, 변환, 라우팅 및 보안 기능을 제공합니다.

Oracle Service Bus 는 프록시 인터페이스의 경량 중간 계층에서 메시지 브로커, 중재 및 서비스 수명 주기 관리 기능을 제공하는 Enterprise Service Bus 의 예입니다.

Oracle Service Bus 를 사용하여 메시지 처리 규칙을 정의하면 다음에 대한 메트릭을 사용하여 Oracle Service Bus 의 작업을 모니터링 할 수 있습니다.

비즈니스 서비스

비즈니스 서비스는 Oracle Service Bus 가 클라이언트인 외부 서비스에 대한 정의입니다.

외부 서비스는 외부 시스템이 구현하고 호스트합니다. 이러한 외부 서비스를 사용하려면 Oracle Service Bus 는 호출하는 데 필요한 인터페이스, 호출하는 방법 및 호출로 예상되는 결과를 알고 있어야 합니다. OSB 내의 비즈니스 서비스는 버스에서 외부 시스템을 호출하고 서로 상호 작용 할 수 있도록 외부 인터페이스를 모델링 합니다. OSB 내에서 비즈니스 서비스 구성에는 인터페이스, 전송 설정 및 보안 설정이 포함됩니다.

SOA extension for OSB 를 사용하면 비즈니스 서비스가 외부 시스템과 어떻게 상호 작용하는지 모니터링하고, "OSB" > "비즈니스 서비스" 노드 아래에서 비즈니스 서비스의 전반적인 상태에 대한 데이터를 수집 할 수 있습니다.

파이프라인

파이프라인은 "요청", "응답" 또는 "오류" 메시지 흐름에 대한 특정 단계를 처리하는 이름이 지정된 순서입니다.

프록시 서비스의 처리 논리를 구현하기 위해 요청 및 응답 파이프라인이 파이프라인 쌍 노드에 쌍으로 연결됩니다. 이러한 파이프라인 쌍 노드는 다른 노드와 함께 루트가 하나인 트리 구조에 결합되어 전반적인 흐름을 제어합니다. 오류 파이프라인은 메시지 흐름에서 단계와 노드에 대한 오류 및 메시지 흐름이나 비즈니스 서비스의 오류를 처리합니다.

SOA extension for OSB 를 사용하면 요청 및 응답 파이프라인의 성능을 모니터링하고 "OSB" > "Pipelines" 노드 아래에서 파이프라인의 전반적인 상태에 대한 데이터를 수집 할 수 있습니다.

프록시 서비스

프록시 서비스는 서비스 버스가 로컬로 구축하고 호스트하는 중간 웹 서비스에 대한 정의입니다.

Oracle Service Bus 는 프록시 서비스를 사용하여 비즈니스 서비스와 프레젠테이션 응용 프로그램이나 기타 비즈니스 서비스 같은 서비스 클라이언트 사이에 메시지를 라우팅합니다.

프록시 서비스 구성에는 인터페이스, 전송 설정, 보안 설정 및 메시지 흐름 정의가 포함됩니다. 메시지 흐름은 프록시 서비스를 통해 메시지를 전달할 때 메시지가 어떻게 처리되는지 결정하는 논리를 정의합니다.

SOA extension for OSB 를 사용하면 프록시 서비스의 성능을 모니터링하고 "OSB" > "프록시 서비스" 노드 아래에서 프록시 서비스의 전반적인 상태에 대한 데이터를 수집할 수 있습니다.

전송

전송은 메시지를 보내고 배달하는 메커니즘을 정의하며, Oracle Service Bus 가 지원하는 모든 전송 프로토콜을 포함할 수 있습니다.

전송 공급자는 전송 끝점의 수명 주기와 런타임 동작을 관리합니다. 대상 끝점은 메시지의 출처 또는 대상이 되는 리소스입니다. OSB 의 네이티브 공급자를 사용하면 이러한 전송 프로토콜을 필요로 하는 프록시 및 비즈니스 서비스를 구성할 수 있습니다. 사용자 지정 전송 공급자를 만들거나 설치할 수도 있습니다.

SOA extension for OSB 를 사용하면 지원되는 모든 전송 프로토콜을 모니터링하고 "OSB" > "전송" 노드 아래에서 인바운드 끝점과 아웃바운드 끝점에 대한 메트릭을 수집할 수 있습니다.

UDDI

UDDI(Universal Description, Discovery, and Integration)는 전 세계의 회사에서 서비스를 인터넷에 게시할 수 있게 하는 플랫폼 독립적인 XML 기반 레지스트리입니다.

UDDI는 회사에서 서비스 목록을 게시하고, 다른 회사의 비즈니스 서비스 목록을 검색하고, 서비스나 소프트웨어 응용 프로그램이 인터넷 상에서 상호 작용하는 방식을 정의하는 개방형 산업 이니셔티브입니다.

Oracle Service Bus 와 UDDI 버전 3.0 과 호환되는 UDDI 레지스트리를 사용하면 다음과 같은 작업을 수행할 수 있습니다.

- 프록시 서비스에 대한 정보를 레지스트리에 게시합니다.
레지스트리는 WSDL(Web Services Description Language), SOAP 또는 XML 기반 레지스트리일 수 있습니다.
- 레지스트리에 포함된 특정 서비스를 문의하거나 사용 가능한 모든 서비스를 나열합니다. 비즈니스 엔터티, 서비스 이름 패턴 또는 둘 다를 대상으로 검색할 수 있습니다.
- 레지스트리에서 비즈니스 서비스를 가져옵니다.

SOA extension for OSB 를 사용하면 UDDI 레지스트리를 모니터링하고 "OSB" > "UDDI" 노드 아래에서 게시, 가져오기 및 조회 작업에 대한 메트릭을 수집할 수 있습니다.

XQuery

Oracle Service Bus 는 Oracle Data Services 플랫폼에 구축된 XQuery 엔진을 사용하여 데이터 변환에 대한 XQuery 를 지원합니다.

Oracle XQuery 엔진은 변환 맵을 사용하여 데이터 유형 간의 매핑을 설명하고, Oracle Service Bus 는 XQuery 를 사용한 데이터 매핑을 지원합니다. 변환을 만들고, 구문 분석하고 실행할 수 있습니다.

SOA extension for OSB 를 사용하면 XQuery 를 만들고, 구문 분석하고 실행하기 위한 XQuery 변환을 모니터링하고 "OSB > XQuery" 노드 아래에서 이러한 작업에 대한 메트릭을 수집할 수 있습니다.

Oracle Service Bus 모니터링을 사용하도록 설정하는 방법

Oracle Service Bus 의 모니터링을 사용하도록 설정할 때는 다음과 같은 간략한 단계를 수행합니다.

1. 지원되는 버전의 Oracle Service Bus 가 설치되어 있는지 확인합니다.
참고: 시스템 요구 사항에 대해서는 *Compatibility Guide*(호환성 안내서)를 참조하십시오.
2. 에이전트 및 CA APM for SOA 가 설치되고 사용하도록 설정되었는지 확인합니다.
3. Oracle Service Bus 모니터링을 위한 적절한 지시문이 포함되도록 에이전트 프로필을 구성하여 [Oracle Service Bus 모니터링을 사용하도록 에이전트를 설정](#) (페이지 131)합니다. Standalone Agent 설치 관리자를 사용하거나 응답 파일을 사용하여 에이전트를 사용하도록 설정한 경우에는 이 단계를 건너뛸 수 있습니다.
4. 파일을 <EM_Home>/examples/SOAExtensionForOSB 디렉터리에서 적절한 Enterprise Manager 디렉터리로 이동하여 [Enterprise Manager 확장을 사용하도록 설정](#) (페이지 134)합니다.

Oracle Service Bus 모니터링을 사용하도록 에이전트 설정

에이전트를 설치할 때 CA APM for Oracle Service Bus 를 추가하고 사용하도록 설정한 다음 WebLogic 을 응용 프로그램 서버로 선택할 수 있습니다. 또한 에이전트를 설치한 후 수동으로 CA APM for Oracle Service Bus 를 사용하도록 설정할 수 있습니다.

에이전트를 설치할 때 CA APM for Oracle Service Bus 를 사용하도록 설정한 경우에는 에이전트 프로필이 기본 설정으로 자동 구성되며 추가 단계 없이도 에이전트 확장이 사용하도록 설정됩니다.

에이전트를 설치할 때 CA APM for Oracle Service Bus 를 사용하도록 설정하지 않은 경우에는 에이전트 프로필을 수동으로 구성하여 모니터링을 사용하도록 설정해야 합니다.

수동으로 CA APM for Oracle Service Bus 를 사용하도록 설정하려면

1. 에이전트 및 CA APM for SOA 가 설치되고 사용하도록 설정되었는지 확인합니다.
2. CA APM for Oracle Service Bus 디렉터리 *SOAExtensionForOSB* 가 *<Agent_Home>/examples* 디렉터리에 있는지 확인하고 *<Agent_Home>/examples/SOAExtensionForOSB/ext* 디렉터리의 파일을 *<Agent_Home>/core/ext* 디렉터리로 복사합니다.
3. *IntroscopeAgent.profile* 파일을 텍스트 편집기에서 엽니다.
4. 적절한 *OSB-full.pbl* 또는 *OSB-typical.pbl* 파일을 *IntroscopeAgent.profile* 파일의 *introscope.autoprobe.directivesFile* 속성에 추가합니다.
적절한 *ProbeBuilder* 지시문 선택에 대한 자세한 내용은 [Oracle Service Bus에 대한 지시문 파일 정보](#) (페이지 133)를 참조하십시오.
5. 기본 설정을 변경하려면 *IntroscopeAgent.profile* 파일에서 추가 SOA 특정 에이전트 구성 속성을 수정합니다.
보안이 구현된 방법에 따라 일부 응용 프로그램은 기본 설정을 변경해야 할 수 있습니다.
예를 들어 웹 서비스에 대한 통신을 암호화한 경우에는 에이전트가 HTTP 상관 관계를 사용하고 SOAP 상관 관계를 사용하지 않도록 에이전트의 기본 구성을 수정해야 합니다. 기본적으로 SOAP 및 HTTP 상관 관계는 모두 Oracle Service Bus에 대해 사용하도록 설정됩니다. HTTP 또는 SOAP 상관 관계에 대한 구성 속성 설정에 대한 자세한 내용은 [상호 관련된 추적 구성](#) (페이지 121)을 참조하십시오.
6. 변경 사항을 *IntroscopeAgent.profile* 파일에 저장하고 텍스트 편집기를 닫습니다.

Oracle Service Bus 에 대한 지시문 파일 정보

IntroscopeAgent.profile 의 *introscope.autoprobe.directivesFile* 속성을 설정할 때 기본값인 표준 계측을 선택하거나 전체 계측을 선택하여 에이전트를 배포하는 환경에 적합하게 모니터링 수준, 메트릭 가시성 및 성능 오버헤드를 제어할 수 있습니다. 그런 다음 특정 추적 프로그램 그룹에 대한 추적을 설정하거나 해제하여 표준 또는 전체 전환 파일을 통해 특정 구성 요소에 대한 모니터링을 세밀하게 조정할 수 있습니다.

OSB-full.pbl

모든 Oracle Service Bus 구성 요소의 작업에 대한 세부적인 가시성을 제공하는 전체 계측입니다.

전체 계측은 세부적인 보고 기능을 제공하지만 오버헤드가 많이 발생합니다. 전체 계측은 테스트 또는 개발 환경에 권장되며 프로덕션 환경에는 일반적으로 권장되지 않습니다.

기본적으로 OSB-full.pbl 파일 목록에는 다음과 같은 지시문 파일이 포함됩니다.

- OSB-toggles-full.pbd
- OSB.pbd

OSB-typical.pbl

오버헤드가 매우 중요한 프로덕션 환경에서 중요 Oracle Service Bus 구성 요소에 대한 표준 모니터링입니다.

표준 계측은 세부 수준은 낮지만 오버헤드를 줄일 수 있으며 프로덕션 환경에 권장됩니다.

기본적으로 OSB-typical.pbl 파일 목록에는 다음과 같은 지시문 파일이 포함됩니다.

- OSB-toggles-typical.pbd
- OSB.pbd

OSB-toggles-full.pbd

전체 계측을 사용할 때 Oracle Service Bus 구성 요소에 대한 모니터링을 켜거나 끄도록 전환합니다.

이 파일은 다른 지시문 파일에 제공되는 추적 기능에 필요한 지시문을 제공합니다. 기본적으로 이 파일에 있는 대부분의 추적 프로그램 그룹은 사용되도록 설정됩니다.

OSB-toggles-typical.pbd

표준 계측을 사용할 때 Oracle Service Bus 구성 요소에 대한 모니터링을 켜거나 끄도록 전환합니다. 이 파일은 다른 지시문 파일에 제공되는 추적 기능에 필요한 지시문을 제공합니다. 기본적으로 추적 프로그램 그룹의 일부만 사용되도록 설정됩니다.

같은 Enterprise Manager 에 데이터를 보고하는 모든 에이전트는 동일한 계측 수준을 사용해야 하며 Enterprise Manager 는 모든 표준 계측 에이전트 또는 모든 전체 계측 에이전트와 작업하도록 구성되어야 합니다. 예를 들어 표준 계측을 사용하여 에이전트를 구성하면 Enterprise Manager 파일의 표준 버전을 배포해야 합니다.

Oracle Service Bus Enterprise Manager 확장을 사용하도록 설정

Enterprise Manager 를 설치할 때 CA APM for Oracle Service Bus 를 추가하고 사용하도록 설정할 수 있습니다. 또한 Enterprise Manager 를 설치한 후 수동으로 CA APM for Oracle Service Bus 를 사용하도록 설정할 수 있습니다.

CA APM for Oracle Service Bus 파일은 기본적으로 Enterprise Manager 를 설치할 때 *<EM_Home>/examples* 디렉터리에 설치됩니다. CA APM for Oracle Service Bus 를 사용하도록 설정하려면 Enterprise Manager 파일을 *<EM_Home>/examples* 디렉터리에서 Enterprise Manager 홈 디렉터리의 적절한 위치로 복사하거나 이동하십시오.

참고: CA APM for Oracle Service Bus 를 사용하려면 Enterprise Manager 에서 CA APM for SOA 를 사용하도록 설정해야 합니다.

다음 단계를 수행하십시오.

1. CA APM for Oracle Service Bus 디렉터리인 SOAExtensionForOSB 가 <EM_Home>/examples 디렉터리에 있는지 확인합니다.
 2. 표준 계측 또는 전체 계측에 대한 적절한 파일을 <EM_Home>/examples/SOAExtensionForOSB 디렉터리에서 Enterprise Manager 디렉터리 구조의 해당 위치로 복사합니다. 예를 들어 표준 계측을 사용하는 경우 com.wily.powerpacks.osb.emext.calculator_typical.jar 파일을 <EM_Home>/examples/SOAExtensionForOSB/ext 디렉터리에서 <EM_Home>/ext 디렉터리로 복사합니다.
- 참고:** 동일한 Enterprise Manager 로 데이터를 보고하는 모든 에이전트에 대하여 동일한 계측 수준을 사용해야 합니다. 예를 들어 Enterprise Manager 파일의 표준 버전을 배포하는 경우 해당 Enterprise Manager 와 통신하는 모든 에이전트를 표준 계측으로 구성해야 합니다. 전체 계측을 사용하는 에이전트는 표준 모드로 구성된 Enterprise Manager 로 데이터를 보고하지 않아야 합니다.
3. Enterprise Manager 가 클러스터 환경에서 수집기인 경우 Oracle Service Bus 관리 모듈 OSB_ManagementModule_typical.jar 또는 OSB_ManagementModule_full.jar 을 <EM_Home>/config/modules 디렉터리에서 제거합니다.
- 관리 모듈만 MOM 컴퓨터로 사용 중인 Enterprise Manager 의 <EM_Home>/config/modules 디렉터리로 복사하십시오. 다른 모든 파일과 스크립트는 Collector Enterprise Manager 와 MOM Enterprise Manager 모두에 설치해야 합니다.
4. 이전 버전의 SOA extension for Oracle Service Bus 에서 업그레이드할 때는 이전 버전의 Enterprise Manager 파일을 제거하십시오.
 - CA APM for Oracle Service Bus 의 이전 릴리스가 Enterprise Manager 에 설치된 경우 새 버전의 CA APM for Oracle Service Bus 를 사용하기 전에 이전 버전의 Enterprise Manager 파일을 수동으로 삭제하십시오.
 - CA APM for Oracle Service Bus 의 이전 릴리스에서 업그레이드한 경우에는 Enterprise Manager 홈 디렉터리에서 다음 파일을 삭제하십시오.
- <EM_home>/config/modules/OSB_ManagementModule<version>.jar
 <EM_home>/ext/xmltv/OSB.overview.tv.xml
 <EM_home>/product/enterprisemanager/plugins/
 com.wily.powerpacks.osb.emext.calculator_<version>.jar

예를 들어 버전 8.1.1에서 업그레이드 중인 경우에는 다음 파일을 삭제합니다.

OSB_ManagementModuleV8.1.1.0.jar
OSB.overview.tv.xml
com.wily.powerpacks.osb.emext.calculator_8.1.1.0.jar

5. Workstation을 다시 시작합니다.

CA APM for Oracle Service Bus에 특정한 대시보드 및 "개요" 탭이 로드됩니다.

추가 정보:

[Enterprise Manager에서 확장을 사용하도록 설정](#) (페이지 41)

대시보드를 사용하여 Oracle Service Bus 모니터링

SOA extension for Oracle Service Bus에는 응용 프로그램 환경의 전반적인 건전성을 모니터링하는 데 사용할 수 있는 몇 개의 사전 구성된 대시보드가 포함되어 있습니다. 대시보드는 배포된 에이전트에서 데이터를 집계하여 성능 정보를 요약하며 문제를 신속하게 진단하고 해결할 수 있도록 해 줍니다.

일반적으로 대시보드는 다음과 같은 작업을 가능하게 하므로 환경을 모니터링하는 시작점으로 사용됩니다.

- Oracle Service Bus의 주요 구성 요소에 대한 전반적인 건전성, 성능, 가용성 및 현재 상태를 한눈에 모니터링
- 하위 수준 메트릭이 주의 또는 위험 임계값이 초과되었음을 알리는 경우 프로덕션 응용 프로그램 환경의 잠재적 문제를 조기에 식별
- 성능 정보를 드릴다운하여 지역 중이거나 오류가 발생 중인 Oracle Service Bus 구성 요소, 전송 프로토콜, 끝점 또는 작업을 격리 및 식별

사전 구성된 Oracle Service Bus 대시보드는 배포한 OSB 관리 모듈(*OSB_ManagementModule_typical.jar* 또는 *OSB_ManagementModule_full.jar*)의 일부로 Enterprise Manager extension for Oracle Service Bus에 패키지되어 있습니다.

OSB 관리 모듈은 Oracle Service Bus 에 대해 다음과 같은 사전 구성된 대시보드를 제공합니다.

OSB 홈

모든 전송, 프록시 서비스, 파일프라인 및 비즈니스 서비스의 전반적인 건전성에 대한 경고 표시기를 포함하여 Oracle Service Bus 의 주요 구성 요소를 통과하는 작업 흐름의 상위 수준 뷰입니다.

OSB 프록시 서비스 - 개요

모든 프록시 서비스의 요약된 상태로, "평균 응답 시간", "간격당 오류 수", "중단 수"에 대한 경고와 그래프, 그리고 가장 느린 10 개의 프록시 서비스 목록을 포함합니다.

OSB 프록시 서비스 - 세부 정보

프록시 서비스를 거치는 요청 및 응답 작업에 대한 별도의 요약 메트릭으로, "평균 응답 시간", "간격당 오류 수", "중단 수"에 대한 경고와 그래프를 포함합니다.

OSB 비즈니스 서비스 - 개요

모든 비즈니스 서비스의 요약된 상태로, "평균 응답 시간"과 "간격당 오류 수"에 대한 경고와 그래프, 가장 느린 10 개의 비즈니스 서비스 목록을 포함합니다.

OSB 파일프라인 - 개요

모든 파일프라인의 요약된 상태로, "평균 응답 시간", "간격당 오류 수", "중단 수"에 대한 경고와 그래프, 그리고 가장 느린 10 개의 파일프라인을 포함합니다.

OSB 전송 I - 개요

많이 사용되는 전송 유형(HTTP/HTTPS, JMS, EJB, FTP, 전자 메일 및 SFTP)의 요약된 건전성으로, 모든 전송 유형의 "평균 응답 시간", "간격당 오류 수", "중단 시간"에 대한 경고, 그리고 HTTP/HTTPS, JMS, EJB, FTP, 전자 메일 및 SFTP 전송 유형에 대한 별도의 "평균 응답 시간" 그래프를 포함합니다.

OSB 전송 II - 개요

추가 전송 유형(Tuxedo, MQ, WS, JPD, SB, DSP, 파일 및 로컬)에 대한 별도의 "평균 응답 시간" 그래프입니다.

OSB 전송 끝점 - 개요

모든 전송 유형의 인바운드 서비스 및 아웃바운드 끝점에 대한 요약된 상태로, "평균 응답 시간" 그래프, 모든 인바운드 서비스 및 아웃바운드 끝점의 "간격당 오류 수" 및 "중단 수"에 대한 경고, 가장 느린 10 개의 인바운드 서비스 목록 및 가장 느린 10 개의 아웃바운드 끝점 목록을 포함합니다.

OSB XQuery - 개요

모든 XQuery 실행 시도에 대한 요약된 상태로, "실행" 작업의 "평균 응답 시간", "간격당 오류 수", "중단 수"에 대한 그래프와 경고, "구문 분석" 작업의 "평균 응답 시간" 및 "간격당 오류 수"에 대한 그래프를 포함합니다.

OSB UDDI - 개요

모든 UDDI 레지스트리 작업에 대한 요약된 상태로, UDDI 레지스트리의 "가져오기" 및 "게시" 작업의 "표준 응답 시간"과 "간격당 오류 수"에 대한 경고와 그래프를 포함합니다.

Workstation 콘솔을 사용하여 사전 구성된 대시보드를 볼 수 있습니다. 또한 사용자 지정 대시보드를 포함하도록 OSB 관리 모듈을 확장하거나 사용자 지정 메트릭이나 경고를 포함하도록 기본 대시보드 정의를 수정할 수도 있습니다.

참고: 대시보드 생성과 수정에 대한 자세한 내용은 *CA APM 구성 및 관리 안내서*를 참조하십시오.

다음 단계를 수행하십시오.

1. Enterprise Manager 가 실행되고 있지 않으면 시작합니다.
2. Workstation 을 시작하고 SOA extension for Oracle Service Bus 가 설치된 Enterprise Manager 에 로그인합니다.
3. "Workstation" > "새 콘솔"을 클릭합니다.
4. "대시보드" 드롭다운 목록에서 OSB 대시보드 중 하나를 선택합니다.

예를 들어 "전송", "프록시 서비스", "파이프라인" 및 "비즈니스 서비스"와 같은 OSB 작업 흐름의 모든 주요 구성 요소에 대한 경고 표시기가 포함된 Oracle Service Bus 의 아키텍처 개요를 보려면 "OSB 홈" 대시보드를 선택합니다.

5. 대시보드에서 경고를 두 번 클릭하면 해당 구성 요소의 대시보드가 열립니다.

예를 들어 "프록시 서비스 응답 시간" 경고를 두 번 클릭하면 "OSB 프록시 서비스 - 개요" 대시보드로 이동하고 가장 느린 프록시 서비스의 목록, 모든 프록시 서비스의 응답 시간, 중단 수 및 오류에 대한 상위 수준 경고 표시기가 표시됩니다.

6. 대시보드에서 특정 프록시 서비스, 파이프라인 또는 전송을 두 번 클릭하면 추가 분석을 위한 **Investigator** 가 열립니다.

예를 들어 "프록시 서비스 개요" 대시보드에서는 "가장 느린 10 개 프록시 서비스" 목록에서 특정 프록시 서비스를 두 번 클릭하여 해당 세부 정보를 **Investigator** 에서 확인하고, 계속해서 집계 메트릭을 통해 선택된 파이프라인의 개별 구성 요소를 드릴다운하여 대시보드 경고가 알린 지연 또는 오류 임계값의 근본 원인을 찾을 수 있습니다.

참고: **Investigator** 에서 OSB 특정 정보를 보는 방법에 대한 자세한 내용은 [Oracle Service Bus 에 대한 메트릭 이해 및 보기](#) (페이지 139)를 참조하십시오. 대시보드 간의 탐색 또는 대시보드와 **Workstation Investigator** 간의 탐색에 대한 자세한 내용은 [CA APM Workstation 사용자 안내서](#)를 참조하십시오.

Oracle Service Bus 에 대한 메트릭 이해 및 보기

Investigator 트리에서 탐색 할 때는 대부분의 Oracle Service Bus 구성 요소와 작업에 대한 표준 CA Introscope 메트릭을 볼 수 있습니다. 표준 메트릭에 대한 데이터는 **Investigator** 트리에서 노드 및 하위 노드로 표시되는 OSB 특정 메트릭 범주로 수집되고 집계됩니다. 표시되는 특정 메트릭 범주 및 노드 이름은 응용 프로그램이 사용하는 Oracle Service Bus 리소스에 따라 달라집니다.

Investigator 트리를 탐색 할 때는 선택한 노드에 따라 개별 작업에 대한 하위 수준 메트릭 표시나 집계 메트릭 표시를 선택할 수 있으므로 여러 OSB 구성 요소의 전반적인 건전성을 모니터링 할 수 있습니다.

CA CPM for SOA가 제공하는 표준 메트릭 및 추가 메트릭의 개요는 [사용할 수 있는 메트릭](#) (페이지 58)을 참조하십시오. OSB 구성 요소의 메트릭에 대한 자세한 내용은 다음 단원을 참조하십시오.

- [비즈니스 서비스에 대한 메트릭](#) (페이지 141)
- [파이프라인에 대한 메트릭](#) (페이지 141)
- [프록시 서비스에 대한 메트릭](#) (페이지 142)
- [전송에 대한 메트릭](#) (페이지 143)
- [UDDI에 대한 메트릭](#) (페이지 144)
- [XQuery에 대한 메트릭](#) (페이지 144)

Investigator에서 Oracle Service Bus 메트릭을 보고 탐색하려면

1. 에이전트 노드를 확장한 다음 OSB를 선택하여 Oracle Service Bus 구성 요소에 대한 "개요" 탭을 표시합니다.
"개요" 탭에는 사용할 Oracle Service Bus에 대해 정의한 모든 프록시 서비스 및 비즈니스 서비스의 요약 정보가 나열됩니다.
2. "개요" 탭에서 프록시 서비스 또는 비즈니스 서비스를 두 번 클릭하면 해당 프록시 서비스 또는 비즈니스 서비스에 대한 모든 집계 메트릭을 그래픽 형식으로 볼 수 있습니다.
3. OSB 노드를 확장하면 최상위 수준 Oracle Service Bus 메트릭 범주에 대한 하위 노드인 "비즈니스 서비스", "파이프라인", "프록시 서비스", "전송", "UDDI" 및 "XQuery"가 표시됩니다.
4. 하위 노드를 클릭하면 해당 메트릭 범주의 요약 정보가 있는 "개요" 탭이 표시됩니다. 예를 들어 "OSB" > "비즈니스 서비스"를 클릭하면 "개요" 탭에 나열된 모든 비즈니스 서비스를 볼 수 있습니다.
5. 하위 노드를 확장하면 개별 서비스나 작업 및 이와 연관된 메트릭이 표시됩니다.
예를 들어 "비즈니스 서비스" 노드를 확장하면 Oracle Service Bus에서 구성한 모든 비즈니스 서비스가 표시됩니다. 그런 다음 개별 비즈니스 서비스 이름을 확장하여 해당 서비스에 대한 표준 메트릭을 볼 수 있습니다.

비즈니스 서비스에 대한 메트릭

Oracle Service Bus의 경우 비즈니스 서비스는 외부 시스템에서 호스팅되는 외부 엔터프라이즈 서비스로 정의됩니다. "OSB" > "비즈니스 서비스" 노드 아래에서는 Oracle Service Bus 비즈니스 서비스에 대해 다음과 같은 표준 메트릭만 사용할 수 있습니다.

- 평균 응답 시간(ms)
- 간격당 오류 수
- 간격당 응답 수

Oracle Service Bus 플랫폼에서 비즈니스 서비스는 Oracle Service Bus가 메시지를 교환할 수 있는 외부 서비스입니다. 비즈니스 서비스로의 메시지는 외부 서비스에 대한 호출을 나타내므로 요청 및 응답 트랜잭션으로 구성되지 않습니다. 따라서 비즈니스 서비스의 "평균 응답 시간" 메트릭 계산은 총 비즈니스 서비스 실행 시간의 확인에 의존합니다.

CA Introscope는 141 Oracle Service Bus 전송 시간, 실제 서비스 실행 시간 및 네트워크 경과 시간의 합을 기반으로 비즈니스 서비스 실행 시간을 계산합니다. 예를 들어 OSB 전송 시간이 10ms이고, 실제 서비스 실행 시간이 18ms이고, 경과 시간이 5ms이면 비즈니스 서비스 실행 시간은 33ms입니다.

파이프라인에 대한 메트릭

파이프라인은 "요청", "응답" 또는 "오류" 메시지 흐름에 대한 특정 단계를 처리하는 이름이 지정된 순서입니다. OSB 전체 계측을 사용 중인 경우 "OSB" > "파이프라인" 노드 아래에서 Oracle Service Bus 파이프라인에 대해 모든 표준 CA Introscope 메트릭을 사용할 수 있습니다.

OSB 표준 계측을 사용 중인 경우에는 Investigator에 "파이프라인" 노드가 표시되지 않습니다. 파이프라인에 대한 메트릭을 Investigator에 표시하려면 *OSB-toggles-typical.pbd* 파일을 수정하여 파이프라인에 대한 추적을 사용하도록 설정하십시오.

프록시 서비스에 대한 메트릭

프록시 서비스는 Oracle Service Bus 가 비즈니스 서비스 사이에서 메시지를 라우트하는 방법을 정의합니다. 외부 웹 서비스 또는 데이터베이스의 메시지는 프레젠테이션 응용 프로그램 또는 다른 비즈니스 서비스와 같은 서비스 클라이언트로 라우트될 수 있습니다. "OSB" > "프록시 서비스" 노드 아래에서 Oracle Service Bus 프록시 서비스에 대해 모든 표준 CA Introscope 메트릭을 사용할 수 있습니다.

프록시 서비스는 요청 및 응답 트랜잭션으로 구성됩니다. 개별 프록시 서비스 이름 아래의 모든 메트릭은 "요청" 및 "응답" 하위 노드 아래에 나열된 메트릭에 기반한 집계된 메트릭입니다.

예를 들어, mortgageBroker1에서 loanGateway1 프록시 서비스로의 동시 호출은 loanGateway1 요청을 동시 loanGateway1 응답에 추가함으로써 결정됩니다.

각 트랜잭션의 응답 시간은 총 왕복 시간으로서 정의됩니다. 트랜잭션은 호출자로부터 요청을 받을 때 시작됩니다. 트랜잭션은 호출자에게 응답이 반환될 때 종료됩니다.

평균 응답 시간은 총 트랜잭션 시간을 트랜잭션 수로 나누어 계산됩니다.

전송에 대한 메트릭

전송은 메시지 전송 및 배달에 대한 메커니즘을 정의합니다. Oracle Service Bus의 경우 Oracle Service Bus에서 기본적으로 지원되는 다음 전송 유형을 포함하여 대부분의 일반적인 전송 프로토콜에 대한 메트릭을 수집할 수 있습니다.

- HTTP/HTTPS
- JMS
- EJB
- FTP
- 전자 메일
- SFTP
- Tuxedo
- MQ
- WS
- JPD
- SB
- DSP
- 파일
- 로컬

프록시 서비스가 특정 전송 유형을 사용하는 경우 그 전송 유형과 메트릭이 Investigator 트리에 표시됩니다. 전송 유형이 서비스에서 사용되지 않는 경우 그 전송 유형은 Investigator 트리에 표시되지 않습니다.

프록시 서비스가 사용자 지정 전송 유형을 사용하는 경우에는 그 전송 유형에 대해 정의된 Java 패키지 이름이 Investigator에 표시됩니다. 예를 들어 OSB 콘솔이 사용자 지정 전송 유형 com.bea.wli.sb.test.service를 사용하는 경우 Investigator에는 com.bea.wli.sb.test.service가 전송 유형으로 표시됩니다.

"OSB" > "전송" 노드 아래에서 사용 중인 특정 전송 프로토콜에 대해 모든 표준 CA Introscope 메트릭을 사용할 수 있습니다. 각 전송 프로토콜에 대해 표시되는 노드는 표준 계측을 구성했는지 전체 계측을 구성했는지에 따라 다릅니다.

예를 들어 표준 계측을 사용하는 경우 메트릭은 보내기 및 수신 노드 아래에 표시됩니다. 전체 계측을 사용 중일 때 전송 유형을 확장하면 인바운드 및 아웃바운드 하위 노드가 표시되고 *getRequestPayload*, *send* 또는 *sendMessageAsync* 와 같은 개별 작업에 대한 메트릭이 전송 유형에 대해 나열됩니다.

UDDI 에 대한 메트릭

UDDI 메트릭을 수집하고 보려면 UDDI 3.0 호환 레지스트리와 상호 작용하도록 AquaLogic 또는 Oracle Service Bus 레지스트리를 구성해야 합니다. UDDI 를 사용하도록 Oracle Service Bus 를 구성한 경우 가져오기, 조회 및 게시 작업에 대한 집계 메트릭을 수집하고 볼 수 있습니다.

"OSB" > "UDDI" 하위 노드 아래에서 가져오기, 조회 및 게시 작업에 대해 모든 표준 CA Introscope 메트릭을 사용할 수 있습니다.

XQuery 에 대한 메트릭

XQuery 를 사용하도록 Oracle Service Bus 를 구성한 경우 생성, 구문 분석 및 실행에 대한 집계 메트릭을 수집하고 볼 수 있습니다.

"OSB" > "XQuery" 하위 노드 아래에서 생성, 구문 분석 및 실행 작업에 대해 모든 표준 CA Introscope 메트릭을 사용할 수 있습니다.

생성 및 구문 분석 메서드에 대한 메트릭은 각 XQuery 에 대해 한 번만 입력됩니다. 실행 메서드에 대한 메트릭은 XQuery 표현식 또는 조건이 실행될 때마다 입력됩니다.

기본 Oracle Service Bus 메트릭 그룹화 보기

SOA extension for Oracle Service Bus 에는 기본 대시보드와 경고를 정의하는 데 사용되는 기본 메트릭 그룹화가 포함됩니다. 사용자 지정 대시보드와 경고에서도 이 기본 메트릭 그룹화를 사용할 수 있습니다.

기본 메트릭 그룹화는 배포한 OSB 관리 모듈(*OSB_ManagementModule_typical.jar* 또는 *OSB_ManagementModule_full.jar*)의 일부로 Enterprise Manager extension for Oracle Service Bus 에 패키지됩니다.

Workstation 관리 모듈 편집기를 사용하여 기본 메트릭 그룹화를 볼 수 있습니다. 또한 사용자 지정 메트릭 그룹화가 포함되도록 OSB 관리 모듈을 확장하거나 사용자 지정 대시보드 또는 경고에서 기본 메트릭 그룹화를 사용할 수도 있습니다.

다음 단계를 수행하십시오.

1. Investigator에서 Workstation > 새 관리 모듈 편집기를 클릭합니다.
2. 배포한 "*SuperDomain*" > "관리 모듈" > "OSB_ManagementModule(*SuperDomain*)"을 확장합니다.
예를 들어 표준 설정을 배포한 경우에는 "OSB_ManagementModule 표준(*SuperDomain*)"을 확장합니다.
3. "메트릭 그룹화" 노드를 확장하여 Oracle Service Bus 관리 모듈에 대해 정의된 모든 메트릭 그룹화를 표시합니다.
4. 특정 메트릭 그룹화를 클릭하여 "뷰어" 창에서 해당 정의를 봅니다.

모든 메트릭 그룹화의 기본 설정을 수정하거나 사용자 지정 메트릭 그룹화를 직접 생성할 수 있습니다.

참고: 메트릭 그룹화 생성 또는 수정에 대한 자세한 내용은 *CA APM Workstation 사용자 안내서*를 참조하십시오.

기본 Oracle Service Bus 경고 보기

SOA extension for Oracle Service Bus 에는 사전 구성된 대시보드에 사용되는 기본 경고 정의가 포함됩니다. 이 기본 경고를 사용자 지정 대시보드에서도 사용할 수 있습니다. 대부분의 기본 경고는 기본 "주의" 및 "위험" 임계값으로 사전 구성되며 임계값이 초과되거나 심각도가 증가할 경우 콘솔에 알림을 보내도록 사전 구성됩니다.

기본 경고 정의는 배포한 OSB 관리 모듈(*OSB_ManagementModule_typical.jar* 또는 *OSB_ManagementModule_full.jar*)의 일부로 Enterprise Manager extension for Oracle Service Bus 에 패키지됩니다.

Workstation 관리 모듈 편집기를 사용하여 기본 경고 정의를 볼 수 있습니다. 또한 OSB 관리 모듈을 확장하여 사용자 지정 경고 정의와 알림 유형을 포함하거나 사용자 지정 대시보드에서 기본 경고 정의를 사용할 수도 있습니다.

다음 단계를 수행하십시오.

1. *Investigator*에서 **Workstation > 새 관리 모듈 편집기**를 클릭합니다.
2. 배포한 "**SuperDomain**" > "관리 모듈" > "*OSB_ManagementModule(*SuperDomain*)*"을 확장합니다.
예를 들어 표준 설정을 배포한 경우에는 "*OSB_ManagementModule 표준(*SuperDomain*)*"을 확장합니다.
3. "경고" 노드를 클릭하여 Oracle Service Bus 관리 모듈에 대해 정의된 모든 경고를 표시합니다.
4. 특정 경고를 클릭하여 "뷰어" 창에서 해당 정의를 봅니다.

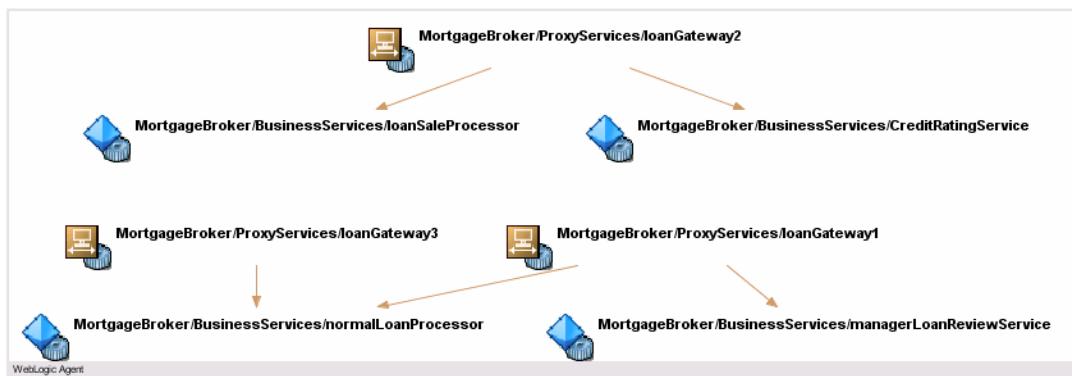
특히 "주의" 및 "위험" 임계값의 기본 설정을 검토하여 필요한 경우 값을 조정하고 알림 또는 수정 조치를 추가하십시오.

경고에 대한 기본 설정을 수정하거나 사용자 지정 경고를 직접 생성할 수 있습니다.

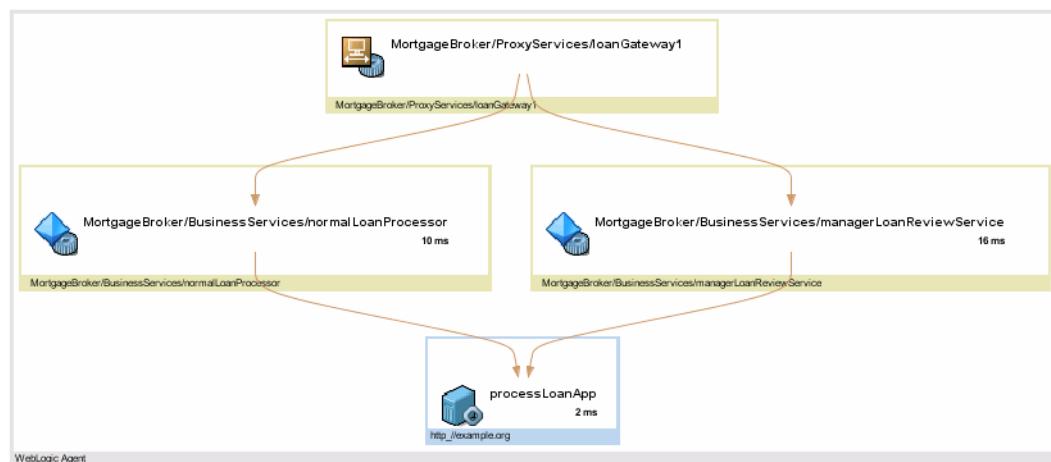
참고: 경고 생성 또는 수정에 대한 자세한 내용은 *CA APM Workstation 사용자 안내서*를 참조하십시오.

Oracle Service Bus 종속성 보기

Investigator 트리에서 "프록시 서비스" 또는 "비즈니스 서비스" 노드를 선택하거나 개별 프록시 서비스나 비즈니스 서비스를 선택한 다음 "SOA 종속성 맵" 탭을 클릭하여 Oracle Service Bus 프록시 서비스와 비즈니스 서비스의 종속성을 볼 수 있습니다. 예를 들어 에이전트의 모든 프록시 서비스에 대한 종속성의 상위 수준 뷰를 보려면 Investigator에서 "프록시 서비스" 노드를 선택한 다음 "SOA 종속성 맵" 탭을 선택합니다.



선택한 노드에 따라 종속성 맵에 표시되는 컨텍스트가 결정됩니다. 종속성 맵이 표시되면 보고자 하는 컨텍스트와 세부 정보 수준을 룰업하여 축소하거나 롤다운하여 확장할 수 있습니다. 예를 들어 Investigator에서 특정 프록시 서비스 이름을 선택한 후 "SOA 종속성 맵" 탭을 선택하면 해당 프록시 서비스의 종속성에 대한 상위 수준 뷰를 볼 수 있습니다. 다음 예제에서는 Investigator 트리에서 대출 요청을 처리하는 프록시 서비스를 선택하고 추가적인 종속성을 표시하도록 종속성 맵을 확장한 경우를 보여 줍니다.



계속해서 필요에 따라 맵에 종속성 수준을 추가하여 비즈니스 프로세스의 전체 작업 흐름을 표시하거나 맵의 특정 노드를 확대할 수 있습니다. 종속성 맵의 탐색에 대한 자세한 내용은 [SOA 종속성 맵 사용](#) (페이지 73)을 참조하십시오.

Oracle Service Bus 에 대한 트랜잭션 추적

트랜잭션 추적은 SOAP 메시지, HTTP 또는 HTTPS 헤더, Java Message Service 호출에서 전달되고 메시지, 데이터 또는 프로토콜 전송을 수행하는 Oracle Service Bus 구성 요소를 통해 전달된 작업을 포함하여 트랜잭션 완료에 관련된 특정 단계를 캡처합니다.

트랜잭션에 Oracle Service Bus 프록시 서비스 또는 다른 Oracle Service Bus 구성 요소를 통해 라우팅된 메시지가 포함된 경우에는 수행된 작업에 대한 정보와 프로세스 간 트랜잭션 추적에서 각 작업의 완료에 걸린 시간을 볼 수 있습니다. 추적 중인 모든 노드에서 CA APM for SOA 및 CA APM for Oracle Service Bus 가 사용하도록 설정되어 있으면 모든 조합의 플랫폼에서 비즈니스 트랜잭션을 추적할 수 있습니다.

크로스 프로세스 트랜잭션 추적의 가치 이해

크로스 프로세스 트랜잭션 추적은 SOA(서비스 지향 아키텍처)에서 느슨하게 결합된 서비스가 수행하는 작업에 대한 중요한 정보를 제공합니다. 크로스 프로세스 트랜잭션 추적을 사용하면 다음을 확인할 수 있습니다.

- Oracle Service Bus 구성 요소를 통해 메시지가 변환되고 라우팅되는 방법
- 트랜잭션 중에 호출되는 비즈니스 서비스와 프록시 서비스
- 트랜잭션 중에 실행되는 호출 순서
- 요청 또는 응답 처리가 가장 느린 지점

샘플 트랜잭션 추적 시작 및 보기

트랜잭션 추적 세션은 다음과 같이 시작할 수 있습니다.

- SOA 종속성 맵의 맵 노드에서 바로 시작합니다.
- "Workstation" > "새 트랜잭션 추적 세션"을 클릭하여 Workstation에서 수동으로 시작합니다.

종속성 맵에서 트랜잭션 추적을 시작하는 경우에는 맵 노드 유형에 따라 기본 필터가 자동으로 설정됩니다. 새 트랜잭션 추적 세션을 수동으로 시작하는 경우에는 Oracle Service Bus에 대한 다음 필터 유형 중 하나를 선택할 수 있습니다.

- 비즈니스 서비스
- 프록시 서비스
- namespace
- 작업

예를 들어 특정 Oracle Service Bus 프록시 서비스에 대한 트랜잭션을 필터링하려면 프록시 서비스 필터를 선택하고 프록시 서비스 이름의 전체 또는 일부를 입력하십시오.

트랜잭션 추적 세션이 시작된 후에는 "트랜잭션 추적 뷰어"를 사용하여 선택된 트랜잭션의 각 세그먼트에 대한 요약 및 세부 정보를 볼 수 있습니다. Oracle Service Bus가 관련된 트랜잭션에는 비동기 호출이 포함되는 경우가 많으므로 "시퀀스 뷰"를 클릭하여 트랜잭션의 일부로 비동기적으로 실행된 프로세스에 대한 트랜잭션 작업 흐름을 확인하면 유용합니다. "시퀀스 뷰"에는 프로세스가 실행되는 순서가 시퀀스를 식별할 수 있는 범위까지 표시됩니다. Oracle Service Bus 트랜잭션의 경우 시퀀스가 반드시 전통적인 호출자-피호출자 관계를 나타내는 것은 아니지만 하나의 프로세스가 다른 프로세스의 실행을 트리거하는 시점을 보여 줍니다.

참고: 트랜잭션 추적에 대한 자세한 내용은 [SOA 환경에서 트랜잭션 추적 사용](#) (페이지 101)을 참조하십시오. 에이전트에 대한 추적의 구성에 대한 자세한 내용은 *CA APM Java Agent* 구현 안내서 또는 *CA APM .NET* 에이전트 구현 안내서를 참조하십시오. 트랜잭션 추적 뷰와 기록 데이터 작업에 대한 자세한 내용은 *CA APM Workstation* 사용자 안내서를 참조하십시오.

제 8 장: TIBCO BusinessWorks 모니터링

TIBCO BusinessWorks 는 여러 인프라 구성 요소와 기능으로 구성된 SOA 플랫폼입니다. CA APM for TIBCO BusinessWorks 를 사용하면 프로세스 스타터, 작업 인스턴스 및 전송 프로토콜을 포함하여 TIBCO BusinessWorks 처리 엔진의 여러 핵심 요소를 모니터링 할 수 있습니다.

이 단원에서는 TIBCO BusinessWorks 환경의 성능, 가용성 및 전반적인 건전성을 모니터링하고 분석하는 데 사용할 수 있는 TIBCO BusinessWorks 특정 대시보드, 메트릭 및 경고에 대해 설명합니다.

이 섹션은 다음 항목을 포함하고 있습니다.

- [TIBCO BusinessWorks 정보](#) (페이지 151)
- [TIBCO BusinessWorks 모니터링을 사용하도록 설정하는 방법](#) (페이지 153)
- [대시보드를 사용하여 TIBCO BusinessWorks 모니터링](#) (페이지 161)
- [TIBCO BusinessWorks 에 대한 메트릭 이해 및 보기](#) (페이지 164)
- [기본 TIBCO BusinessWorks 메트릭 그룹화 보기](#) (페이지 185)
- [기본 TIBCO BusinessWorks 경고 보기](#) (페이지 185)
- [TIBCO BusinessWorks 종속성 보기](#) (페이지 186)
- [TIBCO BusinessWorks 에 대한 트랜잭션 추적](#) (페이지 189)
- [BusinessWorks 의 프런트엔드 및 백엔드 정보](#) (페이지 192)
- [메트릭 만료 처리 및 제거 사용자 지정](#) (페이지 195)
- [TIBCO BusinessWorks 를 위한 상호 관련된 추적 구성](#) (페이지 196)

TIBCO BusinessWorks 정보

TIBCO BusinessWorks 는 회사가 기존 서비스와 레거시 시스템을 표시 및 통합하고, 새로운 서비스를 설계 및 테스트하고, 느슨하게 결합된 서비스를 조정하고 응용 프로그램에 어셈블할 수 있게 도와줍니다.

BusinessWorks 는 통합 플랫폼으로서 여러 개의 분산 하위 시스템을 포함합니다. 예를 들어 TIBCO BusinessWorks 는 여러 메시징 서비스와 전송 프로토콜을 지원합니다.

다음과 같은 상위 수준 구성 요소에 대한 메트릭을 사용하여 TIBCO BusinessWorks 의 작업을 모니터링 할 수 있습니다.

동작

동작은 TIBCO 비즈니스 프로세스 정의에서 작업을 수행하는 개별 작업 단위입니다.

TIBCO BusinessWorks 확장을 사용하면 "Tibco" > "동작" 노드 아래에서 비즈니스 프로세스 동작의 성능과 전반적인 건전성을 모니터링 할 수 있습니다.

그룹 작업

그룹 작업은 서로 관련된 동작 집합의 작업 유형을 나타냅니다.

TIBCO BusinessWorks 확장을 사용하면 "Tibco" > "그룹 작업" 노드 아래에서 비즈니스 프로세스 동작의 성능과 전반적인 건전성을 모니터링 할 수 있습니다.

Hawk 메트릭

TIBCO Hawk 마이크로 에이전트는 로컬 에이전트에 호스트 운영 체제를 모니터링 하는 방법을 제공합니다.

TIBCO BusinessWorks 확장을 사용하면 "Tibco" > "Hawk 메트릭" 노드 아래에서 TIBCO Hawk 마이크로 에이전트가 수집한 엔진 통계를 모니터링하고 처리 할 수 있습니다.

작업

작업은 메모리 내에 생성되고 작업 풀 내에서 TIBCO BusinessWorks 엔진에 의해 실행되는 프로세스 인스턴스의 실행을 의미합니다.

TIBCO BusinessWorks 확장을 사용하면 "Tibco" > "작업" 노드 아래에서 프로세스 스타터와 작업 풀의 성능 및 전반적인 건전성을 모니터링 할 수 있습니다.

프로세스

프로세스는 특정 작업을 완료하도록 설계된 비즈니스 작업 흐름입니다. TIBCO Designer에서 정의한 프로세스에는 상위 프로세스의 일부로 실행되거나 상위 프로세스와 병렬로 실행되는 하위 프로세스가 포함될 수 있습니다. 개별 비즈니스 프로세스는 TIBCO BusinessWorks 프로세스 정의의 런타임 인스턴스입니다.

TIBCO BusinessWorks 확장을 사용하면 "Tibco" > "프로세스" 노드 아래에서 비즈니스 프로세스의 성능과 전반적인 건전성을 모니터링 할 수 있습니다.

전송

전송은 메시지 전송 및 배달에 대한 메커니즘을 정의합니다.

TIBCO BusinessWorks 확장을 사용하면 "Tibco" > "전송" 노드 아래에서 HTTP, SOAP 및 FTP 전송 프로토콜의 성능과 전반적인 건전성을 모니터링하고 "Tibco" > "전송" > "RV" 노드 아래에서 메트릭을 사용하여 Rendezvous 의 전반적인 건전성을 모니터링할 수 있습니다.

WebServices

WebServices 메트릭은 각 서비스 내에서 클라이언트 및 서버의 비즈니스 서비스 끝점과 관련 작업을 나타냅니다.

TIBCO BusinessWorks 확장을 사용하면 "WebServices" 노드 아래에서 클라이언트와 서버 웹 서비스 끝점의 성능과 전반적인 건전성을 모니터링할 수 있습니다.

TIBCO BusinessWorks 모니터링을 사용하도록 설정하는 방법

TIBCO BusinessWorks 모니터링을 사용하도록 설정하려면 다음과 같은 간략한 단계를 수행해야 합니다.

1. 지원되는 버전의 TIBCO BusinessWorks 가 설치되어 있는지 확인합니다.
참고: TIBCO BusinessWorks 요구 사항의 전체 목록은 *Compatibility Guide*(호환성 안내서)의 "SOA 성능 관리" 단원을 참조하십시오.
2. 에이전트 및 CA APM for SOA 가 설치되고 사용하도록 설정되었는지 확인합니다.
3. [TIBCO BusinessWorks 를 모니터링하도록 에이전트 프로필을 구성](#) (페이지 154)하여 에이전트가 CA APM for TIBCO BusinessWorks 를 사용할 수 있도록 설정합니다. Standalone Agent 설치 관리자를 사용하거나 응답 파일을 사용하여 에이전트에서 CA APM for TIBCO BusinessWorks 를 사용하도록 설정한 경우에는 이 단계를 건너뛸 수 있습니다.
4. [에이전트를 사용하도록 TIBCO BusinessWorks 를 구성합니다.](#) (페이지 158)
5. [Enterprise Manager extension for TIBCO BusinessWorks 를 사용하도록 설정합니다.](#) (페이지 161)

TIBCO BusinessWorks 모니터링을 위하여 에이전트를 사용하도록 설정

에이전트를 설치할 때 TIBCO BusinessWorks 모니터링을 사용하도록 설정하고 응용 프로그램 서버로 기본값을 선택하거나 에이전트 설치 후 수동으로 설정할 수 있습니다.

에이전트를 설치할 때 CA APM for TIBCO BusinessWorks 를 사용하도록 설정한 경우에는 에이전트 프로필이 자동으로 기본 설정으로 구성되며 추가 단계가 필요하지 않습니다.

에이전트를 설치할 때 CA APM for TIBCO BusinessWorks 를 선택하지 않은 경우에는 에이전트 프로필을 수동으로 구성하여 모니터링을 사용하도록 설정해야 합니다.

수동으로 SOA extension for TIBCO BusinessWorks 를 사용하도록 설정하려면

1. 에이전트 및 CA APM for SOA 가 설치되고 사용하도록 설정되었는지 확인합니다.
2. CA APM for TIBCO BusinessWorks 디렉터리 *SOAExtensionForTibcoBW* 가 *<Agent_Home>/examples* 디렉터리에 있는지 확인합니다.
3. 파일을 *<Agent_Home>/examples/SOAExtensionForTibcoBW/ext* 디렉터리에서 *<Agent_Home>/core/ext* 디렉터리로 복사합니다.
4. *IntroscopeAgent.profile* 파일을 텍스트 편집기에서 엽니다.
5. 표준 또는 전체 계측에 대한 적절한 지시문 파일을 *IntroscopeAgent.profile* 파일의 *introscope.autoprobe.directivesFile* 속성에 추가합니다.
표준 또는 전체 계측의 선택에 대한 자세한 내용은 [표준 및 전체 계측 정보](#) (페이지 155)를 참조하십시오. *ProbeBuilder* 지시문 파일을 사용하여 추적을 사용자 지정하는 방법에 대한 자세한 내용은 [TIBCO BusinessWorks 에 대한 지시문 파일 정보](#) (페이지 157)를 참조하십시오.
6. 변경 사항을 *IntroscopeAgent.profile* 파일에 저장하고 텍스트 편집기를 닫습니다.

표준 계측 및 전체 계측 정보

IntroscopeAgent.profile 의 *introscope.autoprobe.directivesFile* 속성을 설정할 때 기본값인 표준 계측을 선택하거나 전체 계측을 선택하여 에이전트를 배포하는 환경에 적합하게 모니터링 수준, 메트릭 가시성 및 성능 오버헤드를 제어할 수 있습니다. 그런 다음 특정 추적 프로그램 그룹에 대한 추적을 설정하거나 해제하여 표준 또는 전체 전환 파일을 통해 특정 구성 요소에 대한 모니터링을 세밀하게 조정할 수 있습니다.

tibcobw-full.pbl

모든 TIBCO BusinessWorks 구성 요소의 작업에 대한 세부적인 가시성을 위해 전체 계측을 제공합니다.

전체 계측은 세부적인 보고 기능을 제공하지만 오버헤드가 많이 발생합니다. 전체 계측은 테스트 또는 개발 환경에 권장되며 프로덕션 환경에는 일반적으로 권장되지 않습니다.

tibcobw-full.pbl 파일 목록에는 다음과 같은 지시문 파일이 포함됩니다.

- tibcobw-toggles-full.pbd
- tibcobw-webservices.pbd
- tibcobw-processes.pbd
- tibcobw-tasks.pbd
- tibcobw-RV.pbd
- tibcobw-activities.pbd
- tibcobw-jobs.pbd
- tibcobw-correlation.pbd
- tibcobw-transports.pbd
- tibcobw-hawk.pbd

tibcobw-typical.pbl

오버헤드가 매우 중요한 프로덕션 환경에서 중요 TIBCO BusinessWorks 구성 요소에 대한 표준 모니터링을 정의합니다.

표준 계측은 세부 수준은 낮지만 오버헤드를 줄일 수 있으며 프로덕션 환경에 권장됩니다.

tibcobw-typical.pbl 파일 목록에는 다음과 같은 지시문 파일이 포함됩니다.

- tibcobw-toggles-typical.pbd
- tibcobw-webservices.pbd
- tibcobw-processes.pbd
- tibcobw-tasks.pbd
- tibcobw-RV.pbd
- tibcobw-activities.pbd
- tibcobw-correlation.pbd

tibcobw-toggles-full.pbd

전체 계측을 사용할 때 TIBCO BusinessWorks 구성 요소에 대한 모니터링을 켜거나 끄도록 전환합니다.

이 파일은 다른 지시문 파일에 제공되는 추적 기능에 필요한 지시문을 제공합니다. 기본적으로 이 파일에 있는 대부분의 추적 프로그램 그룹은 사용되도록 설정됩니다.

tibcobw-toggles-typical.pbd

표준 계측을 사용할 때 TIBCO BusinessWorks 구성 요소에 대한 모니터링을 켜거나 끄도록 전환합니다. 이 파일은 다른 지시문 파일에 제공되는 추적 기능에 필요한 지시문을 제공합니다. 기본적으로 추적 프로그램 그룹의 일부만 사용되도록 설정됩니다.

지시문 파일에 대한 자세한 내용은 [TIBCO BusinessWorks 지시문 파일 정보](#) (페이지 157)를 참조하십시오.

TIBCO BusinessWorks 에 대한 지시문 파일 정보

tibcobw-toggles-full.pbd 및 *tibcobw-toggles-typical.pbd* 파일을 사용하면 ProbeBuilder 지시문(*.pbd*) 파일 집합에서 제공되는 기본 추적을 제어할 수 있습니다. 또한 필요에 따라 다음 파일의 설정을 수정하여 제공된 추적을 직접 사용자 지정할 수도 있습니다.

tibcobw-activities.pbd

비즈니스 프로세스 내의 동작 및 그룹화된 동작 작업에 대한 모니터링 규칙을 제공합니다.

tibcobw-correlation.pbd

TIBCO BusinessWorks 구성 요소에 대한 프로세스 간 트랜잭션 추적을 사용하도록 설정합니다.

tibcobw-hawk.pbd

TIBCO BusinessWorks 내에서 Hawk 메트릭에 대한 모니터링 규칙을 제공합니다.

tibcobw-jobs.pbd

TIBCO BusinessWorks에서 실행 중인 작업 및 작업 풀에 대한 모니터링 규칙을 제공합니다.

tibcobw-processes.pbd

프로세스 스타터, 배치된 하위 프로세스와 배치되지 않은 하위 프로세스를 포함하여 TIBCO BusinessWorks 비즈니스 프로세스 정의에 대한 모니터링 규칙을 제공합니다.

tibcobw-RV.pbd

BusinessWorks 내의 Rendezvous 메시징 서비스에 대한 모니터링 규칙을 제공합니다.

tibcobw-tasks.pbd

TIBCO BusinessWorks 비즈니스 프로세스 내의 작업에 대한 모니터링 규칙을 제공합니다.

tibcobw-transports.pbd

FTP, HTTP 및 SOAP 와 같은 개별 전송 프로토콜에 대한 모니터링 규칙을 제공합니다.

tibcobw-webservices.pbd

BusinessWorks 웹 서비스 끝점에 대한 모니터링 규칙을 제공합니다.

참고: 추적 구성에 대한 자세한 내용은 *CA APM Java Agent* 구현 안내서 또는 *CA APM .NET* 에이전트 구현 안내서를 참조하십시오.

에이전트를 사용하도록 TIBCO BusinessWorks 를 구성합니다.

설치 관리자를 통해 또는 설치 후 수동으로 에이전트를 사용하도록 설정한 경우에는 에이전트를 실행하여 응용 프로그램을 계측하도록 TIBCO BusinessWorks 를 구성해야 합니다. 구성 단계는 환경에서 아카이브를 배포하는 방식에 따라 다릅니다.

TIBCO Designer에서 프로세스 정의를 생성할 때는 이를 **프로젝트 아카이브(EAR)**로 내보냅니다. 각 프로젝트 아카이브는 하나 이상의 **프로세스 아카이브(PAR)**로 구성될 수 있습니다.

일반적인 TIBCO BusinessWorks 배포에서는 각 프로세스 아카이브가 자체 JVM 내에서 실행됩니다. 해당 환경에서 각 프로세스 아카이브가 자체 JVM에서 실행되는 경우에는 다음 단계에 따라 에이전트를 사용하도록 TIBCO BusinessWorks 를 구성할 수 있습니다.

서비스 컨테이너 없이 TIBCO BusinessWorks 를 구성하려면

1. TIBCO BusinessWorks 의 *bwengine.tra* 파일 또는 배포된 응용 프로그램의 *<application>.tra* 파일을 열고 *java.extended.properties* 속성에 다음 항목을 추가합니다.
java.extended.properties=-javaagent:<Agent_Home>/Agent.jar -Dcom.wily.introscope.agentProfile=<Agent_Home>/core/config/IntroscopeAgent.profile

참고: UNIX에서는 TIBCO BusinessWorks 의 *java.extended.properties*에서 콜론(:)과 등호(=)를 이스케이프하십시오.

응용 프로그램을 배포한 경우에는 *bwengine.tra* 파일을 변경한 후에 이 응용 프로그램을 제거했다가 다시 배포하십시오. 또는 개별 응용 프로그램의 *.tra* 파일에 속성을 추가할 수도 있습니다. 예를 들어 *<TIBCO_DOMAIN_HOME>/<DOMAIN_NAME>/application/<application_name>* 디렉터리의 *<application_name>.tra* 파일에 속성을 추가할 수 있습니다.

2. SOA extension for TIBCO BusinessWorks 를 설치한 컴퓨터에서 응용 프로그램에 대해 실행 중인 모든 서비스 인스턴스를 중지했다가 다시 시작합니다.

각 프로세스 아카이브를 자체 JVM에서 실행하는 대신 TIBCO BusinessWorks 서비스 컨테이너를 배포하도록 선택할 수 있습니다. 서비스 컨테이너를 사용하면 여러 프로세스 아카이브를 단일 JVM에서 실행할 수 있습니다. 서비스 컨테이너를 사용하여 프로세스 아카이브를 배포하는 경우에는 다음 단계를 통해 에이전트를 사용하도록 TIBCO BusinessWorks를 구성할 수 있습니다.

서비스 컨테이너를 사용할 때 TIBCO BusinessWorks를 구성하려면

1. <TIBCO_BW_HOME>/bin 디렉터리의 *bwcontainer.tra* 파일을 열고 *java.extended.properties* 속성에 다음 항목을 추가합니다.

```
java.extended.properties=-javaagent:<Agent_Home>/Agent.jar
-Dcom.wily.introscope.agentProfile=<Agent_Home>/core/config/IntroscopeAgent.profile
-Dcom.wily.introscope.agent.agentName=<Agent_Name>
```

참고: UNIX에서는 TIBCO BusinessWorks의 *java.extended.properties*에서 콜론(:)과 등호(=)를 이스케이프하십시오.
2. 배포된 응용 프로그램의 *bwengine.tra* 또는 <application>.tra 파일에 에이전트가 구성되지 않았는지 확인합니다.
3. 서비스 컨테이너 인스턴스를 중지했다가 다시 시작합니다.

자동 에이전트 이름 지정 구성

대부분의 TIBCO BusinessWorks 도메인에는 여러 프로세스 아카이브(PAR)에 배포된 여러 비즈니스 프로세스가 있습니다. 각 프로세스 아카이브는 자체 JVM 인스턴스에서 실행되므로 각 프로세스 아카이브는 자체 에이전트 인스턴스를 시작합니다.

각 응용 프로그램의 .tra 파일에서 *com.wily.introscope.agent.agentName* 속성을 사용하여 각 에이전트 인스턴스에 이름을 수동으로 지정할 필요가 없도록, SOA extension for TIBCO BusinessWorks는 TIBCO BusinessWorks 도메인 이름의 접두사와 밑줄(_) 및 응용 프로그램을 차례로 사용하여 에이전트 이름을 동적으로 지정할 수 있는 구성 파일과 속성을 제공합니다. 예를 들어 TIBCO 도메인 이름이 *caDomain*이고 응용 프로그램 이름이 *soap_over_http-Process_Archive*인 경우 자동 에이전트 이름 지정을 사용하도록 설정하면 에이전트 이름은 다음과 같이 동적으로 형식 지정됩니다.

caDomain_soap_over_http-Process_Archive

프로세스 아카이브의 이름 속성이 없으면 에이전트 이름은 배포 이름을 사용하여 자동으로 지정됩니다. 예:

caDomain_soap_over_http

자동 에이전트 이름 지정을 구성하려면

1. 실행 중인 TIBCO BusinessWorks 응용 프로그램을 모두 중지합니다.
 2. *TibcoBWNaming.jar* 파일이 <Agent_Home>/core/ext 디렉터리에 있는지 확인합니다.
- 파일이 없으면 <Agent_Home>/examples/SOAExtensionForTibcoBW/ext 디렉터리에서 <Agent_Home>/core/ext 디렉터리로 복사합니다.
3. <Agent_Home>/core/config/IntroscopeAgent.profile 파일을 텍스트 편집기에서 엽니다.
 4. *introscope.agent.agentAutoNamingEnabled* 속성을 찾아서 true로 설정합니다. 예:
`introscope.agent.agentAutoNamingEnabled=true`
 5. 연결 지연 속성을 설정하여 자동 이름 지정을 시도할 때 에이전트가 Enterprise Manager 연결을 지연할 시간을 정의합니다. 예를 들어 30초 동안 연결을 지연하려면 이 속성을 다음과 같이 설정합니다.
`introscope.agent.agentAutoNamingMaximumConnectionDelayInSeconds=30`
 6. 이름 바꾸기 간격 속성을 설정하여 자동 이름 지정이 사용될 때 에이전트가 동적으로 형식 지정된 이름을 확인할 빈도를 정의합니다. 예를 들어 자동으로 생성된 에이전트의 이름을 1분마다 확인하려면 속성을 다음과 같이 설정합니다.
`introscope.agent.agentAutoRenamingIntervalInMinutes=1`
 7. 변경 사항을 *IntroscopeAgent.profile* 파일에 저장합니다.
 8. TIBCO BusinessWorks 응용 프로그램을 다시 시작합니다.

별도의 프로세스 아카이브에 배포된 TIBCO BusinessWorks 응용 프로그램에 대해서만 자동 에이전트 이름 지정을 사용해야 합니다. BusinessWorks Service Container를 사용하여 프로세스 아카이브를 배포하는 경우에는 *TibcoBWNaming.jar* 파일을 <Agent_Home>/ext 디렉터리에서 제거해야 합니다.

Enterprise Manager 확장을 사용하도록 설정

CA APM for TIBCO BusinessWorks 파일은 기본적으로 Enterprise Manager 를 설치할 때 `<EM_Home>/examples` 디렉터리에 설치됩니다. CA APM for TIBCO BusinessWorks 를 사용하도록 설정하려면 Enterprise Manager 파일을 `<EM_Home>/examples` 디렉터리에서 Enterprise Manager 홈 디렉터리의 적절한 위치로 복사하거나 이동해야 합니다.

참고: SOA extension for TIBCO BusinessWorks 를 사용하려면 Enterprise Manager에서 CA APM for SOA 를 사용하도록 설정해야 합니다. CA APM for SOA Enterprise Manager 확장을 사용하도록 설정하는 자세한 방법은 [Enterprise Manager에서 확장을 사용하도록 설정](#) (페이지 41)을 참조하십시오.

다음 단계를 수행하십시오.

1. CA APM for TIBCO BusinessWorks 디렉터리인 `SOAExtensionForTibcoBW` 가 `<EM_Home>/examples` 디렉터리에 있는지 확인한 다음 `<EM_Home>/examples/SOAExtensionForTibcoBW` 디렉터리의 파일을 Enterprise Manager 디렉터리 구조의 해당 위치로 복사합니다. 예를 들어 `<EM_Home>/examples/SOAExtensionForTibcoBW/ext` 디렉터리의 파일을 `<EM_Home>/ext` 디렉터리로 복사합니다.
2. Enterprise Manager 가 클러스터 환경의 수집기인 경우 CA APM for TIBCO BusinessWorks 관리 모듈 `TibcoBWManagementModule.jar` 을 `<EM_Home>/config/modules` 디렉터리에서 제거하십시오.
관리 모듈만 MOM 컴퓨터로 사용 중인 Enterprise Manager 의 `<EM_Home>/config/modules` 디렉터리로 복사해야 합니다. 다른 모든 파일과 스크립트는 Collector Enterprise Manager 와 MOM Enterprise Manager 모두에 설치해야 합니다.
3. Workstation 을 다시 시작하여 CA APM for TIBCO BusinessWorks 의 고유한 대시보드 및 "개요" 템을 로드합니다.

대시보드를 사용하여 TIBCO BusinessWorks 모니터링

SOA extension for TIBCO BusinessWorks 에는 응용 프로그램 환경의 전반적인 건전성을 모니터링하는 데 사용할 수 있는 몇 개의 사전 구성된 대시보드가 포함되어 있습니다. 대시보드는 배포된 에이전트에서 데이터를 집계하여 성능 정보를 요약하며 문제를 신속하게 진단하고 해결할 수 있도록 해 줍니다.

일반적으로 대시보드는 다음과 같은 작업을 가능하게 하므로 환경을 모니터링하는 시작점으로 사용됩니다.

- TIBCO BusinessWorks 의 주요 구성 요소에 대한 전반적인 건전성, 성능, 가용성 및 현재 상태를 한눈에 모니터링
- 하위 수준 메트릭이 주의 또는 위험 임계값이 초과되었음을 알리는 경우 프로덕션 응용 프로그램 환경의 잠재적 문제를 조기에 식별
- 성능 정보를 드릴다운하여 지연 중이거나 오류가 발생 중인 TIBCO BusinessWorks 프로세스, 전송 프로토콜 또는 웹 서비스를 격리 및 식별

사전 구성된 TIBCO BusinessWorks 대시보드는 Enterprise Manager extension for TIBCO BusinessWorks 에 SOA Extension for TIBCO BusinessWorks 관리 모듈(*TibcoBWMManagementModule.jar*)의 일부로 패키지되어 있습니다.

TIBCO BusinessWorks 관리 모듈은 TIBCO BusinessWorks 에 대해 다음과 같은 사전 구성된 대시보드를 제공합니다.

Tibco BW - 개요

TIBCO BusinessWorks 의 주요 동작에 대한 최상위 수준 개요로, 모든 웹 서비스의 전체 응답 시간, 오류, 중단 및 SOAP 결함에 대한 경고 표시기와 비즈니스 프로세스의 전체 응답 시간, 오류 및 중단에 대한 경고 표시기를 포함합니다.

Tibco BW - 비즈니스 프로세스

모든 비즈니스 프로세스의 요약 상태로, 비즈니스 프로세스의 응답 시간, 오류 수 및 중단 수에 대한 경고 표시기와 그래프 및 가장 느린 비즈니스 프로세스의 목록을 포함합니다.

Tibco BW - 동작

모든 동작의 요약 상태로, 동작의 응답 시간, 오류 수 및 중단 수에 대한 경고 표시기와 그래프, 그리고 가장 느린 동작의 목록을 포함합니다.

Tibco BW - 작업

모든 작업 및 작업 풀에 대한 요약 상태로, 생성된 작업, 실행 중인 작업, 시간당 생성된 작업 및 완료된 작업의 수에 대한 그래프를 포함합니다.

대시보드에는 모니터링되는 프로세스 스타터의 수와 모니터링되는 작업에 대한 흐름 제한 설정을 나타내는 그래프도 포함됩니다.

Tibco BW - 전송

모든 전송 유형의 요약된 상태로, 각 전송 유형의 모든 작업에 대한 평균 응답 유형을 표시하는 그래프를 포함합니다. 예를 들어 SOAP를 사용하는 TIBCO BusinessWorks 프로세스를 모니터링 중인 경우 대시보드에는 `writeSoapEnvelope` 및 `sendMessage`와 같은 SOAP 작업에 대한 평균 응답 시간이 표시됩니다.

Workstation 콘솔을 사용하여 사전 구성된 대시보드를 볼 수 있습니다. 또한 사용자 지정 대시보드가 포함되도록 TIBCO BusinessWorks 관리 모듈을 확장하거나 사용자 지정 메트릭이나 경고가 포함되도록 기본 대시보드 정의를 수정할 수도 있습니다.

참고: 대시보드 생성과 수정에 대한 자세한 내용은 *CA APM 구성 및 관리 안내서*를 참조하십시오.

다음 단계를 수행하십시오.

1. Enterprise Manager 가 실행되고 있지 않으면 시작합니다.
2. Workstation 을 시작하고 SOA extension for TIBCO 가 설치된 Enterprise Manager 에 로그인합니다.
3. "Workstation" > "새 콘솔"을 클릭합니다.
4. "대시보드" 드롭다운 목록에서 TIBCO BusinessWorks 대시보드 중 하나를 선택합니다.

예를 들어 "Tibco BW - 개요" 대시보드를 선택하여 TIBCO BusinessWorks 프런트엔드, 백엔드, 웹 서비스 및 비즈니스 프로세스에 대한 최상위 수준 경고 표시기를 표시합니다.

5. 관련 대시보드를 열어서 더 자세한 정보를 보려면 다른 탭이나 경고를 두 번 클릭합니다.
- 예를 들어 "비즈니스 프로세스" 탭이나 "비즈니스 프로세스 응답 시간" 경고를 두 번 클릭하면 가장 느린 비즈니스 프로세스와 모든 비즈니스 프로세스의 평균 응답 시간, 간격당 오류 수 및 중단 수에 대한 더 자세한 정보를 볼 수 있습니다.

6. 대시보드에서 특정 비즈니스 프로세스, 동작 또는 작업 이름을 두 번 클릭하면 추가 분석을 위한 **Investigator** 가 열립니다. 예를 들어 "Tibco - 비즈니스 프로세스" 대시보드에서 가장 느린 프로세스를 두 번 클릭하면 해당 프로세스를 표시하는 **Investigator** 가 열립니다.

참고: **Investigator**에서 TIBCO 특정 정보를 보는 방법에 대한 자세한 내용은 [TIBCO BusinessWorks 에 대한 메트릭 이해 및 보기](#) (페이지 164)를 참조하십시오.

Workstation 시작과 사용, 대시보드 액세스 또는 **Investigator** 열기와 탐색에 대한 자세한 내용은 [CA APM Workstation 사용자 안내서](#)를 참조하십시오.

TIBCO BusinessWorks 에 대한 메트릭 이해 및 보기

Investigator 트리에서 탐색할 때는 TIBCO BusinessWorks 인프라의 구성 요소 대부분에 대한 표준 CA Introscope 메트릭을 볼 수 있습니다. 표준 메트릭에 대한 데이터는 **Investigator** 트리에서 노드 및 하위 노드로 표시되는 TIBCO 특정 메트릭 범주로 수집되고 집계됩니다. 표시되는 특정 메트릭 범주 및 노드 이름은 환경에 배포한 프로세스, 서비스 및 리소스에 따라 다릅니다.

Investigator 트리를 탐색할 때는 선택한 노드에 따라 개별 작업에 대한 하위 수준 메트릭 또는 집계 메트릭을 볼 수 있으므로 여러 TIBCO BusinessWorks 구성 요소의 전반적인 건전성을 모니터링 할 수 있습니다.

CA APM for SOA에서 제공하는 표준 및 추가 메트릭에 대한 개요는 [사용할 수 있는 메트릭](#) (페이지 58)을 참조하십시오. TIBCO BusinessWorks 또는 특정 BusinessWorks 구성 요소 모니터링에 특정한 메트릭에 대한 추가 정보를 보려면 다음 메트릭 범주를 참조하십시오.

- [동작에 대한 메트릭](#) (페이지 166)
- [Group Actions\(그룹 작업\)에 대한 메트릭](#) (페이지 166)
- [Hawk에 대한 메트릭](#) (페이지 167)
- [작업 및 작업 풀에 대한 메트릭](#) (페이지 174)
- [프로세스에 대한 메트릭](#) (페이지 177)
- [전송에 대한 메트릭](#) (페이지 180)
- [WebServices에 대한 메트릭](#) (페이지 183)

Investigator에서 TIBCO BusinessWorks 메트릭을 보고 탐색하려면

1. 에이전트 노드를 확장한 다음 Tibco 노드를 클릭하면 현재 모니터링 중인 모든 TIBCO BusinessWorks 프로세스에 대한 요약 정보가 나열되는 "개요" 탭이 표시됩니다.
2. 프로세스를 두 번 클릭하여 해당 프로세스에 대한 모든 메트릭을 그래픽 형식으로 표시합니다.
계측의 수준에 따라, 선택된 프로세스와 연관된 모든 작업의 목록을 표시할 수도 있습니다.
3. "Tibco" 노드를 확장하여 최상위 수준 TIBCO BusinessWorks 메트릭 범주에 대한 하위 노드를 표시합니다.
4. 하위 노드를 클릭하거나 확장하여 해당 메트릭 범주의 요약 정보가 있는 "개요" 탭을 표시합니다. 예를 들어 "작업" 노드를 클릭하면 "개요" 탭에 작업 풀 메트릭이 표시됩니다.
전송에 대한 개요 정보를 보려면 "전송" 노드를 확장한 다음 "FTP", "HTTP" 또는 "SOAP" 노드를 선택하여 "개요" 탭을 표시하십시오.
Rendezvous 메트릭의 경우 "전송" > "RV"를 확장한 다음 하위 노드를 선택하여 "개요" 탭을 표시합니다.
5. 원하는 하위 노드를 확장하여 메모리의 하위 프로세스, 동작 또는 작업 및 서로 연관된 메트릭 등과 같은 개별 구성 요소에 대한 세부 정보를 표시합니다.

동작에 대한 메트릭

동작은 TIBCO 비즈니스 프로세스 정의에서 개별 작업 단위를 가리킵니다. 동작은 외부 시스템과 상호 작용하는 작업인 경우가 많지만 내부 처리에 사용될 수도 있습니다. TIBCO Designer에서 동작은 유형별로 그룹화되며 선택한 팔레트에서 사용할 수 있습니다. 예를 들어 프로세스 호출, 타이머 설정 또는 로그 파일 쓰기와 같은 일반적인 작업에 대한 "General Activities"(일반 동작) 팔레트를 사용하여 파일 처리 동작을 정의할 수 있습니다.

Investigator 트리에서 동작은 수행될 동작의 유형에 따라 그룹화되며 노드 이름은 동작 유형에 대한 특정 작업을 반영합니다. 예를 들어 Investigator의 "파일" 노드에는 FileCreateActivity, FileReadActivity 및 FileWriteActivity에 대한 메트릭이 포함될 수 있습니다. 노드 이름은 일부 동작에 대한 TIBCO Designer에 사용되는 팔레트 이름과 비슷할 수 있지만 항상 그렇지는 않습니다. 예를 들어 "General Activities"(일반 동작) 팔레트의 동작은 Investigator 트리에서 "호출 프로세스", "오류 처리" 또는 "흐름" 노드 아래에 나열될 수 있습니다.

"Tibco" > "동작" 노드 아래에서 TIBCO BusinessWorks 동작에 대해 모든 표준 CA Introscope 메트릭을 사용할 수 있습니다.

Group Actions(그룹 작업)에 대한 메트릭

그룹은 특정 유형의 작업을 수행하기 위해 함께 수행되는 동작 집합을 생성하는 데 사용됩니다. 예를 들어 조건이 true가 될 때까지 반복되거나, 조건부로 실행되거나, 동일한 트랜잭션에 참가하고 함께 커밋 및 롤백되는 동작 집합을 정의하기 위해 그룹을 사용할 수 있습니다. Group Actions(그룹 작업)은 if-then 조건을 실행하는 동작 집합에 대한 "If 그룹" 또는 시퀀스의 모든 항목에 대해 반복되는 동작 집합에 대한 "반복 루프 그룹"과 같이 동작이 참가하는 작업의 유형을 식별합니다.

Group Actions(그룹 작업)에 대한 메트릭은 트랜잭션 커밋이나 롤백 또는 일련의 단계 반복과 같은 작업을 실행 중인 그룹의 성능을 모니터링합니다. 메트릭은 그룹 내 개별 동작에서 집계된 데이터를 기반으로 하지 않습니다.

"Tibco" > "Group Actions"(그룹 작업) 노드 아래에서 TIBCO BusinessWorks 그룹에 대해 모든 표준 CA Introscope 메트릭을 사용할 수 있습니다.

Hawk에 대한 메트릭

TIBCO Hawk 마이크로 에이전트는 BusinessWorks 처리 엔진, 프로세스 정의와 동작 메트릭 및 메모리 사용량에 대한 모니터링을 제공합니다.

"Tibco" > "Hawk 메트릭" 노드 아래의 다음과 같은 메트릭 범주를 사용하여 Hawk 마이크로 에이전트 메트릭을 볼 수 있습니다.

- [ExecInfo](#) (페이지 167)(실행 정보) 메트릭은 시스템 작동 시간이나 실행 중인 스레드 수와 같이 프로세스 엔진의 실행 작업에 대한 정보를 제공합니다.
- [MemoryUsage](#) (페이지 168)(메모리 사용) 메트릭은 사용 가능한 메모리와 사용한 메모리를 나타냅니다.
- [ProcessDefs](#) (페이지 168)(프로세스 정의) 메트릭은 개별 프로세스 정의 및 각 프로세스 정의에 대해 실행된 동작에 대한 세부 정보를 제공합니다.
- [프로세스](#) (페이지 171) 메트릭은 활성 프로세스 인스턴스에 대한 요약 정보를 제공합니다. 이러한 메트릭은 Hawk 버전 4.8.1 에만 적용됩니다.
- [ServicesInfo](#) (페이지 172)(서비스 정보) 메트릭은 배포된 모든 서비스에 대한 정보를 제공합니다.
- [상태](#) (페이지 173) 메트릭은 프로세스 엔진의 상태에 대한 일반적인 정보를 제공합니다.

실행 메트릭

다음은 "Tibco" > "Hawk 메트릭" > "ExecInfo"(서비스 정보) 노드 아래에서 TIBCO BusinessWorks 에 대해 사용할 수 있는 메트릭입니다.

Status

실행 엔진의 현재 상태입니다. 이 메트릭 값은 모니터링되는 서버에서 엔진의 상태가 ACTIVE, SUSPENDED, STANDBY 또는 STOPPING 인지를 나타냅니다.

스레드

엔진의 작업자 스레드 수입니다.

Uptime(작동 시간)

프로세스 엔진을 시작한 후 경과된 총 시간입니다.

버전

모니터링되는 서버의 구성 버전 정보입니다.

메모리 사용량 메트릭

다음은 "Tibco" > "Hawk 메트릭" > "MemoryUsage"(서비스 정보) 노드 아래에서 TIBCO BusinessWorks 에 대해 사용할 수 있는 메트릭입니다.

FreeBytes(사용 가능한 바이트)

사용 가능한 총 바이트 수입니다.

PercentUsed

현재 사용 중인 총 바이트의 백분율입니다.

TotalBytes(총 바이트)

엔진 프로세스에 할당된 총 바이트 수입니다.

UsedBytes(사용 중인 바이트)

현재 사용 중인 총 바이트 수입니다.

프로세스 정의 및 동작 메트릭

다음은 개별 프로세스 하위 노드의 "Tibco" > "Hawk 메트릭" > "ProcessDefs"(프로세스 정의) 노드 아래에서 TIBCO BusinessWorks 에 대해 사용할 수 있는 메트릭입니다.

Aborted(중단됨)

선택한 프로세스 정의를 중단한 횟수입니다.

AverageElapsed(평균 경과 시간)

선택한 프로세스 정의를 사용하여 완료된 모든 프로세스의 평균 경과 시간(밀리초)입니다.

AverageExecution(평균 실행 시간)

선택한 프로세스 정의를 사용하여 완료된 모든 프로세스의 평균 실행 시간(밀리초)입니다.

Checkpointed(검사점 확인 횟수)

선택한 프로세스 정의에 대해 검사점을 확인한 횟수입니다.

Completed(완료됨)

선택한 프로세스 정의가 완료된 횟수입니다.

CountSinceReset(다시 설정 후 완료된 수)

마지막 다시 설정 이후 완료된 프로세스 수입니다.

생성됨

선택한 프로세스 정의에 대해 만들어진 프로세스 수입니다.

MaxElapsed(최대 경과 시간)

선택한 프로세스 정의를 사용하여 완료된 모든 프로세스의 최대 경과 시간(밀리초)입니다.

MaxExecution(최대 실행 시간)

선택한 프로세스 정의를 사용하여 완료된 모든 프로세스의 최대 실행 시간(밀리초)입니다.

MinElapsed(최소 경과 시간)

선택한 프로세스 정의를 사용하여 완료된 모든 프로세스의 최소 경과 시간(밀리초)입니다.

MinExecution(최소 실행 시간)

선택한 프로세스 정의를 사용하여 완료된 모든 프로세스의 최소 실행 시간(밀리초)입니다.

MostRecentElapsedTime(가장 최근의 경과 시간)

가장 최근의 경과 시간(밀리초)입니다.

MostRecentExecutionTime(가장 최근의 실행 시간)

가장 최근의 실행 시간(밀리초)입니다.

이름

프로세스 정의의 이름입니다.

큐에 저장됨

선택한 프로세스 정의를 큐에 배치한 횟수입니다.

Starter(스타터)

선택한 프로세스 정의의 프로세스 스타터 동작 이름입니다.

Suspended(일시 중지됨)

선택한 프로세스 정의를 사용하는 프로세스를 일시 중지한 횟수입니다.

Swapped(교체됨)

선택한 프로세스 정의를 교체한 횟수입니다.

TimeSinceLastUpdate(마지막 업데이트 후 시간)

가장 최근에 값이 업데이트된 이후의 경과 시간(밀리초)입니다.

TotalElapsed(총 경과 시간)

선택한 프로세스 정의를 사용하여 완료된 모든 프로세스의 총 경과 시간(밀리초)입니다.

TotalExecution(총 실행 시간)

선택한 프로세스 정의를 사용하여 완료된 모든 프로세스의 총 실행 시간(밀리초)입니다.

다음은 개별 동작 이름에 대해 "Tibco" > "Hawk 메트릭" > "ProcessDefs"(프로세스 정의) > "process_name" 노드 아래에서 TIBCO BusinessWorks 에 대해 사용할 수 있는 메트릭입니다.

ActivityClass(동작 클래스)

선택한 동작을 구현하는 클래스의 이름입니다.

CalledProcessDefs(호출된 프로세스 정의)

선택한 동작에서 호출한 프로세스 정의의 이름입니다.

ElapsedTime(경과된 시간)

Sleep, Call 프로세스 및 Wait 동작에 소요된 모든 시간을 포함하여 선택한 동작의 모든 호출에 사용된 총 경과 시간(밀리초)입니다.

ErrorCount

선택한 동작이 오류를 반환한 횟수입니다.

ExecutionCount(실행 수)

TIBCO BusinessWorks 엔진 인스턴스에서 선택한 동작이 실행된 횟수입니다.

ExecutionCountSinceReset(다시 설정 이후 실행 수)

마지막 다시 설정 이후 선택한 동작이 실행된 횟수입니다.

ExecutionTime(실행 시간)

선택한 동작의 모든 호출에 사용된 총 실행 시간(밀리초)입니다.

이 메트릭에는 Sleep, Call 프로세스 및 Wait 동작에 소요된 대기 시간은 포함되지 않습니다.

LastReturnCode(마지막 반환 코드)

선택한 동작을 가장 최근에 실행했을 때 반환된 상태입니다. 이 메트릭의 유효한 반환 코드는 OK, ERROR, DEAD 또는 DEBUG입니다.

MaxElapsedTime(최대 경과 시간)

선택한 동작의 최대 경과 시간(밀리초)입니다.

MaxExecutionTime(최대 실행 시간)

선택한 동작의 최대 실행 시간(밀리초)입니다.

MinElapsedTime(최소 경과 시간)

선택한 동작의 최소 경과 시간(밀리초)입니다.

MinExecutionTime(최소 실행 시간)

선택한 동작의 최소 실행 시간(밀리초)입니다.

MostRecentElapsedTime(가장 최근의 경과 시간)

가장 최근의 경과 시간(밀리초)입니다.

MostRecentExecutionTime(가장 최근의 실행 시간)

가장 최근의 실행 시간(밀리초)입니다.

이름

동작의 이름입니다.

TimeSinceLastUpdate(마지막 업데이트 후 시간)

가장 최근에 값이 업데이트된 이후의 경과 시간(밀리초)입니다.

Tracing(추적)

선택한 동작에 대해 추적 기능이 설정되었는지 나타냅니다.

프로세스 메트릭

Hawk 버전 4.8.1 에 해당:

다음은 Hawk 버전 4.8.1 을 사용하는 경우에 "Tibco" > "Hawk 메트릭" > "프로세스"에서 TIBCO BusinessWorks 에 대해 사용할 수 있는 메트릭입니다.

CurrentActivityName(현재 동작 이름)

선택한 프로세스에서 현재 실행 중인 동작의 이름입니다.

CustomId(사용자 지정 ID)

선택한 프로세스의 사용자 지정 ID입니다(있는 경우).

Duration(기간)

선택한 프로세스를 시작한 이후 경과한 쿠록 시간(밀리초)입니다.

MainProcessName(기본 프로세스 이름)

기본 프로세스 정의의 이름입니다.

이름

선택한 프로세스의 프로세스 이름입니다.

StartTime(시작 시간)

프로세스가 시작된 시간(밀리초)입니다.

StarterName(스타터 이름)

선택한 프로세스 인스턴스를 시작한 프로세스 스타터의 이름입니다.

Status

선택한 프로세스의 현재 상태입니다. 예를 들어 이 메트릭은 작업이 현재 활성 상태인지 여부를 나타냅니다.

SubProcessName(하위 프로세스 이름)

선택한 프로세스의 하위 프로세스 정의 이름입니다(해당하는 경우).

TrackingId(추적 ID)

선택한 프로세스의 추적 ID입니다(있는 경우).

서비스 메트릭

다음은 "Tibco" > "Hawk 메트릭" > "ServicesInfo"(서비스 정보) 노드 아래에서 TIBCO BusinessWorks 에 대해 사용할 수 있는 메트릭입니다.

설명

서비스 설명입니다.

EndpointURL(끝점 URL)

서비스의 끝점 URL 주소입니다.

이름

서비스의 이름입니다.

PortName(포트 이름)

서비스의 포트 이름입니다.

PortTypeName(포트 유형 이름)

서비스의 포트 유형 설명입니다.

TransportType(전송 유형)

서비스의 전송 유형 설명입니다.

WSDL_Namespace

서비스의 구체적인 WSDL(Web Services Description Language) 네임스페이스입니다.

WSDL_URL

서비스의 WSDL(Web Services Description Language) URL 주소입니다.

bindingName(바인딩 이름)

서비스의 바인딩 이름입니다.

상태 메트릭

다음은 "Tibco" > "Hawk 메트릭" > "상태" 노드 아래에서 TIBCO BusinessWorks 에 대해 사용할 수 있는 메트릭입니다.

Adapter Name(어댑터 이름)

응용 프로그램의 이름입니다.

호스트

TIBCO BusinessWorks 엔진 프로세스가 실행되는 호스트 컴퓨터의 이름입니다.

Instance ID(인스턴스 ID)

엔진 인스턴스의 인스턴스 ID입니다.

New Errors(새 오류 수)

이 메트릭을 마지막으로 호출한 이후에 발생한 새 오류의 수입니다.

프로세스 ID

TIBCO BusinessWorks 엔진 프로세스의 프로세스 ID입니다.

Total Errors(총 오류 수)

시작 이후에 발생한 총 오류의 수입니다.

Uptime(작동 시간)

시작된 이후의 초입니다.

Hawk 메트릭 사용

Hawk 메트릭을 수집하는 기능은 성능에 큰 영향을 미치므로 기본적으로 사용되지 않습니다. 이 옵션을 설정하기 전에 성능 오버헤드를 고려하십시오. Hawk 메트릭 수집 기능은 Hawk 마이크로 에이전트를 사용하여 설정합니다.

다음 단계를 수행하십시오.

1. *IntroscopeAgent.profile* 파일을 텍스트 편집기에서 엽니다.
2. 전체 계측 모니터링 수준이 사용되도록 *introscope.autoprobe.directivesFile* 속성을 설정합니다. 예: *introscope.autoprobe.directivesFile=tibcobw-full.pbl*
3. 다음 Hawk 속성의 주석 처리를 제거하고 해당 속성을 true로 설정합니다.
`com.wily.soaextension.tibcobw.hawkmonitor.enabled=true`
이 속성을 true로 설정하면 TIBCO Hawk 모니터링이 설정됩니다.
4. 다음 속성의 주석 처리를 제거하여 Hawk 마이크로 에이전트가 메트릭을 수집하는 폴링 간격을 제어합니다. 예:
`com.wily.soaextension.tibcobw.hawkmonitor.frequency`
참고: 메트릭을 수집하는 기본 폴링 간격은 30 초(30000 밀리초)입니다.
5. 응용 프로그램 서버를 다시 시작합니다.
변경 내용이 적용됩니다.

작업 및 작업 풀에 대한 메트릭

프로세스 정의에 사용할 프로세스 스타터는 실행을 시작하는 데 필요한 데이터를 수신하면 작업을 만듭니다. 작업은 프로세스 인스턴스에서의 작업 실행을 나타냅니다. 대부분의 경우 작업은 메모리에 만들어지며 TIBCO BusinessWorks 엔진이 작업 풀 내에서 실행합니다. 작업 풀에 포함된 각 작업은 정해진 수의 작업을 실행할 수 있으며 그 후에는 다음 작업이 실행을 계속합니다. 작업 풀에서 실행할 수 있는 최대 작업 수는 **StepCount** 속성으로 제어합니다. BusinessWorks 엔진이 동시에 실행할 수 있는 최대 작업 수는 **ThreadCount** 속성으로 정의합니다.

작업 및 작업 풀 범주의 메트릭은 생성된 작업 수, 메모리에 있는 작업 수 및 각 작업 풀에서 실행된 작업 수를 모니터링하는 데 유용합니다. 작업 및 작업 풀에 대한 메트릭은 30 초마다 에이전트에 의해 새로 고쳐집니다.

다음은 "Tibco" > "작업" 노드 아래에서 TIBCO BusinessWorks 작업 및 작업 풀에 대해 사용할 수 있는 메트릭입니다.

작업 풀의 활성 작업

선택한 작업 풀에 있는 활성 프로세스 및 페이징되지 않은 프로세스의 총 수입니다.

작업 풀의 작업 오류 수

시작 이후 선택한 작업 풀에서 모든 작업에 대해 발생한 총 오류 수입니다.

작업 풀의 실행 중인 작업

선택한 작업 풀에서 실행 중인 프로세스와 큐에 저장된 프로세스의 총 수입니다.

스레드 수

BusinessWorks 엔진에 대해 구성된 작업자 스레드의 최대 수입니다.

완료된 작업

시작 이후 실행이 완료된 프로세스의 수입니다.

생성된 작업

시작 이후 BusinessWorks 엔진 내에 만들어진 프로세스의 수입니다.

흐름 제한

TIBCO BusinessWorks 흐름 제한 설정 값입니다.

시간당 생성된 작업

TIBCO BusinessWorks 엔진 안에서 시간당 만들어지는 프로세스의 수입니다.

프로세스 스타터 상태

작업에 대한 프로세스 스타터의 현재 상태를 나타냅니다.

실행 중인 작업

TIBCO BusinessWorks 엔진에서 실행 중인 비즈니스 프로세스의 수입니다.

각 작업 풀 이름을 확장하여 해당 프로세스 인스턴스에 대해 만들어진 작업의 하위 노드를 표시할 수 있습니다.

작업 및 작업 풀에 대한 메트릭 설정 또는 해제

전체 계측을 사용할 경우 기본적으로 작업 및 작업 풀 모니터링이 30 초의 기본 폴링 간격으로 사용됩니다. 다음과 같은 속성을 사용하면 작업 및 작업 풀에 대한 모니터링 사용 여부를 설정하고 폴링 간격의 빈도를 제어할 수 있습니다.

```
com.wily.soaextension.tibcobw.jobmonitor.enabled  
com.wily.soaextension.tibcobw.jobmonitor.frequency
```

예를 들어 표준 계측을 사용하는 경우 모니터링을 설정하려면 `com.wily.soaextension.tibcobw.jobmonitor.enabled` 속성을 `true`로 설정하고, 전체 계측을 사용하는 경우 모니터링을 해제하려면 이 속성을 `false`로 설정합니다.

```
com.wily.soaextension.tibcobw.jobmonitor.enabled=true
```

그런 다음 `com.wily.soaextension.tibcobw.jobmonitor.frequency` 속성을 사용하여 메트릭을 수집하는 폴링 간격을 제어할 수 있습니다. 예:

```
com.wily.soaextension.tibcobw.jobmonitor.frequency=30000
```

표준 계측을 사용하고 `com.wily.soaextension.tibcobw.jobmonitor.enabled` 속성을 `true`로 설정한 경우에는 `tibcobw-toggles-typical.pbd` 파일에서 다음 행의 주석 처리를 제거하여 작업 및 작업 풀 모니터링이 사용되도록 설정해야 합니다.

```
#TurnOn: JobTracing  
#TurnOn: JobPoolTracing
```

예를 들어 표준 계측 사용 시

`com.wily.soaextension.tibcobw.jobmonitor.enabled` 속성을 `true`로 설정하고 `tibcobw-toggles-typical.pbd` 파일을 수정하여 작업 및 작업 풀 모니터링 추적 프로그램이 사용되도록 설정합니다.

```
TurnOn: JobTracing  
TurnOn: JobPoolTracing
```

두 작업 모니터링 속성 중 하나라도 변경한 경우 변경 사항을 적용하려면 응용 프로그램 서버를 다시 시작해야 합니다.

프로세스에 대한 메트릭

TIBCO BusinessWorks 의 주요 구성 요소는 해당 프로세스 엔진입니다. 프로세스 엔진에는 비즈니스 프로세스를 설계, 배포 및 관리하기 위한 도구가 포함됩니다. TIBCO BusinessWorks 에서 비즈니스 프로세스는 TIBCO 프로세스 정의의 런타임 인스턴스입니다. 프로세스 정의에는 동작, 하위 프로세스 및 작업이 포함될 수 있으며 비즈니스 프로세스의 이러한 요소를 모니터링하여 프로세스 엔진을 모니터링할 수 있습니다.

"Tibco" > "프로세스" 노드 아래에서 비즈니스 프로세스, 배치된 하위 프로세스 및 작업에 대해 모든 표준 CA Introscope 메트릭을 사용할 수 있습니다. 배치되지 않은 하위 프로세스의 경우에는 "평균 응답 시간", "동시 호출", "간격당 오류 수" 및 "간격당 응답 수"만 사용할 수 있습니다.

표준 메트릭 외에도 프로세스 및 배치된 하위 프로세스에는 다음 메트릭을 사용할 수 있습니다.

실행 중인 평균 응답 시간

비즈니스 프로세스가 입력 대기 시간을 제외하고 TIBCO BusinessWorks 엔진에서 능동적으로 실행되는 평균 시간(밀리초 단위)입니다.

실행 중인 동시 호출

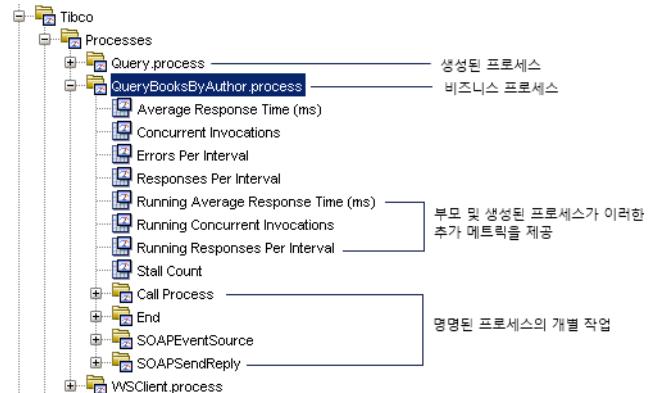
TIBCO BusinessWorks 엔진에서 능동적으로 실행 중인 비즈니스 프로세스의 수입니다.

간격당 실행 중인 응답 수

하나의 간격에서 TIBCO BusinessWorks 엔진에서 성공적으로 실행되는 비즈니스 프로세스의 수입니다.

이 메트릭은 간격 중에 완료된 비즈니스 프로세스를 나타내지 않습니다. 예를 들어 이 메트릭에는 실행이 일시 중지된 비즈니스 프로세스가 포함될 수 있습니다.

예를 들어 "Tibco" > "프로세스"를 확장하면 TIBCO Designer for TIBCO BusinessWorks에서 정의한 비즈니스 프로세스 이름이 표시됩니다. "프로세스" 노드에는 부모 비즈니스 프로세스와 같은 수준에 있는 배포된 하위 프로세스 및 배포되지 않은 하위 프로세스와 개별 작업에 대한 노드가 포함될 수 있습니다.



또한 TIBCO BusinessWorks 프로세스에 대한 종속성 및 편차 메트릭도 수집할 수 있습니다. 표준 종속성 및 편차 메트릭에 대한 자세한 내용은 [Investigator를 사용하여 SOA 성능 메트릭 보기](#) (페이지 58)를 참조하십시오.

비즈니스 프로세스의 전체 및 활성 실행 정보

비즈니스 프로세스 정의가 타이머를 포함하거나 사용자 입력을 대기하기 위해 실행을 일시 중지시킬 수 있으므로 프로세스 메트릭 범주는 다음을 측정하는 메트릭을 제공합니다.

- **비즈니스 프로세스 전체:** 이러한 메트릭은 프로세스가 일시 중지 상태인지 아니면 활성 상태인지에 관계없이 비즈니스 프로세스의 전체 처리 상태를 나타냅니다.
- **비즈니스 프로세스 활성 실행:** 이러한 메트릭은 TIBCO BusinessWorks 엔진에서 활성으로 실행되고 있는 비즈니스 프로세스에 대해서만 처리 상태를 제공합니다.

비즈니스 프로세스의 **평균 응답 시간** 메트릭은 프로세스가 입력이나 사용자 응답을 대기하느라 일시 중지된 시간을 포함하여 프로세스가 시작되어 끝날 때까지의 평균 응답 시간을 측정합니다. 예를 들어 비즈니스 프로세스가 시작된 후 BusinessWorks 엔진에서 15ms 의 활성 실행 시간이 경과한 다음 사용자 입력을 30ms 동안 대기하고 작업 완료를 위해 10ms 동안의 실행 시간이 경과한 경우 시작부터 끝날 때까지의 총 응답 시간은 15ms, 30ms 및 10ms 의 합계인 55ms 입니다. 예:

평균 응답 시간(ms) = 실행 시간 + 일시 중지 시간

"평균 응답 시간" 메트릭에는 처리가 일시 중지된 시간이 포함되기 때문에 이 메트릭은 BusinessWorks 엔진 자체의 성능을 나타내지 않습니다. 별도의 **실행 중인 평균 응답 시간** 메트릭이 비즈니스 프로세스가 BusinessWorks 엔진을 능동적으로 실행하는 데 소비한 시간을 추적합니다.

"**실행 중인 평균 응답 시간**" 메트릭은 비즈니스 프로세스의 활성 실행 시간만 측정합니다. 즉, 입력 대기에 소비한 시간은 포함하지 않습니다. 예를 들어 비즈니스 프로세스가 시작된 후 BusinessWorks 엔진이 15 ms 동안 실행되고 사용자 입력을 대기하느라 중지된 다음 BusinessWorks 엔진이 10 ms 동안 실행되어 비즈니스 프로세스를 완료한 경우 "실행 중인 평균 응답 시간"은 첫 번째 실행의 15 ms 와 두 번째 실행의 10 ms 입니다. 사용자 입력을 대기하느라 프로세스가 일시 중지된 시간은 포함되지 않습니다. 예:

실행 중인 평균 응답 시간(ms) = 총 실행 시간 - 일시 중지 시간

마찬가지로, 표준 **동시 호출 및 간격당 응답 수** 메트릭에는 프로세스가 능동적으로 실행되고 있는지, 아니면 일시 중지되었는지에 관계없이 모든 프로세스가 계산에 포함됩니다. **실행 중인 동시 호출 및 간격당 실행 중인 응답 수** 메트릭은 BusinessWorks 엔진에 실제로 배치된 처리 동작만 포함합니다.

"**실행 중인 평균 응답 시간**", "**실행 중인 동시 호출**" 및 "**간격당 실행 중인 응답 수**" 메트릭은 모두 비즈니스 프로세스 및 생성된 프로세스에만 적용할 수 있으며 생성되지 않은 하위 프로세스나 개별 작업에는 이 메트릭이 제공되지 않습니다.

비즈니스 프로세스의 중단 수 정보

비즈니스 프로세스의 경우 **중단 수** 메트릭에는 BusinessWorks 엔진에서 능동적으로 실행되는 프로세스만 포함됩니다. 사용자 입력을 대기하느라 일시 중지된 프로세스는 중단 수에 포함되지 않습니다.

하위 프로세스의 간격당 오류 수 정보

간격당 오류 수 메트릭은 특정 하위 프로세스를 실행할 때 오류가 발생하여 생성되지 않은 하위 프로세스에 대해서만 표시됩니다. 생성되지 않은 하위 프로세스를 실행할 때 오류가 발생하면 하위 프로세스와 해당 부모 프로세스의 "간격당 오류 수"가 모두 하나씩 증가합니다.

부모 프로세스와 생성된 하위 프로세스의 경우 기본적으로 "간격당 오류 수" 메트릭이 항상 표시됩니다. 생성된 하위 프로세스를 실행할 때 오류가 발생하면 해당 하위 프로세스의 "간격당 오류 수" 메트릭에 오류가 기록됩니다. 하지만 부모 프로세스의 "간격당 오류 수" 계산에는 생성된 하위 프로세스의 오류가 더해지지 않습니다.

전송에 대한 메트릭

전송은 메시지 전송 및 배달에 대한 메커니즘을 정의합니다. TIBCO BusinessWorks 의 경우 TIBCO BusinessWorks 에서 메시징의 라우팅 및 처리에 사용될 수 있는 엔터프라이즈 메시징 시스템인 TIBCO Rendezvous 와 HTTP, SOAP, JMS, FTP 및 XML 등의 표준 전송 프로토콜을 포함한 다양한 전송 프로토콜에 대한 메트릭을 수집할 수 있습니다.

TIBCO BusinessWorks 확장을 사용하면 "Tibco" > "전송" 노드 아래에서 HTTP, SOAP 및 FTP 전송 프로토콜의 성능과 전반적인 건전성을 모니터링하고 "Tibco" > "전송" > "RV" 노드 아래에서 메트릭을 사용하여 Rendezvous 의 전반적인 건전성을 모니터링 할 수 있습니다.

또한 표준 *j2ee.pbd* 파일에서 JMS 및 XML 추적을 활성화하여 JMS 및 XML 의 성능과 전반적인 건전성을 모니터링 할 수도 있습니다. *j2ee.pbd* 파일은 기본적으로 에이전트와 함께 제공되지만 TIBCO BusinessWorks 확장과 함께 사용하여 추가 모니터링을 제공할 수 있습니다.

사용 중인 SOAP 및 HTTP 와 같은 특정 전송 프로토콜에 대한 "전송" 노드 아래에서 작업에 대해 모든 표준 CA Introscope 메트릭을 사용할 수 있습니다. 그런 다음 전송 프로토콜 노드를 확장하여 드라이버, 핸들러 또는 요청자 클래스와 같은 전송 클래스를 표시한 다음 하위 노드를 확장하여 *sendMessage* 또는 *writeSoapEnvelope* 와 같은 개별 작업에 대한 메트릭을 드릴다운할 수 있습니다.

에이전트의 *j2ee.pbd* 파일을 사용하여 JMS 또는 XML 을 모니터링하는 경우에는 JMS 및 XML 전송 프로토콜 노드를 "Tibco" > "전송" 노드가 아니라 에이전트 노드 아래에서 볼 수 있습니다.

Rendezvous(RV)의 메트릭

Rendezvous 는 분산 응용 프로그램이 네트워크를 통해 데이터를 교환할 수 있게 해 주는 엔터프라이즈 메시징 시스템입니다. 통합 플랫폼인 TIBCO BusinessWorks 는 지원 메시징 서비스 중 하나인 Rendezvous 를 사용하여 대량으로 주고받는 메시지를 안정적으로 처리합니다.

Rendezvous 에 대한 메트릭은 Rendezvous 메시지의 배달, 큐 처리 및 전송과 관련된 자세한 정보를 제공합니다. TIBCO BusinessWorks 에서는 "Tibco" > "전송" > "RV" 노드 아래에 있는 다음과 같은 메트릭 범주의 메트릭을 사용하여 Rendezvous 를 모니터링 할 수 있습니다.

Message Listeners(메시지 수신기)

"Message Listeners"(메시지 수신기)는 인바운드 Rendezvous 메시지를 수신 대기하는 개체입니다. 새 인바운드 메시지가 도착하면 수신기가 "Message Processor"(메시지 처리기) 개체를 호출하여 메시지를 처리합니다.

Message Processors(메시지 처리기)

"Message Processors"(메시지 처리기)는 "Message Listeners"(메시지 수신기)로 검색된 메시지를 처리하는 콜백 메서드입니다.

Queue Groups(큐 그룹)

"Queue Groups"(큐 그룹)를 사용하면 이벤트가 발송되는 순서를 정밀하게 제어 할 수 있습니다. 발송 프로세스가 큐를 호출하여 이벤트를 발송하는 순서는 그룹에 지정된 각 큐의 상대적 우선 순위에 의해 결정됩니다.

이 범주의 메트릭은 발송 메서드의 응답 시간, 처리량 및 오류를 모니터링 합니다.

큐

"Queues"(큐)는 인바운드 메시지 또는 타이머 프로세스 같은 비동기 이벤트를 메시지 처리기가 처리할 수 있을 때까지 받은 순서대로 저장하는 데 사용됩니다. 이벤트가 발생하면 이벤트 큐에 해당 이벤트가 배치되고 발송 루틴이 처리를 위해 전송할 때까지 대기하게 됩니다.

이 범주의 메트릭은 발송 메서드의 응답 시간, 처리량 및 오류를 모니터링합니다.

전송

전송은 메시지 전송 및 배달에 대한 메커니즘을 정의합니다. TIBCO BusinessWorks 와 함께 사용하는 Rendezvous 의 경우 "신뢰할 수 있는 전송"(표준 Rendezvous 전송) 및 "인증된 전송"(RVCM) 유형에 대한 메트릭을 수집할 수 있습니다.

"Message Listener"(메시지 수신기), "Message Processor"(메시지 처리기) 및 "전송" 메트릭

"Message Listener"(메시지 수신기), "Message Processor"(메시지 처리기) 및 "Reliable and Certified"(신뢰할 수 있고 인증된) 전송 프로토콜에 대한 "Tibco" > "전송" > "RV" 하위 노드 아래에서 Rendezvous 에 대한 모든 표준 CA Introscope 메트릭을 사용할 수 있습니다.

Queue Groups(큐 그룹) 메트릭

개별 큐 그룹과 연관된 발송 메서드에 대한 "Tibco" > "전송" > "RV" > "Queue Groups"(큐 그룹) 노드 아래에서 다음의 발송 관련 메트릭을 사용할 수 있습니다.

- 평균 응답 시간(ms)
- 간격당 오류 수
- 간격당 응답 수

큐 메트릭

개별 큐와 연관된 발송 메서드에 대한 "큐" 노드 아래에서 다음의 발송 관련 메트릭을 사용할 수 있습니다.

- 평균 응답 시간(ms)
- 간격당 오류 수
- 간격당 응답 수

Rendezvous 에 대한 메트릭을 사용하거나 사용하지 않도록 설정

Rendezvous 모니터링은 30 초의 기본 폴링 간격으로 표준 또는 전체 계측을 사용할 때 기본적으로 사용됩니다. Rendezvous 모니터링을 사용하지 않으려면 *tibcobw-toggles-typical.pbd* 또는 *tibcobw-toggles-full.pbd* 파일에서 다음 행을 주석 처리하십시오.

```
TurnOn: TibrvTransport
TurnOn: TibrvDispatchable
TurnOn: TibcoRVMessageProcessorTracing
TurnOn: TibcoRVMessageListenerTracing
```

예를 들어 표준 계측을 사용하는 경우 *tibcobw-toggles-typical.pbd* 파일을 텍스트 편집기에서 열고 행을 다음과 같이 수정하여 추적을 사용하지 않도록 설정합니다.

```
#TurnOn: TibrvTransport
#TurnOn: TibrvDispatchable
#TurnOn: TibcoRVMessageProcessorTracing
#TurnOn: TibcoRVMessageListenerTracing
```

WebServices 에 대한 메트릭

"WebServices" 노드 아래에서 클라이언트 및 서버 웹 서비스 끝점에 대해 모든 표준 CA Introscope 메트릭을 사용할 수 있습니다. 또한 "간격당 SOAP 오류 수"는 SOA 플랫폼에서 웹 서비스를 모니터링하는 모든 환경에 대한 표준 메트릭입니다. "간격당 오류 수"와 "간격당 SOAP 결함 수"의 차이에 대한 자세한 내용은 "오류" 탭에서 SOAP 결함 및 오류 메트릭 보기 를 참조하십시오.

또한 TIBCO BusinessWorks 웹 서비스 작업에 대한 종속성 및 편차 메트릭도 수집할 수 있습니다. 표준 종속성 및 편차 메트릭에 대한 자세한 내용은 [Investigator 를 사용하여 SOA 성능 메트릭 보기](#) (페이지 58)를 참조하십시오.

Investigator 에서 WebServices 메트릭을 보고 탐색하려면

1. 에이전트 노드를 확장한 다음 "WebServices"를 확장하여 웹 서비스 끝점에 대한 "클라이언트" 및 "서버" 노드를 표시합니다.
2. "클라이언트" 또는 "서버" 노드를 확장하여 메트릭을 표시할 개별 웹 서비스 끝점을 표시합니다.

3. 특정 웹 서비스 끝점 `<namespace>` 또는 `<servicename>`을 확장하여 작업을 표시하거나 해당 웹 서비스에 대한 메트릭을 선택합니다.
4. 웹 서비스 내의 특정 작업을 확장하여 해당 작업에 대한 메트릭을 표시합니다.

서버 끝점의 평균 응답 시간 정보

서버 측 작업의 경우 "평균 응답 시간" 메트릭은 TIBCO BusinessWorks 가 들어오는 SOAP 메시지를 수신하고 구문 분석하는 데 걸린 시간을 나타냅니다. 대부분의 경우 이러한 처리는 보다 복잡한 트랜잭션의 일부에 불과합니다. 예를 들어 표준 BusinessWorks 트랜잭션에는 웹 서비스의 서버 측에서 수행되는 다음과 같은 단계가 포함됩니다.

- 들어오는 SOAP 메시지의 수신 및 구문 분석에 소요된 시간
- BusinessWorks 엔진이 웹 서비스가 노출하는 프로세스 정의를 실행하고 응답을 보낼 준비를 하는 데 소요된 시간
- 응답을 보내는 데 소요된 시간

트랜잭션의 각 단계가 별도의 스레드에서 실행될 수 있으므로 "평균 응답 시간" 메트릭은 각 작업에 대해 트랜잭션의 첫 부분만 측정한 다음 웹 서비스 끝점에 대한 데이터를 집계할 수 있습니다. 이 메트릭에서는 추가 스레드에서 소요된 처리 시간을 계산할 수 없으므로 이 메트릭은 전체 웹 서비스 트랜잭션의 총 응답 시간을 나타내지 않습니다.

클라이언트 및 서버 SOAP 결합 메트릭 정보

서버 측 작업("WebServices" > "서버" > 네임스페이스)의 경우 "간격당 SOAP 오류 수" 메트릭은 BusinessWorks 가 요청에 대한 응답에서 웹 서비스 클라이언트로 보낸 SOAP 결합 수를 나타냅니다.

클라이언트 측 작업("WebServices" > "클라이언트" > 네임스페이스)의 경우 "간격당 SOAP 오류 수" 메트릭은 클라이언트의 SOAPSendReceiveActivity에서 받은 SOAP 결합 수를 나타냅니다.

기본 TIBCO BusinessWorks 메트릭 그룹화 보기

SOA extension for TIBCO BusinessWorks 에는 기본 대시보드와 경고를 정의하는 데 사용되는 기본 메트릭 그룹화가 포함됩니다. 사용자 지정 대시보드와 경고에서도 이 기본 메트릭 그룹화를 사용할 수 있습니다.

기본 메트릭 그룹화는 Enterprise Manager extension for TIBCO BusinessWorks 에 SOA Extension for TIBCO BusinessWorks 관리 모듈(*TibcoBWManagementModule.jar*)의 일부로 패키지되어 있습니다.

다음 단계를 수행하십시오.

1. Investigator에서 Workstation > 새 관리 모듈 편집기를 클릭합니다.
2. "*SuperDomain*" > "관리 모듈" > "Introscope SOA Extension for TibcoBW <version>(*SuperDomain*)"을 확장합니다.
3. "메트릭 그룹화" 노드를 확장하여 TIBCO 관리 모듈에 대해 정의된 모든 메트릭 그룹화를 표시합니다.
4. 특정 메트릭 그룹화를 클릭하여 "뷰어" 창에서 해당 정의를 봅니다.

모든 메트릭 그룹화의 기본 설정을 수정하거나 사용자 지정 메트릭 그룹화를 직접 생성할 수 있습니다.

참고: 메트릭 그룹화 생성 또는 수정에 대한 자세한 내용은 *CA APM Workstation 사용자 안내서*를 참조하십시오.

기본 TIBCO BusinessWorks 경고 보기

SOA extension for TIBCO BusinessWorks 에는 사전 구성된 대시보드에 사용되는 기본 경고 정의가 포함됩니다. 이 기본 경고를 사용자 지정 대시보드에서도 사용할 수 있습니다. 대부분의 기본 경고는 기본 "주의" 및 "위험" 임계값으로 사전 구성되며 임계값이 초과되거나 심각도가 증가할 경우 콘솔에 알림을 보내도록 사전 구성됩니다.

기본 경고 정의는 Enterprise Manager extension for TIBCO BusinessWorks 에 SOA Extension for TIBCO BusinessWorks 관리 모듈(*TibcoBWManagementModule.jar*)의 일부로 패키지되어 있습니다.

TIBCO BusinessWorks 에이전트에 대한 기본 경고 정의를 보려면

1. Investigator에서 Workstation > 새 관리 모듈 편집기를 클릭합니다.
2. "*SuperDomain*" > "관리 모듈" > "Introscope SOA Extension for TibcoBW <version>(*SuperDomain*)"을 확장합니다.
3. "경고" 노드를 확장하여 TIBCO BusinessWorks 관리 모듈에 대해 정의된 모든 경고를 표시합니다.
4. 특정 경고를 클릭하여 "뷰어" 창에서 해당 정의를 봅니다.

특히 "주의" 및 "위험" 임계값의 기본 설정을 검토하여 필요한 경우 값을 조정하고 알림 또는 수정 조치를 추가하십시오.

경고에 대한 기본 설정을 수정하거나 사용자 지정 경고를 직접 생성할 수 있습니다.

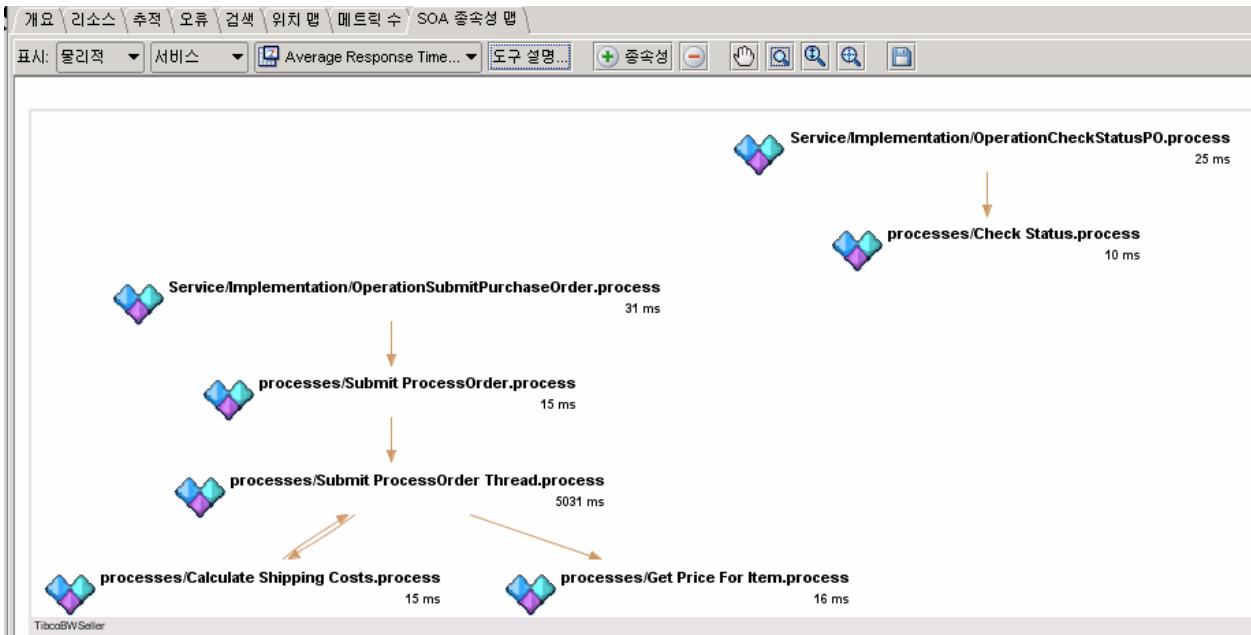
참고: 경고 생성 또는 수정에 대한 자세한 내용은 *CA APM Workstation 사용자 안내서*를 참조하십시오.

TIBCO BusinessWorks 종속성 보기

TIBCO BusinessWorks 웹 서비스, 서비스 작업 및 프로세스 정의에 대한 종속성을 보려면 트리에서 TIBCO BusinessWorks 프로세스 또는 WebServices 노드를 선택하고 "SOA 종속성 맵" 탭을 클릭하십시오.

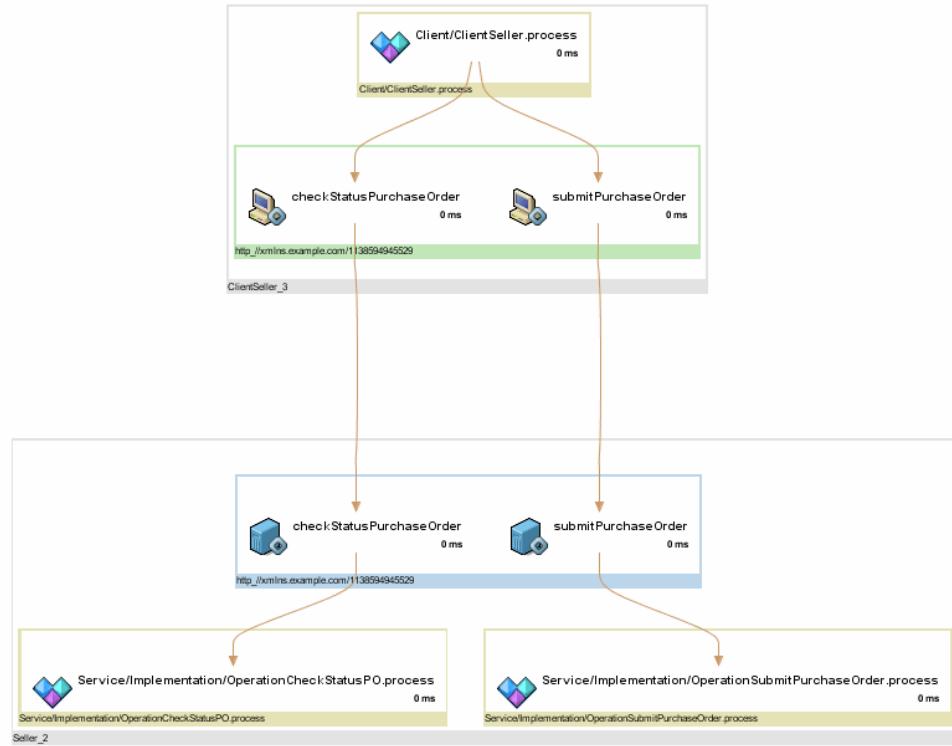
선택한 노드에 따라 종속성 맵에 표시되는 컨텍스트가 결정됩니다. 종속성 맵이 표시되면 보고자 하는 컨텍스트와 세부 정보 수준을 롤업하여 축소하거나 롤다운하여 확장할 수 있습니다.

예를 들어 에이전트에 대한 모든 비즈니스 프로세스에서 종속성의 상위 수준 뷰를 보려면 **Investigator**에서 "프로세스" 노드를 선택한 다음 "SOA 종속성 맵" 탭을 클릭합니다.



에이전트에 대한 특정 비즈니스 프로세스에서 종속성의 상위 수준 뷰를 보려면 **Investigator**에서 비즈니스 프로세스 이름을 선택한 다음 "SOA 종속성 맵" 탭을 클릭합니다. 콘텐츠 유형이나 **Investigator** 트리 노드를 변경하면 종속성 맵의 컨텍스트가 변경된다는 점에 주의하십시오. 예를 들어 콘텐츠 유형을 "서비스"에서 "작업"으로 변경하면 표시된 맵이 변경됩니다. 하지만 맵에 포함된 세부 정보는 프로세스 정의 및 **Investigator** 트리에서 선택한 노드에 따라 동일하게 유지되거나 달라질 수 있습니다.

다음 예에서는 **Investigator** 트리에서 구입 주문 요청을 선택하고 추가 종속성을 표시하도록 확장하는 비즈니스 프로세스를 보여 줍니다.



계속해서 필요에 따라 맵에 종속성 수준을 추가하여 비즈니스 프로세스의 전체 작업 흐름을 표시하거나 맵의 특정 노드를 확대할 수 있습니다. 종속성 수준의 표시 및 숨기기를 비롯하여 종속성 맵 탐색 방법에 대한 자세한 내용은 [SOA 종속성 맵 사용](#) (페이지 73)을 참조하십시오.

TIBCO BusinessWorks 에 대한 트랜잭션 추적

트랜잭션 추적은 비즈니스 트랜잭션 완료에 관련된 특정 단계의 세부 정보 뷰 또는 요약 뷰를 제공합니다. TIBCO BusinessWorks 비즈니스 프로세스 또는 웹 서비스의 경우에는 다음 프로토콜을 통해 라우팅되는 작업이 포함된 트랜잭션을 추적할 수 있습니다.

- SOAP(Simple Object Access Protocol)
- HTTP(Hypertext Transport Protocol)
- HTTPS(Hypertext Transport Protocol Secure)
- JMS(Java Message Service)

TIBCO BusinessWorks 비즈니스 프로세스 또는 웹 서비스가 연관된 트랜잭션에는 동기적 호출과 비동기적 호출 및 서로 다른 스레드를 사용하여 동시에 실행 중인 동작이 포함될 수 있습니다. 여러 스레드나 프로세스에서 실행되는 트랜잭션에 대한 트랜잭션 추적을 사용하도록 설정하기 위해 상관 관계 식별자가 삽입되어 각 구성 요소 및 작업에서 트랜잭션 단계로 사용됩니다. 그러면 수행된 특정 작업에 대한 세부 정보와 각 작업이 완료되는 데 걸린 시간을 볼 수 있습니다.

또한 추적 중인 모든 노드에서 CA APM for SOA 및 CA APM for TIBCO BusinessWorks 가 사용하도록 설정되어 있으면 모든 조합의 플랫폼에서 비즈니스 트랜잭션을 추적할 수 있습니다. 따라서 트랜잭션이 여러 JVM 또는 CLR에서 실행되는 경우에도 트랜잭션에 대한 세부 정보를 볼 수 있습니다.

TIBCO BusinessWorks 의 호출이 모니터링되지 않는 서버의 프로세스에 의해 실행되는 경우 트랜잭션 추적은 모니터링되지 않는 프로세스를 호출하는 비즈니스 프로세스의 실행 시간을 제공합니다. 따라서 모니터링되지 않는 프로세스 자체의 실행 시간이 기록되지 않더라도 회신을 받기까지 걸린 시간을 확인할 수 있습니다. 트랜잭션 추적은 오류 발생 여부도 기록합니다.

Tibco에서 트랜잭션 추적 데이터를 사용하는 방법

평균 응답 시간이 매우 큰 경우, 예를 들어 Tibco 비즈니스 프로세스에서 평균 응답 시간이 x 인 경우 시간 필터를 사용하여 x 보다 큰 트랜잭션을 필터링할 수 있습니다. 비즈니스 프로세스는 일반적으로 둘 이상의 트랜잭션으로 구성되므로 문제 영역을 진단하려면 다음 단계를 수행해야 합니다.

1. TT 창으로 이동하여 해당 비즈니스 프로세스에 대한 필터를 사용하도록 설정합니다.
2. 응용 프로그램을 호출합니다.
해당 비즈니스 프로세스에 대한 여러 트랜잭션 추적이 수행됩니다. 추적은 비즈니스 프로세스의 실행 단위입니다.
3. 긴 실행 단위가 있는지 확인합니다. 해당 추적을 드릴다운하여 자세한 정보를 확인합니다.
실행 추적 간의 시간 차이가 표시됩니다. 이 시간은 비즈니스 프로세스가 일시 중지된 시간을 나타냅니다. 이 정보를 사용하여 비즈니스 프로세스가 일시 중지된 빈도와 시간 길이를 확인합니다.
HTTP 요청이 시간 초과되면 해당 동작에서 오류를 표시합니다.

크로스 프로세스 트랜잭션 추적의 가치 이해

크로스 프로세스 트랜잭션 추적은 SOA(서비스 지향 아키텍처)에서 느슨하게 결합된 서비스가 수행하는 작업에 대한 중요한 정보를 제공합니다. 크로스 프로세스 트랜잭션 추적을 사용하면 다음을 확인할 수 있습니다.

- 비즈니스 프로세스가 구성 요소를 통해 라우팅되는 방법
- 트랜잭션 중에 호출되고 실행되는 비즈니스 프로세스
- 트랜잭션 중에 발생하는 호출의 순서
- 요청 또는 응답 처리가 가장 느린 지점

샘플 트랜잭션 추적 시작 및 보기

비즈니스 프로세스 트랜잭션에 로컬 또는 원격 컴퓨터에 있는 여러 JVM이나 CLR에 대한 호출이 포함되어 있고 호출된 JVM 또는 CLR의 에이전트가 동일한 Enterprise Manager로 호출을 보고하는 경우에는 트랜잭션 추적을 실행하여 비즈니스 프로세스의 일부로 실행된 모든 호출을 볼 수 있습니다.

트랜잭션 추적 세션은 다음과 같이 시작할 수 있습니다.

- SOA 종속성 맵의 맵 노드에서 바로 시작합니다.
- "Workstation" > "새 트랜잭션 추적 세션"을 클릭하여 Workstation에서 수동으로 시작합니다.

필터를 구성하여 트랜잭션 추적 세션에서 캡처할 트랜잭션을 제어하고 세션이 지속될 시간을 지정할 수 있습니다. 종속성 맵에서 트랜잭션 추적을 시작하는 경우에는 맵 노드 유형에 따라 기본 필터가 자동으로 설정됩니다. 새 트랜잭션 추적 세션을 수동으로 시작하는 경우에는 TIBCO BusinessWorks에 대한 다음 필터 유형 중 하나를 선택할 수 있습니다.

- 비즈니스 프로세스
- namespace
- 작업

예를 들어 특정 TIBCO BusinessWorks 프로세스 정의에 대한 트랜잭션을 필터링하려면 비즈니스 프로세스 필터를 선택하고 프로세스 정의 이름의 전체 또는 일부를 입력하십시오.

필터를 구성하고 트랜잭션 추적 세션을 시작하면 "트랜잭션 추적 뷰어"가 표시됩니다. "트랜잭션 추적 뷰어"에서는 추적을 선택하여 트랜잭션에서 생성된 호출에 대한 추가 세부 정보를 볼 수 있습니다. 예를 들어 트랜잭션을 선택한 후 다음을 수행할 수 있습니다.

- "요약 뷰" 탭을 클릭하여 트랜잭션에 관련된 구성 요소를 표시합니다.
- "추적 뷰" 탭을 클릭하여 트랜잭션에서 생성된 호출과 트랜잭션에 관련된 각 호출에 소요된 시간을 표시합니다.
- "시퀀스 뷰"를 클릭하여 트랜잭션 추적에서 가장 많은 시간이 소요된 트랜잭션 및 스레드에 대한 프로세스 작업 흐름 또는 호출 순서를 표시합니다.

참고: 트랜잭션 추적에 대한 자세한 내용은 [SOA 환경에서 트랜잭션 추적 사용](#) (페이지 101)을 참조하십시오. 추적 구성에 대한 자세한 내용은 [CA APM Java Agent 구현 안내서](#) 또는 [CA APM .NET 에이전트 구현 안내서](#)를 참조하십시오. 트랜잭션 추적 뷰와 기록 데이터 작업에 대한 자세한 내용은 [CA APM Workstation 사용자 안내서](#)를 참조하십시오.

BusinessWorks 의 프런트엔드 및 백엔드 정보

SOA 의 트랜잭션에는 여러 시스템에 대한 동기 및 비동기 호출의 복잡한 처리가 필요한 경우가 많습니다. 예를 들어 단일 트랜잭션이 외부 웹 서비스, TIBCO Enterprise Message Service 서버 인스턴스 및 데이터베이스에 대한 호출을 포함할 수 있습니다. TIBCO BusinessWorks 의 경우 모든 백엔드 호출이 서로 다른 스레드를 사용하여 처리될 수 있기 때문에 백엔드 호출이 어떤 트랜잭션에 속하는지 확인하기가 더욱 어렵습니다.

따라서 SOA Extension for TIBCO BusinessWorks 에는 여러 백엔드 호출을 한 트랜잭션의 동일한 시작점에 연결할 수 있게 해 주는 다음과 같은 구성 속성이 포함되어 있습니다.

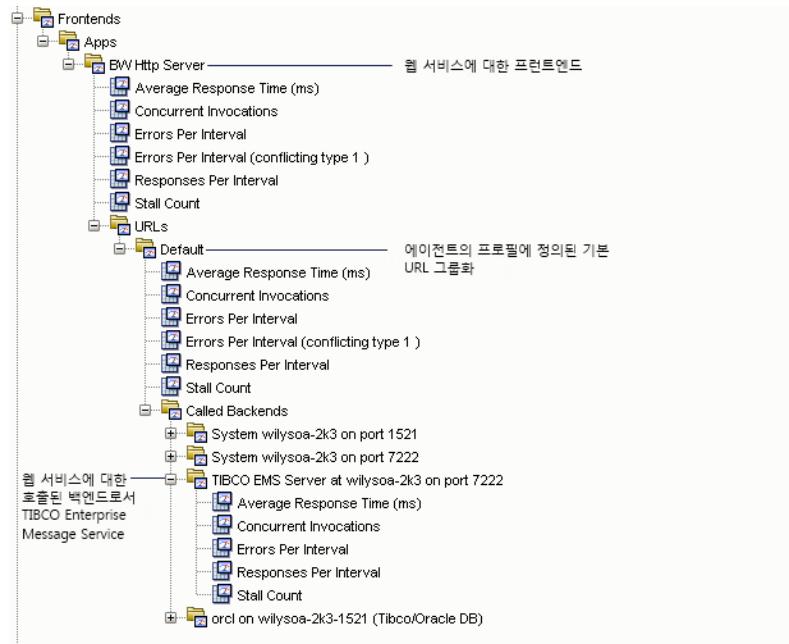
`com.wily.soaextension.tibcobw.mbps.enabled`

기본적으로 이 속성은 `true` 로 설정되어 동일한 트랜잭션에 참여하는 백엔드 호출의 스레드를 "백엔드" 또는 "호출된 백엔드" 노드 아래에 포함하고 적절한 프런트엔드와 연결합니다.

프런트엔드는 트랜잭션의 시작점을 나타냅니다. TIBCO BusinessWorks 에서 기본 프런트엔드는 비즈니스 프로세스의 경우 프로세스 스타터이고 웹 서비스의 경우 TIBCO BusinessWorks HTTP 서블릿입니다.

프런트엔드 메트릭은 "프런트엔드" 노드 아래에 TIBCO BusinessWorks 응용 프로그램 이름별로 수집되고 게시됩니다. 한 트랜잭션의 동일한 프런트엔드에 의해 시작된 모든 백엔드 호출은 해당 프런트엔드의 "호출된 백엔드" 노드 아래에 수집됩니다. 예를 들어 프로세스 스타터가 웹 서비스, 데이터베이스 및 TIBCO Enterprise Message Service 에 대한 호출을 실행한 경우 해당 프로세스 스타터와 연결된 "프런트엔드" 노드 아래에서 호출의 메트릭을 볼 수 있습니다. 비즈니스 프로세스의 백엔드에서 생성된 모든 오류는 연결된 프런트엔드 노드로 전파됩니다.

예를 들어 TIBCO Enterprise Message Service 서버 인스턴스를 호출한 웹 서비스 트랜잭션에 대한 메트릭을 보려면 Investigator에서 BusinessWorks HTTP 서버 프런트엔드를 확장한 다음 "호출된 백엔드" 노드를 확장합니다.



생성된 프로세스에 대한 제한 사항 이해

생성된 하위 프로세스를 생성하는 비즈니스 프로세스가 있는 경우 대부분 프로세스는 생성된 프로세스의 실행이 완료될 때까지 대기하지 않습니다. 하위 프로세스를 생성한 비즈니스 프로세스가 생성된 프로세스가 완료될 때까지 대기하지 않으므로, 생성된 프로세스에서 발생하는 모든 메트릭이나 오류가 호출된 백엔드에 대한 메트릭이나 비즈니스 프로세스의 프런트엔드에 대한 메트릭에 포함되지 않을 수 있습니다.

생성된 프로세스가 실행되어 기본 비즈니스 프로세스가 실행되는 동안 실행이 완료되면 생성된 프로세스의 메트릭과 오류가 비즈니스 프로세스의 프런트엔드에 포함될 수 있습니다.

한 프런트엔드에 여러 백엔드를 연결하지 않도록 설정

기본적으로 `com.wily.soaextension.tibcobw.mbbs.enabled` 속성은 `true`로 설정되어 서로 다른 스레드에서 실행되는 여러 백엔드 시스템에 대한 호출을 트랜잭션을 시작한 프런트엔드에 연결할 수 있도록 합니다. 이 기능이 사용되지 않도록 설정하려면 `com.wily.soaextension.tibcobw.mbbs.enabled` 속성을 `false`로 설정하십시오.

이 속성을 `false`로 설정하면 프런트엔드와 동일한 스레드를 사용하여 실행되는 백엔드 호출만 "호출된 백엔드" 노드 아래에 수집됩니다. 일반적으로 TIBCO BusinessWorks 의 처리에는 다중 스레드가 사용되므로 성능 메트릭과 오류를 해당 프런트엔드와 연관시킬 필요가 없을 때만 이 속성을 `false`로 설정해야 합니다.

사용자 지정 TIBCO BusinessWorks 백엔드 추가

사용자 지정 백엔드를 추가하고 사용자 지정 백엔드를 관련 프런트엔드와 연결하려면 사용자 지정 백엔드 추적 프로그램을 정의하여 사용자 지정 백엔드를 표시한 다음 에이전트에 대한 사용자 지정 `callbackbackend` 추적 프로그램 및 `callbackbackend` 오류 추적 프로그램을 정의해야 합니다. 예를 들어 다음과 같은 백엔드 추적 프로그램을 정의하는 경우

```
SetTracerClassMapping: <CustomBackendTracer>
com.wily.introscope.agent.trace.BackendTracer
com.wily.introscope.probebuilder.validate.ResourceNameValidator
SetTracerParameter: <CustomBackendTracer> nameformatter <CustomNameFormatter>
```

```
TraceOneMethodWithParametersIfFlagged: <CustomTracInFlag> <method>
<CustomBackendTracer> "Metric Path"
```

그런 다음 `CalledBackendTracer` 및 `CalledBackendErrorTracer` 를 다음과 같이 정의할 수 있습니다.

```
SetTracerClassMapping: <CustomCalledBackendTracer>
com.wily.soaextension.tibcobw.multithread.blame.CalledBackendTracer
com.wily.introscope.probebuilder.validate.ResourceNameValidator
SetTracerParameter: <CustomCalledBackendTracer> nameformatter
<CustomNameFormatter>

SetTracerClassMapping: <CustomCalledBackendErrorTracer>
com.wily.soaextension.tibcobw.multithread.blame.CalledBackendErrorTracer
com.wily.introscope.probebuilder.validate.MetricNameValidator
SetTracerParameter: <CustomCalledBackendErrorTracer> nameformatter
<CustomNameFormatter>

TraceOneMethodWithParametersIfFlagged: <CustomTracinFlag> <method>
CustomCalledBackendTracer "Metric Path"
TraceOneMethodWithParametersIfFlagged: <CustomTracinFlag> <method>
CustomCalledBackendErrorTracer "Metric Path:Errors Per Interval"
```

참고: 사용자 지정 추적 프로그램 구성, 이름 형식 지정 정의 및 사용자 지정 프런트엔드와 백엔드 표시에 대한 자세한 내용은 *CA APM Java Agent* 구현 안내서 또는 *CA APM .NET* 에이전트 구현 안내서를 참조하십시오.

메트릭 만료 처리 및 제거 사용자 지정

메트릭 만료 처리는 사용되지 않는 메트릭을 주기적으로 에이전트의 메모리 캐시에서 제거합니다. 사용되지 않는 메트릭은 특정 시간 동안 새 데이터를 보고하지 않은 메트릭입니다. 이렇게 하면 에이전트 성능이 향상되고 시스템에서 처리 가능한 것보다 많은 메트릭이 수집되지 않도록 방지할 수 있습니다.

하지만 장시간 실행되는 트랜잭션이나 비즈니스 프로세스가 있으면 일부 메트릭에 메트릭 만료 처리가 적용되지 않도록 할 수 있습니다. 예를 들어 12 시간 이상 실행되는 비즈니스 프로세스가 있는 경우 다음과 유사한 구성 속성을 사용하여 비즈니스 프로세스의 "동시 호출"에 메트릭 만료 처리가 적용되지 않도록 할 수 있습니다.

```
introscope.agent.metricAging.metricExclude.ignore.2=Tibco\|Processes\|*.*:Concurrent Invocations
```

또한 다른 에이전트 속성을 사용하여 메트릭 만료 처리를 사용자 지정할 수도 있습니다. 예를 들어 속성을 설정하여 메트릭이 제거될 위한 후보가 될 때까지 경과할 시간 또는 각 간격에서 확인할 메트릭의 수를 사용자 지정할 수 있습니다.

참고: 메트릭 만료 처리 및 에이전트 속성 구성에 대한 자세한 내용은 *CA APM Java Agent* 구현 안내서 또는 *CA APM .NET* 에이전트 구현 안내서를 참조하십시오.

TIBCO BusinessWorks 메트릭을 메트릭 만료 처리에서 제외하려면

1. *<Agent_Home>/core/config* 디렉터리로 이동합니다.
2. *IntroscopeAgent.profile* 파일을 텍스트 편집기에서 엽니다.
3. *Agent Metric Aging* 섹션을 찾아서 다음 속성의 주석 처리를 제거합니다.
`introscope.agent.metricAging.metricExclude.ignore.2=Tibco\|Processes\|*.*:Concurrent Invocations`

12 시간 이상 실행되는 비즈니스 프로세스가 있는 경우에는 "동시 호출"을 메트릭 만료 처리에서 제외하는 것이 좋습니다.

TIBCO BusinessWorks 를 위한 상호 관련된 추적 구성

크로스 프로세스 트랜잭션을 추적하려면 에이전트가 한 프로세스에서 다른 프로세스로 전달될 수 있는 상관 관계 식별자를 삽입해야 합니다.

에이전트는 이 상관 관계 식별자를 SOAP 또는 HTTP 헤더에 삽입할 수 있습니다. 기본적으로 SOA Extension for TIBCO BusinessWorks 는 상관 관계 정보의 전달을 SOAP 및 HTTP 헤더 모두에서 지원하도록 구성됩니다. 상관 관계 정보를 사용하여 크로스 프로세스 트랜잭션을 추적하는 방법에 대한 자세한 정보는 [트랜잭션의 세그먼트가 연결되는 방식 이해](#) (페이지 102)를 참조하십시오. 상호 관련된 추적의 속성 구성에 대한 자세한 내용은 [상호 관련된 추적 구성](#) (페이지 121)을 참조하십시오.

제 9 장: TIBCO Enterprise Message Service 모니터링

TIBCO Enterprise Message Service 는 엔터프라이즈의 시스템 간에 동기 및 비동기 통신을 가능하게 하는 SOA 인프라의 핵심 구성 요소입니다. SOA Extension for TIBCO EMS(Enterprise Message Service)를 사용하면 큐, 항목과 브리지, 멀티캐스트 채널, 라우트 같은 고급 구성 요소를 비롯한 TIBCO EMS 서버의 많은 핵심 요소를 모니터링할 수 있습니다.

이 단원에서는 TIBCO EMS 환경의 상태와 작동을 모니터링하고 분석하는 데 사용할 수 있는 대시보드와 메트릭에 대해 설명합니다.

이 섹션은 다음 항목을 포함하고 있습니다.

- [TIBCO Enterprise Message Service 정보](#) (페이지 198)
- [SOA Extension for TIBCO EMS 설치 방법](#) (페이지 198)
- [Standalone Agent 설치 관리자 실행](#) (페이지 199)
- [TIBCO EMS 서버 모니터링 준비](#) (페이지 201)
- [TIBCO EMS Monitor 에이전트 구성](#) (페이지 203)
- [선택적 모니터링을 위한 필터 구성](#) (페이지 206)
- [모니터링 수준 정의](#) (페이지 209)
- [암호화된 암호 만들기](#) (페이지 214)
- [SSL을 사용하여 TIBCO EMS 서버 인스턴스에 연결](#) (페이지 215)
- [EMSMonitor 에이전트 시작](#) (페이지 217)
- [Enterprise Manager 확장을 사용하도록 설정](#) (페이지 219)
- [대시보드를 사용하여 TIBCO EMS 모니터링](#) (페이지 220)
- [TIBCO EMS 메트릭 이해 및 보기](#) (페이지 222)
- [기본 TIBCO EMS 메트릭 그룹화 보기](#) (페이지 253)
- [기본 TIBCO EMS 경고 보기](#) (페이지 254)
- [에이전트 구성 속성 요약](#) (페이지 255)

TIBCO Enterprise Message Service 정보

TIBCO Enterprise Message Service 는 광범위한 플랫폼과 응용 프로그램 기술을 지원하는 JMS-호환 통신 기능을 제공함으로써 SOA(서비스 지향 아키텍처)의 기반 역할을 하는 표준 기반 메시징 계층입니다.

메시지의 정확하고 안전한 배달은 비즈니스 서비스의 제공과 비즈니스 트랜잭션의 완료에 매우 중요하므로 SOA Extension for TIBCO EMS 는 TIBCO Enterprise Message Service 의 핵심 구성 요소와 성능 메트릭을 모니터링하기 위한 별도의 에이전트를 제공합니다.

TIBCO Enterprise Message Service 를 모니터링하는 데 사용되는 에이전트는 핵심 Java 또는 .NET 에이전트의 확장이 아닌 독립 실행형 에이전트입니다. *EMSMonitor* 에이전트는 Java 클래스를 이용하는 대신 TIBCO EMS Admin API 를 사용하여 TIBCO EMS 구성 요소에 대한 성능 정보를 수집한 다음 이 정보를 Enterprise Manager 에 보고합니다.

EMSMonitor 에이전트는 독립 실행형 에이전트이므로 별도의 소프트웨어 패키지로 배포되며 자체 *IntroscopeAgent.profile* 과 고유한 구성 단계가 필요합니다. 예를 들면 모니터링 할 EMS 서버에 대한 연결 정보를 구성하고 필터를 추가하여 메트릭을 수집할 구성 요소를 사용자 지정해야 합니다.

적절한 에이전트 속성을 구성한 후에는 *EMSMonitor* 에이전트를 사용하여 로컬 및 원격 TIBCO EMS 구성 요소를 모니터링 할 수 있습니다.

SOA Extension for TIBCO EMS 설치 방법

EMSMonitor 에이전트는 핵심 에이전트에 종속되지 않습니다. *EMSMonitor* 에이전트는 TIBCO Enterprise Message Service 를 실행하는 하나 이상의 컴퓨터에서 다른 모든 CA Introscope 구성 요소에 독립적으로 설치하고 구성할 수 있습니다.

SOA extension for TIBCO EMS 를 추가할 때는 다음과 같은 간략한 단계를 수행합니다.

1. TIBCO EMS(Enterprise Message Service)의 지원되는 버전이 설치되어 있는지 확인합니다.
참고: 시스템 요구 사항에 대해서는 *Compatibility Guide*(호환성 안내서)를 참조하십시오.
2. 환경에 Enterprise Manager 와 Workstation 이 설치되어 있는지 확인합니다. EMSMonitor 에이전트가 데이터를 전송할 Enterprise Manager 에 대한 연결 정보가 있는지 확인합니다.
3. [Standalone Agent 설치 관리자를 실행](#) (페이지 199)하거나 응답 파일을 사용하여 필요한 에이전트 파일을 환경에 추가합니다.
4. [EMS 서버의 모니터링을 준비합니다.](#) (페이지 201)
5. [TIBCO EMSMonitor 에이전트 프로필을 구성](#) (페이지 203)하여 연결 및 모니터링 속성을 정의합니다.
6. [Enterprise Manager extension for TIBCO BusinessWorks 를 사용하도록 설정합니다.](#) (페이지 219)

Standalone Agent 설치 관리자 실행

Standalone Agent 설치 관리자를 사용하면 핵심 Java 또는 .NET 에이전트를 사용하지 않는 Standalone Agent 를 설치할 수 있습니다. TIBCO EMS(Enterprise Message Service)의 모니터링을 사용하도록 설정하려면 Standalone Agent 설치 관리자를 사용하여 에이전트 관련 파일을 환경에 추가하고 Enterprise Manager 에 대한 에이전트 연결을 구성할 수 있습니다. Standalone Agent 설치 관리자는 필요한 파일을 추출하여 적절한 위치에 넣고 사용자가 이를 필요에 따라 수정하여 에이전트 구성을 완료할 수 있도록 합니다.

TIBCO EMS 의 모니터링을 위해 Standalone Agent 설치 관리자를 실행하려면

1. 운영 환경에 적절한 Standalone Agent 설치 관리자를 시작합니다.
2. "소개" 페이지에서 "다음"을 클릭합니다.
3. 설치 할 모니터링 패키지를 선택하고 "다음"을 클릭합니다. 예를 들어 TIBCO EMS 에 대한 모니터링을 사용하려면 "SOA Extension For TIBCO Enterprise Message Service" 옵션을 선택합니다.

4. 설치 디렉터리에 대해서는 "다음"을 클릭하여 기본 위치를 사용하거나 "찾아보기"를 클릭하여 다른 위치를 지정합니다.
5. Enterprise Manager 연결 설정에 대해서는 에이전트가 데이터를 전송할 Enterprise Manager 호스트 이름 및 포트 번호를 지정하고 "다음"을 클릭합니다.
6. 설정의 요약을 검토한 다음 "설치"를 클릭하여 설치를 시작합니다.
7. 설치가 완료되면 "완료"를 클릭하여 Standalone Agent 설치 관리자를 닫습니다.

자동 설치를 위해 응답 파일 사용

Standalone Agent 설치 관리자를 대화식으로 실행하지 않으려면 독립 실행형 샘플 응답 파일을 편집하여 에이전트 파일을 설치하십시오. 이 방법을 사용하여 SOA extension for TIBCO EMS 를 자동 모드에서 사용하도록 설정할 수 있습니다.

다음 단계를 수행하십시오.

1. Standalone Agent 설치 관리자와 같은 디렉터리에 있는 SampleResponseFile.StandaloneAgentPP.txt 파일을 엽니다.
2. SampleResponseFile.StandaloneAgentPP.txt 파일을 편집하여 shouldInstallTibcoEMS 속성을 true 로 설정하고 SOA extension for TIBCO Enterprise Message Service 를 에이전트에 추가합니다. 예: shouldInstallTibcoEMS=true
3. SampleResponseFile.StandaloneAgentPP.txt 파일을 저장합니다.
4. 명령줄에 적절한 명령을 입력하여 설치 관리자를 호출합니다.

참고: 자동 모드에서 에이전트를 설치하는 방법에 대한 자세한 내용은 **CA APM Java Agent** 구현 안내서 또는 **CA APM .NET** 에이전트 구현 안내서를 참조하십시오.

설치 아카이브 수동 추출

Standalone Agent 설치 관리자 또는 독립 실행형 응답 파일에 대한 액세스 권한이 없는 경우 운영 환경에 적합한 독립 실행형 설치 아카이브를 다운로드할 수 있습니다. [CA Support](#)의 CA APM 소프트웨어 다운로드 영역에서 CA APM 제품을 다운로드할 수 있습니다. 다운로드한 후에는 운영 환경에 적절한 명령을 사용하여 아카이브에서 수동으로 파일을 추출할 수 있습니다. 예를 들어 UNIX 컴퓨터에서는 다음과 같이 *tar* 명령을 사용합니다.

```
tar -xvf IntroscopeStandaloneAgentPPIInstaller<version>.tar  
r
```

TIBCO EMS 서버 모니터링 준비

TIBCO Enterprise Message Service 를 모니터링하려면 EMSMonitor 에이전트가 사용할 사용자 계정을 준비하고 모니터링에 필요한 라이브러리를 로컬에서 사용할 수 있는지 확인해야 합니다.

다음 단계를 수행하십시오.

1. *change-admin-acl* 및 *change-user* 권한이 있는 계정을 사용하여 EMS 서버에 로그인합니다. 예를 들어 *admin* 계정을 사용하여 로그인합니다.
2. EMSMonitor 에이전트에 대한 *view-all* 권한이 있는 사용자 계정을 찾거나 만듭니다. 예를 들어 EMSMonitor 에이전트를 실행하는 새 사용자를 만들려면 EMS 관리 도구에서 다음 명령을 실행합니다.
create user wilyemsuser
grant admin wilyemsuser view-all
3. <*TIBCO_EMS_HOME*>/lib 디렉터리에서 다음과 같은 TIBCO EMS 라이브러리를 사용할 수 있는지 확인합니다.
jms.jar
tibjms.jar
tibjmsadmin.jar

원격 서버 인스턴스를 모니터링 하려면 EMSMonitor 에이전트를 실행하는 컴퓨터가 EMS 클라이언트로 구성되어 있어야 합니다. 일반적으로 이는 jms.jar, tibjms.jar 및 tibjmsadmin.jar 파일이 사용 가능하며 EMSMonitor 에이전트와 동일한 컴퓨터의 클래스 경로에 추가되었다는 의미입니다.

참고: EMS 클라이언트 응용 프로그램을 설정하고 이러한 파일을 클래스 경로에 추가하는 방법에 대한 자세한 내용은 *TIBCO Enterprise Message Service User's Guide*(TIBCO Enterprise Message Service 사용자 안내서)를 참조하십시오.

EMS 버전 5.x 이상에 해당: 에이전트가 SSL(Secure Socket Layer) 연결을 사용하여 EMS 서버에 연결해야 하는 경우에는 버전 5.x 이상의 EMS를 실행하는 로컬 컴퓨터에서 다음과 같은 추가 라이브러리를 사용할 수 있어야 합니다.

tibcrypt.jar
slf4j-api-1.4.2.jar
slf4j-simple-1.4.2.jar

EMS 서버 버전 4.4.x에 해당: *tibcrypt.jar* 라이브러리만 필요합니다.

참고: EMS에 사용할 SSL 통신의 구성에 대한 자세한 내용은 *TIBCO Enterprise Message Service User's Guide*(TIBCO Enterprise Message Service 사용자 안내서)를 참조하십시오.

EMS 서버 버전 4.4.x 및 5.x에 해당: EMS 4.4.x 및 5.x를 포함하는 환경을 모니터링 하려면 로컬 컴퓨터에 있는 모든 필수 .jar 파일이 사용 가능한 최신 .jar 파일인지 확인해야 합니다. 예를 들어 환경에 EMS 4.4.x 및 5.x가 포함되는 경우 EMSMonitor 에이전트가 5.x .jar 파일을 사용하는지 확인하십시오.

TIBCO EMSMonitor 에이전트 구성

Standalone Agent 설치 관리자를 실행한 후에는 *TibcoEMSServer* 디렉터리를 확인해야 합니다. *TibcoEMSServer* 디렉터리는 TIBCO Enterprise Message Service 의 모니터링을 위하여 다음과 같은 파일 및 디렉터리를 제공합니다.

ext 디렉터리

Supportability-Agent.jar 파일이 있습니다. *Supportability-Agent.jar* 파일은 EMSMonitor 에이전트 지원이 필요한 경우 CA Support 에 제출할 수 있는 로그 파일에 시스템 속성, 구성 파일 및 기타 정보를 기록하는 확장입니다.

lib 디렉터리

.jar 파일로 패키지된 필수 EMSMonitor 에이전트 라이브러리(*Agent.jar* and *jline-0.9.9.jar*)가 있습니다.

properties 디렉터리

IntroscopeAgent.profile, *TibcoEMSServer.properties* 및 *MonitoringLevel.xml* 파일이 있습니다.

- *IntroscopeAgent.profile* 을 사용하여 Enterprise Manager 연결을 구성할 수 있습니다.
- *TibcoEMSServer.properties* 파일을 사용하여 로컬 및 원격 TIBCO EMS 서버 인스턴스에 연결하기 위한 매개 변수를 구성하고 메트릭을 수집할 구성 요소와 모니터링 수준을 사용자 지정할 수 있습니다.
- *MonitoringLevel.xml* 파일을 사용하여 최소, 권장 및 전체 모니터링 수준 집합을 구성하는 메트릭을 사용자 지정합니다.

EMSServer.jar 파일

에이전트가 EMS 서버 인스턴스를 모니터링하기 위해 사용하는 클래스를 제공합니다.

emsPwdEncryptor 파일

EMS 서버에 연결하기 위한 SSL 클라이언트 인증서 암호 및 EMS 사용자 암호를 암호화할 수 있는 암호 암호화 스크립트(*emsPwdEncryptor.bat* 또는 *emsPwdEncryptor.sh*)를 제공합니다.

EMSMonitor 파일

EMS 서버 인스턴스를 모니터링 할 준비가 되었을 때 EMSMonitor 에이전트를 실행하기 위한 시작 스크립트(EMSMonitor.bat 또는 EMSMonitor.sh)를 제공합니다.

UNIX 플랫폼에서는 EMSMonitor.sh 스크립트를 사용하여 EMSMonitor 에이전트를 시작, 중지, 다시 시작하거나 상태를 확인할 수 있습니다. Windows에서는 EMSMonitor.bat 또는 Windows 서비스를 사용하여 에이전트를 시작하고 EMSMonitor 에이전트를 중지할 수 있습니다.

TibcoEMSMonitor 디렉터리의 파일을 사용하면 EMSMonitor 에이전트 연결 속성, 모니터링 속성, 보안 통신 속성을 구성할 수 있습니다.

기본 연결 속성 구성

EMSMonitor 에이전트는 TIBCO EMS 서버의 데이터를 Enterprise Manager로 보고합니다. 메트릭의 수집 및 보고가 사용되도록 설정하려면 EMSMonitor 에이전트가 보고하는 Enterprise Manager에 대한 연결 정보를 지정하고 에이전트가 데이터를 수집하는 서버의 목록을 지정해야 합니다.

EMSMonitor 에이전트 연결 속성을 구성하려면

1. *IntroscopeAgent.profile* 파일을 텍스트 편집기에서 열고 Enterprise Manager 연결 매개 변수를 지정합니다. 예:

```
introscope.agent.enterprisemanager.transport.tcp.host.DEFAULT=mercury  
introscope.agent.enterprisemanager.transport.tcp.port.DEFAULT=5001
```

2. *TibcoEMSMonitor.properties* 파일을 텍스트 편집기에서 열고 *ems.server.list* 속성을 모니터링하려는 EMS 서버 인스턴스 이름으로 설정합니다. 예:

```
ems.server.list = tibco_ems_srv01, tibco_ems_srv02
```

이 목록에는 원하는 이름을 사용할 수 있습니다. 모니터링하려는 서버 인스턴스의 실제 이름과 일치시킬 필요는 없습니다. 하지만 목록의 이름은 추가 속성을 정의하는 데 사용하는 서버 인스턴스 이름과 일치해야 합니다. 예를 들어 이름이 *NewYork*인 서버 인스턴스 두 개가 있고 *ems.server.list* 속성에서 별칭 *newyork1* 및 *newyork2*를 사용하는 경우 이 두 서버의 나머지 속성은 이 별칭을 사용하여 정의합니다. 예:

```
ems.server.list = newyork1, newyork2  
newyork1.host = catsmcn01  
newyork1.port = 6001
```

이제 **Investigator** 트리에 표시되는 서버 인스턴스 이름에 각 인스턴스의 URL이 추가되므로 두 인스턴스를 모니터링하고 고유하게 식별할 수 있습니다.

3. *ems.server.list* 속성에 지정한 각 서버 인스턴스에 연결할 호스트 이름 및 포트를 구성합니다. 예:

```
tibco_ems_srv01.host=localhost
tibco_ems_srv01.port=7222
```

```
tibco_ems_srv02.host=vespa
tibco_ems_srv02.port=7222
```

4. EMS 서버에 연결하는 데 사용할 사용자 계정의 암호화된 암호를 생성하려면 *emsPwdEncryptor* 유ти리티를 실행하십시오.

EMS 서버에 연결하는 데 사용할 사용자 계정을 만들거나 선택하는 데 대한 자세한 내용은 [모니터링을 위해 EMS 서버 준비](#) (페이지 201)를 참조하십시오. *emsPwdEncryptor* 유ти리티 사용에 대한 자세한 내용은 [암호화된 암호 만들기](#) (페이지 214)를 참조하십시오.

서버의 폴링 간격 구성

EMSMonitor 에이전트는 각 서버 인스턴스를 정기적인 간격으로 폴링하여 최신 상태 및 구성 정보를 검색합니다. 구성 속성을 사용하면 이러한 쿼리의 빈도를 제어할 수 있습니다.

EMSMonitor 에이전트가 서버를 쿼리하는 빈도를 구성하려면

1. 상태 관련 메트릭을 수집하기 위해 각 서버 인스턴스를 폴링하는 간격을 구성합니다.

지정한 간격은 EMS 서버 인스턴스에서 상태 정보를 쿼리하는 빈도를 제어합니다. 예를 들어 *tibco_ems_srv01* 인스턴스를 60 초마다 한 번씩 쿼리하고 *tibco_ems_srv02* 인스턴스를 120 초마다 한 번씩 쿼리하려면 다음과 같은 속성을 설정합니다.

```
tibco_ems_srv01.delaytime=60
tibco_ems_srv02.delaytime=120
```

간격을 지정하지 않으면 기본적으로 60 초마다 한 번씩 메트릭을 새로고침합니다.

2. 각 서버 인스턴스에 대해 정적 구성 관련 메트릭을 새로 고치는 간격을 구성합니다.

구성 속성은 자주 변경되지 않으므로 **EMSMonitor** 에이전트는 시작 시에 한 번 구성 속성 값을 수집한 후 **<ServerInstance>.report.static.freq** 속성과 **<ServerInstance>.delaytime** 속성을 곱하여 설정되는 값만큼의 시간이 지나면 구성 메트릭을 폴링합니다. **<ServerInstance>.delaytime** 속성이 60 초로 설정되어 있고 **<ServerInstance>.report.static.freq** 속성이 20 으로 설정되어 있으면 정적 메트릭을 1200 초(20 분)마다 한 번씩 확인합니다.

예:

```
tibco_ems_srv01.report.static.freq=20  
tibco_ems_srv02.report.static.freq=20
```

이 속성을 0 으로 설정하면 **EMSMonitor** 에이전트가 시작 시에 모든 구성 메트릭을 보고하지만 그 이후의 구성 메트릭 변경 사항은 보고하지 않습니다.

선택적 모니터링을 위한 필터 구성

기본적으로 **EMSMonitor** 에이전트는 **ems.server.list** 속성에 지정된 서버 인스턴스의 모든 큐 및 항목에 대한 메트릭을 수집합니다. 큐 및 항목을 선택적으로 모니터링하려면 필터를 구성하여 관심 큐 및 항목을 지정할 수 있습니다. 또한 필터를 구성하여 브리지, 멀티캐스트 채널, 라우트 등과 같은 고급 구성 요소의 모니터링을 설정할 수 있으며 정규식을 사용하여 이러한 고급 구성 요소 중에서 메트릭을 수집하고자 하는 하위 집합을 지정할 수 있습니다.

큐 및 항목에 대한 필터 구성

EMS 서버는 정적 및 동적 큐/항목을 모두 지원합니다. 정적 큐 및 항목은 EMS 관리자에 의해 생성되고 관리되며 사용되지 않는 경우에도 서버에 유지됩니다. 임시 동적 큐를 비롯한 동적 큐 및 항목은 필요 시 EMS 클라이언트에 의해 생성되고 삭제되며 서버에는 짧은 기간 동안만 유지됩니다.

기본적으로 **EMSMonitor** 에이전트는 불필요한 오버헤드를 방지하기 위해 정적 큐 및 항목만 모니터링합니다. 하지만 동적 큐 또는 동적 항목을 포함하는 필터를 정의하면 동적 큐 및 항목을 모니터링할 수 있습니다. 또한 정규식을 사용하는 필터를 구성하여 메트릭을 수집할 특정 큐 및 항목을 지정할 수도 있습니다.

동적 큐 또는 항목의 모니터링이 사용되도록 설정하려면

1. *TibcoEMSMonitor.properties* 파일을 텍스트 편집기에서 엽니다.
2. 모니터링하려는 EMS 서버 인스턴스의 이름을 확인합니다.
3. 동적 개체를 모니터링하려는 각 서버 인스턴스의 동적 큐 또는 동적 항목에 대한 포함 필터를 구성합니다. 예를 들어 EMS 서버 인스턴스 *tibco_ems_srv01*에서 동적 큐 및 동적 항목의 모니터링이 사용되도록 설정하려면 다음과 같이 합니다.

```
tibco_ems_srv01.queue.filter.include.dynamic=true
tibco_ems_srv01.topic.filter.include.dynamic=true
```

정규식을 사용하여 특정 큐 또는 항목을 모니터링하려면

1. *TibcoEMSMonitor.properties* 파일을 텍스트 편집기에서 엽니다.
2. 모니터링하려는 EMS 서버 인스턴스의 이름을 확인합니다.
3. 정규식을 사용하여 각 서버 인스턴스에 대한 큐 및 항목 필터를 원하는 대로 구성합니다. 예:

```
tibco_ems_srv01.queue.filter.includeonly.regex=sample.*
tibco_ems_srv01.topic.filter.includeonly.regex=sample.*
```

```
tibco_ems_srv02.queue.filter.includeonly.regex=test.*
tibco_ems_srv02.topic.filter.includeonly.regex=test.*
```

동적 큐 및 항목을 포함하도록 속성을 구성한 경우 필터를 지정하지 않으면 동적 및 임시 큐/항목을 비롯한 모든 큐 및 항목에 대한 메트릭이 수집됩니다. 포함할 큐에 대한 정규식을 지정한 경우에는 정규식과 일치하는 큐만 모니터링됩니다. 포함할 항목에 대한 정규식을 지정한 경우에는 정규식과 일치하는 항목만 모니터링됩니다.

포함할 고급 구성 요소 정의

기본적으로 EMSMonitor 에이전트는 브리지, 채널 또는 라우트에 대한 메트릭을 보고하지 않습니다. 이러한 고급 구성 요소를 모니터링하도록 설정하려면 *TibcoEMSMonitor.properties* 파일에 포함 필터를 지정해야 합니다. 포함하고자 하는 고급 구성 요소에 대한 모니터링이 사용되도록 설정한 후에는 정규식 필터를 사용하여 메트릭을 수집하고자 하는 고급 구성 요소의 목록을 상세하게 정의할 수 있습니다.

브리지, 채널 및 라우트를 모니터링하도록 설정하려면

1. *TibcoEMSMonitor.properties* 파일을 텍스트 편집기에서 엽니다.
2. 모니터링하려는 EMS 서버 인스턴스의 이름을 확인합니다.
3. 모니터링하려는 각 서버 인스턴스에서 각 고급 구성 요소에 대한 포함 필터를 구성합니다. 예를 들어 서버 인스턴스 *tibco_ems_srv1*에서 브리지, 채널 및 라우트를 모니터링하려면 파일에 다음 포함 필터를 추가합니다.

```
tibco_ems_srv1.monitor.bridges=true  
tibco_ems_srv1.monitor.channels=true  
tibco_ems_srv1.monitor.routes=true
```

고급 구성 요소에 대한 정규식 필터 정의

브리지, 채널 또는 라우트를 모니터링하도록 설정하는 포함 필터를 정의한 경우 메트릭을 수집할 특정 브리지, 채널 또는 라우트를 지정하는 정규식 필터도 구성할 수 있습니다.

구성 요소에 대한 포함 필터를 정의하고 정규식 필터를 정의하지 않으면 해당 유형의 모든 구성 요소가 모니터링됩니다. 예를 들어 브리지를 모니터링하도록 설정하고 브리지에 대한 정규식 필터를 추가하지 않으면 모든 브리지에 대한 메트릭이 수집됩니다. 구성 요소에 대한 포함 필터를 정의하지 않은 경우에는 해당 구성 요소에 대한 모든 정규식 필터가 무시됩니다.

브리지, 채널 또는 라우트에 대한 정규식 필터를 정의하려면

1. *TibcoEMSMonitor.properties* 파일을 텍스트 편집기에서 엽니다.
2. 모니터링하려는 EMS 서버 인스턴스의 이름을 확인합니다.
3. 모니터링하려는 각 서버 인스턴스에서 각 고급 구성 요소에 대한 정규식 필터를 구성합니다. 예를 들어 서버 인스턴스 *tibco_ems_srv1*에서 브리지, 채널 및 라우트를 모니터링하도록 설정한 경우 파일에 다음 포함 필터를 추가합니다.

```
<ServerInstance>.bridge.filter.includeonly.regex  
<ServerInstance>.channel.filter.includeonly.regex  
<ServerInstance>.route.filter.includeonly.regex
```

각 필터에 정규식을 사용하여 모니터링하려는 구성 요소의 하위 집합을 나타냅니다. 예:

```
tibco_ems_srv1.bridge.filter.includeonly.regex=new.*  
tibco_ems_srv1.channel.filter.includeonly.regex=.*bulletin  
tibco_ems_srv1.route.filter.includeonly.regex=.*
```

정규식 필터는 *true*로 설정된 포함 필터를 구성한 고급 구성 요소에만 적용됩니다.

EMS 이름의 특수 문자 바꾸기

TIBCO Enterprise Message Service에서는 모든 구성 요소 이름에 콜론(:)이나 파이프(|) 문자와 같은 특수 문자를 포함할 수 있습니다. 하지만 콜론(:) 및 파이프(|) 문자는 CA Introscope에서 209 예약된 문자이며 구성 요소 이름에서는 자동으로 밑줄 문자(_)로 바뀝니다. 예를 들어 이름이 *Queue:WorkInProgress*인 큐가 있는 경우 EMS 에이전트는 콜론 문자를 밑줄로 바꾸고 이 큐에 대한 메트릭을 *Queue_WorkInProgress*라는 이름의 노드 아래에서 보고합니다.

예를 들어 *Queue:WorkInProgress*와 *Queue/WorkInProgress*처럼 비슷한 이름의 큐가 있는 경우 특수 문자가 바뀌면 이름이 중복됩니다. 콜론 및 파이프 문자를 바꾸면 큐 이름이 같아지므로 **Investigator**는 *Queue_WorkInProgress*라는 이름의 큐에 대해 하나의 메트릭 집합만 표시합니다.

특수 문자에 대한 이러한 제한은 모든 EMS 구성 요소에 적용됩니다. 따라서 모니터링을 구성하기 위한 필터를 정의할 때 콜론이나 파이프 문자의 바꾸기가 구성 요소 이름에 미치는 영향을 고려하십시오.

모니터링 수준 정의

EMSMonitor 에이전트는 각 TIBCO EMS 서버 인스턴스에 대해 많은 수의 메트릭을 수집할 수 있습니다. 수집되고 Enterprise Manager에 보고되는 메트릭의 수를 제어하려면 각 TIBCO EMS 서버 인스턴스에 대해 모니터링 수준을 정의합니다. 각 서버 인스턴스에서 서버, 큐 및 항목 메트릭에 별도의 모니터링 수준을 정의할 수 있습니다.

기본 모니터링 수준 정의 사용

각 EMS 서버 및 해당 구성 요소에 대한 메트릭의 수를 제어 할 수 있도록 *EMSMonitor* 에이전트는 서버 메트릭, 큐 메트릭 및 항목 메트릭에 대해 다음과 같은 모니터링 수준을 지원합니다.

Minimum(최소)

서버, 큐 또는 항목을 모니터링 하는 데 필수적인 메트릭만 수집합니다. 미리 구성된 대시보드 및 "개요" 탭에 정보를 표시하려면 기본적으로 **minimum** 모니터링 수준으로 정의된 메트릭이 필요합니다.

EMS 서버 인스턴스에서 서버, 큐 또는 항목 메트릭의 오버헤드를 최소화하려는 경우 이 모니터링 수준을 사용합니다.

Recommended(권장)

minimum 수준의 모든 메트릭과 함께 서버의 내결합성 메트릭, 큐에 대한 인바운드 및 아웃바운드 메시지 통계 또는 항목의 멀티캐스트 관련 메트릭 같은 추가 메트릭을 수집합니다.

서버, 큐 또는 항목의 성능 및 작동을 더 자세히 볼 수 있는 대신 오버헤드가 약간 증가해도 괜찮은 경우 이 모니터링 수준을 사용합니다.

Full(전체)

minimum 및 **recommended** 메트릭을 비롯하여 서버, 큐 또는 항목의 모든 사용 가능한 메트릭을 수집하고 보고합니다.

서버, 큐 또는 항목의 성능 및 작동에 대한 자세한 정보가 필요한 경우 또는 성능 오버헤드가 문제가 되지 않는 경우 이 모니터링 수준을 사용합니다.

모니터링 수준은 각 서버 인스턴스에서 서버, 큐 또는 항목에 대해 수집할 메트릭 집합을 지정하며, 각 서버 인스턴스의 각 구성 요소마다 개별적으로 모니터링 수준을 설정할 수 있습니다.

예를 들어 서버 인스턴스 *tibco_ems_server01*에서 모든 서버 메트릭, **minimum** 큐 메트릭 및 **recommended** 항목 메트릭을 수집하려면 다음 속성을 사용하여 각 구성 요소에 대한 모니터링 수준을 설정할 수 있습니다.

```
tibco_ems_server01.monitoring.level = full  
tibco_ems_server01.queue.monitoring.level = minimum  
tibco_ems_server01.topic.monitoring.level = recommended
```

계속해서 다른 서버 인스턴스에서 서버, 큐 및 항목 모니터링 수준에 대하여 이러한 속성을 구성할 수 있습니다. 예:

```
tibco_ems_server02.monitoring.level = minimum  
tibco_ems_server02.queue.monitoring.level = recommended  
tibco_ems_server02.topic.monitoring.level = recommended
```

TIBCO EMS 서버 인스턴스에 대한 모니터링 수준을 설정하려면

1. *TibcoEMSMonitor.properties* 파일을 텍스트 편집기에서 엽니다.
2. 모니터링하려는 EMS 서버 인스턴스의 이름을 확인합니다.
3. 각 서버 인스턴스에 대해 모니터링 수준 속성
<ServerInstance>.monitoring.level 을 구성합니다. 예를 들어 모니터링 수준을 EMS 서버 인스턴스 *tibco_ems_srv01* 에서는 *minimum* 으로 설정하고 EMS 서버 인스턴스 *tibco_ems_srv02* 에서는 *full* 로 설정하려면 다음과 같이 합니다.

```
tibco_ems_srv01.monitoring.level=minimum  
tibco_ems_srv02.monitoring.level=full
```

TIBCO EMS 큐에 대한 모니터링 수준을 설정하려면

1. *TibcoEMSMonitor.properties* 파일을 텍스트 편집기에서 엽니다.
2. 모니터링하려는 EMS 서버 인스턴스의 이름을 확인합니다.
3. 각 서버 인스턴스에서 큐에 대한 모니터링 수준 속성
<ServerInstance>.queue.monitoring.level 을 설정합니다. 예를 들어 EMS 서버 인스턴스 *tibco_ems_srv01* 의 큐에 대한 모니터링 수준을 *minimum* 으로 설정하고 EMS 서버 인스턴스 *tibco_ems_srv02* 의 큐에 대한 모니터링 수준을 *recommended* 로 설정하려면 다음과 같이 합니다.

```
tibco_ems_srv01.queue.monitoring.level=minimum  
tibco_ems_srv02.queue.monitoring.level=recommended
```

설정한 모니터링 수준은 지정된 서버 인스턴스에 있는 모든 정적 및 동적 큐에 적용됩니다.

TIBCO EMS 항목에 대한 모니터링 수준을 설정하려면

1. *TibcoEMSSMonitor.properties* 파일을 텍스트 편집기에서 엽니다.
2. 모니터링하려는 EMS 서버 인스턴스의 이름을 확인합니다.
3. 각 서버 인스턴스마다 항목에 대한 모니터링 수준 속성 *<ServerInstance>.topic.monitoring.level* 을 설정합니다. 예를 들어 EMS 서버 인스턴스 *tibco_ems_srv01* 의 항목에 대한 모니터링 수준을 *full*로 설정하고 EMS 서버 인스턴스 *tibco_ems_srv02* 의 항목에 대한 모니터링 수준을 *recommended*로 설정하려면 다음과 같이 합니다.
tibco_ems_srv01.topic.monitoring.level=full
tibco_ems_srv02.topic.monitoring.level=recommended

설정한 모니터링 수준은 지정된 서버 인스턴스에 있는 모든 정적 및 동적 항목에 적용됩니다.

기본 모니터링 수준 정의 설정

minimum, *recommended* 또는 *full* 모니터링 수준과 연결된 메트릭을 변경하려면 *MonitoringLevel.xml* 파일을 편집하여 변경할 수 있습니다.

MonitoringLevel.xml 파일에는 EMS 서버 인스턴스, 큐 및 항목에 사용할 수 있는 모든 메트릭이 나와 있습니다. 이 파일은 각 메트릭에 대한 모니터링 수준도 정의합니다. 이 파일에서 모니터링 수준을 변경하면 *TibcoEMSSMonitor.properties* 파일에서 서버, 큐 및 항목에 대한 모니터링 수준을 *minimum*, *recommended* 또는 *full*로 설정하여 수집되는 메트릭을 변경할 수 있습니다.

이 파일을 사용하여 브리지, 채널 또는 라우트와 연결된 메트릭의 모니터링 수준을 구성할 수는 없습니다. 브리지, 채널 또는 라우트에 대한 모니터링을 사용자 지정하려면 필터를 사용해야 합니다. 필터를 사용하여 브리지, 채널 및 라우트에 대한 모니터링을 사용자 지정하는 방법에 대한 자세한 내용은 [선택적 모니터링을 위한 필터 구성](#) (페이지 206)을 참조하십시오.

개별 메트릭의 모니터링 수준을 변경하려면

1. *MonitoringLevel.xml* 파일을 텍스트 편집기에서 엽니다.
2. 메트릭 설정을 수정하려는 구성 요소에 대한 *<MetricGroup>* 단원을 찾습니다. 예를 들어 서버 메트릭에 대한 모니터링 수준을 변경하려면 *<MetricGroup name="Server">* 단원을 찾습니다.

EMS 서버 인스턴스를 모니터링하는 데 사용할 수 있는 모든 메트릭이 각 메트릭 이름의 특성으로 정의된 메트릭 모니터링 수준과 함께 나와 있습니다. 예를 들어 "Queue Count"(큐 수) 및 "Topic Count"(항목 수) 메트릭은 기본적으로 모니터링 수준을 *minimum* 으로 설정하면 수집됩니다.

```
<Metric level="Minimum">Queue Count</Metric>
<Metric level="Minimum">Topic Count</Metric>
```

- 모니터링 수준을 수정하려는 메트릭을 찾은 다음 해당 메트릭의 *level* 특성을 새 모니터링 수준으로 수정합니다.

예를 들어 "Queue Count"(큐 수) 및 "Topic Count"(항목 수)의 모니터링 수준을 *minimum* 에서 *recommended* 로 변경하려면 다음과 같이 합니다.

```
<Metric level="Recommended">Queue Count</Metric>
<Metric level="Recommended">Topic Count</Metric>
```

한 메트릭의 모니터링 수준을 *minimum* 에서 *recommended* 또는 *full* 로 변경하면 *EMSMonitor* 에이전트가 로그 파일에 경고를 기록한 후 경고 메시지를 표시합니다. 다른 유형의 변경에서는 에이전트에서 경고가 나타나지 않습니다.

메트릭의 *minimum* 설정을 변경하면 일부 데이터가 그래프와 대시보드에 표시 또는 포함되지 않을 수 있다는 점에 주의하십시오. 메트릭을 *minimum* 에서 *recommended* 또는 *full* 로 변경하기 전에 미리 구성되거나 사용자 지정된 뷰 또는 대시보드에서 메트릭이 사용되고 있는지 여부를 확인해야 합니다.

- 모니터링 수준 변경 사항을 적용하는 시작 스크립트를 사용하여 *EMSMonitor* 에이전트를 다시 시작합니다.

변경을 수행하고 에이전트를 다시 시작한 후에는 모니터링 수준이 *recommended* 로 설정된 서버 인스턴스에서만 "Queue Count"(큐 수) 및 "Topic Count"(항목 수) 메트릭이 수집됩니다.

<MetricGroup name> 및 *<Metric level>* 특성은 대/소문자를 구분하지 않습니다. 하지만 *MonitoringLevel.xml* 파일의 메트릭 이름은 대/소문자를 구분합니다.

에이전트는 *MonitoringLevel.xml* 파일을 사용하여 각 모니터링 수준의 메트릭을 결정합니다. 메트릭이 이 파일에 나와 있지 않거나 이 파일에서 삭제된 경우 에이전트가 그 메트릭에 대한 정보를 수집하거나 보고할 수 없습니다.

MonitoringLevel.xml 파일이 손상되거나 *EMSMonitor* 에이전트가 해당 파일을 읽을 수 없는 경우 에이전트는 로그에 오류 메시지를 기록하고 시작 시 검색된 메트릭을 보고합니다.

암호화된 암호 만들기

EMSMonitor 에이전트가 사용자가 지정한 EMS 서버에 연결하려면 인증을 위한 유효한 사용자 자격 증명을 제시할 수 있어야 합니다. 클라이언트와 EMS 서버 사이에서 SSL(Secure Socket Layer) 연결을 사용하고 에이전트는 클라이언트의 보안 인증서를 확인하도록 구성된 EMS 서버에 연결하는 경우, 에이전트는 유효성 검사를 위해 서명된 인증서도 제시할 수 있어야 합니다.

사용자 암호 또는 클라이언트 인증서 경로 및 암호를 암호화하고 저장하려면

1. *emsPwdEncryptor* 스크립트를 텍스트 편집기에서 열고 *JAVA_HOME* 환경 변수를 적절한 디렉터리로 설정합니다. 예:
`set JAVA_HOME=C:\Java\jdk1.5.0_10`
2. *emsPwdEncryptor* 스크립트를 실행합니다.

TibcoEMSMonitor.properties 파일의 *ems.server.list* 속성 목록에 지정한 각 EMS 서버 인스턴스에 대해 사용자 이름과 암호를 지정하라는 메시지가 나타납니다.

- 서버 인스턴스에 대해 암호화된 암호를 만들려면 *y*를 입력한 다음 EMS 사용자 이름 및 암호를 입력합니다.
- 서버 인스턴스에 대해 암호화된 암호를 만들지 않으려면 *n*을 입력합니다.

각 인스턴스에 연결할 때 사용할 사용자 이름 및 암호를 입력하는 과정을 마치면 사용자 이름과 암호화된 버전의 암호가 *TibcoEMSMonitor.properties* 파일에 기록됩니다. 연결이 성공하려면 EMS 서버가 지정된 사용자 이름 및 암호를 인증할 수 있어야 합니다.

3. SSL(Secure Socket Layer) 속성을 구성할지 묻는 메시지가 나타나면 *y* 또는 *n*을 입력합니다.

SSL 연결을 사용하고 있고 클라이언트의 보안 인증서를 확인하도록 구성된 EMS 서버가 있는 경우 *y*를 입력합니다. 그런 다음 클라이언트 인증서 경로 및 클라이언트 인증서 읽기에 사용할 암호를 지정할 수 있습니다.

정보 입력을 마치면 *client.identity* 및 암호화된 버전의 *ssl.password* 속성이 *EMSMonitor* 에이전트의 *TibcoEMSMonitor.properties* 파일에 기록됩니다.

- *client.identity* 속성은 EMS 서버가 *EMSMonitor* 에이전트의 ID 를 검증하는 데 사용할 수 있는 인증서의 경로를 지정합니다. 예: *client.identity=C:/Tibco/TibcoEMSMonitor/certs/client_identity.p12*
- *ssl.password* 속성은 클라이언트 보안 인증서에 대한 암호화된 암호를 지정합니다.

SSL 을 사용하여 TIBCO EMS 서버 인스턴스에 연결

TIBCO EMS 가 네트워크 연결에 SSL(Secure Socket Layer) 프로토콜을 사용하도록 구성되어 있는 경우 SSL을 사용하여 TIBCO EMS 서버에 연결하도록 *EMSMonitor* 에이전트를 구성해야 합니다. *EMSMonitor* 에이전트를 SSL을 사용하도록 구성하려면 *TibcoEMSMonitor.properties* 파일에서 여러 속성을 설정해야 합니다. 이러한 속성 중 일부는 각 EMS 서버 인스턴스마다 개별적으로 설정해야 합니다. 그 외의 속성은 모든 EMS 서버 인스턴스에 대해 한 번만 설정하면 됩니다.

TIBCO EMS 서버 인스턴스에 대한 SSL 연결 정보를 정의하려면

1. *TibcoEMSMonitor.properties* 파일을 텍스트 편집기에서 엽니다.
2. 모니터링하려는 EMS 서버 인스턴스의 이름을 확인합니다.
3. <*ServerInstance*>.ssl.connection 속성을 *enable* 또는 *disable*로 구성하여 각 서버 인스턴스의 암호화된 통신 사용 여부를 설정합니다. 이 속성에서는 서버 인스턴스 이름을 지정해야 합니다.

예를 들어 에이전트가 EMS 서버 인스턴스 *tibco_ems_srv01*에 SSL을 사용하여 연결하도록 구성하려면 다음과 같이 합니다.

```
tibco_ems_srv01.ssl.connection=enable
```

4. `<ServerInstance>.verify.host` 속성을 구성하여 EMSMonitor 에이전트가 EMS 서버의 인증서를 검증할지 여부를 지정합니다.

이 속성에서는 서버 인스턴스 이름을 지정해야 합니다. 예를 들어 에이전트가 EMS 서버 인스턴스 `tibco_ems_srv01`에 연결할 때 검증을 요구하도록 하려면 다음과 같이 합니다.

```
tibco_ems_srv01.verify.host=true
```

이 속성을 `true`로 설정하면 에이전트가 `trusted.certificates` 속성에 정의된 목록을 사용하여 EMS 서버의 보안 인증서를 검증합니다.

5. `trusted.certificates` 속성을 구성하여 EMSMonitor 에이전트가 서버의 인증서를 검증하는 데 사용할 신뢰할 수 있는 인증서의 첨표로 구분된 목록을 지정합니다. 예:

```
trusted.certificates=C:/Tibco/wily/TibcoEMSMonitor/certs/tbx_root.cert.pem
```

`verify.host` 속성이 `true`로 설정되어 있으면 이 속성이 필요하며, 이 속성은 SSL 을 사용하는 모든 EMS 서버 인스턴스에 적용할 수 있습니다.

6. `<ServerInstance>.verify.hostname` 속성을 구성하여 에이전트가 서버 인증서의 CN(Common Name) 필드를 검증해야 하는지 여부를 지정합니다.

```
tibco_ems_srv01.verify.hostname=true
```

이 속성을 `true`로 설정하면 에이전트가 연결된 호스트의 이름이나 `<ServerInstance>.expected.name` 속성에 지정된 이름을 서버 인증서의 CN(Common Name) 필드 값과 비교합니다. 두 이름이 일치하지 않으면 에이전트가 연결을 거부합니다.

이 속성이 `false`로 설정되어 있으면 에이전트가 서버 이름을 확인하지 않고 서버와 SSL 연결을 설정합니다.

7. `<ServerInstance>.expected.hostname` 속성을 구성하여 EMSMonitor 에이전트가 서버 인증서의 CN(Common Name) 필드 값과 비교할 이름을 지정합니다.

```
tibco_ems_srv01.expected.hostname=tbxserver
```

8. `cipher.suites` 속성을 구성하여 EMSMonitor 에이전트가 SSL 기반 EMS 서버와의 통신을 암호화하는 데 사용할 수 있는 암호 그룹의 첨표로 구분된 목록을 지정합니다.

cipher.suites 속성은 선택 사항입니다. *EMSMonitor* 에이전트는 모니터링 중인 EMS 서버에서 지원하는 모든 암호화 패키지를 사용할 수 있습니다. 이 속성을 설정하면 모든 SSL 기반 EMS 서버 인스턴스에 설정이 적용됩니다.

암호 그룹에 표준 알고리즘 이름이나 OpenSSL 이름을 사용할 수 있습니다. 예를 들어 *cipher.suites* 속성을 *RC4-MD5*(OpenSSL 이름을 사용하는 경우)로 설정하거나 *SSL_RSA_WITH_RC4_128_MD*(표준 이름을 사용하는 경우)로 설정하여 동일한 암호 그룹을 나타낼 수 있습니다.

예:

`cipher.suites=RC4-MD5,RC4-SHA`

9. 클라이언트의 보안 인증서를 확인하도록 EMS 서버를 구성한 경우 *emsPwdEncryptor* 유ти리티를 실행하여 클라이언트 인증서의 경로 및 암호화된 암호를 설정합니다.

emsPwdEncryptor 유ти리티 사용에 대한 자세한 내용은 [암호화된 암호 만들기](#) (페이지 214)를 참조하십시오.

EMS 서버가 클라이언트 인증서를 확인할 필요가 없는 경우 이 단계를 건너뛸 수 있습니다.

EMSMonitor 에이전트 시작

연결 속성을 구성한 후에는 추가 속성을 구성하거나 에이전트를 시작하여 EMS 서버 인스턴스 모니터링을 시작할 수 있습니다. 기본 모니터링 속성을 사용하려면 다음과 같이 할 수 있습니다.

- 시작 스크립트에서 에이전트 시작
- 에이전트를 Windows 서비스로 실행되도록 구성

시작 스크립트를 사용하여 EMSMonitor 시작

UNIX 컴퓨터에서 *EMSMonitor* 에이전트를 시작 및 중지하려면 *EMSMonitor.sh* 스크립트를 사용합니다. Windows에서는 *EMSMonitor.bat* 스크립트를 사용하거나 *EMSMonitor* 에이전트를 Windows 서비스로 실행되도록 구성할 수 있습니다.

다음 단계를 수행하십시오.

1. EMSMonitor 시작 스크립트를 텍스트 편집기에서 엽니다.
2. TIBCO_EMS_HOME 환경 변수를 TIBCO Enterprise Message Service 설치 디렉터리로 설정합니다. 예:

```
set TIBCO_EMS_HOME=C:\tibco\ems\5.1
```
3. JAVA_HOME 환경 변수를 적절한 디렉터리로 설정합니다. 예:

```
set JAVA_HOME=C:\Java\jdk1.5.0_10
```

JRE 버전 1.5 이상에 해당: TIBCO Enterprise Message Service 를 모니터링하려면 JRE 버전 1.5 이상이 필요합니다.
4. EMSMonitor 스크립트를 실행합니다. UNIX 컴퓨터에서 EMSMonitor 를 실행하는 경우 스크립트가 다음과 같은 명령줄 매개 변수를 지원합니다.
`EMSMonitor.sh [start|stop|restart|status]`
새 에이전트 인스턴스를 시작하려면 start 옵션을 사용합니다. 예를 들어 UNIX 컴퓨터에서는 다음 명령을 사용합니다.
`./EMSMonitor.sh start`

EMSMonitor 에이전트를 Windows 서비스로 실행

Windows에서 EMSMonitor 에이전트를 설치하면 이 에이전트를 Windows 서비스로 실행하는 옵션이 있습니다. 에이전트를 Windows 서비스로 실행하면 다음과 같은 이점이 있습니다.

- 호스트 컴퓨터를 시작하거나 중지하면 에이전트 서비스가 자동으로 시작되거나 중지됩니다.
- 에이전트를 콘솔이 아니라 백그라운드 프로세스로 실행할 수 있으므로 면조 및 무단 액세스로부터 보호할 수 있습니다.
- 사용자가 현재 세션에서 로그오프해도 에이전트가 계속 실행됩니다.

EMSMonitor 에이전트를 Windows 서비스로 실행하려면

1. 이전 단원에서 설명한 대로 EMS 서버, *IntroscopeAgent.profile* 및 *TibcoEMSMonitor.properties* 파일을 설치하고 구성합니다.
2. JAVA_HOME 환경 변수가 올바른 JVM 으로 설정되어 있고, 사용자 변수가 아닌 시스템 변수로 구성되어 있는지 확인합니다.
3. 텍스트 편집기에서 *TibcoEMSMonitor\Windows Service\jsw-3.2.3\conf\wrapper.conf* 파일을 엽니다.

4. *TIBCO_EMS_HOME* 환경 변수를 TIBCO Enterprise Message Service 설치 디렉터리로 설정합니다. 예:
set TIBCO_EMS_HOME=C:\tibco\ems\5.1
5. *wrapper.conf* 파일을 저장합니다.
6. 명령 창을 열고
TibcoEMSMonitor\Windows Service\RegisterEMSMonitorAgentService.bat 파일을 실행하여 EMSMonitor 앤이전트를 Windows 서비스로 등록합니다.
서비스로 등록된 EMSMonitor 앤이전트를 나중에 제거하려면 *DeRegisterEMSMonitorAgentService.bat* 파일을 실행합니다.

Enterprise Manager 확장을 사용하도록 설정

CA APM for TIBCO Enterprise Message Service 파일은 기본적으로 Enterprise Manager를 설치할 때 *<EM_Home>/examples* 디렉터리에 설치됩니다. CA APM for TIBCO Enterprise Message Service를 사용하도록 설정하려면 Enterprise Manager 파일을 *<EM_Home>/examples* 디렉터리에서 Enterprise Manager 흄 디렉터리의 적절한 위치로 복사하거나 이동해야 합니다.

참고: CA APM for TIBCO Enterprise Message Service를 사용하려면 Enterprise Manager에서 CA APM for SOA를 사용하도록 설정해야 합니다. CA APM for SOA Enterprise Manager 확장을 사용하도록 설정하는 자세한 방법은 [Enterprise Manager에서 확장을 사용하도록 설정](#) (페이지 41)을 참조하십시오.

다음 단계를 수행하십시오.

1. CA APM for TIBCO Enterprise Message Service 디렉터리인 *SOAExtensionForTibcoEMS*가 *<EM_Home>/examples* 디렉터리에 있는지 확인한 다음 *<EM_Home>/examples/SOAExtensionForTibcoEMS* 디렉터리의 파일을 Enterprise Manager 디렉터리 구조의 해당 위치로 복사합니다. 예를 들어 *<EM_Home>/examples/SOAExtensionForTibcoEMS/ext* 디렉터리의 파일을 *<EM_Home>/ext* 디렉터리로 복사합니다.
2. Enterprise Manager가 클러스터 환경의 수집기인 경우 CA APM for TIBCO Enterprise Message Service 관리 모듈 *TibcoEMSMgmtModule.jar*을 *<EM_Home>/config/modules* 디렉터리에서 제거합니다.

관리 모듈만 MOM 컴퓨터로 사용 중인 Enterprise Manager 의 `<EM_Home>/config/modules` 디렉터리로 복사해야 합니다. 다른 모든 파일과 스크립트는 Collector Enterprise Manager 와 MOM Enterprise Manager 모두에 설치해야 합니다.

3. Workstation을 다시 시작하여 SOA extension for TIBCO Enterprise Message Service 의 고유한 대시보드 및 "개요" 템을 로드합니다.

대시보드를 사용하여 TIBCO EMS 모니터링

SOA extension for TIBCO Enterprise Message Service 에는 EMS 환경의 전반적인 건전성을 모니터링하는 데 사용할 수 있는 몇 개의 사전 구성된 대시보드가 포함되어 있습니다. 대시보드는 배포된 에이전트에서 데이터를 집계하여 성능 정보를 요약하며 문제를 신속하게 진단하고 해결할 수 있도록 해 줍니다.

일반적으로 대시보드는 다음과 같은 작업을 가능하게 하므로 환경을 모니터링하는 시작점으로 사용됩니다.

- Enterprise Message Service 의 주요 구성 요소에 대하여 전반적인 건전성, 성능, 가용성 및 현재 상태를 한눈에 모니터링
- 하위 수준 메트릭이 주의 또는 위험 임계값이 초과되었음을 알리는 경우 프로덕션 응용 프로그램 환경의 잠재적 문제를 조기에 식별
- 성능 정보를 드릴다운하여 지연 중이거나 오류가 발생 중인 Enterprise Message Service 구성 요소를 격리 및 식별

사전 구성된 TIBCO Enterprise Message Service 대시보드는 TIBCO Enterprise Message Service 관리 모듈(*TibcoEMSMgmtModule.jar*)의 일부로 Enterprise Manager extension for TIBCO Enterprise Message Service 에 패키지되어 있습니다.

TIBCO Enterprise Message Service 관리 모듈은 TIBCO Enterprise Message Service에 대해 다음과 같은 사전 구성된 대시보드를 제공합니다.

Tibco EMS - 개요

Enterprise Message Service의 주요 동작에 대한 상위 수준 개요로, EMS 서비스와 에이전트 사이의 연결 상태, EMS 서버의 전반적인 건전성에 대한 경고 표시기, 최대 큐 크기, 최대 항목 크기, 경로와 채널의 전반적인 건전성 및 사용 중인 연결의 비율을 포함합니다.

Tibco EMS - 서버

모든 EMS 서버의 요약된 상태로, EMS 서버와 백업 서버의 상태에 대한 경고 표시기 및 사용 중인 연결의 비율을 포함합니다.

대시보드에는 연결 수, 소비자 수, 생산자 수, 전체 인바운드 및 아웃바운드 메시지 비율, 보류 중인 메시지 수가 요약된 그래프도 표시됩니다.

Tibco EMS - 큐

모든 EMS 큐에 대한 요약된 상태로, 큐의 인바운드 및 아웃바운드 메시지에 대한 경고 표시기와 그래프, 최대 큐 크기에 대한 경고 표시기, 큐의 보류 중인 메시지 수와 소비자 수에 대한 그래프를 포함합니다.

Tibco EMS - 항목

모든 EMS 항목에 대한 요약된 상태로, 항목의 인바운드 및 아웃바운드 메시지에 대한 경고 표시기와 그래프, 최대 항목 크기에 대한 경고 표시기, 보류 중인 메시지 수에 대한 그래프를 포함합니다.

대시보드에는 총 구독자 수, 활성 및 지속 구독자 수도 포함됩니다.

Tibco EMS - 라우트

모든 EMS 라우트에 대한 요약된 상태로, 라우트의 인바운드 및 아웃바운드 메시지 비율에 대한 경고와 그래프, 경로 상태에 대한 경고 표시기, 백로그 수와 백로그 크기에 대한 그래프(적용 가능한 경우)를 포함합니다.

Tibco EMS - 채널

모든 EMS 채널에 대한 요약된 상태로, 전체적인 채널 상태, 채널 메시지 비율 및 백로그 수에 대한 경고 표시기와 메시지 비율, 바이트 비율, 백로그 수 및 백로그 크기에 대한 그래프를 포함합니다.

다음 단계를 수행하십시오.

1. Enterprise Manager 가 실행되고 있지 않으면 시작합니다.
2. Workstation 을 시작하고 SOA extension for TIBCO EMS 가 설치된 Enterprise Manager 에 로그인합니다.
3. "Workstation" > "새 콘솔"을 클릭합니다.
4. "대시보드" 드롭다운 목록에서 TIBCO Enterprise Message Service 대시보드 중 하나를 선택합니다.
예를 들어 연결 상태, 사용된 연결의 백분율, 최대 큐 및 항목 크기를 포함하여 TIBCO Enterprise Message Service 환경의 전반적인 건전성에 대한 정보를 보려면 "Tibco EMS - 개요" 대시보드를 선택합니다.
5. 구성 요소를 두 번 클릭하여 해당 구성 요소의 대시보드를 열고 서버, 큐, 항목, 라우트 또는 채널에 대하여 더 많은 세부 정보를 확인합니다. 예를 들어 "서버" 탭을 두 번 클릭하여 "Tibco EMS - 서버" 대시보드를 표시합니다.

TIBCO EMS 메트릭 이해 및 보기

EMSMonitor 에이전트는 로컬 및 원격 EMS 서버 인스턴스를 모니터링하여 Tibco EMS 서버 > `<host_name>` > `<EMS_server_instance_name>` 노드 아래에 해당 인스턴스의 종합적인 상태에 대한 데이터를 제공할 수 있습니다.

다음과 같은 메트릭 범주의 메트릭을 사용하여 TIBCO EMS 구성 요소의 성능 및 상태를 모니터링 할 수 있습니다.

Bridges(브리지)

브리지를 사용하면 여러 대상 간에 메시지를 라우팅 할 수 있습니다. 즉 한 대상에 보낸 메시지가 브리지된 모든 대상에 전달됩니다. 한 대상에서 유형이 같거나 다른 하나 이상의 다른 대상으로 연결되는 브리지를 만들 수 있습니다. 예를 들어 항목에서 큐로 연결되거나 큐에서 항목으로 연결되는 브리지를 만들거나 한 대상과 여러 대상 간에 브리지를 만들 수 있습니다.

"Bridges"(브리지) 노드 아래에서 정의한 각 브리지와 연결된 개별 목표 대상의 메트릭을 찾을 수 있습니다.

채널

멀티캐스트 메시징은 구독하는 각 소비자에게 개별적으로 메시지 복사본을 보내는 대신 한 번에 많은 소비자에게 메시지를 브로드캐스팅합니다. 멀티캐스트 메시징을 사용하면 서버가 멀티캐스트 채널을 통해 멀티캐스트 지원 항목으로 메시지를 보냅니다. 채널은 서버가 메시지를 보낼 멀티캐스트 포트 및 멀티캐스트 그룹 주소를 결정합니다.

"Channels"(채널) 노드 아래에서 개별 멀티캐스트 채널에 대한 메트릭을 찾을 수 있습니다.

마지막 검사

"마지막 검사" 노드 아래에서 EMS 서버 인스턴스와 에이전트 간의 연결과 관련된 메트릭을 찾을 수 있습니다.

큐

큐는 Enterprise Message Service 네트워크의 클라이언트나 다른 큐로 전달할 대기 메시지를 저장하는 임시 저장소 개체입니다.

"큐" 노드 아래에서 각 큐의 인바운드 및 아웃바운드 비율과 전달되거나 전송 중인 메시지 수와 같은 개별 큐에 대한 메트릭을 찾을 수 있습니다.

라우트

라우트는 두 TIBCO Enterprise Message Service 서버를 서버 쌍으로 연결하는 데 사용됩니다. 쌍을 이루는 각 서버는 라우트를 따라 다른 서버의 해당하는 대상에 메시지를 전달합니다. 라우트는 두 서버에서 이름이 같은 전역 항목에 대한 메시지나 큐 소유자가 같은 라우트된 큐에 대한 메시지만 전달합니다.

"라우트" 노드 아래에서 각 라우트의 인바운드 및 아웃바운드 비율이나 라우트 백로그의 메시지 수와 같은 서버 사이에 정의된 개별 라우트에 대한 메트릭을 찾을 수 있습니다.

Server

"서버" 범주는 Enterprise Message Service의 기본 런타임 프로세스에 대한 메트릭을 제공합니다. EMS 서버 프로세스는 메시징 트랜잭션을 구성하는 다른 구성 요소를 생성하고 관리합니다.

"서버" 노드 아래에서 연결 수, 사용 가능한 메모리, 서버가 관리 중인 큐 및 항목 수와 같은 서버 프로세스에 대한 메트릭을 찾을 수 있습니다.

항목

항목은 게시자가 메시지를 작성할 수 있고 구독자가 게시된 메시지를 받을 수 있는 논리적 주제를 나타냅니다. 메시지의 복사본이 하나만 저장되고 하나의 수신자만 메시지를 받을 수 있는 큐와 달리, 항목은 메시지의 단일 복사본을 여러 관련 구독자에게 게시할 수 있는 EMS 서버로 관리됩니다.

"항목" 노드 아래에서 각 항목의 인바운드, 아웃바운드 및 보류 메시지 수와 같은 개별 항목에 대한 메트릭을 찾을 수 있습니다.

Investigator에서 TIBCO EMS 메트릭 요약을 보고 탐색하려면

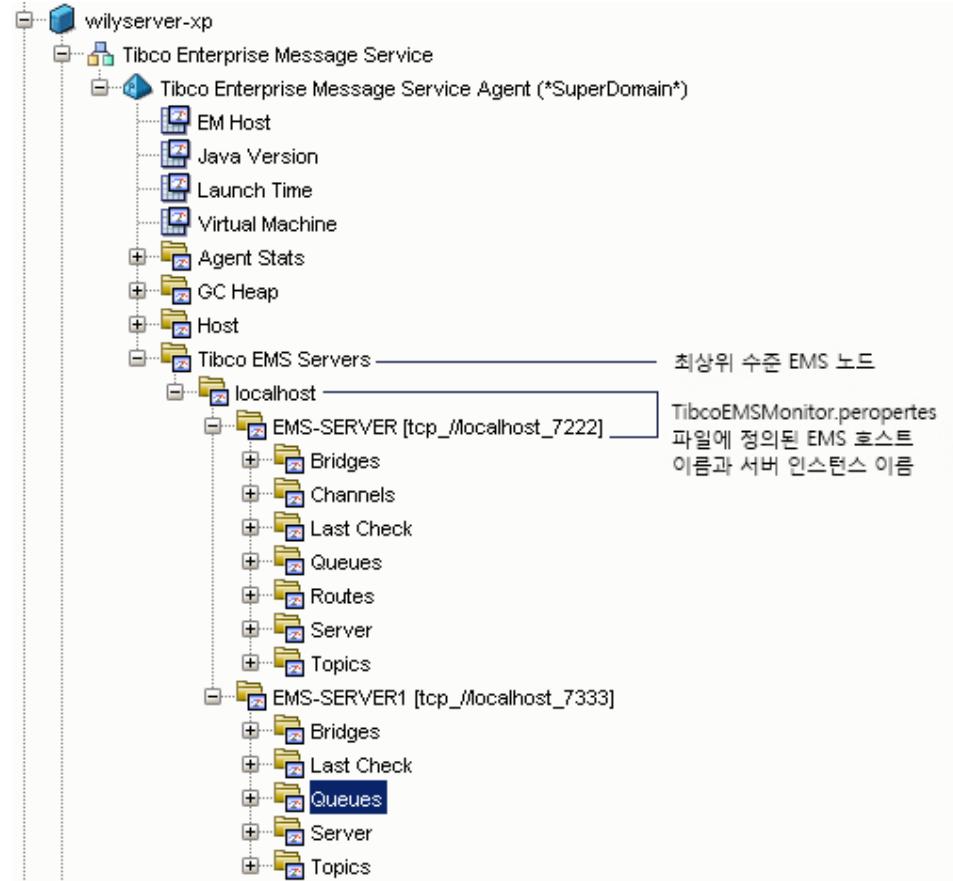
1. 에이전트 노드와 "Tibco Enterprise Message Service" 노드를 확장한 다음 "Tibco EMS 서버"를 클릭하여 모니터링되는 모든 서버 인스턴스와 인스턴스의 현재 상태가 나와 있는 "개요" 탭을 표시합니다.
2. 서버 인스턴스를 선택하여 해당 서버 인스턴스의 가장 중요한 상태 메트릭을 "개요" 탭에 그래픽 형식으로 표시합니다.
3. "구성" 탭을 클릭하여 선택한 서버 인스턴스의 구성 메트릭을 표시합니다.

4. 서버 인스턴스를 확장하고 원하는 하위 노드를 클릭하여 뷰어 창의 탭에 해당 메트릭 범주와 관련된 요약 또는 구성 정보를 표시합니다. 예를 들어 "큐" 노드를 선택한 다음 "임시 큐" 탭을 클릭하여 모든 임시 큐에 대한 메트릭 요약을 표시하거나, "동적 큐" 탭을 클릭하여 모든 동적 큐의 메트릭 요약을 표시합니다.
5. 하위 노드를 확장하거나 특정 큐 또는 항목 같은 개별 구성 요소를 선택하여 개별 구성 요소에 대한 자세한 정보와 각 구성 요소와 연결된 메트릭을 표시합니다.

Investigator에서 TIBCO EMS 메트릭 노드를 보고 탐색하려면

1. 에이전트를 확장한 다음 "Tibco EMS 서버" 노드를 확장하여 모니터링 중인 TIBCO EMS 서버 인스턴스의 호스트 이름을 표시합니다.
2. *TibcoEMSMonitor.properties* 파일에서 정의한 호스트 이름에 해당하는 개별 *server_name* 노드를 확장합니다.

3. *TibcoEMSMonitor.properties* 파일에 지정한 서버 인스턴스의 개별 TIBCO EMS 서버 *instance_name* 노드를 확장하여 최상위 수준 TIBCO EMS 메트릭 범주의 하위 노드를 표시합니다. 예:



4. 하위 노드를 확장하여 해당 메트릭 범주와 관련된 정보를 표시합니다. 예를 들어 "큐" > "정적 큐" > *<queue_name>*을 클릭하여 선택한 큐의 "구성" 또는 "상태" 메트릭을 표시합니다.
5. 하위 노드를 더 확장하여 선택한 큐의 "구성" 또는 "상태" 메트릭을 표시합니다. 예를 들어 "배달되지 않은 메시지 큐" > *<queue_name>* > "상태" 노드를 확장하여 선택한 큐의 상태 메트릭을 표시합니다.

마지막 검사에 대한 메트릭

EMSMonitor 에이전트는 TIBCO EMS 서버 및 해당 구성 요소의 성능 및 작업 관련 메트릭과 함께 각 EMS 서버와 에이전트 간의 연결 관련 메트릭을 수집합니다. "마지막 검사" 노드 아래에 있는 메트릭은 30 초마다 수집됩니다. 이러한 메트릭의 수집 빈도는 지연 시간이나 정적 빈도 구성 속성으로 제어되지 않으며 모든 모니터링 수준에서 메트릭이 수집됩니다.

다음 메트릭을 사용하여 각 EMS 서버와 *EMSMonitor* 에이전트 간의 연결 상태를 평가할 수 있습니다.

에이전트 - EMS 연결 상태

EMSMonitor 에이전트가 현재 TIBCO EMS 서버에 연결되어 있는지 여부를 나타내는 텍스트 문자열입니다.

메트릭 값이 "Connected"(연결됨)이면 연결이 존재함을 나타냅니다. 값이 "Not Connected"(연결되지 않음)이면 에이전트와 EMS 서버의 연결이 끊긴 상태임을 나타냅니다.

에이전트 - EMS 연결 상태 값

EMSMonitor 에이전트가 현재 TIBCO EMS 서버에 연결되어 있는지 여부를 나타내는 숫자 값입니다.

메트릭 값이 0이면 에이전트가 EMS 서버에 연결되어 있음을 나타냅니다. 메트릭 값이 1이면 에이전트가 EMS 서버에 연결되어 있지 않음을 나타냅니다.

EMS 서버 이름

모니터링되는 EMS 서버의 이름입니다.

Timestamp(타임스탬프)

"마지막 검사" 메트릭을 최근에 수집한 날짜 및 시간입니다. 이 메트릭의 형식은 다음과 같습니다.

yyyy-MM-dd HH:mm:ss

큐에 대한 메트릭

다음 구성 메트릭은 TIBCO EMS 의 동적 및 정적 큐에 사용할 수 있습니다.

Bridged Queue(브리지된 큐)

텍스트 문자열에서 큐에 브리지된 대상이 있는지 여부를 나타냅니다.

- 메트릭 값이 "true"이면 큐에 브리지된 대상이 있음을 나타냅니다.
- 값이 "false"이면 큐에 브리지된 대상이 없음을 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 full 로 설정한 경우에만 수집됩니다.

Exclusive(단독)

텍스트 문자열에서 큐에 단독 속성이 설정되어 있는지 여부를 나타냅니다.

- 메트릭 값이 "true"이면 큐가 단독 큐로 구성되어 있음을 나타냅니다.
 - 값이 "false"이면 큐가 단독 큐로 구성되어 있지 않음을 나타냅니다.
- 기본적으로 이 메트릭은 모니터링 수준을 minimum 으로 설정한 경우에 수집됩니다.

Expiration (Sec)(만료 시간(초))

메시지가 선택된 큐에서 만료되기 전의 최대 시간을 나타냅니다. 이 메트릭 값이 0 이면 메시지가 만료되지 않습니다.

서버 만료 속성이 큐에 대해 설정된 경우 이 속성의 값은 메시지 생산자가 설정하는 JMSExpiration 값을 재정의합니다.

기본적으로 이 메트릭은 모니터링 수준을 minimum 으로 설정한 경우에 수집됩니다.

Failsafe(폐일 세이프)

텍스트 문자열에서 큐가 폐일 세이프 대상으로 구성되어 있는지 여부를 나타냅니다.

- 메트릭 값이 "true"이면 큐가 폐일 세이프 대상으로 구성되어 있음을 나타냅니다.
- 값이 "false"이면 큐가 폐일 세이프 대상으로 구성되어 있지 않음을 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 **full**로 설정한 경우에만 수집됩니다.

참고: 이 메트릭은 EMS 버전 4.4에만 적용됩니다.

Global(전역)

텍스트 문자열에서 큐가 한 서버에서 다른 서버로 메시지를 라우팅하는 데 사용할 수 있는 전역 큐로 구성되어 있는지 여부를 나타냅니다.

- 메트릭 값이 "true"이면 큐가 전역 큐임을 나타냅니다.
- 값이 "false"이면 큐가 전역 큐가 아님을 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 **recommended**로 설정한 경우에 수집됩니다.

Is Route Connected(라우트가 연결됨)

선택한 큐의 라우트가 현재 연결되어 있는지 여부를 나타냅니다. 이 메트릭은 선택한 큐가 라우트된 큐인 경우에만 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 **recommended**로 설정한 경우에 수집됩니다.

Maximum Flow Control Bytes(최대 흐름 제어 바이트 수)

이 큐의 흐름 제어를 가능하게 하는 데 사용되는 바이트의 최대 수를 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 **full**로 설정한 경우에만 수집됩니다.

Maximum Messages(최대 메시지 수)

서버에서 이 큐에 대해 보류 중인 메시지로 저장할 수 있는 메시지의 최대 수를 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 **minimum**으로 설정한 경우에만 수집됩니다.

Maximum Redelivery(최대 재배달 횟수)

서버가 선택한 큐에서 큐 수신자로 지정된 메시지 배달을 시도할 수 있는 최대 횟수를 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 **full**로 설정한 경우에만 수집됩니다.

Overflow Policy(오버플로 정책)

최대 메시지 크기 또는 메시지 수를 초과한 경우 큐에 사용되는 오버플로 정책을 나타냅니다. 유효한 오버플로 정책 값은 다음과 같습니다.

- 0 - 기본 오버플로 정책을 나타냅니다. 최대 바이트 수나 메시지 수가 초과되면 서버에서 새 메시지가 거부되고 메시지 생산자에게 오류가 반환됩니다.
- 1 - **discardOld** 오버플로 정책을 나타냅니다. 큐의 메시지가 최대 바이트 수 또는 메시지 수를 초과하면 큐에서 가장 오래된 메시지를 제거하고 메시지 생산자에게 오류를 반환합니다.
- 2 - **rejectIncoming** 오버플로 정책을 나타냅니다. 최대 바이트 수나 메시지 수가 초과되면 서버에서 새 메시지가 거부되고 메시지 생산자에게 오류가 반환됩니다.

기본적으로 이 메트릭은 모니터링 수준을 **full**로 설정한 경우에만 수집됩니다.

Prefetch Count(프리페치 수)

메시지 소비자가 EMS 서버에서 가져올 수 있는 메시지의 최대 수를 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 **recommended**로 설정한 경우에 수집됩니다.

Route Name(라우트 이름)

큐와 연결된 라우트 이름을 식별합니다. 이 메트릭은 선택한 큐가 라우트된 큐인 경우에만 표시됩니다.

기본적으로 이 메트릭은 모니터링 수준을 **recommended**로 설정한 경우에 수집됩니다.

Routed Queue(라우트된 큐)

텍스트 문자열에서 선택된 큐가 라우트된 큐인지 여부를 나타냅니다.

- 메트릭 값이 "true"이면 큐가 라우트된 큐임을 나타냅니다.
- 값이 "false"이면 큐가 라우트된 큐가 아님을 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 **recommended**로 설정한 경우에 수집됩니다.

Secure(보안)

텍스트 문자열에서 큐가 들어오는 연결을 인증하도록 구성되어 있는지 여부를 나타냅니다.

- 메트릭 값이 "true"이면 들어오는 연결이 인증됨을 나타냅니다.
- 값이 "false"이면 들어오는 연결이 인증되지 않음을 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 **minimum**으로 설정한 경우에 수집됩니다.

Sender Name(전송자 이름)

텍스트 문자열에서 큐에 **sender_name** 속성이 구성되어 있는지 여부를 나타냅니다.

- 메트릭 값이 "true"이면 **sender_name** 속성이 설정되어 있음을 나타냅니다.
- 값이 "false"이면 **sender_name** 속성이 설정되어 있지 않음을 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 **full**로 설정한 경우에만 수집됩니다.

Sender Name Enforced(전송자 이름 적용)

텍스트 문자열에서 큐에 **sender_name_enforced** 속성이 구성되어 있는지 여부를 나타냅니다.

- 메트릭 값이 "true"이면 **sender_name** 속성이 적용됨을 나타냅니다.
- 값이 "false"이면 **sender_name** 속성이 적용되지 않음을 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 **full**로 설정한 경우에만 수집됩니다.

Store Name(저장소 이름)

영구 메시지가 저장되는 저장소 이름을 식별합니다.

기본적으로 이 메트릭은 모니터링 수준을 full로 설정한 경우에만 수집됩니다.

참고: 이 메트릭은 EMS 버전 5.x 이상에만 적용됩니다.

다음 상태 메트릭은 TIBCO EMS의 동적 및 정적 큐에 사용할 수 있습니다.

배달된 메시지 수

배달되고 승인된 메시지의 총 수를 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 minimum으로 설정한 경우에만 수집됩니다.

Inbound Byte Rate(인바운드 바이트 비율)

EMS 클라이언트 및 라우트된 서버에서 선택한 큐로 보내는 초당 인바운드 바이트 수를 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 full로 설정한 경우에만 수집됩니다.

In Transit Message Count(전송 중인 메시지 수)

큐 소유자에게 배달되었지만 아직 승인되지 않은 메시지의 총 수를 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 full로 설정한 경우에만 수집됩니다.

인바운드 메시지 비율

EMS 클라이언트 또는 라우트된 서버에서 선택한 큐로 보내는 초당 인바운드 메시지 수를 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 minimum으로 설정한 경우에만 수집됩니다.

Outbound Byte Rate(아웃바운드 바이트 비율)

선택한 큐의 소비자에게 보내거나 다른 서버로 라우트되는 초당 아웃바운드 바이트 수를 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 full로 설정한 경우에만 수집됩니다.

아웃바운드 메시지 비율

선택한 큐의 소비자에게 보내거나 다른 서버로 라우트되는 초당 아웃바운드 메시지 수를 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 **minimum** 으로 설정한 경우에 수집됩니다.

보류 중인 메시지 수

선택한 큐에서 현재 보류 중인 메시지의 총 수를 나타냅니다. 이 메트릭은 모니터링 큐 크기와 동일합니다.

기본적으로 이 메트릭은 모니터링 수준을 **minimum** 으로 설정한 경우에 수집됩니다.

Pending Message Size(보류 중인 메시지 크기)

선택한 큐에서 보류 중인 모든 메시지의 총 크기(KB)를 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 **minimum** 으로 설정한 경우에 수집됩니다.

수신자 수

선택한 큐의 수신자 수를 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 **minimum** 으로 설정한 경우에 수집됩니다.

서버에 대한 메트릭

다음 구성 메트릭은 TIBCO EMS 서버 프로세스에 사용할 수 있습니다.

Authorization Enabled(인증 사용)

서버에서 인증을 사용하는지 여부를 나타냅니다.

이 메트릭이 "True"이면 이 서버에 능동적으로 연결하는 다른 서버가 이름과 암호를 사용하여 스스로를 인증해야 합니다.

기본적으로 이 메트릭은 모니터링 수준을 **minimum** 으로 설정한 경우에 수집됩니다.

Backup Server Name(백업 서버 이름)

선택한 서버 인스턴스의 백업 서버로 구성된 컴퓨터의 호스트 이름 및 서버 인스턴스의 이름을 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 **minimum** 으로 설정한 경우에 수집됩니다.

Client Heartbeat Server Interval (sec)(클라이언트 하트비트 서버 간격(초))

클라이언트에서 서버로 하트비트 메시지를 보내는 시간 간격(초)을 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 full로 설정한 경우에만 수집됩니다.

Client Timeout Server Connection (sec)(클라이언트의 서버 연결 시간 만료(초))

클라이언트가 서버에 대한 연결을 종료하기 전에 서버에서 보낸 하트비트를 대기하는 시간(초)을 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 full로 설정한 경우에만 수집됩니다.

Fault Tolerant Activation Time (sec)(내결함성 활성화 시간(초))

백업 서버가 활성 서버에서 장애가 발생했다고 간주하기 전에 하트비트 메시지를 대기하는 시간(초)을 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 full로 설정한 경우에만 수집됩니다.

Fault Tolerant Reread(내결함성 다시 읽기)

장애 조치 후에 백업 서버가 기본 파일을 제외한 구성 파일을 다시 읽는지 여부를 나타냅니다.

메트릭 값이 "true"이면 파일을 다시 읽음을 나타냅니다. 값이 "false"이면 파일을 다시 읽지 않음을 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 recommended로 설정한 경우에 수집됩니다.

Fault Tolerant Heartbeat Interval (sec)(내결함성 하트비트 간격(초))

활성 서버에서 백업 서버로 하트비트 메시지를 보내는 시간 간격(초)을 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 recommended로 설정한 경우에 수집됩니다.

Fault Tolerant Reconnect Timeout (sec)(내결함성 다시 연결 시간 만료(초))

장애 조치 후 클라이언트가 다시 연결하도록 새 활성 서버가 대기하는 시간(초)을 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 recommended로 설정한 경우에 수집됩니다.

Fault Tolerant Server URL(내결함성 서버 URL)

백업 서버의 URL 을 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 recommended 로 설정한 경우에 수집됩니다.

Flow Control Enabled(흐름 제어 사용)

서버의 메시지 생산자 및 대상에서 흐름 제어를 사용하는지 여부를 나타냅니다.

메트릭 값이 "true"이면 흐름 제어를 사용함을 나타냅니다. 값이 "false"이면 흐름 제어를 사용하지 않음을 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 minimum 으로 설정한 경우에 수집됩니다.

Maximum Log File Size(최대 로그 파일 크기)

로그 파일의 최대 크기(KB)를 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 recommended 로 설정한 경우에 수집됩니다.

최대 연결 수

EMS 서버의 최대 연결 수를 나타냅니다.

이 메트릭이 0 이면 허용되는 연결 수에 제한이 없습니다.

기본적으로 이 메트릭은 모니터링 수준을 recommended 로 설정한 경우에 수집됩니다.

Maximum Message Memory(최대 메시지 메모리)

메시지 저장에 사용할 수 있는 바이트 단위의 최대 메모리 양을 나타냅니다.

이 메트릭이 0 이면 메시지에 사용할 수 있는 메모리 크기에 제한이 없습니다.

기본적으로 이 메트릭은 모니터링 수준을 recommended 로 설정한 경우에 수집됩니다.

Maximum Statistics Memory(최대 통계 메모리)

상세 통계를 수집하는 데 할당할 수 있는 바이트 단위의 최대 메모리 양을 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 full 로 설정한 경우에만 수집됩니다.

Minimum Size Asynchronous Store file(비동기 저장소 파일의 최소 크기)

서버에서 사용하는 비동기 저장소 파일의 최소 크기를 나타냅니다.

이 메트릭의 값은 저장소 파일의 구성에 따라 단위가 MB 또는 GB 일 수 있습니다.

기본적으로 이 메트릭은 모니터링 수준을 full로 설정한 경우에만 수집됩니다.

참고: 이 메트릭은 EMS 버전 4.4에만 적용됩니다.

Minimum Size Store file(저장소 파일의 최소 크기)

서버 저장소 파일의 최소 크기를 나타냅니다.

이 메트릭의 값은 저장소 파일의 구성에 따라 단위가 MB 또는 GB 일 수 있습니다.

기본적으로 이 메트릭은 모니터링 수준을 full로 설정한 경우에만 수집됩니다.

참고: 이 메트릭은 EMS 버전 4.4에만 적용됩니다.

Minimum Size Synchronous Store file(동기 저장소 파일의 최소 크기)

서버에서 사용하는 동기 저장소 파일의 최소 크기를 나타냅니다.

이 메트릭의 값은 저장소 파일의 구성에 따라 단위가 MB 또는 GB 일 수 있습니다.

기본적으로 이 메트릭은 모니터링 수준을 full로 설정한 경우에만 수집됩니다.

참고: 이 메트릭은 EMS 버전 4.4에만 적용됩니다.

Multicast Enabled(멀티캐스트 사용)

서버에서 멀티캐스트 메시징을 사용하는지 여부를 나타냅니다.

메트릭 값이 "true"이면 멀티캐스팅을 사용함을 나타냅니다. 값이 "false"이면 멀티캐스팅을 사용하지 않음을 나타냅니다.

EMS 버전 4.4.x는 멀티캐스트 메시징을 지원하지 않으므로 EMS 4.4.x 서버 인스턴스에서는 이 메트릭이 항상 "false"입니다.

기본적으로 이 메트릭은 모니터링 수준을 minimum으로 설정한 경우에 수집됩니다.

Reserve Memory(예약 메모리)

예약 메모리의 KB 단위 크기를 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 recommended로 설정한 경우에 수집됩니다.

Routing Enabled(라우팅 사용)

서버에서 라우팅을 사용하는지 여부를 나타냅니다.

메트릭 값이 "true"이면 라우팅을 사용함을 나타냅니다. 값이 "false"이면 라우팅을 사용하지 않음을 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 minimum으로 설정한 경우에 수집됩니다.

Server Heartbeat Client Interval (sec)(서버 하트비트 클라이언트 간격(초))

연결을 확인하기 위해 서버에서 클라이언트로 하트비트를 보내는 시간 간격(초)을 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 full로 설정한 경우에만 수집됩니다.

Server Heartbeat Server Interval (sec)(서버 하트비트 서버 간격(초))

이 서버가 다른 서버로 하트비트를 보내는 시간 간격(초)을 나타냅니다. 이 두 서버를 라우트 또는 내결함성 쌍으로 연결할 수 있습니다.

기본적으로 이 메트릭은 모니터링 수준을 full로 설정한 경우에만 수집됩니다.

Server Timeout for Client Connection (sec)(클라이언트 연결의 서버 시간 만료(초))

서버가 클라이언트 연결을 종료하기 전에 클라이언트 하트비트를 대기하는 시간(초)을 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 full로 설정한 경우에만 수집됩니다.

Server Timeout for Server Connection (sec)(서버 연결의 서버 시간 만료(초))

서버가 다른 서버와의 연결을 종료하기 전에 해당 서버의 하트비트를 대기하는 시간(초)을 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 full로 설정한 경우에만 수집됩니다.

서버 시작 시간

yyyy MM-dd HH:mm:ss 형식의 서버 시작 시간을 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 full로 설정한 경우에만 수집됩니다.

Statistics Cleanup Interval (sec)(통계 정리 간격(초))

통계 정리 간격을 초 단위로 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 full로 설정한 경우에만 수집됩니다.

Statistics Enabled(통계 사용)

통계 수집 사용 여부를 나타냅니다.

메트릭 값이 "true"이면 통계를 수집함을 나타냅니다. 값이 "false"이면 통계를 수집하지 않음을 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 full로 설정한 경우에만 수집됩니다.

Store Truncation Enabled(저장소 자르기 사용)

필요한 경우 서버가 저장소 파일 자르기를 시도할지 여부를 나타냅니다.

메트릭 값이 "true"이면 저장소 자르기를 사용함을 나타냅니다. 값이 "false"이면 자르기를 사용하지 않음을 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 recommended로 설정한 경우에 수집됩니다.

참고: 이 메트릭은 EMS 버전 4.4에만 적용됩니다.

Swapping Enabled(스와핑 사용)

메시지 스와핑이 사용되어 서버가 프로세스 메모리에서 디스크로 메시지를 스와핑하도록 허용되는지 여부를 나타냅니다.

메트릭 값이 "true"이면 스와핑을 사용함을 나타냅니다. 값이 "false"이면 스와핑을 사용하지 않음을 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 recommended로 설정한 경우에 수집됩니다.

Tibco RV Transport Enabled(Tibco SmartSockets 전송 사용)

서버에서 TIBCO Rendezvous 메시징을 사용하는지 여부를 나타냅니다.

메트릭 값이 "true"이면 tibrv 및 tibrvcm에서 들어오고 나가는 브리징 전송을 사용함을 나타냅니다. 값이 "false"이면 Rendezvous 전송을 사용하지 않음을 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 full로 설정한 경우에만 수집됩니다.

Tibco SmartSockets Transport Enabled(Tibco SmartSockets 전송 사용)

서버에서 TIBCO SmartSockets 전송 프로토콜을 사용하는지 여부를 나타냅니다.

메트릭 값이 "true"이면 SmartSockets에서 들어오고 나가는 브리징 전송을 사용함을 나타냅니다. 값이 "false"이면 SmartSockets 전송을 사용하지 않음을 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 full로 설정한 경우에만 수집됩니다.

URL

서버 인스턴스 URL을 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 full로 설정한 경우에만 수집됩니다.

버전

EMS 서버 인스턴스의 버전 번호를 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 minimum으로 설정한 경우에만 수집됩니다.

다음 상태 메트릭은 TIBCO EMS 서버 프로세스에 사용할 수 있습니다.

Asynchronous DB Size(비동기 DB 크기)

비동기 저장소 파일의 현재 크기(KB)를 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 **recommended**로 설정한 경우에 수집됩니다.

백업 서버 상태

서버의 현재 상태("실행 중" 또는 "중지됨")를 나타내는 텍스트 문자열을 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 **full**로 설정한 경우에만 수집됩니다.

Backup Server State Value(백업 서버 상태 값)

서버의 현재 상태를 나타내는 숫자 값을 나타냅니다.

0 값은 실행 중인 백업 서버를 나타냅니다. 1 값은 중지된 백업 서버를 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 **minimum**으로 설정한 경우에 수집됩니다.

연결 수

서버의 활성 연결 수를 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 **minimum**으로 설정한 경우에 수집됩니다.

소비자 수

서버에 있는 총 메시지 소비자 수를 나타냅니다. 이 메트릭에는 모든 구독자 및 큐 수신자가 포함됩니다.

기본적으로 이 메트릭은 모니터링 수준을 **minimum**으로 설정한 경우에 수집됩니다.

지속 구독자 수

서버에 있는 지속 구독자 수를 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 **minimum**으로 설정한 경우에 수집됩니다.

Inbound Byte Rate(인바운드 바이트 비율)

서버에서 인바운드 메시지를 받는 초당 바이트 수(Bps) 단위 속도를 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 full로 설정한 경우에만 수집됩니다.

인바운드 메시지 비율

서버에서 받는 초당 인바운드 메시지의 수를 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 minimum으로 설정한 경우에만 수집됩니다.

Log File Size(로그 파일 크기)

로그 파일의 현재 크기(KB)를 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 minimum으로 설정한 경우에만 수집됩니다.

Message Memory Used(사용된 메시지 메모리)

현재 서버에서 메시지 저장에 사용하는 메모리 크기(바이트)를 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 recommended로 설정한 경우에만 수집됩니다.

Outbound Byte Rate(아웃바운드 바이트 비율)

서버에서 아웃바운드 메시지를 보내는 초당 바이트 수(Bps) 단위 속도를 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 full로 설정한 경우에만 수집됩니다.

아웃바운드 메시지 비율

서버에 대한 초당 아웃바운드 메시지의 수를 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 minimum으로 설정한 경우에만 수집됩니다.

보류 중인 메시지 수

서버에 대해 보류 중인 총 메시지 수를 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 minimum으로 설정한 경우에만 수집됩니다.

Pending Message Size(보류 중인 메시지 크기)

서버에 대해 보류 중인 메시지의 총 크기(KB)를 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 minimum으로 설정한 경우에 수집됩니다.

Producers Count(생산자 수)

서버에 있는 총 메시지 생산자 수를 나타냅니다.

이 메트릭에는 모든 항목 게시자 및 큐 전송자가 포함됩니다.

기본적으로 이 메트릭은 모니터링 수준을 minimum으로 설정한 경우에 수집됩니다.

Queue Count(큐 수)

서버의 정적, 동적 및 임시 큐를 비롯한 총 큐 수를 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 minimum으로 설정한 경우에 수집됩니다.

Route Recover Count(라우트 복구 수)

서버에 저장할 수 있는 라우트 복구 메시지의 총 수를 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 full로 설정한 경우에만 수집됩니다.

Route Recover Interval (sec)(라우트 복구 간격(초))

서버에서 라우트 복구 메시지를 확인하는 간격(초)을 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 full로 설정한 경우에만 수집됩니다.

Sessions Count(세션 수)

서버에 있는 클라이언트 응용 프로그램에 의해 생성된 총 세션 수를 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 minimum으로 설정한 경우에 수집됩니다.

서버 상태

서버의 현재 상태("활성" 또는 "대기")를 나타내는 텍스트 문자열을 나타냅니다. 에이전트가 EMS 서버에 연결할 수 없는 경우 서버 상태가 "알 수 없음"으로 표시됩니다.

기본적으로 이 메트릭은 모니터링 수준을 minimum으로 설정한 경우에 수집됩니다.

Server State Value(서버 상태 값)

서버의 현재 상태를 나타내는 숫자 값을 나타냅니다.

0 값은 서버가 "활성" 상태임을 나타냅니다. 1 값은 서버가 "대기" 상태임을 나타냅니다. 2 값은 "알 수 없음" 상태임을 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 minimum으로 설정한 경우에 수집됩니다.

Synchronous DB Size(동기 DB 크기)

동기 저장소 파일의 현재 크기(KB)를 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 recommended로 설정한 경우에 수집됩니다.

Topic Count(항목 수)

서버의 정적, 동적 및 임시 항목을 비롯한 총 항목 수를 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 minimum으로 설정한 경우에 수집됩니다.

추가 정보:

[기본 및 백업 서버 모니터링 정보 \(페이지 243\)](#)

기본 및 백업 서버 모니터링 정보

기본 및 백업 서버 쌍을 모니터링하려는 경우 기본 서버와 백업 서버가 동일한 사용자 이름 및 암호 설정을 사용하도록 구성해야 합니다. 기본 서버가 오프라인일 때 *EMSMonitor* 에이전트가 백업 서버에 연결할 수 있으려면 동일한 사용자 이름 및 암호가 필요합니다.

기본 서버가 오프라인이 될 때 백업 서버가 구성되어 있으면 "서버 상태" 및 "백업 서버 상태" 메트릭이 기본 및 백업 서버 모두의 상태를 계속 보고합니다. 하지만 "Backup Server Name"(백업 서버 이름) 메트릭은 기본 서버에 대해서만 보고됩니다. 쌍을 이루는 서버의 상태에 따라 서버 상태 메트릭이 "활성"에서 "Standby"(대기)로 전환될 수 있으므로 "Backup Server Name"(백업 서버 이름)은 한 번에 하나의 서버에 대해서만 보고될 수 있습니다.

항목에 대한 메트릭

다음 구성 메트릭은 TIBCO EMS 동적 및 정적 항목에 사용할 수 있습니다.

Bridged Topic(브리지된 항목)

텍스트 문자열에서 항목에 대한 브리지된 대상이 있는지 여부를 나타냅니다.

- 메트릭 값이 "true"이면 항목에 대한 브리지된 대상이 있음을 나타냅니다.
- 값이 "false"이면 항목에 브리지된 대상이 없음을 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 **full**로 설정한 경우에만 수집됩니다.

Expiration (Sec)(만료 시간(초))

메시지가 선택된 항목에서 만료되기 전의 최대 시간을 나타냅니다. 이 메트릭 값이 0이면 메시지가 만료되지 않습니다.

서버 만료 속성이 큐에 대해 설정된 경우 속성의 값은 메시지 생산자가 설정하는 **JMSExpiration** 값을 재정의합니다.

기본적으로 이 메트릭은 모니터링 수준을 **minimum**으로 설정한 경우에 수집됩니다.

Failsafe(페일 세이프)

텍스트 문자열에서 항목이 페일 세이프 대상으로 구성되어 있는지 여부를 나타냅니다.

- 메트릭 값이 "true"이면 항목이 페일 세이프 대상으로 구성되어 있음을 나타냅니다.
- 메트릭 값이 "false"이면 항목이 페일 세이프 대상으로 구성되어 있지 않음을 나타냅니다.

참고: 이 메트릭은 EMS 버전 4.4에만 적용됩니다.

Global(전역)

텍스트 문자열에서 항목이 한 서버에서 다른 서버로 메시지를 라우팅하는 데 사용할 수 있는 전역 항목으로 구성되어 있는지 여부를 나타냅니다.

- 메트릭 값이 "true"이면 항목이 전역 항목임을 나타냅니다.
- 값이 "false"이면 항목이 전역 항목이 아님을 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 recommended로 설정한 경우에만 수집됩니다.

Maximum Flow Control Bytes(최대 흐름 제어 바이트 수)

이 항목의 흐름 제어를 가능하게 하는 데 사용되는 바이트의 최대 수를 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 full로 설정한 경우에만 수집됩니다.

Maximum Messages(최대 메시지 수)

서버에서 선택한 항목에 대해 보류 중인 메시지로 저장할 수 있는 메시지의 최대 수를 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 minimum으로 설정한 경우에만 수집됩니다.

Multicast Channel Name(멀티캐스트 채널 이름)

항목에 멀티캐스팅이 사용되도록 설정된 경우 항목과 연결된 멀티캐스트 채널 이름을 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 recommended로 설정한 경우에만 수집됩니다.

Multicast Enabled(멀티캐스트 사용)

항목에 멀티캐스팅을 사용하는지 여부를 나타냅니다.

- 메트릭 값이 "true"이면 항목에 멀티캐스팅이 사용됨을 나타냅니다.
- 값이 "false"이면 멀티캐스팅이 사용되지 않음을 나타냅니다.

참고: EMS 버전 4.4.x에서는 멀티캐스트 메시징이 지원되지 않으므로 이 메트릭은 EMS 4.4.x 서버 인스턴스의 항목에 대해 항상 "false"입니다.

기본적으로 이 메트릭은 모니터링 수준을 recommended로 설정한 경우에만 수집됩니다.

Overflow Policy(오버플로 정책)

최대 메시지 개수 또는 최대 메시지 크기(바이트)를 초과하면 항목에 오버플로 정책이 사용됨을 나타냅니다. 유효한 오버플로 정책 값은 다음과 같습니다.

- 0 - 기본 오버플로 정책을 나타냅니다. 구독자가 메시지의 최대 바이트 수 또는 최대 개수를 초과하면 해당 구독자는 메시지를 받지 않습니다. 메시지 생산자에게는 아무 오류도 반환되지 않습니다.
- 1 - `discardOld` 오버플로 정책을 나타냅니다. 구독자가 메시지의 최대 바이트 수 또는 최대 개수를 초과하면 구독자에게 새 메시지가 배달되기 전에 가장 오래된 메시지가 삭제됩니다.
- 2 - `rejectIncoming` 오버플로 정책을 나타냅니다. 구독자가 메시지의 최대 바이트 수 또는 최대 개수를 초과하면 모든 새로운 메시지가 거부되고 생산자에게 오류가 반환됩니다.

기본적으로 이 메트릭은 모니터링 수준을 `full`로 설정한 경우에만 수집됩니다.

Prefetch Count(프리페치 수)

메시지 소비자가 항목에서 가져올 수 있는 메시지의 최대 수를 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 `recommended`로 설정한 경우에 수집됩니다.

Secure(보안)

텍스트 문자열에서 항목이 들어오는 연결을 인증하도록 구성되어 있는지 여부를 나타냅니다.

- 메트릭 값이 "true"이면 들어오는 연결이 인증됨을 나타냅니다.
- 값이 "false"이면 들어오는 연결이 인증되지 않음을 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 `minimum`으로 설정한 경우에 수집됩니다.

Sender Name(전송자 이름)

텍스트 문자열에서 항목에 `sender_name` 속성이 구성되어 있는지 여부를 나타냅니다.

- 메트릭 값이 "true"이면 `sender_name` 속성이 설정되어 있음을 나타냅니다.
- 값이 "false"이면 `sender_name` 속성이 설정되어 있지 않음을 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 `full`로 설정한 경우에만 수집됩니다.

Sender Name Enforced(전송자 이름 적용)

텍스트 문자열에서 항목에 `sender_name_enforced` 속성이 구성되어 있는지 여부를 나타냅니다.

- 메트릭 값이 "true"이면 `sender_name` 속성이 적용됨을 나타냅니다.
- 값이 "false"이면 `sender_name` 속성이 적용되지 않음을 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 `full`로 설정한 경우에만 수집됩니다.

Store Name(저장소 이름)

영구 메시지가 저장되는 저장소 이름을 식별합니다.

기본적으로 이 메트릭은 모니터링 수준을 `full`로 설정한 경우에만 수집됩니다.

참고: 이 메트릭은 EMS 버전 5.0 이상에만 적용됩니다.

다음 상태 메트릭은 TIBCO EMS 동적 및 정적 항목에 사용할 수 있습니다.

활성 지속 구독자 수

선택한 항목의 현재 활성 지속 구독자 수를 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 `minimum`으로 설정한 경우에 수집됩니다.

지속 구독자 수

선택한 항목의 총 지속 구독자 수를 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 `minimum`으로 설정한 경우에 수집됩니다.

Inbound Byte Rate(인바운드 바이트 비율)

EMS 클라이언트 및 라우트된 서버에서 선택한 항목으로 보내는 초당 인바운드 바이트 수를 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 full로 설정한 경우에만 수집됩니다.

인바운드 메시지 비율

EMS 클라이언트 또는 라우트된 서버에서 선택한 항목으로 보내는 초당 인바운드 메시지 수를 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 minimum으로 설정한 경우에만 수집됩니다.

Outbound Byte Rate(아웃바운드 바이트 비율)

선택한 항목의 소비자에게 보내거나 다른 서버로 라우트되는 초당 아웃바운드 바이트 수를 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 full로 설정한 경우에만 수집됩니다.

아웃바운드 메시지 비율

선택한 항목의 소비자에게 보내거나 다른 서버로 라우트되는 초당 아웃바운드 메시지 수를 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 minimum으로 설정한 경우에만 수집됩니다.

보류 중인 메시지 수

선택한 항목에 대해 보류 중인 메시지 수를 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 minimum으로 설정한 경우에만 수집됩니다.

Pending Message Size(보류 중인 메시지 크기)

선택한 항목에 대해 보류 중인 모든 메시지의 총 크기(KB)를 나타냅니다.

기본적으로 이 메트릭은 모니터링 수준을 minimum으로 설정한 경우에만 수집됩니다.

구독자 수

선택한 항목의 총 구독자 수를 나타냅니다.

이 메트릭에는 지속 및 비지속 구독자가 모두 포함됩니다.

기본적으로 이 메트릭은 모니터링 수준을 **minimum** 으로 설정한 경우에 수집됩니다.

라우트에 대한 메트릭

기본적으로 TIBCO EMS 라우트는 모니터링되지 않지만

TibcoEMSMonitor.properties 파일에 *<ServerInstance>.monitor.routes* 속성을 추가하여 라우트를 모니터링하도록 설정할 수 있습니다. 라우트를 모니터링하도록 설정한 경우 TIBCO EMS 라우트에 대해 다음과 같은 구성 메트릭을 사용할 수 있습니다.

원격 서버 URL

라우트가 연결하는 원격 서버의 URL입니다.

라우트 유형

라우트가 능동 라우트인지 아니면 수동 라우트인지를 나타냅니다.

메트릭 값은 라우트가 "능동"인지 아니면 "수동"인지를 나타냅니다.

영역 이름

선택한 라우트가 속한 라우팅 영역의 이름입니다.

영역 유형

라우트의 라우팅 영역이 단일 흡(1hop) 영역 유형인지 아니면 다중 흡(mhop) 영역 유형인지를 나타냅니다.

영역 유형을 확인할 수 없는 경우 메트릭 값은 UNKNOWN입니다.

라우트를 모니터링하도록 설정한 경우 TIBCO EMS 라우트에 대해 다음과 같은 상태 메트릭을 사용할 수 있습니다.

백로그 수

라우트 백로그의 메시지 수입니다.

이 메트릭은 EMS 버전 5.0 이상에만 적용됩니다.

백로그 크기

라우트 백로그에 있는 모든 메시지의 크기(KB)입니다.

이 메트릭은 EMS 버전 5.0 이상에만 적용됩니다.

Inbound Byte Rate(인바운드 바이트 비율)

인바운드 메시지가 라우팅되는 비율을 초당 바이트 수(Bps)로 나타낸 값입니다.

인바운드 메시지 비율

초당 받은 인바운드 메시지의 수입니다.

연결됨

현재 라우트가 연결되어 있는지 여부를 나타내는 텍스트 문자열입니다.

메트릭 값이 "true"이면 연결된 라우트를 나타냅니다. 값이 "false"이면 연결이 끊긴 라우트를 나타냅니다.

Is Connected Value(연결됨 값)

라우트가 연결되어 있는지 여부를 나타내는 숫자 값입니다.

메트릭 값이 0 이면 연결된 라우트를 나타냅니다. 0 이 아닌 값은 연결이 끊긴 라우트를 나타냅니다.

Outbound Byte Rate(아웃바운드 바이트 비율)

아웃바운드 메시지가 라우팅되는 비율을 초당 바이트 수(Bps)로 나타낸 값입니다.

아웃바운드 메시지 비율

초당 보낸 아웃바운드 메시지의 수입니다.

중단된 대상

라우트에 중단된 대상이 있는지 여부를 나타냅니다.

메트릭 값이 "true"이면 중단된 대상이 있음을 나타냅니다. 값이 "false"이면 중단된 대상이 없음을 나타냅니다.

채널에 대한 메트릭

기본적으로 TIBCO EMS 멀티캐스트 채널은 모니터링되지 않지만 *TibcoEMSMonitor.properties* 파일에 *<ServerInstance>.monitor.channels* 속성을 추가하여 채널을 모니터링하도록 설정할 수 있습니다. 채널을 모니터링하도록 설정한 경우 TIBCO EMS 멀티캐스트 채널 작업에 대해 다음과 같은 구성 메트릭을 사용할 수 있습니다.

채널 인터페이스

멀티캐스트 데이터를 보내는 데 사용되는 멀티캐스트 가능 IP 주소입니다.

최대 시간

서버가 재전송을 위해 보낸 메시지를 유지하는 최대 기간(초)입니다.

최대 비율

멀티캐스트 데이터 전송의 최대 전송 비율을 초당 바이트 수(Bps)로 나타낸 값입니다.

멀티캐스트 그룹 주소

메시지가 전송되는 멀티캐스트 그룹 주소 및 포트입니다.

Multicast Time to Live(멀티캐스트 유지 시간)

서버와 멀티캐스트 데몬 사이에서 메시지에 허용되는 최대 네트워크 흡수입니다.

우선 순위

대역폭 할당 중에 이 채널에 지정된 우선 순위입니다.

가장 높은 우선 순위는 -5이고, 가장 낮은 우선 순위는 5입니다.

채널을 모니터링하도록 설정하면 TIBCO EMS 멀티캐스트 채널 작업에 대해 다음과 같은 상태 메트릭을 사용할 수 있습니다.

백로그 수

버퍼에서 채널을 통해 전송되도록 대기 중인 메시지 수입니다.

백로그 크기

버퍼에서 채널을 통해 전송되도록 대기 중인 메시지의 바이트 단위 크기입니다.

바이트 비율

채널에서 인바운드 및 아웃바운드 메시지가 처리되는 비율을 초당 바이트 수(Bps)로 나타낸 값입니다.

채널 상태

채널이 활성 상태인지, 아니면 서버 구성에 정의되어 있지만 활성 상태가 아닌지를 나타내는 텍스트 문자열입니다.

채널이 활성 상태이면 메트릭이 "활성"으로 표시되고, 채널이 서버에 정의되어 있지만 활성 상태가 아니면 메트릭이 "비활성"으로 표시됩니다.

채널 상태 값

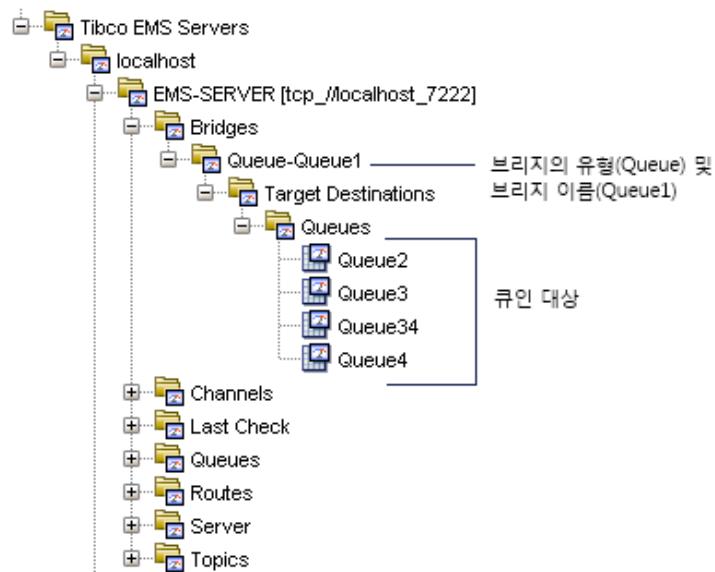
채널이 활성 상태인지(0), 아니면 서버 구성에 정의되어 있지만 활성 상태가 아닌지(0 이 아닌 값)를 나타내는 숫자 값입니다.

메시지 비율

채널에서 초당 처리되는 메시지의 수입니다. 이 메트릭에는 인바운드 및 아웃바운드 메시지가 모두 포함됩니다.

브리지에 대한 메트릭

기본적으로 TIBCO EMS 브리지는 모니터링되지 않지만 *TibcoEMSMonitor.properties* 파일에 *<ServerInstance>.monitor.bridges* 속성을 추가하여 브리지를 모니터링하도록 설정할 수 있습니다. 브리지를 모니터링하도록 설정한 경우 노드 이름은 브리지 유형 및 브리지 원본 이름을 나타내고 하위 노드는 대상 유형별로 대상을 나타냅니다. *Investigator* 트리에 보고되는 메트릭은 대상 및 메시지 필터링에 사용되는 메시지 선택기(선택기가 정의된 경우)뿐입니다. 예:



기본 TIBCO EMS 메트릭 그룹화 보기

SOA extension for TIBCO Enterprise Message Service에는 기본 대시보드와 경고를 정의하는 데 사용되는 기본 메트릭 그룹화가 포함되어 있습니다. 사용자 지정 대시보드와 경고에서도 이 기본 메트릭 그룹화를 사용할 수 있습니다.

기본 메트릭 그룹화는 Enterprise Manager extension for TIBCO Enterprise Message Service에 TIBCO Enterprise Message Service 관리 모듈(*TibcoEMSMgmtModule.jar*)의 일부로 패키지되어 있습니다.

TIBCO Enterprise Message Service 에이전트에 대한 기본 메트릭 그룹화를 보려면

1. Investigator에서 Workstation > 새 관리 모듈 편집기를 클릭합니다.
2. "*SuperDomain*" > "관리 모듈" > "Introscope SOA Extension for TibcoEMS <version>(*SuperDomain*)"을 확장합니다.
3. "메트릭 그룹화" 노드를 확장하여 TIBCO Enterprise Message Service 관리 모듈에 대해 정의된 모든 메트릭 그룹화를 표시합니다.
4. 특정 메트릭 그룹화를 클릭하여 "뷰어" 창에서 해당 정의를 봅니다.

모든 메트릭 그룹화의 기본 설정을 수정하거나 사용자 지정 메트릭 그룹화를 직접 생성할 수 있습니다.

참고: 메트릭 그룹화 생성 또는 수정에 대한 자세한 내용은 *CA APM Workstation* 사용자 안내서를 참조하십시오.

기본 TIBCO EMS 경고 보기

SOA extension for TIBCO Enterprise Message Service에는 사전 구성된 대시보드에 사용되는 기본 경고 정의가 포함됩니다. 이 기본 경고를 사용자 지정 대시보드에서도 사용할 수 있습니다. 대부분의 기본 경고는 기본 "주의" 및 "위험" 임계값으로 사전 구성되며 임계값이 초과되거나 심각도가 증가할 경우 콘솔에 알림을 보내도록 사전 구성됩니다.

기본 경고 정의는 Enterprise Manager extension for TIBCO Enterprise Message Service에 TIBCO Enterprise Message Service 관리 모듈(*TibcoEMSSManagementModule.jar*)의 일부로 패키지되어 있습니다.

TIBCO Enterprise Message Service 에이전트에 대한 기본 경고 정의를 보려면

1. Investigator에서 Workstation > 새 관리 모듈 편집기를 클릭합니다.
2. "*SuperDomain*" > "관리 모듈" > "Introscope SOA Extension for TibcoEMS <version>(*SuperDomain*)"을 확장합니다.

3. "경고" 노드를 확장하여 TIBCO Enterprise Message Service 관리 모듈에 대해 정의된 모든 메트릭 그룹화를 표시합니다.
4. 특정 경고를 클릭하여 "뷰어" 창에서 해당 정의를 봅니다.

특히 가장 중요하다고 생각하는 경고에 대한 "주의" 및 "위험" 임계값의 기본 설정을 검토하고 필요한 경우 값을 조정하십시오. 알림을 추가하거나 일부 경고에 대한 수정 조치를 지정할 수도 있습니다.

경고에 대한 기본 설정을 수정하거나 사용자 지정 경고를 직접 생성할 수 있습니다.

참고: 경고 생성 또는 수정에 대한 자세한 내용은 *CA APM Workstation 사용자 안내서*를 참조하십시오.

에이전트 구성 속성 요약

EMSMonitor 에이전트는 *TibcoEMSMonitor.properties* 파일에 있는 여러 구성 속성을 사용하여 연결할 EMS 서버 인스턴스와 모니터링할 구성 요소를 정의합니다. 다음 목록에서는 유효한 값을 설정하기 위해 빠르게 참조할 수 있도록 이러한 속성의 요약을 제공하고 각 속성에 허용되는 값의 유형을 설명합니다.

여러 구성 속성에서 패턴 일치를 위해 정규식을 지정할 수 있습니다. 정규식 구문 및 유효한 구조에 대한 요약을 보려면 Java 설명서에서 [패턴 일치](#) 관련 정보를 참조하십시오.

ems.server.list

모니터링하고자 하는 EMS 서버 인스턴스 이름의 쉼표로 구분된 목록입니다.

지정한 이름이 모니터링하려는 서버 인스턴스의 실제 이름과 일치할 필요는 없습니다. 예를 들어 이름이 같지만 포트가 다른 두 서버 인스턴스가 있는 경우 `ems.server.list` 속성에서 별칭을 사용하여 두 인스턴스를 구분할 수 있습니다.

예:

```
ems.server.list=mercury01,mercury02,jupiter03
```

<ServerInstance>.host

지정된 서버 인스턴스를 호스팅하는 EMS 서버의 호스트 이름 또는 IP 주소입니다. 기본값은 `localhost`입니다.

예:

```
mercury01.host=winsrvT400
```

<ServerInstance>.port

지정된 서버 인스턴스를 호스팅하는 EMS 서버의 포트 번호입니다. 기본값은 7222입니다.

예:

```
mercury01.port=7200
```

<ServerInstance>.username

관리자 권한이 있으며 EMS 서버에 연결하는 데 사용할 사용자 이름입니다. 이 속성은 대개 `emsPwdEncryptor` 유ти리티를 사용하여 계정의 암호를 암호화할 때 설정됩니다.

기본값은 `admin`입니다.

예:

```
mercury01.username=jgarcia
```

<ServerInstance>.password

<ServerInstance>.username 속성에 지정된 사용자 계정의 암호입니다. 이 속성은 대개 *emsPwdEncryptor* 유ти리티를 사용하여 계정의 암호를 암호화할 때 설정됩니다.

기본값은 빈 문자열(암호 없음)입니다.

예:

```
mercury01.password=YCLhqcwQfpc=
```

<ServerInstance>.delaytime

모니터링 중인 EMS 서버 구성 요소의 메트릭을 새로 고치기 위해 EMS 서버에 대한 쿼리를 수행하는 시간 간격(초)입니다.

이 간격마다 *EMSMonitor* 에이전트가 서버 인스턴스 및 해당 구성 요소의 상태 관련 메트릭을 수집합니다.

기본값은 60 초입니다.

예:

```
mercury01.delaytime=90
```

<ServerInstance>.report.static.freq

정적 메트릭에 대한 쿼리 사이에 수행할 EMS 서버 상태 쿼리의 수입니다. 예를 들어 이 속성이 20 으로 설정되어 있으면 *EMSMonitor* 에이전트가 메트릭을 20 번 수집할 때마다 정적 구성 관련 메트릭이 새로 고쳐집니다.

기본값은 20 입니다.

예:

```
mercury01.report.static.freq=50
```

<ServerInstance>.queue.filter.includeonly.regex

정규식을 지정하여 이 표현식과 이름이 일치하는 큐를 모니터링합니다. 유효한 모든 정규식을 사용할 수 있습니다. 일부 특수 문자는 이스케이프 처리해야 합니다.

이 속성을 사용하여 필터를 지정하지 않으면 기본적으로 *EMSMonitor* 에이전트는 모든 큐의 메트릭을 수집합니다.

예:

```
mercury01.queue.filter.includeonly.regex=[A-H]
```

<ServerInstance>.topic.filter.includeonly.regex

정규식을 지정하여 이 표현식과 이름이 일치하는 항목을 모니터링합니다. 유효한 모든 정규식을 사용할 수 있습니다. 일부 특수 문자는 이스케이프 처리해야 합니다.

이 속성을 사용하여 필터를 지정하지 않으면 기본적으로 *EMSMonitor* 에이전트는 모든 항목의 메트릭을 수집합니다.

예:

```
mercury01.topic.filter.includeonly.regex=[a-hA-H]
```

<ServerInstance>.queue.filter.include.dynamic

이 EMS 서버 인스턴스의 동적 큐를 모니터링 할지 여부를 지정합니다. 동적 큐의 메트릭을 포함하려면 이 속성을 *true*로 설정합니다.

기본적으로 *EMSMonitor* 에이전트는 정적 큐만 모니터링합니다.

예:

```
mercury01.queue.filter.include.dynamic=true
```

<ServerInstance>.topic.filter.include.dynamic

이 EMS 서버 인스턴스의 동적 항목을 모니터링 할지 여부를 지정합니다. 동적 항목의 메트릭을 포함하려면 이 속성을 *true*로 설정합니다.

기본적으로 *EMSMonitor* 에이전트는 정적 항목만 모니터링합니다.

예:

```
mercury01.topic.filter.include.dynamic=true
```

<ServerInstance>.monitor.bridges

이 EMS 서버 인스턴스의 브리지를 모니터링 할지 여부를 지정합니다. 브리지의 메트릭을 포함하려면 이 속성을 *true*로 설정합니다.

기본적으로 *EMSMonitor* 에이전트는 브리지를 모니터링하지 않습니다.

예:

```
mercury01.monitor.bridges=true
```

<ServerInstance>.monitor.channels

이 EMS 서버 인스턴스의 멀티캐스트 채널을 모니터링 할지 여부를 지정합니다. 채널의 메트릭을 포함하려면 이 속성을 *true*로 설정합니다.

기본적으로 *EMSMonitor* 에이전트는 채널을 모니터링하지 않습니다.

예:

```
mercury01.monitor.channels=true
```

<ServerInstance>.monitor.routes

이 EMS 서버 인스턴스의 라우트를 모니터링할지 여부를 지정합니다. 라우트의 메트릭을 포함하려면 이 속성을 *true*로 설정합니다.

기본적으로 *EMSMonitor* 에이전트는 라우트를 모니터링하지 않습니다.

예:

```
mercury01.monitor.routes=true
```

<ServerInstance>.bridge.filter.includeonly.regex

정규식을 지정하여 이 표현식과 이름이 일치하는 브리지를 모니터링합니다. 유효한 모든 정규식을 사용할 수 있습니다. 일부 특수 문자는 이스케이프 처리해야 합니다.

<ServerInstance>.monitor.bridges 속성이 *true*로 설정되어 있고 이 속성을 사용하여 필터를 지정하지 않은 경우 기본적으로 *EMSMonitor* 에이전트는 모든 브리지에 대한 메트릭을 수집합니다.

예:

```
mercury01.bridge.filter.includeonly.regex=test.*
```

<ServerInstance>.channel.filter.includeonly.regex

정규식을 지정하여 이 표현식과 이름이 일치하는 채널을 모니터링합니다. 유효한 모든 정규식을 사용할 수 있습니다. 일부 특수 문자는 이스케이프 처리해야 합니다.

<ServerInstance>.monitor.channels 속성이 *true*로 설정되어 있고 이 속성을 사용하여 필터를 지정하지 않은 경우 기본적으로 *EMSMonitor* 에이전트는 모든 채널에 대한 메트릭을 수집합니다.

예:

```
mercury01.channel.filter.includeonly.regex=test.*
```

<ServerInstance>.route.filter.includeonly.regex

정규식을 지정하여 이 표현식과 이름이 일치하는 라우트를 모니터링합니다. 유효한 모든 정규식을 사용할 수 있습니다. 일부 특수 문자는 이스케이프 처리해야 합니다.

<ServerInstance>.monitor.routes 속성이 *true*로 설정되어 있고 이 속성을 사용하여 필터를 지정하지 않은 경우 기본적으로 *EMSMonitor* 에이전트는 모든 라우트에 대한 메트릭을 수집합니다.

예:

```
mercury01.route.filter.includeonly.regex=test.*
```

<ServerInstance>.monitoring.level

EMS 서버 인스턴스에 대한 모니터링 수준을 정의합니다. 이 속성의 유효한 설정은 *minimum*, *recommended* 및 *full*입니다.

기본 서버 모니터링 수준은 *recommended*입니다.

예:

```
mercury01.monitoring.level=minimum
```

<ServerInstance>.queue.monitoring.level

EMS 서버 인스턴스의 큐에 대한 모니터링 수준을 정의합니다. 이 속성의 유효한 설정은 *minimum*, *recommended* 및 *full*입니다.

기본 큐 모니터링 수준은 *recommended*입니다.

예:

```
mercury01.queue.monitoring.level=recommended
```

<ServerInstance>.topic.monitoring.level

EMS 서버 인스턴스의 항목에 대한 모니터링 수준을 정의합니다. 이 속성의 유효한 설정은 *minimum*, *recommended* 및 *full*입니다.

기본 항목 모니터링 수준은 *recommended*입니다.

예:

```
mercury01.topic.monitoring.level=full
```

client.identity

EMS 서버가 *EMSMonitor*에 이전트의 ID를 검증하는 데 사용할 수 있는 인증서의 경로를 지정합니다. 대부분의 경우 이 속성은 *emsPwdEncryptor* 프로그램을 실행할 때 설정됩니다.

예:

```
client.identity=C:/TibcoEMSMonitor/certs/client.p12
```

ssl.password

클라이언트 보안 인증서의 암호화된 암호를 지정합니다.

대부분의 경우 이 속성은 *emsPwdEncryptor* 프로그램을 실행할 때 설정됩니다.

<ServerInstance>.ssl.connection

SSL(Secure Socket Layer) 프로토콜을 사용하여 서버 인스턴스에 연결할지 여부를 지정합니다.

SSL 을 사용하여 서버 인스턴스에 연결하려면 이 속성을 *enable* 로 설정합니다. 비보안 연결을 허용하려면 이 속성을 *disable* 로 설정합니다. 보안 관련 속성에는 기본값이 없습니다.

<ServerInstance>.verify.host

EMSMonitor 에이전트가 EMS 서버의 인증서를 확인할지 여부를 지정합니다. 에이전트가 *trusted.certificates* 속성에 정의된 목록을 사용하여 EMS 서버의 보안 인증서를 확인해야 한다면 이 속성을 *true* 로 설정합니다.

보안 관련 속성에는 기본값이 없습니다.

예를 들어 에이전트가 EMS 서버 인스턴스 *mercury01* 에 연결할 때 검증을 요구하도록 하려면 다음과 같이 합니다.

```
mercury01.verify.host=true
```

trusted.certificates

EMSMonitor 에이전트가 서버의 인증서를 검증하는 데 사용할 신뢰할 수 있는 인증서의 첨표로 구분된 목록을 지정합니다. *verify.host* 속성이 *true* 로 설정되어 있으면 이 속성이 필요하며, 이 속성은 SSL 을 사용하는 모든 EMS 서버 인스턴스에 적용할 수 있습니다.

보안 관련 속성에는 기본값이 없습니다.

<ServerInstance>.verify.hostname

에이전트가 서버 인증서의 CN(Common Name) 필드를 확인할지 여부를 지정합니다.

에이전트가 연결된 호스트의 이름이나 *<ServerInstance>.expected.name* 속성에 지정된 이름을 서버 인증서의 CN(Common Name) 필드와 비교하게 하려면 이 속성을 *true* 로 설정합니다. 이 속성이 *true* 로 설정되어 있고 이름이 일치하지 않으면 에이전트가 연결을 거부합니다.

보안 관련 속성에는 기본값이 없습니다.

<ServerInstance>.expected.hostname

EMSMonitor 에이전트가 서버 인증서의 CN(Common Name) 필드에서 찾아야 하는 이름을 지정합니다.

보안 관련 속성에는 기본값이 없습니다.

cipher.suites

EMSMonitor 에이전트가 SSL 기반 EMS 서버와의 통신을 암호화하는 데 사용할 수 있는 암호 그룹의 셈표로 구분된 목록을 지정합니다.

EMSMonitor 에이전트는 모니터링 중인 EMS 서버에서 지원하는 모든 암호화 패키지를 사용할 수 있습니다. 이 속성을 설정하면 모든 SSL 기반 EMS 서버 인스턴스에 설정이 적용됩니다.

보안 관련 속성에는 기본값이 없습니다.

제 10 장: webMethods Broker 모니터링

Software AG webMethods Broker 는 webMethods Integration Server 와 별개로 문서를 게시하기 위한 비동기적 처리 및 메시지 처리 서비스를 제공합니다. webMethods Broker 를 사용하면 문서가 목적지에 도착하는지 확인하는 보장된 배달을 사용하여 문서를 게시할 수 있습니다. 보장된 배달 문서가 목적지에 배달될 수 없는 경우 브로커는 해당 문서가 게시되지 않음을 확인합니다. SOA extension for webMethods Broker 를 사용하면 브로커 서버, 개별 브로커 인스턴스, 브로커 클라이언트 및 문서 처리 이벤트를 포함한 주요 webMethods Broker 구성 요소를 모니터링 할 수 있습니다.

이 단원에서는 webMethods Broker 환경의 성능, 가용성 및 전반적인 건전성을 모니터링하고 분석하는 데 사용할 수 있는 브로커 특정 대시보드, 메트릭 및 경고에 대해 설명합니다.

이 섹션은 다음 항목을 포함하고 있습니다.

[webMethods Broker 정보](#) (페이지 263)

[SOA Extension for webMethods Broker 설치 방법](#) (페이지 265)

[대시보드를 사용하여 webMethods Broker 모니터링](#) (페이지 275)

[브로커에 대한 메트릭 이해 및 보기](#) (페이지 278)

[기본 브로커 메트릭 그룹화 보기](#) (페이지 288)

[기본 브로커 경고 보기](#) (페이지 289)

webMethods Broker 정보

webMethods Broker 는 응용 프로그램 간의 문서 라우팅을 관리하고, 비즈니스 프로세스, 엔터프라이즈 시스템과 레거시 시스템, 데이터베이스와 기타 백엔드 시스템, 내부 및 외부 작업 흐름과 웹 서비스, 조직 내부 및 외부 간의 링크 역할을 합니다. 서비스 지향 아키텍처 내에서 webMethods Broker 는 다양한 플랫폼 및 프로토콜 간의 대용량 메시징을 지원하기 위하여 메시지 라우팅, 큐, 저장 및 필터링을 처리합니다.

webMethods Broker 는 클라이언트와 프로그램 사이의 문서 교환을 조정합니다. 또한 다음 서비스를 제공합니다.

- 브로커 클라이언트가 게시하는 문서에 대한 모든 문서 이벤트를 큐에 넣습니다.
- 이벤트를 구독하고 이벤트를 수신할 준비가 된 클라이언트에 문서를 보냅니다.
- 자신이 수신하는 문서의 정확한 배달을 확인합니다.
- 브로커 클라이언트가 수신하는 문서를 내용에 따라 선택적으로 필터링할 수 있도록 하는 필터링 서비스를 제공합니다.
- 문서 유형, 클라이언트 그룹, 게시 권한, 구독 권한, 네트워크 액세스 제어, 각 그룹의 클라이언트에 대한 큐 특성 및 생성된 각 브로커 클라이언트의 상태 정보에 대한 정보를 유지합니다.

메시지의 정확하고 안전한 배달은 비즈니스 서비스의 제공과 비즈니스 트랜잭션의 완료에 매우 중요하므로 **SOA extension for webMethods Broker** 는 webMethods Broker 의 작업 및 성능 메트릭을 모니터링하기 위한 **Standalone Agent** 를 제공합니다.

webMethods Broker 의 모니터링에 사용되는 **Standalone Agent** 는 핵심 Java 또는 .NET 에이전트에 대한 확장이 아닙니다. **WilyWMBrokerMonitor** 에이전트는 Java 클래스를 계측하는 대신 **WmBrokerClient API** 를 사용하여 webMethods Broker 구성 요소 및 작업에 대한 성능 정보를 수집한 다음 이 정보를 **Enterprise Manager** 에 보고합니다.

WilyWMBrokerMonitor 에이전트는 **Standalone Agent** 이기 때문에 별도의 소프트웨어 패키지로 배포되며 자체의 고유한 구성 단계가 필요합니다. 예를 들면 모니터링 할 브로커 서버에 대한 연결 정보를 구성하고 필터를 추가하여 메트릭을 수집할 구성 요소를 사용자 지정해야 합니다.

적절한 에이전트 속성을 구성한 후에는 **WilyWMBrokerMonitor** 에이전트를 사용하여 webMethods Broker 서버 및 클라이언트 작업의 모니터링을 시작할 수 있습니다.

SOA Extension for webMethods Broker 설치 방법

WilyWMBrokerMonitor 에이전트는 핵심 에이전트에 종속되지 않으므로 다른 모든 CA Introscope 구성 요소에 독립적으로 WilyWMBrokerMonitor 에이전트를 설치하고 구성할 수 있습니다.

SOA extension for webMethods Broker 를 추가하려면 다음과 같은 간략한 단계를 수행해야 합니다.

1. SOA extension for webMethods Broker 를 추가하기 위한 [사전 요구 사항을 현재 구현이 충족하는지 확인](#) (페이지 265)합니다.
2. [Standalone Agent 설치 관리자를 실행](#) (페이지 266)하거나 [응답 파일을 사용](#) (페이지 267)하여 필요한 에이전트 파일을 환경에 추가합니다.
3. [webMethods Broker 서버를 모니터링을 위해 준비](#) (페이지 268)합니다.
4. [WilyWMBrokerMonitor](#) (페이지 268) 에이전트 프로필을 구성하여 연결 및 모니터링 속성을 정의합니다.
5. [Enterprise Manager extension for webMethods Broker 를 사용하도록 설정](#) (페이지 275)합니다.

사전 요구 사항 확인

SOA Extension for webMethods Broker 를 추가하기 전에 환경이 특정 요구 사항을 충족하도록 구현되었는지 확인해야 합니다.

다음 단계를 수행하십시오.

1. webMethods Broker 가 설치되어 있는지 확인합니다.
참고: webMethods Broker 요구 사항에 대해서는 *Compatibility Guide*(호환성 안내서)를 참조하십시오.
2. 환경에 Enterprise Manager 와 Workstation 이 설치되어 있는지 확인합니다.
3. WilyWMBrokerMonitor 에이전트가 데이터를 전송할 Enterprise Manager 에 대한 연결 정보가 있는지 확인합니다.

Standalone Agent 설치 관리자 실행

Standalone Agent 설치 관리자를 사용하면 핵심 Java 또는 .NET 에이전트를 사용하지 않는 Standalone Agent를 설치할 수 있습니다. Standalone Agent 설치 관리자를 사용하여 webMethods Broker 모니터링을 사용하도록 설정할 수 있습니다. 설치 관리자는 사용자 환경에 에이전트 관련 파일을 추가하고 Enterprise Manager에 대한 에이전트 연결을 구성합니다. Standalone Agent 설치 관리자는 필요한 파일을 추출하여 적절한 위치에 넣고 사용자가 이를 수정하여 에이전트 구성은 완료할 수 있도록 합니다.

다음 단계를 수행하십시오.

1. 운영 환경에 적절한 Standalone Agent 설치 관리자를 시작합니다.
2. "소개" 페이지에서 "다음"을 클릭합니다.
3. 설치 할 모니터링 패키지를 선택하고 "다음"을 클릭합니다. 예를 들어 SOA Extension For webMethods Broker를 선택합니다.
이 옵션은 webMethods Broker 모니터링을 사용하도록 설정합니다.
4. 설치 디렉터리에 대해서는 "다음"을 클릭하여 기본 위치를 사용하거나 "찾아보기"를 클릭하여 다른 위치를 지정합니다.
5. Enterprise Manager 연결 설정에 대해서는 에이전트가 데이터를 전송할 Enterprise Manager 호스트 이름 및 포트 번호를 지정하고 "다음"을 클릭합니다.
6. 설정 요약을 검토하고 "설치"를 클릭합니다.
설치가 시작됩니다.
7. 설치가 완료되면 "완료"를 클릭합니다.

Standalone Agent 설치 관리자가 닫힙니다.

자동 설치를 위해 응답 파일 사용

Standalone Agent 설치 관리자를 대화식으로 실행하지 않으려면 독립 실행형 샘플 응답 파일을 편집하여 에이전트 파일을 설치하십시오. 이 방법을 사용하여 SOA extension for webMethods Broker 를 자동 모드에서 사용하도록 설정할 수 있습니다.

다음 단계를 수행하십시오.

1. Standalone Agent 설치 관리자와 같은 디렉터리에 있는 SampleResponseFile.StandaloneAgentPP.txt 파일을 엽니다.
2. SampleResponseFile.StandaloneAgentPP.txt 파일을 편집하여 shouldInstallWMBroker 속성을 true 로 설정하고 SOA Extension For webMethods Broker 를 에이전트에 추가합니다. 예: shouldInstallWMBroker=true
3. SampleResponseFile.StandaloneAgentPP.txt 파일을 저장합니다.
4. 명령줄에 적절한 명령을 입력하여 설치 관리자를 호출합니다.

참고: 자동 모드에서 에이전트를 설치하는 방법에 대한 자세한 내용은 **CA APM Java Agent** 구현 안내서 또는 **CA APM .NET** 에이전트 구현 안내서를 참조하십시오.

설치 아카이브 수동 추출

Standalone Agent 설치 관리자 또는 독립 실행형 응답 파일에 대한 액세스 권한이 없는 경우 운영 환경에 적합한 독립 실행형 설치 아카이브를 다운로드할 수 있습니다.

다음 단계를 수행하십시오.

1. [CA Support](#)의 CA APM 소프트웨어 다운로드 영역으로 이동합니다.
2. 운영 환경에 적합한 독립 실행형 설치 아카이브를 선택합니다.
3. 운영 환경에 적합한 명령을 사용하여 아카이브에서 수동으로 파일을 추출할 수 있습니다. 예를 들어 UNIX 컴퓨터에서는 다음과 같이 tar 명령을 사용합니다.

```
tar -xvf IntroscopeStandaloneAgentPPIInstaller<version>unix.tar
```

webMethods Broker 를 모니터링을 위해 준비

webMethods Broker 를 모니터링하려면 먼저 관리 권한을 가진 클라이언트 그룹을 식별하고 모니터링에 필요한 라이브러리를 로컬에서 사용할 수 있는지 확인해야 합니다.

다음 단계를 수행하십시오.

1. WmBrokerAgent 가 사용할 수 있는 브로커 클라이언트를 만드는데 사용할 수 있는 관리 클라이언트 그룹이 있는지 확인합니다.
2. 다음 브로커 라이브러리를 사용할 수 있는지 확인합니다.

Webmethods 7.x:

wmbrokerclient.jar
g11nutils.jar

Webmethods 8.x:

wm-brokerclient.jar
wm-g11nutils.jar

파일은 일반적으로 다음 위치에 있습니다.

- <Broker_Home>/common/lib -- webMethods Broker 6.5 에 해당
- <Broker_Home>/lib -- 지원되는 기타 webMethods Broker 버전에 해당

참고: webMethods Broker 요구 사항에 대해서는 *Compatibility Guide*(호환성 안내서)를 참조하십시오.

webMethods Broker 에 대한 에이전트 구성

설치 관리자를 실행하여 환경에 에이전트 파일을 추가하면 WilyWMBrokerMonitor 디렉터리가 생성되어 있습니다. 이 디렉터리의 파일을 사용하여 webMethods Broker 에 대한 에이전트를 구성합니다.

다음 단계를 수행하십시오.

1. 다음 브로커 라이브러리를 <Broker_Home>/lib 디렉터리 또는 <Broker_Home>/common/lib 에서 WilyWMBrokerMonitor/lib 디렉터리로 복사합니다.
 - wm-brokerclient.jar
 - wm-g11nutils.jar

- WilyWMBrokerMonitor 디렉터리 및 하위 디렉터리의 파일을 사용하여 WmBrokerAgent 연결 및 모니터링 속성을 구성합니다.

lib 디렉터리

필요한 라이브러리가 .jar 파일(jline-0.9.9.jar)로 패키지되어 포함되어 있습니다.

config 디렉터리

webMethods Broker 클라이언트 및 서버에 연결하기 위한 매개 변수를 구성할 때 사용하는 WilyWMBrokerMonitor.properties 파일이 있습니다.

Windows Service 디렉터리

wmBrokerAgent 프로세스를 Windows 서비스로 등록하고 등록 취소할 때 사용하는 파일이 있습니다.

Agent.jar 파일

메트릭을 수집하고 Enterprise Manager로 전달하기 위해 에이전트가 사용하는 클래스를 제공합니다.

IntroscopeAgent.profile 파일

Enterprise Manager 연결을 구성할 수 있는 속성을 제공합니다.

WmBrokerAgent.jar 파일

webMethods Broker 서버 및 클라이언트를 모니터링하기 위해 에이전트가 사용하는 클래스를 제공합니다.

wmBrokerPwdEncryptor 파일

암호 암호화 스크립트(wmBrokerPwdEncryptor.bat 또는 wmBrokerPwdEncryptor.sh)를 제공합니다. 이 스크립트를 사용하면 webMethods Broker 서버에 연결하기 위한 사용자 암호 및 SSL 클라이언트 인증서 암호를 암호화할 수 있습니다.

WmBrokerAgent 파일

webMethods Broker를 모니터링할 준비가 되었을 때 wmBrokerAgent 에이전트를 실행하기 위한 시작 스크립트(WmBrokerAgent.bat 또는 WmBrokerAgent.sh)를 제공합니다.

기본 연결 속성 구성

WmBrokerAgent 는 webMethods Broker 서버의 데이터를 Enterprise Manager 로 보고합니다. 메트릭의 수집 및 보고를 사용하도록 설정하는 방법은 다음과 같습니다.

1. WmBrokerAgent 가 보고하는 Enterprise Manager 의 연결 정보를 지정합니다.
2. 에이전트가 데이터를 수집하는 서버의 목록을 식별합니다.

다음 단계를 수행하십시오.

1. IntroscopeAgent.profile 파일을 텍스트 편집기에서 열고 Enterprise Manager 연결 매개 변수를 확인합니다. 예:

```
introscope.agent.enterprisemanager.transport.tcp.host.DEFAULT=mercury  
introscope.agent.enterprisemanager.transport.tcp.port.DEFAULT=5001
```

Standalone Agent 설치 관리자를 사용하여 에이전트 파일을 설치한 경우 설치할 때 입력한 값을 사용합니다. 에이전트 파일을 수동으로 추출한 경우에는 올바른 Enterprise Manager 에 연결하기 위해 기본 설정을 수정해야 할 수 있습니다.

2. WmBrokerAgent 시작 스크립트를 텍스트 편집기에서 엽니다.
3. JAVA_HOME 환경 변수를 적절한 디렉터리로 설정합니다. 예:

```
set JAVA_HOME=C:\webMethods7\jvm\win150\jre
```

JRE 버전 1.5 이상에 해당: webMethods Broker 를 모니터링하려면 JRE 버전 1.5 이상이 필요합니다.

4. WilyWMBrokerMonitor.properties 파일을 텍스트 편집기에서 열고 호스트 속성을 설정하여 모니터링 할 webMethods Broker 서버를 식별합니다. 예:

```
wily.webmethods.broker.server.host=vepsa09
```

5. 포트 속성을 설정하여 모니터링 할 webMethods Broker 서버에 연결하기 위해 사용할 포트 번호를 지정합니다. 예:

```
wily.webmethods.broker.server.port=6890
```

6. 클라이언트 그룹 속성을 설정하여 BrokerAdminClient 가 생성될 클라이언트 그룹을 지정합니다. 예:

```
wily.webmethods.broker.server.clientgroup=admin
```

메트릭 수집을 위한 폴링 간격 구성

WmBrokerAgent 는 각 브로커 서버를 정기적인 간격으로 폴링하여 최신 상태 및 구성 정보를 검색합니다. 구성 속성을 사용하면 이러한 쿼리의 빈도를 제어 할 수 있습니다.

WmBrokerAgent 에서 서버를 쿼리하는 빈도를 구성하려면 간격 속성을 설정하여 브로커 메트릭 수집에 대한 폴링 간격을 구성합니다. 예를 들어 14000 밀리초마다 한 번씩 메트릭을 수집하여 보고하려면 다음과 같이 합니다.

```
wily.webmethods.broker.interval=14000
```

모니터링할 브로커 클라이언트 그룹 구성

wily.webmethods.broker.clientstat.clientGroups 구성 속성을 사용하면 브로커에서 모니터링할 특정 클라이언트 그룹을 제어 할 수 있습니다.

모니터링할 브로커 클라이언트 그룹을 구성하려면 모니터링할 브로커 클라이언트 그룹의 목록을 지정합니다. 기본적으로 지정한 클라이언트 그룹에 있는 모든 클라이언트에 대한 메트릭이 보고됩니다. 예를 들어 IntegrationServer 및 IS-Backup 클라이언트 그룹의 모든 브로커 클라이언트에 대한 메트릭을 수집하여 보고하려면 속성을 다음과 같이 설정할 수 있습니다.

```
wily.webmethods.broker.clientstat.clientGroups=IntegrationServer,IS-Backup
```

webMethods Broker 를 위한 SSL 연결 구성

클라이언트 그룹과 webMethods Broker 서버 사이에 SSL(Secure Socket Layer) 연결을 사용하는 경우 에이전트는 유효성 검사를 위해 서명된 인증서를 제시해야 합니다. SSL 연결을 지원하려면 SSL 인증서에 대한 암호화된 암호를 생성하고 WilyWMBrokerMonitor.properties 파일에 몇 가지 추가적인 속성을 설정하십시오.

참고: 이 절차는 지원되는 버전의 webMethods Broker 서버(6.5 제외)에 적용해야 합니다. 지원되는 버전에 대한 자세한 내용은 *Compatibility Guide*(호환성 안내서)를 참조하십시오.

wmBrokerPwdEncryptor 스크립트를 사용하여 사용할 적절한 인증서 파일의 경로를 지정하고 SSL 연결에 대한 암호화된 암호를 생성할 수 있습니다.

다음 단계를 수행하십시오.

1. `wmBrokerPwdEncryptor` 스크립트를 텍스트 편집기에서 열고 `JAVA_HOME` 환경 변수를 적절한 디렉터리로 설정합니다. 예:
`set JAVA_HOME=C:\webMethods7\jvm\win150\jre`
 2. `wmBrokerPwdEncryptor` 스크립트를 실행하고 `y`를 입력하여 SSL 연결 속성을 구성합니다.
 - a. 키 저장소 파일의 경로를 묻는 메시지가 나타나면 Broker 의 PKCS12 키 저장소 인증서 파일의 경로를 입력합니다.
 - b. 신뢰 저장소 파일의 경로를 묻는 메시지가 나타나면 Broker 의 JKS(Java Key Store) 신뢰 저장소 인증서 파일의 경로를 입력합니다.
 - c. 키 저장소/인증서 파일의 암호를 묻는 메시지가 나타나면 파일 인증에 사용할 암호를 입력합니다.
 - d. Enter 키를 눌러 구성을 확인합니다.
- `wmBrokerPwdEncryptor` 스크립트가 다음 작업을 수행합니다.
- SSL 키 저장소 또는 인증서 암호를 암호화합니다.
 - `WilyWMBrokerMonitor.properties` 파일에서 Broker 에 대한 적절한 속성을 업데이트합니다.
3. `WilyWMBrokerMonitor.properties` 파일을 텍스트 편집기에서 엽니다.
 - a. `wily.webmethods.broker.connection.ssl_encrypted` 속성을 `true`로 설정하여 SSL 연결 암호화가 사용되도록 설정합니다. 예:
`wily.webmethods.broker.connection.ssl_encrypted=true`
 - b. `wily.webmethods.broker.ssl.distinguished_name` 속성을 Broker 서버에 연결할 때 사용할 전체 고유 이름으로 설정합니다. 예:
`wily.webmethods.broker.ssl.distinguished_name=cn=vespa09,dc=test,dc=org`
 - c. 파일을 저장하고 닫습니다.

추가 정보:

[webMethods Broker 6.5 를 위한 SSL 연결 구성](#) (페이지 273)

webMethods Broker 6.5 를 위한 SSL 연결 구성

클라이언트 그룹과 webMethods Broker 서버 사이에 SSL(Secure Socket Layer) 연결을 사용하는 경우 에이전트는 유효성 검사를 위해 서명된 인증서를 제시해야 합니다. SSL 연결을 지원하려면 SSL 인증서에 대한 암호화된 암호를 생성하고 WilyWMBrokerMonitor.properties 파일에 몇 가지 추가적인 속성을 설정하십시오.

참고: 설정하는 특정 속성은 사용 중인 webMethods Broker 의 버전에 따라 다릅니다. webMethods Broker 6.5 의 경우에는 다음 절차를 사용하십시오. 지원되는 webMethods Broker 버전에 대해서는 *Compatibility Guide*(호환성 안내서)를 참조하십시오.

wmBrokerPwdEncryptor 스크립트를 사용하여 사용할 적절한 인증서 파일의 경로를 지정하고 SSL 연결에 대한 암호화된 암호를 생성할 수 있습니다.

다음 단계를 수행하십시오.

1. **wmBrokerPwdEncryptor** 스크립트를 텍스트 편집기에서 열고 JAVA_HOME 환경 변수를 적절한 디렉터리로 설정합니다. 예:

```
set JAVA_HOME=C:\webMethods<version_number>\jvm\win150\jre
```
2. **wmBrokerPwdEncryptor** 스크립트를 실행하고 **y** 를 입력하여 SSL 연결 속성을 구성합니다.
 - a. 키 저장소 파일의 경로를 묻는 메시지가 나타나면 Enter 키를 눌러 이 속성을 비워 둡니다.
 - b. 신뢰 저장소 파일의 경로를 묻는 메시지가 나타나면 Enter 키를 눌러 이 속성을 비워 둡니다.
 - c. 인증서 파일의 경로를 묻는 메시지가 나타나면 브로커의 SSL 인증서 파일 경로를 입력합니다.

브로커의 인증서는 일반적으로 PKCS10 형식입니다. webMethods Broker Certificate Manager(awcert)를 사용하여 파일을 생성할 수 있습니다.

- d. 키 저장소/인증서 파일의 암호를 묻는 메시지가 나타나면 파일 인증에 사용할 암호를 입력합니다.

e. Enter 키를 눌러 구성은 확인합니다.

wmBrokerPwdEncryptor 스크립트가 다음 작업을 수행합니다.

- SSL 키 저장소 또는 인증서 암호를 암호화합니다.
- WilyWMBrokerMonitor.properties 파일에서 Broker에 대한 적절한 속성을 업데이트합니다.

3. WilyWMBrokerMonitor.properties 파일을 텍스트 편집기에서 엽니다.

- a. wily.webmethods.broker.connection.ssl_encrypted 속성을 true로 설정하여 SSL 연결 암호화가 사용되도록 설정합니다. 예:

```
wily.webmethods.broker.connection.ssl_encrypted=true
```

- b. wily.webmethods.broker.ssl.distinguished_name 속성을 Broker 서버에 연결할 때 사용할 전체 고유 이름으로 설정합니다. 예:

```
wily.webmethods.broker.ssl.distinguished_name=cn=vespa09,dc=test,dc=org
```

- c. 파일을 저장하고 닫습니다.

4. 다음 중 사용하고 있는 운영 환경에 해당하는 작업을 수행합니다.

- Windows에서는 awssl65jn.dll 파일을 <webMethods_Home>\common\lib 디렉터리에서 WilyWMBrokerMonitor\lib 디렉터리로 복사합니다.
- UNIX에서는 libawssl65jn.so 파일을 <webMethods_Home>/common/bin 디렉터리에서 WilyWMBrokerMonitor/lib 디렉터리로 복사합니다.

추가 정보:

[webMethods Broker를 위한 SSL 연결 구성 \(페이지 271\)](#)

에이전트 시작

에이전트의 연결 및 모니터링 속성을 구성하고 나면 에이전트를 시작할 수 있습니다. 에이전트 시작 스크립트를 사용하여 에이전트를 시작하고 브로커 서버 및 클라이언트 모니터링을 시작할 수 있습니다.

webMethods Broker 모니터링을 위한 에이전트를 시작하려면 WmBrokerAgent 시작 스크립트를 실행합니다.

webMethods Broker 서버 모니터링이 시작됩니다.

Enterprise Manager Extension for webMethods Broker 를 사용하도록 설정

CA APM for webMethods Broker 파일은 기본적으로 Enterprise Manager 를 설치할 때 `<EM_Home>/examples` 디렉터리에 설치됩니다. CA APM for webMethods Broker 를 사용하도록 설정하려면 예제 디렉터리의 파일을 Enterprise Manager 홈 디렉터리로 복사하거나 이동하십시오.

다음 단계를 수행하십시오.

1. SOAExtensionForWebMethodsBroker 디렉터리가 `<EM_Home>/examples` 디렉터리에 있는지 확인합니다.
2. `<EM_Home>/examples/SOAExtensionForWebMethodsBroker` 디렉터리의 파일을 Enterprise Manager 디렉터리 구조의 해당 위치로 복사합니다. 예를 들어 `<EM_Home>/examples/SOAExtensionForWebMethodsBroker/ext` 디렉터리의 파일을 `<EM_Home>/ext` 디렉터리로 복사합니다.
3. Enterprise Manager 가 클러스터 환경에서 수집기인 경우 webMethods Broker 관리 모듈 `WebMethodsBrokerManagementModule.jar` 를 `<EM_Home>/config/modules` 디렉터리에서 제거합니다.
참고: 관리 모듈만 MOM 컴퓨터로 사용 중인 Enterprise Manager 의 `<EM_Home>/config/modules` 디렉터리로 복사해야 합니다. 다른 모든 파일과 스크립트는 Collector Enterprise Manager 와 MOM Enterprise Manager 모두에 설치해야 합니다.
4. Workstation 을 다시 시작하여 SOA extension for webMethods Broker 에만 적용되는 대시보드 및 "개요" 탭을 로드합니다.

대시보드를 사용하여 webMethods Broker 모니터링

SOA extension for webMethods Broker 에는 응용 프로그램 환경의 전반적인 건전성을 모니터링하는 데 사용할 수 있는 몇 개의 사전 구성된 대시보드가 포함되어 있습니다. 대시보드는 배포된 에이전트에서 데이터를 집계하여 성능 정보를 요약하며 문제를 신속하게 진단하고 해결할 수 있도록 해 줍니다.

일반적으로 대시보드는 다음과 같은 작업을 가능하게 하므로 환경을 모니터링하는 시작점으로 사용됩니다.

- webMethods Broker의 주요 구성 요소에 대하여 전반적인 건전성, 성능, 가용성 및 현재 상태를 한눈에 모니터링
- 하위 수준 메트릭이 주의 또는 위험 임계값이 초과되었음을 알리는 경우 프로덕션 응용 프로그램 환경의 잠재적 문제를 조기에 식별
- 성능 정보를 드릴다운하여 배달 또는 게시가 지연되고 있거나 트래픽이 가장 많은 브로커, 클라이언트 그룹, 클라이언트 또는 문서 유형을 격리 및 식별

사전 구성된 webMethods Broker 대시보드는 Enterprise Manager extension for webMethods에 webMethods Broker 관리 모듈(*WebMethodsBrokerManagementModule.jar*)의 일부로 패키지되어 있습니다.

webMethods Broker 관리 모듈은 webMethods Broker에 대해 다음의 사전 구성된 대시보드를 제공합니다.

WebMethods Broker - 개요

webMethods Broker 서버의 주요 동작 및 저장소 통계에 대한 최상위 수준 개요로 다음을 포함합니다.

- 게시된 문서 및 큐에 저장된 문서에 대한 그래프와 경고 표시기
- 클라이언트 큐 길이와 검색된 문서에 대한 그래프
- 큐 길이, 큐 크기 및 승인되지 않은 문서의 수에 대한 경고 표시기

WebMethods Broker - 브로커

모든 브로커 인스턴스의 요약된 상태로, 다음과 같은 브로커의 목록을 포함합니다.

- 클라이언트가 가장 많은 브로커
- 문서 유형이 가장 많은 브로커
- 게시된 문서가 가장 많은 브로커
- 큐에 저장된 문서가 가장 많은 브로커
- "다시 시도" 큐에 있는 현재 문서가 가장 많은 브로커
- "다시 시도" 큐에 있는 배달 이벤트의 시도 횟수가 가장 많은 브로커
- "다시 시도" 큐에 있는 게시된 문서의 수가 가장 많은 브로커

WebMethods Broker - 클라이언트

모든 브로커 클라이언트 그룹 및 클라이언트의 요약된 상태로 다음 목록을 포함합니다.

- 배달된 문서가 가장 많은 클라이언트 그룹
- 게시된 문서가 가장 많은 클라이언트 그룹
- 배달된 문서가 가장 많은 클라이언트
- 게시된 문서가 가장 많은 클라이언트
- 큐에 저장된 문서가 가장 많은 클라이언트
- 검색된 이벤트가 가장 많은 클라이언트

이 대시보드에서 "클라이언트 세부 정보" 링크를 두 번 클릭하면 큐 크기, 큐 길이, 검색 수, 최대 큐 길이 및 큐에 저장된 문서가 가장 많은 클라이언트의 가장 많은 승인되지 않은 문서가 나열되는 **WebMethods Broker - 클라이언트 세부 정보** 대시보드가 표시됩니다.

WebMethods Broker - 문서

모든 브로커 문서 유형에 대한 요약된 상태로, 가장 자주 수신 및 전송된 문서 유형의 목록, 가장 많이 배달, 게시 및 수신된 문서 유형의 목록, 클라이언트 구독이 가장 많은 문서 유형의 목록을 포함합니다.

WebMethods Broker - 영역

모든 브로커 영역에 대한 요약된 상태로 다음 목록을 포함합니다.

- 큐에 있는 문서가 가장 많은 영역
- 큐에 있고, 전달되고, 수신된 문서가 가장 많은 영역
- 큐 크기가 가장 크고, 현재 큐 길이가 가장 길고, 최대 큐 길이가 가장 큰 영역

다음 단계를 수행하십시오.

1. Enterprise Manager 가 실행되고 있지 않으면 시작합니다.
2. Workstation 을 시작하고 SOA extension for webMethods 가 설치된 Enterprise Manager 에 로그인합니다.
3. "Workstation" > "새 콘솔"을 클릭합니다.
4. "대시보드" 드롭다운 목록에서 webMethods Broker 대시보드 중 하나를 선택합니다.

예를 들어 webMethods Broker 메트릭의 개요를 보려면 "WebMethods Broker - 개요" 대시보드를 선택합니다.

5. 관련 대시보드를 열어서 더 자세한 정보를 보려면 다른 탭이나 경고를 두 번 클릭합니다.

예를 들어 "클라이언트" 탭을 두 번 클릭하면 **webMethods Broker** 클라이언트 그룹 및 클라이언트에 대한 더 자세한 정보를 볼 수 있습니다.

6. 대시보드에서 특정 브로커, 클라이언트, 문서 유형 또는 영역 메트릭을 두 번 클릭하면 추가 분석을 위한 **Investigator**가 열립니다.

예를 들어 "WebMethods Broker - 클라이언트 세부 정보" 대시보드에서 큐에 저장된 수가 가장 많은 클라이언트를 두 번 클릭하면 해당 클라이언트의 "이벤트" > "큐에 저장됨" 메트릭이 선택된 **Investigator**가 열립니다.

브로커에 대한 메트릭 이해 및 보기

WmBrokerAgent는 브로커 서버 및 클라이언트를 모니터링하여 전반적인 건전성을 브로커 서버 아래의 `<port_number>`의 `<host_name>` 노드에 제공합니다.

다음 메트릭 범주의 메트릭으로 **webMethods Broker** 서버 및 클라이언트의 성능과 건전성을 모니터링할 수 있습니다.

브로커

"브로커" 노드 아래에는 클라이언트, 구독, 게시 및 배달된 문서의 수와 같은 개별 브로커 인스턴스에 대한 메트릭이 있습니다.

클라이언트 그룹

"클라이언트 그룹" 아래에서는 **WilyWMBrokerMonitor.properties** 파일에서 모니터링되도록 식별된 클라이언트 그룹과 게시 및 배달된 문서의 수와 같은 관련 클라이언트에 대한 메트릭을 볼 수 있습니다.

문서 유형

"문서 유형" 노드에서는 정의한 개별 문서 유형에 대한 메트릭을 볼 수 있습니다. 문서 유형에는 파트너 또는 프로그램 사이에 정보를 교환하는 데 사용되는 문서의 구조와 내용을 정의하는 필드 집합이 포함됩니다.

다시 시도 큐

"다시 시도 큐" 노드 아래에는 내부 다시 시도 큐에 대한 메트릭이 있습니다.

영역 통계

"영역 통계" 노드 아래에는 개별 영역에 대한 메트릭이 있습니다.

Trace

"추적" 노드 아래에는 내부 추적 큐에 대한 메트릭이 있습니다.

사용률

"사용률" 노드 아래에는 브로커 구성과 데이터 저장소 통계를 설명하는 메트릭이 있습니다.

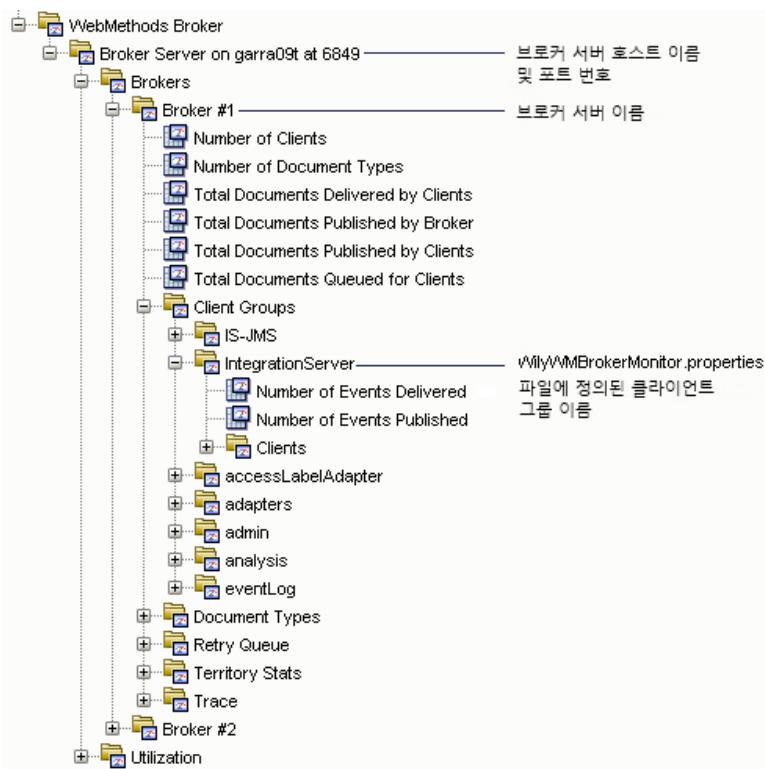
참고: 각 메트릭 범주에서 사용할 수 있는 메트릭에 대한 자세한 내용은 해당 범주에 대한 단원을 참조하십시오.

Investigator에서 webMethods Broker 메트릭 요약을 보고 탐색하려면

1. 에이전트 노드와 *<port_number>*의 *<host_name>*에 있는 브로커 서버 노드를 확장한 다음 "브로커"를 클릭하여 "개요" 탭을 표시합니다. 이 탭에는 모든 브로커 서버 인스턴스가 배달, 게시 및 큐에 저장된 문서의 수에 대한 요약 정보와 함께 나열됩니다.
2. 특정 브로커 이름을 선택하면 해당 인스턴스에 대해 배달 및 게시된 문서에 대한 요약 정보가 "개요" 탭에 그래픽 형식으로 표시됩니다. 예를 들어 개별 브로커 이름을 선택하면 "개요" 탭에 다음에 대한 그레프가 표시됩니다.
 - 클라이언트에 의해 배달된 총 문서
 - 클라이언트에 의해 게시된 총 문서
 - 브로커에 의해 게시된 총 문서
 - 클라이언트에 대해 큐에 저장된 총 문서
 - 클라이언트 수
 - 문서 유형 수
3. 개별 브로커 이름을 확장한 다음 하위 노드를 클릭하면 해당 메트릭 범주에 대한 요약 정보가 표시됩니다. 예를 들어 "클라이언트 그룹" 노드를 클릭하면 브로커에 대한 클라이언트 그룹의 목록과 각 클라이언트 그룹에 대해 배달 및 게시된 문서의 요약이 "개요" 탭에 표시됩니다.

Investigator에서 webMethods Broker 메트릭 노드를 보고 탐색하려면

1. 에이전트 노드와 <port_number>의 <host_name>에 있는 브로커 서버 노드를 확장한 다음 "브로커"를 클릭하여 모니터링 중인 브로커 서버 이름을 표시합니다.
2. 개별 *Broker_server_name* 노드를 확장하여 해당 브로커 서버 인스턴스에 대한 메트릭을 표시합니다.
3. "클라이언트 그룹" 노드를 확장하여 모니터링하려는 *WilyWMBrokerMonitor.properties* 파일에 지정된 클라이언트 그룹을 표시한 다음 특정 <client_group_name>을 확장하여 해당 클라이언트 그룹에 대한 메트릭을 표시합니다. 예:



4. "클라이언트" 노드를 확장한 다음 개별 <client_name> 하위 노드를 확장하여 해당 클라이언트에 대한 메트릭을 표시합니다.
5. "문서 유형" 노드를 확장한 다음 <document_type_name> 하위 노드를 확장하여 선택된 브로커에 대한 문서 유형의 메트릭을 표시합니다.

6. "다시 시도 큐", "영역 통계" 또는 "추적" 하위 노드를 확장하여 선택된 브로커에 대한 다시 시도 큐, 영역 또는 추적 메트릭을 표시합니다.
7. "사용률" > "저장소 통계"를 확장한 다음 "CONFIG" 또는 "DATA" 하위 노드를 확장하여 선택된 브로커 서버의 구성 또는 데이터 저장소에 대한 정보를 표시합니다.

브로커에 대한 메트릭

webMethods Broker 서버는 클라이언트, 브로커 및 다양한 응용 프로그램 간에 문서 흐름을 관리하는 호스트 컴퓨터입니다. 컴퓨터 프로그램은 문서의 형태로 정보를 게시하고 구독하며 브로커 서버는 문서의 라우트, 큐 및 필터 처리를 자동으로 수행합니다. 각 브로커 서버에는 하나 이상의 브로커가 상주합니다.

각 개별 브로커는 클라이언트 연결을 수신하고 자체 문서 유형, 클라이언트 큐 및 구독 관련 정보를 저장합니다. 클라이언트가 문서를 게시하면 브로커가 해당 유형의 문서를 구독하여 수신하는 브로커 클라이언트를 확인한 후 해당하는 브로커 클라이언트 큐에 문서를 배치합니다.

개별 브로커 인스턴스 이름의 "브로커" 하위 노드 아래에서 모니터링 중인 브로커 인스턴스에 대한 다음과 같은 메트릭을 사용할 수 있습니다.

클라이언트 수

선택한 브로커 서버의 총 클라이언트 수입니다.

문서 유형 수

선택한 브로커 서버에 정의된 총 문서 유형 수입니다.

클라이언트에 의해 배달된 총 문서

클라이언트에 의해 선택한 브로커 서버로 배달된 총 문서 수입니다.

브로커에 의해 게시된 총 문서

선택한 브로커 서버가 해당 클라이언트에 게시한 총 문서 수입니다.

클라이언트에 의해 게시된 총 문서

클라이언트가 선택한 브로커 서버에 게시한 총 문서 수입니다.

클라이언트에 대해 큐에 저장된 총 문서

선택한 브로커 서버가 해당 클라이언트로 배달하기 위해 큐에 저장한 총 문서 수입니다.

클라이언트 그룹에 대한 메트릭

webMethods Broker에서 클라이언트 그룹을 사용하여 여러 브로커 클라이언트에 대한 속성을 한 번에 설정할 수 있습니다. 개별 클라이언트 그룹 이름의 "클라이언트 그룹" 하위 노드 아래에서 모니터링 중인 클라이언트 그룹에 대한 다음과 같은 메트릭을 사용할 수 있습니다.

Number of Events Delivered(배달된 이벤트 수)

선택한 클라이언트 그룹의 모든 클라이언트에 대한 문서 배달 이벤트의 총 수입니다.

Number of Events Published(게시된 이벤트 수)

선택한 클라이언트 그룹의 모든 클라이언트에 대한 문서 게시 이벤트의 총 수입니다.

클라이언트에 대한 메트릭

브로커 클라이언트는 클라이언트 프로그램에서 특정 브로커에 연결하기 위해 생성하고 사용하는 webMethods 개체입니다. 클라이언트 프로그램에서는 필요한 경우 하나 이상의 브로커 클라이언트를 생성하여 하나 이상의 브로커에 연결합니다. webMethods Broker 클라이언트의 경우 문서 처리 작업이 이벤트로 기록됩니다. 예를 들어 문서를 클라이언트에 성공적으로 배달하면 해당 작업이 문서 배달 이벤트로 기록됩니다.

클라이언트 > <client_name> 노드 아래에서 다음과 같은 메트릭 범주의 메트릭을 사용하여 브로커 클라이언트를 모니터링할 수 있습니다.

이벤트

이벤트는 문서 처리 작업을 기록합니다. 예를 들어 문서를 클라이언트에 성공적으로 배달하면 해당 작업이 문서 배달 이벤트로 기록됩니다.

큐

큐는 비동기 이벤트를 나중에 처리할 수 있도록 받은 순서대로 저장하는 데 사용됩니다.

세션

세션은 클라이언트의 브로커 연결을 나타냅니다.

이벤트 메트릭

선택한 브로커 클라이언트의 개별 클라이언트에 대한 "이벤트" 하위 노드에서 다음과 같은 메트릭을 사용할 수 있습니다.

배달됨

선택한 클라이언트에 의해 기록된 문서 배달 이벤트의 총 수입니다.

게시됨

선택한 클라이언트에 의해 기록된 문서 게시 이벤트의 총 수입니다.

큐에 저장됨

선택한 클라이언트에 의해 큐에 저장된 문서의 총 수입니다.

검색됨

선택한 클라이언트에 의해 큐에서 검색된 문서의 총 수입니다.

Unacknowledged(승인되지 않음)

클라이언트 프로그램이 클라이언트 큐에서 검색했지만 아직 브로커에게 승인되지 않은 문서의 총 수입니다.

브로커는 승인을 받은 후 클라이언트 큐에서 문서를 제거합니다.

큐 메트릭

선택한 브로커 클라이언트의 개별 클라이언트에 대한 "큐" 하위 노드에서 다음과 같은 메트릭을 사용할 수 있습니다.

바이트 크기

큐 크기(바이트)입니다.

가장 높은 길이

기록된 큐 길이 중 가장 큰 값입니다.

길이

아직 클라이언트 프로그램에서 검색되지 않은 문서와 클라이언트 프로그램에서 검색되었지만 승인되지 않은 문서를 포함하여 현재 클라이언트 큐에 있는 총 문서 수입니다.

세션 메트릭

선택한 브로커 클라이언트의 개별 클라이언트에 대한 "세션" 하위 노드에서 다음과 같은 메트릭을 사용할 수 있습니다.

마지막 연결

마지막으로 설정된 클라이언트 세션의 세션 식별자입니다.

마지막 연결 끊기

브로커에서 연결을 끊은 마지막 클라이언트 세션의 세션 식별자입니다.

마지막 검색

클라이언트 큐에서 마지막으로 문서를 검색한 클라이언트 세션의 세션 식별자입니다.

문서 유형에 대한 메트릭

문서는 네트워크에서 브로커를 통해 게시자에게서 구독자로 이동하는 메시지입니다. 각 문서는 문서 유형의 인스턴스입니다. 각 문서 유형에는 고유한 이름과 속성이 연관되어 있습니다. 예를 들어 문서 폴더 이름, 문서가 만들어진 시간, 문서가 브로커 클라이언트에 의해 게시되고 검색된 횟수, 구독 수 등의 속성이 있습니다.

각 문서 유형의 "문서 유형" 하위 노드 아래에서 모니터링 중인 브로커 인스턴스에 대하여 다음과 같은 메트릭을 사용할 수 있습니다.

클라이언트 구독 수

선택한 브로커에서의 이 문서 유형에 대한 클라이언트의 총 구독 수입니다.

Number of Events Delivered(배달된 이벤트 수)

선택한 브로커에 의해 배달된 선택한 문서 유형의 총 문서 수입니다.

Number of Events Published(게시된 이벤트 수)

선택한 브로커에 게시된 선택한 문서 유형의 총 문서 수입니다.

Number of Forwards Received(받은 전달 수)

동일한 영역의 다른 브로커에서 전달된 선택한 문서 유형의 총 문서 수입니다.

Number of Groups Can Publish(게시할 수 있는 그룹 수)

선택한 문서 유형을 게시할 수 있는 클라이언트 그룹의 총 수입니다.

Number of Groups Can Subscribe(구독할 수 있는 그룹 수)

선택한 문서 유형을 구독할 수 있는 클라이언트 그룹의 총 수입니다.

다시 시도 큐에 대한 메트릭

webMethods Broker 다시 시도 큐는 브로커가 다시 시도해야 하는 요청을 저장하기 위해 사용하는 내부 큐에 대한 통계를 표시합니다. 이러한 통계는 대개 지원 담당자가 문제 해결을 위하여 사용합니다.

"다시 시도 큐" 하위 노드 아래에서 모니터링 중인 브로커 인스턴스에 대하여 다음과 같은 메트릭을 사용할 수 있습니다.

현재 이벤트 수

현재 "다시 시도 큐"에 있는 배달될 문서 수입니다.

현재 게시 수

현재 "다시 시도 큐"에 있는 게시될 문서 수입니다.

Maximum Events(최대 이벤트 수)

"다시 시도 큐"에 보관할 수 있는 배달될 최대 문서 수입니다.

Maximum Publishes(최대 이벤트 수)

"다시 시도 큐"에 보관할 수 있는 게시될 최대 문서 수입니다.

다음 작업 시퀀스 번호

검색할 다음 요청을 식별하는 시퀀스 번호입니다.

시도 횟수

브로커가 "다시 시도 큐"에 있는 문서를 배달하기 위해 시도한 횟수입니다.

예약된 보증된 이벤트 수

보장된 배달 표시가 되어 있어 "다시 시도 큐"에 공간이 예약되어 있는 문서의 수입니다.

예약된 총 게시 수

"다시 시도 큐"에 공간이 예약되어 있는 게시할 문서의 총 수입니다.

예약된 변동적 이벤트 수

"다시 시도 큐"에 공간이 예약되어 있는 배달할 변동적 문서의 수입니다.

예약된 변동적 게시 수

"다시 시도 큐"에 공간이 예약되어 있는 게시할 변동적 문서의 수입니다.

영역 통계에 대한 메트릭

webMethods Broker 영역 통계는 영역별로 정보 이벤트 및 큐를 제공합니다. 이러한 통계는 각 영역에 대해 큐에 저장, 전달 및 받은 이벤트 수와 큐 크기 및 길이를 설명합니다.

"영역 통계" > <territory_name> > "이벤트" 하위 노드 아래에서 모니터링 중인 브로커 인스턴스에 대하여 다음과 같은 메트릭을 사용할 수 있습니다.

Number Enqueued(큐에 저장된 수)

영역의 큐에 저장된 문서 배달 이벤트의 수입니다.

Number Forwarded(전달된 수)

영역에 전달된 문서 배달 이벤트의 수입니다.

Number Received(받은 수)

영역에서 받은 문서 배달 이벤트의 수입니다.

"영역 통계" > <territory_name> > "큐" 하위 노드 아래에서 모니터링 중인 브로커 인스턴스에 대하여 다음과 같은 메트릭을 사용할 수 있습니다.

바이트 크기

선택한 영역의 큐의 총 크기(바이트)입니다.

가장 높은 길이

영역에 기록된 큐 길이 중 가장 큰 값입니다.

길이

아직 클라이언트 프로그램에서 검색되지 않은 문서와 클라이언트 프로그램에서 검색되었지만 승인되지 않은 문서를 포함하여 영역의 현재 클라이언트 큐에 있는 총 문서 수입니다.

추적 큐에 대한 메트릭

webMethods Broker 추적 큐는 추적 이벤트가 저장되는 임시 저장소입니다. 추적 이벤트는 Integration Server에서 계시 또는 승인 트리거에 대한 응답으로 발생합니다. 내부 큐 동작 이벤트 및 추적 이벤트이므로 추적 큐에 대한 통계는 대개 지원 담당자가 문제 해결을 위하여 사용합니다. "추적" 하위 노드 아래에서 모니터링 중인 브로커 인스턴스에 대하여 다음과 같은 메트릭을 사용할 수 있습니다.

큐에 저장된 이벤트 수

선택한 브로커의 추적 큐에 있는 문서의 총 수입니다.

큐 바이트 크기

선택한 브로커의 추적 큐의 총 크기(바이트)입니다.

큐 길이

선택한 브로커의 추적 큐에 있는 총 항목 수입니다.

사용률에 대한 메트릭

"사용률" > "저장소 통계" 하위 노드 아래에서 브로커 서버의 구성(CONFIG) 및 런타임(DATA) 데이터 저장소에 대하여 다음과 같은 메트릭을 사용할 수 있습니다.

현재 예약된 KB

선택한 저장소 파일용으로 예약된 크기(KB)입니다.

현재 사용 중인 KB

선택한 저장소 파일의 현재 크기(KB)입니다.

Maximum KB Available(사용 가능한 최대 KB)

선택한 저장소 파일의 크기가 증가할 수 있는 최대 크기(KB)입니다.

Maximum Transaction (KB)(최대 트랜잭션(KB))

브로커 서버의 최대 트랜잭션 크기(KB)입니다.

Session URL(세션 URL)

저장소 기능 구성의 위치 및 유형입니다.

"사용률" > "저장소 통계" 아래에 있는 메트릭은 CONFIG 및 DATA 저장소로 나뉩니다. CONFIG 메트릭은 브로커 서버, 영역, 브로커, 문서 유형 및 통계를 정의하는 데 필요한 데이터 저장소를 설명합니다. DATA 메트릭은 클라이언트 큐, 문서 및 로그에 필요한 데이터 저장소를 설명합니다. 또한 "파일 통계" > <file_name> 하위 노드 아래에서 브로커 서버의 개별 구성 및 데이터 저장소 파일 이름으로 유사한 메트릭을 사용할 수 있습니다.

기본 브로커 메트릭 그룹화 보기

SOA extension for webMethods 에는 기본 대시보드와 경고를 정의하는 데 사용되는 기본 메트릭 그룹화가 포함되어 있습니다. 사용자 지정 대시보드와 경고에서도 이 기본 메트릭 그룹화를 사용할 수 있습니다.

기본 메트릭 그룹화는 Enterprise Manager extension for webMethods Broker 에 webMethods Broker 관리 모듈(*WebMethodsBrokerManagementModule.jar*)의 일부로 패키지되어 있습니다.

CA Introscope 워크스테이션 관리 모듈 편집기를 사용하여 기본 메트릭 그룹화를 볼 수 있습니다. 또한 사용자 지정 메트릭 그룹화가 포함되도록 webMethods Broker 관리 모듈을 확장하거나 사용자 지정 대시보드 또는 경고에서 기본 메트릭 그룹화를 사용할 수도 있습니다.

webMethods Broker 에이전트에 대한 기본 메트릭 그룹화를 보려면

1. Investigator에서 Workstation > 새 관리 모듈 편집기를 클릭합니다.
2. "*SuperDomain*" > "관리 모듈" > "WebMethodsBrokerManagementModule(*SuperDomain*)"을 확장합니다.
3. "메트릭 그룹화" 노드를 확장하여 webMethods 관리 모듈에 대해 정의된 모든 메트릭 그룹화를 표시합니다.
4. 특정 메트릭 그룹화를 클릭하여 "뷰어" 창에서 해당 정의를 봅니다.

기본 브로커 경고 보기

SOA extension for webMethods 에는 사전 구성된 대시보드에 사용되는 기본 경고 정의가 포함됩니다. 이 기본 경고를 사용자 지정 대시보드에서도 사용할 수 있습니다. 대부분의 기본 경고는 기본 "주의" 및 "위험" 임계값으로 사전 구성되며 임계값이 초과되거나 심각도가 증가할 경우 콘솔에 알림을 보내도록 사전 구성됩니다.

기본 경고 정의는 Enterprise Manager extension for webMethods Broker 에 `webMethods Broker 관리 모듈`(*WebMethodsBrokerManagementModule.jar*)의 일부로 패키지되어 있습니다.

CA Introscope?Workstation 관리 모듈 편집기를 사용하여 기본 경고 정의를 볼 수 있습니다. 또한 `webMethods Broker` 관리 모듈을 확장하여 사용자 지정 경고 정의와 알림 유형을 포함하거나 사용자 지정 대시보드에서 기본 경고 정의를 사용할 수도 있습니다.

webMethods Broker 에 이전트에 대한 기본 경고 정의를 보려면

1. `Investigator`에서 `Workstation > 새 관리 모듈 편집기`를 클릭합니다.
2. `**SuperDomain** > "관리 모듈" > "WebMethodsBrokerManagementModule(*SuperDomain*)"`을 확장합니다.
3. "메트릭 그룹화" 노드를 확장하여 `webMethods Broker` 관리 모듈에 대해 정의된 모든 메트릭 그룹화를 표시합니다.
4. 특정 경고를 클릭하여 "뷰어" 창에서 해당 정의를 봅니다.

특히 "주의" 및 "위험" 임계값의 기본 설정을 검토하여 필요한 경우 값을 조정하고 알림 또는 수정 조치를 추가하십시오.

제 11 장: webMethods Integration Server 모니터링

Software AG webMethods 제품군은 조직에서 비즈니스 프로세스 및 웹 서비스의 생성, 조정 및 통합을 가능하게 만드는 다중 인프라 구성 요소와 기능으로 구성된 SOA 플랫폼입니다. SOA Extension for webMethods 를 사용하면 webMethods 인프라의 많은 주요 요소를 모니터링 할 수 있습니다.

이 단원에서는 webMethods Integration Server 의 상태 및 작업을 모니터링하고 분석하는 데 사용할 수 있는 webMethods 서비스와 관련 대시보드 및 메트릭에 대해 설명합니다.

이 섹션은 다음 항목을 포함하고 있습니다.

[webMethods Integration Server 정보](#) (페이지 291)

[webMethods Integration Server 모니터링을 사용하도록 설정하는 방법](#) (페이지 295)

[대시보드를 사용하여 webMethods Broker 모니터링](#) (페이지 299)

[모니터링 및 표시되는 서비스 필터링](#) (페이지 304)

[webMethods 에 대한 메트릭 보기 및 탐색](#) (페이지 306)

[기본 webMethods 메트릭 그룹화 보기](#) (페이지 323)

[기본 webMethods 경고 보기](#) (페이지 324)

[webMethods 종속성 보기](#) (페이지 324)

[webMethods 에 대한 트랜잭션 추적](#) (페이지 325)

webMethods Integration Server 정보

기업은 webMethods Integration Server 를 사용하여 기존 비즈니스 서비스와 새 비즈니스 서비스를 외부에 공개하고 통합할 수 있습니다. 이 제품에는 새로운 서비스를 설계, 테스트 및 배포하고, 느슨하게 결합된 서비스와 레거시 시스템을 자동화, 조정 및 결합하여 비즈니스 프로세스를 향상시킬 수 있는 도구들이 포함됩니다.

webMethods Integration Server 는 서비스를 실행하고 배포할 수 있는 중앙 집중식 플랫폼을 제공합니다. 이 플랫폼은 클라이언트 요청을 수신 및 해석하고, 요청된 서비스를 식별 및 호출하고, 데이터를 실행 중인 서비스에 예상된 형식으로 전달하고, 서비스에서 생성한 출력을 수신하여 클라이언트로 반환합니다.

통합 플랫폼인 **webMethods**는 주로 응용 프로그램 서버, 데이터베이스 및 사용자 지정 응용 프로그램 간의 작업을 조정하고 회사나 거래 파트너 간에 전자 문서를 교환하는 데 사용됩니다.

다음과 같은 최상위 수준 구성 요소의 메트릭을 사용하여 **webMethods Integration Server**의 작업을 모니터링할 수 있습니다.

어댑터

어댑터를 사용하면 서비스 인터페이스의 공용 어댑터 프레임워크를 통해 외부 응용 프로그램을 **webMethods**에 통합할 수 있습니다. 어댑터의 구성 요소는 다음과 같습니다.

어댑터 연결 - 이를 통해 **Integration Server**가 런타임에 외부 리소스 또는 시스템에 연결할 수 있습니다.

어댑터 서비스 - **Integration Server**에서 실행되는 서비스로, 외부 리소스에 대한 작업을 시작합니다.

어댑터 알림 - 외부 리소스를 모니터링하여 **Integration Server**에 의해 시작되지 않은 이벤트가 발생한 경우 **Integration Server**에 알립니다.

SOA Extension for webMethods에서는 "어댑터 연결 풀", "어댑터 서비스" 및 "어댑터 알림" 노드의 메트릭을 사용하여 **webMethods Integration Server** 용으로 배포한 모든 어댑터의 성능 및 전반적인 건전성을 모니터링할 수 있습니다.

비즈니스 프로세스

비즈니스 프로세스는 특정 비즈니스 규칙 집합을 사용하여 특정 순서로 수행되는 일련의 상호 연관된 비즈니스 작업입니다. 대부분의 비즈니스 프로세스에서는 서로 다른 역할의 여러 인력과 여러 시스템의 상호 작용을 필요로 합니다.

예를 들어 새 직원을 교육하거나, 구입 주문을 처리하거나, 송장을 전송하는 비즈니스 프로세스가 있습니다. 이러한 각 비즈니스 프로세스에는 새 직원을 위한 업무 공간을 할당하고, 인적자원관리(HR) 시스템에 직원을 추가하고, 사무 용품을 주문하는 등의 비즈니스 작업이 포함될 수 있습니다.

SOA Extension for webMethods에서는 "WebMethods" > "비즈니스 프로세스" 노드 아래에 있는 메트릭을 사용하여 정의한 비즈니스 프로세스의 성능 및 전반적인 건전성을 모니터링할 수 있습니다.

흐름 서비스

흐름 서비스는 **webMethods Flow** 언어로 작성되고 **webMethods Integration Server**에 배포됩니다. 흐름 서비스에서는 **webMethods** 서버에서 실행되는 모든 서비스를 호출할 수 있습니다. 여기에는 다른 흐름 서비스, 사용자 정의 서비스, 기본 제공 서비스 및 **webMethods** 어댑터 또는 .NET 플러그인 같은 다른 공급자의 서비스도 포함됩니다.

SOA Extension for webMethods에서는 모든 흐름 서비스의 성능 및 전반적인 건전성을 모니터링하거나, 흐름 서비스를 필터링하여 관심이 없는 흐름 서비스를 모니터링에서 제외할 수 있습니다. 흐름 서비스에서 개별 흐름 단계의 메트릭은 "WebMethods" > "흐름 서비스" 노드 아래에 나열됩니다.

Java 서비스

Java 서비스는 Java로 작성되거나 다른 언어로 작성되어 Java 클래스를 통해 래핑된 기본 제공 서비스 및 사용자 정의 서비스이며 **webMethods Integration Server**에서 서비스로 노출됩니다.

SOA Extension for webMethods에서는 모든 Java 서비스의 성능 및 전반적인 건전성을 모니터링하거나, Java 서비스를 필터링하여 관심이 없는 Java 서비스를 모니터링에서 제외할 수 있습니다. 각 Java 클래스의 Java 메서드에 대한 메트릭은 "WebMethods" > "Java 서비스" 노드 아래에 나열됩니다.

JDBC 연결 풀

webMethods Integration Server는 Java 데이터베이스 연결을 사용하여 네트워크에서 정보를 교환하고 전달합니다.

SOA Extension for webMethods에서는 "WebMethods" > "JDBC 연결 풀" 노드 아래의 메트릭을 사용하여 JDBC 연결의 가용성을 모니터링할 수 있습니다.

스레드 풀

webMethods Integration Server는 서비스 실행, **webMethods Broker**의 문서 검색 및 트리거 실행에 스레드를 사용합니다.

SOA Extension for webMethods에서는 "WebMethods" > "스레드 풀" 노드 아래의 메트릭을 사용하여 스레드의 가용성을 모니터링할 수 있습니다.

트레이딩 네트워크

트레이딩 네트워크를 사용하면 문서를 교환하는 여러 조직이 B2B 관계를 설정하고 다양하게 이용할 수 있습니다.

SOA Extension for webMethods 에서는 "WebMethods" > "트레이딩 네트워크" 노드 아래의 메트릭을 사용하여 문서 인식 및 처리를 모니터링할 수 있습니다.

트리거

트리거는 계시 가능한 문서 유형에 대한 구독을 설정하고 해당 문서의 인스턴스를 처리하는 방법을 지정합니다.

브로커 또는 로컬 트리거는 Integration Server 에 로컬로 계시되거나 브로커로 배달된 문서를 구독하고 이를 처리하는 트리거입니다. 브로커 트리거는 비동기 어댑터 알림과 연결되는 경우가 많습니다.

JMS 트리거는 JMS 공급자의 대상(큐 또는 항목)에서 메시지를 수신한 다음 이 메시지를 처리하는 트리거입니다.

SOA Extension for webMethods 에서는 "WebMethods" > "트리거" 노드 아래의 메트릭을 사용하여 트리거를 모니터링할 수 있습니다.

WebServices

WebServices 메트릭은 각 서비스 내에서 클라이언트 및 서버의 비즈니스 서비스 끝점과 관련 작업을 나타냅니다.

SOA Extension for webMethods 에서는 "WebMethods" > "WebServices" 노드 아래에서 클라이언트 및 서버 웹 서비스 끝점의 성능 및 전반적인 건전성을 모니터링할 수 있습니다.

XSLT 서비스

webMethods 에서는 XSLT 스타일시트를 사용하여 XML 데이터를 다른 형식으로 변환하고 다른 서비스의 변환을 포함할 수 있습니다.

SOA Extension for webMethods 에서는 "WebMethods" > "XSLT 서비스" 노드 아래에서 XSL 변환의 성능 및 전반적인 건전성을 모니터링할 수 있습니다.

webMethods Integration Server 모니터링을 사용하도록 설정하는 방법

관리자는 다음의 간략한 단계를 수행하여 webMethods Integration Server 모니터링을 사용하도록 설정할 수 있습니다.

1. webMethods Integration Server 가 설치되어 있는지 확인합니다.
참고: webMethods Integration Server 요구 사항에 대해서는 *Compatibility Guide*(호환성 안내서)를 참조하십시오.
2. 지원되는 버전의 webMethods WmPRT.jar 파일이 있는지 확인합니다. 웹 브라우저에서 다음 URL 을 엽니다. 나열되는 버전 수준을 확인하십시오.
http://<Integration_Server_Hostname>:<port_number>/WmRoot/Updates.dsp
3. 에이전트 및 CA APM for SOA 가 설치되고 사용되도록 설정되었는지 확인합니다.
4. 에이전트 프로필을 구성하여 에이전트가 CA APM for webMethods Integration Server 를 사용할 수 있도록 설정합니다.
중요! Standalone Agent 설치 관리자를 사용하거나 응답 파일을 사용하여 에이전트에서 CA APM for webMethods Integration Server 를 사용하도록 설정한 경우에는 이 단계를 건너뛰십시오.
5. [Enterprise Manager 확장을 사용하도록 설정합니다](#) (페이지 298).

추가 정보:

[webMethods Integration Server 모니터링을 사용하도록 수동으로 에이전트 설정](#) (페이지 295)

webMethods Integration Server 모니터링을 사용하도록 수동으로 에이전트 설정

다음 방법 중 하나를 사용하여 webMethods Integration Server 에 대한 모니터링을 사용하도록 설정할 수 있습니다.

- 에이전트를 설치할 때 "CA APM for webMethods Integration Server"를 선택합니다.
기본 설정을 사용하여 에이전트 프로필이 자동으로 구성됩니다.
추가적인 단계가 필요 없습니다.

- 에이전트를 설치할 때 "CA APM for webMethods Integration Server"를 선택하지 않습니다. 다음 절차를 따라 에이전트 프로필을 수동으로 구성합니다.

참고: webMethods Integration Server 요구 사항에 대해서는 *Compatibility Guide*(호환성 안내서)를 참조하십시오.

다음 단계를 수행하십시오.

1. 기본 에이전트 및 CA APM for SOA 가 설치되고 사용하도록 설정되었는지 확인합니다.
2. CA APM for webMethods Integration Server 디렉터리가 `<Agent_Home>/examples` 디렉터리에 있는지 확인합니다.
3. `<Agent_Home>/examples/SOAExtensionForWebMethodsIS` 디렉터리의 파일을 해당 `<Agent_Home>` 디렉터리로 복사합니다.
예를 들어
`<Agent_Home>/examples/SOAExtensionForWebMethodsIS/ext` 의 파일을 `<Agent_Home>/core/ext` 디렉터리로 복사합니다.
4. `<Agent_Home>/core/config/IntroscopeAgent.profile` 파일을 텍스트 편집기에서 엽니다.
 - a. `webmethods.pbl`, 기본 에이전트 `pbl`, `spm.pbl` 을 `IntroscopeAgent.profile` 파일의 `introscope.autoprobe.directivesFile` 속성에 추가합니다. 예:

```
introscope.autoprobe.directivesFile=default-typical.pbl,hotdeploy,spm.pbl,webmethods.pbl
```

참고: webMethods Integration Server 용 ProbeBuilder [지시문 파일](#) (페이지 297)을 사용하여 추적을 사용자 지정할 수 있습니다.

- b. `introscope.agent.agentName` 속성 값을 webMethods Agent 로 설정합니다. 예:

`introscope.agent.agentName=webMethods Agent`

이 단계는 대시보드에 webMethods 데이터만 표시되도록 합니다.

- c. 기본 설정을 변경하려면 `IntroscopeAgent.profile` 파일에서 추가 속성을 수정하십시오.

- d. IntroscopeAgent.profile 파일을 저장하고 닫습니다.
5. (webMethods Integration Server 7.x 에만 해당)
<Agent_Home>/core/config/webmethods-toggles.pbd 파일에서 다음과 같이 편집합니다.
 - a. 비즈니스 프로세스 섹션에서 BProcAndStepTracing71 플래그의 주석 처리를 해제합니다.
BProcAndStepTracing71 플래그를 꼽습니다.
BProcAndStepTracing80 플래그를 꼽습니다.
BProcAndStepTracing82 플래그를 꼽습니다.
 - b. 웹 서비스 섹션에서 다음 옵션의 주석 처리를 해제합니다.
WM8xWebServicesTracing 플래그를 꼽습니다.
WM7xWebServicesTracing 플래그를 꼽습니다.
WebMethods8xWSClientTracing 플래그를 꼽습니다.
WebMethods7xWSClientTracing 플래그를 꼽습니다.
WebMethods65WSClientTracing 플래그를 꼽습니다.
 - c. webmethods-toggles.pbd 파일을 저장하고 닫습니다.
6. webMethods Integration Server 프로세스를 다시 시작합니다.

webMethods Integration Server에 대한 지시문 파일 정보

IntroscopeAgent.profile에서 introscope.autoprobe.directivesFile 속성을 설정하면 webMethods Integration Server에 대한 기본 계측이 활성화됩니다. 그런 후 기본 모니터링을 수정하려는 경우 webmethods.pbd 또는 webmethods-toggles.pbd에서 ProbeBuilder 지시문을 사용자 지정할 수 있습니다. 예를 들어 특정 추적 프로그램 그룹에 대한 추적을 설정하거나 해제하여 특정 구성 요소에 대한 모니터링을 세부 조정할 수 있습니다.

webmethods.pbd

흐름 서비스, Java 서비스, 트레이딩 네트워크, 웹 서비스, XSLT 서비스 및 비즈니스 프로세스를 포함한 webMethods Integration Server의 주요 서비스를 모니터링합니다.

webmethods-toggles.pbd

webMethods Integration Server 구성 요소의 모니터링을 설정하거나 해제합니다.

webmethods.pbl

webMethods Integration Server 모니터링을 위해 다음과 같은 ProbeBuilder 지시문 파일의 기본 목록을 제공합니다.

- webmethods.pbd
- webmethods-toggles.pbd

Enterprise Manager 확장을 사용하도록 설정

CA APM for webMethods Integration Server 파일은 기본적으로 Enterprise Manager 를 설치할 때 *<EM_Home>/examples/SOAEExtensionForWebMethodsIS* 디렉터리에 설치됩니다. CA APM for webMethods Integration Server 를 사용하도록 설정하려면 webMethods Integration Server 에 대한 Enterprise Manager 파일을 *<EM_Home>/examples* 디렉터리에서 Enterprise Manager 홈 디렉터리의 적절한 위치로 복사하거나 이동해야 합니다.

참고: CA APM for webMethods Integration Server 를 사용하려면 CA APM for SOA 가 [Enterprise Manager에서 사용하도록 설정](#) (페이지 41)되어 있어야 합니다.

다음 단계를 수행하십시오.

1. CA APM for webMethods Integration Server 디렉터리 *SOAEExtensionForWebMethodsIS* 가 *<EM_Home>/examples* 디렉터리에 있는지 확인한 다음 *<EM_Home>/examples/SOAEExtensionForWebMethodsIS* 디렉터리의 파일을 Enterprise Manager 디렉터리 구조의 해당 위치로 복사합니다. 예를 들어 *<EM_Home>/examples/SOAEExtensionForWebMethodsIS/ext* 디렉터리의 파일을 *<EM_Home>/ext* 디렉터리로 복사합니다.
2. Enterprise Manager 가 클러스터 환경의 수집기인 경우 webMethods Integration Server 관리 모듈 *WebMethodsISManagementModule.jar* 을 *<EM_Home>/config/modules* 디렉터리에서 제거합니다. 관리 모듈만 MOM 컴퓨터로 사용 중인 Enterprise Manager 의 *<EM_Home>/config/modules* 디렉터리로 복사해야 합니다. 다른 모든 파일과 스크립트는 Collector Enterprise Manager 와 MOM Enterprise Manager 모두에 설치해야 합니다.
3. Workstation 을 다시 시작하여 CA APM for webMethods Integration Server 의 고유한 대시보드 및 "개요" 탭을 로드합니다.

대시보드를 사용하여 webMethods Broker 모니터링

SOA extension for webMethods Integration Server에는 응용 프로그램 환경의 전반적인 건전성을 모니터링하는 데 사용할 수 있는 몇 개의 사전 구성된 대시보드가 포함되어 있습니다. 대시보드는 배포된 에이전트에서 데이터를 집계하여 성능 정보를 요약하며 문제를 신속하게 진단하고 해결할 수 있도록 해 줍니다.

일반적으로 대시보드는 다음과 같은 작업을 가능하게 하므로 환경을 모니터링하는 시작점으로 사용됩니다.

- **webMethods Integration Server**의 주요 구성 요소에 대한 전반적인 건전성, 성능, 가용성 및 현재 상태를 한눈에 모니터링
- 하위 수준 메트릭이 주의 또는 위험 임계값이 초과되었음을 알리는 경우 프로덕션 응용 프로그램 환경의 잠재적 문제를 조기에 식별
- 성능 정보를 드릴다운하여 지연 중이거나 오류가 발생 중인 **webMethods Integration Server** 비즈니스 프로세스, 서비스 또는 연결 풀을 격리 및 식별

사전 구성된 webMethods 대시보드는 **webMethods Integration Server** 관리 모듈(*WebMethodsISManagementModule.jar*)의 일부로 패키지되어 있습니다.

webMethods 관리 모듈은 **webMethods Integration Server**에 대해 다음과 같은 사전 구성된 대시보드를 제공합니다.

WebMethods - 흐름

WebMethods Integration Server와 해당 주요 구성 요소의 최상위 수준 아키텍처 개요로 모든 서비스, 비즈니스 프로세스, 트레이딩 네트워크 구성 요소 및 외부 백엔드 시스템의 전반적인 건전성에 대한 경고 표시기를 포함합니다.

WebMethods - IS 서비스 개요

흐름, Java 및 XSLT 서비스와 트리거에 대한 요약된 상태로 "평균 응답 시간"에 대한 그래프 및 오류와 중단에 대한 경고 표시기를 포함합니다.

이 대시보드에서 "IS 가장 느린 서비스" 링크 또는 원하는 "전반적인 건전성" 레이블을 두 번 클릭하면 가장 느린 흐름, Java 및 XSLT 서비스의 목록과 가장 느린 트리거의 목록이 나열되는 "WebMethods - IS 가장 느린 서비스" 대시보드가 표시됩니다.

WebMethods - 비즈니스 프로세스

모든 비즈니스 프로세스에 대한 요약된 상태로, 프로세스 및 단계 수준에서의 "평균 응답 시간"과 "간격당 오류 수"의 그래프를 포함합니다.

대시보드에는 프로세스 수준의 취소, 다시 시작, 일시 중지 및 계속된 작업에 대한 경고 표시기, 단계 수준의 응답 시간, 중단, 동시 호출 및 오류에 대한 경고 표시기, 가장 느린 비즈니스 프로세스 목록이 포함됩니다.

WebMethods - WebServices 개요

모든 클라이언트 및 서버 측 웹 서비스에 대한 요약된 상태로, 클라이언트 측 및 서버 측 성능을 강조 표시하는 "평균 응답 시간" 그래프, 가장 느린 클라이언트 측 및 서버 측 서비스의 목록, 그리고 클라이언트 및 서버 서비스에 대한 응답 시간, SOAP 결합, 오류 및 중단에 대한 경고 표시기를 포함합니다.

이 대시보드에서 "웹 서비스 작업" 링크 또는 원하는 "전반적인 건전성" 레이블 또는 그래프를 두 번 클릭하면 "WebMethods - 웹 서비스 작업" 대시보드가 표시됩니다. 여기에는 클라이언트 측 및 서버 측 작업에 대한 "평균 응답 시간" 그래프, SOAP 결합 및 중단에 대한 경고 표시기, 그리고 가장 느린 클라이언트 측 및 서버 측 웹 서비스 작업의 목록이 포함됩니다.

WebMethods - 어댑터

모든 webMethods 어댑터에 대한 요약된 상태로, 모든 어댑터 서비스, 어댑터 알림, 외부 백엔드 시스템에 대한 "평균 응답 시간"의 그래프를 포함합니다.

어댑터 서비스의 경우 대시보드에 다음 항목도 표시됩니다.

- "간격당 오류 수" 및 "중단 수"에 대한 그래프
- 동시 호출, 오류 및 중단에 대한 경고 표시기
- 가장 느린 어댑터 서비스의 목록

어댑터 알림 및 외부 백엔드의 경우 대시보드에 오류 및 중단에 대한 경고 표시기도 표시됩니다.

WebMethods - 트레이딩 네트워크

"트레이딩 네트워크"에 의해 처리된 모든 문서 처리에 대한 요약된 상태로, 모든 문서 유형 및 처리 규칙에 대한 "평균 응답 시간" 및 "간격당 오류 수" 그래프, 모든 오류 및 중단에 대한 경고 표시기 및 가장 느린 처리된 문서의 목록을 포함합니다.

WebMethods - 연결 및 스레드 풀

어댑터 연결 풀, JDBC 연결 풀 및 스레드 풀에 대한 요약된 상태로, 어댑터 연결과 JDBC 연결에 대한 사용할 수 있는 연결 수와 풀 크기에 대한 그래프, 사용 중인 스레드 수와 스레드 풀의 최대 스레드 수에 대한 그래프를 포함합니다.

Workstation 콘솔을 사용하여 사전 구성된 대시보드를 볼 수 있습니다. 또한 사용자 지정 대시보드가 포함되도록 webMethods Integration Server 관리 모듈을 확장하거나, 사용자 지정 메트릭이나 경고가 포함되도록 기본 대시보드 정의를 수정할 수도 있습니다.

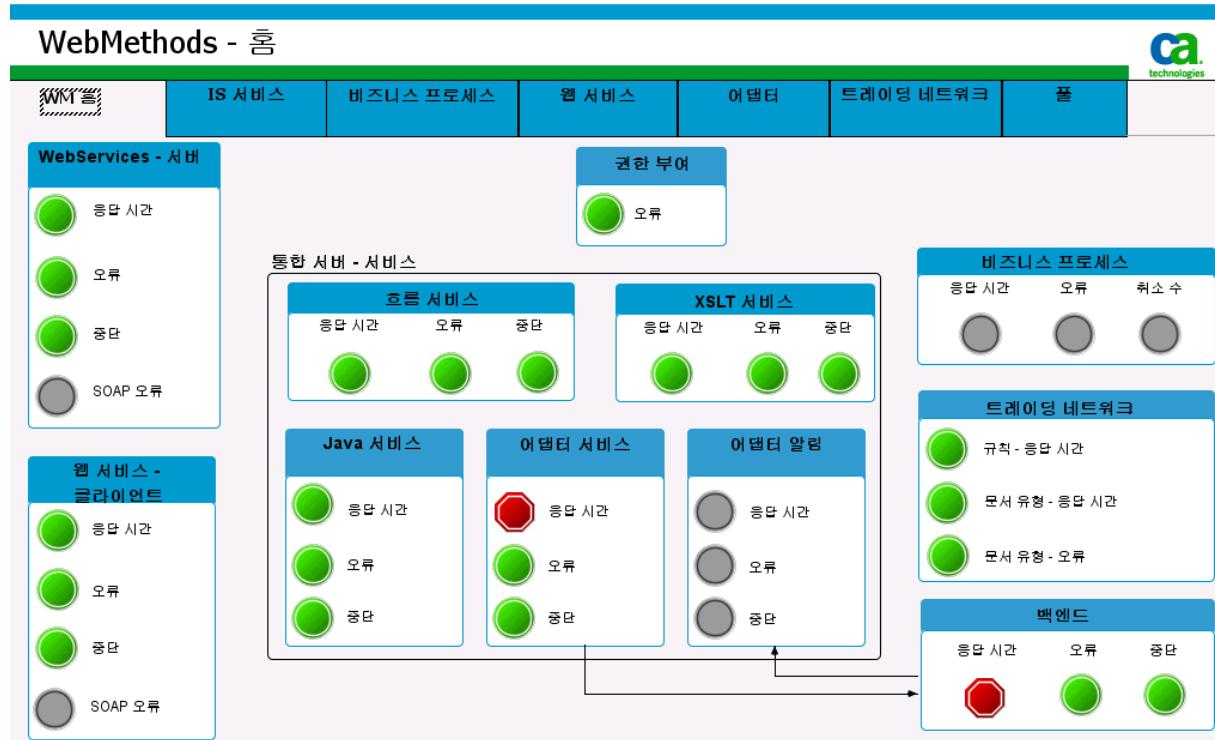
참고: 대시보드 생성과 수정에 대한 자세한 내용은 *CA APM 구성 및 관리 안내서*를 참조하십시오.

다음 단계를 수행하십시오.

1. Enterprise Manager 가 실행되고 있지 않으면 시작합니다.
2. Workstation 을 시작하고 SOA extension for webMethods 가 설치된 Enterprise Manager 에 로그인합니다.
3. "Workstation" > "새 콘솔"을 클릭합니다.

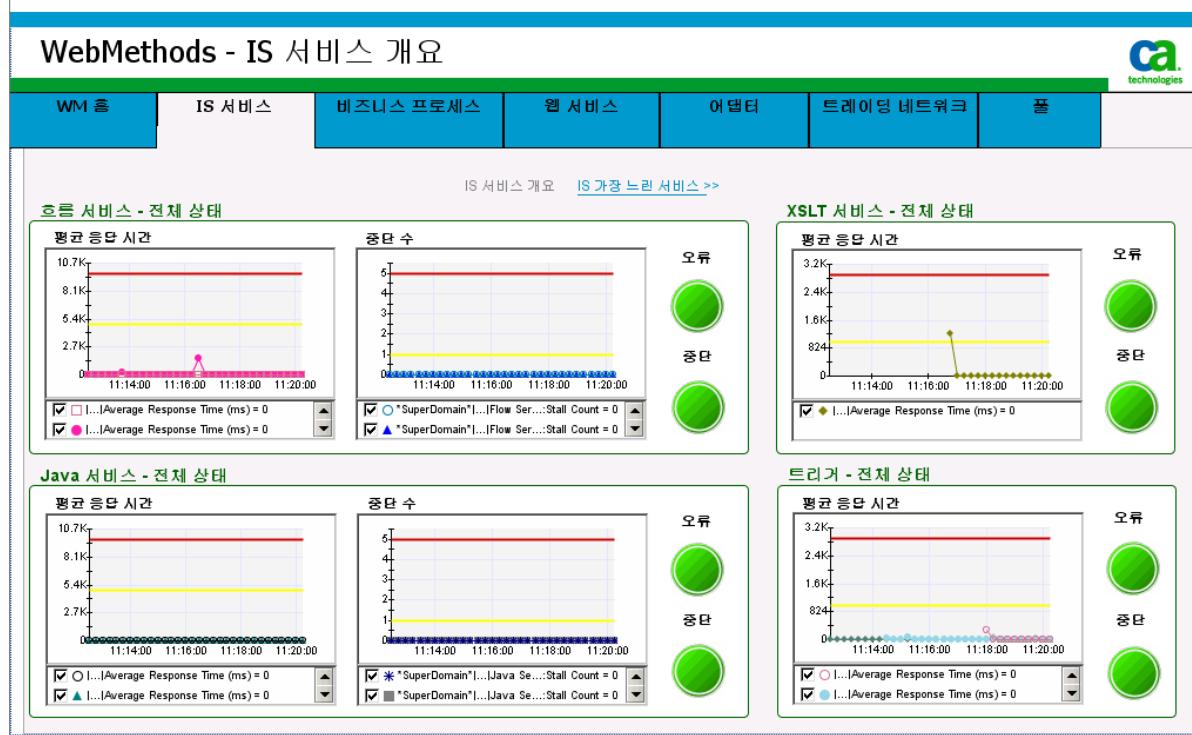
4. "대시보드" 드롭다운 목록에서 webMethods 대시보드 중 하나를 선택합니다.

예를 들어 webMethods 주요 구성 요소 및 내부 작업 흐름의 개요를 보려면 "WebMethods 홈" 대시보드를 선택합니다.

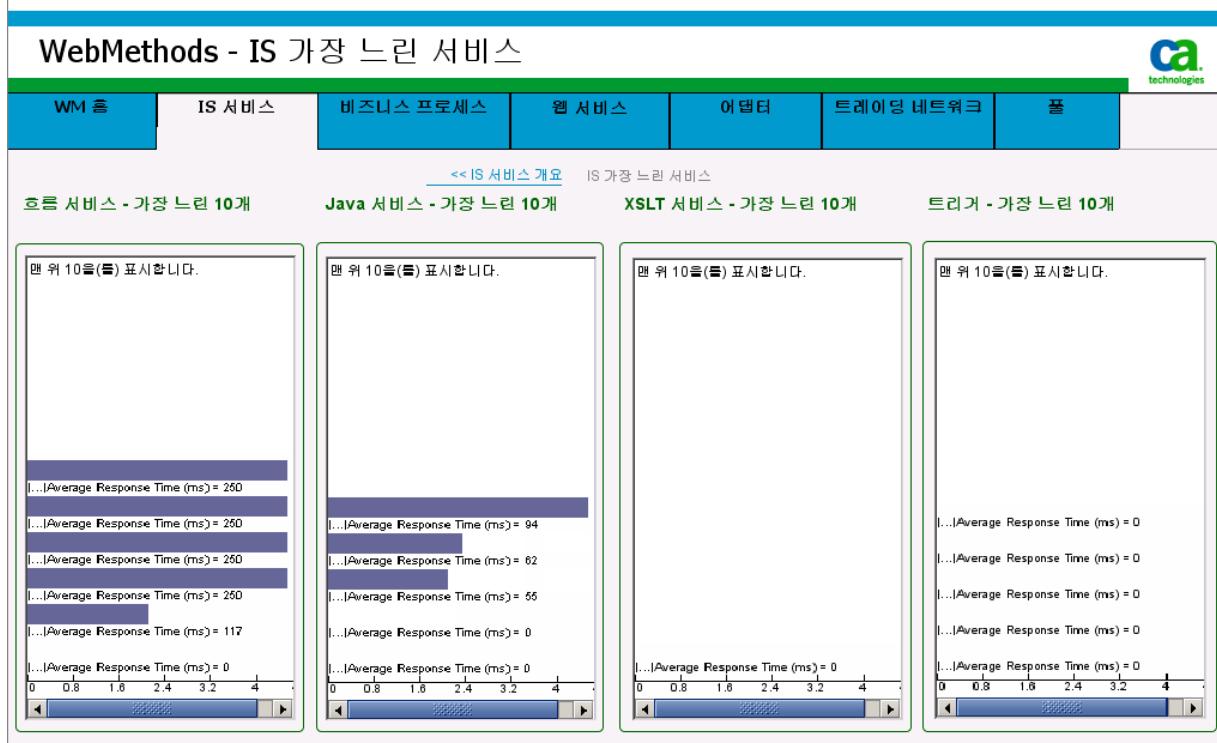


5. 관련 대시보드를 열어서 더 자세한 정보를 보려면 대시보드의 다른 탭이나 경고를 두 번 클릭합니다.

예를 들어 "Java 서비스" 경고를 두 번 클릭하면 "webMethods - IS 서비스 개요"에서 webMethods Integration Server 흐름, Java 및 XSLT 서비스의 전반적인 전전성에 대한 세부 정보를 볼 수 있습니다.



6. "WebMethods - IS 서비스 개요" 대시보드에서 "IS 가장 느린 서비스"를 두 번 클릭하면 가장 느린 개별 흐름, Java, XSLT 및 트리거 서비스의 목록을 볼 수 있습니다. 예:



7. 대시보드에서 특정 서비스, 비즈니스 프로세스 또는 문서 메트릭을 두 번 클릭하면 추가 분석을 위한 Investigator 가 열립니다.

예를 들어 "WebMethods - IS 가장 느린 서비스" 대시보드에서 가장 느린 흐름 서비스를 두 번 클릭하면 선택된 서비스의 "평균 응답 시간"이 표시된 Investigator 가 열립니다.

모니터링 및 표시되는 서비스 필터링

webMethods Integration Server 의 경우 구성 파일에 필터를 지정하여 어떤 흐름이나 Java 서비스를 모니터링하고 Investigator 트리에 포함할지를 제어 할 수 있습니다. 기본 구성 파일은 *wmExtension.config* 입니다. 이 파일을 편집하여 모니터링에서 포함하거나 제외 할 흐름 서비스를 식별하는 필터를 정규식을 사용하여 만들 수 있습니다.

기본 구성 파일 정보

기본 구성 파일인 `wmExtension.config` 는 `<Agent_Home>/common` 디렉터리에 있으며 기본 포함 및 제외 필터로 구성됩니다. 예를 들어 `webMethods` 기본 제공 서비스를 제외하기 위하여 다음과 같은 기본 필터가 정의되어 있습니다.

```
com.wily.wm.service.filter.exclude=wm.* , pub.*
```

또한 구성 파일은 일치하는 `webMethods` 서비스를 포함하기 위하여 다음과 같은 기본 필터도 제공합니다.

```
com.wily.wm.service.include=wm.tn:receive,wm.tn:route:routeBizdoc, pub.prt.tn:handleBizDoc
```

기본 구성 파일을 수정하여 필요에 따라 추가 서비스를 포함 또는 제외하거나, 기본 제공 서비스를 포함한 모든 `webMethods` 서비스를 모니터링하려는 경우 기본 필터를 제거할 수 있습니다. `wmExtension.config` 파일에 필터가 정의되어 있지 않으면 기본 제공 서비스를 포함한 모든 서비스가 `Investigator` 트리에 표시됩니다.

정규식을 사용하여 서비스 포함 및 제외

`wmExtension.config` 구성 파일에서 `com.wily.wm.service.filter.include` 및 `com.wily.wm.service.filter.exclude` 속성을 사용하여 포함하거나 제외할 서비스를 식별할 수 있습니다. 예를 들어 모니터링에서 제외할 서비스를 정의하는 정규식을 사용하여 `com.wily.wm.service.filter.exclude` 속성을 설정할 수 있습니다. `include` 및 `exclude` 속성에는 서비스의 정규화된 이름에 적용할 수 있는 모든 표현식을 사용할 수 있습니다. 예를 들어 `webservice`라는 문자열로 끝나는 모든 흐름 서비스를 제외하려면 `exclude` 속성을 다음과 같이 설정합니다.

```
com.wily.wm.service.filter.exclude=.*webservice
```

여러 개의 필터를 정의하려면 쉼표를 사용하여 정규식을 구분하십시오. 예를 들어 `wm.server` 및 `wm.tomcat`으로 시작하는 모든 흐름 및 Java 서비스를 제외하려면 `com.wily.wm.service.filter.exclude` 속성을 다음과 같이 설정할 수 있습니다.

```
com.wily.wm.service.filter.exclude=wm.server.* ,wm.tomcat.*
```

정규식과 일치하는 모든 흐름 및 Java 서비스가 제외되며 나머지 서비스만 `Investigator`에 표시됩니다. 하지만 유효하지 않은 정규식을 지정하면 아무런 서비스도 모니터링에서 제외되지 않고 기본 제공 서비스를 포함하여 모든 흐름 및 Java 서비스가 표시됩니다.

다른 구성 파일 위치 선택

기본 `wmExtension.config` 파일의 위치는 `<Agent_Home>/common` 디렉터리지만, 서버의 시작 스크립트에서 다른 위치를 지정하여 다른 디렉터리나 다른 구성 파일을 지정할 수 있습니다. 예를 들어 대체 `wmExtension.config` 파일이 `C:\CA-Introscope` 디렉터리에 있는 경우 서버의 시작 스크립트에서 Java 인수에 `com.wily.wm.service.filter.fileloc` 속성을 추가할 수 있습니다.

```
set JAVA_OPTS=%JAVA_OPTS% -Dcom.wily.wm.service.filter.fileloc=C:\CA-Introscope
```

지정한 경로에 이름이 `wmExtension.config` 인 파일이 있는 경우 에이전트는 해당 구성 파일에서 `include` 및 `exclude` 속성을 확인하여 모니터링할 흐름 및 Java 서비스를 결정합니다.

구성 파일의 경로가 서버의 Java 인수에 지정된 경우 에이전트는 해당 파일에 지정된 필터를 사용하여 제외할 흐름 및 Java 서비스를 결정합니다. 구성 파일의 경로가 지정되지 않은 경우 에이전트는 `<Agent_Home>/common/wmExtension.config` 파일에 정의된 필터를 사용합니다. `<Agent_Home>/common/wmExtension.config` 또는 대체 `wmExtension.config` 파일이 없으면 필터링이 수행되지 않으며 기본 제공 서비스를 포함한 모든 흐름 및 Java 서비스가 표시됩니다.

webMethods 에 대한 메트릭 보기 및 탐색

Investigator 트리에서 탐색할 때 `webMethods Integration Server` 인프라의 구성 요소 대부분에 대한 표준 CA Introscope 메트릭을 볼 수 있습니다. 표준 메트릭에 대한 데이터는 `Investigator` 트리에서 노드 및 하위 노드로 표시되는 `webMethods` 특정 메트릭 범주로 수집되고 집계됩니다. 표시되는 특정 메트릭 범주 및 노드 이름은 환경에 배포한 프로세스, 서비스 및 리소스에 따라 다릅니다.

`Investigator` 트리를 탐색할 때는 선택한 노드에 따라 개별 작업에 대한 하위 수준 메트릭 표시나 집계 메트릭 표시를 선택할 수 있으므로 `webMethods Integration Server` 를 통해 배포한 여러 서비스의 전반적인 건전성을 모니터링할 수 있습니다.

Investigator에서 webMethods 메트릭을 보고 탐색하려면

1. 에이전트 노드를 확장한 다음 "WebMethods" 노드를 클릭하여 "개요" 탭을 표시합니다. 여기에는 모니터링 중인 모든 webMethods 흐름 서비스 및 비즈니스 프로세스에 대한 표준 CA Introscope® 메트릭과 요약 정보가 나열됩니다.
 2. 목록에서 흐름 서비스 또는 비즈니스 프로세스를 선택하여 해당 서비스 또는 프로세스에 대한 모든 표준 메트릭을 그래픽 형식으로 표시합니다.
 3. "WebMethods" 노드를 확장하여 webMethods Integration Server의 최상위 수준 메트릭 범주에 대한 하위 노드를 표시합니다.
 4. 하위 노드를 클릭하거나 확장하여 해당 메트릭 범주의 요약 정보가 있는 "개요" 탭을 표시합니다. 예를 들어 "Java 서비스" 노드를 클릭하면 Java 서비스에 대한 요약 메트릭이 "개요" 탭에 표시됩니다.
 5. 원하는 하위 노드를 확장하여 개별 비즈니스 프로세스, 흐름 서비스, Java 서비스 또는 서로 연관된 연결 풀과 메트릭에 대한 세부 정보를 표시합니다.
- 예를 들어 "흐름 서비스" 노드를 확장한 다음 특정 흐름 서비스 이름 및 하위 폴더를 확장하면 개별 흐름 서비스에 대한 메트릭이 표시됩니다.

어댑터에 대한 메트릭

기본 webMethods 어댑터는 조직이 비즈니스 프로세스를 실시간으로 실행할 수 있도록 돋는 엔터프라이즈 응용 프로그램과 정보 리소스에 대한 원활한 연결 기능을 제공합니다. 이러한 어댑터를 사용하면 조직에서 맞춤형 프로그램 작성이나 시간 낭비가 많은 통합 기능 개발을 최소화하고 webMethods를 사용하여 SAP 및 Oracle 응용 프로그램 제품군 같은 패키지 응용 프로그램이나 Microsoft SQL Server 및 Oracle RDBMS 같은 데이터베이스에 연결할 수 있습니다. 사용자가 배포할 수 있는 어댑터는 사용 중인 webMethods Integration Server의 버전에 따라 달라집니다.

어댑터에 대한 메트릭은 사용자가 배포한 어댑터에 대한 자세한 정보를 제공하며, 이러한 메트릭을 통해 해당 어댑터와 관련된 연결, 작업 및 이벤트를 모니터링 할 수 있습니다.

"WebMethods" 노드 아래에 있는 다음과 같은 메트릭 범주의 메트릭을 사용하여 어댑터를 모니터링할 수 있습니다.

어댑터 연결 풀

어댑터 연결을 통해 Integration Server 가 런타임에 외부 응용 프로그램이나 정보 저장소에 연결할 수 있습니다.

어댑터 알림

어댑터 알림을 사용하면 외부 리소스를 모니터링하여 리소스에 의해 시작된 이벤트가 발생할 경우 Integration Server 에 알릴 수 있습니다. 이벤트가 발생하면 어댑터 알림이 webMethods Broker 에 문서를 게시합니다.

어댑터 서비스

어댑터 서비스를 사용하면 외부 리소스에 대한 어댑터 연결을 통해 Integration Server 에서 리소스에 대한 작업을 시작할 수 있습니다.

예를 들어 webMethods 에는 JDBC 드라이버를 사용하여 데이터베이스를 통합하는 데 필요한 사용자 인터페이스, 서비스 및 템플릿 집합을 제공하는 JDBC 어댑터가 있습니다. "어댑터 연결 풀" 범주의 메트릭을 사용하여 데이터베이스에 대한 연결을 모니터링할 수 있습니다.

Integration Server 에서는 JDBC 어댑터 서비스를 사용하여 데이터베이스에서 데이터 삽입, 업데이트 또는 삭제 같은 데이터베이스 작업을 초기화하고 실행할 수 있습니다. 사용자는 어댑터 알림을 통해 데이터베이스를 모니터링할 수 있으며 특정 데이터베이스 테이블에서 업데이트, 삽입 또는 삭제가 발생한 경우 Integration Server 에 알릴 수 있습니다.

예를 들어 어댑터 서비스를 통해 트레이딩 파트너가 사용자의 인벤토리 데이터베이스를 쿼리하여 현재 특정 항목의 재고가 있는지 여부를 확인할 수 있으며, 어댑터 알림을 통해 인벤토리 데이터베이스 테이블이 업데이트된 때를 Integration Server 에 알릴 수 있습니다.

Investigator에서 어댑터 관련 메트릭을 보고 탐색하려면

1. 에이전트 노드를 확장한 다음 "webMethods"를 확장하여 어댑터 메트릭 범주를 표시합니다.
2. "WebMethods" > "어댑터 연결 풀"을 확장한 다음 특정 어댑터 이름을 확장하여 어댑터의 연결 정보를 표시합니다.
3. "WebMethods" > "어댑터 알림"을 확장한 다음 특정 어댑터 이름을 확장하여 어댑터의 알림 정보를 표시합니다. 어댑터에 대한 알림을 구성하지 않은 경우에는 이 메트릭 범주가 표시되지 않습니다.

4. "WebMethods" > "어댑터 서비스"를 확장한 다음 특정 어댑터 유형을 확장하여 활성 연결 목록을 표시합니다.
5. 활성 어댑터 연결 중 하나를 확장하여 Integration Server에서 실행되고 있는 어댑터 서비스 목록을 표시합니다.
6. 개별 서비스를 확장하여 해당 서비스의 메트릭을 표시합니다.

어댑터 연결 풀에 대한 메트릭

개별 어댑터 이름에 대한 "WebMethods" > "어댑터 연결 풀" 하위 노드 아래에서, 구성한 어댑터에 대해 다음과 같은 메트릭을 사용할 수 있습니다.

사용 가능한 연결

어댑터에 사용할 수 있는 여유 연결의 수입니다.

사용 가능한 연결 수는 현재 활성 연결 수에서 사용 중인 연결 수를 빼서 계산합니다. 서버가 한 리소스에 여러 번 연결을 시도하면 사용 가능한 연결 수가 감소합니다.

현재 크기

현재 활성 연결의 총 수입니다.

Maximum Size(최대 크기)

선택한 어댑터에 허용되는 최대 연결 수입니다.

Minimum Size(최소 크기)

선택한 어댑터에 구성된 최소 연결 수입니다.

어댑터 알림에 대한 메트릭

어댑터 알림을 구성한 경우 폴링 프로세스 또는 수신기 프로세스는 적절한 흐름 또는 Java 서비스가 변경 사항에 대응할 수 있도록 데이터베이스 테이블에 대한 삽입, 업데이트 또는 삭제 작업과 같은 외부 리소스 변경 사항을 모니터링합니다.

예를 들어 거래 파트너가 인벤토리 데이터베이스를 쿼리할 수 있도록 어댑터 서비스를 배포한 경우에는 인벤토리 데이터베이스 테이블에 업데이트가 수행될 때마다 이를 Integration Server에 알리는 어댑터 알림을 구성할 수 있습니다. 예를 들어 어댑터 알림에 기반하여 송장을 전송하도록 알림과 연관된 문서를 처리하려면 Integration Server 트리거를 구성할 수 있습니다.

개별 어댑터 서비스 이름의 "WebMethods" > "어댑터 알림" 하위 노드 아래에서 어댑터 알림에 대해 모든 표준 CA Introscope 메트릭을 사용할 수 있습니다.

어댑터 서비스에 대한 메트릭

개별 어댑터 연결 서비스 이름의 "WebMethods" > "어댑터 서비스" 하위 노드 아래에서 어댑터 서비스에 대해 모든 표준 CA Introscope 메트릭을 사용할 수 있습니다.

"간격당 오류 수" 메트릭에는 어댑터 서비스가 실행될 때 발생하는 오류만 포함됩니다. AccessExceptions 또는 서비스가 실행되기 전에 발생하는 다른 모든 유형의 오류는 포함되지 않습니다.

권한 부여에 대한 메트릭

webMethods Integration Server는 대개 여러 사용자 그룹을 포함하며, 이러한 사용자 그룹에는 서로 다른 사용자에게 부여되는 권한을 정의하는 특정 액세스 제어 목록(ACL)이 있습니다. 예를 들어 Administrators, Developers, Monitor Users, Replicators 등과 같은 그룹에 할당되는 기본 권한이 있습니다. 필요한 경우 사용자가 고유한 사용자 그룹과 권한을 만들 수도 있습니다. 사용자가 액세스할 권한이 없는 Integration Server 서비스를 실행하려고 하면 액세스가 거부되고 오류가 기록됩니다.

"WebMethods" > "권한 부여" 노드 아래에서 "간격당 오류 수" 메트릭을 선택하면 권한 부여 오류와 관련된 정보를 볼 수 있습니다.

사용자 액세스가 거부되면 서비스가 호출되지 않기 때문에 서비스의 "간격당 오류 수" 메트릭에 오류가 기록되지 않습니다. 하지만 Investigator에서 "오류" 탭을 클릭하면 "Access Exception"(액세스 예외)" 오류를 볼 수 있습니다. 목록에서 오류를 선택하면 오류에 대한 자세한 정보가 "오류 스냅숏"에 표시됩니다.

비즈니스 프로세스에 대한 메트릭

비즈니스 프로세스는 주문 또는 신입 사원 추가와 같은 비즈니스 이벤트를 완료하는 일련의 단계로 구성됩니다.

"WebMethods" > "비즈니스 프로세스" > "<business_process_name>" 노드 아래의 webMethods 비즈니스 프로세스 및 비즈니스 프로세스 단계에는 다음의 표준 CA Introscope 메트릭만 사용할 수 있습니다.

- 평균 응답 시간(ms)
- 간격당 오류 수
- 간격당 응답 수

비즈니스 프로세스 수준에서 이 메트릭은 프로세스가 성공적으로 완료되는 데 걸리는 시간, 성공적으로 완료된 프로세스의 수 및 오류를 생성하고 실패한 프로세스의 수를 추적합니다. 분산 프로세스의 경우 프로세스 수준의 "평균 응답 시간" 및 "간격당 응답 수"는 비즈니스 프로세스의 마지막 단계가 실행된 에이전트에 대해서만 표시됩니다.

표준 메트릭 이외에 다음 메트릭을 비즈니스 프로세스에 사용할 수 있습니다.

간격당 취소 수

간격에서 취소된 프로세스의 수입니다.

간격당 일시 중지 횟수

간격에서 일시 중지된 프로세스의 수입니다.

간격당 다시 시작 횟수

간격에서 다시 시작된 프로세스의 수입니다.

"Resumes Per Interval"(간격당 계속 수)

간격에서 계속된 프로세스의 수입니다.

또한 webMethods Integration Server 비즈니스 프로세스에 대한 종속성 및 편차 메트릭도 수집할 수 있습니다. 표준 종속성 및 편차 메트릭에 대한 자세한 내용은 [Investigator를 사용하여 SOA 성능 메트릭 보기](#) (페이지 58)를 참조하십시오.

비즈니스 프로세스에 대한 단계 수준 메트릭 보기

기본적으로 webMethods Integration Server 를 모니터링 할 때는 다음과 같이 모든 표준 CA Introscope 메트릭도 비즈니스 프로세스 단계에 사용할 수 있습니다.

WebMethods > BusinessProcesses > *<business_process_name>* > *<step_identifier>* 노드

webMethods Integration Server 6.5.2~6.5.3에 해당: 단일 수준 단계에 대해서만 단계 수준 메트릭이 표시됩니다. 둘 이상의 단계로 내부적으로 실행되는 수동 작업 단계, 참조되는 프로세스, 하위 프로세스 또는 흐름 서비스 단계에 대한 메트릭과 같은 다른 단계 수준 메트릭은 기본적으로 집계되지 않습니다.

webMethods Integration Server 버전 6.5.2~6.5.3 을 모니터링 할 때 비즈니스 프로세스에 대한 전체 단계 수준 메트릭을 확인하려면 webmethods-toggles.pbd 파일에서 다음 행의 주석 처리를 제거하십시오.

```
TurnOn: BProc<version_number>FlowStepMarkTracing  
TurnOn: BProc<version_number>StepFlowTracing
```

참고: webMethods Integration Server 요구 사항에 대해서는 *Compatibility Guide*(호환성 안내서)를 참조하십시오.

다음 단계를 수행하십시오.

1. 에이전트 노드를 확장한 다음 "webMethods" > "비즈니스 프로세스"를 확장합니다.

webMethods Designer에서 정의하고 사용자 환경에 패키지로 배포한 비즈니스 프로세스가 표시됩니다.

실행 중인 여러 버전의 비즈니스 프로세스를 구분하기 위해 비즈니스 프로세스 이름에 밑줄(_)과 버전 번호가 추가됩니다.

2. 프로세스 이름을 모두 확장합니다.
해당 프로세스에 대해 정의한 단계가 나타납니다.
3. 단계 식별자(StepID)를 모두 확장합니다.

Integration Server에서 단계를 나타내는 단계 수준 메트릭 및 흐름 서비스 메트릭이 나타납니다.

완료된 프로세스의 평균 응답 시간 정보

"My WebMethods Server"(내 WebMethods 서버)를 사용하면 실행 중인 webMethods 비즈니스 프로세스를 취소하거나 일시 중지할 수 있습니다. 대부분의 경우 "My WebMethods Server"(내 WebMethods 서버)를 사용하여 취소하거나 일시 중지한 프로세스를 "My WebMethods Server"(내 WebMethods 서버)에서 수동으로 재개할 수 있습니다. 이러한 프로세스를 "My WebMethods Server"(내 WebMethods 서버)에서 다시 제출하여 완료될 때까지 실행할 수 있으므로 프로세스 수준의 "평균 응답 시간" 메트릭에는 성공적으로 완료된 프로세스만 나타납니다.

프로세스가 실패하거나 일시 중지된 후 "My WebMethods Server"(내 WebMethods 서버) 내에서 다시 제출된 경우 "평균 응답 시간" 메트릭에는 프로세스의 초기 호출부터 다시 제출된 후 성공적으로 완료될 때까지의 시간이 반영됩니다. 프로세스가 실패하고 "My WebMethods Server"(내 WebMethods 서버)를 사용하여 다시 제출되지 않은 경우에는 이 프로세스는 "평균 응답 시간" 메트릭에 포함되지 않습니다. 하지만 프로세스 흐름 중에 종료 단계 등을 통해 프로세스가 취소된 경우에는 "평균 응답 시간" 메트릭에 포함됩니다.

비즈니스 프로세스에 여러 Integration Server 가 관여하는 경우 비즈니스 프로세스가 완료된 Integration Server 에 대한 "평균 응답 시간" 메트릭만 보고됩니다.

완료된 프로세스의 간격당 응답 수 정보

"간격당 응답 수" 메트릭은 성공적으로 완료된 프로세스 수를 나타냅니다. 비즈니스 프로세스에 여러 Integration Server 가 관여하는 경우 비즈니스 프로세스가 완료된 Integration Server 에 대한 "간격당 응답 수" 메트릭만 보고됩니다.

실패한 프로세스의 간격당 오류 수 정보

"간격당 오류 수" 메트릭은 프로세스 실행이 완료되지 못한 경우에만 표시됩니다.

프로세스의 여러 단계에 대한 동시 호출 정보

"SubProcess"(하위 프로세스), "Referenced Process"(참조된 프로세스), "Human Task"(수동 작업) 같은 일부 단계 유형은 webMethods Designer에서 단일 단계로 표시되지만 내부적으로는 여러 단계로 실행됩니다. 예를 들어 "Human Task"(수동 작업) 단계는 내부적으로 *PRE_StepID* 및 *POST_StepID*의 두 단계로 실행됩니다. "동시 호출" 메트릭은 이러한 내부의 여러 단계를 집계하여 단일 단계로 취급하지 않고 개별 호출로 계산합니다.

흐름 서비스에 대한 메트릭

흐름 서비스는 webMethods Flow 언어로 작성된 서비스입니다. 흐름 서비스는 각각 입력과 출력이 명확하게 정의된 일련의 흐름 단계로 구성됩니다. 각 흐름 단계는 webMethods 가 런타임에 해석하고 실행하는 기본 작업 단위입니다. 특정 서비스에서 모든 흐름 단계는 같은 스레드를 사용하여 실행되므로 한 단계의 출력을 흐름의 다음 단계에 대한 입력으로 사용할 수 있습니다.

"WebMethods" > "흐름 서비스" 노드 아래에서 개별 webMethods 흐름 서비스에 대해 모든 표준 CA Introscope 메트릭을 사용할 수 있습니다. 데이터가 개별 흐름 단계마다 수집되고 하나의 흐름 서비스 및 여러 흐름 서비스에 대한 메트릭으로 집계되므로 Integration Server에서의 모든 흐름 서비스의 전반적인 건전성을 모니터링할 수 있습니다.

표시되는 노드 이름은 모니터링하도록 선택되고 필터를 사용하여 제외되지 않은 서비스의 전체 이름입니다. 모니터링 할 흐름 서비스의 필터링에 대한 자세한 내용은 [모니터링 및 표시되는 서비스 필터링](#) (페이지 304)을 참조하십시오.

다른 흐름 서비스를 호출하는 흐름 서비스

대개의 경우 모든 표준 CA Introscope 메트릭 및 해당 집계 메트릭 값은 다른 응용 프로그램 구성 요소에 적용될 때와 동일한 방식으로 webMethods 흐름 서비스에 적용됩니다.

하나의 흐름 서비스가 다른 흐름 서비스를 호출하는 경우 두 흐름 서비스 모두에 대한 메트릭이 "Investigator" 트리의 "WebMethods" > "흐름 서비스" 노드 아래에 별도의 노드로 표시됩니다. 이는 다른 노드 아래에 있는 노드와 중첩되지 않습니다.

동일한 webMethods Integration Server에서 실행된 모든 흐름 단계를 호출 순서대로 보려면 트랜잭션 추적 세션을 시작하십시오.

Java 서비스에 대한 메트릭

Java 서비스는 Java 언어를 사용하여 구현되거나, 다른 언어로 작성되고 Java 클래스를 사용하여 래핑되어 webMethods Integration Server에서 서비스로 사용되는 모든 사용자 정의 또는 내부의 기본 제공 webMethods 서비스입니다. 예를 들어 webMethods Integration Server가 제공하는 모든 내장 서비스는 Java 서비스입니다.

"WebMethods" > "Java 서비스" 노드 아래에서 배포한 개별 webMethods Java 서비스에 대하여 모든 표준 CA Introscope 메트릭을 사용할 수 있습니다.

webMethods에서 같은 폴더에 있는 Java 서비스는 같은 클래스의 메서드입니다. 예를 들어 전체 이름이 `recording.user.accounts:createAccount`인 Java 서비스는 Java 패키지(`recording.user`), Java 클래스(`accounts`) 및 Java 메서드(`createAccount`)로 구성됩니다. Investigator에 표시되는 노드 이름은 모니터링하도록 선택되고 필터를 사용하여 제외되지 않은 서비스의 전체 이름을 나타냅니다.

모니터링 할 Java 서비스의 필터링에 대한 자세한 내용은 [모니터링 및 표시되는 서비스 필터링](#) (페이지 304)을 참조하십시오.

JDBC 풀에 대한 메트릭

webMethods Integration Server 는 JDBC 연결 지점을 통해 데이터베이스에 연결하여 사용자, 문서, 내부 서버 함수, 감사 및 오류 로그와 기타 정보를 수집하여 저장합니다. JDBC 풀에서 허용되는 최대 JDBC 연결 수는 Integration Server Administrator 를 사용하여 구성할 수 있습니다.

"WebMethods" > "JDBC 풀" 하위 노드 아래에서 다음과 같은 메트릭을 사용하여 JDBC 사용량을 모니터링 할 수 있습니다.

사용 가능한 연결

JDBC 풀에서 사용 가능한 연결 수입니다.

사용 가능한 연결 수는 현재 활성 연결 수에서 사용 중인 연결 수를 빼서 계산합니다.

현재 크기

현재 활성 JDBC 연결의 총 수입니다.

Maximum Size(최대 크기)

선택한 풀에 구성된 최대 JDBC 연결 수입니다.

Minimum Size(최소 크기)

선택한 풀에 구성된 최소 JDBC 연결 수입니다.

스레드 풀에 대한 메트릭

webMethods Integration Server 는 서비스 실행, webMethods Broker 의 문서 검색 및 트리거 실행에 스레드를 사용합니다. 서버가 시작될 때 스레드 풀은 최소 수의 스레드로 초기화됩니다. 이후에는 허용된 최대 수에 도달할 때까지 서버가 풀에 필요한 스레드를 추가합니다. 최대 수의 스레드가 사용되는 경우 서버는 프로세스가 완료될 때까지 기다린 다음 완료된 프로세스의 스레드를 풀에 반환하여 추가 프로세스를 시작할 수 있는 상태로 만듭니다.

"WebMethods" > "스레드 풀" 하위 노드 아래에서 다음과 같은 메트릭을 사용하여 스레드 사용량을 모니터링 할 수 있습니다.

Maximum Size(최대 크기)

선택한 스레드 풀에 구성된 최대 스레드 수입니다.

Minimum Size(최소 크기)

선택한 스레드 풀에 구성된 최소 스레드 수입니다.

사용된 스레드

선택한 스레드 풀에서 현재 사용되고 있는 스레드 수입니다.

트레이딩 네트워크에 대한 메트릭

트레이딩 네트워크를 사용하면 문서를 교환하는 여러 조직이 B2B 관계를 설정하고 다양하게 이용할 수 있습니다. 예를 들어 구매자, 공급자, 전략적 파트너 또는 기타 조직을 문서를 교환하는 트레이딩 파트너로 식별할 수 있습니다. 문서 교환을 통해 조직 간의 비즈니스 프로세스를 간소화 할 수 있습니다. webMethods Integration Server 에서는 트레이딩 네트워크가 트레이딩 파트너 간의 게이트웨이 역할을 합니다.

"WebMethods" > "트레이딩 네트워크" 노드 아래에서 webMethods 트레이딩 네트워크에 대해 다음과 같은 표준 메트릭을 사용할 수 있습니다.

■ 간격당 오류 수

"WebMethods" > 트레이딩 네트워크 노드 아래의 다음과 같은 메트릭 범주를 사용하여 트레이딩 네트워크에서 XML 문서 인식 및 처리 규칙을 모니터링 할 수 있습니다.

문서 유형

문서 유형은 다양한 파트너 관계에서 교환하는 데이터의 구조 및 유형을 식별합니다.

처리 규칙

처리 규칙은 문서가 Integration Server 를 거쳐 대상으로 라우팅되는 방식을 설명합니다.

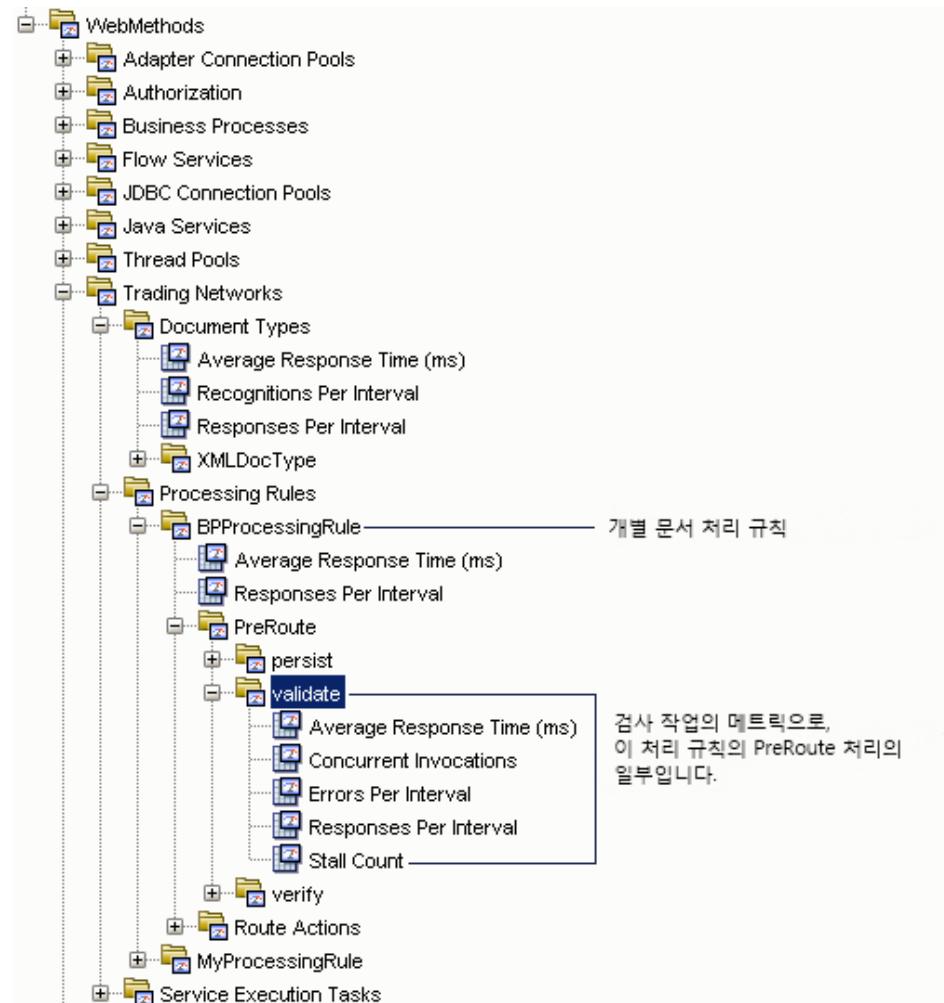
서비스 실행 작업

서비스 실행 작업은 처리 규칙에서 다시 시도 제한을 사용하여 서비스를 비동기로 실행할 때 생성됩니다.

서비스 스레드

서비스 스레드는 다시 시도 제한 없이 라우팅되는 문서의 비동기 처리를 위해 생성됩니다.

예를 들어 "WebMethods" > "트레이딩 네트워크" > "문서 유형" 하위 노드를 확장하여 개별 문서 유형의 문서 인식 및 수락 메트릭을 비롯한 특정 문서 유형에 대한 메트릭을 표시할 수 있습니다. 마찬가지로 "WebMethods" > "트레이딩 네트워크" > "처리 규칙" 하위 노드를 확장하여 개별 사전 라우팅 및 라우팅 작업의 메트릭을 비롯한 특정 처리 규칙에 대한 메트릭을 표시할 수 있습니다.



문서 유형에 대한 메트릭

"WebMethods" > "트레이딩 네트워크" > "문서 유형" 노드 아래에서 webMethods 트레이딩 네트워크에 대해 다음과 같은 표준 CA Introscope 메트릭만 사용할 수 있습니다.

- 평균 응답 시간(ms)
- 간격당 응답 수

"문서 유형"에 대한 "평균 응답 시간" 및 "간격당 응답 수" 메트릭은 문서가 처리될 때 집계됩니다. 이 메트릭은 처리 규칙이 실행될 때만 보고됩니다. 문서가 제출되었지만 처리 규칙에 의해 처리되지 않은 경우에는 "평균 응답 시간" 및 "간격당 응답 수" 메트릭이 보고되지 않습니다.

올바른 문서가 제출되면 표준 메트릭 이외에도 다음 메트릭을 "WebMethods" > "트레이딩 네트워크" > "문서 유형" 노드 아래에서 사용할 수 있습니다.

간격당 인식 수

15 초 간격이 끝날 때 올바른 트레이딩 파트너 문서로 인식되는 문서의 수입니다.

"WebMethods" > "트레이딩 네트워크" > "문서 유형" > *<document_type_name>* 노드 아래에서 인식되는 개별 문서 유형에 대해 모든 표준 CA Introscope 메트릭을 사용할 수 있습니다.

처리 규칙에 대한 메트릭

개별 처리 규칙에 대한 "WebMethods" > "트레이딩 네트워크" > "처리 규칙" > *<processing_rule_name>* 하위 노드 아래에서 webMethods 트레이딩 네트워크에 대해서는 다음 표준 CA Introscope 메트릭만 사용할 수 있습니다.

- 평균 응답 시간(ms)
- 간격당 응답 수

처리 규칙의 특정 작업에 대한 "WebMethods" > "트레이딩 네트워크" > "처리 규칙" > *<processing_rule_name>* > "사전 라우트" 또는 "라우트 작업" 하위 노드 아래에서 webMethods 개별 처리 규칙 작업에 대해 모든 표준 CA Introscope 메트릭을 사용할 수 있습니다.

처리 규칙에 대한 평균 응답 시간 정보

처리 규칙에 대한 "평균 응답 시간" 메트릭은 문서를 동기적으로 라우팅하는 데 걸린 평균 응답 시간을 집계합니다. 처리 규칙에서 동기적 호출의 경우 이 메트릭은 집계된 "문서 유형에 대한 평균 응답 시간"과 동일합니다.

서비스 실행 작업에 대한 메트릭

webMethods 트레이딩 네트워크에 대해 "WebMethods" > "트레이딩 네트워크" > "서비스 실행 작업" 하위 노드 아래에서 다음 메트릭을 사용할 수 있습니다.

- New Tasks Per Interval(간격당 새 작업 수)
- Task Failures Per Interval(간격당 작업 실패 수)
- Tasks Completed Per Interval(간격당 완료된 작업 수)

또한 "WebMethods" > "트레이딩 네트워크" > "서비스 실행 작업" > "호출" 하위 노드 아래에서 개별 호출 작업에 대해 다음의 표준 CA Introscope 메트릭을 사용할 수 있습니다.

- 평균 응답 시간(ms)
- 간격당 응답 수

서비스 스레드에 대한 메트릭

"WebMethods" > "트레이딩 네트워크" > "서비스 스레드" 하위 노드 아래에서 webMethods 트레이딩 네트워크에 대해 다음의 표준 CA Introscope 메트릭을 사용할 수 있습니다.

- 평균 응답 시간(ms)
- 간격당 응답 수

트리거에 대한 메트릭

webMethods Broker 또는 JMS(Java Message Service)를 사용하여 문서를 처리하도록 Integration Server 트리거를 구성할 수 있습니다. webMethods Broker 트리거는 로컬로 게시되거나 webMethods Broker로 배달된 문서를 구독하고 이를 처리하는 트리거입니다. JMS 트리거는 큐 또는 항목과 같은 JMS 공급자의 대상에서 메시지를 수신한 다음 이 메시지를 처리하는 트리거입니다.

개별 트리거 이름에 대한 "WebMethods" > "트리거" 하위 노드 아래에서 트리거에 대해 모든 표준 CA Introscope 메트릭을 사용할 수 있습니다. 브로커, 로컬 또는 JMS 트리거를 구성하지 않은 경우에는 이 메트릭 범주가 표시되지 않습니다.

WebServices 에 대한 메트릭

웹 서비스는 한 단위로 패키지되어 사용자나 소프트웨어 프로그램이 사용할 수 있도록 네트워크에 게시되는 구성 단위입니다.

- 지원되는 버전의 webMethods Integration Server 에 해당 - 웹 서비스 커넥터에 따라 Integration Server 가 웹 서비스 소비자(클라이언트) 역할을 하는지 아니면 웹 서비스 공급자(서버) 역할을 하는지가 정의됩니다. 예를 들어 외부적으로 배포하는 모든 흐름 서비스 또는 Java 기반 서비스를 웹 서비스로 공개할 수 있습니다. 서비스에 대한 정보를 UDDI 레지스트리에 게시하는 공급자 웹 서비스 설명자를 사용할 수 있습니다. 또한 webMethods Integration Server 는 클라이언트로서 서비스를 요청하는 소비자 웹 서비스 설명자를 사용하여 외부 응용 프로그램 서버의 웹 서비스를 호출할 수도 있습니다.
- webMethods Integration Server 6.5.2~6.5.3 에 해당 - WSDL 문서에서 웹 서비스 커넥터를 생성하여 원격 웹 서비스를 호출할 수 있습니다. 웹 서비스 커넥터는 입력 및 출력 서명이 있는 흐름 서비스입니다. 서명은 해당 서명이 생성된 WSDL 문서의 입력 및 출력 메시지에 해당합니다. webMethods Integration Server 에서는 Integration Server 및 Developer 를 사용하여 Integration Server 패키지의 모든 기존 서비스를 웹 서비스로 전환할 수 있습니다.

참고: webMethods Integration Server 요구 사항은 *CA APM 호환성 안내서*를 참조하십시오.

webMethods Integration Server 에서 클라이언트와 서버 웹 서비스 및 작업에 대해 모든 표준 CA Introscope 메트릭을 사용할 수 있습니다. 또한 "간격당 SOAP 오류 수"는 SOA 플랫폼에서 웹 서비스를 모니터링하는 모든 확장에 대한 표준 메트릭입니다.

추가 정보:

["오류" 탭에서 SOAP 결합 및 오류 메트릭 보기](#) (페이지 68)

서버 측 메트릭 정보

웹 서비스 공급자인 webMethods 에는 웹 서비스 요청을 처리하는 다음과 같은 여러 SOAP 프로세서가 있습니다.

- 기본 웹 서비스 SOAP 프로세서
- SOAP 원격 프로시저 호출을 받고 처리하는 SOAP RPC 프로세서
- 프로세스 지시문이 정의되지 않았거나 생략된 경우 메시지를 처리하는 기본 SOAP 메시지 핸들러

서버 측 웹 서비스 메트릭에는 이 세 SOAP 프로세서 모두에 대한 정보가 포함됩니다. 하지만 사용자 지정 SOAP 프로세서는 메트릭에 포함되지 않습니다.

클라이언트 측 메트릭 정보

클라이언트 측 메트릭은 외부 응용 프로그램 서버에서 실행되는 웹 서비스 요청의 실행을 나타냅니다.

webMethods Integration Server 는 웹 서비스 소비자 역할을 할 경우 각 작업에 대한 웹 서비스 커넥터를 자동으로 생성합니다. 클라이언트는 이 커넥터를 통해 웹 서비스 끝점에 직접 바인딩합니다. 웹 서비스 커넥터가 실행되면 웹 서비스를 호출하는 요청이 웹 서비스 구현으로 직접 이동합니다. 내부적으로 웹 서비스 커넥터는 흐름 서비스이며 "흐름 서비스" 메트릭을 사용하여 모니터링할 수 있습니다.

"WebServices" > "클라이언트" 메트릭은 클라이언트가 직접 요청하는 작업이 아니라 외부 웹 서비스를 호출하는 커넥터의 실행을 나타냅니다.

서버 측 메트릭은 webMethods Integration Server 에서 실행되는 웹 서비스의 실행을 나타냅니다.

XSLT 서비스에 대한 메트릭

XSL(Extensible Stylesheet Language) 및 XSLT(XSL Transformations)는 원본 XML 문서를 다른 문서로 변환하기 위한 XML 기반 언어를 제공합니다. 예를 들어 원본 XML 문서를 사용하여 새 XML 문서를 생성하거나, HTML로 변환하여 웹 페이지로 표시하거나, 일반 텍스트로 게시할 수 있습니다.

XSLT 서비스를 호출하면 Integration Server는 연관된 문서인 스타일시트에서 지침을 검색한 다음 이 지침을 적용하여 원본 XML 문서를 스타일시트에 정의된 형식의 새 문서로 변환합니다.

"WebMethods" > "XSLT 서비스" 노드 아래에서 개별 webMethods XSLT 서비스를 모니터링하는 데 모든 표준 CA Introscope 메트릭을 사용할 수 있습니다.

기본 webMethods 메트릭 그룹화 보기

SOA extension for webMethods Integration Server에는 기본 대시보드와 경고를 정의하는 데 사용되는 기본 메트릭 그룹화가 포함되어 있습니다. 사용자 지정 대시보드와 경고에서도 이 기본 메트릭 그룹화를 사용할 수 있습니다.

기본 메트릭 그룹화는 Enterprise Manager extension for webMethods Integration Server에 webMethods Integration Server 관리 모듈(*WebMethodsISManagementModule.jar*)의 일부로 패키지되어 있습니다.

webMethods 에이전트에 대한 기본 메트릭 그룹화를 보려면

1. Investigator에서 Workstation > 새 관리 모듈 편집기를 클릭합니다.
2. "*SuperDomain*" > "관리 모듈" > "WebMethods IS(*SuperDomain*)"를 확장합니다.
3. "메트릭 그룹화" 노드를 확장하여 webMethods 관리 모듈에 대해 정의된 모든 메트릭 그룹화를 표시합니다.
4. 특정 메트릭 그룹화를 클릭하여 "뷰어" 창에서 해당 정의를 봅니다.

모든 메트릭 그룹화의 기본 설정을 수정하거나 사용자 지정 메트릭 그룹화를 직접 생성할 수 있습니다.

참고: 메트릭 그룹화 생성 또는 수정에 대한 자세한 내용은 *CA APM Workstation* 사용자 안내서를 참조하십시오.

기본 webMethods 경고 보기

SOA extension for webMethods에는 사전 구성된 대시보드에 사용되는 기본 경고 정의가 포함됩니다. 이 기본 경고를 사용자 지정 대시보드에서도 사용할 수 있습니다. 대부분의 기본 경고는 기본 "주의" 및 "위험" 임계값으로 사전 구성되며 임계값이 초과되거나 심각도가 증가할 경우 콘솔에 알림을 보내도록 사전 구성됩니다.

기본 경고 정의는 Enterprise Manager extension for webMethods Integration Server에 webMethods Integration Server 관리 모듈(*WebMethodsISManagementModule.jar*)의 일부로 패키지되어 있습니다.

webMethods 에이전트에 대한 기본 경고 정의를 보려면

1. Investigator에서 Workstation > 새 관리 모듈 편집기를 클릭합니다.
2. "*SuperDomain*" > "관리 모듈" > "WebMethods IS(*Super Domain*)"를 확장합니다.
3. "메트릭 그룹화" 노드를 확장하여 webMethods Integration Server 관리 모듈에 대해 정의된 모든 메트릭 그룹화를 표시합니다.
4. 특정 경고를 클릭하여 "뷰어" 창에서 해당 정의를 봅니다.

특히 기본 "주의" 및 "위험" 임계값과 중요 경고에 대한 사전 정의된 작업을 검토하고 환경에 맞게 조정해야 합니다. 예를 들어 필요한 경우 임계값을 조정하거나, 알림을 추가하거나, 수정 조치를 정의해야 할 수 있습니다.

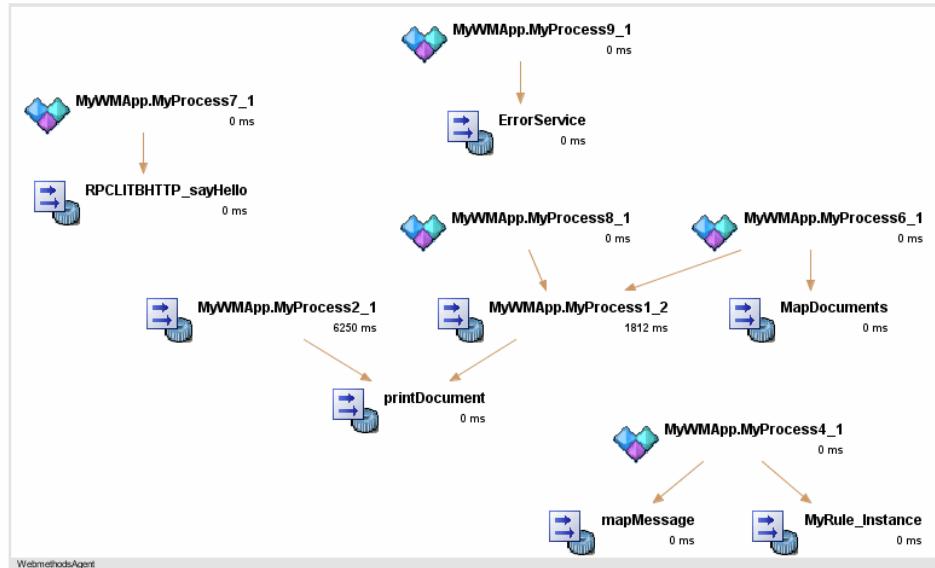
경고에 대한 모든 기본 설정을 수정하거나 사용자 지정 경고를 직접 생성할 수 있습니다.

참고: 경고 생성 또는 수정에 대한 자세한 내용은 *CA APM Workstation 사용자 안내서*를 참조하십시오.

webMethods 종속성 보기

Investigator 트리에서 적절한 webMethods 노드를 선택하고 "SOA 종속성 맵" 탭을 클릭하여 webMethods 흐름, java, 어댑터, 그리고 웹 서비스 및 webMethods 비즈니스 프로세스의 종속성을 확인할 수 있습니다.

선택한 노드에 따라 종속성 맵에 표시되는 컨텍스트가 결정됩니다. 종속성 맵이 표시되면 보고자 하는 컨텍스트와 세부 정보 수준을 룰업하여 축소하거나 롤다운하여 확장할 수 있습니다. 예를 들어 모든 비즈니스 프로세스에 대한 종속성을 최상위 수준에서 표시하려면 Investigator에서 "비즈니스 프로세스" 노드를 선택하고 "SOA 종속성 맵" 탭을 클릭합니다.



특정 비즈니스 프로세스의 종속성을 상위 수준에서 보려면 Investigator에서 비즈니스 프로세스 이름을 선택할 수 있습니다. 계속해서 맵에 종속성 수준을 추가하여 비즈니스 프로세스의 전체 작업 흐름을 표시하거나 필요한 경우 맵의 특정 노드를 확대할 수 있습니다. 종속성 맵의 탐색에 대한 자세한 내용은 [SOA 종속성 맵 사용](#) (페이지 73)을 참조하십시오.

webMethods에 대한 트랜잭션 추적

트랜잭션 추적은 비즈니스 트랜잭션 완료에 관련된 특정 단계의 세부 정보 뷰 또는 요약 뷰를 제공합니다. webMethods Integration Server 비즈니스 프로세스 또는 응용 프로그램 서비스의 경우 다음 프로토콜을 통해 라우팅된 작업이 포함된 트랜잭션을 추적할 수 있습니다.

- SOAP(Simple Object Access Protocol)
- HTTP(Hypertext Transport Protocol)
- HTTPS(Hypertext Transport Protocol Secure)
- JMS(Java Message Service)
- webMethods Broker Message Service

webMethods 서비스가 관련된 트랜잭션에는 동기적 호출과 비동기적 호출 및 서로 다른 스레드를 사용하여 동시에 실행 중인 동작이 포함될 수 있습니다. 여러 스레드나 프로세스에서 실행되는 트랜잭션에 대한 트랜잭션 추적을 사용하도록 설정하기 위해 상관 관계 식별자가 삽입되어 각 구성 요소 및 작업에서 트랜잭션 단계로 사용됩니다. 그러면 수행된 특정 작업에 대한 세부 정보와 각 작업이 완료되는 데 걸린 시간을 볼 수 있습니다.

또한 추적 중인 모든 노드에서 CA APM for SOA 및 CA APM for webMethods Integration Server 가 사용되도록 설정되어 있으면 모든 조합의 플랫폼에서 비즈니스 트랜잭션을 추적할 수 있습니다. 따라서 트랜잭션이 여러 JVM 또는 CLR에서 실행되는 경우에도 트랜잭션에 대한 세부 정보를 볼 수 있습니다.

크로스 프로세스 트랜잭션 추적 구성 정보

일반적으로 상관 관계 정보가 한 프로세스에서 다른 프로세스로 전달될 수 있도록 하여 크로스 프로세스 트랜잭션 추적 기능을 구성할 수 있습니다. 이때 정보는 SOAP 또는 HTTP 헤더를 사용하여 전달됩니다. webMethods Integration Server 6.5.2~6.5.3에 해당 -- RPC 스타일 웹 서비스 클라이언트(커넥터)를 사용할 때는 SOAP 헤더를 상관 관계 식별자 전달에 사용할 수 없습니다.

webMethods Integration Server RPC 웹 서비스 커넥터가 관여된 비즈니스 트랜잭션에 대한 트랜잭션 추적을 보려면 상관 관계 식별자 전달에 HTTP 헤더를 사용하도록 [에이전트가 상호 관련된 추적을 구성](#) (페이지 121)하십시오.

크로스 프로세스 트랜잭션 추적의 가치 이해

크로스 프로세스 트랜잭션 추적은 SOA(서비스 지향 아키텍처)에서 느슨하게 결합된 서비스가 수행하는 작업에 대한 중요한 정보를 제공합니다. 크로스 프로세스 트랜잭션 추적을 사용하면 다음을 확인할 수 있습니다.

- 흐름 또는 Java 서비스를 통해 비즈니스 프로세스가 라우팅되는 방법
- 트랜잭션 중에 호출되고 실행되는 서비스
- 트랜잭션 중에 발생하는 호출의 순서
- 요청 또는 응답 처리가 가장 느린 지점

샘플 트랜잭션 추적 시작 및 보기

다음과 같은 방법으로 트랜잭션 추적 세션을 시작할 수 있습니다.

- SOA 종속성 맵의 맵 노드에서 바로 시작합니다.
- "Workstation" > "새 트랜잭션 추적 세션"을 클릭하여 Workstation에서 수동으로 시작합니다.

종속성 맵에서 트랜잭션 추적을 시작하는 경우에는 맵 노드 유형에 따라 기본 필터가 자동으로 설정됩니다. 새 트랜잭션 추적 세션을 수동으로 시작하는 경우에는 webMethods에 대한 다음 필터 유형 중 하나를 선택할 수 있습니다.

- 비즈니스 프로세스
- namespace
- 작업 이름

필터를 구성하고 트랜잭션 추적 세션을 시작하면 "트랜잭션 추적 뷰어"가 표시됩니다. 추적을 선택하여 트랜잭션에서 생성된 호출에 대한 추가 세부 정보를 볼 수 있습니다. 이 세부 정보에는 webMethods Integration Server에서 실행된 모든 트리거, 흐름 서비스 작업 또는 비즈니스 프로세스 단계가 포함됩니다.

webMethods 프로세스에 대한 시퀀스 뷰 사용

webMethods Integration Server가 관련된 트랜잭션에는 비동기 호출이 포함되는 경우가 많으므로 "시퀀스 뷰"를 클릭하여 트랜잭션의 일부로 비동기적으로 실행된 프로세스에 대한 트랜잭션 작업 흐름을 확인하면 유용합니다. "시퀀스 뷰"에는 프로세스가 실행되는 순서가 시퀀스를 식별할 수 있는 범위까지 표시됩니다. webMethods Integration Server 트랜잭션의 경우 시퀀스가 반드시 전통적 호출자-피호출자 관계를 나타내는 것은 아니지만 하나의 프로세스가 다른 프로세스의 실행을 트리거하는 시점을 보여 줍니다.

하지만 webMethods Integration Server 프로세스의 처리 시간은 호출된 프로세스와 연관된 처리 시간을 포함하여 프로세스의 시작 시점부터 완료 시점까지의 전체 기간을 사용하여 계산됩니다. 호출 프로세스의 기간에서 비차단 동기적 및 비동기적 프로세스의 처리 시간을 뺀 순 기간은 webMethods Integration Server 프로세스에 대해 지원되지 않습니다.

트랜잭션 추적에 대한 자세한 내용은 [SOA 환경에서 트랜잭션 추적 사용](#) (페이지 101)을 참조하십시오.

참고: 추적 구성에 대한 자세한 내용은 *CA APM Java Agent* 구현 안내서 또는 *CA APM .NET* 에이전트 구현 안내서를 참조하십시오. 트랜잭션 추적 뷰와 기록 데이터 작업에 대한 자세한 내용은 *CA APM Workstation* 사용자 안내서를 참조하십시오.

제 12 장: WebSphere Process Server 및 WESB 모니터링

WPS(WebSphere Process Server)는 SCA(Service Component Architecture)를 사용하여 기존 응용 프로그램을 통합하고 새 응용 프로그램을 개발 및 배포할 수 있도록 도와주는 메시징 및 데이터 관리 서비스를 제공하는 포괄적인 SOA(Service-Oriented Architecture) 통합 플랫폼입니다. SOA Extension for WPS(WebSphere Process Server)를 사용하면 WESB(WebSphere Enterprise Service Bus), 비즈니스 서비스 구성 요소 및 지원 서비스 같은 WebSphere Process Server 의 주요 구성 요소를 모니터링 할 수 있습니다.

이 단원에서는 WebSphere Process Server 또는 WebSphere Enterprise Service Bus 환경의 성능, 가용성 및 전반적인 건전성의 모니터링 및 분석에 사용할 수 있는 WPS 및 WESB 관련 대시보드, 메트릭 및 경고에 대해 설명합니다.

이 섹션은 다음 항목을 포함하고 있습니다.

[WebSphere Process Server 및 WESB 정보](#) (페이지 329)

[WebSphere Process Server 또는 WESB 모니터링을 사용하도록 설정하는 방법](#) (페이지 334)

[WPS 또는 WESB 모니터링에 대시보드 사용](#) (페이지 342)

[WPS/WESB에 대한 메트릭 보기 및 탐색](#) (페이지 347)

[기본 WPS 및 WESB 메트릭 그룹화 보기](#) (페이지 355)

[기본 WPS 및 WESB 경고 보기](#) (페이지 356)

[WPS 또는 WESB 종속성 보기](#) (페이지 357)

[WPS 또는 WESB에 대한 트랜잭션 추적](#) (페이지 358)

WebSphere Process Server 및 WESB 정보

WPS(WebSphere Process Server)는 포괄적인 SOA(Service-Oriented Architecture) 통합 플랫폼으로, 이를 통해 새 응용 프로그램과 레거시 응용 프로그램을 통합하고 느슨하게 결합된 서비스를 배포하여 비즈니스 목표를 실현 할 수 있습니다.

WebSphere Process Server 아키텍처의 구성 요소는 다음과 같습니다.

- 서비스 구성 요소
- 지원 서비스
- WebSphere Enterprise Service Bus 메시징

WebSphere Process Server의 서비스 구성 요소에는 비즈니스 프로세스, 비즈니스 상태 컴퓨터, 비즈니스 규칙 및 수동 작업이 포함됩니다.

WebSphere Process Server의 지원 서비스에는 조정 흐름, 인터페이스 맵, 비즈니스 개체 맵, Java 구성 요소, SIB(Service Integration Bus) 통신, 비즈니스 프로세스, 비즈니스 상태 컴퓨터, 관계, 선택기 및 어댑터가 포함됩니다.

WESB(WebSphere Enterprise Service Bus)는 WebSphere Process Server를 통한 메시지의 흐름을 관리하고 서비스와 클라이언트 사이에서 필요한 모든 데이터 변환이나 라우팅을 처리합니다.

SOA Extension for WebSphere Process Server를 사용하면 WebSphere Enterprise Service Bus를 비롯한 전체 WebSphere Process Server 아키텍처를 모니터링할 수 있습니다. 또는 WebSphere Enterprise Service Bus만 모니터링할 수 있습니다. WebSphere Enterprise Service Bus를 독립 실행형 제품으로 설치할 수 있습니다.

WebSphere Process Server를 모니터링하는 경우에는 대시보드와 메트릭에 WPS 및 WESB 모두에 대한 정보가 표시됩니다. WESB를 독립 실행형 제품으로 모니터링하는 경우에는 대시보드와 메트릭에 WESB 관련 정보만 포함됩니다.

WebSphere Process Server 구성 요소 모니터링

WebSphere Process Server를 사용하는 경우 다음과 같은 메트릭을 사용하여 WebSphere Process Server 환경의 작업을 모니터링할 수 있습니다.

비즈니스 개체 맵

비즈니스 개체 맵은 원본 비즈니스 개체 서비스 구성 요소의 값을 기반으로 대상 비즈니스 개체 서비스 구성 요소에 값을 할당합니다. 한 비즈니스 개체는 원본이 되고 다른 비즈니스 개체는 대상이 됩니다. 비즈니스 개체 맵은 원본과 대상을 매핑합니다.

SOA Extension for WPS에서는 "WProcServer" > "BO 맵" 노드 아래에서 비즈니스 개체 맵의 성능 및 전반적인 건전성을 모니터링할 수 있습니다.

비즈니스 프로세스

비즈니스 프로세스는 특정 목표를 달성하기 위해 특정 순서로 실행되는 개별 작업으로 구성됩니다. 비즈니스 프로세스는 요청에 대한 응답에서 메시지를 받는 것과 관련된 하나 이상의 작업으로 구성됩니다.

SOA Extension for WPS 에서는 "WProcServer" > "비즈니스 프로세스" 노드 아래에서 비즈니스 프로세스 작업의 성능 및 전반적인 건전성을 모니터링 할 수 있습니다.

비즈니스 상태 컴퓨터

비즈니스 상태 컴퓨터는 이벤트 중심 응용 프로그램으로, 응용 프로그램은 이벤트가 발생할 때까지 대기하고 있다가 이벤트를 기준으로 실행할 적절한 작업을 선택합니다. 작업 실행 후 컴퓨터는 다른 상태로 전환되거나 또 다른 이벤트가 발생할 때까지 대기할 수 있습니다.

SOA Extension for WPS 에서는 "WProcServer" > "비즈니스 상태 컴퓨터" 노드 아래에서 이벤트 중심 프로세스의 성능 및 전반적인 건전성을 모니터링 할 수 있습니다.

비즈니스 규칙

비즈니스 규칙은 비즈니스 정책 및 관례를 캡처하고 구현합니다. 비즈니스 규칙으로 비즈니스 정책을 적용하거나, 의사 결정을 하거나, 기준 데이터에서 새 데이터를 유추할 수 있습니다.

SOA Extension for WPS 에서는 "WProcServer" > "비즈니스 규칙" 노드 아래에서 정의한 비즈니스 규칙의 성능 및 전반적인 건전성을 모니터링 할 수 있습니다.

데이터 바인딩

데이터 바인딩은 인바운드 및 아웃바운드 처리 중에 비즈니스 개체를 데이터 스트림으로 변환하는 데 사용됩니다.

SOA Extension for WPS 에서는 "WProcServer" > "데이터 바인딩" 노드 아래에서 데이터 바인딩의 성능 및 전반적인 건전성을 모니터링 할 수 있습니다.

수동 작업

수동 작업은 사람에 의해 수행되지만 어떤 방식으로든 WPS 의 프로세스 또는 서비스와 상호 작용하는 작업을 나타냅니다.

SOA Extension for WPS 에서는 "WProcServer" > "수동 작업" 노드 아래에서 사용자 상호 작용이 필요한 작업의 성능 및 전반적인 건전성을 모니터링 할 수 있습니다.

인터페이스 맵

인터페이스 맵은 인터페이스가 서로 다른 구성 요소들이 통신할 수 있도록 변환 및 기타 기초 작업을 사용하여 인터페이스가 서로 다른 구성 요소 간의 차이점을 조정합니다.

SOA Extension for WPS 에서는 "WProcServer" > "인터페이스 맵" 노드 아래에서 인터페이스 맵의 성능 및 전반적인 건전성을 모니터링할 수 있습니다.

J2CA(어댑터)

어댑터는 WebSphere Process Server 가 외부 EIS(Enterprise Information System)와 통신할 수 있도록 해주는 응용 프로그램입니다. WebSphere Process Server 가 EIS 특정 작업을 호출할 때 아웃바운드 통신이 수행됩니다. 응용 프로그램이 특정 EIS 이벤트를 수신 대기할 때 인바운드 통신이 수행됩니다.

SOA Extension for WPS 에서는 "WProcServer" > "J2CA" 노드 아래에서 어댑터의 성능 및 전반적인 건전성을 모니터링할 수 있습니다.

Java 구성 요소

Java 구성 요소를 사용하면 WPS 또는 WESB 의 Java 구현을 사용자 지정할 수 있습니다.

SOA Extension for WPS 에서는 "WProcServer" > "Java 구성 요소" 노드 아래에서 Java 구성 요소의 성능 및 전반적인 건전성을 모니터링할 수 있습니다.

조정 흐름

조정 흐름은 요청 처리에 대한 요청 흐름, 응답 처리에 대한 응답 흐름, 이벤트 처리에 대한 이벤트 흐름 및 결합 처리에 대한 결합 흐름으로 구성됩니다. 요청 흐름은 각 원본 작업에 대해 시작되며 응답 흐름은 각 대상 작업에 대해 시작됩니다.

SOA Extension for WPS 에서는 "WProcServer" > "WESB" > "조정 흐름" 노드 아래에서 조정 흐름의 성능 및 전반적인 건전성을 모니터링할 수 있습니다.

조정 기본

조정 기본은 조정 흐름 구성 요소의 기능을 구현하는 구성 단위입니다. 각 조정 흐름에는 데이터를 변환할 수 있는 조정 기본이 포함됩니다.

SOA Extension for WPS 에서는 "WProcServer" > "WESB" > "조정 기본" 노드 아래에서 조정 기본의 성능 및 전반적인 건전성을 모니터링할 수 있습니다.

관계

관계는 둘 이상의 데이터 형식에서 데이터 간의 연결을 설정합니다. 예를 들어 고객과 고객 ID 간의 상관 관계를 정의하는 데 관계를 사용할 수 있습니다.

SOA Extension for WPS에서는 "WProcServer" > "관계" 노드 아래에서 관계 정의의 성능 및 전반적인 건전성을 모니터링할 수 있습니다.

선택기

선택기를 사용하면 런타임에서 서비스 구성 요소를 유연하게 처리할 수 있습니다. 선택기를 사용하면 클라이언트 응용 프로그램의 호출 하나를 받아 런타임에 선택 기준을 바탕으로 서로 다른 대상을 호출할 수 있습니다.

SOA Extension for WPS에서는 "WProcServer" > "선택기" 노드 아래에서, 정의한 선택기의 성능 및 전반적인 건전성을 모니터링할 수 있습니다.

SIB-Communication(SIB 통신)

비즈니스 프로세스 및 비즈니스 상태 컴퓨터의 실행을 제어하는 Process Server Controller 구성 요소의 SIB(Service Integration Bus) 메트릭을 표시합니다. 이 메트릭은 "WProcServer" > "SIB-Communication"(SIB 통신) 노드 아래에서 모니터링할 수 있습니다.

독립 실행형 WebSphere Enterprise Service Bus 모니터링

WebSphere Enterprise Service Bus를 독립 실행형 제품으로 사용하고 있다면 다음과 같은 메트릭으로 WebSphere Enterprise Service Bus 환경의 작업을 모니터링할 수 있습니다.

비즈니스 개체 맵

독립 실행형 WESB 환경에서는 "WESB" > "BO 맵" 노드 아래에서 비즈니스 개체 맵의 성능 및 전반적인 건전성을 모니터링할 수 있습니다.

데이터 바인딩

독립 실행형 WESB 환경에서는 "WESB" > "데이터 바인딩" 노드 아래에서 데이터 바인딩의 성능 및 전반적인 건전성을 모니터링할 수 있습니다.

J2CA(어댑터)

독립 실행형 WESB 환경에서는 "WESB" > "J2CA" 노드 아래에서 어댑터의 성능 및 전반적인 건전성을 모니터링할 수 있습니다.

Java 구성 요소

독립 실행형 WESB 환경에서는 "WESB" > "Java 구성 요소" 노드 아래에서 Java 구성 요소의 성능 및 전반적인 건전성을 모니터링할 수 있습니다.

조정 흐름

독립 실행형 WESB 환경에서는 "WESB" > "조정 흐름" 노드 아래에서 조정 흐름의 성능 및 전반적인 건전성을 모니터링할 수 있습니다.

조정 기본

독립 실행형 WESB 환경에서는 "WESB" > "조정 기본" 노드 아래에서 조정 기본의 성능 및 전반적인 건전성을 모니터링할 수 있습니다.

관계

독립 실행형 WESB 환경에서는 "WESB" > "관계" 노드 아래에서 관계 정의의 성능 및 전반적인 건전성을 모니터링할 수 있습니다.

WebSphere Process Server 또는 WESB 모니터링을 사용하도록 설정하는 방법

WebSphere Process Server 또는 WebSphere Enterprise Service Bus 의 모니터링을 사용하도록 설정하는 단계는 다음과 같습니다.

1. 지원되는 버전의 WebSphere Process Server 또는 WebSphere Enterprise Service Bus 가 설치되어 있는지 확인합니다.

참고: 시스템 요구 사항에 대해서는 *Compatibility Guide*(호환성 안내서)를 참조하십시오.

2. 에이전트 및 CA APM for SOA 가 설치되고 사용되도록 설정되었는지 확인합니다.

3. [에이전트 프로필을 구성](#) (페이지 351)하여 CA APM for WebSphere Process Server 또는 WebSphere Enterprise Service Bus 를 사용하도록 에이전트를 설정합니다. WebSphere Process Server 또는 WebSphere Enterprise Service Bus 를 모니터링하기 위한 적절한 ProbeBuilder 지시문 파일을 포함합니다.

4. Enterprise Manager extension for WebSphere Process Server 또는 Enterprise Manager extension for WebSphere Enterprise Service Bus [를 사용하도록 설정합니다.](#) (페이지 339) `<EM_Home>/examples/SOAEExtensionForWPSandWESB` 디렉터리의 파일을 적절한 Enterprise Manager 디렉터리로 이동합니다.

WPS 또는 WESB 모니터링을 위하여 에이전트를 사용하도록 설정

에이전트를 설치하고 WebSphere 를 응용 프로그램 서버로 선택하는 경우 또는 에이전트를 설치한 다음 수동으로 WebSphere Enterprise Service Bus 에 대한 모니터링 기능을 포함하는 WebSphere Process Server 모니터링을 사용하도록 설정할 수 있습니다.

에이전트를 설치할 때 CA APM for WebSphere Process Server 및 WESB 를 선택한 경우 WebSphere Process Server 및 WebSphere Enterprise Service Bus 를 모두 모니터링하기 위한 파일을 `<Agent_Home>` 디렉터리에서 사용할 수 있습니다.

에이전트를 설치한 후에는 에이전트 프로필을 구성하여 수동으로 CA APM for WebSphere Process Server 또는 CA APM for WebSphere Enterprise Service Bus 를 사용하도록 설정해야 합니다.

수동으로 CA APM for WPS 또는 WESB 를 사용하도록 설정하려면

1. 에이전트 및 CA APM for SOA 가 설치되고 사용하도록 설정되었는지 확인합니다.
2. CA APM for WebSphere Process Server 디렉터리 `SOAExtensionForWPSandWESB` 가 `<Agent_Home>/examples` 디렉터리에 있는지 확인하고 `<Agent_Home>/examples/SOAExtensionForWPSandWESB/ext` 디렉터리의 파일을 `<Agent_Home>/core/ext` 디렉터리로 복사합니다.
3. `<Agent_Home>/core/config/IntroscopeAgent.profile` 파일을 텍스트 편집기에서 엽니다.
4. 적절한 지시문 파일을 `IntroscopeAgent.profile` 파일의 `introscope.autoprobe.directivesFile` 속성에 추가합니다.
 - WebSphere Enterprise Service Bus 를 포함하여 모든 WebSphere Process Server 를 모니터링하려면 `wps.pbd` 를 추가합니다.
 - 독립 실행형 환경에서 WebSphere Enterprise Service Bus 구성 요소만 모니터링하려면 `wesb.pbd` 를 추가합니다.

기본 모니터링을 수정하려는 경우에는 `wps.pbd` 또는 `wesb.pbd` 파일에서 `ProbeBuilder` 지시문을 사용자 지정할 수 있습니다. 그런 후에는 예를 들어 특정 추적 프로그램 그룹에 대한 추적을 설정하거나 해제하여 특정 구성 요소에 대한 모니터링을 세부 조정할 수 있습니다.

5. `IntroscopeAgent.profile` 파일을 저장하고 닫습니다.

Business Process Manager 8.0 및 8.1 의 필수 구성

Business Process Manager 8.0 및 8.1에서 추적을 사용하려면 wps.pbd를 구성하십시오.

다음 단계를 수행하십시오.

1. 실행 중인 경우 비즈니스 프로세스 관리자를 중지합니다.
2. <Agent_Home>/core/config 디렉터리로 이동하여 “wps.pbd” 파일을 업니다.

참고: 레거시 모드를 사용하는 경우 동일한 절차를 통해 “wps-legacy.pbd” 파일을 수정하십시오.

3. pbd 파일에서 다음과 같은 추적 프로그램 매핑 및 추적 프로그램 지시문을 주석 처리합니다.

```
#SetTracerClassMapping: ThreadCreationTracer
com.wily.powerpack.webspheredprocserver.tracer.hc2.ThreadCreationTracer
com.wily.introscope.probebuilder.validate.ResourceNameValidator

#TraceOneMethodWithParametersOfClass:
com.ibm.bpe.framework.navigation.NavigationWorker doWork ThreadCreationTracer
"Navigation-Work"
```

4. pbd 파일에서 다음과 같은 추적 프로그램 매핑 및 추적 프로그램 지시문의 주석 처리를 해제합니다.

```
SetTracerClassMapping: ThreadCreationTracer80
com.wily.powerpack.webspheredprocserver.tracer.hc2.ThreadCreationTracer80
com.wily.introscope.probebuilder.validate.ResourceNameValidator

TraceOneMethodWithParametersOfClass:
com.ibm.bpe.framework.navigation.NavigationWorker doWork
ThreadCreationTracer80 "Navigation-Work"
```

5. wps.pbd를 저장하고 닫습니다.

기본 구성의 수정이 완료되면 비즈니스 프로세스 관리자를 다시 시작하십시오.

여러 버전의 비즈니스 프로세스 및 비즈니스 상태 컴퓨터에 대한 메트릭 수집

여러 버전의 비즈니스 프로세스 및 비즈니스 상태 컴퓨터에 대해 별도의 메트릭을 제공하도록 wps.pbd를 구성할 수 있습니다. 각 비즈니스 프로세스 및 비즈니스 상태 컴퓨터의 버전을 하나씩만 유지하려면 이 구성 단계를 건너뛸 수 있습니다.

다음 단계를 수행하십시오.

1. <Agent_Home>/core/config 디렉터리로 이동하여 wps.pbd 파일을 업니다.
2. ProcessCustomTracer 및 ProcessFaultTracer 매개 변수를 주석 처리합니다.
예:

```
##SetTracerParameter: ProcessCustomTracer nameformatter
com.wily.powerpack.webspheredprocserver.nameformatter.ProcessContextFormatter
##SetTracerParameter: ProcessFaultTracer nameformatter
com.wily.powerpack.webspheredprocserver.nameformatter.ProcessContextFormatter
##SetTracerParameter: BPMMapTracer nameformatter
com.wily.powerpack.webspheredprocserver.nameformatter.ProcessContextFormatter
##SetTracerParameter: BusinessProcessTracer nameformatter
com.wily.powerpack.webspheredprocserver.nameformatter.BusinessProcessNameAndTi
meFormatter
##SetTracerParameter: BusinessProcessFaultTracer nameformatter
com.wily.powerpack.webspheredprocserver.nameformatter.BusinessProcessNameAndTi
meFormatter
##SetTracerParameter: BPMMap7Tracer nameformatter
com.wily.powerpack.webspheredprocserver.nameformatter.BusinessProcessNameAndTi
meFormatter
```

3. ProcessCustomTracer 및 ProcessFaultTracer 매개 변수의 주석 처리를 제거하여 시간 특성을 가져옵니다. 예:

```
SetTracerParameter: ProcessCustomTracer nameformatter  
com.wily.powerpack.webspheredprocserver.nameformatter.ProcessAndTimeNameFormat  
ter  
SetTracerParameter: ProcessFaultTracer nameformatter  
com.wily.powerpack.webspheredprocserver.nameformatter.ProcessAndTimeNameFormat  
ter  
SetTracerParameter: BPMMapTracer nameformatter  
com.wily.powerpack.webspheredprocserver.nameformatter.ProcessAndTimeNameFormat  
ter  
SetTracerParameter: BusinessProcessTracer nameformatter  
com.wily.powerpack.webspheredprocserver.nameformatter.BusinessProcessNameFormat  
ter  
SetTracerParameter: BusinessProcessFaultTracer nameformatter  
com.wily.powerpack.webspheredprocserver.nameformatter.BusinessProcessNameFormat  
ter  
SetTracerParameter: BPMMap7Tracer nameformatter  
com.wily.powerpack.webspheredprocserver.nameformatter.BusinessProcessNameFormat  
ter
```

참고: 이 단계를 건너뛰면 파일 개발 시 사용한 시간 형식으로 "파일
비즈니스 프로세스" 및 "비즈니스 상태 컴퓨터" 노드에 추가됩니다.

4. wps.pbd 를 저장하고 닫습니다.

기본 구성의 수정이 끝나면 WebSphere Process Server 또는 WESB 서버를
다시 시작할 수 있습니다.

PMI(Performance Monitoring Infrastructure) 메트릭 수집

분산 환경에서 WebSphere Application Server 를 모니터링 중인 경우에는
WebSphere PMI(Performance Monitoring Infrastructure) 메트릭을 수집하고
WebSphere Process Server 구성 요소에 대한 해당 메트릭을 Investigator 에서
볼 수 있습니다.

이렇게 하려면 먼저 *CA APM for IBM WebSphere Distributed Environments*
안내서의 설명에 따라 모니터링을 구성해야 합니다. 그런 다음 응용
프로그램 서버에서 추가 애이전트 속성을 수정하여 WebSphere Process
Server 구성 요소의 모니터링을 사용하도록 설정할 수 있습니다.

다음 단계를 수행하십시오.

1. <Agent_Home>/core/config 의 *IntroscopeAgent.profile* 파일을 엽니다.
2. 다음 속성을 찾습니다.
introscope.agent.pmi.enable.WBISStats.RootGroup
introscope.agent.pmi.enable.SCASStats.RootGroup

3. 이 두 속성을 모두 true 로 설정하여 PMI 메트릭 보고를 사용하도록 설정합니다. 예:

```
introscope.agent.pmi.enable.WBISStats.RootGroup=true  
introscope.agent.pmi.enable.SCAStats.RootGroup=true
```

이 속성이 현재 정의되어 있지 않으면 수동으로 *IntroscopeAgent.profile*에 속성을 추가한 다음 true 로 설정합니다.

4. *IntroscopeAgent.profile* 파일을 저장하고 닫습니다.

기본 구성의 수정을 완료했으면 WebSphere Process Server 또는 WESB 서버를 다시 시작할 수 있습니다.

5. WebSphere Admin 콘솔을 사용하여 Performance Monitoring Infrastructure for WebSphere Process Server 및 WebSphere Enterprise Service Bus 모듈을 사용하도록 설정합니다.

참고: WebSphere Admin 콘솔을 사용하여 모듈을 사용하도록 설정하는데 대한 자세한 내용은 IBM WebSphere 문서를 참조하십시오.

Enterprise Manager Extension for WPS 또는 WESB 를 사용하도록 설정

CA APM for WebSphere Process Server 및 WebSphere Enterprise Service Bus 파일은 기본적으로 Enterprise Manager 를 설치할 때 <EM_Home>/examples 디렉터리에 설치됩니다. CA APM for WebSphere Process Server 또는 WebSphere Enterprise Service Bus 를 사용하도록 설정하려면 Enterprise Manager 파일을 <EM_Home>/examples 디렉터리에서 Enterprise Manager 홈 디렉터리의 적절한 위치로 복사하거나 이동하십시오.

참고: CA APM for WebSphere Process Server 또는 WebSphere Enterprise Service Bus 를 사용하려면 먼저 CA APM for SOA 를 [Enterprise Manager에서 사용하도록 설정](#) (페이지 41)해야 합니다.

Enterprise Manager extension for WebSphere Process Server 를 사용하도록 설정하려면

1. CA APM for WebSphere Process Server 디렉터리
SOAExtensionForWPSandWESB 가 <EM_Home>/examples 디렉터리에 있는지 확인합니다.
<EM_Home>/examples/SOAExtensionForWPSandWESB 디렉터리의 파일을 Enterprise Manager 디렉터리 구조의 해당 위치로 복사합니다. 예를 들어 <EM_Home>/examples/SOAExtensionForWPSandWESB/ext 디렉터리의 파일을 <EM_Home>/ext 디렉터리로 복사합니다.
2. *WPS_Management_Module.jar* 파일을 <EM_HOME>/examples/SOAExtensionForWPSandWESB/config/modules 디렉터리에서 복사하여 <EM_Home>/config/modules 디렉터리에 붙여 넣습니다.
<EM_HOME>/examples/SOAExtensionForWPSandWESB/config/modules 디렉터리에는 WebSphere Process Server 및 WebSphere Enterprise Service Bus 독립 실행형 모두에 대한 관리 모듈이 있습니다. WebSphere Enterprise Service Bus 를 사용하는 WebSphere Process Server 를 모니터링하는 경우에만 *WPS_Management_Module.jar* 파일을 복사하십시오.
클러스터 환경에 여러 개의 Enterprise Manager 가 있는 경우 이 파일만 MOM 컴퓨터로 사용 중인 Enterprise Manager 의 <EM_Home>/config/modules 디렉터리에 복사하십시오. 다른 모든 파일과 스크립트는 Collector Enterprise Manager 와 MOM Enterprise Manager 모두에 설치해야 합니다.
3. 이전 버전의 CA APM for WebSphere Process Server 에서 업그레이드할 때는 이전 버전의 Enterprise Manager 파일을 제거하십시오.
Enterprise Manager 에 이전 릴리스를 설치한 경우 새 버전의 CA APM for WebSphere Process Server 를 사용하기 전에 이전 버전의 Enterprise Manager 파일을 수동으로 삭제하십시오.
이전 릴리스에서 업그레이드한 경우 Enterprise Manager 홈 디렉터리에서 다음 파일을 삭제하십시오.
 - <EM_home>/config/modules/WPS_Management_ModuleV<version>.jar
 - <EM_home>/product/enterprisemanager/plugins/com.wily.powerpack.webspherereprocserver.em.ext_<version>.jar

예를 들어 버전 8.1에서 업그레이드하는 경우에는 다음 파일을 삭제하십시오.

- WPS_Management_ModuleV8.1.0.0.jar
- com.wily.powerpack.websphereprocserver.em.ext_8.1.0.jar

4. Workstation을 다시 시작합니다.

SOA extension for WebSphere Process Server 또는 WebSphere Enterprise Service Bus에 특정한 대시보드 및 "개요" 탭이 로드됩니다.

Enterprise Manager extension for WESB standalone을 사용하도록 설정하려면

1. CA APM for WebSphere Process Server 디렉터리 `SOAExtensionForWPSandWESB` 가 `<EM_Home>/examples` 디렉터리에 있는지 확인합니다.
`<EM_Home>/examples/SOAExtensionForWPSandWESB` 디렉터리의 파일을 Enterprise Manager 디렉터리 구조의 해당 위치로 복사합니다. 예를 들어 `<EM_Home>/examples/SOAExtensionForWPSandWESB/ext` 디렉터리의 파일을 `<EM_Home>/ext` 디렉터리로 복사합니다.
2. `WESB_Management_Module.jar` 파일을 `<EM_HOME>/examples/SOAExtensionForWPSandWESB/config/modules` 디렉터리에서 복사하여 `<EM_Home>/config/modules` 디렉터리에 붙여 넣습니다.
`<EM_HOME>/examples/SOAExtensionForWPSandWESB/config/modules` 디렉터리에는 WebSphere Process Server 및 WebSphere Enterprise Service Bus 독립 실행형 모두에 대한 관리 모듈이 있습니다. WebSphere Enterprise Service Bus 독립 실행형을 모니터링하는 경우에만 `WESB_Management_Module.jar` 파일을 복사하십시오.
클러스터 환경에 여러 개의 Enterprise Manager가 있는 경우 이 파일만 MOM 컴퓨터로 사용 중인 Enterprise Manager의 `<EM_Home>/config/modules` 디렉터리에 복사하십시오. 다른 모든 파일과 스크립트는 Collector Enterprise Manager와 MOM Enterprise Manager 모두에 설치해야 합니다.
3. 이전 버전의 CA APM for WebSphere Enterprise Service Bus에서 업그레이드할 때는 이전 버전의 Enterprise Manager 파일을 제거하십시오.

Enterprise Manager에 이전 릴리스를 설치한 경우 새 버전의 CA APM for WESB를 사용하기 전에 이전 버전의 Enterprise Manager 파일을 직접 삭제하십시오. 이전 릴리스에서 업그레이드한 경우 Enterprise Manager 홈 디렉터리에서 다음 파일을 삭제하십시오.

- <EM_home>/config/modules/WESB_Management_ModuleV<version>.jar
- <EM_home>/product/enterprisemanager/plugins/com.wily.powerpack.websphereprocserver.em.ext_<version>.jar

예를 들어 버전 8.1에서 업그레이드하는 경우에는 다음 파일을 삭제하십시오.

- WESB_Management_ModuleV8.1.0.0.jar
- com.wily.powerpack.websphereprocserver.em.ext_8.1.0.jar

4. Workstation을 다시 시작합니다.

CA APM for WebSphere Process Server 또는 WebSphere Enterprise Service Bus에 특정한 대시보드 및 "개요" 탭이 로드됩니다.

WPS 또는 WESB 모니터링에 대시보드 사용

SOA Extension for WebSphere Process Server에는 응용 프로그램 환경의 전반적인 건전성을 모니터링하는 데 사용할 수 있는 몇 개의 사전 구성된 대시보드가 포함되어 있습니다. 대시보드는 배포된 에이전트에서 데이터를 집계하여 성능 정보를 요약하며 문제를 신속하게 진단하고 해결할 수 있도록 해 줍니다.

일반적으로 대시보드는 다음과 같은 작업을 가능하게 하므로 환경을 모니터링하는 시작점으로 사용됩니다.

- WebSphere Process Server의 주요 구성 요소에 대한 전반적인 건전성, 성능, 가용성 및 현재 상태를 한눈에 모니터링
- 하위 수준 메트릭이 주의 또는 위험 임계값이 초과되었음을 알리는 경우 프로덕션 응용 프로그램 환경의 잠재적 문제를 조기에 식별
- 성능 정보를 드릴다운하여 지연 중이거나 오류가 발생 중인 WebSphere 비즈니스 프로세스, 비즈니스 규칙, 조정 흐름 또는 기타 구성 요소를 격리 및 식별

사전 구성된 WebSphere Process Server 및 WebSphere Enterprise Service Bus 대시보드는 SOA Extension for WebSphere Process Server 관리 모듈(*WPS_Management_Module.jar*) 또는 SOA Extension for WebSphere Enterprise Service Bus 관리 모듈(*WESB_Management_Module.jar*)의 일부로 Enterprise Manager Extension for WebSphere에 패키지되어 있습니다.

WebSphere Process Server 대시보드 정보

WebSphere Process Server 관리 모듈은 WebSphere Process Server에 대해 다음과 같은 사전 구성된 대시보드를 제공합니다.

WPS - 개요

WebSphere Process Server의 건전성에 대한 최상위 수준 개요로, WebSphere Process Server 아키텍처의 모든 주요 구성 요소에 대한 전체 응답 시간, 오류, 중단 등을 포함합니다.

WPS - 비즈니스 프로세스 & 상태 컴퓨터

모든 비즈니스 프로세스 및 비즈니스 상태 컴퓨터에 대한 요약 상태로, 모든 비즈니스 프로세스 및 상태 컴퓨터에 대한 평균 응답 시간 그래프, 오류 및 중단에 대한 경고 표시기, 가장 느린 비즈니스 프로세스 및 상태 컴퓨터의 목록을 포함합니다.

WPS - 비즈니스 규칙 및 수동 작업

모든 비즈니스 규칙 및 수동 작업에 대한 요약 상태로, 모든 비즈니스 규칙 수동 작업에 대한 평균 응답 시간 그래프, 오류 및 중단에 대한 경고 표시기, 가장 느린 비즈니스 규칙 및 수동 작업의 목록을 포함합니다.

WPS - 인터페이스 맵, BO 맵 및 관계

모든 인터페이스 맵, 비즈니스 개체 맵 및 관계에 대한 요약 상태로, 인터페이스 맵, 비즈니스 개체 맵 및 관계에 대한 평균 응답 시간 그래프, 오류 및 중단에 대한 경고 표시기, 가장 느린 인터페이스 맵, 비즈니스 개체 맵 및 관계의 목록을 포함합니다.

WPS - 선택기 및 Java 구성 요소

모든 선택기 및 Java 구성 요소에 대한 요약 상태로, 모든 선택기 및 Java 구성 요소에 대한 평균 응답 시간 그래프, 오류 및 중단에 대한 경고 표시기, 가장 느린 선택기 및 Java 구성 요소의 목록을 포함합니다.

WPS - 데이터 바인딩

모든 데이터 바인딩에 대한 요약 상태로, 평균 응답 시간, 간격당 오류 수 및 중단 수에 대한 그래프 및 경고 표시기, 가장 느린 데이터 바인딩 목록을 포함합니다.

WPS - 어댑터

모든 인바운드 및 아웃바운드 어댑터에 대한 요약 상태로, 모든 인바운드 및 아웃바운드 어댑터에 대한 평균 응답 시간 그래프, 오류 및 중단에 대한 경고 표시기, 가장 느린 인바운드 및 아웃바운드 어댑터의 목록을 포함합니다.

WESB - 조정 흐름

조정 흐름의 전반적인 건전성에 대한 요약 상태와 요청 흐름, 응답 흐름 및 결합 흐름에 대한 별도의 그래프 및 경고 표시기입니다.

조정 흐름은 요청, 응답 및 결합 흐름으로 구성되므로 이 대시보드에는 조정 흐름, 요청 흐름, 응답 흐름 및 결합 흐름에 대한 평균 응답 시간 그래프, 조정 흐름, 요청 흐름, 응답 흐름 및 결합 흐름의 오류 및 중단에 대한 경고 표시기, 가장 느린 조정 흐름 목록이 표시됩니다.

WESB - 조정 기본

조정 기본의 전반적인 건전성에 대한 요약 상태로, 모든 조정 기본의 응답 시간 및 오류에 대한 경고 표시기와 요청 흐름, 응답 흐름 및 결합 흐름의 응답 시간, 오류 및 중단에 대한 별도의 경고 표시기를 포함합니다.

WESB 대시보드 정보

WebSphere Enterprise Service Bus 를 독립 실행형 제품으로 모니터링하는 경우 다음과 같은 대시보드만 사용할 수 있습니다.

WESB - 개요

WebSphere Enterprise Service Bus 의 건전성에 대한 최상위 수준 개요로, WebSphere 아키텍처에서 WebSphere Enterprise Service Bus 계층의 모든 주요 구성 요소에 대한 전체 응답 시간, 오류, 중단 등을 포함합니다.

WESB - 조정 흐름

WebSphere Enterprise Service Bus 에 대한 조정 흐름의 전반적인 건전성에 대한 요약 상태로, 요청 흐름, 응답 흐름 및 결합 흐름에 대한 별도의 그래프 및 경고 표시기를 포함합니다.

WESB - 조정 기본

WebSphere Enterprise Service Bus에 대한 조정 기본의 전반적인 건전성에 대한 요약 상태로, 모든 조정 기본의 응답 시간 및 오류에 대한 경고 표시기와 요청 흐름, 응답 흐름 및 결함 흐름의 응답 시간, 오류 및 중단에 대한 별도의 경고 표시기를 포함합니다.

WESB - 어댑터

WebSphere Enterprise Service Bus에 대한 모든 인바운드 및 아웃바운드 어댑터에 대한 요약 상태로, 모든 인바운드 및 아웃바운드 어댑터에 대한 평균 응답 시간 그래프, 오류 및 중단에 대한 경고 표시기, 가장 느린 인바운드 및 아웃바운드 어댑터의 목록을 포함합니다.

WESB - BO 맵 및 관계

WebSphere Enterprise Service Bus에 대한 모든 비즈니스 개체 맵 및 관계에 대한 요약 상태로, 모든 비즈니스 개체 맵 및 관계에 대한 평균 응답 시간 그래프, 오류 및 중단에 대한 경고 표시기, 가장 느린 비즈니스 개체 맵 및 관계의 목록을 포함합니다.

WESB - 데이터 바인딩

WebSphere Enterprise Service Bus에 대한 모든 데이터 바인딩에 대한 요약 상태로, 평균 응답 시간, 간격당 오류 수 및 중단 수에 대한 그래프 및 경고 표시기, 가장 느린 데이터 바인딩 목록을 포함합니다.

WESB - Java 구성 요소

WebSphere Enterprise Service Bus에 대한 모든 Java 구성 요소에 대한 요약 상태로, 모든 Java 구성 요소에 대한 평균 응답 시간, 간격당 오류 수 및 중단 수에 대한 그래프 및 경고 표시기, 가장 느린 Java 구성 요소의 목록을 포함합니다.

WebSphere Enterprise Service Bus의 대시보드는 WebSphere Process Server에서 제공되는 대시보드와 유사하지만 WebSphere Enterprise Service Bus 구성 요소에 한정된 경고 표시기 및 메트릭만 제공합니다.

WebSphere Process Server 또는 WESB 대시보드 보기

WebSphere Process Server 또는 WebSphere Enterprise Service Bus에 대한 대시보드에서는 시스템 건전성을 한눈에 평가하는 데 도움이 되는 경고 표시기를 제공합니다.

다음 단계를 수행하십시오.

1. Enterprise Manager가 실행되고 있지 않으면 시작합니다.
2. Workstation을 시작하고 SOA extension for WebSphere Process Server 또는 WESB가 설치된 Enterprise Manager에 로그인합니다.
3. "Workstation" > "새 콘솔"을 클릭합니다.
4. "대시보드" 드롭다운 목록에서 WPS 또는 WESB 대시보드 중 하나를 선택합니다.

예를 들어 모든 서비스 구성 요소 및 지원 서비스에 대한 경고 표시기를 포함한 WebSphere Process Server 건전성의 개요를 보려면 **WPS - 개요** 대시보드를 선택합니다.

"WPS - 개요" 또는 "WESB - 개요" 대시보드에서 보고자 하는 대시보드의 이름이 지정된 탭을 두 번 클릭하면 다른 대시보드로 이동할 수 있습니다.

5. 관련 대시보드를 열어서 더 자세한 정보를 보려면 대시보드의 다른 탭이나 경고를 두 번 클릭합니다.

예를 들어 "비즈니스 프로세스"에 대한 "응답 시간" 경고를 두 번 클릭하면 **WPS - 비즈니스 프로세스 및 상태 컴퓨터** 대시보드에서 가장 많은 시간을 점유하고 있는 특정 비즈니스 프로세스에 대한 자세한 정보를 볼 수 있습니다.

"WPS - 비즈니스 프로세스 및 상태 컴퓨터" 대시보드에는 가장 느린 비즈니스 프로세스 및 상태 컴퓨터의 목록이 응답 시간 그래프, 오류에 대한 경고 표시기 및 중단 수와 함께 표시됩니다.

6. "WPS - 비즈니스 프로세스 및 상태 컴퓨터" 대시보드에서 "가장 느린 비즈니스 프로세스"의 목록 중 특정 비즈니스 프로세스 이름을 두 번 클릭하면 해당 비즈니스 프로세스의 응답 시간이 추가 분석을 위해 선택되어 있는 **Investigator**를 열 수 있습니다.

참고: Workstation의 시작과 사용, 대시보드 액세스 또는 Investigator의 열기와 탐색에 대한 자세한 내용은 *CA APM Workstation 사용자 안내서*를 참조하십시오.

WPS/WESB 에 대한 메트릭 보기 및 탐색

Investigator 트리에서 탐색 할 때는 WebSphere Process Server 또는 WebSphere Enterprise Service Bus 인프라의 구성 요소 대부분에 대한 표준 CA Introscope 메트릭을 볼 수 있습니다. 표준 메트릭에 대한 데이터는 Investigator 트리에서 노드 및 하위 노드로 표시되는 WebSphere Process Server 또는 WESB 특정 메트릭 범주로 수집되고 집계됩니다. 표시되는 특정 메트릭 범주 및 노드 이름은 환경에서 배포하고 액세스한 구성 요소, 서비스 및 리소스에 따라 다릅니다.

Investigator 트리를 탐색 할 때는 선택한 노드에 따라 개별 작업에 대한 하위 수준 메트릭 표시나 집계 메트릭 표시를 선택할 수 있으므로 여러 Process Server 또는 WESB 구성 요소의 전반적인 건전성을 모니터링 할 수 있습니다. 프로세스에 대해 정의된 수동 작업이 없는 경우처럼 구성 요소가 전혀 사용되지 않는 경우에는 구성 요소에 대한 메트릭 범주가 Investigator 트리에 표시되지 않습니다.

Investigator 에서 메트릭 요약을 보고 탐색하려면

1. 에이전트 노드를 확장한 다음 WProcServer 또는 WESB 노드를 클릭하여 요약 정보에 대한 "개요" 탭을 표시합니다.
예를 들어 "WProcServer" 노드를 선택한 경우 WebSphere Process Server 가 사용하고 있는 모든 비즈니스 프로세스, 비즈니스 상태 컴퓨터 및 조정 흐름에 대한 요약 정보가 "개요"에 나열됩니다.
2. 비즈니스 프로세스, 비즈니스 상태 컴퓨터 또는 조정 흐름을 두 번 클릭하면 연관된 모든 메트릭이 그래픽 형식으로 표시됩니다.
3. WProcServer 또는 WESB 노드를 확장하면 최상위 수준 WebSphere Process Server 또는 WebSphere Enterprise Service Bus 메트릭 범주에 대한 하위 노드가 표시됩니다.
4. 하위 노드를 클릭하거나 확장하여 해당 메트릭 범주의 요약 정보가 있는 "개요" 탭을 표시합니다. 예를 들어 "비즈니스 규칙" 노드를 클릭하여 비즈니스 규칙의 목록을 "개요" 탭에 표시합니다.
5. 하위 노드를 확장하면 개별 구성 요소에 대한 세부 정보를 볼 수 있습니다. 예를 들어 특정 비즈니스 프로세스 이름을 선택하면 해당 비즈니스 프로세스에 대한 메트릭이 그래픽 형식으로 표시됩니다.
6. 개별 개체를 확장하면 해당 개체에 대한 메트릭이 표시됩니다. 예를 들어 개별 Java 구성 요소 이름을 확장한 다음 작업을 확장하여 해당 작업에 대한 메트릭을 표시합니다.

비즈니스 개체 맵에 대한 메트릭

비즈니스 개체 맵은 원본 비즈니스 개체 서비스 구성 요소의 값을 기반으로 대상 비즈니스 개체 서비스 구성 요소에 값을 할당합니다.

"WProcServer" > "BOMaps" 또는 "WESB" > "BOMaps" 노드 아래에서 대상 네임스페이스 및 개별 비즈니스 개체 맵에 대해 모든 표준 CA Introscope 메트릭을 사용할 수 있습니다.

비즈니스 프로세스에 대한 메트릭

비즈니스 프로세스는 특정 비즈니스 목표를 달성하기 위해 특정 순서로 실행되는 개별 작업으로 구성됩니다. 이 메트릭 범주는 WebSphere Process Server에만 적용 가능합니다. WebSphere Enterprise Service Bus를 독립 실행형 제품으로서 모니터링하는 경우에는 적합하지 않습니다.

"WProcServer" > "비즈니스 프로세스" > <*business_process_name*>" 노드 아래에서 개별 WebSphere 비즈니스 프로세스 및 비즈니스 프로세스 단계에 대해 모든 표준 CA Introscope 메트릭을 사용할 수 있습니다.

여러 버전의 비즈니스 프로세스에 대한 메트릭 수집 (페이지 337)의 설명에 따라 동일한 비즈니스 프로세스의 여러 버전에 대해 개별적인 메트릭을 표시하도록 추적을 구성한 경우 "비즈니스 프로세스" 노드 아래의 비즈니스 프로세스 이름에는 동일한 프로세스의 여러 버전을 구분하기 위해 밑줄[_]과 타임스탬프가 추가됩니다.

비즈니스 규칙에 대한 메트릭

비즈니스 규칙은 비즈니스 정책 및 관례를 캡처하고 구현합니다. 이 메트릭 범주는 WebSphere Process Server에만 적용 가능합니다. WebSphere Enterprise Service Bus를 독립 실행형 제품으로서 모니터링하는 경우에는 적합하지 않습니다.

"WProcServer" > "비즈니스 규칙" > <*rule_name*>" 노드 아래에서 배포한 개별 비즈니스 규칙에 대한 모든 표준 CA Introscope 메트릭을 사용할 수 있습니다.

비즈니스 상태 컴퓨터에 대한 메트릭

비즈니스 상태 컴퓨터는 이벤트에 대한 응답으로 작업을 실행하는 이벤트 중심 응용 프로그램입니다. 이 메트릭 범주는 WebSphere Process Server에만 적용 가능합니다. 이 범주는 WebSphere Enterprise Service Bus를 독립 실행형 제품으로 모니터링하는 경우에는 적용할 수 없습니다.

"WProcServer" > "비즈니스 상태 컴퓨터" > "*<business_state_machine_name>*" 노드 아래에서 개별 비즈니스 상태 컴퓨터에 대해 모든 표준 CA Introscope 메트릭을 사용할 수 있습니다.

동일한 비즈니스 상태 컴퓨터의 각 버전마다 별도의 메트릭을 표시하도록 추적을 구성 (페이지 337)한 경우 "비즈니스 상태 컴퓨터" 노드 아래의 이름은 다음과 같이 표시됩니다.

- 이름에 밑줄[_]이 추가됩니다.
- 타임스탬프를 통해 동일한 프로세스의 여러 버전이 식별됩니다.

데이터 바인딩에 대한 메트릭

데이터 바인딩은 인바운드 및 아웃바운드 처리 중에 비즈니스 개체를 데이터 스트림으로 변환하는 데 사용됩니다.

"WProcServer" > "데이터 바인딩" > "*<data_binding_name>*" 또는 "WESB" > "데이터 바인딩" > "*<data_binding_name>*" 노드 아래에서 개별 데이터 바인딩에 대해 모든 표준 CA Introscope 메트릭을 사용할 수 있습니다.

수동 작업에 대한 메트릭

수동 작업은 사람에 의해 수행되지만 WebSphere Process Server의 프로세스 또는 서비스와 상호 작용하는 작업을 나타냅니다. 이 메트릭 범주는 WebSphere Process Server에만 적용 가능합니다. WebSphere Enterprise Service Bus를 독립 실행형 제품으로서 모니터링하는 경우에는 적합하지 않습니다.

"WProcServer" > "수동 작업" 노드 아래에서 개별 수동 작업에 대해 모든 표준 CA Introscope 메트릭을 사용할 수 있습니다.

인터페이스 맵에 대한 메트릭

인터페이스 맵은 인터페이스가 서로 다른 구성 요소가 통신할 수 있도록 구성 요소 간의 차이를 조정합니다. 이 메트릭 범주는 WebSphere Process Server에만 적용 가능합니다. WebSphere Enterprise Service Bus를 독립 실행형 제품으로서 모니터링하는 경우에는 적합하지 않습니다.

"WProcServer" > "인터페이스 맵" 노드 아래에서 정의된 인터페이스 맵에 대해 모든 표준 CA Introscope 메트릭을 사용할 수 있습니다.

동기적 호출의 경우 개별 인터페이스 맵에 대한 메트릭은 인터페이스 맵 구성 요소에 소요된 총 시간을 나타냅니다. 비동기적 호출의 경우 메트릭은 입력 및 출력 변환을 개별적으로 나타냅니다.

J2CA 어댑터에 대한 메트릭

어댑터는 WebSphere Process Server가 외부 Enterprise Information System과 통신할 수 있도록 해주는 응용 프로그램입니다. WebSphere Process Server가 EIS 특정 작업을 호출할 때 아웃바운드 통신이 수행됩니다. 응용 프로그램이 특정 EIS 이벤트를 수신 대기할 때 인바운드 통신이 수행됩니다.

"WProcServer" > "J2CA" 또는 "WESB" > "J2CA" 노드 아래에서 WebSphere 서버와 외부 시스템 간의 인바운드 및 아웃바운드에 대해 모든 표준 CA Introscope 메트릭을 사용할 수 있습니다.

Java 구성 요소에 대한 메트릭

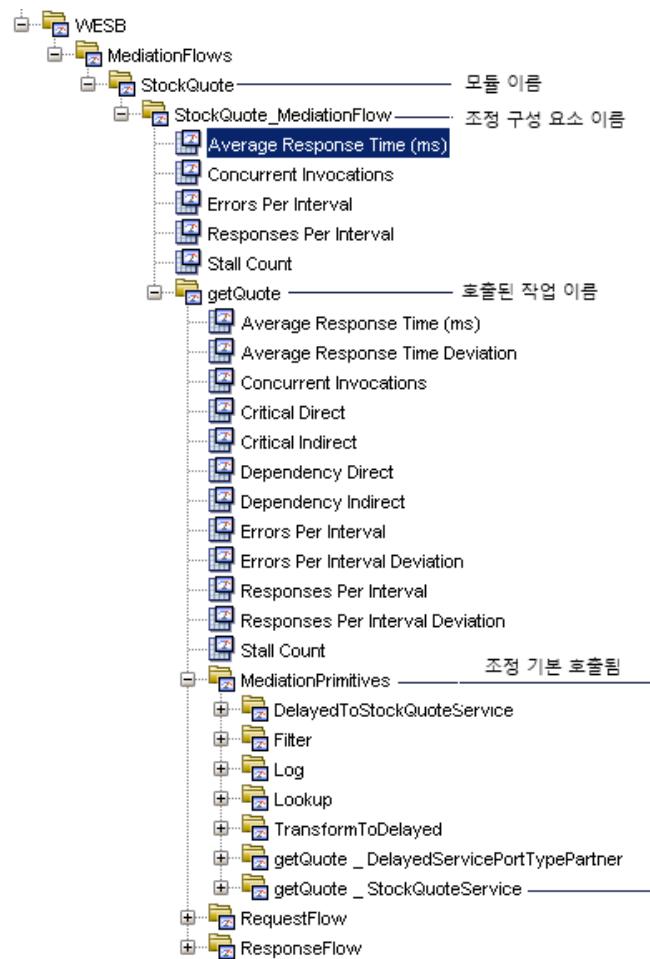
Java 구성 요소를 사용하면 WPS 또는 WESB의 Java 구현을 사용자 지정할 수 있습니다.

"WProcServer" > "Java 구성 요소" 또는 "WESB" > "Java 구성 요소" 노드 아래에서 개별 Java 구성 요소 및 작업에 대해 모든 표준 CA Introscope 메트릭을 사용할 수 있습니다.

조정 흐름 및 조정 기본에 대한 메트릭

조정 흐름은 요청 처리에 대한 요청 흐름, 응답 처리에 대한 응답 흐름, 이벤트 처리에 대한 이벤트 흐름 및 결합 처리에 대한 결합 흐름으로 구성됩니다. 요청 흐름은 각 원본 작업에 대해 시작되며 응답 흐름은 각 대상 작업에 대해 시작됩니다. 조정 기본은 조정 흐름의 기능을 구현하는 구성 단위입니다. 각 조정 흐름에는 필요한 경우 데이터를 변환할 수 있는 조정 기본이 포함됩니다.

"WProcServer" > "WESB" > "조정 흐름" 또는 "WESB" > "조정 흐름" 노드 아래에서 조정 흐름과 조정 기본에 대해 모든 표준 CA Introscope 메트릭을 사용할 수 있습니다. "조정 흐름" 노드는 조정 흐름 구성 요소가 있는 모듈을 나열합니다. 그런 다음 조정 흐름 구성 요소를 확장하여 조정 흐름과 해당 작업, 흐름 및 조정 기본에 대한 집계 메트릭을 볼 수 있습니다. 예:



응답, 이벤트 또는 결합 흐름이 비동기적 호출로 호출된 경우 노드 이름은 Investigator에 *<flow_type>_Asynch*로 표시됩니다. 여기서 *<flow_type>*은 ResponseFlow, EventFlow 또는 FaultFlow의 흐름 유형을 나타냅니다. 예를 들어 응답 흐름이 콜백을 통해 비동기적으로 호출된 경우 Investigator에 표시되는 노드 이름은 ResponseFlow_Asynch입니다.

조정 흐름 메트릭의 컨텍스트에서 비동기적 호출은 콜백 함수를 사용하여 비동기적으로 호출되는 작업입니다.

동기 및 비동기 응답 시간 메트릭 정보

조정 흐름 구성 요소에 대한 메트릭은 흐름 이름 아래에 작업에 대한 모든 "평균 응답 시간" 메트릭에서 집계됩니다. 조정 흐름 작업에 대한 메트릭은 작업이 동기 호출을 사용하는지 아니면 비동기 호출을 사용하는지에 따라 다르게 집계됩니다.

- 작업이 동기 호출을 사용하는 경우 요청 흐름이 동기 호출 중의 응답 흐름을 간접적으로 포함하기 때문에 작업 수준 메트릭은 작업의 요청 흐름에서 집계됩니다.
- 작업이 비동기 호출을 사용하는 경우 작업 수준 메트릭은 작업 이름 아래에 기존 요청 흐름(RequestFlow)과 비동기 응답(ResponseFlow_Asynch), 결합(FaultFlow_Asynch) 및 이벤트 흐름(EventFlow_Asynch)에서 집계됩니다.

예를 들어 평균 응답 시간이 요청 흐름의 경우 10 ms, 비동기 응답 흐름의 경우 12 ms, 비동기 결합 흐름의 경우 8 ms, 비동기 이벤트 흐름의 경우 10 ms이고 개수 값이 요청 흐름의 경우 1, 응답 흐름의 경우 1, 결합 흐름의 경우 2, 이벤트의 흐름의 경우 3이라면 작업의 평균 응답 시간은 다음과 같이 계산됩니다.

$$((10 \times 1) + (12 \times 1) + (8 \times 2) + (10 \times 3))/3$$

동기 및 비동기 동시 호출 정보

조정 흐름 구성 요소에 대한 메트릭은 흐름 이름 아래에 작업에 대한 모든 "동시 호출" 메트릭에서 집계됩니다. 조정 흐름 작업에 대한 메트릭은 작업이 동기 호출을 사용하는지 아니면 비동기 호출을 사용하는지에 따라 다르게 집계됩니다.

- 작업이 동기 호출을 사용하는 경우 요청 흐름은 동기 호출 중의 응답 흐름을 간접적으로 포함하므로 작업 수준의 "동시 호출" 메트릭은 작업에 대한 요청 흐름의 동시 호출을 나타냅니다.
- 작업이 비동기 호출을 사용하는 경우 작업 수준 메트릭은 작업 이름 아래에 기존 요청 흐름(RequestFlow)과 비동기 응답(ResponseFlow_Asynch), 결함(FaultFlow_Asynch) 및 이벤트 흐름(EventFlow_Asynch)의 동시 호출에서 집계됩니다.

동기 및 비동기 오류 메트릭 정보

조정 흐름 구성 요소에 대한 메트릭은 흐름 이름 아래에 작업에 대한 모든 "간격당 오류 수" 메트릭에서 집계됩니다. 조정 흐름 작업에 대한 "간격당 오류 수" 메트릭은 작업이 동기 호출을 사용하는지 아니면 비동기 호출을 사용하는지에 따라 다르게 계산됩니다.

- 작업이 동기 호출을 사용하는 경우 작업 수준 메트릭은 요청 흐름의 "간격당 오류 수" 메트릭을 더하여 집계됩니다.
- 작업이 비동기 호출을 사용하는 경우 작업 수준 메트릭은 작업 이름 아래에 기존 요청 흐름(RequestFlow)과 비동기 응답(ResponseFlow_Asynch) 및 이벤트(EventFlow_Asynch) 흐름의 기존 "간격당 오류 수" 메트릭을 비동기 결함(FaultFlow_Asynch)의 "간격당 응답 수" 메트릭에 더하여 집계됩니다.

결함이 발생할 때 결함 흐름이 발생한다고 예상되므로 비동기 결함 흐름에 대한 "간격당 응답 수" 메트릭 값은 오류가 해당 작업의 호출 중에 발생했음을 나타내기 위해 작업 수준의 "간격당 오류 수" 메트릭 값에 더해집니다.

동기 및 비동기 응답 메트릭 정보

조정 흐름 구성 요소에 대한 메트릭은 흐름 이름 아래에 작업에 대한 모든 "간격당 응답 수" 메트릭에서 집계됩니다. 조정 흐름 작업에 대한 "간격당 응답 수" 메트릭은 작업이 동기 호출을 사용하는지 아니면 비동기 호출을 사용하는지에 따라 다르게 계산됩니다.

- 작업이 동기 호출을 사용하는 경우 작업 수준 메트릭은 요청 흐름의 "간격당 응답 수"를 더하여 집계됩니다.
- 작업이 비동기 호출을 사용하는 경우 "간격당 응답 수" 메트릭은 작업 이름 아래에 있는 요청 흐름(RequestFlow)과 비동기 응답(ResponseFlow_Asynch), 결함(FaultFlow_Asynch) 및 이벤트 흐름(EventFlow_Asynch)의 최대 "간격당 응답 수"로 결정됩니다.

동기 및 비동기 중단 수 메트릭 정보

조정 흐름 구성 요소에 대한 메트릭은 흐름 이름 아래에 작업에 대한 모든 "중단 수" 메트릭에서 집계됩니다. 조정 흐름 작업에 대한 "중단 수" 메트릭은 작업이 동기 호출을 사용하는지 아니면 비동기 호출을 사용하는지에 따라 다르게 계산됩니다.

- 작업이 동기 호출을 사용하는 경우 "중단 수"는 요청 흐름에서 기록된 중단 수를 모두 더하여 계산됩니다.
- 작업이 비동기 호출을 사용하는 경우 "중단 수"는 작업 이름 아래에 있는 요청 흐름과 비동기 응답(ResponseFlow_Asynch), 결함(FaultFlow_Asynch) 및 이벤트 흐름(EventFlow_Asynch)으로 기록된 중단 수를 더하여 계산됩니다.

관계에 대한 메트릭

관계는 의미적으로 동일한 비즈니스 개체를 나타냅니다.

"WProcServer" > "관계" 또는 "WESB" > "관계" 노드 아래에서 개별 관계에 대해 모든 표준 CA Introscope 메트릭을 사용할 수 있습니다.

선택기에 대한 메트릭

선택기를 사용하면 런타임에서 서비스 구성 요소를 유연하게 처리할 수 있습니다. 이 메트릭 범주는 WebSphere Process Server에만 적용 가능합니다. WebSphere Enterprise Service Bus를 독립 실행형 제품으로서 모니터링하는 경우에는 적합하지 않습니다.

"WProcServer" > "선택기" 노드 아래에서 모든 표준 CA Introscope 메트릭을 사용할 수 있습니다.

SIB(Service Integration Bus) 통신에 대한 메트릭

SIB(Service Integration Bus)는 메시지 기반 및 SOA(Service-Oriented Architecture)를 사용하는 응용 프로그램을 지원합니다. SIB 통신 메트릭은 "WProcServer" > "SIB-Communication"(SIB 통신) 노드 아래에서 볼 수 있습니다.

WebSphere Process Server 결함에 대한 메트릭

WPS 계층에서 결함이 발생할 때 결함이 처리되지 않으면 예외가 throw 됩니다. "WPS Faults Per Interval"(간격당 WPS 결함 수) 메트릭은 발생한 예외의 수를 15초 시간 간격으로 표시합니다. 이 메트릭은 WebSphere Process Server에만 적용 가능합니다. WebSphere Enterprise Service Bus를 독립 실행형 제품으로서 모니터링하는 경우에는 적합하지 않습니다.

기본 WPS 및 WESB 메트릭 그룹화 보기

SOA extension for WebSphere Process Server 및 WESB에는 기본 대시보드와 경고를 정의하는 데 사용되는 기본 메트릭 그룹화가 포함되어 있습니다. 사용자 지정 대시보드와 경고에서도 이 기본 메트릭 그룹화를 사용할 수 있습니다.

기본 메트릭 그룹화는 SOA Extension for WebSphere Process Server 관리 모듈(*WPS_Management_Module.jar*) 또는 SOA Extension for WebSphere Enterprise Service Bus 관리 모듈(*WESB_Management_Module.jar*)의 일부로 Enterprise Manager 확장에 패키지되어 있습니다.

기본 WebSphere Process Server 또는 WESB 메트릭 그룹화를 보려면

1. Investigator에서 Workstation > 새 관리 모듈 편집기를 클릭합니다.
2. 모든 WebSphere Process Server 구성 요소를 모니터링하는 경우에는 "*SuperDomain*" > "관리 모듈" > "WPS_ManagementModule(*SuperDomain*)"을 확장하고 독립 실행형 WESB 서버를 모니터링하는 경우에는 "WESB_ManagementModule (*SuperDomain*)"을 확장합니다.
3. "메트릭 그룹화" 노드를 확장하면 모든 WebSphere Process Server 구성 요소를 모니터링하는 경우 WebSphere Process Server에 대한 모든 기본 메트릭 그룹화가 표시되고 독립 실행형 WESB 서버를 모니터링하는 경우 WebSphere Enterprise Service Bus에 대한 기본 메트릭 그룹화가 표시됩니다.
4. 특정 메트릭 그룹화를 클릭하여 "뷰어" 창에서 해당 정의를 봅니다.
메트릭 그룹화의 기본 설정을 수정하거나 사용자 지정 메트릭 그룹화를 직접 생성할 수 있습니다.

참고: 메트릭 그룹화 생성 또는 수정에 대한 자세한 내용은 *CA APM Workstation 사용자 안내서*를 참조하십시오.

기본 WPS 및 WESB 경고 보기

SOA Extension for WebSphere Process Server 및 WESB에는 사전 구성된 대시보드에 사용되는 기본 경고 정의가 포함됩니다. 이 기본 경고를 사용자 지정 대시보드에서도 사용할 수 있습니다. 대부분의 기본 경고는 기본 "주의" 및 "위험" 임계값으로 사전 구성되며 임계값이 초과되거나 심각도가 증가할 경우 콘솔에 알림을 보내도록 사전 구성됩니다.

기본 경고 정의는 SOA Extension for WebSphere Process Server 관리 모듈(*WPS_Management_Module.jar*) 또는 SOA Extension for WebSphere Enterprise Service Bus 관리 모듈(*WESB_Management_Module.jar*)의 일부로 Enterprise Manager 확장에 패키지되어 있습니다.

WebSphere Process Server 또는 WESB 에이전트의 기본 경고 정의를 보려면

1. Investigator에서 Workstation > 새 관리 모듈 편집기를 클릭합니다.
2. 모든 WebSphere Process Server 구성 요소를 모니터링하는 경우에는 "*SuperDomain*" > "관리 모듈" > "WPS_ManagementModule (*SuperDomain*)"을 확장하고, 독립 실행형 WESB 서버를 모니터링하는 경우에는 "WESB_ManagementModule (*SuperDomain*)"을 확장합니다.
3. "메트릭 그룹화" 노드를 확장하여 WebSphere Process Server 관리 모듈에 대해 정의된 모든 메트릭 그룹화를 표시합니다.
4. 특정 경고를 클릭하여 "뷰어" 창에서 해당 정의를 봅니다.

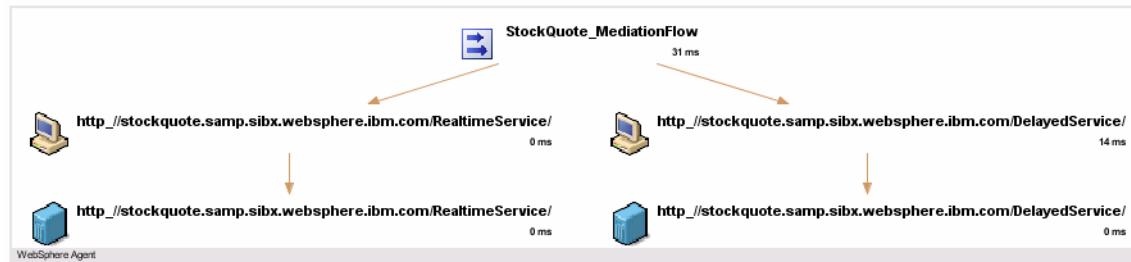
특히 기본 "주의" 및 "위험" 임계값과 중요 경고에 대한 사전 정의된 작업을 검토하고 환경에 맞게 조정해야 합니다. 예를 들어 필요한 경우 임계값을 조정하거나, 알림을 추가하거나, 수정 조치를 정의해야 할 수 있습니다.

WPS 또는 WESB 종속성 보기

트리의 노드를 사용하여 WebSphere Process Server에서 "비즈니스 프로세스", "비즈니스 상태 컴퓨터" 또는 "조정 흐름"을 선택하거나 WebSphere Enterprise Service Bus에서 "조정 흐름"을 선택한 다음 "SOA 종속성 맵" 탭을 클릭하여 비즈니스 프로세스, 비즈니스 상태 컴퓨터, 조정 흐름 및 어댑터 아웃바운드 구성 요소에 대한 종속성을 표시할 수 있습니다. EIS 작업을 호출하는 구성 요소를 선택하면 종속성 맵에서 어댑터 아웃바운드 종속성도 확인할 수 있습니다. 예를 들어 조정 흐름이 어댑터를 사용하여 JDBC 테이블의 행을 검색하는 경우 조정 흐름을 선택하고 해당 종속성을 표시하여 종속성 맵에 조정 흐름과 아웃바운드 어댑터 간의 종속성을 표시할 수 있습니다.

선택한 노드에 따라 종속성 맵에 표시되는 컨텍스트가 결정됩니다. 종속성 맵이 표시되면 보고자 하는 컨텍스트와 세부 정보 수준을 룰업하여 축소하거나 롤다운하여 확장할 수 있습니다. 예를 들어 비즈니스 프로세스의 상위 수준 종속성을 표시하려면 Investigator에서 비즈니스 프로세스 이름을 선택한 다음 "SOA 종속성 맵" 탭을 클릭합니다.

다음 예에서는 **Investigator** 트리에서 선택한 **Stock Quote** 비즈니스 서비스의 조정 흐름과 추가 종속성을 표시하도록 확장된 종속성 맵을 보여 줍니다.



계속해서 필요에 따라 맵에 종속성 수준을 추가하여 비즈니스 프로세스의 전체 작업 흐름을 표시하거나 맵의 특정 노드를 확대할 수 있습니다. 종속성 맵의 탐색에 대한 자세한 내용은 [SOA 종속성 맵 사용](#) (페이지 73)을 참조하십시오.

WPS 또는 WESB에 대한 트랜잭션 추적

트랜잭션 추적은 비즈니스 트랜잭션 완료에 관련된 특정 단계의 세부 정보 뷔 또는 요약 뷔를 제공합니다. WebSphere Process Server 또는 WebSphere Enterprise Service Bus 구성 요소나 웹 서비스가 포함된 트랜잭션의 경우 다음 프로토콜을 통해 라우팅되는 작업이 포함된 트랜잭션을 추적할 수 있습니다.

- SOAP(Simple Object Access Protocol)
- HTTP(Hypertext Transport Protocol)
- HTTPS(Hypertext Transport Protocol Secure)
- JMS(Java Message Service)

WebSphere Process Server 또는 WebSphere Enterprise Service Bus 구성 요소 외부에서 시작된 트랜잭션에는 WebSphere Process Server 또는 WebSphere Enterprise Service Bus 환경에서 시작된 비동기적 호출이 포함될 수 있습니다. 트랜잭션에 참가하는 모든 구성 요소에서 이러한 트랜잭션을 추적하기 위해 상관 관계 식별자가 삽입되고 각 구성 요소 및 작업에서 트랜잭션 단계로 사용됩니다. 그런 후에는 트랜잭션의 세부 정보를 드릴다운하여 수행된 작업과 각 작업의 완료에 걸린 시간을 포함하여 트랜잭션에 관여한 WebSphere Process Server 또는 WebSphere Enterprise Service Bus 구성 요소를 이해할 수 있습니다.

또한 추적 중인 모든 노드에서 CA APM for SOA 및 CA APM for WebSphere Process Server 가 사용하도록 설정되어 있으면 모든 조합의 플랫폼에서 비즈니스 트랜잭션을 추적할 수 있습니다. 따라서 트랜잭션이 여러 JVM 또는 CLR에서 실행되는 경우에도 트랜잭션에 대한 세부 정보를 볼 수 있습니다.

크로스 프로세스 트랜잭션 추적의 가치 이해

크로스 프로세스 트랜잭션 추적은 SOA(서비스 지향 아키텍처)에서 느슨하게 결합된 서비스가 수행하는 작업에 대한 중요한 정보를 제공합니다. 크로스 프로세스 트랜잭션 추적을 사용하면 다음을 확인할 수 있습니다.

- WebSphere Enterprise Service Bus 구성 요소를 통해 메시지가 라우팅되는 방식
- 트랜잭션 중에 호출 및 실행되는 구성 요소와 작업
- 트랜잭션 중에 발생하는 호출의 순서
- 요청 또는 응답 처리가 가장 느린 지점

샘플 트랜잭션 추적 시작 및 보기

트랜잭션 추적 세션은 다음과 같이 시작할 수 있습니다.

- SOA 종속성 맵의 맵 노드에서 바로 시작합니다.
- "Workstation" > "새 트랜잭션 추적 세션"을 클릭하여 Workstation에서 수동으로 시작합니다.

종속성 맵에서 트랜잭션 추적을 시작하는 경우에는 맵 노드 유형에 따라 기본 필터가 자동으로 설정됩니다. 새 트랜잭션 추적 세션을 수동으로 시작하는 경우에는 WebSphere Process Server에 대한 다음 필터 유형 중 하나를 선택할 수 있습니다.

- 어댑터 노드
- 비즈니스 프로세스
- 비즈니스 상태 컴퓨터
- 조정 흐름
- 조정 흐름 작업

예를 들어 특정 조정 흐름에 대한 트랜잭션을 필터링하려면 조정 흐름 필터를 선택하고 조정 흐름 이름의 전체 또는 일부를 입력하십시오.

필터를 구성하고 트랜잭션 추적 세션을 시작하면 "트랜잭션 추적 뷰어"가 표시됩니다. 추적을 선택하여 트랜잭션에서 생성된 호출에 대한 추가 세부 정보를 볼 수 있습니다. 여기에는 WebSphere Process Server를 사용하여 실행된 모든 비즈니스 프로세스, 상태 변경 사항 또는 Java 구성 요소가 포함됩니다.

WebSphere 프로세스에 대한 시퀀스 뷰 사용

WebSphere Process Server가 관련된 트랜잭션에는 비동기 호출이 포함되는 경우가 많으므로 "시퀀스 뷰"를 클릭하여 트랜잭션의 일부로 비동기적으로 실행된 프로세스에 대한 트랜잭션 작업 흐름을 확인하면 유용합니다. "시퀀스 뷰"에는 프로세스가 실행되는 순서가 시퀀스를 식별할 수 있는 범위까지 표시됩니다. WebSphere Process Server 트랜잭션의 경우 시퀀스가 반드시 전통적인 호출자-피호출자 관계를 나타내는 것은 아니지만 하나의 프로세스가 다른 프로세스의 실행을 트리거하는 시점을 보여 줍니다.

하지만 WebSphere Process Server 및 WESB 프로세스의 처리 시간은 호출된 프로세스와 연관된 처리 시간을 포함하여 프로세스의 시작 시점부터 완료 시점까지의 전체 기간을 사용하여 계산됩니다. 호출 프로세스의 기간에서 비차단 동기적 및 비동기적 프로세스의 처리 시간을 뺀 순 기간은 WebSphere Process Server 및 WESB 프로세스에 대해 지원되지 않습니다.

트랜잭션 추적에 대한 자세한 내용은 [SOA 환경에서 트랜잭션 추적 사용](#) (페이지 101)을 참조하십시오.

참고: 추적 구성에 대한 자세한 내용은 *CA APM Java Agent* 구현 안내서 또는 *CA APM .NET* 에이전트 구현 안내서를 참조하십시오. 트랜잭션 추적 뷰와 기록 데이터 작업에 대한 자세한 내용은 *CA APM Workstation* 사용자 안내서를 참조하십시오.

비즈니스 프로세스/비즈니스 상태 컴퓨터 동작이 포함되도록 추적 구성

기본적으로 WebSphere Process Server 비즈니스 프로세스/비즈니스 상태 컴퓨터에 대한 트랜잭션 추적에는 동작 수준까지의 성능 정보가 포함됩니다. 이러한 추적에는 동작 자체의 실행에 대한 추가적인 세부 정보는 포함되지 않습니다. 동작 이름을 검색하면 성능 오버헤드가 커지므로 추가적인 동작 수준 추적은 기본적으로 해제되어 있습니다.

동작 추적을 수동으로 켜서 비즈니스 프로세스/비즈니스 상태 컴퓨터에서 실행되는 동작에 대한 세부 정보를 볼 수 있습니다. 하지만 동작 추적은 제한된 시간 동안만 켜십시오.

다음 단계를 수행하십시오.

1. WebSphere Process Server 또는 WESB 응용 프로그램 서버를 중지합니다.
2. <Agent_Home> 디렉터리로 이동한 다음 **wps.pbd** 파일을 텍스트 편집기에서 엽니다.
3. **ActivityTracing** 및 **ActivityResponseTracing** 플래그의 주석 처리를 제거합니다.

```
TurnOn: ActivityTracing  
TurnOn: ActivityResponseTracing
```

추가적인 동작 수준 추적이 사용하도록 설정됩니다.

참고: 시간 특성을 가져오기 위한 [매개 변수를 사용하도록 설정](#) (페이지 337)한 경우에는 다음과 같이 **ActivityTracer**의 주석 처리를 제거합니다. 그러면 "동작" 노드가 별도의 비즈니스 프로세스/비즈니스 상태 컴퓨터 노드가 아니라 동일한 비즈니스 프로세스/비즈니스 상태 컴퓨터 노드 아래에 나타납니다.

```
#SetTracerParameter: ActivityTracer nameformatter  
com.wily.powerpack.webspherereprocserver.nameformatter.ActivityTimeNameformatte  
r
```

4. **wps.pbd** 를 저장하고 닫습니다.
5. WebSphere Process Server 또는 WESB 서버를 시작합니다.

에이전트가 새 구성으로 다시 시작됩니다.

부록 A: SOA 특정 에이전트 구성 속성

이 단원에서는 CA APM for SOA 구성에 사용되는 에이전트 속성을 설명합니다.

이 섹션은 다음 항목을 포함하고 있습니다.

[에이전트 속성의 위치 이해](#) (페이지 363)

[에이전트 속성 구성](#) (페이지 363)

[SOA 특정 에이전트 속성 정보](#) (페이지 364)

에이전트 속성의 위치 이해

CA APM for SOA 에이전트 속성은 에이전트 프로필 *IntroscopeAgent.profile*에 저장됩니다. 프로필은 속성 이름 및 값이 포함된 텍스트 파일입니다. 에이전트는 다음과 같은 방식으로 에이전트 프로필을 찾습니다.

- *com.wily.introscope.agentProfile* 시스템 속성으로 지정된 위치에서 찾습니다.
- *com.wily.introscope.agentProfile* 속성이 설정되지 않은 경우 에이전트는 *IntroscopeAgent.profile* 파일을 사용합니다.

대개의 경우 에이전트 속성의 변경 사항을 적용하려면 에이전트가 모니터링하는 응용 프로그램 서버를 다시 시작해야 합니다.

에이전트 속성 구성

에이전트 속성을 사용하면 에이전트의 동작 및 작동을 제어하고 환경에 맞게 설정을 사용자 지정할 수 있습니다. CA APM for SOA에는 기본 설정이 적용된 몇 개의 SOA 특정 에이전트 속성이 있습니다. 이러한 속성을 사용하려면 *IntroscopeAgent.profile* 파일에 속성 이름을 추가하고 속성에 올바른 값을 설정해야 합니다.

다음 단계를 수행하십시오.

1. *IntroscopeAgent.profile* 파일을 텍스트 편집기에서 엽니다.
2. *IntroscopeAgent.profile* 파일에 추가할 속성을 [SOA 특정 에이전트 속성](#) (페이지 364) 목록에서 식별합니다.

3. *com.wily.introscope* 접두사를 포함한 속성의 전체 이름을 *IntroscopeAgent.profile* 파일의 SOA 특정 섹션에 입력합니다. 예:
#####
SOA Performance Management Agent Settings
=====
com.wily.introscope.agent.httpheaderread.enabled
 4. 필요에 맞게 속성 값을 설정합니다. 예:
com.wicom.wily.introscope.agent.httpheaderread.enabled=true
 5. *IntroscopeAgent.profile* 파일을 저장하고 닫습니다.
- 속성을 *IntroscopeAgent.profile* 파일에 추가한 후에는 필요에 따라 속성 값을 변경할 수 있습니다.

SOA 특정 에이전트 속성 정보

*IntroscopeAgent.profile*에서 다음과 같은 SOA 특정 속성을 설정할 수 있습니다. 모든 속성은 *com.wily.introscope* 접두사로 시작하지만 읽기 편하도록 접두사 없이 나열되어 있습니다.

참고: 에이전트 속성에 대한 자세한 내용은 *CA APM Java Agent* 구현 안내서 또는 *CA APM .NET* 에이전트 구현 안내서를 참조하십시오.

[agent.httpheaderinsertion.enabled](#) (페이지 366)

HTTP 헤더에 클라이언트 측 상관 관계 정보를 삽입하거나 삽입하지 않도록 설정합니다.

[agent.httpheaderread.enabled](#) (페이지 367)

HTTP 헤더의 상관 관계 정보를 서버 측에서 검색하거나 검색하지 않도록 설정합니다.

[agent.soapheaderinsertion.enabled](#) (페이지 368)

SOAP 헤더에 클라이언트 측 상관 관계 정보를 삽입하거나 삽입하지 않도록 설정합니다.

[agent.soapheaderread.enabled](#) (페이지 369)

SOAP 헤더의 상관 관계 정보를 서버 측에서 검색하거나 검색하지 않도록 설정합니다.

[agent.soa.metricNameFormatting](#) (페이지 369)

CA APM for SOA 메트릭 이름을 수정하여 지정된 문자를 밑줄[_]로 바꿉니다.

agent.transactiontrace.boundaryTracing.cacheFlushFrequency (페이지 371)

캐시를 플러시하기 전에 에이전트 메모리에 종속성 데이터를 보관할 날짜 수를 지정합니다.

agent.transactiontrace.boundaryTracing.enable (페이지 372)

트랜잭션에 대한 경계 추적을 사용하거나 사용하지 않도록 설정합니다.

soa.client.prependhandler (페이지 373)

클라이언트 측에서 SOAP 헤더의 삽입 지점을 제어합니다.

soa.server.appendhandler (페이지 374)

서버 측에서 SOAP 헤더의 검색 및 제거를 제어합니다.

대개의 경우 에이전트 속성을 변경하고 응용 프로그램 서버를 다시 시작해야 속성 변경 사항이 적용됩니다.

agent.httpheaderinsertion.enabled

HTTP 헤더에 클라이언트 측 상관 관계 식별자를 삽입하거나 삽입하지 않도록 설정합니다.

SAP NetWeaver 에 대해서는 이 속성이 지원되지 않습니다.

속성 설정

속성은 true 또는 false 로 설정할 수 있습니다.

- True - 클라이언트 측 상관 관계 및 종속성 정보를 HTTP 헤더에 추가합니다.
- False - 클라이언트 측 상관 관계 및 종속성 정보에 대한 HTTP 헤더의 변경을 허용하지 않습니다.

기본값

CA APM for SOA, CA APM for WebSphere Process Server 및 CA APM for WebSphere Enterprise Service Bus 의 경우 False

CA APM for Oracle Service Bus, CA APM for TIBCO BusinessWork 및 CA APM for webMethods Integration Server 의 경우 True

예

```
com.wily.introscope.agent.httpheaderinsertion.enabled=true
```

참고

이 속성의 변경 내용을 적용하려면 관리되는 응용 프로그램을 다시 시작해야 합니다.

agent.httpheaderread.enabled

HTTP 헤더의 상관 관계 식별자를 서버 측에서 검색하거나 검색하지 않도록 설정합니다.

SAP NetWeaver에 대해서는 이 속성이 지원되지 않습니다.

속성 설정

속성은 true 또는 false로 설정할 수 있습니다.

- True - 서버 측 상관 관계 및 종속성 정보를 HTTP 헤더에서 읽습니다.
- False - HTTP 헤더에서 상관 관계 및 종속성 정보를 확인하지 않습니다.

기본값

CA APM for SOA, CA APM for WebSphere Process Server 및 CA APM for WebSphere Enterprise Service Bus의 경우 False

CA APM for Oracle Service Bus, CA APM for TIBCO BusinessWork 및 CA APM for webMethods Integration Server의 경우 True

예

```
com.wily.introscope.agent.httpheaderread.enabled=true
```

참고

이 속성의 변경 내용을 적용하려면 관리되는 응용 프로그램을 다시 시작해야 합니다.

agent.soapheaderinsertion.enabled

SOAP 헤더에 클라이언트 측 상관 관계 식별자를 삽입하거나 삽입하지 않도록 설정합니다.

SOA 종속성 맵 및 크로스 프로세스 트랜잭션을 추적하려면 한 프로세스에서 다른 프로세스로 상관 관계 식별자를 전달해야 합니다. 에이전트는 이 상관 관계 식별자를 SOAP 또는 HTTP 헤더에 삽입할 수 있습니다.

사용 중인 SOAP 기반 응용 프로그램이 예기치 않은 헤더 항목으로 인해 제대로 작동하지 않는 경우 이 속성을 `false`로 설정해야 합니다. 이 속성을 `false`로 설정하고 HTTP 헤더에 상관 관계 식별자를 삽입하도록 설정하지 않으면 SOA 종속성 맵이나 크로스 프로세스 트랜잭션 추적에 관련 정보를 표시할 수 없습니다.

속성 설정

속성은 `true` 또는 `false`로 설정할 수 있습니다.

- `True` - 클라이언트 측 상관 관계 및 종속성 정보를 SOAP 헤더에 추가합니다.
- `False` - 클라이언트 측 상관 관계 및 종속성 정보에 대한 SOAP 헤더의 변경을 허용하지 않습니다.

기본값

`True`

예

```
com.wily.introscope.agent.soapheaderinsertion.enabled=true
```

참고

이 속성의 변경 내용을 적용하려면 관리되는 응용 프로그램을 다시 시작해야 합니다.

agent.soapheaderread.enabled

SOAP 헤더의 상관 관계 식별자를 서버 측에서 검색하거나 검색하지 않도록 설정합니다.

속성 설정

속성은 true 또는 false로 설정할 수 있습니다.

- True - 서버 측 상관 관계 및 종속성 정보를 SOAP 헤더에서 읽습니다.
- False - SOAP 헤더에서 상관 관계 및 종속성 정보를 확인하지 않습니다.

기본값

True

예

com.wily.introscope.agent.soapheaderread.enabled=true

참고

이 속성의 변경 내용을 적용하려면 관리되는 응용 프로그램을 다시 시작해야 합니다.

agent.soa.metricNameFormatting

CA APM for SOA 메트릭 이름에서 지정한 모든 문자를 밑줄 문자[_]로 바꿉니다.

예를 들어 속성 com.wily.introscope.agent.soa.metricNameFormatting 을 설정하여 슬래시(/)를 바꾸면 URL
<http://CheckingAccount/demobank.ca.com> 은 다음과 같이 표시됩니다.
http_CheckingAccount_demobank.ca.com

메트릭 이름에서 지정한 문자를 밑줄(_)로 바꾸려면 이 속성을 설정하십시오.

메트릭 이름을 일부 Web Services Manager에 사용되는 형식으로 표시하려고 할 때 이 방법을 사용할 수 있습니다. 이 경우 webservices.pbd 파일에서 네임스페이스도 서비스 이름으로 바꿔야 합니다. 이렇게 변경하면 CA APM for SOA 메트릭 이름이 Investigator 트리와 "SOA 종속성 맵"에 네임스페이스 대신 서비스 이름을 사용하여 표시됩니다.

속성 설정

속성은 메트릭 이름에서 밑줄(_) 문자로 바꾸고자 하는 하나 이상의 문자로 설정할 수 있습니다.

속성을 정의하지 않거나 속성에 값이 없으면 문자 대체가 수행되지 않습니다.

기본값

없음

예

com.wily.introscope.agent.soa.metricNameFormatting=/:

참고

이 속성의 변경 내용을 적용하려면 관리되는 응용 프로그램을 다시 시작해야 합니다.

agent.transactiontrace.boundaryTracing.cacheFlushFrequency

캐시를 플러시하기 전에 에이전트 메모리에 종속성 데이터를 보관할 날짜 수를 지정합니다. 이 속성을 설정하여 주기적으로 유휴 스레드를 플러시하고 에이전트 캐시를 지울 수 있습니다. 지정한 기간이 끝나면 캐시 플러시의 결과로 "SOA 종속성 맵" 종속성 데이터가 에이전트에서 제거됩니다. 그러면 에이전트는 종속성 데이터를 다시 발견하고 데이터를 Enterprise Manager로 전달하여 SOA 종속성 맵에 표시된 정보를 새로고침합니다.

에이전트가 Enterprise Manager에 알려진 종속성에 대한 정보를 불필요하게 전송하지 않도록 이전에 검색된 종속성은 기본적으로 에이전트 캐시에 30일 동안 보관됩니다. 대개의 경우 종속성 데이터는 정적이며 이전에 발견된 종속성을 에이전트 캐시에 저장하면 에이전트가 Enterprise Manager에 정기적으로 전송해야 하는 데이터의 양을 줄일 수 있으므로 30일의 기간이 적절합니다.

실행 중인 에이전트와 독립 실행형 Enterprise Manager 사이의 연결이 끊기거나 클러스터되지 않은(MOM이 아닌) 환경에서 에이전트가 새 Enterprise Manager로 장애 조치된 경우 이 속성을 1로 설정하면 에이전트 캐시를 하루만에 플러시할 수 있습니다. 속성을 1로 설정하면 알려진 종속성이 캐시에서 제거되고 에이전트가 종속성을 다시 발견하여 새 Enterprise Manager로 데이터를 전송할 수 있게 됩니다.

속성 설정

속성은 0 보다 큰 모든 정수로 설정할 수 있습니다.

기본값

30 일

예

```
com.wily.introscope.agent.transactiontrace.boundaryTracing.cacheFlushFrequency=1
```

참고

이 속성의 변경 사항은 즉시 적용되며 관리되는 응용 프로그램 서버를 다시 시작할 필요가 없습니다.

agent.transactiontrace.boundaryTracing.enable

트랜잭션 추적 경계 추적이 사용되거나 사용되지 않도록 설정합니다. SOA 종속성 맵에 정보를 표시하려면 경계 추적이 사용되도록 설정해야 합니다. SOA 종속성 맵을 사용하지 않는다면 이 속성이 사용되지 않도록 설정하여 오버헤드를 줄일 수 있습니다.

오버헤드 문제를 방지하기 위해 경계 추적은 에이전트 캐시에서 특정 크기로 전송됩니다.

속성 설정

속성은 true 또는 false로 설정할 수 있습니다.

- True - SOA 종속성 맵에 필요한 트랜잭션 추적 경계 추적 정보를 수집하여 Enterprise Manager로 보냅니다.
- False - SOA 종속성 맵에 필요한 트랜잭션 추적 경계 추적 정보를 수집하지 않습니다.

기본값

True

예

```
com.wily.introscope.agent.transactiontrace.boundaryTracing.enable=true
```

참고

이 속성의 변경 내용을 적용하려면 관리되는 응용 프로그램을 다시 시작해야 합니다.

soa.client.prependhandler

클라이언트 측에서 CA APM for SOA SOAP 헤더의 삽입 지점을 제어합니다. 기본적으로 SOAP 핸들러 체인의 첫 번째 핸들러가 CA APM for SOA SOAP 헤더를 삽입합니다. 응용 프로그램이 SOAP 메시지를 처리하는 방식에 따라, 이 속성을 사용하여 SOAP 헤더를 SOAP 핸들러 체인의 처음 또는 마지막에 삽입할 수 있습니다.

이 속성은 SOAP 핸들러를 사용하는 응용 프로그램 서버에만 적용 가능하며 사용 중인 SOAP 엔진과 API에 따라 달라집니다. 예를 들어 응용 프로그램 서버가 이전 버전의 WebLogic 또는 WebSphere를 실행하거나 응용 프로그램 서버가 SAP NetWeaver를 실행하는 경우 이 속성을 사용할 수 있습니다. Apache Axis, Apache CXF, 기본 JBoss, WebLogic 10.x(JAX-WS), WebSphere 7.0(JAX-WS), .NET 또는 Spring 웹 서비스를 사용 중인 경우에는 속성을 적용할 수 없습니다.

속성 설정

속성은 true 또는 false로 설정할 수 있습니다.

- true - SOAP 핸들러 체인의 첫 번째 핸들러를 사용하여 SOAP 헤더를 삽입합니다.
- false - SOAP 핸들러 체인의 마지막 핸들러를 사용하여 SOAP 헤더를 추가합니다.

기본값

True

예

```
com.wily.introscope.soa.client.prependhandler=true
```

참고

이 속성의 변경 내용을 적용하려면 관리되는 응용 프로그램을 다시 시작해야 합니다.

soa.server.appendhandler

서버 CA APM for SOA SPM SOAP 헤더의 검색 및 제거를 제어합니다. 기본적으로 SOAP 핸들러 체인의 마지막 핸들러가 CA APM for SOA SOAP 헤더를 읽습니다. 응용 프로그램이 SOAP 메시지를 처리하는 방식에 따라, 이 속성을 사용하여 SOAP 헤더를 SOAP 핸들러 체인의 처음 또는 마지막에 읽고 제거할 수 있습니다.

이 속성은 SOAP 핸들러를 사용하는 응용 프로그램 서버에만 적용 가능하며 사용 중인 SOAP 엔진과 API에 따라 달라집니다. 예를 들어 응용 프로그램 서버가 이전 버전의 WebLogic 또는 WebSphere를 실행하거나 응용 프로그램 서버가 SAP NetWeaver를 실행하는 경우 이 속성을 사용할 수 있습니다. Apache Axis, Apache CXF, 기본 JBoss, WebLogic 10.x(JAX-WS), WebSphere 7.0(JAX-WS), .NET 또는 Spring 웹 서비스를 사용 중인 경우에는 속성을 적용할 수 없습니다.

속성 설정

속성은 true 또는 false로 설정할 수 있습니다.

- true - SOAP 핸들러 체인의 마지막 핸들러를 사용하여 CA APM for SOA SOAP 헤더를 읽은 다음 제거합니다.
- false - SOAP 핸들러 체인의 첫 번째 핸들러를 사용하여 CA APM for SOA SOAP 헤더를 읽은 다음 제거합니다.

기본값

True

예

```
com.wily.introscope.soa.server.appendhandler=false
```

참고

이 속성의 변경 내용을 적용하려면 관리되는 응용 프로그램을 다시 시작해야 합니다.

부록 B: SOA 특정 Enterprise Manager 구성 속성

이 단원에서는 CA APM for SOA 의 구성에 사용되는 Enterprise Manager 속성을 설명합니다.

이 섹션은 다음 항목을 포함하고 있습니다.

[Enterprise Manager 속성 구성](#) (페이지 375)

[SOA 특정 Enterprise Manager 속성 정보](#) (페이지 377)

Enterprise Manager 속성 구성

Enterprise Manager 구성 속성을 사용하면 Enterprise Manager 의 동작과 작동을 제어하고 설정을 환경에 맞게 사용자 지정할 수 있습니다. CA APM for SOA 에는 기본 설정이 적용된 여러 개의 SOA 특정 Enterprise Manager 속성이 포함되어 있습니다. 하지만 이러한 속성을 사용하려면 속성 이름을 *IntroscopeEnterpriseManager.properties* 파일에 추가하고 해당 값을 업데이트해야 합니다.

구성 속성을 *IntroscopeEnterpriseManager.properties* 에 추가하려면

1. <EM_Home>/*IntroscopeEnterpriseManager.properties* 파일을 텍스트 편집기에서 엽니다.
2. "SOA 특정 Enterprise Manager 속성 정보"에 있는 속성 목록에서 *IntroscopeEnterpriseManager.properties* 파일에 추가할 속성을 확인합니다.

3. *com.wily.introscope* 접두사를 포함한 속성의 전체 이름을 *IntroscopeEnterpriseManager.properties* 파일의 SOA 특정 섹션에 입력합니다. 예:
#####
SOA Performance Management EM Settings
=====
com.wily.introscope.soa.dashboard.typeviewer.mostcritical.count
4. 필요에 맞게 속성 값을 설정합니다. 예를 들어 이 속성의 기본 설정을 사용하려면 다음과 같이 지정합니다.
com.wily.introscope.soa.dashboard.typeviewer.mostcritical.count=10
5. *IntroscopeEnterpriseManager.properties* 파일을 저장하고 닫습니다.

Enterprise Manager 확장 속성이 *IntroscopeEnterpriseManager.properties* 파일에 추가된 후에는 필요에 따라 속성 값을 변경할 수 있습니다.

Enterprise Manager 구성 속성 값을 변경하려면

1. <EM_Home>/*IntroscopeEnterpriseManager.properties* 파일을 텍스트 편집기에서 엽니다.
2. 변경할 속성을 찾아서 환경에 적절한 새 값으로 설정합니다. 예:
com.wily.introscope.soa.dashboard.typeviewer.mostcritical.count=15
위의 예는 설명을 위한 것이며 CA APM for SOA 설정 권장 사항이 아닙니다.
3. *IntroscopeEnterpriseManager.properties* 파일을 저장하고 닫습니다.
4. Enterprise Manager를 다시 시작합니다.

SOA 특정 Enterprise Manager 속성 정보

IntroscopeEnterpriseManager.properties 파일에서 다음의 SOA 특정 속성을 설정할 수 있습니다. 모든 속성은 *com.wily.introscope* 접두사로 시작하지만 읽기 편하도록 접두사 없이 나열되어 있습니다.

참고: 다른 Enterprise Manager 속성의 설정에 대한 자세한 내용은 *CA APM 구성 및 관리 안내서*를 참조하십시오.

[soa.dependencymap.aging.refresh.interval](#) (페이지 380)

이전에 검색된 종속성의 경과 시간을 확인할 간격을 지정합니다.

[soa.dependencymap.aging.expire.interval](#) (페이지 381)

종속성이 만료 전 SOA 종속성 맵에서 유지될 수 있는 최대 날짜 수를 지정합니다.

[soa.dependencymap.heuristics.clientside.enable](#) (페이지 382)

클라이언트 측 논리적 동등성 추론 규칙을 사용하거나 사용하지 않도록 설정합니다.

[soa.dependencymap.heuristics.namematch.enable](#) (페이지 383)

클라이언트 측 논리적 동등성 추론 이름 일치 규칙을 사용하거나 사용하지 않도록 설정합니다.

[soa.dependencymap.heuristics.serverside.enable](#) (페이지 384)

서버 측 논리적 동등성 추론 규칙을 사용하거나 사용하지 않도록 설정합니다.

[soa.dependencymap.log.suppress](#) (페이지 385)

로그 파일에 쓸 수 있는 동일한 오류 또는 경고 메시지의 최대 수를 지정합니다.

[soa.dependencymap.max.edge.ratio](#) (페이지 386)

맵에서 노드에 대한 종속성의 최대 비율을 지정합니다.

[soa.dependencymap.max.vertices](#) (페이지 387)

독립 실행형 또는 Collector Enterprise Manager에서 종속성 맵에 대해 저장할 수 있는 최대 노드 수를 지정합니다.

[soa.deviation.enable](#) (페이지 388)

편차 메트릭 생성에 필요한 계산을 사용하거나 사용하지 않도록 설정합니다.

[soa.deviation.art.enable](#) (페이지 389)

"평균 응답 시간" 편차 메트릭이 사용되거나 사용되지 않도록 설정합니다.

[soa.deviation.dependencymetric.enable](#) (페이지 389)

종속성 메트릭 생성에 필요한 계산을 사용하거나 사용하지 않도록 설정합니다.

[soa.deviation.count.per.metric](#) (페이지 390)

편차 메트릭당 최대 작업 수를 지정합니다.

[soa.deviation.dependency.refreshrate](#) (페이지 391)

캐시된 SOA 종속성 맵 종속성 관계 데이터를 새로 고칠 빈도를 시간 단위로 지정합니다.

[soa.deviation.errors.enable](#) (페이지 392)

"간격당 오류 편차" 메트릭을 사용하거나 사용하지 않도록 설정합니다.

[soa.deviation.max.metric.count](#) (페이지 393)

보고할 최대 편차 메트릭 수를 지정합니다.

[soa.deviation.metric.expressionlist](#) (페이지 394)

편차 메트릭 표현식에 대한 이름의 목록을 정의합니다.

[soa.deviation.metric.calledbackends](#) (페이지 395)

deviation.metric.expressionlist 속성에 정의된 이름을 사용하여 편차 메트릭을 생성하기 위한 새로운 속성을 구성합니다.

[soa.deviation.usage.enable](#) (페이지 396)

"간격당 응답 편차" 메트릭이 사용되거나 사용되지 않도록 설정합니다.

[soa.dashboard.typeviewer.average.enable](#) (페이지 397)

선택된 작업에 대한 "평균 응답 시간 편차" 그래프를 표시하거나 표시하지 않습니다.

[soa.dashboard.typeviewer.response.enable](#) (페이지 399)

선택된 작업에 대한 "간격당 응답 편차" 그래프를 표시하거나 표시하지 않습니다.

[soa.dashboard.typeviewer.errors.enable](#) (페이지 398)

선택된 작업에 대한 "간격당 오류 편차" 그래프를 표시하거나 표시하지 않습니다.

soa.dashboard.typeviewer.mostcritical.enable (페이지 400)

"가장 중요한 작업" 대시보드 및 "가장 중요한" 탭을 표시하거나 표시하지 않습니다.

soa.dashboard.typeviewer.mostcritical.count (페이지 401)

보고되는 중요 작업의 최대 수를 지정합니다.

soa.dashboard.typeviewer.mostdependent.enable (페이지 402)

"가장 종속된 작업" 대시보드 및 "가장 종속된" 탭을 표시하거나 표시하지 않습니다.

soa.dashboard.typeviewer.mostdependent.count (페이지 403)

보고되는 종속 작업의 최대 수를 지정합니다.

대개의 경우 Enterprise Manager 를 다시 시작해야 속성 변경 사항이 적용됩니다.

soa.dependencymap.aging.refresh.interval

Enterprise Manager 가 다음 번 SOA 종속성 맵 종속성 만료 처리 확인을 수행하기 전에 대기할 시간을 시간 단위로 지정합니다.

Enterprise Manager 는 검색된 각 종속성의 경과 시간을 추적하고 주기적으로 종속성을 다시 검색하여 여전히 존재하는지 확인합니다. 종속성의 경과 시간은 가장 최근의 검색을 기준으로 합니다.

SOA 종속성 맵 종속성 만료 처리 확인 중에 Enterprise Manager 는 SOA 종속성 맵에서 각 종속성의 경과 시간을 확인합니다.

SOA 종속성 맵 새로 고침 간격은 기본적으로 1 시간으로 정의됩니다. 따라서 여섯 개의 SOA 종속성 맵 새로 고침 간격은 6 시간 동안 지속됩니다.

속성 설정

속성은 0 보다 큰 모든 정수로 설정할 수 있습니다.

기본값

6 시간

예

```
com.wily.introscope.soa.dependencymap.aging.refresh.interval=6
```

참고

- 이 속성과 `com.wily.introscope.soa.dependencymap.aging.expire.interval` 속성을 함께 사용하여 종속성을 제거해야 하는지 여부를 결정합니다.
- 이 속성의 변경 사항은 즉시 적용되며 Enterprise Manager 를 다시 시작할 필요가 없습니다.

soa.dependencymap.aging.expire.interval

SOA 종속성 맵에서 종속성의 최대 경과 시간(일)입니다. 종속성 만료 처리 확인 중에 Enterprise Manager 는 지정된 값보다 오래된 모든 종속성을 SOA 종속성 맵에서 제거합니다.

예를 들어 *com.wily.introscope.soa.dependencymap.aging.expire.interval* 값이 60 일인 경우 만료된 종속성은 지난 60 일 동안 다시 검색되지 않은 종속성입니다. 이 기간 동안 종속성이 다시 검색되지 않은 경우 종속성 맵에서 종속성이 제거됩니다.

속성 설정

속성은 0 보다 큰 모든 정수로 설정할 수 있습니다.

기본값

60 일

예

```
com.wily.introscope.soa.dependencymap.aging.expire.interval=90
```

참고

- 이 속성과 *com.wily.introscope.soa.dependencymap.aging.refresh.interval* 속성이 함께 작동합니다.
- 이 속성은 SOA 종속성 맵 종속성 만료 처리 확인 중에 적용할 종속성 제거 기준을 결정합니다.
- 이 속성의 변경 사항은 즉시 적용되며 Enterprise Manager 를 다시 시작할 필요가 없습니다.

soa.dependencymap.heuristics.clientside.enable

CA APM for SOA 클라이언트 측 논리적 동등성 추론 규칙을 사용하거나 사용하지 않도록 설정합니다.

클라이언트 측 논리적 동등성 추론 규칙은 다음과 같습니다.

클라이언트 유형의 두 물리적 서비스 작업이 모두 서버유형의 동일한 물리적 서비스 작업에 종속된 경우 논리적으로 동등하다고 간주합니다.

이러한 설정은 서로 다른 두 개의 응용 프로그램이 동일한 클라이언트 측 서비스 작업 호출을 수행할 때 발생할 수 있습니다. 이 규칙은 두 작업의 메트릭 경로(에이전트 설명자 제외)가 같을 때와 같지 않을 때 모두 이러한 논리적 동등성을 감지합니다.

속성 설정

속성은 true 또는 false로 설정할 수 있습니다.

- true - CA APM for SOA 클라이언트 측 논리적 동등성 추론 규칙을 적용합니다.
- false - CA APM for SOA 클라이언트 측 논리적 동등성 추론 규칙을 사용하지 않습니다.

기본값

True

예

com.wily.introscope.soa.dependencymap.heuristics.clientside.enable=true

참고

이 규칙을 변경할 경우 이전에 검색된 종속성에는 변경 사항이 적용되지 않습니다. 따라서 Enterprise Manager를 중지하고 이전에 저장된 SOA 종속성 맵 파일 유형을 모두 삭제하지 않고 규칙 값을 변경하면 예기치 않은 결과가 발생할 수 있습니다.

이 속성에 대한 변경 사항은 즉시 적용되므로 Enterprise Manager를 다시 시작할 필요가 없습니다.

추가 정보:

[저장된 종속성 데이터 무시 또는 제거](#) (페이지 78)

soa.dependencymap.heuristics.namematch.enable

CA APM for SOA 논리적 동등성 추론 이름 일치 규칙을 사용하거나 사용하지 않도록 설정합니다.

논리적 동등성 추론 이름 일치 규칙은 두 작업의 메트릭 경로(에이전트 설명자 제외)가 동일할 때 논리적 동등성을 감지합니다.

CA APM for SOA 논리적 동등성 추론 이름 일치 규칙은 에이전트 설명자 삭제 후의 메트릭 경로가 동일한 두 물리적 웹 서비스 작업을 논리적으로 동등한 것으로 간주합니다.

속성 설정

속성은 true 또는 false로 설정할 수 있습니다.

- true - 논리적 동등성 추론 이름 일치 규칙을 사용합니다.
- false - 논리적 동등성 추론 이름 일치 규칙을 사용하지 않습니다.

기본값

False

예

com.wily.introscope.soa.dependencymap.heuristics.namematch.enable=true

참고

이 규칙을 변경할 경우 이전에 검색된 종속성에는 변경 사항이 적용되지 않습니다. 따라서 Enterprise Manager를 중지하고 이전에 저장된 SOA 종속성 맵 파일 유형을 모두 삭제하지 않고 규칙 값을 변경하면 예기치 않은 결과가 발생할 수 있습니다.

CA APM for SOA에서 가상 에이전트를 생성할 때는 이 속성을 사용하도록 설정합니다.

이 속성에 대한 변경 사항은 즉시 적용되므로 Enterprise Manager를 다시 시작할 필요가 없습니다.

추가 정보:

[저장된 종속성 데이터 무시 또는 제거](#) (페이지 78)

soa.dependencymap.heuristics.serverside.enable

CA APM for SOA 서버 측 논리적 동등성 추론 규칙을 사용하거나 사용하지 않도록 설정합니다.

서버 측 논리적 동등성 추론 규칙은 유형 서버의 두 물리적 서비스 작업에 모두 종속된 유형 클라이언트의 물리적 서비스 작업이 있는 경우에만 유형 서버의 두 물리적 서비스 작업이 논리적으로 동등하다고 간주합니다.

일반적으로 이런 경우는 클라이언트 측 웹 서비스 작업 호출이 몇 개의 사용 가능한 서버 측 웹 서비스 작업 구현 중 하나로 발송되는 경우처럼 특정 유형의 부하 분산 메커니즘이 SOA에 통합되어 있을 때 발생합니다.

속성 설정

속성은 true 또는 false로 설정할 수 있습니다.

- true - CA APM for SOA 서버 측 논리적 동등성 추론 규칙을 사용합니다.
- false - CA APM for SOA 서버 측 논리적 동등성 추론 규칙을 사용하지 않습니다.

기본값

True

예

com.wily.introscope.soa.dependencymap.heuristics.serverside.enable=true

참고

이 규칙을 변경할 경우 이전에 검색된 종속성에는 변경 사항이 적용되지 않습니다. 따라서 Enterprise Manager를 중지하고 이전에 저장된 SOA 종속성 맵 파일 유형을 모두 삭제하지 않고 규칙 값을 변경하면 예기치 않은 결과가 발생할 수 있습니다. 이 속성의 변경 사항은 즉시 적용되며 Enterprise Manager를 다시 시작할 필요가 없습니다.

추가 정보:

[저장된 종속성 데이터 무시 또는 제거](#) (페이지 78)

soa.dependencymap.log.suppress

로그 파일에 쓸 수 있는 동일한 오류 또는 경고 메시지의 최대 수를 지정합니다. 이 수에 도달하면 중복 메시지가 억제됩니다.

이 속성을 사용하여 반복되는 오류나 경고 메시지가 발생할 때마다 Enterprise Manager 로그 파일로 전송되는 것을 방지할 수 있습니다. 예를 들어 허용된 최대 종속성 수나 최대 노드 수를 초과하는 종속성 맵이 있는 경우 최대값에 도달할 때마다 로그 파일에 경고 메시지가 기록됩니다. 이 속성을 사용하면 지정된 횟수만큼 로그 파일에 오류나 경고가 게시된 다음 마지막 오류나 경고에 이제부터 메시지가 억제됨을 나타내는 텍스트 문자열이 추가되며, Enterprise Manager 를 다시 시작하거나 이 속성을 수정할 때까지 더 이상 해당 오류나 경고의 복사본이 로그 파일에 기록되지 않습니다.

속성 설정

속성은 0 보다 큰 모든 정수로 설정할 수 있습니다.

기본값

5 개의 로그 항목

예

com.wily.introscope.soa.dependencymap.log.suppress=5

참고

이 속성을 사용하기 위해 Enterprise Manager 를 다시 시작할 필요는 없습니다.

하지만 속성 값을 변경하면 오류 및 경고 메시지를 억제하기 위해 셈하는 횟수가 처음부터 다시 시작됩니다. 예를 들어 특정 오류 메시지가 5 번 발생하면 기본적으로 해당 오류 메시지가 억제됩니다. 이 경우 속성 값을 6 으로 변경하면 오류 메시지가 더 이상 억제되지 않으며 오류가 앞으로 6 번 더 발생해야만 오류 메시지가 억제됩니다.

soa.dependencymap.max.edge.ratio

맵에서 노드에 대한 종속성의 최대 비율을 지정합니다. 이 속성을 사용하여 복잡한 대규모 SOA 환경에서 Enterprise Manager에 저장되는 종속성 맵의 전반적인 복잡성을 제어할 수 있습니다.

Enterprise Manager에 저장되는 종속성 맵 데이터는 대개 검색된 응용 프로그램 간의 모든 종속성을 나타내며, 배포한 SOA(Service-Oriented Architecture)의 완전한 모델을 제공합니다. 하지만 매우 크거나 복잡한 SOA 환경에서는 모든 SOA 구성 요소와 해당 종속성의 전체 표현이 Enterprise Manager 자체의 성능과 작업에 영향을 미칠 수 있습니다. 이 속성을 사용하면 노드와 종속성 간의 비율을 지정하여 허용되는 복잡성 수준을 제어할 수 있습니다. 이 제한에 도달하면 추가 종속성이 저장되지 않고 Enterprise Manager 로그 파일에 경고가 기록됩니다.

이 속성과 com.wily.introscope.soa.dependencymap.max.vertices 속성을 함께 사용하여 종속성 맵에 저장되는 총 노드 수를 제한할 수도 있습니다.

클러스터 환경에서는 이 속성이 Collector Enterprise Manager에만 적용됩니다. MOM을 사용하면 모든 수집기의 통합된 SOA 환경에 대한 전체 표현을 저장할 수 있습니다.

속성 설정

속성은 0보다 큰 모든 정수로 설정할 수 있습니다.

기본값

노드당 5 개의 종속성

예

```
com.wily.introscope.soa.dependencymap.max.edge.ratio=5
```

참고

com.wily.introscope.soa.dependencymap.max.vertices 및 com.wily.introscope.soa.dependencymap.max.edge.ratio 속성의 기본값을 사용하면 종속성 맵이 5000 개의 노드와 최대 25000 개의 종속성(5 x 5000)으로 제한됩니다.

대부분의 SOA 네트워크에서 구성 요소 수는 5000 개보다 적으며 종속성 비율은 구성 요소당 종속성 1 개 또는 2 개입니다. 따라서 기본 설정만으로 대부분의 조직에서 필요한 수준보다 훨씬 복잡한 구성을 처리할 수 있습니다. 이러한 속성을 사용하여 종속성 맵의 크기와 복잡성을 의도적으로 제한할 수 있지만 그럴 경우 SOA 종속성 맵 모델이 불완전해질 수 있습니다.

soa.dependencymap.max.vertices

독립 실행형 또는 Collector Enterprise Manager에서 종속성 맵에 대해 저장할 수 있는 최대 노드 수를 지정합니다. 이 속성을 사용하여 복잡한 대규모 SOA 환경에서 Enterprise Manager에 저장되는 종속성 맵의 최대 크기를 제어할 수 있습니다.

Enterprise Manager에 저장되는 종속성 맵 데이터는 대개 검색된 응용 프로그램 간의 모든 종속성을 나타내며, 배포한 SOA(Service-Oriented Architecture)의 완전한 모델을 제공합니다. 하지만 매우 크거나 복잡한 SOA 환경에서는 모든 SOA 구성 요소와 해당 종속성의 전체 표현이 Enterprise Manager 자체의 성능과 작업에 영향을 미칠 수 있습니다. 이 속성을 사용하면 포함할 노드의 최대 수를 지정하여 SOA 모델의 크기를 제한할 수 있습니다. 이 제한에 도달하면 추가 정보가 저장되지 않고 Enterprise Manager 로그 파일에 경고가 기록됩니다.

이 속성과 com.wily.introscope.soa.dependencymap.max.edge.ratio 속성을 함께 사용하여 종속성 맵에 저장되는 총 종속성 관계 수를 제한할 수도 있습니다.

클러스터 환경에서는 이 속성이 Collector Enterprise Manager에만 적용됩니다. MOM을 사용하면 모든 수집기의 통합된 SOA 환경에 대한 전체 표현을 저장할 수 있습니다.

속성 설정

속성은 0 보다 큰 모든 정수로 설정할 수 있습니다.

기본값

5000 개의 노드

예

```
com.wily.introscope.soa.dependencymap.max.vertices=5000
```

soa.deviation.enable

다음 편차 메트릭을 생성하는 데 필요한 계산이 사용되거나 사용되지 않도록 설정합니다.

- 평균 응답 시간 편차
- 간격당 오류 편차
- 간격당 응답 편차

이 속성을 `true`로 설정하면 편차 메트릭을 수집하여 보고할 수 있습니다. 계속해서 각 편차 메트릭이 개별적으로 사용되거나 사용되지 않도록 설정할 수 있습니다.

속성 설정

속성은 `true` 또는 `false`로 설정할 수 있습니다.

- `True` - 편차 메트릭 데이터를 제공하기 위한 계산을 수행합니다.
- `False` - 편차 메트릭을 보고하지 않습니다.

기본값

`True`

예

```
com.wily.introscope.soa.deviation.enable=true
```

참고

이 속성의 변경 사항을 적용하려면 Enterprise Manager를 다시 시작해야 합니다.

soa.deviation.art.enable

"평균 응답 시간" 편차 메트릭이 사용되거나 사용되지 않도록 설정합니다. 이 속성은 편차 메트릭을 계산하는 경우 평균 응답 시간 편차 메트릭을 보고할지 여부를 결정합니다.

속성 설정

속성은 **true** 또는 **false**로 설정할 수 있습니다.

- **True** - 계산을 수행하고 "평균 응답 시간 편차" 메트릭 데이터를 보고합니다.
- **False** - "평균 응답 시간 편차" 메트릭을 보고하지 않습니다.

기본값

True

예

```
com.wily.introscope.soa.deviation.art.enable=true
```

참고

이 속성의 변경 사항을 적용하려면 Enterprise Manager를 다시 시작해야 합니다.

soa.deviation.dependencymetric.enable

이 속성은 다음 종속성 메트릭을 생성하는 데 필요한 계산을 사용하거나 사용하지 않도록 설정합니다.

- 중요 직접
- 중요 간접
- 종속성 직접
- 종속성 간접

이 속성은 CA APM for SOA 종속성 메트릭이 보고되는지 여부를 결정합니다.

속성 설정

속성은 true 또는 false로 설정할 수 있습니다.

- true - 종속성 메트릭 데이터를 제공하기 위한 계산을 수행합니다.
- false - 종속성 메트릭 계산을 수행하지 않습니다.

기본값

True

예

```
com.wily.introscope.soa.deviation.dependencymetric.enable=true
```

참고

이 속성의 변경 사항을 적용하려면 Enterprise Manager를 다시 시작해야 합니다.

soa.deviation.count.per.metric

이 속성은 편차 메트릭이 계산되는 "WebServices" 노드 아래의 최대 작업 수를 지정합니다. 보고된 작업 수가 이 숫자를 초과하는 경우 모든 "서버" 및 "클라이언트" 노드 네임스페이스 아래의 가장 중요한 작업이 사용됩니다.

속성 설정

속성은 0보다 큰 모든 정수로 설정할 수 있습니다.

기본값

종속성 메트릭당 25개의 작업

예

```
com.wily.introscope.soa.deviation.count.per.metric=25
```

참고

이 속성을 적용하려면 Enterprise Manager를 다시 시작하십시오.

soa.deviation.dependency.refreshrate

캐시된 SOA 종속성 맵 종속성 관계 데이터를 새로 고침 빈도를 시간 단위로 지정합니다.

SOA 종속성 맵 데이터는 편차 메트릭 서비스별로 캐시됩니다. 15 초마다 한 번씩 이러한 서비스 및 작업에 대한 중요, 종속 및 편차 메트릭이 보고됩니다.

캐시된 SOA 종속성 맵 데이터는 이 속성을 기준으로 주기적으로 새로 고쳐집니다.

이 맵은 거의 변경되지 않으므로 대개 배포가 변경된 경우에만 새로 고침이 수행됩니다.

속성 설정

속성은 0 보다 큰 모든 정수로 설정할 수 있습니다.

기본값

1 시간

예

```
com.wily.introscope.soa.deviation.dependency.refreshrate=1
```

참고

이 속성의 변경 사항을 적용하려면 Enterprise Manager 를 다시 시작해야 합니다.

soa.deviation.errors.enable

오류 편차 메트릭이 사용되거나 사용되지 않도록 설정합니다. 이 속성은 편차 메트릭을 계산하는 경우 오류 수 편차 메트릭을 보고할지 여부를 결정합니다.

속성 설정

속성은 **true** 또는 **false**로 설정할 수 있습니다.

- **True** - 계산을 수행하고 "간격당 오류 편차" 메트릭 데이터를 보고합니다.
- **False** - "간격당 오류 편차" 메트릭을 보고하지 않습니다.

기본값

True

예

```
com.wily.introscope.soa.deviation.errors.enable=true
```

참고

이 속성의 변경 사항을 적용하려면 Enterprise Manager를 다시 시작해야 합니다.

soa.deviation.max.metric.count

보고할 최대 편차 메트릭 수를 지정합니다.

CA Technologies에서는 이 속성의 기본값을 변경하는 것을 권장하지 않습니다. 기본값을 사용하면 Enterprise Manager 성능에 영향을 주지 않으면서 유용한 정보를 얻는 데 필요한 충분한 메트릭이 보고됩니다. 기본값을 변경하면 Enterprise Manager에서 성능 문제가 발생할 수 있습니다.

속성 설정

속성은 0 보다 큰 모든 정수로 설정할 수 있습니다.

기본값

총 1000 개 편차 메트릭

예

```
com.wily.introscope.soa.deviation.metric.count=1000
```

참고

이 속성의 변경 사항을 적용하려면 Enterprise Manager를 다시 시작해야 합니다.

soa.deviation.mean.days

이 속성은 평균 값을 계산하기 위해 작업을 누적할 기간(일)을 지정합니다. 평균 값은 지정된 기간(일) 후에 자동으로 계산됩니다.

속성 설정

속성은 0 보다 큰 모든 정수로 설정할 수 있습니다.

기본값

7 일

예

```
com.wily.introscope.soa.deviation.mean.days=3
```

참고

이 속성을 적용하려면 Enterprise Manager 를 다시 시작하십시오.

soa.deviation.metric.expressionlist

이 속성은 편차 메트릭 표현식에 대한 이름 목록을 정의합니다. 편차 메트릭 만들기에 사용할 이름의 쉼표로 구분된 목록으로 이 속성을 설정하십시오. 각 이름은 별도의 메트릭 표현식으로 사용됩니다.

deviation.metric.calledbackends 속성은 이 속성을 참조하여 사용자 정의 편차 메트릭을 만듭니다.

속성 설정

속성은 이름의 사용자 정의 목록으로 설정할 수 있습니다.

기본값

calledbackends

예

deviation.metric.expressionlist=alpha1, beta2, gama3

추가 편차 계산

기본적으로 편차 메트릭은 개별 작업에 대해서만 계산됩니다. 편차를 WebService, 클라이언트 또는 서버 네임스페이스 수준에서 사용할 수 있게 하려면 다음 문을 *IntroscopeEnterpriseManager.properties* 파일에 추가하십시오.

```
com.wily.introscope.soa.deviation.metric.expressionlist=test,test1  
com.wily.introscope.soa.deviation.metric.test=WebServices\\|Client\\|.*  
com.wily.introscope.soa.deviation.metric.test1=WebServices\\|Server\\|.*
```

중요! 이런 방식으로 여러 수준에서 편차를 계산하는 경우 추가 처리 오버헤드가 필요합니다.

참고

이 속성의 변경 사항을 적용하려면 Enterprise Manager 를 다시 시작하십시오.

soa.deviation.metric.calledbackends

deviation.metric.expressionlist 속성에 정의된 이름을 사용하여 편차 메트릭을 생성하기 위한 새로운 속성을 구성합니다.

새 속성 이름의 형식은 다음과 같습니다.

`com.wily.introscope.soa.deviation.metric.<user-defined_name> = <user-defined regular expression>`

deviation.metric.calledbackends 속성에 할당되는 값은 메트릭 표현식입니다.

예:

`com.wily.introscope.soa.deviation.metric.alpha1=Frontends|Called Backends
com.wily.introscope.soa.deviation.metric.beta2=Frontends|Called Backends`

기본값

calledbackends

예

`com.wily.introscope.soa.deviation.metric.calledbackends=Frontends|Frontends|FBApp
$Frontend|Called Backends(백엔드) |FBApp$Backend`

참고

이 속성의 변경 사항을 적용하려면 Enterprise Manager 를 다시 시작해야 합니다.

soa.deviation.usage.enable

"간격당 응답 편차" 메트릭이 사용되거나 사용되지 않도록 설정합니다. 이 속성은 편차 메트릭을 계산하는 경우 응답 수 편차 메트릭을 보고할지 여부를 결정합니다.

속성 설정

속성은 **true** 또는 **false**로 설정할 수 있습니다.

- **True** - 계산을 수행하고 "간격당 응답 편차" 메트릭 데이터를 보고합니다.
- **False** - "간격당 응답 편차" 메트릭을 보고하지 않습니다.

기본값

True

예

```
com.wily.introscope.soa.deviation.usage.enable=true
```

참고

이 속성의 변경 사항을 적용하려면 Enterprise Manager를 다시 시작해야 합니다.

soa.dashboard.typeviewer.average.enable

선택된 작업에 대해 수집된 편차 데이터를 다음 위치의 "평균 응답 시간 편차" 그래프에 표시합니다.

- "SOA 성능 - 가장 중요한 작업" 대시보드
- "SOA 성능 - 가장 종속된 작업" 대시보드
- "가장 중요한" 탭
- "가장 종속된" 탭

편차 데이터는 에이전트를 통해 Enterprise Manager에 보고되고, Workstation에 표시하도록 허용된 경우에만 대시보드 및 탭에서 사용할 수 있습니다. 편차 데이터의 보고에 대한 자세한 내용은 ["편차 탭에서 편차 메트릭 보기"](#) (페이지 66)를 참조하십시오.

속성 설정

속성은 true 또는 false로 설정할 수 있습니다.

- True - 선택된 작업의 데이터를 "평균 응답 시간 편차" 그래프에 표시합니다.
- False - 선택된 작업의 데이터를 "평균 응답 시간 편차" 그래프에 표시하지 않습니다.

기본값

True

예

```
com.wily.introscope.soa.dashboard.typeviewer.average.enable=true
```

참고

이 속성은 *IntroscopeEnterpriseManager.properties* 파일 또는 *IntroscopeWorkstation.properties* 파일에서 정의할 수 있습니다. 속성이 *IntroscopeWorkstation.properties* 파일과 *IntroscopeEnterpriseManager.properties* 파일에 모두 정의된 경우에는 *IntroscopeWorkstation.properties* 속성 설정이 사용됩니다.

이 속성의 변경 사항을 적용하려면 Enterprise Manager를 다시 시작해야 합니다.

soa.dashboard.typeviewer.errors.enable

선택된 작업에 대해 수집된 편차 데이터를 다음 위치의 "간격당 오류 편차" 그래프에 표시합니다.

- "SOA 성능 - 가장 중요한 작업" 대시보드
- "SOA 성능 - 가장 종속된 작업" 대시보드
- "가장 중요한" 탭
- "가장 종속된" 탭

편차 데이터는 에이전트를 통해 Enterprise Manager에 보고되고, Workstation에 표시하도록 허용된 경우에만 대시보드 및 탭에서 사용할 수 있습니다. 편차 데이터의 보고에 대한 자세한 내용은 ["편차 탭에서 편차 메트릭 보기"](#) (페이지 66)를 참조하십시오.

속성 설정

속성은 true 또는 false로 설정할 수 있습니다.

- True - 선택된 작업의 데이터를 "간격당 오류 편차" 그래프에 표시합니다.
- False - 선택된 작업의 데이터를 "간격당 오류 편차" 그래프에 표시하지 않습니다.

기본값

True

예

```
com.wily.introscope.soa.dashboard.typeviewer.response.enable=true
```

참고

이 속성은 *IntroscopeEnterpriseManager.properties* 파일 또는 *IntroscopeWorkstation.properties* 파일에서 정의할 수 있습니다. 이 속성이 *IntroscopeWorkstation.properties* 파일과 *IntroscopeEnterpriseManager.properties* 파일 모두에 정의된 경우에는 *IntroscopeWorkstation.properties* 속성 설정이 사용됩니다.

이 속성의 변경 사항을 적용하려면 Enterprise Manager를 다시 시작해야 합니다.

soa.dashboard.typeviewer.response.enable

선택된 작업에 대해 수집된 편차 데이터를 다음 위치의 "간격당 응답 편차" 그래프에 표시합니다.

- "SOA 성능 - 가장 중요한 작업" 대시보드
- "SOA 성능 - 가장 종속된 작업" 대시보드
- "가장 중요한" 탭
- "가장 종속된" 탭

편차 데이터는 에이전트를 통해 Enterprise Manager에 보고되고, Workstation에 표시하도록 허용된 경우에만 대시보드 및 탭에서 사용할 수 있습니다. 편차 데이터의 보고에 대한 자세한 내용은 ["편차 탭에서 편차 메트릭 보기"](#) (페이지 66)를 참조하십시오.

속성 설정

속성은 true 또는 false로 설정할 수 있습니다.

- True - 선택된 작업의 데이터를 "간격당 응답 편차" 그래프에 표시합니다.
- False - 선택된 작업의 데이터를 "간격당 응답 편차" 그래프에 표시하지 않습니다.

기본값

True

예

```
com.wily.introscope.soa.dashboard.typeviewer.response.enable=true
```

참고

이 속성은 *IntroscopeEnterpriseManager.properties* 파일 또는 *IntroscopeWorkstation.properties* 파일에서 정의할 수 있습니다. 이 속성이 *IntroscopeWorkstation.properties* 파일과 *IntroscopeEnterpriseManager.properties* 파일 모두에 정의된 경우에는 *IntroscopeWorkstation.properties* 속성 설정이 사용됩니다.

이 속성의 변경 사항을 적용하려면 Enterprise Manager를 다시 시작해야 합니다.

soa.dashboard.typeviewer.mostcritical.enable

"가장 중요한 작업" 대시보드를 "콘솔"에 표시하고 "가장 중요한" 탭을 Investigator에 표시합니다.

속성 설정

속성은 true 또는 false로 설정할 수 있습니다.

- True - "SOA 성능 - 가장 중요한 작업" 대시보드 옵션을 "콘솔" 대시보드 메뉴에 표시하고 "가장 중요한" 탭을 Investigator에 표시합니다.
- False - "SOA 성능 - 가장 중요한 작업" 대시보드 및 "가장 중요한" 탭에 액세스할 수 없습니다.

이 속성을 false로 설정하면 Investigator의 모든 노드에 대해 "가장 중요한" 탭이 표시되지 않습니다. "가장 중요한 작업" 대시보드는 "콘솔" 대시보드 드롭다운 목록에 계속 나열되지만 선택하더라도 대시보드가 표시되지 않습니다. "가장 중요한 작업" 대시보드 대신 대시보드를 사용할 수 없다는 오류 메시지가 나타납니다.

기본값

True

예

```
com.wily.introscope.soa.dashboard.typeviewer.mostcritical.enable=true
```

참고

이 속성은 IntroscopeEnterpriseManager.properties 파일 또는 IntroscopeWorkstation.properties 파일에서 정의할 수 있습니다. 이 속성이 IntroscopeWorkstation.properties 파일과 IntroscopeEnterpriseManager.properties 파일 모두에 정의된 경우에는 IntroscopeWorkstation.properties 속성 설정이 사용됩니다.

이 속성의 변경 사항을 적용하려면 Enterprise Manager와 Workstation을 다시 시작해야 합니다.

soa.dashboard.typeviewer.mostcritical.count

"SOA 성능 - 가장 중요한 작업" 대시보드 및 "가장 중요한" 탭에서 모든 에이전트가 보고하는 중요한 작업의 총 수를 결정합니다.

속성 설정

최대값은 0입니다.

최대값은 기본값이 25인 *com.wily.introscope.soa.deviation.count.per.metric* 속성의 값보다 작거나 같아야 합니다.

기본값

10 개 작업

예

```
com.wily.introscope.soa.dashboard.typeviewer.mostcritical.count=5
```

참고

이 속성은 *IntroscopeEnterpriseManager.properties* 파일 또는 *IntroscopeWorkstation.properties* 파일에서 정의할 수 있습니다. 이 속성이 *IntroscopeEnterpriseManager.properties* 파일과 *IntroscopeWorkstation.properties* 파일 모두에 정의된 경우에는 *IntroscopeWorkstation.properties* 속성 설정이 사용됩니다.

이 속성의 변경 사항을 적용하려면 Enterprise Manager 와 Workstation 을 다시 시작해야 합니다.

soa.dashboard.typeviewer.mostdependent.enable

"가장 종속된 작업" 대시보드를 "콘솔"에 표시하고 "가장 종속된" 탭을 Investigator에 표시합니다.

속성 설정

속성은 true 또는 false로 설정할 수 있습니다.

- True - "SOA 성능 - 가장 종속된 작업" 대시보드 옵션을 "콘솔" 대시보드 메뉴에 표시하고 "가장 종속된" 탭을 Investigator에 표시합니다.
- False - "SOA 성능 - 가장 종속된 작업" 대시보드 및 "가장 종속된" 탭에 액세스할 수 없습니다.

이 속성을 false로 설정하면 Investigator에 "가장 종속된" 탭이 표시되지 않습니다. "가장 종속된 작업" 대시보드는 "콘솔" 대시보드 드롭다운 목록에 계속 나열되지만 선택하더라도 대시보드가 표시되지 않습니다. "가장 종속된 작업" 대시보드 대신 대시보드를 사용할 수 없다는 오류 메시지가 나타납니다.

기본값

True

예

```
com.wily.introscope.soa.dashboard.typeviewer.mostdependent.enable=true
```

참고

이 속성은 *IntroscopeEnterpriseManager.properties* 파일 또는 *IntroscopeWorkstation.properties* 파일에서 정의할 수 있습니다. 이 속성이 *IntroscopeEnterpriseManager.properties* 파일과 *IntroscopeWorkstation.properties* 파일 모두에 정의된 경우에는 *IntroscopeWorkstation.properties* 속성 설정이 사용됩니다.

이 속성의 변경 사항을 적용하려면 Enterprise Manager와 Workstation을 다시 시작해야 합니다.

soa.dashboard.typeviewer.mostdependent.count

"SOA 성능 - 가장 종속된 작업" 대시보드 및 "가장 종속된" 탭에서 에이전트가 보고하는 가장 종속된 작업의 총 수를 확인합니다.

속성 설정

최대값은 0입니다.

최대값은 기본값이 25인 com.wily.introscope.soa.deviation.count.per.metric 속성의 값보다 작거나 같아야 합니다.

기본값

10 개 작업

예

```
com.wily.introscope.soa.dashboard.typeviewer.mostdependent.count=5
```

참고

이 속성은 IntroscopeEnterpriseManager.properties 파일 또는 IntroscopeWorkstation.properties 파일에서 정의할 수 있습니다. 이 속성이 IntroscopeEnterpriseManager.properties 파일과 IntroscopeWorkstation.properties 파일 모두에 정의된 경우에는 IntroscopeWorkstation.properties 속성 설정이 사용됩니다.

이 속성의 변경 사항을 적용하려면 Enterprise Manager 와 Workstation 을 다시 시작해야 합니다.

부록 C: SOA 특정 Workstation 구성 속성

Workstation 구성 속성을 사용하면 Workstation 의 동작 및 작업을 제어할 수 있습니다. 또한 이 속성을 사용하면 Workstation 을 환경에 가장 적합하게 사용자 지정할 수 있습니다. Workstation 속성은 CA APM for SOA 를 구체적으로 구성할 수 있도록 해 줍니다.

이 섹션은 다음 항목을 포함하고 있습니다.

[Workstation 속성 구성](#) (페이지 406)

[SOA 특정 Workstation 속성 정보](#) (페이지 407)

Workstation 속성 구성

CA APM for SOA 에는 기본 설정이 적용된 몇 개의 SOA 특정 Workstation 속성이 있습니다. 하지만 이러한 속성을 구성하려면 먼저 속성 이름을 *IntroscopeWorkstation.properties* 파일에 추가해야 합니다.

Workstation 속성을 *IntroscopeWorkstation.properties*에 추가하려면

1. <EM_Home>/config 디렉터리에 있는 *IntroscopeWorkstation.properties* 파일을 엽니다.
2. [SOA 특정 Workstation 속성 정보](#) (페이지 407)에 나열된 속성에서 *IntroscopeWorkstation.properties* 파일에 추가할 속성을 찾습니다.
3. *com.wily.introscope* 접두사를 포함한 속성의 전체 이름을 *IntroscopeWorkstation.properties* 파일에 추가합니다. 예:
`com.wily.introscope.soa.dashboard.typeviewer.mostcritical.count`
4. 필요에 맞게 속성 값을 설정합니다. 예를 들어 이 속성의 기본 설정을 사용하려면 다음과 같이 지정합니다.
`com.wily.introscope.soa.dashboard.typeviewer.mostcritical.count=10`
5. *IntroscopeEnterpriseManager.properties* 파일을 저장하고 닫습니다.

Workstation 속성이 *IntroscopeWorkstation.properties* 파일에 추가된 후에는 필요에 따라 속성 값을 구성할 수 있습니다.

Workstation 구성 속성 값을 변경하려면

1. <EM_Home>/config 디렉터리에 있는 *IntroscopeWorkstation.properties* 파일을 엽니다.
2. 변경할 속성을 찾아서 환경에 적절한 새 값을 설정합니다. 예:
`com.wily.introscope.soa.dashboard.typeviewer.mostcritical.count=15`
이 설정은 예제일 뿐이며 CA APM for SOA 권장 설정이 아닙니다.
3. *IntroscopeWorkstation.properties* 파일을 저장하고 닫습니다.
4. Workstation 을 다시 시작하여 속성 변경 사항을 적용합니다.

SOA 특정 Workstation 속성 정보

IntroscopeWorkstation.properties 파일에서 다음의 SOA 특정 속성을 설정할 수 있습니다. 모든 속성은 *com.wily.introscope*로 시작하지만 읽기 편리하도록 이 접두사 없이 나열됩니다.

참고: 다른 Workstation 속성의 설정에 대한 자세한 내용은 *CA APM* 구성 및 관리 안내서를 참조하십시오.

[soa.dependencymap.ui.view.nodecount](#) (페이지 408)

SOA 종속성 맵에 표시할 최대 맵 노드 수를 지정합니다.

[com.wily.introscope.soa.dependencymap.ui.view.edgecount](#) (페이지 409)

SOA 종속성 맵에 표시할 최대 맵 가장자리 수를 지정합니다.

[soa.dashboard.typeviewer.average.enable](#) (페이지 397)

선택된 작업에 대한 "평균 응답 시간 편차" 그래프를 표시하거나 표시하지 않습니다.

[soa.dashboard.typeviewer.response.enable](#) (페이지 399)

선택된 작업에 대한 "간격당 응답 편차" 그래프를 표시하거나 표시하지 않습니다.

[soa.dashboard.typeviewer.errors.enable](#) (페이지 398)

선택된 작업에 대한 "간격당 오류 편차" 그래프를 표시하거나 표시하지 않습니다.

[soa.dashboard.typeviewer.mostcritical.enable](#) (페이지 400)

"가장 중요한 작업" 대시보드 및 "가장 중요한" 탭을 표시하거나 표시하지 않습니다.

[soa.dashboard.typeviewer.mostcritical.count](#) (페이지 401)

보고되는 중요 작업의 최대 수를 지정합니다.

[soa.dashboard.typeviewer.mostdependent.enable](#) (페이지 402)

"가장 종속된 작업" 대시보드 및 "가장 종속된" 탭을 표시하거나 표시하지 않습니다.

[soa.dashboard.typeviewer.mostdependent.count](#) (페이지 403)

보고되는 종속 작업의 최대 수를 지정합니다.

[workstation.soa.dependencymap.fetchmetrics](#) (페이지 418)

SOA 종속성 맵의 메트릭 표시를 제어합니다.

workstation.traceview.crossprocess.duration.full (페이지 419)

트랜잭션 추적에 전체 기간 시간을 사용하는지 여부를 결정합니다.

workstation.traceview.crossprocess.duration.net (페이지 420)

트랜잭션 추적에 순 기간 시간을 사용하는지 여부를 결정합니다.

대개의 경우 속성 변경 사항을 적용하려면 Workstation 을 다시 시작해야 합니다. 속성이 *IntroscopeWorkstation.properties* 파일과 *IntroscopeEnterpriseManager.properties* 파일에 정의된 경우에는 *IntroscopeWorkstation.properties* 속성 설정이 사용됩니다.

soa.dependencymap.ui.view.nodecount

이 속성은 Workstation 의 "SOA 종속성 맵"에 표시되는 최대 맵 노드 수를 지정합니다.

Investigator 노드를 선택한 경우 "SOA 종속성 맵" 노드의 수가 com.wily.introscope.soa.dependencymap.ui.view.nodecount 값이 초과하면 오류 메시지가 표시됩니다. 이때 "SOA 종속성 맵"은 표시되지 않습니다.

새 컨텍스트를 선택한 경우 SOA 종속성 맵 노드의 수가 com.wily.introscope.soa.dependencymap.ui.view.nodecount 값이 초과하면 오류 메시지가 표시되고 "SOA 종속성 맵"이 이전 뷰로 돌아갑니다. 예를 들어 "에이전트"에 대한 물리적 뷰를 "서비스" 뷰에 대한 물리적 뷰로 변경한 경우 노드 수 제한이 초과하면 "SOA 종속성 맵"에 오류 메시지가 표시되고 에이전트에 대한 물리적 뷰로 돌아갑니다.

ShowAllOperations 또는 ShowAllServices 를 사용하여 표시된 정보를 확장한 경우 노드 수 값이 초과하면 SOA 종속성 맵에 오류 메시지가 표시되고 최근에 추가된 SOA 종속성 맵 노드가 표시됩니다.

속성 설정

속성은 0 보다 큰 모든 정수로 설정할 수 있습니다.

기본값

200 개 맵 노드

예

```
com.wily.introscope.soa.dependencymap.ui.view.nodecount=200
```

참고

이 속성의 변경 사항을 적용하려면 Workstation 을 다시 시작해야 합니다.

com.wily.introscope.soa.dependencymap.ui.view.edgecount

이 속성은 Workstation 의 "SOA 종속성 맵"에 표시되는 최대 맵 가장자리 수를 지정합니다.

Investigator 노드를 선택한 경우 "SOA 종속성 맵" 가장자리 수가 com.wily.introscope.soa.dependencymap.ui.view.edgecount 속성 값으로 초과하면 오류 메시지가 표시됩니다. 이때 "SOA 종속성 맵"은 표시되지 않습니다.

새 컨텍스트를 선택한 경우 "SOA 종속성 맵" 가장자리 수가 com.wily.introscope.soa.dependencymap.ui.view.edgecount 속성 값으로 초과하면 오류 메시지가 표시되고 "SOA 종속성 맵"이 이전 뷰로 돌아가며 "SOA 종속성 맵"이 표시되지 않습니다. 예를 들어 "에이전트"에 대한 물리적 뷰에서 "서비스" 뷰에 대한 물리적 뷰로 변경한 경우 노드 가장자리 수 제한이 초과되면 "SOA 종속성 맵"에 오류 메시지가 표시되고 "에이전트"에 대한 물리적 뷰로 돌아가며 맵이 표시되지 않습니다.

ShowAllOperations 또는 ShowAllServices 를 사용하여 표시된 정보를 확장한 경우 가장자리 수 값이 초과되면 "SOA 종속성 맵"에 오류 메시지가 표시되고 최근에 추가된 "SOA 종속성 맵" 가장자리가 표시됩니다.

SOA 종속성 맵이 응용 프로그램 성능에 미치는 영향을 방지하기 위해 1000에서 400으로 기본값을 줄여서 맵의 크기와 복잡성을 제한할 수 있습니다.

속성 설정

속성은 0 보다 큰 모든 정수로 설정할 수 있습니다.

기본값

1000

권장되는 값

400

예

com.wily.introscope.soa.dependencymap.ui.view.edgecount=250

참고

이 속성의 변경 사항을 적용하려면 먼저 Workstation 응용 프로그램을 다시 시작해야 합니다.

soa.dashboard.typeviewer.average.enable

선택된 작업에 대해 수집된 편차 데이터를 다음 위치의 "평균 응답 시간 편차" 그래프에 표시합니다.

- "SOA 성능 - 가장 중요한 작업" 대시보드
- "SOA 성능 - 가장 종속된 작업" 대시보드
- "가장 중요한" 탭
- "가장 종속된" 탭

편차 데이터는 에이전트를 통해 Enterprise Manager에 보고되고, Workstation에 표시하도록 허용된 경우에만 대시보드 및 탭에서 사용할 수 있습니다. 편차 데이터의 보고에 대한 자세한 내용은 ["편차 탭에서 편차 메트릭 보기"](#) (페이지 66)를 참조하십시오.

속성 설정

속성은 true 또는 false로 설정할 수 있습니다.

- True - 선택된 작업의 데이터를 "평균 응답 시간 편차" 그래프에 표시합니다.
- False - 선택된 작업의 데이터를 "평균 응답 시간 편차" 그래프에 표시하지 않습니다.

기본값

True

예

```
com.wily.introscope.soa.dashboard.typeviewer.average.enable=true
```

참고

이 속성은 *IntroscopeEnterpriseManager.properties* 파일 또는 *IntroscopeWorkstation.properties* 파일에서 정의할 수 있습니다. 속성이 *IntroscopeWorkstation.properties* 파일과 *IntroscopeEnterpriseManager.properties* 파일에 모두 정의된 경우에는 *IntroscopeWorkstation.properties* 속성 설정이 사용됩니다.

이 속성의 변경 사항을 적용하려면 Enterprise Manager를 다시 시작해야 합니다.

soa.dashboard.typeviewer.errors.enable

선택된 작업에 대해 수집된 편차 데이터를 다음 위치의 "간격당 오류 편차" 그래프에 표시합니다.

- "SOA 성능 - 가장 중요한 작업" 대시보드
- "SOA 성능 - 가장 종속된 작업" 대시보드
- "가장 중요한" 탭
- "가장 종속된" 탭

편차 데이터는 에이전트를 통해 Enterprise Manager에 보고되고, Workstation에 표시하도록 허용된 경우에만 대시보드 및 탭에서 사용할 수 있습니다. 편차 데이터의 보고에 대한 자세한 내용은 ["편차 탭에서 편차 메트릭 보기"](#) (페이지 66)를 참조하십시오.

속성 설정

속성은 true 또는 false로 설정할 수 있습니다.

- True - 선택된 작업의 데이터를 "간격당 오류 편차" 그래프에 표시합니다.
- False - 선택된 작업의 데이터를 "간격당 오류 편차" 그래프에 표시하지 않습니다.

기본값

True

예

```
com.wily.introscope.soa.dashboard.typeviewer.response.enable=true
```

참고

이 속성은 *IntroscopeEnterpriseManager.properties* 파일 또는 *IntroscopeWorkstation.properties* 파일에서 정의할 수 있습니다. 이 속성이 *IntroscopeWorkstation.properties* 파일과 *IntroscopeEnterpriseManager.properties* 파일 모두에 정의된 경우에는 *IntroscopeWorkstation.properties* 속성 설정이 사용됩니다.

이 속성의 변경 사항을 적용하려면 Enterprise Manager를 다시 시작해야 합니다.

soa.dashboard.typeviewer.response.enable

선택된 작업에 대해 수집된 편차 데이터를 다음 위치의 "간격당 응답 편차" 그래프에 표시합니다.

- "SOA 성능 - 가장 중요한 작업" 대시보드
- "SOA 성능 - 가장 종속된 작업" 대시보드
- "가장 중요한" 탭
- "가장 종속된" 탭

편차 데이터는 에이전트를 통해 Enterprise Manager에 보고되고, Workstation에 표시하도록 허용된 경우에만 대시보드 및 탭에서 사용할 수 있습니다. 편차 데이터의 보고에 대한 자세한 내용은 ["편차 탭에서 편차 메트릭 보기"](#) (페이지 66)를 참조하십시오.

속성 설정

속성은 true 또는 false로 설정할 수 있습니다.

- True - 선택된 작업의 데이터를 "간격당 응답 편차" 그래프에 표시합니다.
- False - 선택된 작업의 데이터를 "간격당 응답 편차" 그래프에 표시하지 않습니다.

기본값

True

예

```
com.wily.introscope.soa.dashboard.typeviewer.response.enable=true
```

참고

이 속성은 *IntroscopeEnterpriseManager.properties* 파일 또는 *IntroscopeWorkstation.properties* 파일에서 정의할 수 있습니다. 이 속성이 *IntroscopeWorkstation.properties* 파일과 *IntroscopeEnterpriseManager.properties* 파일 모두에 정의된 경우에는 *IntroscopeWorkstation.properties* 속성 설정이 사용됩니다.

이 속성의 변경 사항을 적용하려면 Enterprise Manager를 다시 시작해야 합니다.

soa.dashboard.typeviewer.mostcritical.enable

"가장 중요한 작업" 대시보드를 "콘솔"에 표시하고 "가장 중요한" 탭을 Investigator에 표시합니다.

속성 설정

속성은 true 또는 false로 설정할 수 있습니다.

- True - "SOA 성능 - 가장 중요한 작업" 대시보드 옵션을 "콘솔" 대시보드 메뉴에 표시하고 "가장 중요한" 탭을 Investigator에 표시합니다.
- False - "SOA 성능 - 가장 중요한 작업" 대시보드 및 "가장 중요한" 탭에 액세스할 수 없습니다.

이 속성을 false로 설정하면 Investigator의 모든 노드에 대해 "가장 중요한" 탭이 표시되지 않습니다. "가장 중요한 작업" 대시보드는 "콘솔" 대시보드 드롭다운 목록에 계속 나열되지만 선택하더라도 대시보드가 표시되지 않습니다. "가장 중요한 작업" 대시보드 대신 대시보드를 사용할 수 없다는 오류 메시지가 나타납니다.

기본값

True

예

```
com.wily.introscope.soa.dashboard.typeviewer.mostcritical.enable=true
```

참고

이 속성은 IntroscopeEnterpriseManager.properties 파일 또는 IntroscopeWorkstation.properties 파일에서 정의할 수 있습니다. 이 속성이 IntroscopeWorkstation.properties 파일과 IntroscopeEnterpriseManager.properties 파일 모두에 정의된 경우에는 IntroscopeWorkstation.properties 속성 설정이 사용됩니다.

이 속성의 변경 사항을 적용하려면 Enterprise Manager와 Workstation을 다시 시작해야 합니다.

soa.dashboard.typeviewer.mostcritical.count

"SOA 성능 - 가장 중요한 작업" 대시보드 및 "가장 중요한" 탭에서 모든 에이전트가 보고하는 중요한 작업의 총 수를 결정합니다.

속성 설정

최대값은 0입니다.

최대값은 기본값이 25인 *com.wily.introscope.soa.deviation.count.per.metric* 속성의 값보다 작거나 같아야 합니다.

기본값

10 개 작업

예

```
com.wily.introscope.soa.dashboard.typeviewer.mostcritical.count=5
```

참고

이 속성은 *IntroscopeEnterpriseManager.properties* 파일 또는 *IntroscopeWorkstation.properties* 파일에서 정의할 수 있습니다. 이 속성이 *IntroscopeEnterpriseManager.properties* 파일과 *IntroscopeWorkstation.properties* 파일 모두에 정의된 경우에는 *IntroscopeWorkstation.properties* 속성 설정이 사용됩니다.

이 속성의 변경 사항을 적용하려면 Enterprise Manager 와 Workstation 을 다시 시작해야 합니다.

soa.dashboard.typeviewer.mostdependent.enable

"가장 종속된 작업" 대시보드를 "콘솔"에 표시하고 "가장 종속된" 탭을 Investigator에 표시합니다.

속성 설정

속성은 true 또는 false로 설정할 수 있습니다.

- True - "SOA 성능 - 가장 종속된 작업" 대시보드 옵션을 "콘솔" 대시보드 메뉴에 표시하고 "가장 종속된" 탭을 Investigator에 표시합니다.
- False - "SOA 성능 - 가장 종속된 작업" 대시보드 및 "가장 종속된" 탭에 액세스할 수 없습니다.

이 속성을 false로 설정하면 Investigator에 "가장 종속된" 탭이 표시되지 않습니다. "가장 종속된 작업" 대시보드는 "콘솔" 대시보드 드롭다운 목록에 계속 나열되지만 선택하더라도 대시보드가 표시되지 않습니다. "가장 종속된 작업" 대시보드 대신 대시보드를 사용할 수 없다는 오류 메시지가 나타납니다.

기본값

True

예

```
com.wily.introscope.soa.dashboard.typeviewer.mostdependent.enable=true
```

참고

이 속성은 *IntroscopeEnterpriseManager.properties* 파일 또는 *IntroscopeWorkstation.properties* 파일에서 정의할 수 있습니다. 이 속성이 *IntroscopeEnterpriseManager.properties* 파일과 *IntroscopeWorkstation.properties* 파일 모두에 정의된 경우에는 *IntroscopeWorkstation.properties* 속성 설정이 사용됩니다.

이 속성의 변경 사항을 적용하려면 Enterprise Manager와 Workstation을 다시 시작해야 합니다.

soa.dashboard.typeviewer.mostdependent.count

"SOA 성능 - 가장 종속된 작업" 대시보드 및 "가장 종속된" 탭에서 에이전트가 보고하는 가장 종속된 작업의 총 수를 확인합니다.

속성 설정

최대값은 0입니다.

최대값은 기본값이 25인 com.wily.introscope.soa.deviation.count.per.metric 속성의 값보다 작거나 같아야 합니다.

기본값

10 개 작업

예

```
com.wily.introscope.soa.dashboard.typeviewer.mostdependent.count=5
```

참고

이 속성은 IntroscopeEnterpriseManager.properties 파일 또는 IntroscopeWorkstation.properties 파일에서 정의할 수 있습니다. 이 속성이 IntroscopeEnterpriseManager.properties 파일과 IntroscopeWorkstation.properties 파일 모두에 정의된 경우에는 IntroscopeWorkstation.properties 속성 설정이 사용됩니다.

이 속성의 변경 사항을 적용하려면 Enterprise Manager 와 Workstation 을 다시 시작해야 합니다.

workstation.soa.dependencymap.fetchmetrics

SOA 종속성 맵의 메트릭 표시를 제어합니다.

속성 설정

속성은 true 또는 false로 설정할 수 있습니다.

- True - Enterprise Manager에서 선택한 기본 메트릭 및 도구 설명 메트릭을 요청합니다. 이 속성을 true로 설정하면 SOA 종속성 맵에 모든 맵 노드에 대한 기본 메트릭이 표시됩니다. 맵 노드 위로 마우스를 이동하면 도구 설명에 추가 메트릭이 표시됩니다.
- False - SOA 종속성 맵 노드에 기본 메트릭을 표시하거나 도구 설명 창에 추가 메트릭을 표시하지 않습니다.

기본값

True

예

```
com.wily.introscope.workstation.soa.dependencymap.fetchmetrics=true
```

참고

이 속성의 변경 사항을 적용하려면 Workstation을 다시 시작해야 합니다.

workstation.traceview.crossprocess.duration.full

트랜잭션 추적 뷰어의 "시퀀스 뷰" 탭에서 트랜잭션 추적의 전체 기간 시간을 표시하고 추적의 기본값으로 사용할지 여부를 결정합니다.

전체 기간 시간은 프로세스의 시작부터 끝까지의 시간을 사용하여 계산되며 다른 프로세스가 실행될 때 프로세스가 대기한 시간을 포함합니다. 이 방법을 사용하면 동기 트랜잭션에서 루트 추적이 항상 가장 느립니다. 순 기간은 추적의 전체 기간 시간에서 추적의 비동기 자식의 기간 시간을 제하여 계산합니다.

Workstation 속성 파일을 변경하지 않은 경우 기본적으로 전체 기간이 표시되며 "시퀀스 뷰" 탭에서 전체 기간 시간 또는 순 기간 시간을 표시하도록 선택할 수 있습니다.

속성 설정

Workstation 속성 파일에

introscope.workstation.traceview.crossprocess.duration.full 속성만 추가한 경우 "시퀀스 뷰" 탭에 전체 기간만 표시됩니다.

Workstation 속성 파일에

introscope.workstation.traceview.crossprocess.duration.full 및 *introscope.workstation.traceview.crossprocess.duration.net* 을 모두 추가하면 "시퀀스 뷰" 탭에 전체 기간과 순 기간을 모두 표시할 수 있으며 기본값은 값이 가장 작은 속성 값으로 결정됩니다.

예를 들어 전체 기간이나 순 기간을 선택할 수 있도록 허용하고 기본적으로 순 기간을 표시하고 싶다면 Workstation 속성 파일에서 다음과 같이 속성을 설정합니다.

```
com.wily.introscope.workstation.traceview.crossprocess.duration.full=2
com.wily.introscope.workstation.traceview.crossprocess.duration.net=1
```

기본값

1 (전체 기간이 표시됨)

예

```
com.wily.introscope.workstation.traceview.crossprocess.duration.full=1
```

참고

이 속성의 변경 사항을 적용하려면 Workstation 을 다시 시작해야 합니다.

workstation.traceview.crossprocess.duration.net

트랜잭션 추적 뷰어의 "시퀀스 뷰" 탭에서 트랜잭션 추적의 순 기간 시간을 표시하고 추적의 기본값으로 사용할지 여부를 결정합니다.

순 기간은 추적의 전체 기간 시간에서 추적의 비동기 자식의 기간 시간을 제하여 계산합니다. 이 방법의 주요 이점은 동기 트랜잭션의 린 스레드를 즉시 알 수 있다는 것입니다.

Workstation 속성 파일을 변경하지 않은 경우 기본적으로 전체 기간이 표시되며 "시퀀스 뷰" 탭에서 전체 기간 시간 또는 순 기간 시간을 표시하도록 선택할 수 있습니다.

속성 설정

Workstation 속성 파일에

introscope.workstation.traceview.crossprocess.duration.net 속성만 추가한 경우 "시퀀스 뷰" 탭에 순 기간만 표시됩니다.

Workstation 속성 파일에

introscope.workstation.traceview.crossprocess.duration.full 및 *introscope.workstation.traceview.crossprocess.duration.net* 을 모두 추가하면 "시퀀스 뷰" 탭에 전체 기간과 순 기간을 모두 표시할 수 있으며 기본값은 값이 가장 작은 속성 값으로 결정됩니다.

예를 들어 전체 기간이나 순 기간을 선택할 수 있도록 허용하고 기본적으로 순 기간을 표시하고 싶다면 Workstation 속성 파일에서 다음과 같이 속성을 설정합니다.

```
com.wily.introscope.workstation.traceview.crossprocess.duration.full=2  
com.wily.introscope.workstation.traceview.crossprocess.duration.net=1
```

기본값

2(순 기간이 기본적으로 표시되지 않음)

예

```
com.wily.introscope.workstation.traceview.crossprocess.duration.net=1
```

참고

이 속성의 변경 사항을 적용하려면 Workstation 을 다시 시작해야 합니다.

부록 D: SOA 특정 WebView 구성 속성

WebView 구성 속성을 사용하면 WebView 응용 프로그램의 동작 및 작업을 제어할 수 있습니다. 또한 이 속성을 사용하면 WebView 를 사용자 환경에 가장 적합하게 사용자 지정할 수 있습니다. WebView 속성은 CA APM for SOA 를 구체적으로 구성할 수 있도록 해 줍니다.

이 섹션은 다음 항목을 포함하고 있습니다.

[WebView 속성 구성](#) (페이지 421)

[SOA 특정 WebView 속성 정보](#) (페이지 422)

WebView 속성 구성

CA APM for SOA 에는 기본 설정이 적용된 SOA 특정 WebView 속성이 있습니다. 구성을 시작하기 전에 속성 이름을 *IntroscopeWebView.properties* 파일에 추가하십시오.

WebView 속성을 *IntroscopeWebView.properties*에 추가하려면

1. *<EM_Home>/config* 디렉터리에 있는 *IntroscopeWebView.properties* 파일을 엽니다.
2. *IntroscopeWebView.properties* 파일에 추가할 [SOA 특정 WebView 속성](#) (페이지 422)을 찾습니다.
3. *com.wily.introscope* 접두사를 포함한 속성의 전체 이름을 *IntroscopeWebView.properties* 파일에 추가합니다. 예:
`com.wily.introscope.soa.dependencymap.ui.view.edgecount`
4. 필요에 맞게 속성 값을 설정합니다. 예를 들어 이 속성의 기본 설정을 사용하려면 다음과 같이 지정합니다.
`com.wily.introscope.soa.dependencymap.ui.view.edgecount=1000`
5. *IntroscopeWebView.properties* 파일을 저장하고 닫습니다.

WebView 속성은 *IntroscopeWebView.properties* 파일에 추가된 후에는 필요에 따라 속성 값을 구성할 수 있습니다.

WebView 구성 속성 값을 변경하려면

1. <EM_Home>/config 디렉터리에 있는 *IntroscopeWebView.properties* 파일을 엽니다.
2. 변경할 속성을 찾아서 환경에 적절한 새 값을 설정합니다. 예: com.wily.introscope.soa.dependencymap.ui.view.edgecount=250
이 설정은 예제일 뿐이며 CA APM for SOA 권장 설정이 아닙니다.
3. *IntroscopeWebView.properties* 파일을 저장하고 닫습니다.
4. WebView 응용 프로그램을 다시 시작하여 속성 변경 사항을 적용합니다.

SOA 특정 WebView 속성 정보

IntroscopeWebView.properties 파일에서 다음의 SOA 특정 속성을 설정할 수 있습니다. 모든 속성은 *com.wily.introscope*로 시작하지만 읽기 편리하도록 이 접두사 없이 나열됩니다.

참고: 다른 WebView 속성 설정에 대한 자세한 내용은 *CA APM* 구성 및 관리 안내서를 참조하십시오.

[soa.dependencymap.ui.view.nodecount](#) (페이지 423)

WebView의 "SOA 종속성 맵"에 표시되는 최대 맵 노드 수를 지정합니다.

[com.wily.introscope.soa.dependencymap.ui.view.edgecount](#) (페이지 424)

WebView의 "SOA 종속성 맵"에 표시되는 최대 맵 가장자리 수를 지정합니다.

soa.dependencymap.ui.view.nodecount

이 속성은 WebView 의 "SOA 종속성 맵"에 표시되는 최대 맵 노드 수를 지정합니다.

Investigator 노드를 선택한 경우 "SOA 종속성 맵" 노드 수가 com.wily.introscope.soa.dependencymap.ui.view.nodecount 속성 값을 초과하면 오류 메시지가 표시됩니다. 이 때 "SOA 종속성 맵"은 표시되지 않습니다.

새 컨텍스트를 선택한 경우 "SOA 종속성 맵" 노드 수가 com.wily.introscope.soa.dependencymap.ui.view.nodecount 속성 값을 초과하면 오류 메시지가 표시되고 "SOA 종속성 맵"이 표시되지 않습니다. 예를 들어 "에이전트"에 대한 물리적 뷰에서 "서비스" 뷰에 대한 물리적 뷰로 변경한 경우 노드 수 제한이 초과되면 "SOA 종속성 맵"에 오류 메시지가 표시되고 맵이 표시되지 않습니다.

ShowAllOperations 또는 ShowAllServices 를 사용하여 표시된 정보를 확장한 경우 노드 수 값이 초과하면 SOA 종속성 맵에 오류 메시지가 표시되고 최근에 추가된 SOA 종속성 맵 노드가 표시됩니다.

SOA 종속성 맵이 응용 프로그램 성능에 미치는 영향을 방지하기 위해 200 에서 더 낮은 값으로 속성 기본값을 줄여서 맵의 크기와 복잡성을 제한할 수 있습니다.

속성 설정

속성은 0 보다 큰 모든 정수로 설정할 수 있습니다.

기본값

200 개 맵 노드

권장되는 값

200 개 맵 노드

예

```
com.wily.introscope.soa.dependencymap.ui.view.nodecount=200
```

참고

이 속성의 변경 사항을 적용하려면 먼저 WebView 응용 프로그램을 다시 시작해야 합니다.

com.wily.introscope.soa.dependencymap.ui.view.edgecount

이 속성은 WebView 의 "SOA 종속성 맵"에 표시되는 최대 맵 가장자리 수를 지정합니다.

Investigator 노드를 선택한 경우 "SOA 종속성 맵" 가장자리 수가 com.wily.introscope.soa.dependencymap.ui.view.edgecount 값을 초과하면 오류 메시지가 표시됩니다. 이때 "SOA 종속성 맵"은 표시되지 않습니다.

새 컨텍스트를 선택한 경우 "SOA 종속성 맵" 가장자리 수가 com.wily.introscope.soa.dependencymap.ui.view.edgecount 값을 초과하면 오류 메시지가 표시되고 "SOA 종속성 맵"이 표시되지 않습니다. 예를 들어 "에이전트"에 대한 물리적 뷰를 "서비스" 뷰에 대한 물리적 뷰로 변경한 경우 가장자리 수 제한이 초과되면 "SOA 종속성 맵"에 오류 메시지가 표시되고 맵이 표시되지 않습니다.

ShowAllOperations 또는 ShowAllServices 를 사용하여 표시된 정보를 확장한 경우 가장자리 수 값이 초과되면 "SOA 종속성 맵"에 오류 메시지가 표시되고 최근에 추가된 "SOA 종속성 맵" 가장자리가 표시됩니다.

SOA 종속성 맵이 응용 프로그램 성능에 미치는 영향을 방지하기 위해 1000에서 250으로 기본값을 줄여서 맵의 크기와 복잡성을 제한할 수 있습니다.

속성 설정

속성은 0 보다 큰 모든 정수로 설정할 수 있습니다.

기본값

1000 개의 맵 가장자리

권장되는 값

250 개의 맵 가장자리

예

com.wily.introscope.soa.dependencymap.ui.view.edgecount=250

참고

이 속성의 변경 사항을 적용하려면 먼저 WebView 응용 프로그램을 다시 시작해야 합니다.