

Rational Build Forge



# Information Center

버전 7.1.3.5 이 정보는 개발 중이며 언제든지 변경될 수 있습니다.  
릴리스 버전은 [ibm.com](http://ibm.com)에서 사용할 수 있습니다.

참고

이 정보와 이 정보가 지원하는 제품을 사용하기 전에, 647 페이지의 『주의사항』의 정보를 읽으십시오.

이 개정판은 새 개정판에 별도로 명시하지 않는 한 Rational Build Forge 버전 7.1.3.5 및 모든 후속 릴리스와 수정에 적용됩니다.

© Copyright IBM Corporation 2003, 2013.

# **목차**

제 1 장 IBM Rational Build Forge 정보 . . . . .	1
Build Forge 제품 개정판 . . . . .	1
프로세스 자동화 . . . . .	1
Build Forge 개념 및 오브젝트 . . . . .	3
제 2 장 Build Forge 시작하기 . . . . .	7
콘솔 액세스 및 사용 . . . . .	7
목록 필터링 및 정렬 . . . . .	10
Hello world 프로젝트 작성 . . . . .	11
서버 설정 . . . . .	11
프로젝트 정의 . . . . .	14
작업 실행 . . . . .	14
작업 결과 보기 . . . . .	15
프로젝트 샘플 . . . . .	16
제 3 장 IBM Rational Build Forge 문서 주의사항 . . . . .	17
제 4 장 요구사항 . . . . .	21
하드웨어, 소프트웨어 및 데이터베이스 요구사항 . . . . .	21
데이터베이스 요구사항 . . . . .	21
Installation Manager 요구사항 . . . . .	22
제품 개정판에 대한 라이선싱 요구사항 . . . . .	22
설치 중에 라이선스 파일 지정 . . . . .	22
Build Forge의 Rational License Server 구성 . . . . .	23
라이선스 키 얻기 및 Rational Licensing Server . . . . .	25
설정 . . . . .	25
IPv6 지원에 대한 네트워킹 요구사항 . . . . .	26
자국어 지원 요구사항 . . . . .	28
관리 콘솔 및 에이전트에 대한 언어 설정 . . . . .	28
데이터베이스 호스트에 대한 국제적 데이터 지원 . . . . .	29
관리 콘솔에서 사용자 언어 환경 설정 변경 . . . . .	29
UNIX/Linux 호스트의 언어/문자 세트 판별 . . . . .	29
Windows 호스트의 언어 코드 페이지 판별 . . . . .	30
제 5 장 설치 계획 . . . . .	31
컴포넌트 . . . . .	31
배치 유형 . . . . .	32
설치 방법 . . . . .	33
구성 옵션 . . . . .	33
기타 제품과 통합 . . . . .	33
제 6 장 설치 전 설정 . . . . .	35

국제적 데이터 설정 . . . . .	35
데이터베이스 백업(필수) . . . . .	37
데이터베이스 설정 . . . . .	37
DB2 Express 설정 . . . . .	37
DB2 설정 . . . . .	39
Microsoft SQL Server 설정 . . . . .	43
MySQL 설정 . . . . .	48
Oracle 설정 . . . . .	52
보안 설정 . . . . .	61
제 7 장 관리 콘솔 설치 . . . . .	63
런치패드를 사용하여 Installation Manager 시작 . . . . .	64
제품 DVD에서 런치패드 시작 . . . . .	64
다운로드한 패키지에서 런치패드 시작 . . . . .	65
Installation Manager에서 설치 단계 . . . . .	66
설치 후 점검 목록 . . . . .	71
Linux용 파일 핸들 수 늘리기 . . . . .	72
제 8 장 대체 설치 방법 . . . . .	75
자체 컴포넌트를 사용하여 설치 . . . . .	75
전제조건 . . . . .	75
국제적 데이터 지원 . . . . .	76
데이터베이스 설치 및 구성 . . . . .	77
Apache HTTP Server 설치 및 구성 . . . . .	78
PHP 설치 및 구성 . . . . .	80
Apache Tomcat 설치 및 구성 . . . . .	84
Installation Manager 수동으로 설치 . . . . .	86
Installation Manager 시작 . . . . .	86
저장소 URL 지정 . . . . .	87
제품 컴포넌트의 자동 설치 수행 . . . . .	88
Installation Manager로 응답 파일 작성 . . . . .	88
자동 모드에서 Installation Manager 설치 및 실행 . . . . .	89
제품 컴포넌트의 자동 업그레이드 수행 . . . . .	90
Installation Manager로 업데이트 응답 파일 작성 . . . . .	90
자동 모드에서 업데이트 설치 실행 . . . . .	91
VMware에 Build Forge 시스템 설치 . . . . .	92
System z의 Linux에 관리 콘솔 설치 . . . . .	92
제 9 장 관리 콘솔에서 추가 기능 구성 . . . . .	99
Build Forge 구성 파일(buildforge.conf) . . . . .	99
buildforge.conf 파일 업데이트 . . . . .	99
Buildforge.conf 참조 . . . . .	101

대체 포트를 사용하도록 관리 콘솔 구성 . . . . .	102
중복 구성 . . . . .	103
중복 정보 . . . . .	103
백업 시스템 설치 . . . . .	103
중복 작업 . . . . .	105
IPv6 네트워크 지원 사용 . . . . .	105
httpd.conf 수정 . . . . .	105
FLEXlm 클라이언트 설정 . . . . .	106
보안 기능 . . . . .	106
싱글 사인온 구현 . . . . .	107
SSL 및 HTTPS 사용 . . . . .	123
비밀번호 암호화 사용 . . . . .	147
<b>제 10 장 에이전트 설치.</b> . . . .	161
Windows 플랫폼에 에이전트 설치 . . . . .	161
사용자 모드 에이전트 . . . . .	162
Windows 운영 체제에서 자동 에이전트 설치 수 행 . . . . .	163
UNIX 및 Linux 시스템에 에이전트 설치 . . . . .	163
System i 플랫폼에 에이전트 설치 . . . . .	165
System z 플랫폼에 에이전트 설치 및 실행 . . . . .	167
z/OS에서 에이전트 설치 문제점 해결 . . . . .	169
에이전트 업데이트 및 배치 유틸리티 . . . . .	170
지원되는 운영 체제 . . . . .	171
설치 . . . . .	171
UNIX 또는 Linux를 실행하는 대상 컴퓨터의 연결 설정 . . . . .	172
Windows를 실행하는 대상 컴퓨터의 연결 설정 사용법 개요. . . . .	175
XML 파일에서 대상 시스템 지정 . . . . .	176
에이전트 업데이트 유틸리티 실행. . . . .	180
agentupdate 유틸리티 참조. . . . .	182
XML 파일 매개변수, 명령행 옵션 및 환경 변수 의 우선순위. . . . .	183
로그 파일을 통해 유틸리티 조치 보기 . . . . .	184
문제점 해결. . . . .	184
에이전트 실행 . . . . .	190
Windows에서 에이전트 실행 . . . . .	190
UNIX, Linux 및 MacOS에서 에이전트 실행 System i에서 에이전트 실행 . . . . .	191
bfagent 참조 . . . . .	193
에이전트 구성 . . . . .	193
에이전트 구성 파일 찾기 . . . . .	193
에이전트 포트 변경 . . . . .	194
다른 셸 구성 . . . . .	194
네트워크 공유 파일 시스템에서 에이전트 명령 실행(Windows) . . . . .	194

에이전트 간 직접 파일 전송 구성 . . . . .	196
트리거 변수 및 에이전트 성능. . . . .	196
bfagent.conf 참조. . . . .	196
에이전트 문제점 해결. . . . .	207
호스트 이름 해석 테스트 . . . . .	208
연결 테스트. . . . .	208
Windows에서 에이전트의 문제점 해결 . . . . .	210
UNIX, Linux 또는 MacOS에서 에이전트 문제 점 해결 . . . . .	210
<b>제 11 장 설치 후 태스크</b> . . . . .	213
엔진 시작 및 중지 . . . . .	213
Windows에서 엔진 시작 및 중지 . . . . .	213
UNIX 또는 Linux 시스템에서 엔진 시작 및 중 지 . . . . .	213
사용자 설정. . . . .	214
루트 사용자. . . . .	214
사용자 작성 및 편집. . . . .	214
설치 확인 . . . . .	217
서버 인증 . . . . .	217
서버 구성 . . . . .	217
테스트 프로젝트 작성 . . . . .	218
프로젝트 실행 . . . . .	218
로그 위치 및 설정. . . . .	219
공통 문제점 해결 . . . . .	220
<b>제 12 장 이전 버전에서 업그레이드.</b> . . . .	223
업그레이드를 위한 필수 소프트웨어 . . . . .	223
버전 7.1 콘솔 업그레이드 . . . . .	224
업데이트 설치 수행 . . . . .	224
버전 7.1.3.1로 업그레이드 후에 어댑터 복사 버전 7.0.2.x 콘솔 업그레이드 . . . . .	225
에이전트 업그레이드 . . . . .	226
<b>제 13 장 제품 컴포넌트 설치 제거</b> . . . . .	227
Installation Manager를 사용하여 제품 설치 제거 Installation Manager 설치에 실패한 경우 수동 으로 제품 설치 제거 . . . . .	228
Build Forge 파일의 설치 후 제거 정리 . . . . .	228
Installation Manager를 사용하여 제품 다시 설 치 및 기존 DB2 Express 사용 . . . . .	229
Windows Build Forge 에이전트 설치 제거 . . . . .	230
UNIX 또는 Linux Build Forge 에이전트 설치 제 거 . . . . .	230
<b>제 14 장 릴리스 정보</b> . . . . .	233
릴리스 정보 - <i>IBM Rational Build Forge 버전</i> 7.1.3.5 . . . . .	233

제 15 장 대시보드 탭 . . . . .	237
제 16 장 관리 . . . . .	239
관리 정보 . . . . .	239
액세스 그룹 . . . . .	239
액세스 개요 . . . . .	241
액세스 예제: 작업을 실행할 수 있는 기능을 그룹에 제공 . . . . .	242
팀 및 프로젝트 보안 계획 . . . . .	243
액세스 특성 관리 . . . . .	243
사용자 . . . . .	244
사용자 특성 . . . . .	244
사용자 관리 . . . . .	246
사용자 유형을 사용하여 액세스 제어 . . . . .	247
계정 잠금 . . . . .	248
루트 사용자 . . . . .	249
API 사용자 . . . . .	249
권한 . . . . .	250
권한 부여 연습 . . . . .	250
LDAP 통합 . . . . .	251
LDAP 통합 정보 . . . . .	251
LDAP 도메인 특성 . . . . .	252
태스크 . . . . .	256
시스템 구성 설정 . . . . .	259
기타 시스템 설정 . . . . .	270
감사 로깅 . . . . .	270
메시지 . . . . .	276
번역된 메시지 . . . . .	277
작업 상태에 대한 RSS 데이터 공급에 등록 . . . . .	277
시스템 메시지에 대한 RSS 데이터 공급 필터링 . . . . .	278
보안 패널 . . . . .	278
키 저장소 패널 . . . . .	281
SSL 패널 . . . . .	282
SSO 패널 . . . . .	283
라이선스 관리 . . . . .	284
새 라이선스 키 입력 . . . . .	285
토큰 라이선스 로그 . . . . .	285
엔진 관리 . . . . .	285
엔진 일시정지 . . . . .	286
엔진 시작 및 중지 . . . . .	286
데이터베이스 관리 . . . . .	287
데이터베이스 로그 파일 삭제 . . . . .	287
오류 메시지 . . . . .	287
활성 단계 없음 . . . . .	287
라이선스 키가 올바르지 않거나 또는 Build Forge 라이선스 키가 손상 또는 누락되었습니다 . . . . .	287
데이터베이스 라이선스 필요 . . . . .	288

제 17 장 서버 . . . . .	289
서버 자원 정보 . . . . .	289
서버 패널 정보 . . . . .	290
서버 작성 . . . . .	291
서버 테스트 . . . . .	292
서버 변경 . . . . .	293
서버 사용 및 사용 안함 . . . . .	293
서버의 동시 작업 수 제한 . . . . .	293
작업 계수 재설정 . . . . .	294
서버에서 작업 계수를 0으로 재설정 . . . . .	294
모든 서버에서 작업 계수를 0으로 재설정 . . . . .	295
서버 상태에 대한 RSS 데이터 피드 . . . . .	295
서버 인증 . . . . .	296
서버 인증 정보 . . . . .	296
서버 인증 작성 . . . . .	297
서버 인증 대체 . . . . .	298
제한된 서버 인증 사용 허용 . . . . .	298
선택기 . . . . .	299
선택기 정보 . . . . .	299
선택기 설정 방법 . . . . .	300
선택기 변수 유형 . . . . .	301
선택기 변수 비교 규칙 . . . . .	302
올바른 서버 자원에 대한 선택기 평가 . . . . .	302
컬렉터 . . . . .	303
컬렉터 정보 . . . . .	303
컬렉터 패널 정보 . . . . .	303
컬렉터 작성 . . . . .	304
Manifest 및 동적 서버 선택 . . . . .	305
Manifest 보기 . . . . .	307
Manifest를 수동으로 새로 고치기 . . . . .	307
서버 Manifest의 업데이트 빈도 설정 . . . . .	307
정적 및 동적 서버 선택 설정 예제 . . . . .	308
특성 참조 . . . . .	309
스냅샷을 사용하여 선택기의 새 인스턴스 작성 . . . . .	314
선택기 스냅샷 개요 . . . . .	314
선택기 스냅샷 계획 . . . . .	316
기존 선택기 또는 선택기 스냅샷에서 선택기 스냅샷 작성 . . . . .	317
기본 선택기 스냅샷 변경 . . . . .	317
선택기 스냅샷의 스냅샷 이름 변경 . . . . .	318
선택기 스냅샷 세트의 스냅샷 액세스 및 보기 . . . . .	319
선택기 스냅샷 삭제 . . . . .	319
제 18 장 환경에 대한 작업 . . . . .	321
환경 정보 . . . . .	321
환경 상속 . . . . .	321
특수 상속 사례 . . . . .	322

작업 시작 시 프로젝트 변수 변경 내용 . . . . .	323
변수 정보 . . . . .	323
단계에서 변수 해석 . . . . .	324
선언되지 않은 변수 해석 . . . . .	325
환경 패널 정보 . . . . .	325
세부사항 탭 . . . . .	326
스냅샷 탭 . . . . .	329
환경 작성 . . . . .	329
변수 사용 . . . . .	329
변수의 풀다운 작성 . . . . .	330
다른 환경 포함 . . . . .	331
단계 실행 중 변수 값 변경 . . . . .	331
Windows 드라이브 맵핑 . . . . .	332
변수에서 점 명령 사용 . . . . .	334
시스템 변수 참조 . . . . .	335
트리거 변수 참조 . . . . .	337
환경 스냅샷 . . . . .	342
환경 스냅샷 개요 . . . . .	342
환경 스냅샷 계획 . . . . .	344
환경 스냅샷 작성 . . . . .	345
기본 환경 스냅샷 변경 . . . . .	345
환경 스냅샷 이름 변경 . . . . .	346
스냅샷 액세스 및 보기 . . . . .	347
환경 스냅샷 삭제 . . . . .	347
<b>제 19 장 프로젝트에 대한 작업 . . . . .</b>	<b>349</b>
프로젝트 정보 . . . . .	349
프로젝트 패널 정보 . . . . .	349
프로젝트 특성 변경 . . . . .	353
프로젝트 복사 . . . . .	355
프로젝트 삭제 . . . . .	356
단계를 서버에 고정 . . . . .	356
체인: 다른 프로젝트 또는 라이브러리의 조건부 실행 . . . . .	357
체인 프로젝트의 환경 변수 상속 . . . . .	358
대기 사용 시 체인 프로젝트 취소 . . . . .	359
태그 정의 . . . . .	359
프로젝트의 태그 형식 편집 . . . . .	360
태그 동기화 . . . . .	360
태그의 시스템 정의 변수 . . . . .	362
태그 변수 작성 또는 편집 . . . . .	362
라이브러리 . . . . .	363
라이브러리 정보 . . . . .	363
라이브러리 복사 . . . . .	364
로그 필터 . . . . .	364
로그 필터 정보 . . . . .	365
로그 필터 작성 . . . . .	365

단계에 로그 필터 지정 . . . . .	366
필터 패턴 . . . . .	366
필터 조치 . . . . .	368
필터 알림 . . . . .	369
오류 임계값 . . . . .	369
오류 및 경고 계수 . . . . .	370
클래스 . . . . .	370
클래스 정보 . . . . .	370
알림 설정 . . . . .	373
알림 템플릿 정보 . . . . .	374
SMTP 서버 구성 . . . . .	377
프로젝트 및 단계의 알림 특성 설정 . . . . .	377
알림 연습 . . . . .	377
알림 템플릿 사용자 정의 . . . . .	378
인라인 프로젝트에 대한 알림 . . . . .	382
스냅샷을 사용하여 프로젝트의 새 인스턴스 작성 . . . . .	382
프로젝트 스냅샷 개요 . . . . .	382
프로젝트 스냅샷 계획 . . . . .	384
프로젝트 스냅샷 옵션 . . . . .	384
스냅샷 권한의 액세스 그룹 확인 및 편집 . . . . .	385
기본 프로젝트 또는 프로젝트 스냅샷에서 프로젝트 스냅샷 작성 . . . . .	386
기본 프로젝트 스냅샷 변경 . . . . .	387
프로젝트 스냅샷의 스냅샷 이름 변경 . . . . .	387
프로젝트 스냅샷 세트의 스냅샷 액세스 및 보기 . . . . .	388
기본 프로젝트 스냅샷에 대한 작업 시작 . . . . .	389
비기본 프로젝트 스냅샷에 대한 작업 시작 . . . . .	389
프로젝트 스냅샷 삭제 . . . . .	390

<b>제 20 장 단계에 대한 작업 . . . . .</b>	<b>393</b>
단계 정보 . . . . .	393
단계 패널 정보 . . . . .	393
단계 추가 . . . . .	398
단계 편집 . . . . .	398
단계 사용 안함 . . . . .	398
추가 단계 조작 . . . . .	399
실행 플로우 제어 . . . . .	399
단계가 실행하는 방법 . . . . .	400
인라인: 프로젝트 또는 라이브러리의 단계 포함 . . . . .	402
단계의 패스 체인 및 실패 체인 . . . . .	403
스레딩: 병렬로 단계 실행 . . . . .	403
다중 서버에 단계 브로드캐스트 . . . . .	404
조건부 단계 실행 . . . . .	406
While 루프 실행 . . . . .	406
조건 함수 . . . . .	407
단계에서 프로젝트 실행 . . . . .	410
로그 출력 사용자 정의 . . . . .	410

단계의 로그 출력 레이블 지정 . . . . .	410
단계 출력을 색상 또는 활성 링크로 강조표시 . . . . .	411
작업 데이터에 대한 작업 . . . . .	412
프로젝트 파일에 빌드 번호 임베드 . . . . .	412
작업 중에 빌드 태그 변경 . . . . .	413
작업 중에 환경 변수 값 변경 . . . . .	413
레지스터 사용 . . . . .	415
프로젝트 레지스터 . . . . .	416
단계에서 서버 자원과의 파일 복사 . . . . .	417
서버 자원에서 파일 복사 사용 . . . . .	418
서버에서 파일 가져오기 . . . . .	418
서버에 파일 넣기 . . . . .	418
직접 파일 전송 구성 . . . . .	419
문제점 해결 단계 처리 . . . . .	419
작업이 ANT 빌드 명령 다음의 단계 명령을 처리하지 않음 . . . . .	419
Windows 일괄처리 명령 다음의 단계 내 명령이 실행되지 않음 . . . . .	420
점 명령 참조 . . . . .	420
점 명령 구문 . . . . .	420
.bom . . . . .	421
.bomexport . . . . .	422
.break . . . . .	423
.bset . . . . .	423
.buildstatus . . . . .	424
.date . . . . .	425
.defect . . . . .	426
.drill . . . . .	427
.edit . . . . .	429
.email . . . . .	430
.export . . . . .	430
.get . . . . .	430
.load . . . . .	431
.lock . . . . .	434
.mkdir . . . . .	435
.monitor . . . . .	435
.pack . . . . .	435
.pop . . . . .	436
.poptag . . . . .	437
.purge . . . . .	437
.push . . . . .	437
.put . . . . .	438
.rem . . . . .	438
.report . . . . .	438
.retag . . . . .	439
.retry . . . . .	439

.rget . . . . .	439
.rmdir . . . . .	440
.rput . . . . .	440
.run 및 .runwait . . . . .	440
.scan . . . . .	443
.semget . . . . .	444
.semput . . . . .	444
.set . . . . .	444
.sleep . . . . .	445
.snapshot . . . . .	445
.source . . . . .	446
.stop . . . . .	447
.strsub . . . . .	447
.test . . . . .	448
.tset . . . . .	449
.unlock . . . . .	449

제 21 장 작업에 대한 작업 . . . . .	451
작업 정보 . . . . .	451
홈 패널 정보 . . . . .	451
작업 패널 정보 . . . . .	451
작업 실행 및 결과 보기 . . . . .	454
작업 시작 . . . . .	454
작업 결과 보기 . . . . .	455
작업 상태에 대한 RSS 데이터 피드 . . . . .	457
실패한 작업 다시 시작 . . . . .	457
명세서 사용 . . . . .	459
작업 스케줄링 . . . . .	461
스케줄 매개변수 . . . . .	464
작업 클래스 제거 스케줄링 . . . . .	466
작업 관리 . . . . .	466
작업 잠금 . . . . .	466
작업 삭제 . . . . .	467
작업용 작업 디렉토리 . . . . .	468
세마포어 . . . . .	471

제 22 장 보고서에 대한 작업 . . . . .	475
보고서 정보 . . . . .	475
보고서 출력에 데이터를 표시하기 위한 전제조건 . . . . .	476
보고서 데이터 표시를 위한 액세스 권한 . . . . .	476
기본 보고서에 대한 결과를 CSV 파일로 내보내기 . . . . .	476
성능에 대한 표준 보고서 . . . . .	477
프로젝트의 작업 성능 통계 보기 . . . . .	477
프로젝트의 작업 지속 시간 보기 . . . . .	478
프로젝트별로 단계 및 서버 성능 보기 . . . . .	478
조회에 대한 사전 정의 보고서 . . . . .	479



선택기 활용도 히스토리 보기 . . . . .	479
서버별로 현재 서버 Manifest 보기 . . . . .	480
작업 패스/실패/경고 결과 보기. . . . .	481
날짜 범위의 작업에 대한 서버 활용도 보기 . . . . .	481
MD5 값을 사용하여 작업 파일 검색 . . . . .	482
빠른 보고서를 사용하여 보고서 작성 . . . . .	483
빠른 보고서에 필요한 보고서 그룹 권한 . . . . .	483
빠른 보고서에 대한 보고서 유형 참조 . . . . .	484
빠른 보고서에 대한 보고서 형식 및 프리젠테이션 참조 . . . . .	489
샘플 보고서 참조 . . . . .	495
제공된 보고서 유형을 사용하여 보고서 작성 . . . . .	496
작업 BOM에 보고서 출력 추가 . . . . .	497
빠른 보고서에서 보고서 수정 및 관리 . . . . .	497
<b>제 23 장 유틸리티에 대한 작업 . . . . .</b>	<b>501</b>
명령행 유틸리티 액세스 및 실행 . . . . .	501
프로젝트 내보내기. . . . .	501
bfexport 참조 . . . . .	501
.export 사용 . . . . .	504
프로젝트 가져오기. . . . .	504
가져오기 유틸리티를 사용하여 프로젝트 및 기타 오브젝트 가져오기. . . . .	505
bfimport 참조 . . . . .	506
가져온 오브젝트에 액세스 그룹을 지정하는 방법	509
액세스 그룹 지정 및 보안 . . . . .	510
가져오기에서 오브젝트 및 오브젝트 이름 바꾸기	510
<b>제 24 장 UI 구성 탭에서 웹 자원 링크 . . . . .</b>	<b>513</b>
<b>제 25 장 Build Catalyst . . . . .</b>	<b>515</b>
지원되는 운영 체제 . . . . .	515
설치 개요 . . . . .	516
Linux 및 Solaris 운영 체제에 Build Catalyst 설치 . . . . .	516
Windows에 Build Catalyst 설치. . . . .	517
Build Catalyst 예제 . . . . .	518
rafmake 유틸리티 참조. . . . .	521
Build Catalyst 환경 변수 . . . . .	524
<b>제 26 장 Rational Build Forge 통합 방법. . . . .</b>	<b>527</b>
명령행 인터페이스를 사용하여 통합 . . . . .	527
어댑터를 사용하여 통합. . . . .	528
어댑터 요구사항 . . . . .	529
어댑터. . . . .	529
어댑터 태스크 개요 . . . . .	538
핵심 어댑터 태스크 . . . . .	541
ClearQuest 빌드 레코드 업데이트 . . . . .	548

고급 어댑터 태스크 . . . . .	549
어댑터 참조. . . . .	556
어댑터 XML 참조 . . . . .	559
<b>제 27 장 Rational Build Forge 및 Rational ClearCase . . . . .</b>	<b>571</b>
<b>제 28 장 Rational Build Forge 및 Rational ClearQuest. . . . .</b>	<b>579</b>
<b>제 29 장 Rational Build Forge 및 CVS . . . . .</b>	<b>585</b>
<b>제 30 장 IDE 통합 . . . . .</b>	<b>587</b>
IDE 통합 정보. . . . .	587
테스트 프로젝트의 특수 변수 . . . . .	588
Eclipse 및 Rational Application Developer의 플러그인. . . . .	589
Eclipse 또는 Rational Application Developer 플러그인 설치 . . . . .	590
SSL 사용 시 대체 설치. . . . .	591
Eclipse 및 Rational Application Developer의 플러그인 사용 . . . . .	593
Rational Team Concert 플러그인 . . . . .	593
Rational Team Concert 플러그인 사용 . . . . .	594
Rational Team Concert 플러그인 문제점 해결	595
<b>제 31 장 Rational Build Forge 및 Perforce . . . . .</b>	<b>597</b>
<b>제 32 장 Rational Build Forge 및 Rational Quality Manager . . . . .</b>	<b>599</b>
<b>제 33 장 Rational Build Forge 및 Rational Team Concert . . . . .</b>	<b>601</b>
Rational Team Concert 버전 3.x . . . . .	601
Rational Team Concert 버전 1.x 및 2.x. . . . .	601
Rational Team Concert 서버 확장 설치 . . . . .	602
Rational Team Concert의 클라이언트 플러그인 설치 . . . . .	603
Rational Team Concert 어댑터 구성 . . . . .	607
<b>제 34 장 Rational Build Forge 및 StarTeam . . . . .</b>	<b>611</b>
<b>제 35 장 Rational Build Forge 및 Subversion . . . . .</b>	<b>613</b>
<b>제 36 장 Rational Build Forge 및 WebSphere 제품 . . . . .</b>	<b>615</b>
Apache Tomcat 대신 WebSphere Application Server 사용. . . . .	615



Build Forge 애플리케이션을 실행하도록 Websphere Application Server 구성 . . . . .	616
Websphere Application Server에서 Build Forge에 대해 Java 2 보안 구성 . . . . .	619
Websphere Application Server에서 Build Forge에 대해 SSL 및 비밀번호 암호화 지원 구 성 . . . . .	619
Build Forge 애플리케이션을 실행하기 위해 Websphere Application Server를 사용하기 위 한 시작 순서 . . . . .	620
Apache HTTP Server 대신 IBM HTTP Server 사용 . . . . .	621
사전 컴파일된 PHP 파일 설정 . . . . .	622
IBM HTTP Server 구성 파일 편집. . . . .	623
PHP에서 프록시 서버 지정. . . . .	624
IHS에 필요한 SSL 구성 . . . . .	624
Windows: DLL 대체 . . . . .	626
AIX 시스템: 추가 설정 . . . . .	627
<b>제 37 장 API에 대한 작업 . . . . .</b>	<b>629</b>
API 프로그램용 Build Forge 사용자 작성 . . . . .	629
Java 클라이언트 API . . . . .	630
Java 클라이언트 API 패키지 가져오기 . . . . .	630
Java 클라이언트 API 설정. . . . .	630
Perl 클라이언트 API. . . . .	631
Perl 클라이언트 API 패키지 가져오기 . . . . .	631
Perl 클라이언트 API 설정 . . . . .	632
<b>제 38 장 관리 콘솔 버전 번호 판별. . . . .</b>	<b>635</b>
<b>제 39 장 성능 및 확장성 . . . . .</b>	<b>637</b>
<b>제 40 장 제품과 함께 설치된 실행 가능 명령 . . . . .</b>	<b>639</b>
<b>제 41 장 용어집 . . . . .</b>	<b>641</b>
액세스 그룹(access group). . . . .	641
어댑터(adaptor) . . . . .	641

에이전트(agent) . . . . .	641
아카이브(archive) . . . . .	641
BOM . . . . .	641
클래스(class) . . . . .	642
정리(clobber) . . . . .	642
컬렉터(collector) . . . . .	642
데이터베이스(database) . . . . .	642
동적(dynamic) . . . . .	642
엔진(engine) . . . . .	642
환경(environments) . . . . .	642
데이터 교환(handshake). . . . .	643
인터셉터(interceptor). . . . .	643
인터페이스(interface). . . . .	643
작업(job). . . . .	643
라이브러리(library) . . . . .	643
LDAP(Lightweight Directory Access Protocol)	643
Manifest . . . . .	644
관리 콘솔(Management Console) . . . . .	644
알림 템플릿(notification templates) . . . . .	644
플러그인(plugin) . . . . .	644
프로젝트(project) . . . . .	644
선택기(selector) . . . . .	644
세마포어(semaphore). . . . .	645
서버(server). . . . .	645
서비스(services) . . . . .	645
스냅샷(snapshot) . . . . .	645
정적(static) . . . . .	645
단계(step) . . . . .	646
단계 로그(step log) . . . . .	646
스레딩(threading) . . . . .	646
사용자(user). . . . .	646
<b>부록. 주의사항 . . . . .</b>	<b>647</b>
<b>상표 . . . . .</b>	<b>649</b>



---

## 제 1 장 IBM Rational Build Forge 정보

이 절에서는 제품 에디션 및 기본 개념에 대해 설명합니다.

---

### Build Forge 제품 개정판

다음 제품 개정판을 사용할 수 있습니다.

- IBM Rational Build Forge Standard Edition
- IBM Rational Build Forge Enterprise Edition
- IBM Rational Build Forge Enterprise Plus Edition

다음 표는 각 개정판의 구분된 컴포넌트 또는 기능을 나열합니다.

컴포넌트 및 기능	Standard Edition	Enterprise Edition	Enterprise Plus Edition
관리 콘솔	Microsoft Windows, UNIX 또는 Linux	Windows, UNIX 또는 Linux	Windows, UNIX 또는 Linux
데이터베이스	지원되는 모든 데이터베이스	지원되는 모든 데이터베이스	지원되는 모든 데이터베이스
에이전트	지원되는 모든 운영 체제	지원되는 모든 운영 체제	지원되는 모든 운영 체제
라이선스 서버	필수(25명의 동시 사용자)	필수(150명의 동시 사용자)	필수 아님(250명의 동시 로그인)
어댑터 툴킷	지원됨	지원됨	지원됨
빠른 보고서	지원됨	지원됨	지원됨
API(Perl, Java)	지원됨	지원됨	지원됨
동적 서버 관리	지원되지 않음	지원됨	지원됨

---

### 프로세스 자동화

프로세스 자동화를 사용하면 비즈니스에서 서로 다른 애플리케이션, 사용자 및 시스템 사이에 프로세스를 연출하여 비효율성 제거, 비용 최적화, 규제 준수 및 생산성 신장을 이룰 수 있습니다.

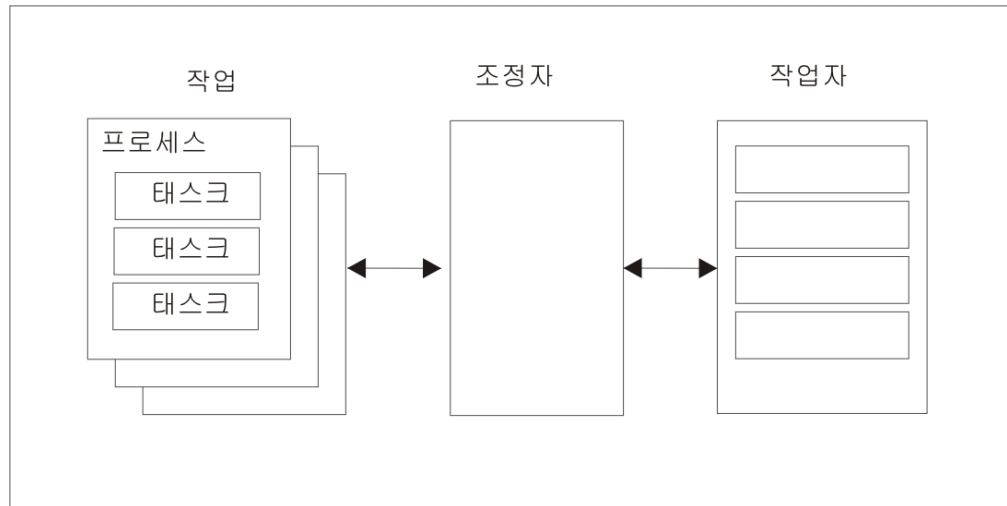
Build Forge는 제품 개발 어셈블리 라인에서 프로세스를 자동화, 조정, 관리 및 추적합니다. 소프트웨어 빌드 및 패키지 프로세스 자동화를 위해 공통적으로 사용됩니다.

이 절에서는 프로세스에 대한 사용자 모델을 설명하고 Build Forge 기능에 매핑합니다.

#### 사용자 모델

다음 역할이 프로세스 자동화의 사용자 기반 모델을 구성합니다.

- **작업:** 수행될 사용 가능한 작업은 문서화된 프로세스 세트입니다. 각 프로세스는 태스크로 구성됩니다.
- **조정자:** 조정자는 작업을 "실행"합니다. 조정자는 실행할 프로세스를 선택한 후 태스크를 순서대로 조사합니다. 태스크마다, 프로세스는 수행해야 할 작업자를 정의합니다. 조정자는 태스크를 작업자에게 넘기고 결과를 기다립니다.
- **작업자:** 작업자는 프로세스에 정의된 태스크를 수행합니다.



## Build Forge 모델

사용자 기반 모델은 다음과 같이 Build Forge 모델에 맵핑됩니다.

- **작업:** 작업 프로세스는 Build Forge 프로젝트 오브젝트에서 정의됩니다. 각 프로젝트에는 하나 이상의 단계 오브젝트가 있습니다. 프로젝트의 단계는 사용자 모델의 프로세스에 있는 태스크에 해당됩니다.
- **조정자:** 조정자는 실행 중인 Build Forge 소프트웨어입니다. 프로세스 엔진이라고 하는 컴포넌트가 포함됩니다. Build Forge에서 프로젝트를 시작할 때 프로세스 엔진은 이 프로젝트를 작업 오브젝트로 실행합니다. 작업은 단지 실행 중인 프로젝트입니다.
- **작업자:** 작업자는 호스트 컴퓨터입니다. Build Forge에서 서버 오브젝트로 표시됩니다. 호스트에는 에이전트가 설치되어 있어야 합니다.

이는 매우 단순화된 모델입니다.

- Build Forge에는 프로세스 자동화를 지원하기 위한 다른 많은 오브젝트 유형이 포함됩니다. 서버 및 프로젝트를 지원하는 사용자, 권한 및 기타 오브젝트는 모두 데이터베이스에 저장됩니다. Build Forge 파이프 역시 데이터베이스에 저장됩니다(예: UI 위젯). Build Forge가 실행되려면 데이터베이스가 실행 중 상태여야 합니다.
- 프로젝트 및 단계는 복잡한 종속 항목 및 워크플로우가 있는 복잡한 프로세스를 지원하기 위해 많은 다양한 방법으로 실행되도록 구성할 수 있습니다. 프로젝트가 다른 프로젝트를 실행할 수 있고, 단계 자체가 프로젝트를 실행할 수 있습니다.

- 단계 및 프로젝트는 서버 정의의 일부인 기준 데이터를 기반으로 동적으로 서버를 선택할 수 있습니다. 예를 들어, 단계는 Windows 또는 UNIX/Linux 중 어느 것을 실행 중인지 여부에 따라 실행할 서버를 선택할 수 있습니다. 서버는 또한 풀링된 자원으로 구성될 수 있으므로, 단계는 현재 가용성 또는 로드를 기반으로 서버를 선택합니다.

## 설치할 Build Forge 컴포넌트

Build Forge를 사용하려면 다음 컴포넌트를 설치해야 합니다.

- 데이터베이스: Build Forge는 사용하는 모든 오브젝트를 데이터베이스를 사용하여 저장합니다.
- Build Forge: 데이터베이스에 대한 액세스를 관리하고 UI 코드를 실행하며 실행 중인 작업의 작업을 수행하는 기술 세트입니다.
- 에이전트: 작업을 수행할 호스트마다 에이전트 소프트웨어를 설치합니다.
- 웹 브라우저: 사용자는 일반적으로 웹 브라우저에 표시된 콘솔을 통해 Build Forge에 액세스합니다.

---

## Build Forge 개념 및 오브젝트

Build Forge는 간단한 오브젝트 세트를 사용하여 프로세스 자동화 설명을 생성합니다.

이 절에서는 주요 개념 및 오브젝트의 개요와 이 개념 및 오브젝트의 관련 방식을 제공합니다. 편리를 위해 자세한 설명에 대한 링크가 포함되어 있습니다.

### 사용자 및 역할

Build Forge에 액세스하려면 사용자에게 사용자에게 맞게 설정된 사용자 오브젝트가 있어야 합니다. 프로덕션 시스템에서 ID 관리 시스템인 LDAP에 액세스하도록 Build Forge를 구성하면 됩니다.

Build Forge는 권한 시스템을 사용하여 모든 오브젝트에 대한 사용자 액세스를 제어합니다. 액세스 그룹 오브젝트에는 권한 목록이 포함됩니다. 사용자는 하나 이상의 액세스 그룹에 속합니다.

사용자 고유의 액세스 그룹을 작성하거나 제공된 액세스 그룹을 수정할 수 있습니다. 시스템에 정의된 액세스 그룹은 다음과 같습니다.

- 빌드 엔지니어 - 프로세스를 정의합니다(프로젝트 및 단계를 작성합니다).
- 시스템 관리자 - 서버 및 기타 시스템 전반의 기능을 관리합니다.
- 보안 - 사용자, 액세스 및 보안을 관리합니다.
- 개발자 - 프로세스 개발을 돕고, 작업을 실행하며, 결과를 봅니다.
- 운영자 - 프로젝트를 복사하고, 작업을 실행하며, 결과를 봅니다.

- 게스트 - 프로젝트를 복사하고, 작업을 실행하며, 결과를 봅니다.

## 서버

서버 오브젝트는 프로젝트 및 단계가 실행할 수 있는 위치를 정의합니다. 프로젝트 및 단계는 동일한 서버를 사용하거나 하나를 독립적으로 선택할 수 있습니다.

서버 오브젝트는 작업이 수행되는 호스트를 표시합니다. 호스트는 시스템에서 작업을 수신하고 결과를 리턴하기 위해 Build Forge 에이전트를 실행 중이어야 합니다.

다른 오브젝트는 서버에 관련되며 프로젝트 및 단계 정의 이전에 설정해야 합니다.

- 서버 인증: 서버 인증은 서버가 서버에서 지정된 호스트에 액세스하기 위해 사용할 로그인 정보를 저장합니다. 서버 인증은 이 인증을 사용하는 서버를 작성하기 전에 작성해야 합니다. 서버 정의에서, 시스템에 정의된 모든 서버 인증의 목록에서 서버 인증을 선택합니다.
- 콜렉터" 콜렉터 오브젝트는 지정된 서버 특성을 수집합니다. 데이터는 *Manifest*에 저장됩니다. 서버에는 지정된 기본 특성 세트가 있습니다. 이 빌트인 특성에는 호스트 아키텍처, 네트워크 연결 및 자원(CPU, 메모리, 로드)에 대한 정보가 포함됩니다. 콜렉터 오브젝트를 정의하여 다른 특성을 추가할 수 있습니다. 콜렉터는 서버에 추가하기 전에 작성해야 합니다. 서버 정의에서, 시스템에 정의된 모든 콜렉터의 목록에서 콜렉터를 선택합니다.
- 선택기: 선택기 오브젝트는 프로젝트 또는 단계에서 사용하기 위해 서버를 선택하는 방법을 정의합니다. 선택기가 사용되는 프로젝트에 이 선택기를 추가하기 전에 선택기를 작성해야 합니다. 프로젝트 정의에서, 시스템에 정의된 모든 선택기의 목록에서 선택기를 선택합니다.
- 환경: 환경 오브젝트는 단계에서 사용할 수 있는 변수 세트입니다. 작업 중에, 단계가 실행되기 전에 서버 호스트에서 변수가 설정됩니다. 환경은 서버 오브젝트, 프로젝트 오브젝트 및 단계 오브젝트와 연관될 수 있습니다. 동일한 변수가 다른 환경에서 다른 값으로 설정되는 경우, 상속 방식(scheme)은 사용되는 값을 판별합니다. 서버, 프로젝트 또는 단계에 환경을 추가하기 전에 이 환경을 작성해야 합니다. 환경 오브젝트 정의에서, 시스템에 정의된 모든 환경의 목록에서 환경을 선택합니다.

## 환경

환경은 변수 세트입니다. 환경은 서버, 프로젝트 및 단계 오브젝트에 지정할 수 있습니다. 단계가 실행될 때 해당 오브젝트 각각에서 설정된 환경이 결합되어 사용할 단계에 대한 변수가 제공됩니다. 321 페이지의 『환경 정보』의 내용을 참조하십시오.

단계가 실행되는 대로 변수를 변경할 수 있습니다. (331 페이지의 『단계 실행 중 변수 값 변경』을 참조하십시오.) 변경 범위는 단계에 국한되거나, 프로젝트에 국한되거나, 영구적(변수가 저장된 환경에서 변경됨)이 될 수 있습니다.

사전정의된 시스템 변수와, 사용자가 정의하는 변수를 사용할 수 있습니다.

## 프로젝트

프로젝트는 프로세스에서 수행될 작업을 정의합니다. 프로젝트가 시작될 때 작업(job)으로 실행됩니다.

수행될 작업(work)은 단계 목록에 포함됩니다.

기타 오브젝트는 프로젝트에 관련됩니다.

- **선택기:** 선택기는 시작될 프로젝트를 결정합니다. 프로젝트에 대해 선택기가 지정되지 않은 경우 프로젝트는 독립적으로 실행될 수 없으며 라이브러리를 호출합니다. 선택기는 작업에 지정되도록 이미 정의되어 있어야 합니다.
- **환경:** 환경 오브젝트는 단계에서 사용할 수 있는 변수 세트입니다. 작업 중에, 단계가 실행되기 전에 서버 호스트에서 변수가 설정됩니다. 환경은 서버 오브젝트, 프로젝트 오브젝트 및 단계 오브젝트와 연관될 수 있습니다. 상속 방식은 동일한 변수가 다른 값으로 설정된 경우 사용되는 값을 결정합니다. 환경을 사용하는 서버, 프로젝트 또는 단계를 작성하기 전에 이 환경을 작성해야 합니다. 환경 오브젝트 정의에서, 시스템에 정의된 모든 환경의 목록에서 환경을 선택합니다.
- **알림 템플릿:** 알림 템플릿은 작업 활동(작업 시작, 작업 패스, 작업 실패, 기타)에 대한 알림을 전송할 방법을 정의합니다. 알림 오브젝트는 액세스 그룹을 통해 알릴 사용자를 정의합니다. 시스템 설정에서 SMTP 서버의 위치를 지정합니다.
- **클래스:** 클래스 오브젝트는 유지보수 목적으로 프로젝트를 그룹화하기 위해 사용됩니다. 일반적으로 클래스는 완료된 작업을 정기적으로 제거하거나 아카이브하기 위해 사용됩니다. 프로젝트를 사용하여 실행된 모든 작업은 클래스에 의해 영향을 받습니다.
- **어댑터:** 어댑터는 외부 시스템(일반적으로 소스 코드 관리 시스템)과의 통합을 정의합니다. 여러 샘플 템플릿이 시작점으로 제공됩니다. 연결을 관리하고 시스템에서 지정된 조치를 수행하기 위해 어댑터 템플릿을 추가 구성해야 합니다.
- **어댑터 링크:** 어댑터 링크는 어댑터와, 이 어댑터를 사용하는 프로젝트 사이의 관계를 정의합니다.

## 단계

단계는 수행할 가장 작은 작업 단위입니다. (393 페이지의 『단계 정보』을 참조하십시오.) 키 컴포넌트는 선택된 서버에서 실행할 명령을 포함하는 명령 특성입니다.

명령 특성은 점(dot) 명령을 실행하기 위해 사용할 수도 있습니다. 점 명령은 프로세스 엔진에서 실행되고 추가 기능을 제공하는 명령입니다.

기타 오브젝트는 프로젝트에 관련됩니다.

- **선택기:** 단계에는 고유한 선택기가 있을 수 있습니다. 지정하지 않으면 해당 프로젝트의 선택기를 사용합니다.



- 환경: 단계에는 고유한 환경이 있을 수 있습니다. 서버, 프로젝트 및 단계에서 제공되는 환경이 결합됩니다. 기본적으로 이 순서로 적용되므로, 단계 환경에서 정의된 변수가 동일한 변수 정의에서 우선합니다. 우선순위는 시스템 설정에서 제어할 수 있습니다.
- 로그 필터: 단계에는 결과 특성에서 로그 필터가 지정될 수 있습니다. 로그 필터 오브젝트는 단계가 패스 또는 실패인지 여부를 표시하는 조건을 지정하기 위해 사용됩니다. Perl 표현식을 사용하여 특정 패턴에 대해 로그를 스캔합니다. 보통 명령의 종료 상태가 사용되지만, 로그 필터는 대체 수단을 제공합니다. 로그 필터는 단계에서 지정하지 하기 전에 작성해야 합니다. 단계에 대한 결과 특성에서, 시스템에 정의된 모든 로그 필터 목록에서 로그 필터를 선택합니다.
- 알림 템플릿: 알림 템플릿은 단계 활동(단계 시작, 단계 패스, 단계 실패, 기타)에 대한 알림을 전송할 방법을 정의합니다. 알림 오브젝트는 액세스 그룹을 통해 알릴 사용자를 정의합니다. 시스템 설정에서 SMTP 서버의 위치를 지정합니다.
- Build Catalyst: 단계는 rafmake(Build Catalyst의 주요 유틸리티)를 실행할 수 있습니다. Build Catalyst는 make-based 빌드를 가속화하기 위한 수단을 제공합니다. Build Catalyst는 Build Forge 에이전트 외에, make 빌드가 실행되는 동일한 호스트에 설치해야 합니다.

## 작업(Job)

작업은 실행 중인 프로젝트입니다. 작업이 시작되면, 프로세스 엔진은 작업을 큐에 넣고 실행합니다. 사용자는 작업 패널에서 상태를 확인할 수 있습니다. 완료되면 다음 정보를 사용할 수 있습니다.

- 결과: 작업을 열어서 모든 단계의 결과를 검토할 수 있습니다. 또한 실행 중인 작업을 열어서 해당 진행상태를 모니터링할 수도 있습니다.
- 단계 로그: 단계 로그는 Manifest 및 환경 설정 정보와 실행 결과를 포함하여, 단계가 실행된 방법에 대한 포괄적인 정보를 기록합니다. 로그를 보려면 결과 페이지에서 단계 결과 링크를 클릭하십시오.
- 명세서: 명세서(BOM)에는 작업 단계 및 단계 Manifest에 대한 정보가 있습니다. .bom 점(dot) 명령을 사용하여 BOM에 대한 추가 정보를 형식화하고 데이터를 BOM에 기록할 수 있습니다. .scan 점 명령을 사용하여 기준선 및 체크포인트를 BOM에 추가할 수 있습니다.

작업을 취소하고 다시 시작할 수 있습니다. 스케줄에 프로젝트를 추가하여 프로젝트가 스케줄된 시간에 실행되도록 할 수 있습니다.

---

## 제 2 장 Build Forge<sup>®</sup> 시작하기

이 주제에서는 시스템에 대해 간략하게 소개합니다.

### 설치

시스템이 아직 설치되어 있지 않은 경우, 시스템을 설치하십시오. 시작을 단순화하도록 단일 호스트에 시스템(관리 콘솔 및 에이전트)을 설치할 수 있습니다.

1. 요구사항 확인
  - 21 페이지의 『데이터베이스 요구사항』
2. 다음 단계를 완료하십시오.
  - a. 31 페이지의 제 5 장 『설치 계획』(독립형, 일반 프로덕션, 분산 등)
  - b. 35 페이지의 제 6 장 『설치 전 설정』(국제 데이터, 데이터베이스 및 보안)
  - c. 63 페이지의 제 7 장 『관리 콘솔 설치』
  - d. 161 페이지의 제 10 장 『에이전트 설치』

---

### 콘솔 액세스 및 사용

이 주제에서는 시스템 메뉴 및 탭을 사용하는 데 대한 기본 정보를 제공합니다.

#### 콘솔 액세스

콘솔에 액세스하려면 다음을 수행하십시오.

1. 브라우저 창을 여십시오.
2. 콘솔 URL을 입력하십시오.

`http://host:port/`

*Host*는 콘솔을 실행 중인 호스트의 완전한 네트워크 이름 또는 IP 주소입니다.

*Port*는 HTTP 기본 포트가 사용되지 않는 경우에만 필요합니다. SSL이 사용되지 않으면 기본 포트가 80입니다. 기본 HTTPS 포트는 443입니다. HTTPS는 콘솔이 SSL을 사용하도록 구성되어 있는 경우에만 사용됩니다.

예제:

- `http://localhost/` 또는 `http://127.0.0.1:` 콘솔이 설치되어 있는 호스트에서 브라우저를 실행 중인 경우에만 사용 가능합니다.
- `http://my.company.com/`

- `http://my.company.com:81`: 포트 81을 사용하는 콘솔이 설치되어 있으므로 포트 번호를 지정해야 합니다.

### 3. 로그인하십시오. 다음 정보를 제공한 후 로그인을 클릭하십시오.

- 사용자 이름
- 비밀번호
- 도메인

기본적으로 다른 URL로 재지정되어 신임 정보를 입력하게 되며, 로그인한 후에 원래대로 재지정됩니다. URL은 서비스 컴포넌트용입니다. 이는 인증용 보안 서블릿을 제공합니다. 이 사용은 구성 가능하며, 꺼져 있거나 사용자 인증에 LDAP이 사용되는 경우에는 사용되지 않습니다. 자세한 정보는 124 페이지의 『기본 로그인 보안 정보』의 내용을 참조하십시오.

루트 사용자 또는 Build Forge에만 정의된 다른 사용자로 로그인하면 도메인에 대한 프롬프트가 표시되지 않습니다. 가장 좋은 방법은 사용자를 인증하는 데 LDAP를 사용하도록 Build Forge를 구성하는 것입니다. 설정 시 도메인 필드가 표시됩니다. 사용자 인증에 사용되는 LDAP 도메인을 나타냅니다.

251 페이지의 『LDAP 통합』의 내용을 참조하십시오.

## 사용자 세션

사용자 세션은 처음에 쿠키에 생성된 세션 ID로 추적됩니다. 일단 생성된 세션 ID는 데이터베이스에 저장됩니다. 모든 사용자 상호작용은 세션 ID에 대해 검사됩니다.

일반적으로 동일한 로그인을 사용하여 여러 브라우저 창을 열 수 있지만 이러한 창은 동일한 호스트에 있어야 하며 동일한 유형의 브라우저(FireFox, Internet Explorer)를 사용해야 합니다.

다음 조건이 충족되는 경우 세션이 닫힙니다.

- 동일한 사용자 계정을 사용하여 다른 호스트에 로그인합니다.
- 동일한 사용자 계정을 사용하여 동일한 호스트에 로그인하지만 다른 유형의 브라우저를 사용합니다.
- 사용자가 관리 콘솔 세션이 있는 브라우저 창 또는 탭을 닫습니다.
- 세션 시간 한계에 도달합니다. 시간 한계는 자동 로그오프 시스템 설정에 지정됩니다. 기본적으로 0으로 설정되며, 이는 시간 한계가 없음을 의미합니다.

## 제한조건과 팁

다음 사용에 대한 제한조건과 팁에 유의하십시오.

- 디스플레이를 1024 x 768보다 낮지 않게 설정하십시오. 최선의 결과를 얻으려면 1280 x 1024 이상을 사용하십시오.

- 브라우저 창 크기를 1024 x 768보다 작게 줄이지 마십시오.
- 브라우저 설정을 사용하여 글꼴 크기, 색상 및 기타 내게 필요한 옵션 기능을 제어하십시오. 새 브라우저 설정을 적절히 표시하려면 페이지를 새로 고쳐야 할 수도 있습니다.
- 글꼴이 너무 크거나 브라우저를 축소하면 Microsoft Internet Explorer의 작업 > 시작 페이지에서 관리 콘솔(MC)이 표시되지 않습니다.

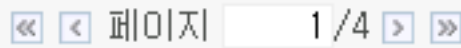
## 콘솔 창 사용



애플리케이션을 보려면 오른쪽 상단에 있는 탭을 클릭하십시오. 대시보드(기본값), UI 구성, 콘솔 또는 보고서를 클릭할 수 있습니다. 시스템은 사용자가 로그아웃할 때 위치했던 탭을 기억하여 다음에 로그인할 때 표시합니다.


다음과 같이 콘솔 애플리케이션에서 탐색하십시오.

- 메뉴가 왼쪽에 표시되고 기본 보기 패널이 오른쪽에 표시됩니다.
- 패널을 표시하거나 패널의 하위 메뉴를 열려면 메뉴 항목을 클릭하십시오.
- 보기 영역보다 더 긴 패널의 경우, 오른쪽 상단에 있는 페이지징 제어(



)를 사용하십시오.

- 목록을 볼 때 필터 필드를 사용하십시오. 문자열을 입력하고 필터를 클릭하면 목록이 업데이트됩니다. 이름에 문자열이 있는 항목만 표시합니다.
- 크기를 조정하려면 메뉴의 오른쪽 가장자리를 끄십시오.
- 데이터를 작성하거나 편집할 수 있는 패널은 일반적으로 다음과 같은 제어를 가집니다.

- 패널의 상단 부분을 사용하여 항목을 보고 선택할 수 있습니다. 항목 내용을 확인하려면 항목 이름을 클릭하십시오. 항목을 편집하려면 편집 아이콘()을 클릭하십시오.
- 패널의 하단 부분에 선택한 항목의 내용이 표시됩니다.
- 예를 들어, 목록에서 항목을 선택하면 왼쪽의 메뉴 아래에 추가 정보가 표시됩니다. 예를 들어 다음과 같습니다.
  1. 관리 > 사용자를 엽니다.
  2. 사용자를 클릭합니다.

왼쪽의 메뉴 아래에 해당 사용자의 권한이 나타납니다.

---

## 목록 필터링 및 정렬

목록을 빠르게 필터링 또는 정렬할 수 있습니다.

목록을 필터링하려면 다음을 수행하십시오.

1. 필터 텍스트 상자에 텍스트를 입력하십시오. 입력은 대소문자를 구분합니다.
2. 필터를 클릭하십시오.
3. 결과를 확인하십시오. 모든 결과를 볼 수도 있고 페이지를 매길 수도 있습니다.

팁:

- 필터는 항목이 링크(파란색 또는 밑줄)인 모든 열에서 작동합니다. 값이 일반 검은색 텍스트로 표시되는 열에 대해서는 필터링할 수 없습니다.
- 단일 열에 대해서만 필터링하려면 열 이름을 입력한 다음 필터링 문자열을 입력하십시오. 예를 들어, 선택기 열이 있는 경우 선택기: 모두 선택을 입력할 수 있습니다.
- 시스템은 사용자가 입력하는 필터 문자열을 보유하고 있습니다. 필터 상자 옆에 있는 화살표를 클릭하여 사용자나 다른 사용자가 입력한 필터 목록을 표시하십시오. 필터 문자열을 삭제하려면 문자열을 강조표시하고 문자열 오른쪽에 있는 휴지통 아이콘을 클릭하십시오.
- 모두 표시 선택사항은 필터 문자열의 목록에 항상 표시됩니다.
- 필터 텍스트를 입력했으나 필터에 일치하는 항목이 없는 경우에는 항목이 표시되지 않습니다.

목록을 정렬하려면 다음을 수행하십시오.

1. 열 레이블 옆에 있는 이중 화살표를 클릭하십시오.
2. 결과를 확인하십시오. 모든 결과를 볼 수도 있고 페이지를 매길 수도 있습니다.

팁:

- 이중 화살표를 다시 한 번 클릭하면 정렬 순서가 반대가 됩니다.

## Hello world 프로젝트 작성

이 주제에서는 단순 프로젝트를 작성 및 실행하여 빌드 시스템이 올바르게 설치되었는지 확인하는 방법에 대해 설명합니다.

다음 항목에서는 "Hello World" 프로젝트를 작성하는 데 필요한 태스크에 대한 개요를 제공합니다.

- 다음과 같이 서버를 설정하십시오.
  - Rational Build Forge가 서버에서 명령에 액세스하고 이를 실행할 수 있도록 서버 인증(로그인 이름 및 비밀번호) 작성
  - 에이전트가 설치된 서버를 표시하도록 서버 정의 작성
  - HelloWorld 프로젝트에서 실행할 서버를 판별하는 데 사용할 선택기 작성
  - 마스터 콘솔 및 서버의 에이전트가 서로 통신할 수 있도록 서버 연결 테스트
- 다음과 같이 프로젝트를 정의하십시오.
  - 프로젝트 이름을 HelloWorld로 지정하고 선택기를 사용하여 프로젝트를 실행할 서버 지정
  - Hello World 메시지를 표시하는 명령을 실행할 단계 작성
- 다음과 같이 작업을 실행하십시오.
  - 작업이라고도 하는 프로젝트의 인스턴스 시작
  - 왼쪽 패널에서 작업 항목을 클릭하여 작업 상태 확인
- 다음과 같이 작업 결과를 보십시오.
  - 완료된 작업 목록에서 작업 태그를 클릭하여 작업 단계의 목록 또는 HelloWorld 예제의 경우 단일 단계 표시
  - 작업 단계를 클릭하여 단계 로그 표시
  - 단계 로그에서 명령의 Hello World 출력 찾기

## 서버 설정

이 주제에서는 빌드 시스템이 올바르게 설정되었는지 확인하기 위해 "Hello World" 연습에서 사용할 서버를 설정하는 방법을 설명합니다.

### 서버 인증 작성

Rational Build Forge는 에이전트가 설치된 서버에 연결하도록 서버 인증(로그인 이름 및 비밀번호)을 사용합니다.

## 이 태스크 정보

Rational Build Forge에서 사용할 자원으로 해당 서버를 정의하기 전에 서버에 대한 서버 인증을 작성해야 합니다. 로그인 이름 및 비밀번호와 연관된 권한은 서버에서 실행할 때 Rational Build Forge 프로젝트가 사용하는 권한을 결정합니다.

### 프로시저

1. Rational Build Forge의 왼쪽 패널에서 서버 > 서버 인증을 클릭하십시오.

관리 콘솔에서는 맨 아래 빈 서버 인증 세부사항 패널을 표시합니다.

2. 서버 인증 추가를 클릭하십시오.
3. 이름에서 인증에 대한 이름을 입력하십시오. 인증 이름에 대한 로그인 이름을 사용하십시오.
4. 로그인에서 서버의 계정에 대한 로그인 이름을 지정하십시오.

참고: 로그인 이름의 출처가 도메인 사용자인 경우, 이 필드에 도메인을 포함시키십시오. 예를 들어, MYDOMAIN/joeuser를 입력하십시오.

5. 비밀번호에 비밀번호를 입력하십시오.
6. 확인에 비밀번호를 다시 입력하십시오.
7. 서버 인증 저장을 클릭하십시오.

### 결과

새 서버 인증은 맨 위 패널에 표시되면 서버를 정의할 때 사용할 수 있습니다.

## 서버 작성

관리 콘솔에서 서버는 프로젝트 또는 단계를 실행할 수 있는 호스트를 나타냅니다.

### 프로시저

1. Rational Build Forge의 왼쪽 패널에서 서버를 클릭하십시오.
2. 서버 추가를 클릭하십시오.
3. 서버 세부사항을 제공하십시오.

(새 서버) 저장 복사 삭제 작업 계수 다시 설정 큐 Manifest 새로 고치기 연결 테스트

세부사항 Manifest 테스트 결과 노트 (0)

☐ 사용 안함 이름: 경로:

호스트: 폴백터: 환경:

인증: -- 없음 -- 비밀번호 암호화: -- 없음 --

맥세스: 사용 불가능 파일: -- 없음 --

개발자 SSL 사용: -- 없음 --

최대 작업: 3 아니오



- a. 이름: HelloWorldServer 이름을 서버에 제공합니다. 나중에 선택기를 정의할 때 이 동일한 이름을 사용합니다.
- b. 경로: 프로젝트 및 작업 디렉토리를 작성할 때 서버에서 사용하는 디렉토리(예: C:\buildforgeprojects)를 지정합니다. 시스템은 빌드 디렉토리를 사용할 때 이 경로 값을 시작점으로 사용합니다.

팁: 시스템은 서버 경로를 작성하지 않습니다. 빌드가 서버에 액세스를 시도하기 전에 경로가 존재해야 합니다. 경로가 존재하지 않으면 빌드가 실패합니다.

- c. 호스트: 에이전트를 실행 중인 실제 컴퓨터의 호스트 이름을 제공하십시오. 관리 콘솔 컴퓨터를 서버로 정의하고 있는 경우 localhost 값을 사용하십시오(에이전트도 관리 콘솔에 설치되어야 함).

참고: 호스트 이름 앞에 프로토콜이 오지 않게 하십시오. 예를 들어, http://를 사용하지 마십시오.

- d. 인증: 이 서버에 사용할 서버 인증을 선택하십시오.
  - e. 다른 필드는 기본값 상태로 두십시오.
4. 저장을 클릭하십시오. 새 서버가 콘텐츠 패널 맨 위의 서버 목록에 표시됩니다.
  5. 서버를 올바르게 구성했는지 확인하려면 목록에서 서버를 선택한 후 **연결 테스트**를 클릭하십시오.

서버와 통신할 수 없는 경우 시스템이 오류를 보고합니다. 오류를 수신한 경우 서버에서 에이전트가 실행 중인지 확인하십시오.

## 선택기 작성

선택기는 프로젝트 또는 단계를 실행할 서버를 결정합니다.

## 이 태스크 정보

HelloWorld 프로젝트가 해당 단계를 완료한 위치를 결정할 수 있도록 선택기를 작성합니다.

### 프로시저

1. Rational Build Forge의 왼쪽 패널에서 **서버 > 선택기**를 클릭하십시오.
2. 선택기 추가를 클릭하십시오.
3. 이름에서 HelloWorldSelector를 입력하십시오.
4. 저장을 클릭하십시오. 선택할 서버를 판별하는 선택기 특성을 정의할 수 있도록 패널 맨 아래 부분이 변경됩니다.

5. 12 페이지의 『서버 작성』에 지정된 서버 이름을 사용하여 실행할 서버를 설정하십시오. 이름이 BF\_NAME으로 설정되고 연산자가 EQ(기본값)로 설정된 경우 값에서 HelloWorldServer를 입력하십시오.
6. 저장을 클릭하십시오.

## 프로젝트 정의

이 주제에서는 "Hello World" 연습의 일부로 단순 프로젝트를 작성하여 빌드 시스템이 올바르게 설치되었는지 확인하는 방법에 대해 설명합니다.

### 프로젝트 작성

#### 시작하기 전에

HelloWorld 프로젝트 정의에서 사용할 선택기 및 서버를 설정해야 합니다.

#### 프로시저

1. 프로젝트를 선택하십시오. 프로젝트 세부사항 패널이 기본 콘텐츠 패널 맨 아래 표시됩니다.
2. 이름 필드에 HelloWorld를 입력하십시오.
3. 선택기를 HelloWorldSelector로 설정하십시오.
4. 저장을 클릭하십시오. 시스템이 프로젝트의 빈 단계 목록과 빈 단계 세부사항 페이지를 표시합니다. 『단계 작성』에서 설명한 대로 단계를 정의하십시오.

### 단계 작성

#### 시작하기 전에

HelloWorld 프로젝트에서 첫 번째 단계를 정의하려면 프로젝트 정의를 시작해야 합니다.

#### 프로시저

1. 단계 세부사항 페이지의 이름 필드에 EchoHelloWorld를 입력하십시오.
2. 명령 필드에는 선택한 서버의 표준 출력에 Hello world를 쓸 명령을 입력하십시오. 예를 들어, 이 명령은 Windows®, Solaris, Linux®, UNIX® 및 Apple Macintosh OS X 운영 체제에서 작동합니다.  
echo Hello World
3. 단계 저장을 클릭하십시오. 단계 목록에 단계가 표시됩니다.

## 작업 실행


이 주제에서는 "Hello World" 연습의 일부로 작업(실행 중인 프로젝트의 인스턴스)을 시작하고 해당 상태를 확인하여 빌드 시스템이 올바르게 설치되었는지 확인하는 방법에 대해 설명합니다.

## 작업 시작

### 시작하기 전에

HelloWorld 프로젝트를 정의해야 합니다.

### 프로시저

1. 프로젝트를 선택하여 프로젝트 목록을 다시 표시하십시오.
2. HelloWorld 프로젝트 옆에 있는 빠른 시작 아이콘  을 클릭하십시오.

## 작업 상태 확인

### 프로시저

1. 작업 및 실행 중 탭을 차례로 클릭하여 실행 중으로 나열된 HelloWorld 작업을 표시하십시오. 작업이 완료되면 완료됨 탭의 페이지로 이동됩니다.

참고: HelloWorld 작업이 나열되지 않으면 다음 단계를 계속 진행하십시오.

2. 완료 탭을 클릭하십시오. 완료된 작업 목록에 HelloWorld 작업이 표시됩니다.

## 작업 결과 보기

이 주제에서는 "Hello World" 연습의 일부로 작업 결과를 보고 빌드 시스템이 올바르게 설치되었는지 확인하는 방법에 대해 설명합니다.

### 시작하기 전에

HelloWorld 프로젝트를 정의 및 실행해야 합니다.

### 완료된 작업 목록 보기

### 프로시저

1. 작업을 클릭하십시오.
2. 완료 탭을 클릭하십시오. 완료된 작업 목록에서는 HelloWorld 작업이 성공적으로 완료된 것으로 표시합니다.

## 단계 로그 보기

### 프로시저

1. 작업을 클릭하십시오.
2. 완료 탭을 클릭하십시오.
3. HelloWorld 작업에 대한 작업 태그를 클릭하십시오. 초기 작업의 기본 작업 태그는 BUILD\_1입니다. 시스템은 작업 단계 이름을 표시합니다. 이 예제에서 작업에는 하나의 단계, EchoHelloWorld만 포함합니다.
4. EchoHelloWorld 단계를 클릭하여 해당 로그를 검사하십시오. 대부분의 Hello World 예제에서는 콘솔 창이나 팝업 창에 "Hello World" 텍스트가 표시됩니다. 관리 콘

솔은 대상 서버의 에이전트 프로세스로 명령을 전송함으로써 작업을 수행합니다. 그런 다음, 에이전트는 이러한 명령의 출력을 다시 관리 콘솔로 보내고 관리 콘솔은 해당 출력을 단계 로그에 저장합니다. 로그에는 많은 섹션이 있습니다. 관련 섹션은 최종 EXEC 섹션입니다. 모든 선택란을 지우고 EXEC 선택란을 선택하고 새로 고치기 링크를 클릭하여 EXEC 섹션만 표시할 수 있습니다. 명령 결과는 아래 표시 됩니다.

```
80 04/19/10 11:06AM EXEC Locale set to 'English_United States.1252'
253 04/19/10 11:06AM EXEC Locale set to 'English_United States.1252'
354 04/19/10 11:06AM EXEC Performing variable expansion on command line
356 04/19/10 11:06AM EXEC start [C:\buildforgeprojects\HelloWorld\BUILD_1@mcsystem]
357 04/19/10 11:06AM EXEC Hello World
358 04/19/10 11:06AM EXEC end [C:\buildforgeprojects\HelloWorld\BUILD_1@mcsystem]
```

## 결과

이 프로젝트는 시스템을 올바르게 구성했으며, 프로젝트가 서버에 성공적으로 액세스하여 서버에서 실행되고 출력을 생성할 수 있음을 보여줍니다. echo 명령을 대상 서버에서 실행할 수 있는 명령으로 바꿀 수 있습니다.

---

## 프로젝트 샘플

시스템에 익숙해지도록 도움을 주는 프로젝트 샘플이 포함되어 있습니다.

프로젝트 샘플에는 다음 디렉토리가 있습니다.

```
<bfinstall>/samples/projects/
```

기본적으로 Build Forge 설치 디렉토리 또는 <bfinstall>은 Windows의 경우 C:\Program Files\IBM\Build Forge 및 UNIX와 Linux의 경우 /opt/buildforge 입니다.

샘플 프로젝트를 사용하려면 다음을 수행하십시오.

- bfimport 명령을 사용하여 샘플 프로젝트를 관리 콘솔로 가져오십시오.
- 프로젝트를 실행하십시오.

---

## 제 3 장 IBM® Rational® Build Forge 문서 주의사항

© Copyright IBM Corporation 2003, 2013.

U.S. Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

이 정보는 미국에서 제공되는 제품 및 서비스용으로 작성된 것입니다.

IBM은 다른 국가에서 이 책에 기술된 제품, 서비스 또는 기능을 제공하지 않을 수도 있습니다. 현재 사용할 수 있는 제품 및 서비스에 대한 정보는 한국 IBM 담당자에게 문의하십시오. 이 책에서 IBM 제품, 프로그램 또는 서비스를 언급했다고 해서 해당 IBM 제품, 프로그램 또는 서비스만을 사용할 수 있다는 것을 의미하지는 않습니다. IBM의 지적 재산을 침해하지 않는 한, 기능상으로 동등한 제품, 프로그램 또는 서비스를 대신 사용할 수도 있습니다. 그러나 비IBM 제품, 프로그램 또는 서비스의 운영에 대한 평가 및 검증은 사용자의 책임입니다.

IBM은 이 책에서 다루고 있는 특정 내용에 대해 특허를 보유하고 있거나 현재 특허 출원 중일 수 있습니다. 이 책을 제공한다고 해서 특허에 대한 라이선스까지 부여하는 것은 아닙니다. 라이선스에 대한 의문사항은 다음으로 문의하십시오.

135-700

서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩

한국 아이.비.엠 주식회사

고객만족센터

전화번호: 080-023-8080

2바이트(DBCS) 정보에 관한 라이선스 문의는 한국 IBM 고객만족센터에 문의하거나 다음 주소로 서면 문의하시기 바랍니다.

*Intellectual Property Licensing*

*Legal and Intellectual Property Law*

*IBM Japan, Ltd.*

*1623-14, Shimotsuruma, Yamato-shi*

*Kanagawa 242-8502 Japan*

다음 단락은 현지법과 상충하는 영국이나 기타 국가에서는 적용되지 않습니다. IBM은 타인의 권리 비침해, 상품성 및 특정 목적에의 적합성에 대한 묵시적 보증을 포함하여 (단, 이에 한하지 않음) 묵시적이든 명시적이든 어떠한 종류의 보증 없이 이 책을 "현 상태대로" 제공합니다. 일부 국가에서는 특정 거래에서 명시적 또는 묵시적 보증의 면책사항을 허용하지 않으므로, 이 사항이 적용되지 않을 수도 있습니다.

이 정보에는 기술적으로 부정확한 내용이나 인쇄상의 오류가 있을 수 있습니다. 이 정보는 주기적으로 변경되며, 변경된 사항은 최신판에 통합됩니다. IBM은 이 책에서 설명한 제품 및/또는 프로그램을 사전 통지 없이 언제든지 개선 및/또는 변경할 수 있습니다.

이 정보에서 언급되는 비IBM의 웹 사이트는 단지 편의상 제공된 것으로, 어떤 방식으로든 이들 웹 사이트를 옹호하고자 하는 것은 아닙니다. 해당 웹 사이트의 자료는 본 IBM 제품 자료의 일부가 아니므로 해당 웹 사이트 사용으로 인한 위험은 사용자 본인이 감수해야 합니다.

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

(i) 독립적으로 작성된 프로그램과 기타 프로그램(본 프로그램 포함)간의 정보 교환 및  
(ii) 교환된 정보의 상호 이용을 목적으로 본 프로그램에 관한 정보를 얻고자 하는 라이선스 사용자는 다음 주소로 문의하십시오.

135-700

서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩

한국 아이.비.엠 주식회사

고객만족센터

이러한 정보는 해당 조건(예를 들면, 사용료 지불 등)하에서 사용될 수 있습니다.

이 정보에 기술된 라이선스가 부여된 프로그램 및 프로그램에 대해 사용 가능한 모든 라이선스가 부여된 자료는 IBM이 IBM 기본 계약, IBM 국제 프로그램 라이선스 계약(IPLA) 또는 이와 동등한 계약에 따라 제공한 것입니다.

본 문서에 포함된 모든 성능 데이터는 제한된 환경에서 산출된 것입니다. 따라서 다른 운영 환경에서 얻어진 결과는 상당히 다를 수 있습니다. 일부 성능은 개발 단계의 시스템에서 측정되었을 수 있으므로 이러한 측정치가 일반적으로 사용되고 있는 시스템에서도 동일하게 나타날 것이라고는 보증할 수 없습니다. 또한 일부 성능은 추정을 통해 추측되었을 수도 있으므로 실제 결과는 다를 수 있습니다. 이 책의 사용자는 해당 데이터를 본인의 특정 환경에서 검증해야 합니다.

비IBM 제품에 관한 정보는 해당 제품의 공급업체, 공개 자료 또는 기타 범용 소스에서부터 얻은 것입니다. IBM에서는 이러한 비IBM 제품을 반드시 테스트하지 않았으므로, 이들 제품과 관련된 성능의 정확성, 호환성 또는 기타 주장에 대해서는 확인할 수 없습니다. 비IBM 제품의 성능에 대한 의문사항은 해당 제품의 공급업체에 문의하십시오.

IBM이 제시하는 방향 또는 의도에 관한 모든 언급은 특별한 통지 없이 변경될 수 있습니다.

이 정보에는 일상의 비즈니스 운영에서 사용되는 자료 및 보고서에 대한 예제가 들어 있습니다. 이들 예제에는 개념을 가능한 완벽하게 설명하기 위하여 개인, 회사, 상표 및 제품의 이름이 사용될 수 있습니다. 이들 이름은 모두 가공의 것이며 실제 기업의 이름 및 주소와 유사하더라도 이는 전적으로 우연입니다.

## 저작권 라이선스

이 정보에는 여러 운영 플랫폼에서의 프로그래밍 기법을 보여주는 원어로 된 샘플 응용프로그램이 들어 있습니다. 귀하는 이러한 샘플 프로그램의 작성 기준이 된 운영 플랫폼의 애플리케이션 프로그래밍 인터페이스(API)에 부합하는 애플리케이션을 개발, 사용, 판매 또는 배포할 목적으로 추가 비용 없이 이들 샘플 프로그램을 어떠한 형태로든 복사, 수정 및 배포할 수 있습니다. 이러한 샘플 프로그램은 모든 조건하에서 완전히 테스트된 것은 아닙니다. 따라서 IBM은 이러한 프로그램의 신뢰성, 서비스 가능성 또는 기능을 보증하거나 진술하지 않습니다. 샘플 프로그램은 인체의 보증 없이 "현상태대로" 제공됩니다. IBM은 샘플 프로그램의 사용으로 발생된 어떠한 손해에 대하여도 책임을 지지 않습니다.

이러한 샘플 프로그램 또는 파생 제품의 각 사본이나 그 일부에는 반드시 다음과 같은 저작권 표시가 포함되어야 합니다.

© (귀하의 회사명) (연도). 이 코드의 일부는 IBM Corp.의 샘플 프로그램에서 파생됩니다. © Copyright IBM Corp. 2003, 2013.

이 정보를 소프트웨어로 확인하는 경우에는 사진과 컬러 삽화가 제대로 나타나지 않을 수도 있습니다.

## 상표 정보

<a href="http://www.ibm.com/legal/copytrade.shtml">www.ibm.com/legal/copytrade.shtml</a> 의 내용을 참조하십시오.
--





---

## 제 4 장 요구사항

Rational® Build Forge® 제품 컴포넌트는 하드웨어 및 소프트웨어 요구사항을 갖고 있습니다.

다음 영역에 대한 요구 사항이 나와 있습니다.

- 하드웨어, 소프트웨어 및 데이터베이스
- Installation Manager
- Licensing
- IPv6 지원에 대한 네트워킹 요구사항
- 통합
- 자국어 지원

---

### 하드웨어, 소프트웨어 및 데이터베이스 요구사항

하드웨어, 소프트웨어 및 데이터베이스 요구사항은 특정 제품에 대한 세부 시스템 요구 사항 사이트에서 조회를 통해 보고서에서 확인할 수 있습니다. 사용 중인 제품, 제품 에디션, 버전 및 운영 체제를 제공합니다. 하드웨어, 운영 체제, 데이터베이스, 에이전트 호스트 및 웹 클라이언트에 대한 요구사항이 규정되어 있습니다.

#### 데이터베이스 요구사항

Rational Build Forge를 사용하려면 데이터베이스가 설치되어 있어야 합니다. Clearing House의 하드웨어 및 소프트웨어 요구사항에서 지원되는 데이터베이스를 확인하십시오.

#### 제품에 포함된 데이터베이스 인타이틀먼트

DB2 Workgroup 버전 9.7 데이터베이스는 Passport Advantage에서 개별적으로 다운로드하여 사용할 수 있습니다. 사용자의 제품 라이선스에 이 제품에 대한 라이선스가 포함되어 있습니다. 제품과 함께 DB2 Workgroup 9.7를 사용하려는 경우 Passport Advantage에서 이를 다운로드하여 콘솔을 설치하기 전에 먼저 설치해야 합니다. Rational Build Forge에서 사용하는 것과 관련된 플랫폼 또는 호스트 위치 제한사항은 없습니다.

참고: DB2 9.1.1은 이전에 제품에 포함되어 제공되었습니다. 현재 이 기능은 지원되지  
만 더 이상 제품에 포함되어 제공되지 않습니다.

---

## Installation Manager 요구사항

Installation Manager가 설치된 경우 Build Forge를 설치하기 위한 버전 요구사항을 만족해야 합니다.

설치 프로그램이 설치된 Installation Manager를 찾으려고 합니다. 버전 요구사항을 만족해야 합니다.

설치 프로그램이 찾지 못하는 경우 Installation Manager를 설치합니다.

- 최소 지원 버전은 1.3.0입니다.
- 콘솔 설치를 위해 Solaris 11을 사용할 경우 Installation Manager 1.4.4를 사용하십시오.

---

## 제품 개정판에 대한 라이선싱 요구사항

사용하는 라이선싱 메커니즘은 가지고 있는 제품 개정판에 따라 다릅니다. 1 페이지의 『Build Forge 제품 개정판』의 내용을 참조하십시오.

개정판	라이선스 메커니즘
Rational Build Forge Standard Edition	라이선스 서버
Rational Build Forge Enterprise Edition	라이선스 서버
Rational Build Forge Enterprise Plus Edition	라이선스 서버

이 주제에는 Build Forge의 Rational License Server 설정에 대한 세부사항이 포함되어 있습니다. Build Forge UI를 통해 새 라이선스 서버를 지정할 수 있습니다. IBM Installation Manager를 통한 라이선스 관리는 지원되지 않습니다.

### 설치 중에 라이선스 파일 지정

이 주제는 파일 기반 라이선싱이 필요한 Build Forge Enterprise Plus Edition을 사용하는 고객에게만 적용됩니다.

설치 중에 라이선스 파일을 지정하려면 다음 단계를 완료하십시오.

1. Passport Advantage에서 라이선스 키 파일인 irbf\_license.properties를 확보하십시오. 해당 파일을 루트 설치 디렉토리에 배치하십시오.

Windows	C:\Program Files\IBM\Build Forge
UNIX/Linux	/opt/buildforge

2. Installation Manager 설치 프로그램의 라이선스 서버 구성 페이지에서 라이선스 파일을 선택하십시오.
3. 찾아보기를 클릭하여 루트 설치 디렉토리에서 라이선스 키 파일을 찾으십시오.

4. **irbf\_license.properties** 파일을 두 번 클릭하여 해당 파일을 선택하고 다음을 클릭하여 계속해서 설치를 수행하십시오.

## Build Forge의 Rational License Server 구성

라이선스 관리자가 Rational License Server를 설정하고 설치하는 동안 지정한 라이선스 서버 호스트 이름을 제공합니다.

이 절은 서버 기반 라이선싱이 필요한 제품 개정판에만 적용됩니다.

- Build Forge Standard Edition
- Build Forge Enterprise Edition

### 라이선스 서버에 대한 Build Forge 구성 요구사항

Build Forge의 라이선스 서버를 구성하기 전에 다음 요구사항을 검토합니다.

- 사용자 환경의 모든 관리 콘솔 호스트 컴퓨터가 Rational License Server 호스트 컴퓨터에 연결될 수 있어야 합니다.
- UNIX/Linux 또는 Windows Rational License Server가 자동으로 시작되고 서비스로 실행되도록 설정되어 있어야 합니다.
- 제품을 설치하고 라이선스 서버를 사용하도록 구성하려면 라이선스 관리자가 Rational License Server의 호스트 이름과 포트 번호를 제공해야 합니다.
- 라이선스 서버와 통신하는 데 필요한 FLEXlm 클라이언트 소프트웨어가 관리 콘솔 설치의 파트로 설치되고 구성됩니다.

**참고:** Build Forge 버전 7.1.3.4에서 Rational License Key Server의 지원 버전이 7.x에서 8.1.2로 변경되었습니다. 7.1.3.4 이전 버전에서 업그레이드하려면 이 라이선스 서버도 업그레이드해야 합니다. 자세한 정보는 자세한 시스템 요구사항 보고서를 참조하십시오.

### 설치 중에 라이선스 서버 지정

설치하는 동안 Rational License Server의 호스트 이름 및 TCP/IP 포트를 제공합니다. 라이선스 관리자로부터 이 정보를 얻습니다.

서버 기반 라이선스를 구성하려면 다음을 수행하십시오.

1. Installation Manager 설치 프로그램의 라이선스 서버 구성 페이지에서 **라이선스 서버**를 선택하십시오.
2. (UNIX 및 Linux) **Build Forge**를 실행해야 하는 사용자에서, 기본 사용자(루트)를 승인하거나 다른 사용자를 지정하십시오. 다른 사용자를 지정하는 경우에도 루트로서 Build Forge를 시작해야 하지만, Build Forge는 설치 단계 중에 지정된 사용자로서 엔진과 Apache Tomcat을 시작합니다. 사용자는 Installation Manager의 데이터베이스 구성 페이지에 지정된 JDBC jar 파일 및 데이터베이스 라이브러리에 대한 읽기 및 실행 권한이 있어야 합니다.

3. 라이선스 서버에 Rational License Server에 대한 올바른 호스트 이름을 제공하십시오.

호스트 이름을 제공하면 이 정보가 Build Forge 시스템 설정에 자동으로 추가됩니다.

호스트 이름을 모르는 경우 이 필드에 특성 또는 값을 입력하고 설치가 완료된 후 UI에서 Build Forge 시스템 설정을 업데이트합니다.

**중요사항:** 이 필드를 공백으로 두지 마십시오. 이 필드를 공백으로 두면 설치가 불완전하게 되어 제품을 사용할 수 없게 될 수 있습니다.

4. 포트에 라이선스 서버에 대한 TCP/IP 포트를 제공하십시오. 기본 포트는 27000입니다.
5. 다음을 클릭하여 계속해서 설치를 수행하십시오.

### 관리 콘솔의 라이선스 서버 변경

관리 콘솔에 대한 Rational License Server를 변경하려면 제품 라이선스 서버 구성을 다음과 같이 수정합니다. 다음과 같은 경우 이 프로시저를 수행해야 합니다.

- 설치하는 동안 잘못된 호스트 이름 또는 기타 값을 입력한 경우
  - 설치하는 동안 라이선스 서버 호스트 이름이 회색으로 표시되는 경우(FLEXlm 클라이언트가 호스트에 대한 라이선스 서버를 이미 등록했음을 나타냄)
1. 이 단계는 Build Forge 버전 7.1에서 7.1.3.3에만 적용됩니다. 이후 버전의 경우 단계 2로 진행하십시오. 사용 중인 운영 체제에서 RATIONAL\_LICENSE\_FILE 변수 값을 변경하십시오.

Windows	RATIONAL_LICENSE_FILE이 HKEY_LOCAL_MACHINE\SOFTWARE\FLEXlm License Manager의 레지스트리에 있습니다.
UNIX/Linux	RATIONAL_LICENSE_FILE이 Build Forge를 실행하는 사용자의 홈 디렉토리에 있는 .flexlmrc 파일에 설정되어 있습니다.

이 변수를 올바른 라이선스 서버의 호스트 이름으로 설정하십시오.

port@hostname | @hostname

라이선스 서버 포트가 27000(기본값)인 경우에는 포트가 필요하지 않습니다.

2. 관리 콘솔 UI에서 관리 → 시스템을 선택하십시오.
3. 라이선스 서버 설정을 찾고 해당 값을 새 Rational License Server의 호스트 이름으로 설정하십시오.

다음 형식 중 하나를 사용하십시오.

<host\_name>:<port> | <host\_name> | <port>:<host\_name>

라이선스 서버 포트가 27000(기본값)인 경우에는 포트가 필요하지 않습니다.

4. 웹 브라우저에서 새로 고침을 클릭하여 관리 콘솔을 새 라이선스 서버에 연결할 수 있는지 확인하십시오.

## 라이선스 키 얻기 및 Rational Licensing Server 설정

기존 라이선스 서버를 사용할 수 없는 경우 다음 표에서 라이선스 관리자가 라이선스 키를 얻고 Rational License Server를 설정하기 위해 완료하는 일반 태스크를 식별합니다.

Rational License Server를 설치하고 구성하려면 Rational License Server 소프트웨어의 버전에 대한 문서를 검토하십시오. <http://www-306.ibm.com/>으로 이동하여 지원 및 다운로드를 선택하고 Rational 라이선스 관리 안내서를 검색하십시오.

라이선스 관리자 태스크	자원
Rational License 키 센터에서 라이선스 키를 얻습니다.	Rational 라이선스 관리 안내서  <a href="http://www-306.ibm.com/software/rational/support/licensing">http://www-306.ibm.com/software/rational/support/licensing</a> 에 있는 Rational License 키 센터의 빠른 시작 안내서
관리 콘솔 호스트와 Rational License Server의 네트워크 연결성을 확인합니다.	Rational 라이선스 관리 안내서
다음을 수행하여 필요한 버전의 Rational License Server 소프트웨어를 얻습니다. <ul style="list-style-type: none"> <li>• 제품 분배 시 포함되어 있는 소프트웨어에 액세스합니다.</li> <li>• 다음 주소에 있는 IBM Rational 다운로드 및 라이선싱 센터에서 소프트웨어를 다운로드합니다.</li> </ul> <a href="https://www14.software.ibm.com/webapp/iwm/web/reg/signup.do?source=rational">https://www14.software.ibm.com/webapp/iwm/web/reg/signup.do?source=rational</a>	Rational 라이선스 관리 안내서
Rational License Server를 설치합니다.	Rational 라이선스 관리 안내서
다음과 같이 라이선스 키를 설치하거나 Rational License Server로 가져옵니다. <ul style="list-style-type: none"> <li>• Windows의 경우 많은 IBM Rational 제품 및 Rational License Server 소프트웨어와 함께 설치된 IBM Rational LKAD(License Key Administrator)를 사용합니다.</li> <li>• UNIX/Linux의 경우 다음 주소에 있는 IBM Rational 다운로드 및 라이선싱 센터에서 license_setup 스크립트 및 라이선싱 실행 파일을 사용합니다.</li> </ul> <a href="https://www14.software.ibm.com/webapp/iwm/web/reg/signup.do?source=rational">https://www14.software.ibm.com/webapp/iwm/web/reg/signup.do?source=rational</a>	Rational 라이선스 관리 안내서

라이선스 관리자 태스크	자원
<p>Rational License Server를 시작하고 다음과 같이 Rational License Server를 자동으로 시작하고 서비스로 실행되도록 설정합니다.</p> <ul style="list-style-type: none"> <li>Windows의 경우 컴퓨터를 시작할 때 서비스로 시작하도록 Rational License Server가 자동으로 설정됩니다. 자동으로 시작되지 않는 경우 Windows에서 라이선스 서버를 자동으로 시작할 수 있는 지시사항을 참조하십시오.</li> <li>UNIX/Linux의 경우 제공된 템플릿 시작 스크립트를 사용하여 시작 스크립트를 작성하고 설치를 위해 해당 스크립트를 수정합니다. 다음 주소에 있는 IBM Rational 다운로드 및 라이선싱 센터에서 템플릿을 얻습니다.</li> </ul> <p><a href="https://www14.software.ibm.com/webapp/iwm/web/reg/signup.do?source=rational">https://www14.software.ibm.com/webapp/iwm/web/reg/signup.do?source=rational</a></p>	Rational 라이선스 관리 안내서

## IPv6 지원에 대한 네트워킹 요구사항

Build Forge 시스템을 인터넷 프로토콜 버전 6(IPv6) 및 혼합 IPv6-IPv4 네트워크에서 실행하는 데 몇 가지 제한사항이 있습니다.

IPv6을 지원하려면 컴퓨터와 네트워크가 IPv6을 지원하도록 올바르게 구성되어 있어야 합니다. 네트워크 구성 문제점은 Build Forge 시스템에서 지정된 호스트 이름과 주소가 올바르게 해석되지 않도록 합니다.

IPv6에 대해 Build Forge를 수동으로 구성해야 합니다. 이를 수행하려면 105 페이지의 『httpd.conf 수정』의 내용을 참조하십시오.

Build Forge에서 IP 주소를 입력할 때 올바른 주소 형식을 사용합니다. Build Forge에서는 관리자 또는 사용자가 일부 위치에만 호스트 이름 또는 IP 주소를 입력합니다.

- 설치하는 동안 관리자가 관리 콘솔에서 라이선스 서버(Standard 및 Enterprise Editions) 및 데이터베이스와 통신하는 데 사용하는 호스트 이름 또는 IP 주소를 지정합니다.
- 에이전트 구성(bfagent.conf 파일)에서 선택적 설정을 통해 에이전트에 대한 연결을 특정 주소 또는 주소 범위로 제한합니다.
- 사용자가 관리 콘솔 사용자 인터페이스를 보려면 브라우저에 URL을 입력합니다. URL은 관리 콘솔이 실행되는 서버의 호스트 이름 또는 IP 주소로 구성됩니다. 예를 들어, IPv4 및 IPv6 주소가 구성된 BFmachine이라는 서버에 설치된 관리 콘솔에 액세스하기 위해 사용자는 웹 브라우저에 다음과 같은 주소를 입력할 수 있습니다.

– <http://BFmachine/>



- http://localhost/
- http://127.0.0.1/(IPv4 루프백 주소)
- http://::127.0.0.1/, http://0:0:0:0:0:0:127.0.0.1/(IPv4 루프백 주소의 IPv6 약어) 또는 간단히 http://::1/(루프백 주소의 IPv6 압축 표기법)

IPv4 및 IPv6 주소는 형식 및 길이가 다릅니다.

- **IPv4 형식:** 길이는 32비트입니다. 주소는 소수점으로 구분된 네 개의 10진수 값(예 : 255.255.255.255)으로 지정됩니다.
- **IPv6 형식:** 길이는 128비트입니다. 주소는 콜론으로 구분된 8개의 16진 값으로 지정됩니다(예: FE80:0000:0000:0000:0202:B3FF:FE1E:8329). 상위 치수 필드 사용에 대한 여러 가지 규칙이 있습니다. 약어에 대한 규칙도 있습니다. Build Forge는 IP 주소를 해석하지 않고 네트워크에 직접 전달합니다. 따라서 올바르고 유효한 약어는 작동합니다. IPv6 주소 규칙에 대한 자세한 정보는 기타 참조를 확인하십시오.

## **/etc/hosts에서 localhost 별명 파하기**

IPv4와 IPv6을 둘 다 사용할 경우 UNIX 및 Linux 시스템의 /etc/hosts에서 별명으로 localhost를 사용하지 마십시오. 서버의 완전한 도메인 이름 또는 IP 주소를 사용하십시오.

localhost를 /etc/hosts에서 호스트 이름으로 사용해야 할 경우 IPv4 주소 127.0.0.1만 정의해야 합니다.

## **IPv6을 지원하지 않는 컴포넌트**

IPv6를 지원하지 않는 컴포넌트는 IPv4 주소를 갖는 호스트 컴퓨터에 설치되어야 합니다. 이러한 컴포넌트를 IPv4 주소가 있는 컴퓨터에 설치합니다.

- Build Forge Standard 및 Enterprise Edition의 라이선스 서버인 Rational License Server 7.0이 제품 분배 시 포함되어 있습니다.

IPv6 주소를 지원하려면 Rational License Server 7.1을 설치해야 합니다. IBM Rational 지원 또는 IBM Rational 다운로드 및 라이선싱 센터에서 다운로드합니다. IPv6을 지원하는 FLEXlm 클라이언트 소프트웨어의 버전도 구성해야 합니다.106 페이지의 『FLEXlm 클라이언트 설정』의 내용을 참조하십시오.

- DB2 데이터베이스: PHP 데이터베이스 드라이버는 아직 IPv6를 지원하지 않습니다.
- MySQL 데이터베이스: PHP 데이터베이스 드라이버는 아직 IPv6를 지원하지 않습니다.
- 11g-the PHP 데이터베이스 이외의 Oracle 데이터베이스 드라이버는 아직 IPv6를 지원하지 않습니다.

---

## 자국어 지원 요구사항

Build Forge 제품은 프랑스어, 독일어, 이탈리아어, 포르투갈어(브라질), 스페인어, 일본어, 한국어, 중국어 및 대만어로 된 자국어 지원을 제공합니다.

이 절에서는 Build Forge의 언어 지원에 대한 정보를 제공합니다.

### 관리 콘솔 및 에이전트에 대한 언어 설정

이 주제에서는 Build Forge가 관리 콘솔, 에이전트 및 엔진 컴포넌트의 언어 설정을 구성하는 방법에 대해 설명합니다.

#### 관리 콘솔 언어 설정

현재 사용자를 위한 언어 설정은 관리 콘솔의 인터페이스 컨트롤을 표시하는데 사용되는 언어를 결정합니다.

다음과 같이 사용자 계정에 대한 관리 콘솔의 언어를 설정하십시오.

##### 루트 사용자

설치 프로그램은 루트 사용자 계정을 생성하며 관리자는 그 계정을 통해 처음으로 콘솔에 로그인합니다.

최초 로그인 시, 루트 사용자를 위한 언어 설정은 Build Forge 엔진 호스트의 운영 체제 언어를 기본으로 합니다.

##### 기타 사용자의 기본 언어

관리자가 생성한 사용자 계정에 있어서 기본 콘솔 언어는 사용자가 로그인하는 웹 브라우저의 언어 설정을 초기값으로 합니다. (이것은 Build Forge가 아닌, 웹 브라우저의 구성 설정입니다.)

관리자가 사용자에게 대한 언어 환경 설정을 변경하지 않으면 해당 값이 웹 브라우저로부터 상속됩니다.

관리자가 사용자에게 대한 언어를 변경하려는 경우 다음 프로시저를 완료합니다. UI에서 **관리 > 사용자 > 언어**를 클릭한 후 언어를 선택하십시오.

LDAP에서 작성된 사용자 계정에서는 항상 웹 브라우저에 구성된 언어 환경 설정을 사용합니다.

##### 사용자 기본 언어 변경

콘솔에 로그인한 후 관리자는 Build Forge 창의 왼쪽 탐색 패널에서 **관리 > 사용자 > 언어**를 선택하여 개별 Build Forge 사용자에게 대해 서로 다른 언어 설정을 선택할 수 있습니다.

사용자를 위한 언어 설정이 이런 식으로 이루어 지는 경우, 관리 콘솔은 웹 브라우저에서 설정된 언어에 관계 없이 사용자를 위해 선택된 언어로 인터페이스 컨트롤을 표시합니다.

## 에이전트 및 Build Forge 엔진 언어 설정

Build Forge 엔진 호스트의 운영 체제 언어는 다음을 의미합니다.

- Build Forge 엔진에서 사용하는 언어
- Build Forge 에이전트의 기본 언어

에이전트 언어 설정이 시스템 메시지 및 작업 출력의 언어를 제어합니다.

관리 콘솔 언어에 대해 설정된 언어가 웹 브라우저 또는 Build Forge 환경 설정에 설정되어 있는지 여부에 관계없이 에이전트는 Build Forge 엔진 호스트의 운영 체제 언어로 시스템 메시지 및 작업 출력에 대한 데이터의 로그를 기록합니다.

관리 콘솔 인터페이스에서 여러 언어가 혼합되지 않도록 하려면 관리 콘솔에 대해 선택된 언어가 Build Forge 엔진 호스트에서 사용되는 언어와 일치해야 합니다.

### 언어 설정

관리 콘솔에 표시되는 언어가 에이전트에서 시스템 메시지 및 작업 출력에 대한 데이터의 로그를 기록하는 데 사용하는 언어와 일치되도록 하려면 웹 브라우저, 관리 콘솔 호스트의 운영 체제 및 Build Forge 사용자에게 동일한 언어를 사용합니다.

## 데이터베이스 호스트에 대한 국제적 데이터 지원

국제적 데이터를 표시하고 조작하려면 다음과 같이 관리 콘솔 데이터베이스에 대한 호스트 컴퓨터를 구성하십시오.

- 유니코드 UTF-8 문자 세트를 사용합니다.
- 데이터를 표시하는 데 사용할 글꼴을 설치합니다.

## 관리 콘솔에서 사용자 언어 환경 설정 변경

모든 관리 콘솔 사용자에게 대한 기본 언어는 초기에 웹 브라우저의 언어로 설정됩니다.

Build Forge 사용자에게 대한 언어 설정을 변경하려면 관리 > 사용자 > 언어를 선택하십시오.

관리 콘솔은 사용자 인터페이스 제어를 사용자가 사용하는 언어로 표시하지만, 에이전트는 시스템 메시지 및 작업 출력에 대한 데이터를 Build Forge 엔진 호스트의 운영 체제 언어로 기록합니다.

## UNIX/Linux 호스트의 언어/문자 세트 판별

관리 콘솔 또는 에이전트가 UNIX/Linux 호스트에 설치된 경우 다음과 같이 locale 명령을 사용하십시오.

- 운영 체제가 현재 사용 중인 언어/문자 세트를 판별하려면 다음과 같이 하십시오.  
\$ locale

- 운영 체제에 사용 가능한 언어/문자 세트 조합을 판별하려면 다음과 같이 하십시오.  
\$ locale -a
- 로그인 시 로케일을 설정하려면 rc 또는 프로파일 스크립트를 사용하십시오.

## Windows 호스트의 언어 코드 페이지 판별

관리 콘솔 또는 에이전트가 Windows 호스트에 설치된 경우, 다음과 같이 chcp 명령을 사용하십시오.

- 활성 코드 페이지 번호를 판별하려면 다음을 입력하십시오.  
> chcp
- 코드 페이지를 설정하려면 언어에 해당하는 번호를 입력하십시오.  
> chcp code\_page

다음 표는 Build Forge가 지원하는 NLV1 언어에 대한 Windows 문자 인코딩을 나열합니다.

언어	코드 페이지
영어	1252
프랑스어	
스페인어	
이탈리아어	
독일어	
포르투갈어	
일본어 Shift-JIS	932
한국어	949
중국어 GBK	936
대만어 Big5	950

## 제 5 장 설치 계획

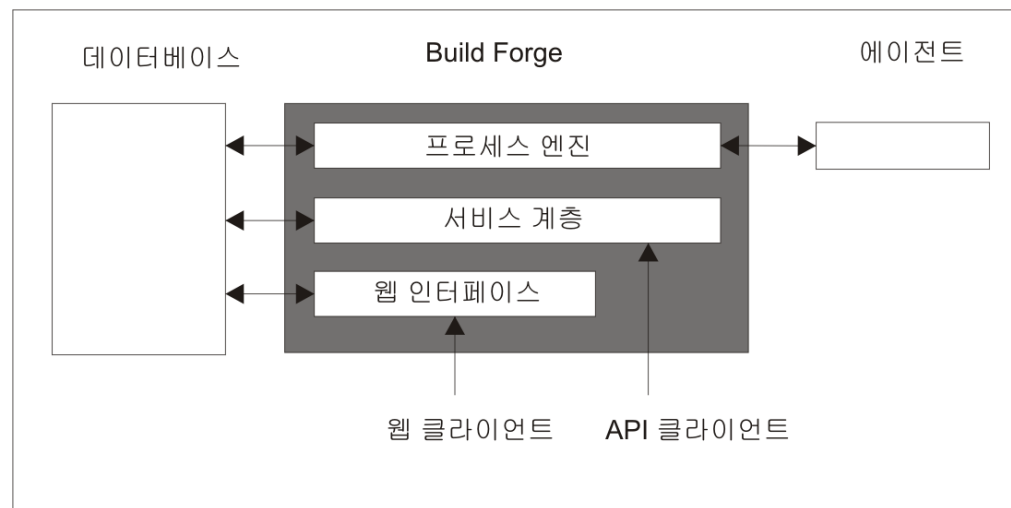
Rational® Build Forge® 설치 시 여러 제품 컴포넌트를 고려해야 합니다.

이 주제에서는 Rational® Build Forge® 제품 컴포넌트를 설치하는 데 필요한 계획에 대해 설명하며, 이는 설치 및 구성할 항목과 방법에 대해 선택할 수 있는 항목에 대한 로드맵 역할을 합니다.

637 페이지의 제 39 장 『성능 및 확장성』에 설명되어 있는 계획 자원도 참조하십시오.

### 컴포넌트

Build Forge® 시스템에는 다음 컴포넌트가 있습니다.



- **웹 클라이언트:** 시스템에 액세스하기 위해 웹 브라우저를 사용하는 사용자 및 관리자가 사용합니다. 브라우저 클라이언트는 웹 인터페이스 컴포넌트에 액세스합니다.
- **API 클라이언트:** Build Forge에 액세스할 때 Java API 또는 Perl API를 사용하는 모든 프로그램입니다. API 클라이언트는 서비스 계층에 직접 액세스합니다.
- **Build Forge:** 시스템에 대한 포괄적인 용어입니다. 설치하는 동안 시스템에 다음과 같은 코어 제품 기능이 구성된 것으로 표시됩니다.
  - **웹 인터페이스:** 관리 콘솔 또는 콘솔이라고도 합니다. 이 컴포넌트는 PHP 모듈 세트로 구성됩니다.
  - **프로세스 엔진:** 엔진이라고도 합니다. 엔진은 작업 스케줄링 및 실행을 관리합니다.

- 서비스 계층: API 클라이언트, 웹 인터페이스 및 프로세스 엔진에서 요청하는 데이터베이스 추상 계층입니다.

단순 설치에서는 세 가지 컴포넌트가 모두 동일한 호스트에 설치됩니다. 이러한 컴포넌트를 개별 호스트에 설치할 수 있습니다. 이러한 경우 컴포넌트 간에 통신할 수 있도록 Build Forge 구성 파일 buildforge.conf를 수정해야 합니다.

- 데이터베이스: 시스템의 정보 스토리지입니다. 데이터베이스는 프로젝트 정의, 시스템 구성 및 사용자 구성을 저장합니다.
- 에이전트: 호스트에 설치된 프로그램입니다. 에이전트는 관리 콘솔이 서버 자원으로 사용할 모든 호스트에 설치해야 합니다. 에이전트가 작업(단계) 수행 요청을 수신하여 에이전트가 설치되어 있는 호스트에서 해당 작업을 실행합니다.

컴포넌트는 여러 가지 방법으로 배치할 수 있습니다. 단일 호스트에 모든 컴포넌트를 설치하거나 클러스터 콘솔 및 다수의 분배 서버 자원을 사용하는 시스템에 설치할 수도 있습니다.

---

## 배치 유형

이 절에서는 다음과 같은 배치 유형을 설명합니다.

- 독립형
- 일반 프로덕션 설치
- 계층별 설치

### 독립형

독립형 배치에는 단일 호스트에 배치된 하나의 에이전트, 관리 콘솔 및 데이터베이스가 포함됩니다. 일반적으로, 이 배치는 평가 또는 개발 용도로 사용됩니다.

### 일반

일반 설치에서는 다른 호스트에 데이터베이스, 관리 콘솔 및 에이전트를 배치합니다. 이 배치는 가장 일반적으로 사용되는 프로덕션 환경입니다.

### 계층별 설치

계층별 배치에는 일반적으로 다음과 같은 세 개의 일반 배치가 있습니다.

- 개발(스크래치 또는 샌드박스라고도 함): 새 프로젝트를 작성하기 위해 개발 직원이 사용합니다. 또한 업그레이드의 초기 영역으로도 사용됩니다.
- 테스트: 시스템을 테스트하기 위해 품질 보증 및 개발 직원이 사용합니다.
- 프로덕션: 일상적인 비즈니스 과정에서 작업을 실행하기 위해 모든 사용자가 사용합니다.

Build Forge 데이터는 하나의 계층에서 가져와 다음 계층으로 내보내집니다.

**중요사항:** 세 개의 배치는 서로 대칭적이어야 합니다. 각 배치에서 동일한 운영 체제, 데이터베이스 및 컴포넌트를 사용하십시오.

---

## 설치 방법

이 절에서는 사용자가 사용할 수 있는 설치 방법을 설명합니다.

- Installation Manager, 대화식: 설치가 수행되면 입력 제공
- Installation Manager, 자동: 파일에 입력을 저장하고 해당 파일을 사용하여 설치 중 입력 제공
- 고유한 컴포넌트를 사용하는 설치: Build Forge에 필요한 기존 컴포넌트를 사용하여 계속 진행
- 가상 이미지에서 설치: VMware에서 설치에 대한 가이드라인 읽기
- IBM System z에서 설치: System z의 SUSE Linux에서 콘솔 설치에 필요한 정보 및 도구 이해

---

## 구성 옵션

이 주제에서는 제품 구성을 대체하는 데 필요한 기능을 설명합니다.

- 포트 지정: 기본 포트 80이 아닌 다른 포트를 사용하는 방법 학습
- 124 페이지의 『기본 로그인 보안 정보』: 설치 중 키 저장소 비밀번호를 제공합니다. 키 저장소는 추가 설정이 필요한 SSL 및 기본적으로 제공되는 보안 로그인 메커니즘에서 모두 사용됩니다. 원하는 경우 보안 로그인을 사용하지 않을 수 있습니다.
- 비밀번호 암호화: Build Forge 엔진, 에이전트, 서비스 계층 및 데이터베이스에서 사용하는 비밀번호 암호화
- SSL 및 HTTPS: Build Forge 시스템 전체에서 SSL을 사용하는 데 필요한 단계 이해
- 싱글 사인온: 보안을 유지보수하는 중 싱글 사인온 단순화
- 콘솔 중복을 사용하여 더 많은 처리량 지원: 하나의 데이터베이스를 사용하도록 여러 콘솔 설정
- Build Catalyst를 사용하여 작성 기반 빌드 가속화: 단일 시스템 병렬 빌드 및 다중 시스템 분산 빌드 설정

---

## 기타 제품과 통합

- WebSphere: Apache Tomcat 대신 WebSphere Application Server 사용
- Rational Team Concert IDE: RTC 빌드 서버로 Build Forge 설정 및 기타 태스크 수행

- Eclipse 및 Rational Application Developer IDE: Eclipse 및 Rational Application Developer IDE에서 콘솔에 액세스하는 방법 학습
- 어댑터를 사용하여 소스에 액세스: 제공된 어댑터 사용 및 기타 제품과 상호작용하도록 고유한 디자인



---

## 제 6 장 설치 전 설정

이 절에서는 설치 프로그램을 실행하기 전에 필요한 설치 전 설정에 대해 설명합니다.

참고: Build Forge 버전 7.1 이상이 이미 설치되어 있는 경우에는 223 페이지의 제 12 장 『이전 버전에서 업그레이드』의 내용을 참조하십시오.

- 설치 역할(필수). 관리 레벨 사용자로 소프트웨어를 설치해야 합니다.
- 국제적 데이터 지원(필수)을 설정해야 합니다. Build Forge를 설치하기 전에 UTF-8 문자 세트를 지원하도록 웹 브라우저 및 데이터베이스를 구성해야 합니다.
- (필수사항) 데이터베이스를 설정해야 합니다. Build Forge에 사용할 데이터베이스를 설정해야 합니다. 일반적으로 데이터베이스 오브젝트(데이터베이스 사용자 및 비밀번호 포함)를 작성하고, Build Forge가 실행되는 호스트에 데이터베이스 클라이언트를 설치하고, Build Forge를 설치하는 동안 프롬프트된 정보를 어셈블해야 합니다.
- (선택사항)보안을 설정해야 합니다. SSL/HTTPS를 사용하려는 경우 인증서를 제공하거나 Build Forge에서 자가 서명 인증서를 설치하도록 해야 합니다.

---

### 국제적 데이터 설정

국제적 데이터를 지원하도록 Build Forge 컴포넌트를 설정해야 합니다.

#### 프로시저

1. 웹 브라우저를 구성하십시오.
  - a. 언어를 설정하십시오.
  - b. 올바른 글꼴이 설치되어 있는지 확인하십시오.
2. UTF-8 문자 세트를 사용하도록 에이전트 호스트를 구성하십시오.

Windows의 경우 다음과 같은 chcp 명령을 사용하여 코드 페이지를 확인합니다.

```
> chcp
```

UNIX 또는 Linux의 경우 다음 명령을 사용하여 로케일 및 문자 세트를 확인합니다.

```
locale
```

언어 및 문자 세트를 지정하는 값을 확인해야 합니다. 다음은 언어가 미국 영어이고 UTF-8 문자 세트를 사용하는 Solaris 시스템의 예제입니다.

```
LANG=en_US.UTF-8  
LC_CTYPE="en_US.UTF-8"
```

3. 국제적 데이터를 지원하는 UTF-8 문자 세트 및 글꼴을 사용하도록 데이터베이스를 구성하십시오.

- **DB2:**

- a. 코드세트 및 지역을 설정하십시오. 예: `CREATE DATABASE USING CODESET UTF-8 TERRITORY US`(또는 제어 센터에서 해당 코드세트 및 지역을 선택하십시오).
- b. 관리 콘솔 컴퓨터의 `DB2CODEPAGE` 환경 변수를 1208로 설정하십시오.

Windows의 경우 다음 명령을 사용하십시오.

```
set DB2CODEPAGE=1208
```

UNIX 또는 Linux의 경우 다음 명령을 사용하십시오.

```
export DB2CODEPAGE=1208
```

기존 데이터베이스에 UTF-8로 마이그레이션해야 하는 데이터가 있는 경우, 다음 문서를 참조하십시오. <http://publib.boulder.ibm.com/infocenter/db2luw/v9/index.jsp?topic=/com.ibm.db2.udb.admin.doc/doc/t0024033.htm>

- **Microsoft SQL Server**(Windows의 관리 콘솔과 함께 사용하는 경우에만): 국제적 데이터를 지원하지 않습니다. Microsoft SQL Server는 UCS-2를 사용하여 유니코드 데이터를 저장하며 UTF-8을 지원하지 않습니다.
- **MySQL**: 서버 문자 세트 및 조합을 설정하십시오. 현재 MySQL 설치가 국제적 데이터를 지원하지 않는 경우, 소스에서 재컴파일하여 `./configure --with-charset=utf8 --with-collation=utf8_bin`을 사용할 수 있습니다. 이 지원을 구성하지 않으면 Build Forge 엔진이 시작되지 않습니다.
- **Oracle**: 이를 설치한 인스턴스에서 문자 세트를 **UTF8 - Unicode 3.0**으로 설정하십시오. 데이터베이스 구성 보조의 문자 세트 탭에 있는 초기화 매개변수 단계에서 설정됩니다. 명령행을 사용하는 경우 문자 세트를 **AL32UTF8**로 설정하십시오.

4. (선택사항) Build Catalyst 기능에 대한 메시지를 구성하십시오.

Microsoft® Windows® 플랫폼에서 Build Catalyst 메시지는 자동으로 올바른 변환으로 표시됩니다.

이들 메시지를 UNIX 및 Linux 플랫폼에서 보려면 `NLSPATH` 환경 변수를 다음 값으로 설정하십시오.

```
/opt/rational/buildforge/buildcatalyst/lib/nls/%L/utf8/%N.cat:$NLSPATH
```

여기서,

- `%L`은 `LANG` 환경 변수의 값을 대체합니다.

- %N은 catopen(3C)으로 패스된 이름 매개변수의 값을 대체합니다.

또는, 변환을 명시적으로 지정할 수 있습니다. 다음 값은 ja\_JP 변환에 대한 값입니다.

```
/opt/rational/buildforge/buildcatalyst/lib/nls/ja_JP/utf8/%N.cat:$NLSPATH
```

---

## 데이터베이스 백업(필수)

새로 설치, 다시 설치 또는 업그레이드를 진행하기 전에 먼저 데이터베이스를 백업해 두는 것이 중요합니다.

설치 중에 문제가 발생하면 데이터베이스 백업이 데이터베이스를 설치를 시도하기 전의 마지막 상태로 복원하는 기능을 제공합니다. 데이터베이스 백업은 제공되는 유일한 롤백 방법입니다.

---

## 데이터베이스 설정

이 절에는 지원되는 각 데이터베이스에 대한 설정 지시사항이 포함되어 있습니다.

데이터베이스는 관리 콘솔과 동일한 호스트 또는 다른 호스트에 설치할 수 있습니다.

**참고:** Rational Build Forge 7.1 릴리스는 빈 데이터베이스를 사용하여 설치해야 합니다. 데이터베이스가 비어 있지 않은 경우 설치에 실패합니다.

설정하려면 데이터베이스에 따라 다음 태스크 중 일부 또는 모두를 완료해야 합니다.

- 사용할 Build Forge에 대한 데이터베이스 오브젝트(데이터베이스 및 데이터베이스 사용자)를 작성합니다.
- 사용할 Build Forge의 필수 클라이언트 소프트웨어를 설치합니다.
- 설치 중에 필요한 추가 정보를 판별합니다. 일반적으로 이러한 정보는 Apache Tomcat 애플리케이션 서버에서 사용되는 JDBC 드라이버의 위치입니다.
- UTF-8 문자 세트 지원 및 적절한 조합을 구성합니다. 일반적으로 국제적 데이터 지원은 데이터베이스를 작성할 때 지정됩니다. 국제적 데이터 지원은 데이터베이스를 작성한 후에 구성될 수 없습니다. 데이터를 표시하는 데 사용할 글꼴을 설치해야 합니다. Build Forge에서는 국제적 데이터를 사용해야 합니다(UTF-8 문자 세트).

### DB2 Express 설정

Build Forge에 사용할 DB2 Express를 설정하려면 다음 절을 참조하십시오.

#### DB2 Express용 데이터베이스 오브젝트

설치 프로그램을 실행하기 전에 다음 오브젝트를 정의해야 합니다.

- 사용자: 사용자 계정은 데이터베이스 호스트의 Windows에 정의되어야 합니다. 계정은 데이터베이스에 액세스하기 위해 Build Forge에서만 사용되어야 합니다.

- 데이터베이스 오브젝트: DB2 Express 인스턴스에 다음 오브젝트를 작성합니다.
  - 데이터베이스: 데이터베이스를 작성하고 BUILD로 이름을 지정합니다.
  - 스키마: 데이터베이스 스키마를 작성하고 BUILD로 이름을 지정합니다. 작성한 사용자에게 이 스키마에 대한 권한이 있어야 합니다.

## 설치 중에 필요한 DB2 Express 정보

기존 DB2 Express 인스턴스를 사용하므로, 설치하는 동안 데이터베이스 구성 패널에서 **DB2 Express** 설치에 아니오를 선택합니다. 다음 필드가 표시됩니다.

데이터베이스 구성 패널에 추가 정보를 제공해야 합니다.

### 데이터베이스 구성

- 데이터베이스 유형: **DB2**를 선택합니다.
- 설치 시 이 데이터베이스를 채우시겠습니까?: 설치할 때 Build Forge가 자동으로 채우도록 하려면 예를 선택하고 설치 후에 수동으로 채우려면 아니오를 선택하십시오.
- 데이터베이스 호스트: DB2 Express 설치가 있는 호스트 이름을 입력합니다.
- 데이터베이스 이름: 사용할 Build Forge에 대해 작성한 데이터베이스 이름을 입력합니다.
- 데이터베이스 스키마 이름: 사용할 Build Forge에 대해 작성한 스키마 이름을 입력합니다.
- 데이터베이스 포트: 기본값은 50000입니다. 포트 번호가 다른 경우 사용하는 포트 번호를 입력합니다.
- 데이터베이스 사용자 이름: 데이터베이스에 액세스할 때 사용할 Build Forge의 사용자 이름입니다. 사용할 Build Forge에 대해 작성한 Windows 계정을 입력합니다.
- 비밀번호: 데이터베이스 사용자 이름에 대한 비밀번호입니다. 사용자 계정의 비밀번호를 입력합니다.
- 비밀번호 확인: 사용자 계정의 비밀번호를 입력합니다.

### 데이터베이스 구성 테스트

- DB2 클라이언트 라이브러리에 대한 경로: 기본적으로 이 라이브러리는 DB2 Express를 설치한 드라이브의 C:/Program Files/IBM/SQLLIB/java에 있습니다. Build Forge에서는 db2jcc.jar 및 db2jcc\_license\_cu.jar과 같은 파일을 사용해야 합니다.
- JDBC 드라이버 위치: 기본적으로 이러한 라이브러리는 DB2 Express를 설치한 드라이브의 C:/Program Files/IBM/SQLLIB/java에 있습니다. JDBC 드라이버는 Apache Tomcat에서 데이터베이스에 액세스할 때 사용됩니다.

참고: 설치 시 이 데이터베이스를 채우시겠습니까?에 대해 아니오를 선택하면 테스트는 올바른 JDBC 드라이버 위치만 검사합니다.

## DB2 설정

이 프로시저를 사용하여 DB2 지원을 설정하십시오.

### DB2에 대한 Red Hat 및 SuSe Linux 6 요구사항 시작하기 전에

Rational Build Forge를 설치하기 전에 Red Hat Linux 6의 DB2에 대한 추가 설정이 필요합니다.

#### 프로시저

- 호환성 공유 라이브러리를 설치하십시오. 다음 패키지를 설치해야 합니다.
  - libgcc-4.4.4-13.i686
  - glibc.i686(nss-softoken-freebl.i686에 해당).
  - gtk2.i686(다음과 같은 다양한 추가 파일에 해당) atk.i686, audit-libs.i686, avahi-libs.i686, cairo.i686, cups-libs.i686, dbus-libs.i686, expat.i686, fontconfig.i686, freetype.i686, gamin.i686, glib2.i686, gnutls.i686, jasper-libs.i686, keyutils-libs.i686, krb5-libs.i686, libX11.i686, libXau.i686, libXcomposite.i686, libXcursor.i686, libXdamage.i686, libXext.i686, libXfixes.i686, libXft.i686, libXi.i686, libXinerama.i686, libXrandr.i686, libXrender.i686, libcom\_err.i686, libgcrypt.i686, libgpg-error.i686, libjpeg.i686, libpng.i686, libselinux.i686, libstdc++.i686, libtasn1.i686, libthai.i686, libtiff.i686, libxcb.i686, pango.i686, pixman.i686, zlib.i686
  - libXtst.i686
  - libuuid.i686
- 모든 파일은 사전순 목록을 따릅니다. 파일 확장자는 Intel 플랫폼용입니다. System z 플랫폼의 경우 .s390 확장자를 사용하십시오.

```
atk-1.28.0-2.el6.i686
audit-libs-2.0.4-1.el6.i686
avahi-libs-0.6.25-8.el6.i686
cairo-1.8.8-3.1.el6.i686
cups-libs-1.4.2-35.el6_0.1.i686
dbus-libs-1.2.24-4.el6_0.i686
expat-2.0.1-9.1.el6.i686
fontconfig-2.8.0-3.el6.i686
freetype-2.3.11-6.el6_0.2.i686
gamin-0.1.10-9.el6.i686
glib2-2.22.5-5.el6.i686
glibc-2.12-1.7.el6_0.5.i686
gnutls-2.8.5-4.el6.i686
gtk2-2.18.9-4.el6.i686
jasper-libs-1.900.1-15.el6.i686
keyutils-libs-1.4-1.el6.i686
```

```

krb5-libs-1.8.2-3.el6_0.7.i686
libcom_err-1.41.12-3.el6.i686
**libgcc-4.4.4-13.el6.i686
libgcrypt-1.4.5-3.el6.i686
libgpg-error-1.7-3.el6.i686
libjpeg-6b-46.el6.i686
libpng-1.2.44-1.el6.i686
libselinux-2.0.94-2.el6.i686
libstdc++-4.4.4-13.el6.i686
libtasn1-2.3-3.el6.i686
libthai-0.1.12-3.el6.i686
libtiff-3.9.4-1.el6_0.3.i686
libuuid.i686
libX11-1.3-2.el6.i686
libXau-1.0.5-1.el6.i686
libxcb-1.5-1.el6.i686
libXcomposite-0.4.1-2.el6.i686
libXcursor-1.1.10-2.el6.i686
libXdamage-1.1.2-1.el6.i686
libXext-1.1-3.el6.i686
libXfixes-4.0.4-1.el6.i686
libXft-2.1.13-4.1.el6.i686
libXi-1.3-3.el6.i686
libXinerama-1.1-1.el6.i686
libXrandr-1.3.0-4.el6.i686
libXrender-0.9.5-1.el6.i686
libXtst-1.0.99.2-3.el6.i686
nss-softokn-freebl-3.12.8-1.el6_0.i686
pango-1.28.1-3.el6_0.5.i686
pixman-0.18.4-1.el6_0.1.i686
zlib-1.2.3-25.el6.i686

```

- SuSe Linux 64비트 시스템에서 실행 중인 경우 pam.i686 패키지도 설치해야 합니다.

## DB2용 데이터베이스 오브젝트

### 프로시저

1. 운영 체제에서 사용자를 작성하십시오. 관리 콘솔은 이 이름을 사용하여 데이터베이스에 로그인합니다. 예: 사용자 이름 **BUILD**, 비밀번호 **BUILD**.

DB2에서 나머지 단계를 수행하십시오. 중요: 데이터베이스 이름 및 스키마 이름에는 모두 대문자를 사용해야 합니다.

2. **BUILD**라는 데이터베이스를 작성하십시오.
3. **BUILD**라는 스키마를 작성하십시오.

### 결과

참고: Build Forge는 사용자의 스키마를 사용하여 데이터베이스에 액세스합니다.

## 샘플 DB2 SQL 명령 스크립트

DB2 명령행 프로세서에서 다음 명령을 사용하여 테이블스페이스 및 데이터베이스를 작성하십시오.

```
// Create database and schema
db2 "CREATE DATABASE BUILD USING CODESET UTF-8
    TERRITORY US PAGESIZE 16 K AUTOCONFIGURE
    USING MEM_PERCENT 20 APPLY DB ONLY
db2 "CONNECT TO BUILD"
db2 "CREATE SCHEMA BUILD"
db2 "CONNECT RESET"

// Grant permissions to the user
db2 "CONNECT TO BUILD"
db2 "GRANT CREATEIN,DROPIN,ALTERIN ON SCHEMA BUILD
    TO USER BUILD WITH GRANT OPTION"
db2 "GRANT DBADM,CREATETAB,BINDADD,CONNECT,
    CREATE_NOT_FENCED_ROUTINE,IMPLICIT_SCHEMA, LOAD,
    CREATE_EXTERNAL_ROUTINE,QUIESCE_CONNECT
    ON DATABASE TO USER BUILD"
db2 "commit work"
db2 "CONNECT RESET"db2 "terminate"
```

## DB2에 권장되는 매개변수 조정

### 이 태스크 정보

DB2 조정 매개변수를 설정하면 기존 DB2 데이터베이스를 사용하는 Build Forge 시스템의 성능 및 확장성을 개선할 수 있습니다.

**참고:** Build Forge를 설치하고 실행한 후 이러한 매개변수를 변경하는 경우 변경하기 전에 Build Forge를 중지하십시오. DB2를 다시 시작한 후 Build Forge를 다시 시작하십시오.

### 프로시저

1. 조정 매개변수를 설정하십시오. 다음 DB2 명령을 실행하십시오.

```
db2set DB2_EVALUNCOMMITTED=ON
db2set DB2_SKIPDELETED=ON
db2set DB2_SKIPINSERTED=ON
```

이러한 설정의 영향에 대한 자세한 정보는 DB2 문서를 참조하십시오.

2. DB2를 다시 시작하십시오. 매개변수를 적용하려면 이 단계를 수행해야 합니다. 먼저 데이터베이스에서 실행 중인 세션이 없는지 확인하십시오.

```
db2stop force
db2start
```

## DB2 클라이언트 드라이버

제품을 설치하기 전에 호스트에 IBM DB2 데이터베이스 클라이언트 드라이버를 설치해야 합니다. UNIX 또는 Linux에서는 32비트 드라이버를 사용하십시오.

## 프로시저

1. DB2 Data Server Client를 설치하여 드라이버를 제공합니다.
2. DB2 클라이언트를 설치한 후 컴퓨터를 다시 시작하십시오. 그렇지 않으면 제품 설치에 실패합니다.

## 설치 중에 필요한 DB2 정보

### 이 태스크 정보

설치하는 동안 데이터베이스 구성 패널에 다음 정보를 제공해야 합니다.

#### 데이터베이스 구성

- 데이터베이스 호스트: DB2가 설치되는 호스트입니다.
- 데이터베이스 포트: Build Forge에서는 이 필드에서 DB2에 대한 기본 포트인 50000을 사용합니다. 다른 포트를 사용하는 경우 포트 번호를 입력하십시오.
- 데이터베이스 이름: 사용할 Build Forge의 데이터베이스 이름입니다. 이전 설정 단계에서 이 데이터베이스를 작성했습니다.
- 데이터베이스 스키마 이름: 사용할 Build Forge의 스키마 이름입니다.
- 데이터베이스 사용자 이름: 데이터베이스에 액세스할 때 사용할 Build Forge의 사용자 이름입니다. 이전 설정 단계에서 이 사용자를 작성했습니다.
- 비밀번호: 데이터베이스 사용자 이름에 대한 비밀번호입니다.

#### 데이터베이스 구성 테스트

- **DB2 클라이언트 라이브러리의 경로** - DB2 클라이언트 라이브러리가 있는 디렉토리입니다.

**중요사항:** UNIX 또는 Linux에서 Build Forge를 설치하는 경우 이 디렉토리는 32 비트 클라이언트 드라이버 라이브러리가 포함되어 있는 디렉토리여야 합니다.

- **JDBC 드라이버 위치** - JDBC 드라이버가 있는 디렉토리입니다. 이 드라이버는 Apache Tomcat에서 데이터베이스에 액세스할 때 사용됩니다. 일반적인 위치는 다음과 같습니다.

- Windows: <db2install>/IBM/SQLLIB/java
- UNIX 또는 Linux: 시스템에 대한 문서를 참조하십시오.

다음 정보가 표시됩니다.

- **필수 드라이버 JAR 파일** - 필수 드라이버 JAR 파일이 표시됩니다. DB2의 경우 해당 파일은 db2jcc.jar 및 db2jcc\_license\_cu.jar입니다.
- **필수 JDBC 드라이버 클래스** - 필수 JDBC 드라이버 클래스가 표시됩니다. DB2의 경우 이 클래스는 com.ibm.db2.jcc.DB2Driver입니다.



## Microsoft SQL Server 설정

다음 프로시저를 사용하여 Windows® 기반 관리 콘솔에서 Microsoft SQL Server 데이터베이스에 대한 액세스를 설치하고 구성합니다.

### Microsoft SQL Server용 데이터베이스 오브젝트

#### 이 태스크 정보

다음 단계에서는 데이터베이스 소유자로 사용할 사용자 및 사용할 Build Forge용 데이터베이스를 작성합니다. 필요한 경우 Build Forge에서 사용할 스키마를 작성할 수도 있습니다.

#### 프로시저

1. 데이터베이스 소유자로 사용할 사용자를 작성하십시오. Build Forge는 데이터베이스에 로그인하는 데 이 사용자 이름을 사용합니다. 다른 이름을 사용해야 하는 경우가 아니라면 **build**를 사용하십시오. 사용자는 전체 권한이 있어야 합니다.
  - a. SQL Server 관리 스튜디오를 여십시오.
  - b. 오브젝트 탐색기(왼쪽 패널)에서 데이터베이스 서버를 여십시오.
  - c. 보안 폴더를 마우스 오른쪽 단추로 클릭하고 **새로 작성 > 로그인**을 선택하십시오.
  - d. 로그인 - 새로 작성 대화 상자에서 로그인 이름을 지정하고 다음과 같이 옵션을 선택하십시오. 중요: 다음 로그인에서 사용자가 비밀번호를 변경해야 함을 선택 취소하십시오.
    - **SQL Server** 인증을 선택하고 비밀번호를 제공하십시오.
    - 비밀번호 만기 강제 실행을 선택 취소하십시오.
    - 다음 로그인에서 사용자가 비밀번호를 변경해야 함을 선택 취소하십시오.
2. 데이터베이스를 작성하십시오. 혼합 모드 인증을 사용해야 합니다. 다음 대화식 및 스크립트 예제에서는 사용자가 DBO의 기본 스키마를 사용한다고 가정합니다.
  - a. SQL Server 관리 스튜디오를 여십시오.
  - b. 오브젝트 탐색기(왼쪽 패널)에서 데이터베이스 서버를 여십시오.
  - c. 데이터베이스 폴더를 마우스 오른쪽 단추로 클릭하고 **새 데이터베이스**를 선택하십시오.
  - d. 새 데이터베이스 대화 상자에서 데이터베이스에 대한 매개변수를 지정하십시오.
    - 데이터베이스 이름을 지정하십시오. 다른 이름을 사용해야 하는 경우가 아니라면 **build**를 사용하십시오. 이름은 대소문자를 구분합니다. 데이터 및 로그 파일의 이름은 데이터베이스 파일 상자에서 자동으로 업데이트됩니다.
    - 데이터베이스 소유자를 지정하십시오.
      - 필드의 오른쪽에 있는 [...] 제어를 클릭하십시오.

- 데이터베이스 소유자 선택 대화 상자에서 **찾아보기**를 클릭하십시오.
- 작성한 사용자의 이름을 선택한 다음 **확인**을 클릭하십시오.
- 데이터베이스 소유자 선택 대화 상자에서 **확인**을 클릭하십시오.
- 데이터베이스 파일 매개변수를 지정하십시오. 데이터베이스 파일 테이블에서 다음을 수행하십시오.
  - 두 파일 모두에 대해 초기 크기를 500(MB)으로 설정하십시오.
  - 두 파일 모두에 대해 자동 증가를 설정하십시오. 자동 증가 열에서 **[..]** 제어를 클릭하여 대화 상자를 여십시오. 자동 증가 사용 상자를 선택하고 증가를 500MB로 설정한 다음 무제한 증가를 선택하고 **확인**을 클릭하십시오.

또는 다음 스크립트를 사용하여 데이터베이스를 작성할 수 있습니다.

```
CREATE DATABASE [build] ON PRIMARY
( NAME = N'build', FILENAME = N'C:\Program Files\Microsoft SQL
Server\MSSQL.2\MSSQL\DATA\build.mdf' , SIZE = 2048KB , FILEGROWTH = 1024KB )
LOG ON
( NAME = N'build_log', FILENAME = N'C:\Program Files\Microsoft SQL
Server\MSSQL.2\MSSQL\DATA\build_log.ldf' , SIZE = 1024KB , FILEGROWTH = 10%)
GO
EXEC dbo.sp_dbcmptlevel @dbname=N'build', @new_cmptlevel=90
GO
IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))
begin
EXEC [build].[dbo].[sp_fulltext_database] @action = 'disable'
end
GO
ALTER DATABASE [build] SET ANSI_NULL_DEFAULT OFF
GO
ALTER DATABASE [build] SET ANSI_NULLS OFF
GO
ALTER DATABASE [build] SET ANSI_PADDING OFF
GO
ALTER DATABASE [build] SET ANSI_WARNINGS OFF
GO
ALTER DATABASE [build] SET ARITHABORT OFF
GO
ALTER DATABASE [build] SET AUTO_CLOSE OFF
GO
ALTER DATABASE [build] SET AUTO_CREATE_STATISTICS ON
GO
ALTER DATABASE [build] SET AUTO_SHRINK OFF
GO
ALTER DATABASE [build] SET AUTO_UPDATE_STATISTICS ON
GO
ALTER DATABASE [build] SET CURSOR_CLOSE_ON_COMMIT ON
GO
ALTER DATABASE [build] SET CURSOR_DEFAULT GLOBAL
GO
ALTER DATABASE [build] SET CONCAT_NULL_YIELDS_NULL OFF
GO
ALTER DATABASE [build] SET NUMERIC_ROUNDABORT OFF
GO
ALTER DATABASE [build] SET QUOTED_IDENTIFIER OFF
GO
```

```

ALTER DATABASE [build] SET READ_COMMITTED_SNAPSHOT ON
GO
ALTER DATABASE [build] SET RECURSIVE_TRIGGERS OFF
GO
ALTER DATABASE [build] SET AUTO_UPDATE_STATISTICS_ASYNC OFF
GO
ALTER DATABASE [build] SET DATE_CORRELATION_OPTIMIZATION OFF
GO
ALTER DATABASE [build] SET PARAMETERIZATION SIMPLE
GO
ALTER DATABASE [build] SET READ_WRITE
GO
ALTER DATABASE [build] SET RECOVERY FULL
GO
ALTER DATABASE [build] SET MULTI_USER
GO
ALTER DATABASE [build] SET PAGE_VERIFY CHECKSUM
GO
USE [build]
GO
IF NOT EXISTS (SELECT name FROM sys.filegroups WHERE is_default=1 AND name = N'PRIMARY') \
    ALTER DATABASE [build] MODIFY FILEGROUP [PRIMARY] DEFAULT
GO

```

**참고:** READ\_COMMITTED\_SNAPSHOT 속성은 ON으로 설정되어야 합니다.

READ\_COMMITTED\_SNAPSHOT을 테스트하려면 다음 SQL문을 실행하십시오.

```
SELECT is_read_committed_snapshot_on FROM sys.databases WHERE name='build'
```

build를 사용하지 않은 경우 실제 데이터베이스 이름으로 대체하십시오. 이 명령문은 "1"을 리턴해야 합니다.

명령문이 "0"을 리턴하는 경우 다음 SQL문을 실행하십시오.

```
ALTER DATABASE [build] SET READ_COMMITTED_SNAPSHOT ON
```

### 3. 사용자의 기본 데이터베이스를 설정하십시오.

- SQL Server 관리 스튜디오를 여십시오.
- 오브젝트 탐색기(왼쪽 패널)에서 데이터베이스 서버를 여십시오.
- 오브젝트 탐색기에서 보안 > 로그인을 여십시오.
- 작성한 사용자를 마우스 오른쪽 단추로 클릭하고 특성을 선택하십시오.
- 일반 페이지에서 기본 데이터베이스를 선택하십시오. 사용자가 작성한 데이터베이스를 선택하십시오.
- 확인을 클릭하십시오.

- 필요한 경우 스키마를 작성합니다. 기본 스키마 **DBO**를 사용할 수 있는 경우 스키마를 작성하지 않아도 됩니다. 기본값을 사용할 것을 권장합니다. 다른 스키마를 사용해야 할 경우 Build Forge 데이터베이스에서 작성하고 소유자를 작성한 데이터베이스 사용자로 지정하십시오. 다음 예제에서는 bf\_schema 스키마를 작성하고 build를 소유자로 설정합니다.

```
USE build;
GO
CREATE SCHEMA bf_schema AUTHORIZATION build;
GO
```

## Microsoft SQL Server에 TCP/IP 설정

### 이 태스크 정보

TCP/IP를 관리 콘솔과 함께 사용하려면 Microsoft SQL Server에서 이를 사용할 수 있도록 설정해야 합니다.

MS SQL Server에서 TCP/IP를 사용하도록 설정하려면 다음을 수행하십시오.

### 프로시저

1. MSSQLServer 구성 관리자를 여십시오.
2. **SQL Server** 네트워크 구성 아래에서 **MSSQLSERVER**용 프로토콜을 클릭하십시오.
3. **TCP/IP**를 마우스 오른쪽 단추로 클릭한 후 메뉴에서 **사용**을 선택하십시오.

## Microsoft SQL Server 클라이언트 및 JDBC 드라이버 설치

### 이 태스크 정보

관리 콘솔 호스트에 Microsoft SQL Server 클라이언트 및 JDBC 드라이버를 설치해야 합니다. 관리 콘솔에서 이를 사용하여 데이터베이스에 액세스합니다.

### 프로시저

1. SQL 기본 클라이언트를 설치하십시오. 버전 2005.90.4035.00이 필요합니다. 이는 Microsoft SQL Server의 서비스 팩 3에 포함되어 있습니다.
2. JDBC 드라이버를 설치하십시오.
  - 버전 2005: JDBC 라이브러리 버전 1.2가 필요합니다. 다음 위치에 있는 Microsoft에서 다운로드하십시오.

<http://www.microsoft.com/downloads/details.aspx?FamilyId=C47053EB-3B64-4794-950D-81E1EC91C1BA&displaylang=en>

Microsoft 설치 프로세스를 완료한 후 JAR의 위치는 다음과 같습니다.

/sqljdbc\_1.2/enu/sqljdbc.jar

- 버전 2008: JDBC 라이브러리 버전 3.0이 필요합니다.

<http://www.microsoft.com/downloads/en/details.aspx?FamilyID=%20a737000d-68d0-4531-b65d-da0f2a735707&displaylang=en>

## Microsoft SQL Server용 ODBC 데이터 원본 설정

### 이 태스크 정보

이 단계에서는 Windows 기반 관리 콘솔이 Microsoft SQL Server에서 작성한 데이터베이스에 액세스할 때 사용하는 ODBC 데이터 원본을 작성합니다. 실행하려는

Windows 시스템이 32비트이든 64비트이든 관계없이 32비트 ODBC 데이터 원본을 작성해야 합니다.

## 프로시저

1. 데이터 원본 관리자를 시작하십시오.

- 32비트 Windows 시스템: Windows 시작 메뉴에서 **설정 > 제어판 > 관리 도구 > 데이터 원본**을 선택하십시오.
- 64비트 Windows 시스템: C:\Windows\SysWOW64\odcbad32.exe 파일을 두 번 클릭하십시오. 이렇게 하면 데이터 원본 관리자 대화 상자가 열리고 64비트 시스템에서 필요한 32비트 ODBC 데이터 원본을 작성할 수 있습니다.

**ODBC 데이터 원본 관리자** 대화 상자가 표시됩니다.

2. 시스템 **DSN** 탭에서 **추가**를 클릭하십시오. 드라이버 목록이 표시됩니다.

3. 드라이버 목록에서 **SQL Server**를 선택한 후, **마침**을 클릭하십시오. 표시되는 **ODBC 설정** 대화 상자에서 다음 정보를 입력하십시오.

- **데이터 소스 이름**: 이 데이터 소스의 이름은 데이터베이스 이름과 동일해야 하며 데이터베이스와 연관된 스키마 이름과 달라야 합니다.
- **설명**: 이 데이터 원본에 대한 설명
- **서버 이름**: MS SQL Server 데이터베이스가 설치된 호스트의 호스트 이름
- **데이터베이스**: 위에서 작성한 데이터베이스 이름

**확인**을 클릭하여 **ODBC** 설정을 닫은 후, **확인**을 클릭하여 **ODBC 데이터 원본 관리자**를 닫으십시오. 데이터 원본이 작성됩니다.

## 결과

다음 정보를 적어두십시오. 이 정보는 관리 콘솔을 설치할 때 설치 프로그램에서 요청합니다.

- **데이터 원본 이름**: ODBC 데이터 원본 작성 시 지정한 이름
- **데이터 원본 유형**: SQL Server
- **사용자 이름**: 데이터베이스에 작성한 사용자 이름(예: **build**)
- **비밀번호**: 사용자 이름의 비밀번호(예: **build**)

**중요사항**: 데이터베이스 이름과 데이터 원본 이름으로 동일한 값을 사용하십시오. JDBC 드라이버의 제한사항으로 인해 이 제한조건이 요구됩니다. 값이 다르면 빠른 보고서 보고 기능과 Java 및 Perl용 서비스-계층 API가 작동되지 않습니다.

## 설치 중에 필요한 Microsoft SQL Server 정보

### 이 태스크 정보

설치하는 동안 데이터베이스 구성 패널에 다음 정보를 제공해야 합니다.

#### 데이터베이스 구성

- **데이터베이스 호스트:** SQL Server가 설치되는 호스트입니다.
- **데이터베이스 포트:** Build Forge에서는 이 필드에서 SQL Server에 대한 기본 포트인 1521을 사용합니다. 다른 포트를 사용하는 경우 포트 번호를 입력하십시오.
- **데이터베이스 이름:** 사용할 Build Forge의 데이터베이스 이름입니다. 이전 설정 단계에서 이 데이터베이스를 작성했습니다.
- **데이터베이스 스키마 이름:** 사용할 Build Forge 스키마 이름이며, 기본값인 DBO(권장)이거나 사용자가 이전 설정 단계에서 작성한 것입니다.
- **ODBC 데이터 소스 이름:** ODBC 데이터 원본의 이름입니다.
- **데이터베이스 사용자 이름:** 데이터베이스에 액세스할 때 사용할 Build Forge의 사용자 이름입니다. 이전 설정 단계에서 이 사용자를 작성했습니다.
- **비밀번호:** 데이터베이스 사용자 이름에 대한 비밀번호입니다.

#### 데이터베이스 구성 테스트

- **SQL Server 클라이언트 라이브러리** - SQL Server 클라이언트 라이브러리가 있는 디렉토리입니다.

**중요사항:** Microsoft SQL Server는 Windows 플랫폼에서만 지원됩니다.

- **JDBC 드라이버 위치** - JDBC 드라이버가 있는 디렉토리입니다. 이 드라이버는 Apache Tomcat에서 데이터베이스에 액세스할 때 사용됩니다.
- **2005 버전:** 2005 버전에서 제공하는 드라이버를 사용하십시오.
- **2008 버전:** 버전 3.0 드라이버를 사용하십시오.

다음 정보가 표시됩니다.

- **필수 드라이버 JAR 파일** - 필수 드라이버 JAR 파일이 표시됩니다. SQL Server의 경우 이 파일은 /sqljdbc\_1.2/enu/sqljdbc.jar입니다.
- **필수 JDBC 드라이버 클래스** - 필수 JDBC 드라이버 클래스가 표시됩니다. SQL Server의 경우 이 클래스는 com.mysql.jdbc.Driver입니다.

## MySQL 설정

이 프로시저를 사용하여 MySQL 지원을 설치하고 구성하십시오.

### 시작하기 전에

다음 항목을 설치하고 구성하십시오. 다음 절의 지시사항을 사용하십시오.

## MySQL에 대한 Red Hat Linux 4 요구사항

### 시작하기 전에

Rational Build Forge를 구성하기 전에 MySQL on Red Hat Linux 4의 추가 설정이 필요합니다.

### 프로시저

- ncurses-libs.i686을 설치하십시오. 다음 단계에서 파일을 설치하기 위해서는 이 파일이 꼭 있어야 합니다.
- 호환성 공유 라이브러리를 설치하십시오. 패키지 또는 RPM(둘 다는 아님)을 설치해야 합니다.

```
MySQL-client-community-5.0.51a-0.rhel5.i386
MySQL-shared-compat-5.0.51a-0.rhel5.i386
```

- 시작 스크립트의 이름을 바꾸십시오. 시작 프로세스에서 MySQL이 기본적으로 로드되는 것보다 더 빨리 로드되도록 하려면 시작 스크립트의 이름을 바꾸어야 합니다. 다음 예제는 MySQL을 위로 충분히 이동시킵니다.

```
mv /etc/rc3.d/S99mysql /etc/rc3.d/S50mysql
```

## MySQL의 데이터베이스 오브젝트

### 프로시저

1. **build**라는 비어 있는 데이터베이스를 작성하십시오.
2. 데이터베이스와 연관된 사용자를 작성하십시오(사용자 이름 **build** 및 비밀번호 **build**).

### 결과

다음 명령을 사용하여 데이터베이스 **build**를 작성하고 비밀번호(『식별자』)가 **build**인 사용자 **build@localhost**를 작성할 수 있습니다.

```
mysql -u root
mysql> create database build;
mysql> grant all on build.* to build@localhost
-> identified by "build";
```

## MySQL 라이브러리 및 클라이언트 드라이버

### 시작하기 전에

Build Forge를 설치하기 전에 Build Forge 호스트에 MySQL 기본 클라이언트 드라이버를 설치해야 합니다. UNIX 또는 Linux에서는 32비트 드라이버를 사용하십시오.

MySQL 5.1을 사용하는 경우 MySQL 5.0 클라이언트를 사용해야 합니다.

Build Forge 호스트와 동일한 호스트에 MySQL을 설치할 경우 다음을 설치하십시오.



MySQL database server  
MySQL-client-community-5.0.51a-0.rhel5.i386 - client drivers  
MySQL-shared-compat-5.0.51a-0.rhel5.i386 - required compatibility libraries  
Connector/J JDBC driver

다른 호스트에 MySQL을 설치할 경우 Build Forge 호스트에 다음을 설치하십시오.

MySQL-client-community-5.0.51a-0.rhel5.i386 - client drivers  
MySQL-shared-compat-5.0.51a-0.rhel5.i386 - required compatibility libraries  
Connector/J JDBC driver

#### AIX 시스템의 경우에만 해당:

Connector/J 드라이버 및 5.0-51a 클라이언트 드라이버를 설치해야 합니다. 해당 드라이버는 AIX 버전 6.1과 호환 가능합니다. 해당 드라이버는 MySQL 웹 사이트의 아카이브 영역에서 찾을 수 있습니다.

설치 프로그램이 MySQL 클라이언트 공유 라이브러리를 리패키지하려고 합니다. 파일이 mysql.com에 게시될 때 파일의 패키징은 Build Forge에서 사용할 수 없습니다. 설치 프로그램을 실행하는 사용자가 클라이언트 드라이버 파일이 설치되는 디렉토리(대개 /opt/mysql)에 대한 쓰기 액세스 권한이 있어야 합니다. 설치 프로그램이 libmysqlclient.a 및 libmysqlclient.so.15 파일을 대체합니다. 설치 프로그램이 파일을 리패키지할 수 없는 경우 설치 중에 사용자에게 알리고 설치를 계속합니다. 이 경우에는 사용자가 Build Forge를 실행하기 전에 수동으로 파일을 리패키지해야 합니다.

**참고:** MySQL 클라이언트를 사용하는 다른 애플리케이션을 실행 중인 경우 파일을 별도의 디렉토리에 수동으로 리패키지할 수 있습니다.

파일을 수동으로 패키징하려면 다음을 수행하십시오.

1. MySQL이 설치된 디렉토리에서 새 디렉토리 libbf를 작성하십시오.

```
root@myaix:/opt/mysql/> mkdir libbf
```

2. lib/libmysqlclient.so.15 파일을 새 디렉토리로 복사하십시오.

```
root@myaix:/opt/mysql/> cp lib/libmysqlclient.so.15 libbf
```

3. 새 디렉토리로 변경하고 새 아카이브 파일을 빌드하십시오.

```
root@myaix:/opt/mysql/> cd libbf
```

```
root@myaix:/opt/mysql/libbf/> ar -q libmysqlclient.a libmysqlclient.so.15
```

설치 프로세스가 이미 실행되지 않은 경우 이제 이 디렉토리를 설치 프로세스에 대한 클라이언트 라이브러리 경로로 사용할 수 있습니다. 자동 또는 명령행 메소드 중 하나가 사용된 경우 이 경로를 사용하도록 \$BFROOT/rc/buildforge의 LIBPATH를 업데이트하십시오.



## MySQL 구성

### 프로시저

1. 최대 데이터베이스 연결 수를 200으로 늘리십시오. `<mysql-installdir>/my.ini` (Windows) 또는 `/etc/my.cnf` (UNIX/Linux)에서 `[mysqld]` 섹션을 다음과 같이 편집하십시오.

```
max_connections=200
```

값은 최소한 최대 콘솔 프로세스 수 및 실행 큐 크기 시스템 설정의 총계와 동일해야 합니다(관리 콘솔의 관리 → 시스템 페이지).

2. MySQL 인스턴스가 InnoDB 스토리지 엔진을 사용 중인지 확인하십시오. Rational Build Forge 데이터베이스는 외부 키 제한조건을 사용합니다. MySQL MyISAM 스토리지 엔진은 외부 키 제한조건을 검사하지 않습니다. MyISAM이 사용되는 경우, 특히 API를 통해 새 데이터베이스 오브젝트가 작성될 때 문제점이 발생할 수 있습니다.

## 설치 중에 필요한 MySQL 정보

### 이 태스크 정보

설치하는 동안 데이터베이스 구성 패널에 다음 정보를 제공해야 합니다.

#### 데이터베이스 구성

- 데이터베이스 호스트: MySQL이 설치되는 호스트입니다.
- 데이터베이스 포트: Build Forge에서는 이 필드에서 MySQL에 대한 기본 포트인 3306을 사용합니다. 다른 포트를 사용하는 경우 포트 번호를 입력하십시오.
- 데이터베이스 이름: 사용할 Build Forge의 데이터베이스 이름입니다. 이전 설정 단계에서 이 데이터베이스를 작성했습니다.
- 데이터베이스 사용자 이름: 데이터베이스에 액세스할 때 사용할 Build Forge의 사용자 이름입니다. 이전 설정 단계에서 이 사용자를 작성했습니다.
- 비밀번호: 데이터베이스 사용자 이름에 대한 비밀번호입니다.

#### 데이터베이스 구성 테스트

- MySQL 클라이언트 라이브러리의 경로 - MySQL 클라이언트 라이브러리가 있는 디렉토리입니다.

**중요사항:** UNIX 또는 Linux에서 Build Forge를 설치하는 경우 이 디렉토리는 32 비트 클라이언트 드라이버 라이브러리가 포함되어 있는 디렉토리여야 합니다.

- JDBC 드라이버 위치 - JDBC 드라이버가 있는 디렉토리입니다. 이 드라이버는 Apache Tomcat에서 데이터베이스에 액세스할 때 사용됩니다.

다음 정보가 표시됩니다.

- 필수 드라이버 **JAR** 파일 - 필수 드라이버 JAR 파일이 표시됩니다. MySQL의 경우 이 파일은 mysql-connector-java-5.\*-bin.jar입니다.
- 필수 **JDBC** 드라이버 클래스 - 필수 JDBC 드라이버 클래스가 표시됩니다. MySQL의 경우 이 클래스는 com.mysql.jdbc.Driver입니다.

## MySQL 문제점 해결

### 이 태스크 정보

이 태스크는 "Database extension 'mysqli' not found."라는 메시지와 함께 MySQL를 사용하는 콘솔이 실패할 경우에 대한 임시 해결책을 설명합니다. 이 문제는 다음 파일 사용 시 Red Hat Linux 버전 6.x에서 발생합니다.

/opt/buildforge/server/apache/lib/php/extensions/no-debug-non-zts-20090626

### 프로시저

1. 누락된 라이브러리 및 권한을 검사하십시오. ldd 명령을 사용하십시오. 예제 응답은 다음과 같습니다.

```
ldd: warning: you do not have execution permission for `./mysqli.so'
linux-gate.so.1 => (0x00413000)
libmysqlclient.so.15 => not found
.....
libssl.so.6 => not found
libcrypto.so.6 => not found
```

2. 다음 태스크 중 하나를 수행하여 libmysqlclient.so.15 라이브러리를 찾으십시오.

- libmysqlclient.so.15의 전체 경로를 LD\_LIBRARY\_PATH에 추가하십시오.
- libmysqlclient.so.15를 /usr/lib에 복사하십시오.

3. 링크를 이전 라이브러리에 추가하십시오. 링크를 이전 라이브러리에 수동으로 추가해야 합니다. 예를 들어 다음과 같습니다.

```
ln -s /usr/lib/libcrypto.so.1.0.0 /usr/lib/libcrypto.so.6
ln -s /usr/lib/libssl.so.1.0.0 /usr/lib/libssl.so.6
```

4. 콘솔을 다시 시작하십시오. 다시 시작해야 합니다.

## Oracle 설정

다음 프로시저를 사용하여 Oracle 데이터베이스에 대한 지원을 설정합니다.

### 시작하기 전에

다음 항목을 설치하거나 구성하십시오. 다음 절의 지시사항을 사용하십시오.

## Oracle 10에 대한 Red Hat Linux 6 요구사항 시작하기 전에

Rational Build Forge를 설치하기 전에 Red Hat Linux 6의 Oracle 10의 추가 설정이 필요합니다.

### 프로시저

호환성 공유 라이브러리를 설치하십시오. 다음 패키지를 설치해야 합니다.

- `compat-libs-5.2-1.i386.rpm`
- `compat-libstdc++-33-3.2.3-68.i686`

## Oracle의 데이터베이스 오브젝트 이 태스크 정보

Oracle 호스트에서 로컬 사용자를 작성하십시오: 사용자 이름 **build** 및 비밀번호 **build**

- CREATE SESSION 및 CREATE TABLE을 포함하여 적절한 권한을 추가하십시오.
- DEFAULT TABLESPACE에 적절한 QUOTA 크기를 추가하여 시스템이 데이터를 저장할 수 있는 충분한 공간을 제공하십시오.

```
create user build
  identified by password
  default tablespace users
  quota unlimited on users;
```

```
grant create session, create table
to build;
```

**중요사항:** 설치 중에 데이터베이스 이름과 Oracle SID로 동일한 값이 사용됩니다. JDBC 드라이버의 제한사항으로 인해 이 제한조건이 요구됩니다. 값이 다르면 빠른 보고서 보고 기능과 Java 및 Perl용 서비스-계층 API가 작동되지 않습니다.

## Oracle에 권장되는 매개변수 조정 이 태스크 정보

Build Forge를 올바르게 실행하려면 일부 Oracle 매개변수를 기본값에서 다른 값으로 변경해야 합니다.

**참고:** Build Forge를 설치하고 실행한 후 이러한 매개변수를 변경하는 경우 변경하기 전에 Build Forge를 중지하십시오. 데이터베이스 서버를 다시 시작한 후 Build Forge를 다시 시작하십시오.

### 프로시저

1. 조정 매개변수를 설정하십시오. 다음 명령을 실행하십시오.

```
ALTER SYSTEM SET open_cursors=1000 SCOPE=BOTH
ALTER SYSTEM SET processes=500 SCOPE=BOTH
```

**참고:** ALTER SYSTEM SET processes=500 SCOPE=BOTH를 실행할 때 "SQL Error: ORA-02095: specified initialization parameter cannot be modified" 메시지가 표시되는 경우 대신 ALTER SYSTEM SET processes=500 SCOPE=SPFile을 사용하십시오.

이 설정의 영향에 대한 자세한 정보는 Oracle 문서를 참조하십시오.

2. 데이터베이스 서버를 다시 시작하십시오. 매개변수를 적용하려면 이 단계를 수행해야 합니다. 먼저 데이터베이스에서 실행 중인 세션이 없는지 확인하십시오.

## 설치 중에 필요한 Oracle 정보

### 이 태스크 정보

설치하는 동안 데이터베이스 구성 패널에 다음 정보를 제공해야 합니다.

#### 데이터베이스 구성

- **데이터베이스 호스트:** Oracle이 설치되는 호스트입니다.
- **데이터베이스 포트:** Build Forge에서는 이 필드에서 Oracle에 대한 기본 포트인 1521을 사용합니다. 다른 포트를 사용하는 경우 포트 번호를 입력하십시오.
- **데이터베이스 이름:** 사용할 Build Forge의 데이터베이스 이름입니다. 이전 설정 단계에서 이 데이터베이스를 작성했습니다.
- **데이터베이스 사용자 이름:** 데이터베이스에 액세스할 때 사용할 Build Forge의 사용자 이름입니다. 이전 설정 단계에서 이 사용자를 작성했습니다.
- **비밀번호:** 데이터베이스 사용자 이름에 대한 비밀번호입니다.

#### 데이터베이스 구성 테스트

- **Oracle 클라이언트 라이브러리의 경로** - Oracle 클라이언트 라이브러리가 있는 디렉토리입니다.

**중요사항:** UNIX 또는 Linux에서 Build Forge를 설치하는 경우 이 디렉토리는 32 비트 클라이언트 드라이버 라이브러리가 포함되어 있는 디렉토리여야 합니다.

- **ORACLE\_HOME 환경 변수** - Oracle이 설치되는 디렉토리입니다.
- **tnsnames.ora 파일의 경로(TNS\_ADMIN)** - tnsnames.ora 파일이 포함되어 있는 디렉토리입니다. tnsnames.ora 파일과 관련하여 전체 액세스 권한이 설정되었는지 확인하십시오.
- **JDBC 드라이버 위치** - JDBC 드라이버가 있는 디렉토리입니다. 이 드라이버는 Apache Tomcat에서 데이터베이스에 액세스할 때 사용됩니다.

다음 정보가 표시됩니다.

- 필수 드라이버 **JAR** 파일 - 필수 드라이버 JAR 파일이 표시됩니다. Oracle의 경우 이 파일은 ojdbc14.jar입니다.
- 필수 **JDBC** 드라이버 클래스 - 필수 JDBC 드라이버 클래스가 표시됩니다. Oracle의 경우 이 클래스는 oracle.jdbc.driver.OracleDriver입니다.

## Oracle 클라이언트 구성

### 이 태스크 정보

Oracle용 클라이언트를 설치 및 구성하려면 다음을 수행하십시오.

#### 프로시저

1. 관리 콘솔 호스트에 Oracle Instant Client를 설치하십시오. 32비트 또는 64비트 플랫폼에서 실행 중인지 여부에 관계 없이 32비트 버전을 설치해야 합니다. 운영 체제에 버전 10.2 또는 11.2의 인스턴트 클라이언트 패키지 - 기본만 사용하십시오.
  - Oracle 10.2의 경우 버전 10.2 클라이언트를 사용하십시오.
  - Oracle 11g의 경우 버전 11.2 클라이언트를 사용하십시오.

Oracle에서 클라이언트를 다운로드하십시오(<http://www.oracle.com/technology/software/tech/oci/instantclient/index.html>).

2. 관리 콘솔 호스트에서 환경을 설정하십시오. 다음과 같은 여러 환경 변수를 설정해야 합니다.
  - LD\_LIBRARY\_PATH: 클라이언트 설치 디렉토리를 포함하도록 설정하십시오.

참고: Installation Manager의 데이터베이스 구성 페이지에서 이 값을 지정할 수 있습니다.

- NLS\_LANG: 자국어 지원에 적절한 값으로 설정하십시오.
  - UNIX 및 Linux 시스템: 값이 AL32UTF8을 포함해야 합니다. 예: AMERICAN\_AMERICA.AL32UTF8.
  - Windows 시스템: 값이 Windows 레지스트리의 활성 코드 페이지 설정에 대응하는 문자 맵 스펙을 포함해야 합니다. 56 페이지의 『Windows의 Oracle 클라이언트 예제 구성』의 내용을 참조하십시오.

설명대로 *NLS\_LANG*을 명시적으로 설정해야 합니다. 클라이언트 설치 시 설정된 기본 문자 세트는 *Build Forge*에서 사용하기에 올바르지 않습니다.

- ORACLE\_HOME: Oracle 클라이언트 설치 디렉토리 경로로 설정하십시오.

참고: Installation Manager의 데이터베이스 구성 페이지에서 이 값을 지정할 수 있습니다.

- ORA\_NLS10: 서버에서 문자 세트 데이터가 있는 경로를 설정하십시오.

- PATH: 클라이언트 설치 디렉토리를 포함하도록 설정하십시오.
- TNS\_ADMIN: 서버에서 tnsnames.ora 파일이 있는 경로로 설정하십시오. tnsnames.ora 파일과 관련하여 전체 액세스 권한이 설정되었는지 확인하십시오.

참고: Installation Manager의 데이터베이스 구성 페이지에서 이 값을 지정할 수 있습니다.

Oracle 서버에서 현재 언어 설정을 확인하려면 Oracle에 로그인하여 다음 명령을 실행하십시오.

```
SQL> host echo $NLS_LANG
```

## 다음에 수행할 작업

**중요사항:** 설치 중에 tnsnames.ora는 데이터베이스 이름 및 Oracle SID의 값과 동일한 값을 사용하도록 설정됩니다. JDBC 드라이버의 제한사항으로 인해 이 제한조건이 요구됩니다. 값이 다르면 빠른 보고서 보고 기능과 Java 및 Perl용 서비스-계층 API가 작동되지 않습니다.

## Windows의 Oracle 클라이언트 예제 구성:

### 이 태스크 정보

예제 환경:

- Instant Client - C:\instantclient\_11\_2에서 기본이며, 미국 영어를 사용합니다.
- Oracle 11.2가 국제적 데이터를 지원하기 위해 Windows 시스템의 C:\oracle\product\11.2.0\db\_1에 설치됩니다.

클라이언트와 Build Forge가 설치된 시스템의 변수 설정:

- LD\_LIBRARY\_PATH에 C:\instantclient\_11\_2\ 포함
- NLS\_LANG=AMERICAN\_AMERICA.WE8MSWIN1252
- ORACLE\_HOME=C:\instantclient\_11\_2\
- ORA\_NLS10=C:\oracle\ocommon\nls\admin\data
- PATH에 C:\instantclient\_11\_2\포함
- TNS\_ADMIN=C:\oracle\product\11.2\db\_1\network\admin

## Windows 시스템의 NLS\_LANG 정보

Windows의 Oracle 클라이언트는 Windows의 설정을 사용하여 로컬 문자 매핑을 수행합니다.

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Nls\CodePage\ACP
```

클라이언트가 서버로 데이터를 전송할 때 문자가 서버의 문자 매핑으로 전송됩니다.

위의 예제 NLS\_LANG 설정은 US Windows의 기본값인 1252의 ACP 설정에 해당합니다. 기타 언어의 경우 Oracle InstantClient 문서를 참조하여 올바른 언어, 로케일 및 NLX\_LANG의 문자 맵 파트를 가져오십시오. 반드시 Windows 시스템의 실제 ACP 설정을 확인하고 설정의 문자 맵핑 파트가 그와 일치하게 하십시오.

기타 언어에 대한 문자 맵은 다음과 같습니다. 올바른 언어 및 로케일과 조합으로 사용하여 NLS\_LANG 설정을 작성하십시오.

```
1250 EE8MSWIN1250
1251 CL8MSWIN1251
1252 WE8MSWIN1252
1253 EL8MSWIN1253
1254 TR8MSWIN1254
1255 IW8MSWIN1255
1256 AR8MSWIN1256
1257 BLT8MSWIN1257
1258 VN8MSWIN1258
874  TH8TISASCII
932  JA16SJIS
936  ZHS16GBK
949  KO16MSWIN949
950  ZHT16MSWIN950
```

**참고:** 문자 맵핑이 올바르게 설정되지 않은 경우 경고가 수신되지 않으며 처음에는 동작이나 성능에서 명백한 차이가 없습니다. 클라이언트가 메모리에서 문자 변환을 수행하려고 시도합니다. 메모리 부족이 발생하고 변환을 수행하기 위해 스왑해야 할 때 클라이언트와 서버 간의 통신 성능이 급격히 저하됩니다.

## UNIX 또는 Linux의 Oracle 클라이언트 예제 구성:

### 이 태스크 정보

#### 예제 환경:

- Instant Client - /usr/local/instantclient\_11\_2에서 기본이며, 미국 영어를 사용합니다.
- 국제적 데이터를 지원하기 위해 UNIX 시스템의 /usr/local/oracle/product/11.2.0/db\_1에 Oracle 11.2가 설치되었습니다.

#### 클라이언트와 Build Forge가 설치된 시스템의 변수 설정:

- LD\_LIBRARY\_PATH(Solaris, Linux) 또는 LIBPATH(AIX)에 /usr/local/instantclient\_11\_2 포함
- NLS\_LANG=AMERICAN\_AMERICA.AL32UTF8
- ORACLE\_HOME=/usr/local/instantclient\_11\_2
- ORA\_NLS10=/usr/local/oracle/ocommon/nls/admin/data
- PATH에 /usr/local/instantclient\_11\_2 포함
- TNS\_ADMIN=/usr/local/oracle/product/11.2/db\_1/network/admin



## Oracle Instant Client 11.2를 사용하는 UNIX 및 Linux 시스템

Build Forge 애플리케이션은 버전 10.2 클라이언트의 파트인 libclntsh.so.10.1 라이브러리에 종속성을 갖습니다. 설치 중에 설치 프로그램이 ORACLE\_HOME에서 이 파일을 점검합니다. 해당 파일이 없는 경우 설치 프로그램은 버전 11.2 클라이언트를 사용 중이라고 가정하고 libclntsh.so.10.1에서 대응하는 버전 11.2 라이브러리인 libclntsh.so.11.1로의 기호 링크를 작성하려고 합니다.

```
ln -s libclntsh.so.11.1 libclntsh.so.10.1
```

**중요사항:** Build Forge 설치 프로그램을 실행하는 사용자는 ORACLE\_HOME 디렉토리에 대한 쓰기 액세스 권한이 있어야 합니다. 이것이 사이트에서 문제가 되는 경우 Build Forge를 설치하기 전에 권한있는 사용자가 수동으로 링크를 작성하도록 하십시오. 일부 사이트에서는 클라이언트가 읽기 전용으로 마운트되는 파일 시스템에 설치될 수 있습니다(Solaris 시스템의 일반적 관행임).

## Oracle RAC을 사용하여 장애 복구 구현

이 프로시저를 사용하여 Oracle RAC을 사용할 때 콘솔 장애 복구를 사용 가능하게 하십시오.

### 시작하기 전에

Oracle RAC의 기본 데이터베이스 설정은 기타 Oracle 데이터베이스의 설정과 동일합니다. 해당 설정을 사용하여 콘솔을 설치하십시오. 콘솔이 설치된 후 해당 프로시저를 사용하여 장애 복구를 구현하십시오.

### 이 태스크 정보

Build Forge로 작업하기 위해 Oracle RAC 장애 복구를 사용 가능하게 하려면 추가 구성이 필요합니다. 사용 가능한 두 가지 방법이 있습니다.

- **SCAN 방법:** buildforge.conf를 수정하십시오. SCAN(single client access name)은 Oracle 11g의 기능입니다. 사용되는 경우, 클러스터의 모든 노드에 장애 복구가 적용됩니다. 최소 세 개의 노드가 사용 가능해야 하며 라운드 로빈 기초의 장애 복구로 설정해야 합니다.
- **명시적 연결 문자열 방법:** Oracle Instant Client의 tnsnames.ora 파일을 수정하고 buildforge.conf를 수정하십시오. 이 방법에서 클러스터 및 원하는 각 클러스터 노드를 나타내는 서비스를 지정합니다. 최소 세 개의 노드가 사용 가능해야 하며 라운드 로빈 기초의 장애 복구로 설정해야 합니다. LOAD\_BALANCE 및 FAILOVER 매개변수를 설정해야 합니다.



서비스 이름과 사용할 노드 목록을 제공하기 위해 Oracle DBA가 필요할 수 있습니다. Oracle Instant Client에 대해 tnsnames.ora 파일을 구성할 수 있고 buildforge.conf에서 JDBC 연결 문자열을 설정할 수 있습니다.

두 방법은 클러스터를 전체로 나타내는 SERVICE\_NAME을 지정해야 합니다. 이는 비 RAC 구성과 다르며 여기서 SERVICE\_NAME은 일반적으로 SID와 동일합니다.

## 프로시저

- SCAN 방법

1. buildforge.conf를 편집하십시오. db\_url 매개변수를 buildforge.conf에 추가하고 JDBC 연결 문자열을 매개변수로 사용하십시오. 서비스는 Oracle(mysvc.mycorp.com)에서 설정해야 하며 다중 노드가 있어야 함을 참고하십시오. 정확성을 위해 행 바꾸기가 포함됩니다. buildforge.conf에서 행을 바꾸지 마십시오.

```
db_hostname myoracledbhost.mycompany.com
db_url jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)
      (HOST=myoracledbhost.mycompany.com) (PORT=1521))
      (CONNECT_DATA=(SERVICE_NAME=mysvc.mycorp.com)))
db_tcp_port 1521
db_database orclrac
db_username db_password db_type odbc
db_provider oracle
```

2. 모든 위치의 buildforge.conf를 업데이트하십시오.

- Windows

```
<bfinstall>
<bfinstall>/Apache/tomcat/webapps/rbf-services/WEB-INF/
classes/buildforge.conf
```

- UNIX 또는 Linux

```
<bfinstall>/Platform
<bfinstall>/server/tomcat/webapps/rbf-services/WEB-INF/
classes/buildforge.conf
```

- 명시적 연결 문자열 방법 이 예제에서 다음 이름이 사용됩니다.

- 서비스 이름: myorascv.mycorp.com
- 노드 이름: clusnode-1vip, clusnode-2vip, clusnode-3vip, clusnode-4vip

1. tnsnames.ora를 편집하십시오.

```
myorascv.mycorp.com=
(DESCRIPTION=
  (ADDRESS=(PROTOCOL=TCP) (HOST=clusnode-1vip) (PORT=1521))
  (ADDRESS=(PROTOCOL=TCP) (HOST=clusnode-2vip) (PORT=1521))
  (ADDRESS=(PROTOCOL=TCP) (HOST=clusnode-3vip) (PORT=1521))
  (ADDRESS=(PROTOCOL=TCP) (HOST=clusnode-4vip) (PORT=1521))
  (LOAD_BALANCE=yes) (FAILOVER=ON)
  (CONNECT_DATA=(SERVER=DEDICATED))
```

```
(SERVICE_NAME=myorasvc.mycorp.com)
(failover_mode=(type=select)
(method=basic)(retries=180)(delay=5)))
)
```

2. buildforge.conf를 편집하십시오. db\_url 매개변수를 JDBC 연결 문자열로 설정하십시오. 정확성을 위해 행 바꾸기가 포함됩니다. buildforge.conf에서 행을 바꾸지 마십시오.

```
db_url jdbc:oracle:thin:@(DESCRIPTION=(LOAD_BALANCE=on)
(AADDRESS=(PROTOCOL=TCP)(HOST=clusnode-1vip)(PORT=1521))
(AADDRESS=(PROTOCOL=TCP)(HOST=clusnode-2vip)(PORT=1521))
(AADDRESS=(PROTOCOL=TCP)(HOST=clusnode-3vip)(PORT=1521))
(AADDRESS=(PROTOCOL=TCP)(HOST=clusnode-4vip)(PORT=1521))
(CONNECT_DATA=(SERVICE_NAME=myorasvc.mycorp.com)))
```

3. 모든 위치의 buildforge.conf를 업데이트하십시오.

- Windows

```
<bfinstall>
<bfinstall>/Apache/tomcat/webapps/rbf-services/WEB-INF/
classes/buildforge.conf
```

- UNIX 또는 Linux

```
<bfinstall>/Platform
<bfinstall>/server/tomcat/webapps/rbf-services/WEB-INF/
classes/buildforge.conf
```

## 설치 중에 필요한 Oracle 정보

### 이 태스크 정보

설치하는 동안 데이터베이스 구성 패널에 다음 정보를 제공해야 합니다.

#### 데이터베이스 구성

- 데이터베이스 호스트: Oracle이 설치되는 호스트입니다.
- 데이터베이스 포트: Build Forge에서는 이 필드에서 Oracle에 대한 기본 포트인 1521을 사용합니다. 다른 포트를 사용하는 경우 포트 번호를 입력하십시오.
- 데이터베이스 이름: 사용할 Build Forge의 데이터베이스 이름입니다. 이전 설정 단계에서 이 데이터베이스를 작성했습니다.
- 데이터베이스 사용자 이름: 데이터베이스에 액세스할 때 사용할 Build Forge의 사용자 이름입니다. 이전 설정 단계에서 이 사용자를 작성했습니다.
- 비밀번호: 데이터베이스 사용자 이름에 대한 비밀번호입니다.

#### 데이터베이스 구성 테스트

- Oracle 클라이언트 라이브러리의 경로 - Oracle 클라이언트 라이브러리가 있는 디렉토리입니다.

**중요사항:** UNIX 또는 Linux에서 Build Forge를 설치하는 경우 이 디렉토리는 32 비트 클라이언트 드라이버 라이브러리가 포함되어 있는 디렉토리여야 합니다.

- **ORACLE\_HOME** 환경 변수 - Oracle이 설치되는 디렉토리입니다.
- **tnsnames.ora** 파일의 경로(**TNS\_ADMIN**) - tnsnames.ora 파일이 포함되어 있는 디렉토리입니다. tnsnames.ora 파일과 관련하여 전체 액세스 권한이 설정되었는지 확인하십시오.
- **JDBC** 드라이버 위치 - JDBC 드라이버가 있는 디렉토리입니다. 이 드라이버는 Apache Tomcat에서 데이터베이스에 액세스할 때 사용됩니다.

다음 정보가 표시됩니다.

- 필수 드라이버 **JAR** 파일 - 필수 드라이버 JAR 파일이 표시됩니다. Oracle의 경우 이 파일은 ojdbc14.jar입니다.
- 필수 **JDBC** 드라이버 클래스 - 필수 JDBC 드라이버 클래스가 표시됩니다. Oracle의 경우 이 클래스는 oracle.jdbc.driver.OracleDriver입니다.

---

## 보안 설정

설치하는 동안 보안을 설정할 방법에 대해 묻는 메시지가 표시됩니다.

- 키 저장소 비밀번호: 키 저장소의 비밀번호를 제공해야 합니다. 보안 로그인(신임 정보 암호화) 사용 및 HTTPS/SSL 사용을 위한 시작점으로 사용됩니다.
- 인증서: 개인 인증서를 설치하거나 이미 보유한 인증서를 가져올 수 있는 옵션을 사용할 수 있습니다.
- 보안 HTTP: HTTPS/SSL에 사용할 수 있는 Apache 서버를 설치할 것인지 묻는 메시지가 표시됩니다. 선택한 인증서가 사용됩니다. 기본값인 443이 아닌 포트 번호를 사용해야 하는 경우 포트 번호를 입력해야 합니다.

### 제공된 개인 인증서 사용

제공된 인증서에 다음과 같은 속성 세트가 들어 있습니다.

- 주제 DN: "CN=hostname", 여기서 *hostname*은 설치를 수행할 호스트의 완전한 이름입니다.
- 만기 기간: 10년(3650일로 표시됨). 이 값을 변경할 수 있습니다. 일반적으로 만기 기간은 1년 - 2년입니다. 만기 기간이 이 기간보다 길어지면 키 추측을 시도하는 보안 공격에 대한 취약성이 증가합니다.

제공된 인증서를 수정할 수 있습니다. 인증서를 수정하는 경우 다음 필드를 지정할 수 있습니다.

- 공통 이름(필수)

- 구/군/시
- 시/도
- 조직 이름
- 국가/지역 이름(필수)
- 주소

공통 이름 및 국가/지역 이름은 인증서를 작성하는 동안 지정되는 X500Principal 유형 주제 DN에 연결됩니다.

설치 프로그램에서 작성되는 키 저장소에 사용할 비밀번호에 대한 프롬프트가 표시됩니다. 이 비밀번호를 기록하십시오. HTTPS/SSL 설정을 완료하기 위해 필요합니다.

**중요사항:** 나중에 비밀번호를 변경할 수는 있지만 프로세스가 깁니다. 복잡도에 대한 로컬 요구사항을 충족하는 강력한 비밀번호를 사용하십시오.

## 고유 인증서 사용

인증서가 있는 경우 SSL을 사용하는 시스템의 모든 컴포넌트 및 연결에서 사용할 수 있도록 해당 인증서를 가져올 수 있습니다.

- Build Forge를 설치할 호스트에서 인증서를 사용할 수 있어야 합니다. 인증서를 임시 디렉토리에 복사합니다. 설치하는 동안 완전한 경로에 대한 프롬프트가 표시됩니다.
- 설치하는 동안 키 저장소 비밀번호를 지정해야 합니다.
- 인증서가 PKCS12 키 저장소 유형이어야 합니다. 인증서가 다른 유형인 경우 OpenSSL openssl 유틸리티 또는 JDK keytool 유틸리티를 사용하여 사본을 PKCS12로 변환합니다.
- 설치 프로그램에서 작성되는 키 저장소에 사용할 비밀번호에 대한 프롬프트가 표시됩니다. 이 비밀번호를 기록하십시오. HTTPS/SSL 설정을 완료하기 위해 필요합니다.

**중요사항:** 나중에 비밀번호를 변경할 수는 있지만 프로세스가 깁니다. 복잡도에 대한 로컬 요구사항을 충족하는 강력한 비밀번호를 사용하십시오.

---

## 제 7 장 관리 콘솔 설치

이 절에서는 Windows, UNIX 및 Linux 플랫폼에 관리 콘솔을 설치하는 방법에 대해 설명합니다. Installation Manager에서 지원되는 모든 설치 시나리오에 다음 프로시저를 사용합니다.

1. 35 페이지의 제 6 장 『설치 전 설정』에 설명된 대로 설치 전 설정을 수행하십시오. 태스크에는 다음이 포함됩니다.
  - 국제적 데이터 지원 설정(필수)
  - 데이터베이스 설정(필수). 일반적으로 이 단계는 데이터베이스 오브젝트 작성, 데이터베이스 클라이언트 설치 및 Installation Manager에서 설치 단계를 수행하는 동안 필요한 정보 수집에 사용됩니다.
  - 보안 설정(필요에 따라 선택적)
2. 런치패드를 시작하여 Installation Manager를 시작하십시오.

**참고:** 런치패드가 해당 런치패드를 실행하는 호스트에서 Installation Manager를 검색합니다. Installation Manager가 없는 경우 런치패드가 패키징된 Installation Manager를 실행하여 호스트에 Installation Manager를 설치합니다. 그런 다음 설치된 Installation Manager를 사용하여 Build Forge를 설치합니다. 원하는 경우 런치패드를 사용하지 않고 Installation Manager를 수동으로 설치할 수 있습니다.

3. Installation Manager에서 설치 단계를 수행하십시오.
4. 필수 사후 설치 검사를 수행하십시오.

다음 시나리오에 대한 설치 프로시저는 75 페이지의 제 8 장 『대체 설치 방법』의 내용을 참조하십시오.

- Build Forge에서 제공되고 설치되는 애플리케이션 대신 다음 필수 애플리케이션의 자체 설치 사용
  - Apache HTTP Server
  - PHP
  - Apache Tomcat
- IBM Installation Manager를 사용하여 제품 컴포넌트 자동 설치
- VMware에 설치
- System z의 SUSE Linux에 관리 콘솔 설치

---

## 런치패드를 사용하여 Installation Manager 시작

런치패드를 사용하여 Installation Manager를 시작하고 Build Forge를 설치합니다.

다음과 같은 방법으로 런치패드를 시작할 수 있습니다.

- 제품 DVD에서 런치패드를 시작합니다.
- 다운로드한 파일 패키지에서 런치패드를 시작합니다.

런치패드에서 호스트에 Installation Manager가 설치되어 있는지 감지합니다.

- Installation Manager가 있는 경우 런치패드에서 이를 시작합니다. Installation Manager에서 설치할 **Build Forge** 패키지를 선택합니다.

**중요사항:** Installation Manager 버전은 최소 버전 요구사항을 만족해야 합니다. 22 페이지의 『Installation Manager 요구사항』의 내용을 참조하십시오.

- Installation Manager가 없는 경우 런치패드에서 패키지된 Installation Manager를 사용합니다. 패키지된 Installation Manager에서 설치할 **Installation Manager** 및 **Build Forge** 패키지를 선택합니다. 패키지된 Installation Manager에서 Installation Manager를 설치하고 이를 시작하여 Build Forge를 설치합니다.

또한 호스트가 Windows 호스트인 경우 런치패드에서 로컬 호스트에 에이전트를 설치할 수 있습니다. Windows가 아닌 운영 체제에서는 런치패드를 사용하여 에이전트를 설치할 수 없습니다. 에이전트 설치 지시사항은 161 페이지의 제 10 장 『에이전트 설치』의 내용을 참조하십시오.

### 제품 DVD에서 런치패드 시작

다음 지시사항을 사용하여 제품 DVD에서 런치패드를 시작합니다.

운영 체제에 대한 지시사항을 선택합니다.

- UNIX 또는 Linux
  1. Build Forge를 설치할 호스트에 첫 번째 DVD를 삽입하십시오.
  2. 드라이브를 마운트하십시오.
  3. 드라이브의 루트 디렉토리에서 `launchpad.sh`를 실행하십시오.
  4. Installation Manager에서 설치할 패키지를 선택하십시오.
    - 호스트에 IBM Installation Manager가 있는 경우 런치패드에서 이를 시작합니다.

첫 번째 패키지 설치 페이지에서 **Build Forge** 패키지를 선택한 후 다음을 클릭하십시오.

- IBM Installation Manager가 없는 경우 Installation Manager를 설치하기 위해 패키징된 Installation Manager가 시작되고 이를 사용하여 Build Forge가 설치됩니다.

첫 번째 패키지 설치 페이지에서 **Installation Manager** 및 **Build Forge** 패키지를 선택한 후 다음을 클릭하십시오.

- Windows

1. Build Forge를 설치할 호스트에 첫 번째 DVD를 삽입하십시오.
2. 자동 실행을 사용하는 경우 런치패드가 자동으로 시작됩니다. 자동 실행을 사용할 수 없는 경우 드라이브의 루트 디렉토리에서 launchpad.exe를 실행하십시오.
3. Installation Manager에서 설치할 패키지를 선택하십시오.
  - 호스트에 IBM Installation Manager가 있는 경우 런치패드에서 이를 시작합니다.

첫 번째 패키지 설치 페이지에서 **Build Forge** 패키지를 선택한 후 다음을 클릭하십시오.

- IBM Installation Manager가 없는 경우 Installation Manager를 설치하기 위해 패키징된 Installation Manager가 시작되고 이를 사용하여 Build Forge가 설치됩니다.

첫 번째 패키지 설치 페이지에서 **Installation Manager** 및 **Build Forge** 패키지를 선택한 후 다음을 클릭하십시오.

## 다운로드한 패키지에서 런치패드 시작

다음 지시사항을 사용하여 설치 패키지를 다운로드하고 런치패드를 시작합니다.

1. IBM Passport Advantage에서 Build Forge를 설치할 호스트에 있는 임시 디렉토리에 운영 체제에 대한 설치 패키지를 다운로드하십시오.
2. 설치 이미지를 다운로드한 파일에서 로컬 디렉토리로 추출하십시오. 파일의 콘텐츠가 로컬 디렉토리에 추출됩니다.
3. 다음과 같이 파일을 추출한 디렉토리에서 런치패드 프로그램을 시작하십시오.
  - Windows: launchpad.exe를 실행합니다.
  - UNIX/Linux: launchpad.sh를 실행합니다.
4. Installation Manager에서 설치할 패키지를 선택하십시오.
  - 호스트에 IBM Installation Manager가 있는 경우 런치패드에서 이를 시작합니다.

첫 번째 패키지 설치 페이지에서 **Build Forge** 패키지를 선택한 후 다음을 클릭하십시오.



- IBM Installation Manager가 없는 경우 Installation Manager를 설치하기 위해 패키징된 Installation Manager가 시작되고 이를 사용하여 Build Forge가 설치됩니다.

첫 번째 패키지 설치 페이지에서 **Installation Manager** 및 **Build Forge** 패키지를 선택한 후 다음을 클릭하십시오.

---

## Installation Manager에서 설치 단계

대부분의 플랫폼에서 IBM Installation Manager를 사용하여 제품 컴포넌트를 설치합니다.

### 시작하기 전에

다음 지시사항을 수행하려면 Installation Manager를 시작하고 설치할 **Build Forge** 패키지를 선택해야 합니다.

### 이 태스크 정보

지시사항에 따라 원하는 패키지를 설치하십시오.

### 프로시저

1. 패키지 설치 - **Build Forge** 및 버전 선택란을 선택하십시오. Installation Manager가 아직 설치되어 있지 않은 경우, **Installation Manager** 선택란을 선택하십시오. 선택한 후 다음을 클릭하십시오.
2. 패키지 설치: 위치 - 공유 자원 - 공유 자원을 설치할 디렉토리를 입력하거나 선택한 후 다음을 클릭하십시오.
3. 패키지 설치: 위치 - 패키지 그룹 - 설치 패키지를 설치할 디렉토리를 선택한 후 다음을 클릭하십시오. 기본 위치는 C:\Program Files\IBM\Build Forge입니다.
4. 패키지 설치: 기능 - 기본적으로 웹 인터페이스, 프로세스 엔진 및 서비스 계층과 같은 세 개의 모든 코어 제품 모듈이 설치됩니다. 다음을 클릭하십시오.
5. 패키지 설치: 라이선스 서버 구성 - 콘솔에서 사용할 라이선스 유형 및 "실행 도구" 사용자를 선택하십시오. 다음 정보를 입력한 후 다음을 클릭하십시오.
  - (UNIX 및 Linux) **Build Forge**를 실행할 사용자?에서 기본 사용자(루트)를 승인하거나 다른 사용자를 지정하십시오. 이 사용자는 Build Forge 엔진 및 제공되는 Apache Tomcat을 시작합니다. 사용자는 Installation Manager의 데이터베이스 구성 페이지에 지정된 JDBC jar 파일 및 데이터베이스 라이브러리에 대한 읽기 및 실행 권한이 있어야 합니다.
  - **Rational License Server** 기반: Rational License Server의 호스트 이름을 입력하십시오.



- 라이선스 서버의 올바른 호스트 이름을 입력합니다. 호스트 이름을 나중에 제공하려는 경우 이 필드를 공백으로 두지 마십시오. 이 필드에 문자 또는 값을 입력하십시오. 이 필드를 공백으로 두면 제품이 불완전하게 되어 사용할 수 없습니다.

설치를 완료하면 올바른 호스트 이름이 제공됩니다. 지시사항은 23 페이지의 『Build Forge의 Rational License Server 구성』의 내용을 참조하십시오.

- 라이선스 서버가 표시되지만 회색으로 나타나면 FLEXlm License Client에 호스트의 라이선스 서버가 이미 등록되어 있는 것입니다.

설치가 완료되면 올바른 호스트 이름이 제공됩니다. 지시사항은 24 페이지의 『관리 콘솔의 라이선스 서버 변경』의 내용을 참조하십시오.

- 파일 기반: 라이선스 파일을 다운로드한 위치를 찾으십시오.

## 6. 데이터베이스 구성

Build Forge를 설치할 OS 플랫폼 및 설치할 데이터베이스에 따라 특정 정보를 지정해야 합니다. 37 페이지의 『데이터베이스 설정』에서 다음에 대한 데이터베이스 설정 지시사항을 참조하십시오.

- DB2 Express의 경우 37 페이지의 『DB2 Express 설정』의 내용을 참조하십시오.
- DB2의 경우 39 페이지의 『DB2 설정』의 내용을 참조하십시오.
- Microsoft SQL Server의 경우 43 페이지의 『Microsoft SQL Server 설정』의 내용을 참조하십시오. 지정할 JDBC 드라이버는 선택한 버전(SQL Server 2005 또는 SQL Server 2008)에 따라 달라집니다.
- MySQL의 경우 48 페이지의 『MySQL 설정』의 내용을 참조하십시오.
- Oracle의 경우 52 페이지의 『Oracle 설정』의 내용을 참조하십시오. 설치한 Oracle Instant Client 버전(Oracle 10 또는 Oracle 11)에 해당하는 버전을 선택해야 합니다.

참고: UNIX 및 Linux 플랫폼에서, Oracle, DB2 또는 MySQL 데이터베이스를 사용하는 경우에는 32비트 데이터베이스 클라이언트 드라이버를 설치하고 사용해야 합니다. Installation Manager의 데이터베이스 구성 페이지에서는 특정 데이터베이스 유형에 대해 데이터베이스 구성 페이지의 [DB2|Oracle|MySQL] 클라이언트 라이브러리의 경로 필드에 32비트 버전의 드라이버 라이브러리가 지정되어 있는지 확인하십시오. 또한 DB2의 경우 db2profile을 로드하십시오.

## 7. 패키지 설치: 애플리케이션 및 웹 서버 구성 - 요청된 정보를 입력한 후 다음을 클릭하십시오.

### a. 웹 서버/PHP 구성

- 자체 웹 서버를 제공하시겠습니까? 자체 웹 서버를 제공하려면 예를 선택하십시오. 기본값은 아니오입니다.
- 보안 HTTP를 사용하시겠습니까? 보안 HTTP를 사용하려면 예를 선택하십시오. 기본값은 아니오입니다.
- 웹 서버에서 사용해야 할 포트는 무엇입니까? 웹 서버에 포트 80을 사용하지 않으려면 다른 포트 번호를 입력하십시오. 기본값은 80입니다.
- PHP의 메모리 한계: 기본값을 사용하지 않으려면 PHP의 메모리 한계를 입력하십시오. 기본값은 256MB입니다.

b. 애플리케이션 서버 구성

- 자체 애플리케이션 서버를 제공하시겠습니까? 이미 설치한 애플리케이션 서버를 사용하도록 Build Forge를 구성하려면 예를 선택하십시오. 기본값은 아니오입니다.

참고: 이미 설치된 하나 이상의 컴포넌트를 사용하도록 Build Forge를 구성하는 경우에만 이를 수행합니다. 일반적으로 Build Forge는 설치 중에 이러한 컴포넌트를 설치합니다. 75 페이지의 『자체 컴포넌트를 사용하여 설치』의 내용을 참조하십시오.

- c. 보안 구성 설치 프로그램이 다음 필드에 Build Forge에 대해 작성하는 키 저장소에 대한 비밀번호를 입력하고 확인하십시오.

참고: 이러한 필드가 표시되지 않는 경우 아래로 스크롤하여 해당 필드를 찾으십시오.

- 키 저장소 비밀번호
- 비밀번호 확인

기본 보안 로그인(신임 정보 암호화)을 사용하려면 비밀번호가 필요합니다. 또한 Build Forge가 HTTPS/SSL을 사용하도록 구성하려는 경우에도 필요합니다.

중요사항: 나중에 비밀번호를 변경할 수는 있지만 프로세스가 깎니다. 복잡도에 대한 로컬 요구사항을 충족하는 강력한 비밀번호를 사용하십시오.

다음과 같은 옵션이 있습니다.

- 설치 프로그램에서 그대로 작성되는 자가 서명 인증서를 사용합니다. 다음을 수행하십시오.
  - 1) 기본값을 수정하거나 사용자 정의 인증서를 업로드하시겠습니까? 아니오를 선택하십시오.
  - 2) 기존 보안 인증서가 있습니까? 아니오를 선택합니다.
- 설치 프로그램에서 작성되는 자가 서명 인증서를 사용하지만 해당 필드를 수정합니다. 다음을 수행하십시오.

1) 기본값을 수정하거나 사용자 정의 인증서를 업로드하시겠습니까? 예를 선택하십시오. 인증에 대한 추가 필드가 표시됩니다. 해당 필드를 입력하십시오.

2) 기존 보안 인증서가 있습니까? 아니오를 선택합니다.

- 자체 인증서의 위치를 제공합니다. 호스트에 인증서가 있고 pkcs12 형식이어야 하며 해당 인증서가 있는 키 저장소에 대한 기존 비밀번호를 제공해야 합니다. 다음을 수행하십시오.

1) 기본값을 수정하거나 사용자 정의 인증서를 업로드하시겠습니까? 예를 선택하십시오.

2) 기존 보안 인증서가 있습니까? 예를 선택합니다. 추가 필드가 표시됩니다.

3) **pkcs12** 유형의 키 저장소에 서명된 인증서를 지정하십시오. 파일 이름을 입력하거나 찾아보기를 사용하여 해당 파일 이름을 찾습니다.

4) 키 저장소 비밀번호 인증서가 포함된 키 저장소의 비밀번호를 입력합니다.

5) 비밀번호 확인 인증서가 포함된 키 저장소의 비밀번호를 다시 입력합니다.

참고: 일반적으로 브라우저에 자가 서명 인증서가 있는 보안 사이트의 액세스에 대한 경고가 표시됩니다. 일반적으로 사용자에게 계속해서 수행할 수 있는 옵션이 제공되지만 예외를 확인해야 할 수도 있습니다.

8. 서비스 구성 요청된 정보를 입력한 후 다음을 클릭하십시오.

- 이 시스템에서 서비스 계층이 실행됩니까? 이 시스템에서 서비스 계층을 실행하지 않으려면 아니오를 클릭하십시오. 기본값은 예입니다.

아니오를 클릭하면 서비스 계층 애플리케이션이 실행되는 애플리케이션 서버의 호스트 이름에 대한 프롬프트가 표시됩니다.

- 수신 대기 포트: 기본값은 3966입니다. 하나 또는 두 포트 모두 서비스 포트로 선택해야 합니다.

- 수신 대기 보안 포트: 기본값은 49150입니다. 하나 또는 두 포트 모두 서비스 포트로 선택해야 합니다.

9. 패키지 설치: 콘솔 시작 옵션 - 요청된 정보를 입력한 후 다음을 클릭하십시오.

a.

- 데스크탑에 단축 아이콘을 작성하시겠습니까? 기본적으로 데스크탑에 단축 아이콘이 작성됩니다(Windows의 경우).

참고: Linux에서는 데스크탑 단축 아이콘이 작성되지 않습니다.

b. 시작 옵션

- **콘솔 시작 안 함:** 기본적으로 콘솔이 시작됩니다. 콘솔을 자동으로 시작하지 않으려면 단일 선택 단추를 클릭하십시오.
  - **서비스 모드에서 콘솔 시작:** 기본적으로 콘솔은 서비스 모드에서 시작됩니다.
  - **포그라운드 모드에서 콘솔 시작:** 기본적으로 콘솔은 포그라운드 모드에서 시작되지 않습니다. 포그라운드 모드에서 콘솔을 시작하려면 단일 선택 단추를 클릭하십시오.
10. **패키지 설치: 요약 검토** – 이 페이지에서 요약 정보를 검토하여 Build Forge 컴포넌트가 설치되는 위치를 확인한 다음 설치를 클릭하십시오.
- a. **대상 위치**
    - **패키지 그룹 이름:** 기본 패키지 이름은 buildforge.console입니다.
    - **설치 디렉토리:** 기본 설치 디렉토리는 C:\Program Files\IBM\Build Forge입니다.
    - **공유 자원 디렉토리:** 기본 공유 자원 디렉토리입니다. 기본값은 다음과 같습니다.
      - Windows: C:\Program Files\IBM\SDP70Shared
      - UNIX 및 Linux: /opt/IBM/SDP70Shared
  - b. **기능**
    - **Build Forge** 기능 설치되는 기능 또는 모듈을 검토할 수 있습니다. 예를 들어, 코어 제품 모듈은 웹 인터페이스, 프로세스 엔진 및 서비스 계층입니다.
  - c. **환경**
    - 기본 환경은 영어입니다.
  - d. **저장소 정보**
    - **다음 위치에서 파일 검색:** – 이 섹션을 사용하여 저장소 위치를 검토하고 확인합니다.
11. **관리 콘솔 액세스** 브라우저를 시작하십시오. 관리 콘솔의 URL로 이동하십시오.
- **일반 양식:** http://<hostname>[:<portnumber>]. HTTP 기본 포트 80을 사용한 경우, 포트 번호는 선택적입니다.
  - **로컬:** 브라우저를 관리 콘솔과 동일한 호스트에서 실행하는 경우, http://localhost를 사용할 수 있습니다.
- 참고:** 설치 후 즉시 로그인할 수 없으면 몇 분 정도 기다렸다가 다시 시도해 보십시오. 처음 시작할 때 엔진(bfengine)이 데이터베이스 스키마를 로드해야 합니다.

**중요사항:** 설치 후 bfengine을 즉시 중지하지 마십시오. 그러면, 데이터베이스 스키마가 손상될 수 있습니다. 이 경우, 데이터베이스에서 모든 Build Forge 테이블을 삭제하고 Build Forge를 다시 설치해야 합니다.

12. 로그인하십시오. 사용자 이름 **root**, 비밀번호 **root**. root 비밀번호를 즉시 변경하십시오.

---

## 설치 후 점검 목록

이 절에서는 Build Forge를 설치한 후 수행해야 할 작업에 대해 설명합니다.

- PATH 변수 확인
- 관리 콘솔에서 데이터베이스에 액세스할 때 프록시 서버를 통과해야 하는 경우 사용할 PHP에 대한 프록시 서버 식별
- JVM 메모리를 설정합니다.
- 메시지 마이그레이션을 로그합니다.

### Windows에서 PATH 변수 확인

Windows의 PATH 환경 변수에는 데이터베이스 클라이언트 또는 드라이버 DLL의 경로가 포함되어야 합니다. 다음 데이터베이스의 PATH를 수동으로 확인하십시오.

- DB2 - db2cli.dll 및 sqlar.dll이 있는 디렉토리
- MySQL - libmysql.dll이 있는 디렉토리
- Oracle - oci.dll이 있는 디렉토리

Microsoft SQL Server의 경우 ODBC 연결을 설정할 때 이 요구사항에 주의해야 합니다.

### 프록시 서버 지정

선택적: 이 단계는 관리 콘솔이 데이터베이스에 액세스할 때 프록시 서버를 사용해야 하는 경우에만 필요합니다. 프록시 서버를 사용하도록 PHP를 구성해야 합니다.

- php.ini 파일을 편집하십시오. 이는 <bfinstall>/Apache/php(예: C:\Program Files\IBM\Build Forge\Apache\php)에 있습니다.

다음 항목을 추가하십시오.

```
bf_proxyHost=<your_proxy_server_hostname>
bf_proxyPath=<your_proxy_path>
bf_symlinkPath=<symlink_to_proxy_path>
```

### JVM 메모리 설정(빠른 보고서의 경우 필수)

선택사항: 이 단계는 개정판에 빠른 보고서(Standard Edition, Enterprise Edition 및 Enterprise Plus Edition)가 포함되어 있는 경우에만 필요합니다.

- JVM의 최소 메모리를 1MB 이상으로 설정하십시오. 보고서를 실행하려면 최소 1GB(1024MB) 힙 크기가 필요합니다. 보고서 실행 중에 메모리 부족 오류가 발생하면(보고서가 큰 경우에 발생할 수 있음), 이 설정을 늘리십시오. 이 경우, 호스트에 메모리를 추가해야 합니다.

`JAVA_OPTS -Xmx1024M`

## 메시지 마이그레이션 로그

Build Forge는 이전 릴리스와는 다른 방식으로 작업 메시지를 저장합니다. 마찬가지로 Build Forge 설치가 이전 릴리스의 업그레이드인 경우 작업 출력 페이지의 "메시지" 열은 해당 특정 작업의 마이그레이션이 완료될 때까지 임시로 비어 있습니다. 우선순위가 낮은 소규모 일괄처리에서 발생한 경우 마이그레이션에 시간이 걸릴 수 있습니다. 작업에 빈 메시지가 있는 경우 작업의 메시지 마이그레이션이 완료될 때까지 기다리는 것이 좋습니다.

## Linux용 파일 핸들 수 늘리기

### 시작하기 전에

**중요:** 최상의 결과를 얻으려면 Rational 제품을 사용하기 전에 Rational Build Forge에 사용할 수 있는 파일 핸들 수를 늘리십시오. 시스템 관리자가 이를 변경해야 합니다.

Linux에서 다음 단계에 따라 파일 디스크립터를 늘리는 경우 주의하여 실행하십시오. 지시사항을 수행하는 데 실패하면 컴퓨터가 올바르게 시작되지 않을 수 있습니다. 최상의 결과를 얻으려면 시스템 관리자가 이 프로시저를 수행해야 합니다.

파일 디스크립터를 늘리려면 다음을 수행하십시오.

### 프로시저

1. 루트로 로그인하십시오. 루트 액세스 권한이 없는 경우 계속하기 전에 이 권한을 받아야 합니다.
2. `etc` 디렉토리로 변경하십시오.
3. `initscript` 쉘 스크립트를 찾으십시오. 파일을 열거나 Linux 텍스트 편집기를 사용하여 해당 파일을 작성하십시오.

**중요:** 컴퓨터에서 `initscript` 파일을 비워두지 마십시오. 이 파일이 비어 있는 경우 다음에 컴퓨터를 켜거나 다시 시작할 때 컴퓨터가 시작되지 않습니다.

4. 첫 번째 행에서 `ulimit`를 대부분의 Linux 컴퓨터에서 기본값인 1024보다 훨씬 큰 숫자로 설정하십시오.

`ulimit -n 4096`

주의: ulimit를 너무 높게 설정하면 시스템 전체 성능에 영향을 줄 수 있습니다.

5. 두 번째 행에 `eval exec "$4"`를 입력하십시오.
6. 셸 스크립트를 저장하고 닫으십시오.

## 결과

ulimit 명령에 대한 자세한 정보는 ulimit에 대한 man 페이지를 참조하십시오.





---

## 제 8 장 대체 설치 방법

이 절에서는 관리 콘솔 설치의 대체 방법에 대해 설명합니다. 다음 시나리오에 대한 지시사항이 제공됩니다.

- Build Forge에서 제공되고 설치되는 애플리케이션 대신 다음 애플리케이션의 자체 설치 사용
  - Apache HTTP Server 및 PHP
  - Apache Tomcat
- IBM Installation Manager를 사용하여 제품 컴포넌트 자동 설치
- VMware에 설치
- System z의 SUSE Linux에 설치

---

### 자체 컴포넌트를 사용하여 설치

이 컴포넌트가 이미 설치되어 있고 이미 있는 컴포넌트 중 Rational® Build Forge®에서 제공하지 않은 컴포넌트를 사용하려는 경우 이 섹션을 사용하여 필수 기술을 설정합니다.

Build Forge에서는 다음 필수 컴포넌트와 기술의 설치 및 구성을 자동화합니다.

- Apache HTTP Server 및 PHP
- Apache Tomcat

**중요사항:** 사용하는 각 컴포넌트의 버전은 제품과 함께 제공되는 컴포넌트와 동일해야 합니다. 자세한 정보는 21 페이지의 『하드웨어, 소프트웨어 및 데이터베이스 요구사항』의 내용을 참조하십시오.

Build Forge 요구사항을 충족하도록 각 컴포넌트의 설치를 구성한 다음 Installation Manager를 실행하여 Build Forge 컴포넌트를 설치해야 합니다. 설치하는 동안 이미 설정한 기술을 사용할 수 있습니다.

다음 절에서는 Build Forge에서 사용할 각 기술을 설정하는 방법 및 Build Forge를 설치하여 해당 기술을 사용하는 방법에 대해 설명합니다.

### 전제조건

UNIX 또는 Linux에서 설치를 수행하려면 다음 항목이 필요합니다.

- 인터넷 액세스. Build Forge를 설치하는 컴퓨터에서 인터넷에 액세스할 수 없는 경우, 액세스할 수 있는 컴퓨터에서 파일을 다운로드한 후 Build Forge 시스템으로 전송하여 단계를 완료해야 합니다.

- 플랫폼에서 올바르게 작동하는 C 컴파일러(예: Linux의 gcc 컴파일러)
- 컴파일러 제조업체에서 제안하는 기능 작성(예: gcc와 함께 사용할 gnu-make)
- root로서의 권한
- SSL을 사용하려면 OpenSSL을 컴파일해야 합니다.

## 국제적 데이터 지원

Build Forge가 관리 콘솔에서 국제적 데이터를 지원하도록 설정해야 합니다.

### 시작하기 전에

- 웹 브라우저:
  - 언어를 설정해야 함
  - 데이터를 표시하는 데 사용할 글꼴을 설치해야 함
- 에이전트:

Build Forge는 에이전트 서버에서 UTF-8 문자 세트를 사용할 것을 권장합니다.

UNIX/Linux에서 다음 명령을 사용하여 로케일 및 문자 세트를 확인하십시오.

```
locale
```

언어 및 문자 세트를 지정하는 값을 확인해야 합니다. 다음은 언어가 미국 영어이고 UTF-8 문자 세트를 사용하는 Solaris 시스템의 예제입니다.

```
LANG=en_US.UTF-8
LC_CTYPE="en_US.UTF-8"
```

- 모든 데이터베이스:

일반적으로 국제적 데이터 지원은 데이터베이스를 작성할 때 지정됩니다. 국제적 데이터 지원은 데이터베이스 작성 이후에 구성될 수 없습니다.

데이터를 표시하는 데 사용할 글꼴을 데이터베이스 호스트 컴퓨터에 설치해야 합니다.

Build Forge에서는 국제적 데이터를 사용해야 합니다(UTF-8 문자 세트).

- **DB2:**
  1. 코드셋 및 지역을 설정하십시오. 예: CREATE DATABASE USING CODESET UTF-8 TERRITORY US(또는 제어 센터에서 해당 코드셋 및 지역을 선택하십시오).
  2. 관리 콘솔 컴퓨터의 DB2CODEPAGE 환경 변수를 1208로 설정하십시오.

Windows의 경우 다음 명령을 사용하십시오.

```
set DB2CODEPAGE=1208
```

UNIX 또는 Linux의 경우 다음 명령을 사용하십시오.

```
export DB2CODEPAGE=1208
```

기존 데이터베이스에 UTF-8로 마이그레이션해야 하는 데이터가 있는 경우, 다음 문서를 참조하십시오. <http://publib.boulder.ibm.com/infocenter/db2luw/v9/index.jsp?topic=/com.ibm.db2.udb.admin.doc/doc/t0024033.htm>

- **MySQL:** 서버 문자 세트 및 조합을 설정하십시오. 현재 MySQL 설치가 국제적 데이터를 지원하지 않는 경우, 소스에서 재컴파일하여 `./configure --with-charset=utf8 --with-collation=utf8_bin`을 사용할 수 있습니다. 이 지원을 구성하지 않으면 Build Forge 엔진이 시작되지 않습니다.
- **Oracle:** 이를 설치한 인스턴스에서 문자 세트를 **UTF8 - Unicode 3.0**으로 설정하십시오. 데이터베이스 구성 보조의 문자 세트 탭에 있는 초기화 매개변수 단계에서 설정됩니다.

## 데이터베이스 설치 및 구성

이 절을 사용하여 자체 컴포넌트를 사용하는 Build Forge 관리 콘솔 설치에 사용할 데이터베이스를 설치합니다.

### 시작하기 전에

기타 기술과 Build Forge를 설치 및 구성하기 전에 데이터베이스를 설치하여 데이터베이스 오브젝트로 구성해야 합니다. 다음을 수행해야 합니다.

일반적으로, 다음 태스크를 완료해야 합니다.

- 사용할 데이터베이스 시스템을 지정하십시오. Build Forge가 지원하는 데이터베이스이며 데이터베이스 호스트와 Build Forge 호스트 간에 필요한 네트워크 연결이 있는지 확인하십시오. 데이터베이스에 액세스하는 데 프록시 서버가 필요한 경우, 프록시 서버 이름 및 경로를 알아야 합니다.
- 내부 데이터를 사용하려는 경우, 데이터베이스가 UTF-8 문자 세트를 사용하도록 구성되어 있는지 확인하십시오.
- 일반적으로, 다음과 같이 데이터베이스 오브젝트 및 권한을 작성하십시오.
  - 데이터베이스: 예제에서 데이터베이스 이름은 **build**지만 다른 이름을 사용할 수도 있습니다.
  - 데이터베이스 사용자: 관리 콘솔은 이 사용자 이름을 사용하여 데이터베이스에 액세스합니다. 예제에서 사용자 이름은 **build**지만 다른 사용자 이름을 사용할 수도 있습니다.
  - 데이터베이스 사용자 비밀번호
  - **build** 사용자가 **build** 데이터베이스에 테이블 공간을 작성하는 권한입니다. 데이터를 작성, 수정 및 삭제하려면 소유자 권한이 필요합니다.

각 데이터베이스 유형에 따른 특정 지시사항이 제공됩니다.

## Apache HTTP Server 설치 및 구성

Installation Manager가 Build Forge의 웹 서버로 Apache HTTP Server를 설치하고 구성합니다. 제공된 Apache HTTP Server를 사용하면 Build Forge의 웹 서버를 가장 빠르게 구성할 수 있습니다.

표준 구성에 대한 대안으로 Build Forge에서 설치되고 구성된 Apache HTTP Server 대신 기존 Apache HTTP Server를 구성할 수 있습니다. 제공된 지시사항에서는 운영 체제에서 Apache HTTP Server를 설정하고 구성한 경험이 있는 것으로 간주합니다.

기존 Apache HTTP Server를 사용하려면 다음과 같이 설치를 수정하십시오.

1. Build Forge 애플리케이션을 지정하도록 Apache HTTP Server 구성 파일 (httpd-vhosts.conf)을 수정하십시오.
2. Apache HTTP Server, Build Forge 데이터베이스 및 비밀번호 암호화(이 보안 기능을 사용하려는 경우)에 필요한 PHP를 설치하고 PHP 모듈을 구성하십시오.
3. 데이터베이스에 맞게 Apache 구성

## Installation Manager를 사용하여 Build Forge 설치

Installation Manager의 애플리케이션 및 웹 서버 구성 페이지에 있는 자체 웹 서버 제공 프롬프트에서 예를 선택합니다.

### 필수 소프트웨어

- Apache HTTP Server 2.2.4
- PHP 5.2.4

## Apache 서버 구성 파일 편집

1. 서버 설치의 추가 디렉토리에서 Apache http-vhosts.conf 파일을 찾으십시오.

```
cd <apache-dir>/conf/extras/  
vi httpd-vhosts.conf
```

2. Apache http-vhosts.conf 파일을 편집하십시오. Build Forge에 대한 정보를 httpd-vhosts.conf에 추가하려면 다음 행을 추가하십시오.

```
<VirtualHost *:80>  
    ServerAdmin build@yourdomain.com  
    DocumentRoot /opt/buildforge/webroot/public  
    ServerName ausbuild01.yourdomain.com  
    ServerAlias build.yourdomain.com mc.yourdomain.com  
    ErrorLog logs/ausbuild.error_log  
    CustomLog logs/ausbuild.access_log common  
</VirtualHost>
```

3. Build Forge 웹 애플리케이션을 지정하도록 DocumentRoot 설정을 수정하십시오. 예제에서 Build Forge 설치 디렉토리는 /opt/buildforge입니다.

4. 포트를 80으로 그대로 두거나 로컬에서 Apache HTTP Server를 실행하는 포트로 변경하십시오.

<VirtualHost \*:80>

**중요사항:** 포트 8080을 사용하지 마십시오. 이 포트는 Apache Tomcat의 기본 포트입니다.

5. Apache HTTP Server에 필요한 대로 http-vhosts.conf에서 다른 설정을 수정하십시오.
  - ServerAdmin: Build Forge 관리자의 이메일 주소
  - DocumentRoot: Build Forge 애플리케이션 시작 페이지의 위치
  - ServerName: Build Forge 애플리케이션이 설치된 서버
  - ServerAlias: Build Forge ServerName URL의 선택적 별명
  - ErrorLog: Build Forge 애플리케이션에 대한 Apache 오류 로그
  - CustomLog: Build Forge 애플리케이션에 대한 액세스를 로깅하는 Apache 오류 로그

## Apache HTTP Server의 PHP 설치 및 구성

PHP는 Apache HTTP Server와 함께 설치되지 않습니다. PHP 5.2.4를 설치하고 Apache HTTP Server의 httpd-vhosts.conf를 지정하도록 구성해야 합니다.

## Build Forge 데이터베이스의 PHP 설치 및 구성

PHP를 설치하는 동안 Build Forge 데이터베이스로 사용하는 데이터베이스 유형에 대한 PHP 확장을 선택하고 설치합니다.

### (선택사항) 비밀번호 암호화를 지원하도록 PHP OpenSSL 모듈 구성

SSL을 지원하기 위해 Build Forge에서는 PHP OpenSSL 모듈을 사용합니다. 이 지원은 PHP 5.2.4와 함께 제공되며 추가 구성이 필요하지 않습니다.

비밀번호 암호화를 지원하려면 몇 가지 추가 구성이 필요합니다. 이 구성을 지원하기 위해 PHP 5.2.4가 필요합니다. 다음과 같이 OpenSSL 확장에 대한 패치 파일을 찾아 OpenSSL 디렉토리에 설치한 다음 PHP를 다시 컴파일해야 합니다.

1. Build Forge 설치 디렉토리에 있는 misc 디렉토리에서 php\_openssl.h 및 openssl.c 패치 파일을 찾으십시오. 예를 들어 다음과 같습니다.

Windows	C:\Program Files\IBM\Build Forge\misc
UNIX/Linux	/opt/buildforge/Platform/misc

2. 패치 파일을 Build Forge 설치 디렉토리에 있는 openssl 디렉토리에 복사하십시오.

3. --with-openssl=<path\_to\_openssl> 구성 옵션을 사용하여 PHP를 컴파일하십시오.  
여기서, <path\_to\_openssl>은 Build Forge openssl 디렉토리입니다.

## 데이터베이스에 Apache를 구성하십시오.

데이터베이스에 따라 httpd.conf에 특정 정보를 추가해야 합니다.

### DB2용 Apache 구성

1. 다음 행을 Apache 시작 스크립트의 시작 부분에 추가하십시오(일반적으로  
분배에 따라 /etc/init.d/httpd 또는 /etc/init.d/apache2임).

```
source /home/db2bf/sql1lib/db2profile
```

2. 다음 행을 httpd.conf에 추가하십시오.

```
PassEnv LD_LIBRARY_PATH
PassEnv CLASSPATH
PassEnv LIBPATH
PassEnv VWSPATH
```

### MySQL용 Apache 구성

추가 구성이 필요하지 않습니다.

### Oracle용 Apache 구성

1. 다음 행을 httpd.conf에 추가하십시오.

```
PassEnv LD_LIBRARY_PATH
PassEnv NLS_LANG
PassEnv ORACLE_HOME
PassEnv ORA_NLS
PassEnv ORA_NLS32
PassEnv TNS_ADMIN
```

2. 시동 시 Apache를 시작하는 스크립트에 다음 행을 추가하고(일반적으로,  
/etc/init.d/httpd 또는 /etc/init.d/apache2) 다음 설정에 대한 값을 제공하십  
시오.

```
export LD_LIBRARY_PATH=<value>
export NLS_LANG=<value>
export ORACLE_HOME=<value>
export ORA_NLS=<value>
export ORA_NLS32=<value>
export TNS_ADMIN=<value>
```

## IBM HTTP Server 시작

Build Forge 엔진을 시작하고 관리 콘솔을 시작하기 전에 Apache HTTP Server를 시  
작합니다.

## PHP 설치 및 구성

이 프로시저를 사용하여 관리 콘솔과 함께 사용할 PHP를 설정합니다.

## 시작하기 전에

요구사항:

- 버전: PHP는 5.2.4 이상이어야 함
- 데이터베이스 드라이버: Build Forge 데이터베이스의 PHP 모듈이 설치됨

## 이 태스크 정보

이 절의 지시사항에 따라 Apache HTTP Server 또는 기타 웹 서버에 대한 PHP를 구성합니다. 지시사항에서는 필수 버전의 PHP를 이미 다운로드한 것으로 가정합니다.

- PHP 다운로드
- PHP 설치
- PHP 구성
- Apache 구성 파일 편집
- (선택사항) 데이터베이스에 액세스할 때 사용할 프록시 서버 식별(관리 콘솔 호스트가 프록시 서버를 통해 데이터베이스에 액세스하는 경우에만 필요)

## PHP 설치

### 이 태스크 정보

이 절은 PHP를 소스에서 컴파일하고 설치하는 방법을 보여줍니다. 이미 설치된 PHP가 있고 다시 컴파일하지 않으려면 적절한 데이터베이스 드라이버가 설치되어 있는지 확인하십시오. 데이터베이스 드라이버를 설치해야 한다면 사용할 설치 방법을 PHP 문서에서 찾아 참고하십시오. Build Forge와 사용할 다음 데이터베이스 드라이버를 설치하십시오.

- DB2: ibm\_db2 드라이버
- MySQL: mysqli 드라이버
- Oracle Instant Client: oci8 드라이버

**참고:** 현재 전체 Oracle 클라이언트가 PHP oci8과 호환되지는 않습니다. Oracle Instant Client만 호환됩니다.

## 프로시저

1. 작성한 작업 디렉토리에 PHP를 설치하도록 구성하십시오.

```
$ ./configure --prefix=/usr/local/php-5.2.4 --with-<database>=shared \  
--with-apxs2 --with-ldap=shared --enable-mbstring --enable-shmop \  
--with-xml --with-zlib=shared
```

--with-<database>를 다음과 같이 바꾸십시오.

- DB2: --with-ibm\_db2[=*dir*]. 만약 =*dir*이 지정되지 않으면 다음 기본값이 사용됩니다:/home/db2inst1/sqllib

- MySQL: `--with-mysqli[=file]`. 선택적 파일 매개변수는 `mysql_config`에 대한 경로 이름입니다.
- Microsoft SQL Server: `--with-mssql[=dir]`
- Oracle: **PHP oci8**을 사용하려면 별도의 **Oracle Instant Client**의 설치를 설치해야 합니다. Oracle Instant Client를 사용하여 데이터베이스에 연결하는 경우 `--with-oci8=instantclient,lib`를 사용하십시오. 여기서, *lib*는 Instant Client 라이브러리 디렉토리에 대한 경로입니다.

코드 블록에서 행 연속 문자 \에 유의하십시오. 이 단계는 PHP 설치 위치 및 설치 시 사용할 옵션을 지정합니다. 기본적으로, Apache는 `/usr/local`에 설치됩니다. 예제는 PHP를 `/usr/local/php-5.2.4`에 설치하는 방법을 보여줍니다. 이 위치는 나중에 예제에서 사용됩니다.

## 2. PHP를 컴파일하십시오.

```
$ make
```

이 단계는 로컬 디렉토리에서 실행 파일을 컴파일합니다.

## 3. PHP를 설치하십시오(root로).

```
# make install
```

이 단계는 Apache 설치 디렉토리(이 예제의 경우, `/usr/local/apache-2.2.4`)에 쓰기 권한을 갖는 사용자로 수행해야 합니다. 대개, root로 수행됩니다. 로컬 관리 설정은 다를 수 있습니다.

# PHP 구성

## 프로시저

1. 확장 파일을 확장 디렉토리로 복사하십시오. 데이터베이스의 확장 파일은 저장소에서 활성 확장 디렉토리까지 복사해야 합니다. 다음 예제는 PHP가 `/usr/local/php-5.2.4`에 설치되었다고 가정합니다. `<datestamp>`는 일련의 숫자입니다.

```
$ cd /usr/local/php-5.2.4/lib/php/extensions/no-debug-non-zts-<datestamp>/
$ cp <db-extensions> ..
```

`<db-extensions>` 파일은 다음과 같이 Build Forge의 데이터베이스에 해당합니다.

- DB2: `ibm_db2.so`
- MySQL: `mysql.so` 및 `mysqli.so`
- Oracle: `oci8.so`

2. PHP 구성 파일 `php.ini`를 편집하십시오. 다음 예제는 PHP가 `/usr/local/php-5.2.4`에 설치되었다고 가정합니다.

```
$ cd /usr/local/php-5.2.4/lib/
$ vi php.ini
```



다음 항목을 추가하십시오.

```
extension_dir=/usr/local/php-5.2.4/lib/php/extensions
upload_tmp_dir=<directory>
extension=<db-extension-so>
```

<db-extensions-so> 파일 이름을 다음과 같이 데이터베이스에 대해 사용하십시오.

- DB2: extension=ibm\_db2.so
- MySQL: 두 항목 -  
extension=mysql.so  
extension=mysqli.so
- Oracle: extension=oci8.so

참고: upload\_tmp\_dir에 사용되는 디렉토리는 Apache 웹 서버를 실행하는 사용자가 쓸 수 있어야 합니다. 일반적으로, 이 사용자는 nobody이지만 로컬 관리 실행은 다를 수 있습니다.

## Apache 구성 파일 편집

### 프로시저

Apache 구성 파일을 편집하십시오. httpd.conf에 PHP에 대한 정보를 추가하십시오.

```
cd <apache-dir>
vi httpd.conf
```

다음 행을 추가하십시오.

```
LoadModule php5_module modules/libphp5.so
AddHandler php5-script .php
AddType text/html .php
DirectoryIndex index.php
```

## 프록시 서버 지정

### 이 태스크 정보

선택적: 이 단계는 관리 콘솔이 데이터베이스에 액세스할 때 프록시 서버를 사용해야 하는 경우에만 필요합니다.

### 프로시저

PHP 구성 파일 php.ini를 편집하십시오. 이 파일은 <php-install>/lib에 있습니다(예:/usr/local/php-5.2.4).

다음 항목을 추가하십시오.

```
bf_proxyHost=<your_proxy_server_hostname>
bf_proxyPath=<your_proxy_path>
bf_symlinkPath=<symlink_to_proxy_path>
```

## Apache Tomcat 설치 및 구성

Installation Manager가 Build Forge의 애플리케이션 서버로 Apache Tomcat을 설치하고 구성합니다. 제공된 Apache Tomcat 애플리케이션 서버를 사용하면 Build Forge의 애플리케이션 서버를 가장 빠르게 구성할 수 있습니다.

표준 구성에 대한 대안으로 Build Forge에서 제공되는 Apache Tomcat 대신 기존 Apache Tomcat을 사용할 수 있는 옵션이 있습니다. 이 절에서는 이 대안에 대한 필수 소프트웨어, 설치 전 설정, 설치 및 설치 후 요구사항에 대해 설명합니다. 제공된 지시사항에서는 Apache Tomcat을 설정하고 구성한 경험이 있는 것으로 가정합니다.

### 필수 소프트웨어

- Apache Tomcat 서버:
  - Solaris 플랫폼용 5.5.28
  - 기타 모든 플랫폼용 5.5.9
- J2SE 5: IBM 또는 Sun
- Build Forge 데이터베이스의 JDBC 데이터베이스 드라이버: Apache Tomcat과 함께 사용하려면 JDBC(Java Database Connectivity) 드라이버가 필요합니다. Sun은 <http://developers.sun.com/product/jdbc/drivers>에서 JDBC 벤더 목록을 제공합니다.

### JDBC 드라이버의 jar 파일 설치

데이터베이스의 JDBC 드라이버를 다운로드하여 압축 해제하십시오.

**중요사항:** JDBC 드라이버 다운로드에는 여러 가지 파일 및 서브디렉토리가 포함되어 있을 수 있습니다. JDBC 드라이버의 jar 파일을 찾고 해당 jar 파일을 \$CATALINA\_HOME/common/lib.

에만 복사합니다.\$CATALINA\_HOME은 Tomcat 설치 루트이며 환경 변수로 설정해야 합니다. JDBC 드라이버는 설치 문서를 참조하십시오.

- DB2 - <http://www-306.ibm.com/software/data/db2/express/download.html>

**JDBC 및 SQLJ의 DB2 드라이버** 옆의 다운로드 링크를 클릭하십시오. IBM 계정을 등록해야 합니다. 또한, licensing.jar인 db2jcc\_license\_cu.jar를 찾아서 설치해야 합니다.

- MySQL - <http://www.mysql.com/products/connector/j/>

**MySQL Connector/J 5.0** 또는 **5.1**의 링크를 클릭하십시오. MySQL 버전에 해당하는 JDBC 드라이버 버전을 선택하십시오.

- Oracle - [http://www.oracle.com/technology/software/tech/java/sqlj\\_jdbc/index.html](http://www.oracle.com/technology/software/tech/java/sqlj_jdbc/index.html)

Oracle 버전 옆의 다운로드 링크를 클릭하십시오. 계정을 등록해야 합니다.

- Microsoft SQL Server- <http://msdn.microsoft.com/en-us/data/aa937724.aspx>

SQL Server JDBC 드라이버 다운로드 링크를 클릭하십시오.

## Installation Manager에서 Apache Tomcat 서버 구성

다음 지시사항은 Installation Manager를 통해 Apache Tomcat을 구성하는 데 필요한 정보를 식별합니다.

1. Apache Tomcat을 종료하십시오.

**중요사항:** Installation Manager를 시작하기 전에 Apache Tomcat을 중지해야 합니다.

2. Installation Manager를 시작하십시오.
3. 시작 페이지에서 설치를 클릭하십시오.
4. Installation Manager 마법사의 지시사항에 따라 제품을 설치하십시오.
5. 애플리케이션 서버 구성 페이지에서 예를 클릭하여 고유 애플리케이션 서버를 구성하십시오.

**점검 목록:** 애플리케이션 서버 구성

✓	필드	설명
	URL 리디렉션	애플리케이션 서버의 호스트 이름과 포트 번호를 입력합니다. 컨텍스트 경로로 rbf-services를 지정해야 합니다. 예를 들어 다음과 같습니다. <code>http:   https://&lt;app_server_host&gt;:&lt;app_server_port&gt;/rbf-services</code> .
	BF 서비스 플러그인을 설치할 디렉토리 지정	애플리케이션 서버 호스트에 로컬 디렉토리를 지정합니다. Installation Manager가 이 디렉토리에 Build Forge 서비스 계층 애플리케이션 플러그인 확장을 설치합니다. 애플리케이션 서버를 실행하는 사용자에게 이 디렉토리에 대한 읽기, 쓰기 및 실행 권한이 있어야 합니다.

**참고:** Build Forge의 이전 버전에서는 WAR 배치 디렉토리를 지정해야 합니다. WAR 배치 디렉토리는 자동으로 `bfinstall/PrepForExternal`로 설정됩니다.

6. Installation Manager를 통해 설치를 완료하십시오.

## Apache Tomcat의 설치 후 구성

Installation Manager를 통해 설치를 완료한 후, 다음의 설치 후 단계를 완료합니다.

1. buildforge.conf 파일을 rbf-services.war 파일에 수동으로 추가해야 합니다.

rbf-services.war의 위치는 운영 체제에 따라 달라집니다. UNIX 및 Linux에서는 이 파일은 /opt/buildforge/PrepForExternal에 배치합니다. Windows에서는 이 파일은 C:\Program Files\IBM\Build Forge\PrepForExternal에 배치합니다. 99 페이지의 『buildforge.conf 파일 업데이트』의 내용을 참조하여 관련 단계를 완료하십시오.

2. Apache Tomcat 서버의 JVM 힙 크기를 늘리십시오.

JVM 최대 힙 크기 옵션을 -Xmx에서 1024M로 설정하십시오.

catalina.bat 또는 catalina.sh에서 CATALINA\_OPTS 또는 JAVA\_OPTS 환경 변수를 사용하여 이 JVM 옵션을 설정하십시오.

3. Build Forge를 시작하기 전에 Apache Tomcat을 시작하십시오(

\$CATALINA\_HOME/bin/catalina.sh start).

---

## Installation Manager 수동으로 설치

런치패드 프로그램을 사용하여 제품 설치를 시작하는 경우 IBM Installation Manager가 자동으로 설치되거나 업데이트됩니다. 64 페이지의 『런치패드를 사용하여 Installation Manager 시작』의 내용을 참조하십시오.

Installation Manager를 사용해 본 사용자 또는 자동 설치를 설정하고자 하는 사용자가 Installation Manager를 수동으로 설치할 수 있습니다. 다음 단계를 수행하십시오.

1. Passport Advantage에서 다운로드하거나 제품 DVD를 사용하여 제품 설치 패키지를 가져오십시오.
2. 플랫폼에 대한 Installation Manager 파일을 찾으십시오.
  - InstallerImage\_linux
  - InstallerImage\_solaris
  - InstallerImage\_win32
3. 다음 명령 중 하나를 입력하여 설치 프로그램을 시작하십시오.
  - 관리 사용자로 설치를 시작하려면 다음 명령을 실행하십시오.  
install
  - 비관리 사용자로 설치를 시작하려면 다음 명령을 실행하십시오.  
userinst
4. 설치 지시사항에 따라 Installation Manager를 설치하십시오.

설치 후 Installation Manager 또는 Installation Manager 설치 프로그램을 사용하여 패키지를 자동으로 설정할 수 있습니다.

## Installation Manager 시작

Windows 또는 UNIX/Linux에서 Installation Manager를 시작합니다.

## 시작하기 전에

런치패드 프로그램을 사용하여 프로젝트 설치를 시작하는 경우 Installation Manager가 자동으로 시작됩니다. Installation Manager를 이미 설치한 경우 다음 방법 중 하나로 시작할 수 있습니다.

- Windows: 시작 > 모든 프로그램 > **IBM Installation Manager** > **IBM Installation Manager**를 클릭하십시오.
- <IM-installdir>로 변경하고 ./IBMIM을 실행합니다.

## 저장소 URL 지정

IBM Installation Manager에서는 각 제품 패키지에 있는 임베디드 URL을 사용하여 인터넷을 통해 저장소 서버에 연결하고 최신 제품 설치 패키지를 검색합니다.

## 시작하기 전에

Installation Manager의 환경 설정 창에 있는 저장소 페이지에서 저장소 위치를 설정할 수 있습니다. 조직의 경우 저장소가 인트라넷 사이트를 사용하도록 재지정해야 할 수 있습니다.

**참고:** 설치 프로세스를 시작하기 전에 관리자 또는 IBM으로부터 설치 패키지 저장소 URL을 얻어야 합니다.

저장소를 지정하려면 다음 단계를 완료하십시오.

1. IBM Installation Manager를 시작하십시오.
2. 시작 페이지에서 파일 > 환경 설정을 클릭하십시오.
3. 환경 설정 창에서 저장소를 클릭하십시오. 사용 가능한 저장소, 위치 및 액세스 가능 여부가 표시되는 저장소 페이지가 열립니다.
4. 저장소 페이지에서 저장소 추가를 클릭하십시오.
5. 저장소 추가 대화 상자에 저장소 위치의 URL을 입력하거나 찾아보기를 사용하여 저장소, diskTag.inf 파일 또는 확장된 저장소의 repository.config 파일이 포함되어 있는 .zip 또는 JAR 파일을 찾은 다음 확인을 클릭하십시오.

새 저장소 위치가 나열됩니다. 저장소가 연결되지 않으면 연결 열에 빨간색 x가 표시됩니다.

**참고:** 업데이트된 패키지를 검색하려면 설치 및 업데이트 중 서비스 저장소 검색이 선택되어 있어야 합니다. 이 옵션은 기본적으로 선택됩니다.

6. 확인을 클릭하여 환경 설정 창을 닫으십시오.

## 제품 컴포넌트의 자동 설치 수행

자동 설치 모드에서 Installation Manager를 실행하여 Rational Build Forge 제품 컴포넌트를 자동으로 설치할 수 있습니다. 자동 모드에서는 사용자 인터페이스를 사용할 수 없습니다. 대신, 응답 파일에 제품 패키지를 설치하는 데 필요한 명령이 입력됩니다.

자동 설치에는 다음 태스크가 필요합니다.

1. Installation Manager를 설치하십시오.
2. 응답 파일을 작성하십시오.
3. 자동 설치 모드에서 Installation Manager를 실행하십시오.

참고: 다음 경우에는 자동 설치를 사용할 수 없습니다.

- X11이 설치되지 않은 Linux 서버에 설치하는 경우

Installation Manager 및 자동 설치에 대한 자세한 정보는 Installation Manager Information Center(<http://www.ibm.com/software/awdtools/installmanager/support/index.html>)를 참조하십시오.

## Installation Manager로 응답 파일 작성

Installation Manager를 사용하여 제품 패키지를 설치할 때 조치를 레코딩하여 응답 파일을 작성할 수 있습니다. 응답 파일을 레코딩할 때 Installation Manager UI에서 선택한 사항은 모두 XML 파일에 저장됩니다. Installation Manager를 자동 모드에서 실행하면 Installation Manager에서 XML 응답 파일을 사용하여 설치를 완료합니다.

응답 파일을 작성하고 제품을 설치하거나, `-skipInstall <agentDataLocation>` 인수를 사용하여 제품 설치를 건너뛰고 응답 파일만 작성할 수 있습니다. 다음 지시사항은 두 옵션 모두에 대한 예제 문을 제공합니다.

설치할 응답 파일을 작성하려면 다음을 수행하십시오.

1. 명령행에서 디렉토리를 Installation Manager를 설치한 디렉토리에 있는 eclipse 서브디렉토리로 변경하십시오. 예를 들어, 다음과 같습니다.

Windows	C:\Program Files\IBM\Installation Manager\eclipse
UNIX/Linux	/opt/IBM/InstallationManager/eclipse

2. 명령행에서 다음 명령 중 하나를 사용하여 Installation Manager를 시작하십시오. 응답 파일 및 로그 파일(선택사항)의 파일 이름과 위치를 사용자의 파일 이름과 위치로 대체합니다.

입력한 파일 경로가 있는지 확인하십시오. Installation Manager가 응답 파일 및 로그 파일의 디렉토리를 작성하지 않습니다. `-skipInstall` 옵션을 사용하는 경우 `<agentDataLocation>`은 쓸 수 있는 디렉토리여야 합니다.

- 응답 파일 레코드 및 제품 설치:

```
IBMIM -record <response file path and name> -log <log file path and name>
```

- 제품을 설치하지 않고 응답 파일 레코드:

```
IBMIM -record <response file path and name> -log <log file path and name> -skipInstall <agentDataLocation>
```

3. 패키지 설치 마법사의 화면 지시사항에 따라 설치 관련 선택사항을 선택하십시오.
4. 완료를 클릭한 다음 Installation Manager를 닫으십시오.

XML 응답 파일이 작성되어 명령에 지정된 위치에 저장됩니다.

## 자동 모드에서 Installation Manager 설치 및 실행

Installation Manager를 사용하여 명령행에서 제품 패키지를 자동으로 설치합니다.

자동 설치에는 다음 태스크가 필요합니다.

Installation Manager를 자동 모드에서 실행하려면 eclipse 서브디렉토리에서 다음 명령을 실행하십시오.

Windows	<pre>IBMIM.exe --launcher.ini silent-install.ini -input &lt;response file path and name&gt; -log &lt;log file path and name&gt;</pre> <p>예를 들어 다음과 같습니다. IBMIM.exe --launcher.ini silent-install.ini -input C:\mylog\responsefile.xml -log C:\mylog\silent_install_log.xml</p>
UNIX/Linux	<pre>IBMIM --launcher.ini silent-install.ini -input &lt;response file path and name&gt; -log &lt;log file path and name&gt;</pre> <p>예: IBMIM --launcher.ini silent-install.ini -input /root/mylog/responsefile.xml -log /root/mylog/silent_install_log.xml</p>

Installation Manager를 자동 설치 모드에서 실행하면 응답 파일을 읽고 사용자가 지정한 디렉토리에 로그 파일을 기록합니다. 응답 파일은 필수이며, 로그 파일은 선택적입니다. 이 실행의 결과는 성공 시 상태가 0이고 실패 시 0이 아닌 숫자입니다.

다음 표에서는 자동 설치 명령과 함께 사용할 인수에 대해 설명합니다.

인수	설명
-vm	Java 실행기를 지정합니다. 자동 모드에서는 Windows의 경우 항상 java.exe를 사용하고 다른 플랫폼에서는 java를 사용합니다.
-nosplash	플래시 화면을 표시하지 않습니다.
--launcher.suppressErrors	JVM 오류 대화 상자를 표시하지 않습니다.
-silent	Installation Manager 설치 프로그램을 자동 모드에서 실행합니다.



인수	설명
-input	Installation Manager의 입력으로 사용할 응답 파일을 지정합니다. 응답 파일에는 설치 프로그램 또는 Installation Manager에서 실행하는 명령이 포함되어 있습니다.
-log	(선택사항) 자동 설치의 결과를 기록하는 로그 파일을 작성합니다. 로그 파일은 XML 파일입니다.

## 제품 컴포넌트의 자동 업그레이드 수행

자동 설치 모드에서 Installation Manager를 실행하여 Rational Build Forge 제품 컴포넌트를 자동으로 업그레이드할 수 있습니다.

자동 업그레이드를 수행하려면 다음 전제조건이 필요합니다.

- 기존 Build Forge 콘솔 설치가 Installation-Manager 기반 자동 설치를 사용하여 설치되어 있어야 합니다.
- Installation Manager가 Build Forge 콘솔 호스트와 동일한 호스트에 설치되어야 합니다.

자동 설치에는 다음 태스크가 필요합니다.

1. 업그레이드 응답 파일을 작성하십시오.
2. 응답 파일을 입력으로 지정하여 자동 모드에서 Installation Manager를 실행하십시오.

Installation Manager 및 자동 설치에 대한 자세한 정보는 Installation Manager Information Center(<http://www.ibm.com/software/awdtools/installmanager/support/index.html>)를 참조하십시오.

## Installation Manager로 업데이트 응답 파일 작성

Installation Manager를 사용하여 제품 패키지를 설치할 때 조치를 레코딩하여 응답 파일을 작성합니다.

응답 파일을 레코딩할 때 Installation Manager UI에서 선택한 사항은 모두 XML 파일에 저장됩니다.

업데이트 설치에 사용할 응답 파일을 작성하려면 다음과 같이 수행하십시오.

1. Installation Manager를 실행하십시오. 환경 설정에서 업데이트 설치에 사용할 제품 저장소의 URL을 IM 저장소의 목록에 추가하고 해당 URL이 선택되었는지 확인하십시오.
2. Installation Manager를 종료하십시오.
3. 명령행에서 디렉토리를 Installation Manager를 설치한 디렉토리에 있는 eclipse 서브디렉토리로 변경하십시오. 예를 들어, 다음과 같습니다.



Windows	C:\Program Files\IBM\Installation Manager\eclipse
UNIX/Linux	/opt/IBM/InstallationManager/eclipse

4. 실제로 제품을 설치하지 말고 설치 레코딩을 시작하십시오.

파일 이름을 포함하여 `response_file` 및 `log_file`의 전체 경로를 입력하십시오. 입력한 파일 경로가 존재하는지 확인하십시오. Installation Manager는 응답 파일 및 로그 파일의 디렉토리를 작성하지 않습니다. `agentDataLocation`은 쓰기 가능한 디렉토리여야 합니다.

`IBMIM -record response_file -log log_file -skipInstall agentDataLocation`

5. Installation Manager가 시작됩니다. Installation Manager에서 업데이트를 클릭한 후 프롬프트에 응답하십시오.
6. Installation Manager가 완료되면 완료를 클릭하십시오.
7. Installation Manager를 종료하십시오.

XML 응답 파일이 작성되어 명령에 지정된 위치에 저장됩니다.

## 자동 모드에서 업데이트 설치 실행

Installation Manager를 사용하여 명령행에서 제품 패키지를 자동으로 설치합니다.

Installation Manager를 자동 모드에서 실행하기 위한 일반적인 양식의 명령은 다음과 같습니다.

`IBMIMc.exe --launcher.ini silent-install.ini -input response_file -log log_file`

`response_file` 및 `log_file`에 전체 경로와 파일 이름을 사용하십시오.

- Windows 예제

`IBMIMc.exe --launcher.ini silent-install.ini -input C:\mylog\responsefile.xml -log C:\mylog\silent_install_log.xml`

- UNIX 또는 Linux 예제

`IBMIM --launcher.ini silent-install.ini -input /root/mylog/responsefile.xml -log /root/mylog/silent_install_log.xml`

Installation Manager를 자동 설치 모드에서 실행하면 응답 파일을 읽고 사용자가 지정한 디렉토리에 로그 파일을 기록합니다. 응답 파일은 필수이며, 로그 파일은 선택적입니다. 이 실행의 결과는 성공 시 상태가 0이고 실패 시 0이 아닌 숫자입니다.

다음 표에서는 자동 설치 명령과 함께 사용할 인수에 대해 설명합니다.

인수	설명
-vm	Java 실행기를 지정합니다. 자동 모드에서는 Windows의 경우 항상 <code>java.exe</code> 를 사용하고 다른 플랫폼에서는 <code>java</code> 를 사용합니다.
-nosplash	플래시 화면을 표시하지 않습니다.
--launcher.suppressErrors	JVM 오류 대화 상자를 표시하지 않습니다.
-silent	Installation Manager 설치 프로그램을 자동 모드에서 실행합니다.

인수	설명
-input	Installation Manager의 입력으로 사용할 응답 파일을 지정합니다. 응답 파일에는 설치 프로그램 또는 Installation Manager에서 실행하는 명령이 포함되어 있습니다.
-log	(선택사항) 자동 설치의 결과를 기록하는 로그 파일을 작성합니다. 로그 파일은 XML 파일입니다.

## VMware에 Build Forge 시스템 설치

VMWare에 Build Forge를 설치하고 실행할 수 있습니다.

다음 가이드라인을 사용하십시오.

- 개별 호스트(VMWare 이미지가 아닌 실제 호스트)에 Build Forge에서 사용하는 데이터베이스를 설치하십시오.
- VMware 워크스테이션이 가상 시스템을 실행하는 데 사용하는 메모리를 최소 1GB로 설정하십시오. VMware 워크스테이션에서 편집 > 환경 설정 > 메모리를 클릭하고 이 값을 조정하십시오.
- VMWare의 성능을 최대화하기 위해 다른 시스템 자원 매개변수를 지정해야 할 수도 있습니다.

## System z의 Linux에 관리 콘솔 설치

설치 매체에서 제공되는 mc-<version>-<build>.tar.gz tar 파일을 사용하여 z/Linux에 관리 콘솔을 설치하고 구성하십시오. IBM Installation Manager는 이 설치에 사용되지 않습니다.

z/Linux용 콘솔은 Apache 웹 서버가 아닌 IBM HTTP Server와 패키징되어 있습니다.

관리 콘솔을 설치한 후 z/Linux에서 에이전트 rpm 패키지(zlinux-bfagent-<version>.rpm)를 설치하여 Build Forge에 대해 z/Linux 서버를 설정하십시오. 설치 지시사항은 163 페이지의 『UNIX 및 Linux 시스템에 에이전트 설치』의 내용을 참조하십시오.

### 설치 중에 필요한 정보

설치하는 동안 다음과 같은 정보를 제공해야 합니다.

1. 설치 디렉토리
  - Build Forge를 설치할 위치의 절대 경로를 제공하십시오.
2. 데이터베이스 정보
  - Build Forge에서 사용할 데이터베이스 유형(DB2, Oracle 또는 MySQL)

- 데이터베이스 서버 호스트 이름
- 데이터베이스 포트 번호
- 사용할 데이터베이스 이름
- Build Forge에서 데이터베이스에 연결하는 데 사용할 데이터베이스 사용자 이름
- 데이터베이스 사용자 이름의 비밀번호
- 데이터베이스에 액세스하는 데 사용되는 클라이언트 라이브러리의 위치
- JDBC 드라이버 JAR 파일의 위치

### 3. 애플리케이션 서버 정보

- 사용할 애플리케이션 서버(제공되는 Tomcat 또는 사용자가 설정한 WebSphere Application Server 설치)

WebSphere Application Server를 선택하는 경우 다음과 같은 추가 정보를 제공해야 합니다.

- Build Forge 서비스 컴포넌트를 WAS에 설치할 경우 해당 서비스 컴포넌트의 위치, 도메인, 포트 및 rbf-services에 대한 경로가 필요합니다. 예를 들어 다음과 같습니다.

`http://mydomain.com:9080/rbf-services`

- 플러그인에 사용할 읽기 및 쓰기 가능한 디렉토리. 현재 플러그인은 Rational Team Concert와 통합하는 경우에만 사용됩니다.
- WAR 배치 디렉토리: Build Forge 서비스 WAR의 임시 위치. 설치 후에 WAR을 애플리케이션 서버에 배치합니다.
- Java 실행 파일(java.jar)의 경로
- 서비스 계층의 임시 스토리지 디렉토리. 디렉토리는 읽기 및 쓰기가 가능해야 합니다. 임시 정보를 저장하기 위해 서비스 컴포넌트에서 이 디렉토리를 사용합니다.
- Build Forge 서비스의 HTTP 포트(기본은 3966)
- Build Forge 서비스의 SSL 포트(기본은 49150)

### 4. 웹 서버 정보

- 사용할 웹 서버(제공되는 IBM HTTP Server 또는 사용자가 설정한 웹 서버)

제공되는 IHS 서버를 선택하는 경우 SSL을 사용할지 여부, 사용할 포트와 메모리 및 기존 보안 인증서를 사용할지 또는 새 인증서를 작성할지 여부와 같은 SSL에 대한 추가 정보를 제공해야 합니다.

## 설치 프로그램 실행

1. tar 파일에서 콘솔의 패키지 압축을 풀 디렉토리로 이동하십시오.
2. 다음 명령을 실행하십시오.

```
./cmdline-install.sh
```

## 예제

다음 어노테이션이 있는 목록에는 설치 프로그램 단계를 진행하는 방법이 표시되어 있습니다. 이는 어노테이션이 있는 실행이며 실제 설치를 반영하지 않습니다. 기본값을 사용할 수 있는 경우 기본값이 대괄호 안에 표시됩니다(예: [y]). 기본값을 채택하려면 Enter를 누르십시오.

```
Install directory [/opt/buildforge]
```

```
What database will you be using?
```

```
Enter the # of the database you will use
```

- 1) DB2
- 2) Oracle
- 3) MySQL

참고: 이후로는 Oracle을 선택한 것으로 가정합니다.

```
What is your database hostname? 127.0.0.1
```

```
What is your database port number? [1521]
```

```
What is your database name? build
```

```
What is your database user name? build
```

```
What is your database user password?
```

```
Confirm your database user password?
```

```
Would you like this installation to create the Build Forge database schema? (y|n) [y]
```

이 지점에서 클라이언트 라이브러리 및 정보를 지정합니다. 선택한 데이터베이스에 따라 다음과 같은 프롬프트가 표시됩니다. 절대 경로를 사용하십시오.

- DB2

```
Where are your 32-bit DB2 client libraries (libdb2.so.1)?
```

참고: 라이브러리는 32비트 라이브러리어야 합니다. 절대 경로를 입력하십시오.

```
Where is your DB2 (DB2_HOME) installed?
```

- MySQL

```
Where are your MySQL client libraries (libmysqlclient.so)?
```

참고: 라이브러리는 32비트 라이브러리어야 합니다. 절대 경로를 입력하십시오.

- Oracle

```
Where are your Oracle client libraries (libclntsh.so)?
```

참고: 라이브러리는 64비트 라이브러리어야 합니다. 절대 경로를 입력하십시오.

```
Where is your Oracle instant client (ORACLE_HOME) installed?
```

참고: 이 경로는 인스턴트 클라이언트 라이브러리의 루트에 대한 절대 경로입니다.

```
Where is your tns.names file located (TNS_ADMIN)?
```

참고: 이 위치는 tns.names 파일이 포함되어 있는 디렉토리입니다. 절대 경로를 입력하십시오.

Please enter the directory with your database JDBC jar file?

참고: ojdbc14.jar의 절대 경로를 입력하십시오.

Will you be using the supplied Tomcat app server? (y|n) n

참고: 나머지 목록에서는 애플리케이션 서버로 WAS를 사용한다고 가정합니다.

Enter the full URL used to contact the services layer on your application server:

http://mydomain.com:9080/rbf-services

Enter the directory to install the Build Forge Services plugins to:

참고: 이 디렉토리는 애플리케이션 서버 호스트에서 읽기 및 쓰기가 가능한 디렉토리여야 합니다. Rational Team Concert를 Build Forge와 통합할 경우 서비스 컴포넌트에서 이 디렉토리를 사용합니다.

Specify the war deployment directory:

참고: 로컬 호스트에 디렉토리를 지정하십시오. 설치가 완료되면 설치 프로그램이 서비스 .war 파일을 여기에 배치합니다. 그런 다음 이 파일을 애플리케이션 서버에 배치할 수 있습니다.

Enter the path to a jar executable (which should be included in any JDK):

참고: 사용자의 데이터베이스 드라이버에 적합한 .jar 파일의 경로를 입력하십시오.

Enter the temporary storage path for the Services Layer:

참고: 서비스 컴포넌트에서 사용할 수 있는 디렉토리의 경로를 입력하십시오. 이 경로는 읽기 및 쓰기가 가능해야 합니다.

What http port will the Build Forge services layer use? [3966]

What ssl port will the Build Forge services layer use? [49150]

Will you be using the supplied Apache web server? (y|n)[y]

Note: The rest of the listing assumes you entered 'y'

Would you like Apache to use SSL? (y|n)[n] Enter 'y' to configure Apache for SSL

참고: 나머지 목록에서는 'y'를 입력했다고 가정합니다.

What ssl port will Apache use? [443]

Please enter a memory limit for PHP (in MB): [256]

Would you like to modify or specify a custom SSL certificate? (y|n) [n]

참고: 이후로는 'y'를 입력했으며 사용자 정의 인증서를 작성하려는 것으로 가정합니다.

Do you have an existing secure certificate? (y|n)[n]

A validity period is required for this cert please enter in [number][period] format

Examples: 10Y = 10 years, 6M = 6 months, 350D = 350 daysEnter the validity period for this cert:

```

Enter the common name for the certificate (usually the name of the server)
[linux142.rtp.raleigh.ibm.com]:
Please enter your Locality/City:
Please enter your State/Province:
Please enter your Organization Name:
Please enter your Organization Unit:
Please enter your Country from the list below:
    France
    Taiwan
    Italy
    Germany
    Korea
    United States
    China
    Brazil
    Spain
    Japan
Please enter your Street Address:
A keystore password is required, and must be at least 6 characters long
Please enter a keystore password

```

웹 서버로 Apache 대신 IBM HTTP Server를 사용하는 경우 이를 설정하고 SSL을 사용하는 방법에 대한 추가 정보는 621 페이지의 『Apache HTTP Server 대신 IBM HTTP Server 사용』를 참조하십시오.

## 콘솔 시작

1. 관리 콘솔을 시작하십시오.

```
<bfinstall>/rc/buildforge start
```

2. 시작된 서비스 컴포넌트(Apache Tomcat 서버)를 확인하십시오. catalina.out을 열어 시작 메시지가 로그되었는지 확인하십시오.

```
<bfinstall>/server/tomcat/logs/catalina.out
```

3. 웹 브라우저를 시작하고 완전한 z/Linux 호스트 이름을 입력하십시오. 예: `http://myhost.mycompany.com`.

관리 콘솔이 시작되고 로그인 프롬프트가 표시됩니다. **root/root**로 로그인하십시오.

## 라이선스 파일 설치

z/Linux용 라이선스 파일은 <bfinstall> 디렉토리에 있습니다. 라이선스 파일의 이름은 IRBF\_license입니다. 설치 후 이 파일에는 Passport Advantage에서 실제 라이선스 파일을 다운로드하도록 설명하는 텍스트 메시지가 포함됩니다.

라이선스 파일을 다운로드하여 <bfinstall> 디렉토리에 저장한 후 이를 사용하려면 다음과 같이 Build Forge를 구성하십시오.

1. 관리 콘솔을 시작하십시오.
2. **root/root**로 로그인하십시오.
3. 관리>시스템을 선택하십시오.
4. 라이선스 서버 설정을 찾고 해당 값을 라이선스 파일의 완전한 경로로 설정하십시오.

예를 들어 다음과 같습니다. <bfinstall>/IRBF\_license.

## 관리 콘솔에 SSL 사용

SSL을 사용하여 다음과 같은 Build Forge 컴포넌트 간에 전송되는 데이터를 암호화할 수 있습니다.

- 웹 브라우저 클라이언트와 Apache HTTP Server
- Apache Tomcat Server와 Apache HTTP Server

프롬프트가 표시될 때 예로 응답한 경우 설치 프로그램이 SSL을 사용하기 위해 필요한 몇몇 작업을 수행합니다. 이 절에서 설명하는 태스크를 완료하여 SSL을 사용하십시오.

1. 개인용 인증서 및 키 저장소 검토
2. SSL에 적합하게 IBM HTTP Server 구성
3. 관리 콘솔 UI에서 SSL 사용
4. SSL을 위한 디버깅 사용

**참고:** 비밀번호 암호화 및 싱글 사인온(SSO)과 같은 기타 보안 기능은 이번 릴리스의 z/Linux의 Build Forge에서 지원되지 않습니다.

### 개인용 인증서 및 키 저장소 검토

설치 프로그램이 다음 키 저장소를 작성합니다.

키 저장소	설명
buildForgeKeyStore.p12	비밀번호로 보호되는 keyEntry(공용/개인용 키 쌍이 있는 개인 인증)를 포함합니다.
buildForgeTrustStore.p12	비밀번호로 보호되는 trustedCertEntry(공용 키만 있는 인증)를 포함합니다.
buildForgeKey.pem	비밀번호로 보호되는 개인용 키를 포함합니다.
buildForgeCert.pem	buildForgeKey.pem의 개인용 키에 해당하는 공용 키가 있으며 비밀번호로 보호되지 않는 인증을 포함합니다.
buildForgeCA.pem	초기에는 buildForgeCert.pem과 동일한 정보를 포함합니다. 신뢰를 구축하기 위해 기타 피어가 추가됩니다.
buildForgeKeyForApache.pem	이 키 저장소는 Apache HTTP Server에 대해 SSL을 사용하는 데 필요합니다. buildForgeKey.pem과 달리 비밀번호로 보호되지 않으므로 비밀번호 프롬프트를 표시하지 않고 Apache HTTP Server를 시작할 수 있습니다.

기존 PEM 인증서 변환 및 인증서 관리에 대한 자세한 정보는 137 페이지의 『인증서 관리』의 내용을 참조하십시오.

### SSL에 적합하게 IBM HTTP Server 구성

621 페이지의 『Apache HTTP Server 대신 IBM HTTP Server 사용』의 "IHS에 적합하게 SSL 구성"을 참조하십시오. 키 저장소 형식을 변경하고 httpd.conf에 항목을 추가해야 합니다.

## 관리 콘솔 UI에서 SSL 사용

관리 콘솔 UI(관리 > 보안) 설정을 사용하여 관리 콘솔에서 SSL을 사용하고 Build Forge 데이터베이스를 업데이트하십시오. 그런 다음 `bfclient.conf` 구성 파일에서 필수 특성 값이 업데이트되었는지 확인하십시오.

1. Build Forge를 시작하십시오.
2. UI에 로그인하십시오.
3. 관리 > 보안으로 이동하십시오.
4. SSL 사용:을 예로 변경하십시오.
5. 저장을 클릭하십시오.
6. 마스터 **BFClient.conf** 업데이트를 클릭하십시오.

## SSL을 위한 디버깅 사용

관리 콘솔에서 SSL에 대한 문제를 디버깅하려면 다음 지시사항을 사용하여 SSL에 필요한 추가 정보를 로그하십시오.

1. 엔진에서 디버깅을 사용 가능하게 하십시오. Build Forge 엔진을 시작하기 전에 다음 환경 변수를 설정하십시오.  
`export BFDEBUG_SECURITY=1`
  - a. Build Forge 엔진을 다시 시작하십시오.
  - b. IHS를 다시 시작하십시오. IHS를 다시 시작하면 PHP에서 이 디버그 매개변수를 사용할 수 있습니다.
2. Tomcat에서 디버깅을 사용 가능하게 하십시오. `<bfinstall>/server/tomcat/common/classes/logging.properties`에서 다음과 같이 변경하십시오.
  - a. 다음 행을 추가하십시오.  
`com.buildforge.level = ALL`
  - b. 핸들러 섹션에서 기타 모든 레벨을 FINE에서 ALL로 변경하십시오.

변경사항을 적용하려면 Tomcat을 다시 시작하십시오.



---

## 제 9 장 관리 콘솔에서 추가 기능 구성

이 절에서는 추가 기능을 사용하거나 기본 구성에 대한 대안을 제공하도록 Build Forge®를 구성하는 방법에 대해 설명합니다.

---

### Build Forge 구성 파일(buildforge.conf)

buildforge.conf 파일은 Build Forge 제품 구성 파일입니다. 이 파일에는 Build Forge 데이터베이스를 시작하고 통신하기 위해 다른 Build Forge 컴포넌트에서 사용되는 구성 설정이 포함되어 있습니다.

buildforge.conf 파일은 두 개의 위치에 저장되며, 설치 후 수정해야 하는 경우 두 위치에서 모두 업데이트해야 합니다. 『buildforge.conf 파일 업데이트』의 내용을 참조하십시오.

- 애플리케이션 서버에서 사용되는 rbf-services.war 파일을 서비스 계층이라고도 합니다.
- 설치 루트 디렉토리에 있습니다. 다음 표에 제품에 대한 기본 또는 표준 설치 디렉토리가 나열되어 있습니다.

Windows	C:\Program Files\IBM\Build Forge
UNIX/Linux	/opt/buildforge/Platform

### buildforge.conf 파일 업데이트

buildforge.conf 파일은 두 개의 위치에 있으며, 설치 후 업데이트가 필요한 경우 두 위치에서 모두 업데이트해야 합니다.

Build Forge 데이터베이스가 다른 호스트 컴퓨터로 이동된 경우 buildforge.conf 파일을 편집하여 데이터베이스 호스트를 업데이트해야 할 수 있습니다. 이 파일을 편집하는 또 다른 일반적인 이유는 네트워크 보안 정책을 준수하기 위해 정기적으로 변경해야 하는 데이터베이스 비밀번호를 업데이트하기 위한 것입니다.

다음 프로시저를 사용하여 buildforge.conf 파일을 업데이트한 다음 buildforge.conf의 업데이트된 사본으로 rbf-services.war 파일을 다시 작성하십시오.

**참고:** buildforge.conf를 rbf-services.war로 추가해야 하는 경우에만 buildforge.conf를 편집하는 단계를 건너뛸니다.

1. Build Forge 엔진을 중지하십시오.
2. Build Forge 설치 루트 디렉토리에서 buildforge.conf 파일을 찾으십시오.

3. 텍스트 편집기를 사용하여 파일을 열고 구성 설정을 수정한 후 해당 파일을 저장하십시오.

**참고:** 이 파일을 편집하려면 루트 또는 관리자 특권이 필요합니다.

4. rbf-services.war 파일이 포함되어 있는 디렉토리로 이동하십시오. 예를 들어, 다음과 같습니다.

Apache Tomcat 서버	<bfinstall>/Apache/tomcat/webapps \$CATALINA_HOME/webapps
------------------	--

5. rbf-services.war의 백업 사본을 작성하십시오. 예를 들어, 다음과 같습니다.

Windows	copy rbf-services.war rbf-services.war.bak
UNIX/Linux	cp rbf-services.war rbf-services.war.bak

6. rbf-services 디렉토리를 삭제하십시오. 예를 들어, 다음과 같습니다.

Windows	rmdir rbf-services /s /q
UNIX/Linux	rm -rf rbf-services

7. Windows의 경우 다음 명령을 완료하십시오.

- a. classes 디렉토리를 다시 작성하십시오. 예를 들어 다음과 같습니다.

```
mkdir WEB-INF\classes
```

- b. 업데이트된 buildforge.conf 파일을 새 classes 디렉토리에 복사하십시오. 예를 들어 다음과 같습니다.

```
copy <path_to_bf_conf>\buildforge.conf WEB-INF\classes
```

- c. 다음과 같이 rbf-services.war 파일을 업데이트하십시오.

```
jar -uvf rbf-services.war WEB-INF
```

8. UNIX 및 Linux에서 다음 명령을 완료하십시오.

- a. unzip 유틸리티를 사용하여 rbf-services 디렉토리를 작성하고 채우십시오.

```
unzip -d rbf-services rbf-services.war
```

- b. 업데이트된 buildforge.conf 파일을 새 classes 디렉토리에 복사하십시오. 예를 들어 다음과 같습니다.

```
cp <path_to_bf_conf>/buildforge.conf rbf-services/WEB-INF/classes
```

- c. bf-services에서 실행 파일에 대한 권한을 설정하십시오.

```
chmod 755 <bfinstall>/server/tomcat/rbf-services/bin/*
```

9. Build Forge 엔진을 다시 시작하십시오.

Windows에서 rbf-services 파일이 재배포되고 업데이트된 buildforge.conf와 함께 rbf-services 디렉토리를 자동으로 다시 작성합니다.

## Buildforge.conf 참조

buildforge.conf 파일은 Build Forge 관리 콘솔의 실행 방법에 대한 설정을 저장합니다.

이 파일은 설치 디렉토리에 있으며, 설치 프로그램에서 자동으로 빌드됩니다. 편집해야 하는 경우 파일을 ASCII 텍스트 파일 또는 XML 파일로 저장할 수 있습니다. 구문은 다음과 같습니다.

- 한 행에서 키워드 및 값을 입력하십시오.
- 동일한 부호(공백 없음)로 키워드 및 값을 구분하십시오. 예를 들어, 다음과 같은 키워드를 입력합니다. `bf_file_storage=C:\Program Files\IBM\BuildForge\files`

키워드	값
bf_file_storage	임시 Build Forge 파일이 있는 디렉토리입니다. 예를 들어 다음과 같습니다. C:\Program Files\IBM\BuildForge\temp
bf_plugin_dir	관리 콘솔에 연결 중인 IDE 플러그인이 있는 디렉토리입니다.
birt_home	Eclipse 보고 도구(BIRT)의 파일 위치입니다.
db_database	콘솔 사용을 위해 작성한 데이터베이스의 이름입니다.
db_hostname	데이터베이스를 실행하는 컴퓨터의 호스트 이름/IP 주소입니다. db_hostname에 대한 값을 입력할 때 실제 이름 또는 IP 주소를 사용합니다. localhost의 기본값을 사용하지 마십시오.
db_password	데이터베이스 사용자 이름에 대해 작성한 비밀번호입니다.
db_provider	Build Forge를 설치하도록 선택한 데이터베이스입니다. 이 값을 편집하지 마십시오.
db_schema	데이터베이스의 스키마 이름입니다(일반적으로 db_username과 동일하게 사용되지만, 다른 스키마 이름을 선택할 수도 있음).
db_tcp_port	사용 중인 데이터베이스 연결 포트입니다.
db_type	Build Forge가 설치된 데이터베이스 유형입니다. 기본값은 odbc입니다. 이 값을 편집하지 마십시오.
db_username	데이터베이스의 사용자 이름입니다. 이는 설치 프로그램을 실행하기 전에 설정됩니다.  DB2 및 DB2 Express의 경우 DB2가 아닌 Windows에 사용자를 작성합니다.  기타 모든 데이터베이스 유형의 경우 데이터베이스 사용자 이름을 작성합니다.  37 페이지의 『데이터베이스 설정』의 내용을 참조하십시오.  이 값은 모든 콘솔 유형에 요구됩니다.
services_hostname	Build Forge 서비스 계층을 실행 중인 컴퓨터의 호스트 이름/IP 주소입니다. 이는 서비스 구성에서 완전한 도메인 이름(FQDN)입니다.
services_ssl_port	Build Forge 서비스에 안전하게 연결하기 위한 SSL 포트입니다.

키워드	값
services_tcp_port	SSL이 지정되어 있지 않은 경우 Build Forge 서비스에 연결하기 위한 TCP 포트입니다.
services_url	서비스 계층에 사용되는 포트를 지정하는 URL입니다. 예를 들어 다음과 같습니다. services_url http://mybfhost.com:8080

## 대체 포트를 사용하도록 관리 콘솔 구성

기본 포트 80이 아닌 포트에서 관리 콘솔을 실행할 수 있습니다.

### 이 태스크 정보

다음과 같은 두 가지 방법으로 관리 콘솔이 대체 포트에서 실행되도록 구성할 수 있습니다.

- 설치하는 동안 포트를 원하는 값으로 설정하십시오.
- 관리 콘솔이 이미 설치되어 있는 경우 다음 단계를 완료하십시오.
  1. httpd.conf(Windows 설치의 경우 <bfinstall>/Apache/conf/ 및 \*nix 설치의 경우 <bfinstall>/server/apache/conf/에 있음)에서 두 개 설정을 변경하십시오. 예를 들어, myHost가 로컬 컴퓨터이고 포트 81을 사용하려는 경우 다음 설정을 지정합니다.
 

```
Listen 81
ServerName myHost:81
```
  2. 콘솔을 시작하고 root 또는 관리 특권이 있는 사용자 이름으로 로그인하십시오.
  3. 관리 > 시스템을 선택하고 "콘솔 URL" 시스템 구성 설정을 관리 콘솔이 실행되는 URL 및 포트로 변경하십시오. 예를 들어 다음과 같습니다. http://myHost:81(후행 슬래시는 사용하지 않음).
  4. 엔진을 중지한 후 다시 시작하십시오.
    - Windows: 시작 > 프로그램 > **IBM Rational Build Forge 관리 콘솔 > 엔진 서비스** 중지를 클릭한 후 엔진 서비스 시작을 클릭하십시오.

Build Forge가 포그라운드에서 실행 중인 경우 Build Forge가 실행 중인 Windows 콘솔로 이동한 다음 Ctrl-C를 누르십시오.

    - UNIX 또는 Linux: rc 파일에 제공되는 스크립트를 사용하십시오.
 

```
$ /opt/buildforge/rc/buildforge start
$ /opt/buildforge/rc/buildforge stop
```

수동 명령을 사용할 수도 있습니다.

      - a. 중지하려면 프로세스 ID를 찾아 프로세스를 강제 종료하십시오.

```
$ ps aux | grep buildforge
$ kill ${<PID>}
```

- b. 시작하려면 다음 명령을 사용하십시오. 여기서 *<bfinstall>*은 설치 디렉토리의 경로입니다.

```
<bfinstall>/Platform/buildforge
```

---

## 중복 구성

여러 컴퓨터에서 Build Forge를 실행하고 모두 동일한 데이터베이스와 통신하도록 설정할 수 있습니다. 이 설정을 중복이라고 합니다.

### 중복 정보

중복은 하나의 설치에 실패한 경우 사용가능성을 증가시키고 작업 처리를 조정하는 데 도움이 됩니다.

**중요사항:** 중복은 페일오버(Failover) 기능 또는 기타 고효용 기능을 제공하지 않습니다. 단지 작업 처리 용량만 늘릴 수 있습니다. 중복 설치 중 하나가 실패하면 해당 설치에서 관리되는 모든 실행 중인 작업이 손실되지만, 나머지 설치에서는 실행 중인 작업을 계속해서 처리하고 새 작업을 허용합니다.

사용자가 작업을 시작하면 데이터베이스에 해당 작업에 대한 항목이 작성됩니다. 프로세스 엔진에서 데이터베이스를 폴링하여 새 작업이 있는지 여부를 확인합니다. 다중 프로세스 엔진이 있는 경우 엔진이 사용량이 적을 때 각각 독립적으로 폴링하므로 로드 밸런싱이 발생합니다.

중복을 설정하는 경우 일반적인 관리 콘솔 설치를 수행한 다음 추가 호스트에 관리 콘솔을 설치합니다. 모든 설치는 동일한 Build Forge 데이터베이스에 액세스하도록 구성됩니다.

**중요사항:** 모든 설치는 자체 호스트에 설치되어야 합니다. 동일한 호스트에 다중 관리 콘솔을 설치할 수 없습니다.

## 백업 시스템 설치

### 이 태스크 정보

다음 지시사항은 데이터베이스를 이미 설정하고 해당 데이터베이스를 사용할 첫 번째 관리 콘솔의 설치를 설치한 것으로 가정합니다. 다른 호스트에 추가 설치를 작성하려면 다음을 수행하십시오.

## 프로시저

1. 데이터베이스의 지시사항에 따라 호스트에 설치 전 설정을 수행하십시오. 데이터베이스에 따라 설정 시 호스트에 데이터베이스 클라이언트를 설치하고 기타 구성을 수행해야 할 수 있습니다. 35 페이지의 제 6 장 『설치 전 설정』의 내용을 참조하십시오.
2. 데이터베이스 서버가 외부 연결(TCP)을 허용하도록 구성되었는지 확인하십시오.
3. 설치 지시사항을 따르십시오. 63 페이지의 제 7 장 『관리 콘솔 설치』에서 시작할 때 다음을 수행하십시오.
  - Installation Manager를 설치하십시오(필요한 경우).
  - Installation Manager를 시작하십시오.
  - 설치를 실행하십시오.

다음 단계에는 특정 항목이 필요한 패널만 설명되어 있습니다.

4. 패키지 설치: 기능 패널에서 모든 기능이 선택되어 있는지 확인하십시오.
5. 데이터베이스 구성 패널에서 다음을 수행하십시오.
  - 데이터베이스 이름 및 스키마 이름을 제공하십시오. 이러한 이름은 첫 번째 콘솔에 지정된 이름과 동일해야 합니다.
  - 데이터베이스 사용자 이름과 비밀번호를 제공하십시오. 첫 번째 콘솔에 지정된 것과 동일한 이름 및 비밀번호를 사용하십시오.
  - 설치 시 이 데이터베이스를 채우시겠습니까?에서 아니오를 선택하십시오.

주의:

데이터 손실 위험성: 예를 선택하면 데이터베이스에서 **Build Forge** 데이터베이스 스키마를 겹쳐씹니다. 첫 번째 설치에서 **Build Forge** 오퍼레이션의 데이터가 모두 손실됩니다.

  - 테스트 수행을 클릭하십시오. 테스트에 통과하면 다음을 클릭하여 계속해서 수행할 수 있습니다.

참고: 아니오를 선택하면 JDBC 드라이버의 경로가 올바른지 검사하는 테스트만 수행됩니다.

6. 콘솔 시작 옵션 패널에서 **Build Forge** 시작 안 함 옵션이 선택되어 있습니다. 또한 이는 회색으로 표시되므로 변경할 수 없습니다. 설치 후 콘솔을 수동으로 시작해야 합니다.
7. 필요한 설치 후 구성을 수행하십시오. 213 페이지의 제 11 장 『설치 후 태스크』의 내용을 참조하십시오. 필요한 경우 데이터베이스에 데이터베이스 클라이언트를 연결할 수 있도록 해당 데이터베이스를 카탈로그합니다.

## 중복 작업

중복을 설정한 후 다음과 같은 작업을 수행하십시오.

- 첫 번째 설치의 URL에 사용자를 지정합니다.
- 사용자가 Apache 서버에 액세스하지 않도록 하려면 추가 설치 시 Apache 서버를 중지할 수 있습니다.
- HTTP 요청을 처리하는 데 사용할 용량을 늘리려면 모든 설치에서 Apache를 실행하십시오. 로드 밸런서를 설치하여 설치 간의 요청을 분배합니다.

---

## IPv6 네트워크 지원 사용

IPv6 및 혼합 IPv6/IPv4 네트워크와 함께 관리 콘솔을 구성할 수 있습니다.

### 이 태스크 정보

IPv6에 대한 관리 콘솔을 구성하려면 다음을 수행해야 합니다.

1. httpd.conf에서 서버 항목을 변경하십시오.
2. IPv6 네트워크를 사용하는 데 필요한 FLEXlm License Client를 설정하십시오.

IPv6과 함께 Build Forge를 사용할 수 있는 구성 단계 및 요구사항을 검토하십시오. 26 페이지의 『IPv6 지원에 대한 네트워킹 요구사항』의 내용을 참조하십시오.

## httpd.conf 수정

### 이 태스크 정보

IPv6을 지원하려면 컴퓨터와 네트워크가 IPv6을 지원하도록 올바르게 구성되어 있어야 합니다. 네트워크 구성 문제점은 Build Forge 시스템에서 지정된 호스트 이름과 주소가 올바르게 해석되지 않도록 합니다.

**IPv6에 대해 Build Forge를 수동으로 구성해야 합니다.** 이를 수행하려면 주 Apache 구성 파일 httpd.conf에서 항목을 수정하십시오.:

1. httpd.conf 파일(buildforge/server/apache/conf/httpd.conf)로 이동하십시오.
2. 완전한 도메인 이름(FQDN)을 참조하는 서버 이름을 추가하십시오(예: ServerName qlnx500-v6.ipv6.lexma.ibm.com).
3. 다음과 같이 0.0.0.0:80에서 청취 지시문을 수정하십시오.
  - Windows: [::]:80
  - UNIX 또는 Linux:80

## FLEXlm 클라이언트 설정

### 이 태스크 정보

이 태스크는 Build Forge 버전 7.1에서 7.1.3.3에만 적용됩니다. IPv6 네트워크에서 Build Forge를 실행하려면 다른 버전의 FLEXlm License Client가 필요합니다.

제품과 함께 설치되고 구성되는 FLEXlm 클라이언트에서 Rational License Server의 버전 7.0 및 7.1을 지원합니다.

**중요사항:** Rational License Server 7.1에서 IPv6 주소를 명시적으로 사용해야 하는 경우가 아니면 이 태스크를 완료하지 마십시오.

IPv6 버전의 FLEXlm 클라이언트를 구성하려면 다음을 수행하십시오.

1. Build Forge 엔진을 중지하십시오.
2. FLEXlm 클라이언트 소프트웨어가 설치되어 있는 디렉토리로 이동하십시오.

Windows	C:\<bfinstall>\Apache\tomcat\webapps\rbf-services\bin. 여기서 <bfinstall>은 Build Forge 설치 디렉토리입니다.  기본 설치 위치는 C:\Program Files\IBM\Build Forge입니다.
UNIX/Linux	<bfinstall>/server/tomcat/webapps/rbf-services/bin. 여기서 <bfinstall>은 Build Forge 설치 디렉토리입니다.  UNIX/Linux에서 기본 설치 디렉토리는 /opt/buildforge입니다.

3. 운영 체제에 대한 현재 FLEXlm 클라이언트 라이선스 파일의 이름을 바꿔 해당 파일을 백업하십시오. 예를 들어, 다음과 같습니다.

Windows	flexhelper.exe를 flexhelp.exe.bak로 이름을 바꿉니다.
UNIX/Linux	flexhelper-Linux-i386을 flexhelper-Linux-i386.bak로 이름을 바꿉니다.

4. 제품에서 현재 FLEXlm 클라이언트 라이선스 파일로 사용할 수 있도록 IPv6에 대한 FLEXlm 클라이언트 라이선스 파일의 이름을 바꾸십시오.

Windows	flexhelper-IPv6.exe를 flexhelper.exe로 이름을 바꿉니다.
Linux	flexhelper-Linux-i386-IPv6을 flexhelper-Linux-i386으로 이름을 바꿉니다.

5. Build Forge 엔진을 시작하십시오.

## 보안 기능

이 절에서는 Build Forge®에서 보안 기능을 사용하는 방법에 대해 설명합니다.

- 설치 중 기본적으로 사용 가능한 보안 로그인
- 싱글 사인온(SSO)
- HTTPS 및 SSL 사용



- 비밀번호 암호화 사용
- 보안 구성을 위한 `bfcclient.conf` 파일

관리 > 보안에 있는 관리 콘솔에서 선택사항의 조합 및 구성 파일의 수동 설정을 통해 이러한 기능을 사용할 수 있습니다. 이 절에는 보안 기능을 사용 가능하도록 하는 데 사용되는 구성 파일인 `bfcclient.conf`의 참조 절이 포함되어 있습니다.

**참고:** 이 절의 내용은 z/Linux에서 Build Forge를 실행하는 사용자에게는 해당되지 않습니다. z/Linux의 Build Forge에서 사용 가능한 보안 기능에 대한 정보는 92 페이지의 『System z의 Linux에 관리 콘솔 설치』의 내용을 참조하십시오.

## 싱글 사인온 구현

Build Forge와 함께 싱글 사인온 프레임워크가 제공됩니다.

싱글 사인온은 사용자가 매번 사용자 이름과 비밀번호를 입력할 필요 없이 애플리케이션에 액세스할 수 있는 인증 스키마입니다. Build Forge에서는 싱글 사인온을 구현하기 위해 써드파티 HTTP 인터셉터와의 조합에 사용할 수 있는 프레임워크를 제공합니다.

### 싱글 사인온 프레임워크 정보

Build Forge SSO 프레임워크에서는 업계에서의 여러 SSO 솔루션과 통합할 수 있는 기능을 제공합니다. SSO 프레임워크는 인터셉터 기반이므로, HTTP 요청을 인터셉트하고 해당 요청을 처리할 수 있는 메소드를 제공합니다. HTTP 요청에서 보안 아티팩트를 받고 유효성을 검증할 수 있도록 사용자 정의 인터셉터를 쓸 수 있습니다. 특히, 인터셉터에서 HTTP 응답에 토큰을 설정한 다음 연속 요청에서 해당 토큰을 찾을 수 있습니다.

다음과 같은 두 가지 SSO 솔루션이 Build Forge와 함께 제공됩니다.

- SPNEGO(Simple and Protected Negotiation Protocol)의 인터셉터입니다. 112 페이지의 『Active Directory 도메인에서 SPNEGO를 사용하여 싱글 사인온 구현』의 내용을 참조하십시오.
- WebSphere SSO와의 통합을 위한 인터셉터입니다. 118 페이지의 『사용자 정의 인터셉터를 사용하여 WAS 보안과 통합』의 내용을 참조하십시오.

**SSO 프레임워크 메소드:** SSO 인터셉터는 Build Forge SSO 프레임워크에서 사용되는 인터페이스를 구현하는 Java 클래스입니다.

```
com.buildforge.services.server.sso.ISSOIntercaptor
```

이는 서비스 계층 컴포넌트에 있습니다.

```
<bfinstall>/Apache/tomcat/webapps/rbf-services/WEB-INF/classes
```

인터페이스에서 다음 메소드를 제공합니다.

### **initInterceptor**

인터셉터가 로드되면 호출됩니다. 구성 특성의 맵이 `initInterceptor()` 메소드에 전달됩니다. 구성 특성이 **관리 > 보안 > SSO**의 Build Forge 콘솔에서 작성됩니다.

### **isTargetInterceptor**

인바운드 요청에서 속성을 검토하여 해당 속성에서 이 인터셉터를 작동해야 하는지 여부를 판별합니다. 작동해야 하는 경우 인터셉터에서 `authenticateRequest()` 메소드가 있는 요청을 인증해야 합니다. 그렇지 않으면 이 인터셉터를 건너뛵니다. 인터셉터 선택 시 실행 중에 다중 인터셉터가 구성되었다고 가정합니다. 해당 인터셉터가 순서대로 처리됩니다.

### **authenticateRequest**

요청에 있는 데이터를 사용하여 요청을 인증합니다. 이는 응답 속성을 사용하여 클라이언트에 데이터를 다시 보냅니다.

### **logoutRequest**

요청이 처리된 후 사용자 관련 보안 정보를 정리합니다.

**인터셉터 구성 및 순서 지정:** 인터셉터 구성은 **관리 > 보안 > SSO**에 정의되어 있습니다. 다음 구성이 Build Forge와 함께 제공됩니다.

- 양식 SSO 인터셉터 - 기본적으로 활성 상태이며, 단일 로그인 양식을 구현합니다.
- SPNEGO SSO 인터셉터 - 기본적으로 비활성 상태이며, SPNEGO를 구현하여 인증을 수행합니다.

인터셉터 클래스를 구현하고 해당 인터셉터 클래스를 Build Forge Apache Tomcat 애플리케이션 서버에 배치한 후 해당 위치에 새 SSO 구성을 구성합니다. 클래스는 하나의 SSO 구성 특성입니다.

이 목록의 순서는 요청을 처리하기 위해 인터셉터가 참조되는 순서를 결정합니다. 다중 인터셉터를 구성하여 요청을 처리할 수 있습니다. 로그인 중에 각 인터셉터가 순서대로 참조됩니다. 요청을 처리하는 인터셉터는 속성이 요청에서의 속성에 적합한 첫 번째 활성 인터셉터입니다. 하나의 인터셉터에서만 요청을 처리합니다. 이러한 인터셉터는 항상 `isTargetInterceptor`에 대해 `true`를 응답하는 첫 번째 인터셉터입니다.

**참고:** 오류가 발생하는 경우 대체를 제공하려면 양식 SSO 인터셉터를 활성 상태로 두어야 합니다. 대체를 제공하려면 먼저 사용자 정의 인터셉터가 목록에 있어야 합니다.

**사용자 정의 SSO 인터셉터 추가:** Build Forge에서 사용자 정의 인터셉터를 작성하려면 다음을 수행하십시오.

1. 사용자 정의 Java 클래스를 작성하십시오.

클래스가 `ISSOInterceptor` 인터페이스를 구현해야 합니다.

2. 서비스 계층 컴포넌트 WAR에 사용자 정의 클래스를 배치하십시오.
  - a. 컴파일된 SSO 인터셉터 클래스를 포함하는 JAR 파일을 작성하십시오.
  - b. 다음 위치에서 Build Forge 서비스 계층 컴포넌트에 JAR 파일을 복사하십시오.  
`<bfinstall>/Apache/tomcat/webapps/rbf-services/WEB-INF/classes`
  - c. 이 디렉토리에서 JAR 파일의 압축을 해제하십시오. SSOManager가 이 위치에서 로드할 클래스를 찾습니다.
  - d. Build Forge를 다시 시작하십시오.
3. 선택사항: 환경을 정의하십시오. 이 환경은 `initInterceptor()` 메소드에 특성 오브젝트로 전달될 수 있습니다.
  - a. 관리 콘솔에서 환경으로 이동하십시오.
  - b. 환경 추가를 클릭하십시오.
  - c. 초기화하기 위해 SSO 인터셉터에 필요한 모든 특성을 정의하십시오.
4. Build Forge에 SSO 인터셉터를 추가하십시오.
  - a. 관리 콘솔에서 관리 > 보안 > SSO로 이동하십시오.
  - b. SSO 구성 추가를 클릭하십시오. 다음과 같은 해당 특성을 입력하십시오.
    - 이름 - SSO 구성의 이름을 입력합니다.
    - 활성 - 예로 설정합니다. 인증 요청 중에 모든 활성 구성에 액세스됩니다. 해당 구성은 이 패널에 표시되는 순서대로 액세스됩니다.
    - Java 클래스 - 클래스의 전체 패키지 이름을 입력합니다. 지정된 클래스는 하나의 SSO 인터셉터에만 지정할 수 있습니다.
    - 환경 - 이 SSO 인터셉터에 사용할 환경을 정의한 경우 해당 환경을 선택합니다.
  - c. 저장을 클릭하십시오.

이제 SSO 인터셉터가 목록에 표시됩니다.

5. SSO 구성의 순서를 지정하십시오. SSO 인터셉터의 왼쪽에 있는 아이콘을 클릭한 후 맨 위로 이동을 선택하십시오.

요청하는 동안 이 패널에 표시되는 순서대로 활성 SSO 구성에 액세스됩니다. 사용자 구성은 기본적으로 활성 상태이고, 액세스할 때 항상 `true`를 리턴하므로 해당 구성을 양식 SSO 구성 앞에 배치해야 합니다. 기본적으로 SPNEGO SSO 구성은 비활성 상태입니다.

**예제 authenticateRequest 구현:** 다음 예제는 Build Forge와 WebSphere 보안을 통합하는 데 사용되는 WebSphere SSO 인터셉터의 일부입니다.

인터셉터에서는 리플렉션을 사용하여 WebSphere 클래스 WSSubject를 찾습니다. 클래스에 AuthServlet에 로그인하는 데 사용되는 프린시פל을 리턴하는 getCallerPrincipal 메소드가 들어 있습니다. WAS에서 인증하기 전에 AuthServlet을 보호해야 합니다.

더 많은 정보를 리턴할 수 있는 기타 메소드를 사용할 수 있습니다. 모든 애플리케이션 서버에 대해 작업하는 데 유사한 메소드를 사용할 수 있습니다.

```
public Result authenticateRequest
    (Request requestAttributes, Response responseAttributes)
    throws SSOException {

    Result result = null;

    try {
        Class<?> c1 =
            Class.forName("com.ibm.websphere.security.auth.WSSubject");
        Method theMethod = c1.getMethod("getCallerPrincipal",
            (Class[])null);
        String principal = (String)theMethod.invoke((Object[])null,
            (Object[])null);

        if (principal != null
            && principal.length() > 0
            && !principal.equals("UNAUTHENTICATED")) {
            result = new Result(Result.UseridOnlyOID, domain, principal);
            responseAttributes.setStatus(HttpServletResponse.SC_OK);
        } catch (Exception e) {
            throw new SSOException(e);
        }

        return result;
    }
}
```

authenticateRequest를 구현하는 동안 리턴하기 전에 응답 상태를 설정해야 합니다.

- 리디렉션을 수행할 필요가 없고 찾은 정보가 올바른 경우 다음을 리턴할 수 있습니다.

```
responseAttributes.setStatus(HttpServletResponse.SC_OK);
```

- 요청에 계속해서 올바른 로그인을 수행하는 데 필요한 정보가 부족한 경우 다음을 리턴합니다.

```
responseAttributes.setStatus(HttpServletResponse.SC_FORBIDDEN);
```

- 리디렉션을 수행하여 추가 정보를 수집해야 하는 경우 다음을 리턴합니다.

```
responseAttributes.setStatus(HttpServletResponse.SC_MOVED_TEMPORARILY);
responseAttributes.sendRedirect(url);
```

사용할 수 있는 추가 상태 값이 있습니다. HttpServletResponse의 JavaDoc를 참조하십시오.

**로그인 오류 복구:** 테스트할 때 사용자 정의 인터셉터가 올바르게 작동되지 않는 경우 인증에 문제가 있을 수 있습니다. 다음 정보가 포함된 오류 페이지가 표시됩니다.

## Build Forge Error

Access is denied to the Buil Forge console

"Error authenticating:  
com.buildforge.services.common.api.APIException - API:  
Authentication Error."

Please click [here](#) to try the same type of login again  
or click [here](#) to force a form login (user ID/password).

복구에 대한 다음과 같은 두 가지 옵션이 있습니다.

- 로그인을 재시도합니다. 이렇게 하면 구성된 인터셉터 목록을 동일한 방식으로 다시 수행합니다.
- 양식 로그인을 강제 실행합니다. 이렇게 하면 사용자 정의 인터셉터가 생략되고 양식 로그인 페이지가 사용됩니다.

**메소드 소스 목록:** 다음 주석 및 소스 목록은 ISSOInterceptor 인터페이스에서의 메소드에 대한 자세한 정보를 제공합니다.

### initInterceptor

```
/**
 * This method is called when the interceptor is loaded. A map of the
 * configuration properties is passed into the init method. You can create
 * the configuration properties from a BuildForge Environment and associate
 * it with the SSO configuration.
 *
 * @param initializationProps used to configure the implementation
 * @return true if successful, false if an error should be reported.
 * @throws SSOException if the initialization fails
 */
public boolean initInterceptor (Properties initializationProps) throws SSOException;
```

### isTargetInterceptor

```
/**
 * This methods will review the attributes in the requestAttributes Map
 * to determine if there is something that this interceptor should
 * act on. If the interceptor return is "true", then the interceptor will
 * be responsible for authenticating the request and the authenticateRequest
 * method is invoked. If the interceptor return is "false", then this
 * interceptor is skipped and the next isTargetInterceptor in the list will
 * be called. Ordering of the interceptors during the configuration will
 * return which interceptor has the first shot at authenticating a request.
 *
 * @param requestAttributes attributes found in the inbound request
 * @return true if this interceptor will authenticate the request,
 *         false if it will not.
 * @throws SSOException
 */
public boolean isTargetInterceptor(Request requestAttributes) throws SSOException;
```

### authenticateRequest

```
/**
```

```

* This method is called on an interceptor that returns true for the
  isTargetInterceptor method. The Request will contain data used
  to perform the authentication. The Response is for the interceptor
  to send information back to the client. The Result returned will contain
  the following information if the status code is 200:
*
* OID: an object identifier of the SecurityContext that can process token
  information stored in this map when going to an Agent.
* Domain: a valid BF domain name or <default> if not known
  (the username must be valid in the configured realm).
* Username: a valid BF username. This will be used to lookup BFUser attributes
  that are used in checking authorization policy.
* @see com.buildforge.services.common.security.context.Result
*
* @param requestAttributes attributes found in the inbound request
* @param responseAttributes sent back in the outbound response
* @return com.buildforge.services.common.security.context.Result - result
  information that tells BF how to handle the authentication request.
* @throws com.buildforge.services.server.sso.SSOException
**/
public Result authenticateRequest(
    Request requestAttributes,
    Response responseAttributes)
    throws SSOException;

```

### logoutRequest

```

/**
* This method is called to logout a request. The first interceptor that
  returns true for the isTargetInterceptor method will perform the logout.
  The main point is to clean up any user-related security information that
  should not be kept. The interceptor can inspect the request and response
  objects to determine what needs to be removed.
*
* @param requestAttributes attributes found in the inbound request
* @param responseAttributes sent back in the outbound response
* @return boolean - true if request redirect to exit page,
  false if redirect to login page.
* @throws com.buildforge.services.server.sso.SSOException
**/
public boolean logoutRequest(
    Request requestAttributes,
    Response responseAttributes)
    throws SSOException;

```

## Active Directory 도메인에서 SPNEGO를 사용하여 싱글 사인온 구현

Active Directory 도메인에서 싱글 사인온을 구현할 수 있도록 SPNEGO(Simple and Protected GSS-API Negotiation) 메커니즘이 제공됩니다.

### 시작하기 전에

이 태스크에서는 네트워크에 다음 요소가 필요합니다.

- Active Directory 도메인
- 디렉토리 서버 호스트 이름

- Kerberos 키 분배 센터(KDC) 호스트 이름
- Active Directory 도메인의 호스트에 Build Forge 설치
- Active Directory 도메인의 클라이언트 호스트
- 각 클라이언트의 Kerberos 구성 파일
- 디렉토리 서버 호스트에 설치된 Windows Server 2003 SP2 자원 톨킷
- 지원되는 웹 브라우저

참고: Internet Explorer 6는 SPNEGO에 사용할 수 없습니다. 지원되는 브라우저를 사용합니다.

다음 프로시저에는 다음 설정을 기반으로 하는 예제가 포함되어 있습니다.

- mycompany.com은 도메인의 모든 호스트에서 사용되는 TCP/IP 도메인의 이름입니다.
- ITDEV.COM은 Active Directory 도메인의 이름입니다.
- it\_directory.mycompany.com은 디렉토리 서버가 실행되는 호스트입니다. 또한 Kerberos KDC도 실행됩니다.
- it\_domain.mycompany.com은 Active Directory 도메인 제어가 실행되는 호스트입니다.
- it\_buildforge.mycompany.com은 Build Forge가 설치된 호스트입니다.
- bfuser는 Build Forge 시스템의 도메인 사용자 이름입니다.
- happy\_user는 웹 브라우저에서 SSO를 사용하여 Build Forge에 액세스하는 예제 사용자의 도메인 사용자 이름입니다.

## 이 태스크 정보

다음 태스크를 수행하여 Active Directory 도메인 및 KDC에서 SPNEGO SSO를 구현합니다. 각각 자세한 프로시저가 포함된 섹션이 제공됩니다.

참고: Active Directory가 아닌 KDC와 함께 SPNEGO 인터셉터를 사용할 수 있습니다.

## 프로시저

1. Active Directory 사용자 및 서비스 프린시פל을 설정하십시오.
2. Kerberos 파일을 설정하십시오.
3. Active Directory 및 SPNEGO를 사용하도록 Build Forge를 구성하십시오.
4. 보안 액세스에 대한 브라우저 클라이언트를 구성하십시오.
5. SSO를 통해 Build Forge에 액세스하십시오.

## Active Directory 사용자 및 서비스 프린시פל 설정:



Active Directory 도메인에 Build Forge 서버와 Build Forge 클라이언트를 설정해야 합니다.

### 시작하기 전에

다음 프로시저에는 Windows 2003 SP2의 지원 도구가 필요합니다. 이러한 지원 도구에는 Active Directory에서 서비스 프린시פל을 설정하는 데 필요한 `setspn` 명령이 포함됩니다. Windows Server 2003 제품 CD 또는 Microsoft 다운로드 센터에서 지원 도구를 설치합니다.

### 이 태스크 정보

Active Directory에 Build Forge 클라이언트와 서버가 있으면 사용자가 Windows 호스트에 로그인할 때 Kerberos 신임 토큰을 생성합니다. 그런 다음 사용자가 Build Forge 서버에 액세스하려고 시도하면 SPNEGO 인터셉터에서 사용자 토큰을 수신하고 유효성 검증합니다. 유효성 검증된 ID는 구성된 Microsoft Active Directory LDAP 서버를 통해 로그인을 수행할 수 있도록 Build Forge에 전달됩니다.

### 프로시저

1. 도메인 제어기 호스트에 로그인하십시오. 예제에서 호스트는 `it_example.mycompany.com`입니다.
2. 멤버가 아닌 경우 Build Forge 호스트를 Active Directory 도메인에 추가하십시오. 이 예제에서는 `it_buildforge` 호스트를 `ITDEV.COM` 도메인에 추가합니다. 이제 호스트에서 도메인에 `it_buildforge.ITDEV.COM`과 같은 완전한 이름이 사용됩니다.
3. Active Directory 도메인에 Build Forge 사용자를 추가하십시오. 이 예제에서는 `bfuser` 사용자를 작성합니다.

#### 중요사항:

- 비밀번호 사용 기간 제한 없음을 선택합니다. 다른 비밀번호 관리를 선택할 수도 있습니다. 그러나 비밀번호가 만기될 때마다 Build Forge 서버에 대한 새 비밀번호를 입력해야 합니다.
  - 계정 탭에서 위임에 대해 계정 신뢰할 수 있음을 선택합니다.
4. 사용자 계정이 없는 경우 Microsoft Active Directory에 모든 클라이언트에 대한 사용자 계정을 작성하십시오. 이 예제에서는 `happy_user`와 같이 한 명의 사용자를 작성합니다.
  5. Build Forge에 대한 서비스 프린시פל 이름(SPN)을 작성하십시오. 예제에서는 Build Forge 서버에 대한 SPN `it_buildforge`를 작성하기 위해 Active Directory 사용자 `bfuser`가 서비스 이름 `HTTP/it_buildforge.mycompany.com`과 연관되어 있습니다.

```
setspn -A HTTP/it_buildforge.mycompany.com bfuser
```



HTTP는 Build Forge 서비스의 서비스 이름입니다.

### Kerberos 인증에 대한 파일 설정:

Build Forge 호스트에 시작 파일(Kerberos 클라이언트 구성 파일) 및 키탭(keytab) 파일을 설정해야 합니다.

#### 프로시저

1. Build Forge가 실행되는 호스트에 시작 파일을 설정하십시오.

- Windows 시스템:
  - krb.ini 파일의 이름을 지정하고 C:\winnt에 배치하십시오. C:\winnt가 없는 경우 작성하십시오.
  - default\_keytab\_name을 FILE:C:\winnt\krb5.keytab으로 설정하십시오.
- UNIX 및 Linux 시스템:
  - krb.conf 파일의 이름을 지정하고 C:\winnt에 배치하십시오.
  - default\_keytab\_name을 FILE:/etc/krb5.keytab으로 설정하십시오.

예제 시스템의 도메인 및 범주 설정을 사용하여 Windows에 다음과 같은 예제 파일이 설정됩니다.

```
[libdefaults]
default_realm = ITDEV.COM
default_keytab_name = FILE:C:\winnt\krb5.keytab
default_tkt_enctypes = rc4_hmac
default_tgs_enctypes = rc4_hmac
# kdc_default_options = 0x40800000
forwardable = true
renewable = true
noaddresses = true
clockskew = 300
[realms]
ITDEV.COM = {
    kdc = it_directory.itdev.com:88
    default_domain = mycompany.com
[domain_realm]
.mycompany.com = ITDEV.COM
```

**참고:** 클라이언트 호스트와 Build Forge 서버 호스트 간의 클럭 편차가 300초 이상인 경우에는 토큰이 작동되지 않습니다. 시간, 날짜 및 시간대를 클라이언트와 서버 호스트의 편차 한계 내로 설정합니다.

2. Kerberos 키탭(keytab) 파일을 설정하십시오. Build Forge 서버에서 클라이언트가 Build Forge 서버 URL에 액세스하려고 시도할 때 Kerberos 토큰을 유효성 검증하기 위해 키탭(keytab) 파일을 사용합니다. 도메인 제어기에서 ktpass 명령을 사용하여 파일을 작성합니다. 전제조건 Windows 자원 툴킷에 ktpass 명령이 포함되어 있습니다. 다음 예제에서는 예제 시나리오에서 Build Forge에 대해 설정된 Active Directory 사용자 이름 및 Build Forge 서비스의 프린시펄 이름을 사용합니다.

니다. -pass Rational의 자체 비밀번호를 대체합니다. 쉽게 구별할 수 있도록 예제에 행 바꾸기가 표시됩니다. ktpass 명령에는 행 바꾸기를 사용하지 마십시오.

```
ktpass -out C:\it_buildforge.keytab  
-princ HTTP/it_buildforge.mycompany.com@ITDEV.COM  
-mapuser bfuser -mapop set  
-pass Rational /crypto RC4-HMAC-NT /rndpass /ptype KRB5_NT_SRV_HST
```

it\_buildforge.keytab to krb5.keytab의 이름을 바꾸고 Kerberos 시작 파일이 포함되어 있는 디렉토리의 Build Forge 호스트에 배치합니다.

- Windows: C:\winnt\
- UNIX 및 Linux: /etc

## Active Directory 및 SPNEGO를 사용하도록 Build Forge 구성:

### 프로시저

1. Build Forge에서 Active Directory 도메인 제어를 지정하도록 LDAP를 설정하십시오.
  - a. Build Forge에서 **관리 > LDAP**을 클릭하십시오.
  - b. 새 LDAP 구성을 작성하고 다음과 같이 특성을 설정하여 도메인 제어기에 대한 액세스를 설정하십시오.
    - 이름: Active Directory 도메인의 이름으로 설정합니다. 예제 환경에서 이 이름은 itdev입니다.
    - 관리자 DN: 도메인에서 관리자 사용자로 설정합니다.
    - 액세스 그룹 맵핑: 아니오
    - 호스트: 도메인 제어기 호스트의 IP 주소로 설정합니다.
    - 사용자 계정 바인드: 예
    - 프로토콜: LDAP
    - 표시 이름: displayname
    - 식별 이름: distinguishedname
    - 그룹 이름: memberof
    - 메일 이름: displayname
    - 검색 기준: on=users,do=domainname,do=domainextension입니다. 예제 환경에서는 on=users,do=itdev,do=.com입니다.
    - 고유 ID: sAMAccountNames=%
  - c. 기본값으로 설정을 클릭하십시오. 이 구성은 기본 LDAP 구성이어야 합니다.
2. SPNEGO에 대한 Build Forge 환경 변수를 설정하십시오.
  - a. Build Forge에서 **환경 > SPNEGO SSO**의 환경으로 이동하십시오.

- b. bf\_spnego\_service\_name을 HTTP로 설정하십시오. 이는 서비스 프린시플 이름과 일치합니다.
  - c. bf\_spnego\_server\_name을 Build Forge 서버 호스트의 완전한 호스트 이름인 it\_buildforge.mycompany.com으로 설정하십시오. 이 변수가 설정되어 있지 않으면 INetAddress API가 호스트 이름을 찾으려 시도합니다.
  - d. bf\_spnego\_realm을 Kerberos 범주 이름인 ITDEV.COM으로 설정하십시오. 이 변수가 설정되어 있지 않으면 Kerberos 시작 파일에 있는 값이 사용됩니다.
3. SPNEGO 인터셉터를 사용 가능하도록 설정하십시오.
- a. Build Forge에서 관리 > 보안 > SSO > SPNEGO SSO 인터셉터로 이동하십시오.
  - b. 활성 특성을 예로 설정한 다음 저장을 클릭하십시오.
  - c. 관리 > 보안 > SSO에서 SPNEGO SSO 인터셉터를 목록의 맨 위로 이동하십시오. SPNEGO SSO 인터셉터에 대한 SSO 옵션 메뉴에서 맨 위로 이동을 선택한 다음 저장을 클릭합니다.

### SSO에 대한 클라이언트 브라우저 구성:

클라이언트 브라우저에는 SPNEGO를 사용하기 위한 보안 설정이 설정되어 있어야 합니다.

#### 이 태스크 정보

Microsoft Internet Explorer 또는 Mozilla Firefox와 같이 Build Forge에 액세스하는 데 사용된 브라우저에 클라이언트 설정 지시사항을 사용합니다.

#### 프로시저

- Internet Explorer의 경우 다음을 수행하십시오.
  1. Active Directory 도메인에 로그인하십시오. 예제 구성에서는 itdev.com에 로그인합니다.
  2. Internet Explorer를 여십시오.
  3. 도구 > 인터넷 옵션 > 보안을 클릭하십시오.
  4. 로컬 인트라넷을 선택한 다음 사이트를 클릭하십시오.
  5. 로컬 인트라넷 대화 상자에서 다른 영역에 없는 로컬(인트라넷) 사이트를 모두 포함을 선택한 다음 고급을 클릭하십시오.
  6. Build Forge가 실행되는 호스트를 웹 사이트 목록에 추가한 다음 확인을 클릭하십시오.
  7. 확인을 클릭하여 로컬 인트라넷 대화 상자를 닫으십시오.
  8. 인터넷 옵션 창에서 고급 탭을 클릭하십시오.

9. 보안 그룹으로 스크롤하고 통합된 **Windows** 인증 사용(다시 시작해야 함)이 선택되어 있지 않은 경우 이를 선택하십시오.
  10. 확인을 클릭하십시오.
  11. Internet Explorer를 다시 시작하십시오.
- Mozilla Firefox의 경우 다음을 수행하십시오.
    1. Active Directory 도메인에 로그인하십시오. 예제 구성에서는 itdev.com에 로그인합니다.
    2. Firefox를 여십시오.
    3. 주소 필드에 **about:config**를 입력하십시오.
    4. 필터 상자에 **network.n**을 입력하십시오. 목록이 업데이트됩니다.
    5. **network.negotiate-auth.trusted-uris**를 두 번 클릭하십시오. 신뢰할 수 있는 도메인 목록을 입력합니다. 여기에는 디렉토리 서버 호스트 및 Build Forge 서버 호스트(예제에서 it\_directory.mycompany.com 및 it\_buildforge.mycompany.com)가 포함되어 있어야 합니다. 확인을 클릭하십시오.
    6. 위임을 설정하십시오. **network.negotiate-auth.delegation-uris**를 두 번 클릭하고 브라우저에서 사용자 인증을 위임할 수 있는 사이트 목록을 입력합니다.

### SSO를 통해 Build Forge에 액세스하십시오.:

SSO를 통해 로그인을 테스트할 서버 URL을 입력합니다.

#### 프로시저

1. 사용자의 Active Directory 목록에 있는 사용자 이름을 사용하여 Active Directory 도메인에 있는 호스트에 로그인하십시오.
2. 브라우저를 여십시오.
3. Build Forge 서버 호스트의 URL을 입력하십시오. 예제 구성 파일에서는 http://it\_buildforge.mycompany.com을 사용합니다. SSO가 올바르게 구성된 경우 Build Forge 관리 콘솔이 표시됩니다.
4. Build Forge 콘솔의 오른쪽 상단에 표시된 사용자 이름이 클라이언트의 Windows 로그인 이름과 일치하는지 확인하십시오.

### 사용자 정의 인터셉터를 사용하여 WAS 보안과 통합

이 절에서는 SSO 인터셉터를 작성하여 WebSphere Application Server(WAS) 보안과 통합하는 방법에 대해 설명합니다.

## 시작하기 전에

제공된 양식 SSO 인터셉터가 양식 기반 로그인 페이지에서 사용자를 인증합니다. 다음은 사용자 정의 SSO 인터셉터를 작성하는 방법의 예제입니다. 사용자 정의 인터셉터가 사용자 정의 인터셉터 클래스를 사용합니다.

인터셉터 클래스에서 인증된 사용자 신임을 얻기 위해 WAS에 액세스합니다. 이러한 신임 정보를 얻으면 해당 신임 정보가 캐시됩니다. 후속 로그인에서 캐시된 신임을 사용합니다.

**전제조건:** 사용자가 LDAP 사용자 신임과 함께 WAS에 구성되어야 합니다.

**참고:** WAS 사용자가 포함되어 있는 Build Forge LDAP 도메인을 "기본" LDAP 서버로 설정해야 합니다. 이를 수행하려면 이 Build Forge LDAP 도메인으로 이동하고 "기본값으로 설정"을 선택합니다.

### 권한 서비스(AuthServlet) 보호:

일반적으로 Build Forge에서는 해당 서비스를 제공된 Apache Tomcat 애플리케이션 서버의 애플리케이션으로 실행합니다.

### 이 태스크 정보

다음 지시사항은 Tomcat 대신 WAS를 사용하도록 Build Forge를 구성합니다. 615 페이지의 『Apache Tomcat 대신 WebSphere Application Server 사용』 절의 지시사항을 수행하십시오. 단, rbf-services.war 파일에 보안 제한조건을 추가하기 위해 추출하여 수정해야 하는 web.xml이라는 파일이 포함된 경우는 예외입니다. WAS에서 이 애플리케이션을 설치하기 전에 이 파일의 보호 버전을 사용하도록 수정한 후 war 파일을 재생성해야 합니다.

이를 수행하려면 다음 지시사항을 참조하십시오.

### 프로시저

1. rbf-services.war 파일이 들어 있는 디렉토리로 이동하십시오(Tomcat 서버 루트에 있는 webapps 디렉토리). 이 파일을 C:\rbf와 같은 임시 위치에 복사하십시오.
2. WAR 파일을 펼치십시오. 명령행에서 %IBM\_JAVA\_HOME%\jar -xvf rbf-services.war 명령을 실행하여 war 파일의 콘텐츠를 펼칩니다.

**참고:** Java가 사용 가능하고 IBM\_JAVA\_HOME 환경 변수가 이미 작성되어 있어야 합니다.

3. 나중에 검색하려면 rbf-services.war 파일을 저장하십시오.
  - a. Windows: copy rbf-services.war rbf-services.war.bak
  - b. UNIX 또는 Linux: cp rbf-services.war rbf-services.war.bak

4. WEB-INF 디렉토리에서 web.xml 파일(war 파일에서 펼쳐진 파일)을 찾으십시오. 이 파일을 편집하여 보안 컨텍스트를 추가하십시오. 예를 들어, </web-app> 태그 바로 앞의 파일 끝 부분에 다음 행을 추가합니다.

```
<security-constraint id="SecurityConstraint_1">
    <web-resource-collection id="WebResourceCollection_1">
        <web-resource-name>*/</web-resource-name>
        <url-pattern>/AuthServlet/*</url-pattern>
        <http-method>GET</http-method>
        <http-method>POST</http-method>
        <http-method>PUT</http-method>
        <http-method>DELETE</http-method>
    </web-resource-collection>
    <auth-constraint id="AuthConstraint_1">
        <description>myconstraint:+:</description>
        <role-name>User</role-name>
    </auth-constraint>
    <user-data-constraint id="UserDataConstraint_1">
        <transport-guarantee>NONE</transport-guarantee>
    </user-data-constraint>
</security-constraint>
<login-config id="LoginConfig_1">
    <auth-method>BASIC</auth-method>
    <realm-name>full-qualified-domain</realm-name>
</login-config>
<security-role id="SecurityRole_1">
    <role-name>User</role-name>
</security-role>
```

참고: <auth-method>는 WAS에서 지원되는 모든 J2EE 인증 메소드일 수 있습니다. 가장 공통적인 인증 메소드는 추가 구성 매개변수가 필요한 FORM입니다. 애플리케이션에서의 FORM 구성에 대한 지시사항은 WebSphere 문서를 참조하십시오.

5. 명령행에서 다음 명령을 실행하여 war 파일(파일이 추출된 동일한 디렉토리에서 호출됨)을 재생성하십시오.

```
%IBM_JAVA_HOME%\jar -cvf rbf-services.war
```

이제 J2EE 제한조건과 함께 AuthServlet을 보호하도록 수정된 새 버전의 rbf-services를 사용할 수 있습니다. WAS에서의 실행을 위한 설치 지시사항을 완료하고 애플리케이션->새 애플리케이션 설치를 통해 이 버전의 rbf-services.war을 설치합니다. WAS를 구성하는 동안 Build Forge가 실행되고 있지 않은지 확인하십시오.

설치한 후 애플리케이션->엔터프라이즈 애플리케이션으로 이동합니다. 애플리케이션 이름을 클릭하여 구성합니다. 세부사항 특성에서 사용자/그룹에 대한 보안 역할 맵핑이라는 제목의 링크를 클릭합니다. 사용자의 경우 모두 인증됨 선택란을 선택합니다. 내용을 변경한 후 마스터 구성을 저장해야 합니다.

또한 WAS에서 애플리케이션 보안을 사용할 수 있어야 합니다. 이를 수행하려면 보안->보안 관리, 애플리케이션 및 인프라스트럭처로 이동합니다. 애플리케이션 보안 사용이 선택되어 있는지 확인하십시오.

이 때, WAS 서버를 다시 시작한 후 Build Forge 서버를 다시 시작하십시오.

참고: Build Forge 전에 WAS를 시작해야 하므로 rbf-services가 WAS 시작 프로세스의 일부로 시작됩니다.

## 새 SSO 구성 작성:

새 SSO 구성을 작성하여 인터셉터를 사용하십시오.

### 프로시저

1. Build Forge 콘솔에서 관리->보안->SSO로 이동하십시오.
2. **SSO** 구성 추가를 클릭하십시오.
3. 구성에 대한 특성을 설정하십시오.
  - 이름 - 이 구성의 이름을 입력합니다.
  - **Java** 클래스 -  
`com.buildforge.services.server.sso.was.WebSphereSSOInterceptor`를 입력합니다.
  - 활성 - 예를 선택합니다.
4. 저장을 클릭하십시오.
5. 목록에서 처음 표시되도록 이 구성을 이동하십시오. 구성 이름의 왼쪽에 있는 메뉴에서 맨 위로 이동을 선택합니다.

## LDAP 사용자 또는 LDAP 그룹을 로컬 사용자에게 맵핑:

LDAP 사용자 또는 LDAP 그룹을 로컬 사용자(예: 로컬 루트 사용자)에 맵핑합니다.

### 이 태스크 정보

WebSphere SSO 인터셉터를 작성한 후 인터셉터를 구성하여 다음과 같은 작업을 수행할 수 있습니다.

- LDAP 사용자를 로컬 사용자에게 맵핑(루트 포함)
- LDAP 그룹을 루트와 같은 로컬 사용자에게 맵핑

맵핑을 설정하려면 환경을 정의한 후 인터셉터에서 해당 환경을 참조하십시오.

### 환경 정의:



## 프로시저

1. Build Forge 콘솔에서 **환경**을 클릭하십시오.
2. 환경의 이름을 지정하고 **환경 저장**을 클릭하십시오.
3. LDAP 사용자를 로컬 사용자에게 매핑하려면 다음 변수를 설정하고 저장하십시오.
  - a. `ldap_user_list_mapping`을  
`LDAP_user1|local_user1;LDAP_user2|local_user2;LDAP_user3|local_user3`  
으로 설정
  - b. `ldap_realm_name`을 `LDAP_domain_name`으로 설정

예를 들어, 다음 환경이 설정된 것으로 가정합니다.

```
ldap_user_list_mapping = user1|root;user2|root;user3|root
ldap_realm_name = bluepages.ibm.com:389
```

이 경우 LDAP user1, user2 및 user3이 모두 로컬 루트 사용자에게 매핑됩니다.

4. 특정 LDAP 그룹의 구성원을 로컬 사용자에게 매핑하려면 다음 변수를 설정하고 저장하십시오.
  - a. `ldap_group_name_mapping`을 `LDAP_group|local_user`로 설정
  - b. `ldap_realm_name`을 `LDAP_domain_name`으로 설정

**참고:** 두 가지 유형의 매핑을 모두 사용하는 경우 사용자 매핑이 그룹 매핑보다 우선적으로 적용됩니다.

그룹 매핑의 예로 다음 환경이 설정된 것으로 가정합니다.

```
ldap_group_name_mapping = cn=bf_admin,ou=memberlist,ou=ibmgroups,o=ibm.com|root
ldap_realm_name = bluepages.ibm.com:389
```

이 경우 LDAP 그룹 `cn=bf_admin,ou=memberlist,ou=ibmgroups,o=ibm.com`의 구성원이 로컬 루트 사용자에게 매핑됩니다.

## 환경 참조:

### 프로시저

1. Build Forge 콘솔에서 **관리->보안->SSO**를 클릭하십시오.
2. WebSphere SSO 인터셉터의 이름을 클릭하십시오.
3. 구성의 **환경** 특성을 방금 정의한 환경으로 설정하십시오.
4. **저장**을 클릭하십시오.
5. 목록에서 처음 표시되도록 이 구성을 이동하십시오. 구성 이름의 왼쪽에 있는 메뉴에서 맨 위로 이동을 선택합니다.

### SSO 사용자 정의 인터셉터 실행:

이제 새 구성을 사용하여 로그인할 수 있습니다.



## 이 태스크 정보

이제 이 사용자 정의된 SSO 인터셉터에서 Build Forge에 사용자로 전달되는 AuthServlet 요청을 통해 사용자를 인증할 수 있는 WAS 보안 기술을 사용할 수 있습니다.

### 프로시저

1. 웹 브라우저를 여십시오. `http://localhost` 주소를 입력하십시오.
2. Build Forge 로그인 양식 대신 인증 페이지가 표시됩니다. 사용자 신임을 입력하고 **Enter** 키를 누르십시오.
3. 인증 후 로그인이 자동으로 수행되어야 합니다.
4. 로그아웃하면 로그인 양식 대신 기본 jsp 페이지가 표시됩니다. 사용자가 여전히 인증된 상태인 경우 후속 로그인도 자동으로 수행됩니다.

### 양식 기반 SSO로 되돌리기:

SSO 로그인 양식을 사용하도록 되돌릴 수 있습니다.

## 이 태스크 정보

양식 로그인을 사용하도록 시스템을 다시 구성하려면 WAS에서 `rbf-services`를 설치 제거하고 원래 `rbf-services.war` 파일을 다시 설치해야 합니다. Build Forge에서 양식 기반 SSO 인터셉터가 사용 가능하고 맨 위 항목으로 나열되는지 확인하십시오. 사용자 정의 WAS 인터셉터를 사용하지 않도록 설정하십시오. 이러한 변경을 적용하려면 WAS와 Build Forge를 다시 시작하십시오.

### 프로시저

1. `rbf-services`를 설치 제거하고 원래 `rbf-services.war` 파일을 다시 설치하십시오.
2. Build Forge에서 양식 SSO 인터셉터가 사용 가능하고 맨 위 항목으로 나열되는지 확인하십시오(위의 "Build Forge SSO 보안 구성" 참조).
3. 사용자 정의 WAS 인터셉터를 사용하지 않도록 설정하십시오.
4. WAS를 다시 시작하십시오.
5. Build Forge를 다시 시작하십시오.

## SSL 및 HTTPS 사용

SSL 및 HTTPS를 사용하도록 Build Forge 시스템을 구성하면 시스템 보안이 증가합니다. SSL은 엔드포인트 인증 및 데이터 암호화를 포함합니다.

기본적으로 Build Forge 시스템에서는 Apache Tomcat의 인증 서블릿을 사용하는 로그인 양식에만 SSL을 사용합니다. Build Forge 시스템 전체에 추가 SSL 보호를 사용하려면 다음 설정을 수행해야 합니다.

1. Apache 서버에서 SSL을 사용하도록 설정하십시오. 이 단계는 설치하는 동안 SSL을 사용하도록 지정하지 않은 경우에만 필요합니다.
2. 클라이언트 및 내부 통신에 SSL을 사용하도록 설정하십시오.
3. 에이전트에 SSL을 사용하도록 설정하십시오.

**참고:** WebSphere 컴포넌트와 통합할 경우 SSL 지원에 필요한 전제조건이 충족되는지 확인하십시오.

- WebSphere Application Server와의 통합은 615 페이지의 『Apache Tomcat 대신 WebSphere Application Server 사용』을 참조하십시오.

## 기본 로그인 보안 정보

기본적으로 Build Forge에는 로그인 보안이 포함되어 있습니다. 사용자가 로그인하면 요청이 인증 서블릿에 리디렉션됩니다. 입력한 사용자 이름 및 비밀번호는 서블릿에서 사용하도록 암호화됩니다. 로그인에 성공하면 콘솔 사용자 인터페이스 홈이 표시됩니다. 클라이언트 및 콘솔 간 후속 세션 통신은 http(기본값) 또는 https에서 이루어질 수 있습니다. https를 사용하려면 시스템의 추가 구성이 필요합니다. 123 페이지의 『SSL 및 HTTPS 사용』의 내용을 참조하십시오.

설치 중 암호화에 사용되는 키 저장소의 비밀번호를 제공합니다. 또한 자체 서명 인증서를 설치하는 옵션이 있습니다.

**자가 서명 인증서에 대한 인증 메시지:** Build Forge에서 자가 서명 인증서를 설치하는 경우 보안 사용 브라우저를 통해 시스템에 액세스하는 사용자에게 인증서에 대한 경고 메시지가 표시됩니다.

이러한 경고를 방지하려면 사용자에게 사용자의 브라우저에 설치할 수 있도록 인증서를 분배합니다. 인증서 설치 특성은 브라우저에 따라 다릅니다. 브라우저 문서를 참조하십시오.

인증서는 `<bfinstall>/keystore`에 있습니다.

**기본 로그인 보안 사용 안함:** 로그인 보안을 사용하지 않는 경우 로그인 중에 사용자 신임 정보가 암호화되지 않은 일반 텍스트로 콘솔에 전달됩니다. 로그인 보안을 사용하지 않도록 설정해도 콘솔이 HTTPS/SSL을 사용하도록 설정되었으면 콘솔의 HTTPS/SSL 사용에 영향을 미치지 않습니다.

인증 서블릿을 사용하지 않으려면 다음을 수행하십시오.

1. Build Forge가 실행 중인 경우 중지하십시오.

2. <bfinstall>/buildforge.conf를 편집하여 서비스 계층과의 통신에 HTTP 및 포트 8080을 지정하십시오.

다음 행을 변경하십시오.

```
services_url https://hostname:8443/rbf-services
```

행을 다음으로 변경하십시오.

```
services_url http://hostname:8080/rbf-services
```

3. 서비스 계층 구성 파일을 편집하여 강제 실행된 SSL을 끄십시오. <bfinstall>/Apache/tomcat/webapps/rbf-services/WEB-INF/web.xml을 편집하십시오. ForceHttps 설정을 false로 변경하십시오. 파일에 설정이 없는 경우 굵은체로 표시된 대로 해당 설정을 추가하십시오.

```
<display-name>A Services Layer Bootstrap Servlet</display-name>
<servlet>
  <servlet-name>ServicesBootstrap</servlet-name>
  <servlet-class>com.buildforge.services.server.web.

BootstrapServlet</servlet-class>
  <init-param>
    <param-name>port</param-name>
    <param-value>3966</param-value>
  </init-param>
  <init-param>
    <param-name>sslPort</param-name>
    <param-value>49150</param-value>
  </init-param>
  <init-param>
    <param-name>ForceHttps</param-name>
    <param-value>>false</param-value>
  </init-param>
  <load-on-startup>0</load-on-startup>
</servlet>
```

4. Build Forge를 시작하십시오.

**참고:** 인증 서블릿이 사용되지 않도록 설정되면 일반 텍스트에서 네트워크를 통해 사용자 신임이 통신됩니다. 이는 보안상 위험합니다.

## SSL 및 Build Forge 컴포넌트 정보

SSL이 사용 가능한 경우 기본적으로 Build Forge 컴포넌트는 특정 포트와 보안 설정을 사용하도록 설정됩니다.

**기본 SSL 설정:** 기본 인증을 사용하는 경우 비교적 간단하게 SSL을 사용할 수 있습니다. 이 절의 프로시저는 해당 시나리오를 기반으로 합니다.

그러나 일반적으로 각 시스템에서 동일한 인증(개인용 키)을 사용하지 않는 것이 좋습니다. 하나의 시스템에서 개인용 키가 손상되면 전체 인프라스트럭처가 손상될 수 있습니다. 실제 보안을 실행하여 손상될 가능성을 줄일 수 있습니다.

더욱 안전한 시스템에서는 각 프로세스에 대한 인증을 사용합니다. Build Forge에서 다음을 수행하십시오.

- 모든 에이전트에 대한 인증서를 작성하십시오.
- 모든 엔진에 대한 인증서를 작성하십시오. 이는 중복이 설정된 경우 적용됩니다. 103 페이지의 『중복 구성』의 내용을 참조하십시오.

이 설정을 수행하려면 추가 인증서 관리가 필요합니다. 다음을 선택할 수 있습니다.

- 인증서를 생성하기 위해 CA(Certificate Authority)를 사용할 수 있습니다. 이를 수행하면 서명자 교환 횟수가 감소됩니다.
- 모든 신뢰 저장소 또는 CA 저장소에 연결하는 데 필요한 서명자가 있는지 확인할 수 있습니다.

다음 절에서는 SSL 보안이 강제 실행되는 Build Forge 시스템에서의 인터페이스에 대해 설명합니다.

**클라이언트 인터페이스:** 사용자는 클라이언트 인터페이스를 통해 Build Forge 시스템에 액세스합니다.

#### **Build Forge의 웹 클라이언트**

웹 클라이언트는 해당 Apache 웹 서버를 통해 Build Forge에 액세스합니다. SSL이 사용 가능하고 보안을 사용하도록 설정한 웹 브라우저가 사용되는 경우 다음 인터페이스가 사용됩니다.

- **Apache 웹 서버 포트 443**

웹 클라이언트는 해당 URL을 통해 Build Forge에 액세스합니다. SSL이 사용 가능한 경우 URL은 다음과 같습니다.

`https://host/`

*host*는 Build Forge가 실행되는 호스트입니다. Apache에 대한 보안 액세스를 위해 443이 아닌 포트를 설정하는 경우 사용자는 포트도 지정해야 합니다.

`https://host:port/`

웹 클라이언트가 Apache Tomcat 서버에서 실행 중인 인증 서블릿으로 경로가 재지정됩니다.

- **Apache Tomcat 애플리케이션 서버 포트 8443**

인증 서블릿에서 로그인 신임을 허용하고 사용자를 인증합니다. 서블릿에서 유선을 통해 일반 텍스트에 신임이 표시되지 않도록 해당 신임을 암호화합니다.

Apache Tomcat 서블릿에서 사용되는 리스너 포트의 구성은 구성 파일을 통해 관리됩니다. 이는 <bfinstall>/Apache/tomcat/conf/server.xml에 있습니다. 다음 커넥터 구성을 찾습니다.

```
<Connector port="8443" maxHttpHeaderSize="8192" algorithm="IbmX509"
  maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
  enableLookups="false" disableUploadTimeout="true"
  acceptCount="100" scheme="https" secure="true"
  clientAuth="false" sslProtocol="SSL_TLS"
  keystoreFile="C:\BuildForge71.536\keystore\buildForgeKeyStore.p12"
  keystorePass="password"
  keystoreType="PKCS12"
  truststoreFile="C:\BuildForge71.536\keystore\buildForgeTrustStore.p12"
  truststorePass="password"
  truststoreType="PKCS12"/>
```

## Build Forge의 API 프로그램 클라이언트

- Apache Tomcat 애플리케이션 서버 포트 49150

API 클라이언트는 Apache Tomcat 애플리케이션 서버에서 실행 중인 애플리케이션인 해당 서비스 계층 컴포넌트를 통해 Build Forge에 액세스합니다. API 클라이언트에 올바른 bfclient.conf 파일이 있어야 합니다.

서비스 계층 컴포넌트에서는 인바운드 통신에 SSL 구성을 사용합니다. 이는 관리 > 보안 > SSL에 있는 Build Forge 콘솔에 정의되어 있습니다. 사용되는 기본값은 기본 JSSE 인바운드 SSL입니다.

**내부 인터페이스:** Build Forge는 웹 인터페이스 컴포넌트(Apache 웹 서버 및 PHP), 서비스 계층 컴포넌트 및 엔진 컴포넌트로 구성되어 있습니다. 웹 인터페이스와 엔진 컴포넌트는 서비스 계층 컴포넌트의 클라이언트입니다. API 프로그램 클라이언트도 서비스 계층 컴포넌트의 클라이언트입니다.

### 서비스 계층 인바운드

#### Apache Tomcat 애플리케이션 서버 포트 49150

서비스 계층 컴포넌트에서는 인바운드 통신에 SSL 구성을 사용합니다. 이는 관리 > 보안 > SSL에 있는 Build Forge 콘솔에 정의되어 있습니다. 사용되는 기본값은 기본 JSSE 인바운드 SSL입니다.

### 서비스 계층 클라이언트 아웃바운드

Apache Tomcat 애플리케이션 서버 포트 49150 PHP를 통한 웹 인터페이스 컴포넌트와 엔진 컴포넌트에서는 모두 서비스 계층 컴포넌트에 대한 아웃바운드 통신에 SSL 구성을 사용합니다. 이는 관리 > 보안 > SSL에 있는 Build Forge 콘솔에 정의되어 있습니다. 사용되는 기본값은 기본 JSSE 아웃바운드 SSL입니다.

SSL 데이터 교환을 성공적으로 수행하기 위해 클라이언트 아웃바운드 구성 및 서비스 계층 인바운드 구성에 대한 SSL 특성이 호환 가능해야 합니다. 관리 > 보안 > SSL에서 유형 및 핸드셰이크 프로토콜 특성이 일치해야 합니다.

각 SSL 구성에 다음과 같은 참조 키 저장소 구성이 있습니다.

- 키 저장소 구성: 개인용 인증이 포함되어 있는 키 저장소의 특성을 정의합니다.
- 신뢰 저장소 구성: 신뢰할 수 있는 서명자가 포함되어 있는 키 저장소의 특성을 정의합니다.

구성은 이름별로 지정됩니다. 관리 > 보안 > **SSL**에서 이를 정의합니다. 몇 가지 기본 값이 제공됩니다.

**외부 인터페이스:** 외부 인터페이스는 외부 시스템과 통신하기 위해 Build Forge에서 사용되는 인터페이스입니다.

- Build Forge 엔진은 에이전트와 통신합니다.
- Build Forge 서비스 계층 컴포넌트는 데이터베이스와 통신합니다.

#### 에이전트 통신을 위한 **Build Forge** 엔진

이 인터페이스에 SSL을 사용하려면 다음이 필요합니다.

- 에이전트를 구성합니다. 이를 수행하려면 에이전트 구성 파일의 내용을 변경하고 에이전트 호스트에 인증서를 배치해야 합니다.
- 에이전트를 사용하는 각 서버 자원에 SSL 통신을 사용 가능하도록 설정합니다. 서버 패널에 있는 콘솔에서 이를 수행합니다.

131 페이지의 『에이전트 통신에 SSL 사용』의 내용을 참조하십시오.

#### 데이터베이스 통신을 위한 **Build Forge** 서비스 계층 컴포넌트

이 인터페이스에 대한 SSL 구성은 데이터베이스에 대한 디바이스 드라이버에 정의되어 있습니다.

### **Apache 서버에서 SSL 사용**

SSL이 사용 가능한 경우 기본적으로 Build Forge 컴포넌트는 특정 포트와 보안 설정을 사용하도록 설정됩니다.

설치하는 동안 웹 및 애플리케이션 서버 패널에 다음과 같은 질문이 표시됩니다. 보안 **HTTP**를 사용하시겠습니까?

예로 응답하면 Apache 서버에서 SSL을 사용할 수 있습니다. 이 섹션을 건너뛸니다.

아니오로 응답하지만 SSL을 사용하려는 경우 Build Forge 호스트에 로그인하고 Apache 서버를 다음과 같이 구성합니다.

- httpd.conf 파일

<bfinstall>/Apache/conf/httpd.conf를 편집하여 이 설정을 사용하십시오.

```
Listen 0.0.0.0:443
ServerName localhost:443
```

- ssl.conf 파일

<bfinstall>/Apache/conf/ssl/ssl.conf를 편집하여 이 설정을 사용하십시오.

```
<VirtualHost *:443>
ServerName localhost:443
SSLEngine on
```

선택하는 경우 호스트에 대한 완전한 도메인 이름을 지정할 수 있습니다.

## 클라이언트 및 내부 연결에 SSL 사용

SSL이 사용 가능한 경우 기본적으로 Build Forge 컴포넌트는 특정 포트와 보안 설정을 사용하도록 설정됩니다.

클라이언트에서 Build Forge로의 SSL 연결 및 Build Forge 내부 컴포넌트 간에 SSL 연결을 사용하려면 정확하게 지정된 순서에 따라 다음을 수행합니다.

- 콘솔에서 **관리 > 보안**으로 이동하십시오.
- **SSL 사용**을 예로 설정하십시오. 여러 가지 추가 특성이 표시됩니다. 해당 특성을 기본값으로 설정된 상태로 둡니다. 필요한 경우 나중에 사용자 정의할 수 있습니다.
- **저장**을 클릭하십시오. Build Forge 데이터베이스에 SSL 구성이 저장됩니다.
- 마스터 **BFClient.conf** 업데이트를 클릭하십시오. SSL 구성은 BFClient.conf 파일을 업데이트하는 데 사용됩니다. 이 파일에 사용할 Build Forge에 대한 설정이 있어야 합니다.
- Build Forge를 중지하고 다시 시작하십시오. 이는 필수입니다.

시스템이 기본 설정으로 작동되는 경우 SSL에 영향을 주는 특성을 편집할 수도 있습니다.

**참고:** 클라이언트에서 SSL을 사용하지만 내부 클라이언트(웹 인터페이스 컴포넌트와 엔진 컴포넌트)와 서비스 계층 컴포넌트 간에 SSL을 사용하지 않으려면 다음을 수행하십시오.

- BFClient.conf 파일을 수동으로 편집하십시오. 이는 <bfinstall>에 있습니다.
- bf\_services\_preferred\_protocol 설정을 tcp로 변경하십시오.  
bf\_services\_preferred\_protocol to tcp

이 구성은 Build Forge 호스트가 실제로 보안되는 경우 보안 위험성이 최소인 상태에서 성능을 향상시킵니다.

## 잠긴 시스템에서 TCP 통신 다시 사용

SSL에 잘못된 구성이 있는 경우 시스템이 잠깁니다.

잠긴 시스템에 액세스하려면 다음을 수행하십시오.

- Build Forge를 중지하십시오.



- <bfinstall> 디렉토리에서 문서 편집기로 bfclient.conf 파일을 여십시오.
- 다음과 같이 프로토콜 특성을 변경하십시오.  
bf\_services\_preferred\_protocol=tcp
- Build Forge를 시작하십시오.

로그인할 수 있어야 합니다.

**참고:** 프로토콜을 변경해도 기본적으로 사용할 수 있는 보안 로그인 인증은 사용할 수 있습니다. 사용자가 Build Forge에 대한 로그인 신임의 보안 통신을 허용하는 보안 연결에 리디렉션됩니다.

## API 클라이언트에 적합한 SSL 사용(Perl 또는 Java)

bfclient.conf를 사용하여 Perl 또는 Java로 작성된 API 클라이언트에 적합한 SSL 연결을 구성할 수 있습니다.

## 이 태스크 정보

API 클라이언트 프로그램으로 SSL 연결을 작성하려면 bfclient.conf를 설정하십시오. 이 파일에는 SSL 구성 특성이 포함되어 있습니다. 이 프로시저를 간략하게 하기 위해 아래 단계에서는 첫 번째 단계에서 언급된 클라이언트 디렉토리에서 클라이언트를 실행하는 것으로 가정합니다.

## 프로시저

1. 스크립트를 실행할 클라이언트 디렉토리를 작성하십시오.

Windows: mkdir c:\client

UNIX 또는 Linux: mkdir /temp/client

2. 클라이언트 디렉토리에 bfclient.conf를 복사하십시오.

Windows: copy c:\BuildForge\bfclient.conf c:\client

UNIX 또는 Linux: cp /opt/BuildForge/Platform/bfclient.conf /temp/client

3. 클라이언트 디렉토리에 키 저장소 서브디렉토리를 작성하십시오.

Windows: mkdir c:\client\keystore

UNIX 또는 Linux: mkdir /temp/client/keystore

4. BuildForge 설치에서 클라이언트 키 저장소 디렉토리로 키 저장소를 복사하십시오.

Windows: copy \BuildForge\keystore\\*.pem \client\keystore



UNIX 또는 Linux: `cp /opt/BuildForge/Platform/keystore/*.pem /temp/client/keystore`

5. Perl의 경우 `bfclient.conf` 위치를 찾을 수 있도록 다음 OS 환경 변수를 설정하십시오.

Windows: `set BFCLIENT_CONF=c:\client\bfclient.conf`

UNIX 또는 Linux: `export BFCLIENT_CONF="/temp/client/bfclient.conf"`

6. Java의 경우 다음 단계를 완료하십시오.
- `bfclient.conf`를 사용하여 SSL 연결을 작성하려면 코드에서 `APIClientConnection` 대신 `SecureAPIClientConnection`을 사용하십시오.
  - 스크립트 실행 시 Java 명령행에서 다음 시스템 특성을 설정하십시오.

Windows: `-Dcom.buildforge.client.config=c:\client\bfclient.conf`

UNIX 또는 Linux: `-Dcom.buildforge.client.config=/temp/client/bfclient.conf`

7. `bfclient.conf`를 편집하고 다음 특성을 확인하십시오.

- `bf_services_preferred_protocol=ssl`

값이 `tcp`가 아니라 `ssl`이어야 합니다.

- `bf_services_ssl_port=49150`

49150은 서비스 계층 SSL 포트여야 합니다.

- `bf_keystore_location=./keystore/buildForgeKey.pem`

키 저장소 위치는 여러 개입니다. 클라이언트 디렉토리 이외의 디렉토리에서 스크립트를 실행하는 경우 완전한 경로를 사용하도록 각 키 저장소 위치를 변경하십시오.

8. 스크립트에서 `bfclient.conf`를 올바르게 사용하려면 사용자 환경에서 다음 디버그 특성을 설정하십시오. 스크립트 실행 시 SSL 연결 특성에 대한 추가 출력이 표시되어야 합니다.

Windows: `set BFDEBUG_SECURITY=1`

UNIX 또는 Linux: `export BFDEBUG_SECURITY=1`

## 에이전트 통신에 SSL 사용

SSL이 사용 가능한 경우 기본적으로 Build Forge 컴포넌트는 특정 포트와 보안 설정을 사용하도록 설정됩니다.

Build Forge와 에이전트 간에 SSL 통신을 사용하려면 다음을 수행해야 합니다.

- UNIX 컴퓨터에서 에이전트를 호스트하는 컴퓨터에 GCC 라이브러리가 설치되었는지 확인하십시오.
- 전제조건: 클라이언트와 내부 통신에 SSL을 사용합니다. 129 페이지의 『클라이언트 및 내부 연결에 SSL 사용』의 내용을 참조하십시오.
- 각 에이전트를 구성합니다. 이 태스크에는 다음과 같은 내용이 포함됩니다.
  - 에이전트 호스트에 인증 추가
  - 에이전트에 대한 bfaagent.conf 파일 편집
- 콘솔에서 에이전트에 연결되는 각 서버 정의에 SSL을 사용합니다.

## SSL을 위한 에이전트 구성

1. 에이전트가 실행 중인 경우 해당 에이전트를 중지하십시오.
2. 인증을 위한 .PEM 파일을 에이전트의 루트 설치 디렉토리에 배치하십시오.

SSL을 빠르게 구현하고 테스트하려면 Build Forge 설치에서 .PEM 파일을 복사할 수 있습니다. 파일은 `<bfinstall>/keystore`에 있습니다.

SSL을 위한 우수 사례는 각 에이전트에 개별 인증을 사용하는 것입니다.

- a. 다음에 대해 각각 개별 키 저장소 파일(.PEM)을 작성합니다.
    - 개인용 키(key)
    - 개인용 키(cert)의 공용 인증서
    - 신뢰할 수 있는 서명자(CA 또는 인증 권한)
  - b. Build Forge 엔진의 인증서 복사본 대신 에이전트의 고유 인증서를 사용하는 경우 Build Forge의 CA(Certificate Authority) 키 저장소인 `<bfinstall>/keystore/buildForgeCA.pem`에 에이전트의 인증서를 추가합니다. 다중 엔진(중복 엔진)을 실행 중인 경우 각 엔진의 CA(Certificate Authority) 키 저장소에 인증서를 추가합니다.
  - c. System z에서 kdb를 사용 중인 경우 gskkeyman을 사용하여 kdb를 작성하십시오.
  - d. SAF(System Authorization Facility) 키 링을 사용하는 경우 RACF를 사용하여 키 링을 작성하고 인증서를 키 링에 연결하십시오.
3. BFAgent.conf를 편집하십시오. 파일에서 다음 행이 주석 처리됩니다. 주석 접두부를 제거하십시오.

```
ssl_key_location buildForgeKey.pem
ssl_key_password password
ssl_cert_location buildForgeCert.pem
ssl_ca_location buildForgeCA.pem
ssl_protocol TLSv1
ssl_cipher_group ALL
```

비밀번호는 buildForgeKey.pem 키 저장소에 사용됩니다. 비밀번호를 암호화하려면 152 페이지의 『buildforge.conf 및 bfaagent.conf에서 비밀번호 암호화』의 내용을 참조하십시오.

에이전트에 연결할 때 클라이언트 인증이 필요한 경우 다음 행의 주석을 해제합니다.

```
ssl_client_authentication true
```

이 설정을 수행하려면 엔진의 CA(Certificate Authority) 키 저장소인 buildForgeCA.pem에 엔진 인증을 추가해야 합니다.

특정 암호를 사용하려는 경우 이 행의 주석을 해제하고 암호 목록을 추가합니다.

```
ssl_cipher_override cipher_list
```

4. System z에서 실행 중인 에이전트의 경우 BFAgent.conf를 편집하고 다음 행의 주석을 해제하십시오.

```
gsk_ssl_key_location <root/cert SAF>
gsk_ssl_key_location /etc/key.kdb
gsk_ssl_kdb_password <password>
gsk_keyring_label <certificate label>
gsk_ssl_protocol ALL
gsk_ssl_cipher_v2 6321
gsk_ssl_cipher_v3 0906030201
gsk_ssl_client_authentication true
gsk_password_encrypt false
```

하나의 gsk\_ssl\_key\_location만 사용하십시오. SAF 키 링 또는 kdb를 선택할 수 있습니다. SAF 키 링은 <userid>/<keyring>에서 지정합니다.

5. 에이전트를 시작하십시오. 콘솔에서 연결을 테스트하려면 에이전트가 실행 중이어야 합니다.

## 서버 정의에 SSL 사용

콘솔에서는 서버 정의를 사용하여 에이전트에 연결합니다.

SSL 사용 에이전트에 연결되는 각 서버 정의에 다음을 수행하십시오.

- 콘솔에서 서버 패널로 이동하십시오.
- 서버 정의 이름을 클릭하십시오.
- 서버 정의의 세부사항 탭에서 다음을 수행하십시오.
  - SSL 사용을 예로 설정하십시오.
- 저장을 클릭하십시오.
- 연결 테스트를 클릭하십시오.

## 에이전트와의 SSL 통신 문제점 해결

다음 점검 목록은 SSL을 사용할 때의 공통적인 문제에 대해 설명합니다.

- 에이전트
  - 보안 및 서버 정의에서 SSL을 사용할 수 있지만 에이전트 SSL이 구성되어 있지 않습니다.
  - Build Forge 엔진에서 에이전트의 인증을 신뢰할 수 없습니다. 엔진의 CA 키 저장소에 에이전트 인증을 추가해야 합니다.  
`<bfinstall>/keystore/buildForgeCA.pem`
  - BFAgent.conf에 지정된 키 저장소의 비밀번호가 올바르지 않습니다.
  - BFAgent.conf에 클라이언트 인증이 지정되어 있지만 에이전트의 CA(Certificate Authority)인 buildForgeCA.pem에 엔진의 인증이 추가되지 않았습니다.
- Build Forge 콘솔
  - 콘솔에서 SSL을 사용할 수 있었습니다. SSL이 저장되지 않았거나, 저장하기 전에 **Master Bfclient.conf** 업데이트를 클릭하지 않았거나, 저장한 후에 **마스터 Bfclient.conf** 업데이트를 클릭하지 않았습니다.
- 서버 정의
  - SSL을 사용할 수 없습니다. 연결 테스트를 수행하기 전에 저장을 클릭하여야 합니다.
- 엔진과 에이전트 설정 일치
  - 데이터 교환 프로토콜이 일치하지 않습니다. TLSv1 또는 SSLv3과 같이 엔진 SSL 구성과 에이전트 SSL 구성에 대한 데이터 교환 프로토콜을 동일한 값으로 설정해야 합니다. 기본값은 TLSv1입니다.
  - 암호 스위트가 일치하지 않습니다. 엔진 SSL 구성 및 에이전트 SSL 구성에 지정된 암호 스위트에 공통적인 암호가 있어야 합니다. 기본 암호 스위트 그룹은 ALL입니다.

## 디버깅 메시지 사용

엔진 및 에이전트에서 디버깅을 사용할 수 있습니다. 디버깅을 사용할 수 있는 경우 구성에서의 문제점을 식별할 수 있는 자세한 추가 출력이 생성됩니다.

- 엔진 디버깅 사용:
  1. SSL 사용 서버 정의에서 사용되는 환경에서 다음 변수를 추가하십시오.  
`BFDEBUG_SECURITY=1`
  2. 엔진을 중지하고 다시 시작하십시오.
    - Windows의 경우 포그라운드로 엔진을 시작합니다. 명령 창에 출력이 표시됩니다.
    - UNIX 또는 Linux의 경우 디버깅이 켜진 상태로 엔진을 시작합니다.

```
cd <bfinstall>/rc
./buildforge start
```

엔진의 출력이 <bfinstall>/log의 엔진 로그 파일로 이동됩니다.

- 에이전트 디버깅 사용:

1. 에이전트를 중지하십시오.
2. BFAgent.conf에 다음 행을 추가하십시오.

```
activity_log bfaagent.log
```

이 예제에서는 에이전트가 bfaagent.log에 출력을 씁니다. 다른 파일 이름을 지정할 수 있습니다.

**참고:** 에이전트가 서비스로 실행되는 경우 절대 경로를 지정합니다.

3. 에이전트를 시작하십시오.

## 성공적인 SSL 연결을 위한 예제 엔진 디버그 출력

엔진이 에이전트에 성공적으로 연결되면 다음과 같은 출력이 생성됩니다.

```
SSL_ca_file: ./keystore/buildForgeCA.pem
SSL_cert_file: ./keystore/buildForgeCert.pem
SSL_key_file: ./keystore/buildForgeKey.pem
SSL_verify_mode: 0x01
SSL_version: TLSv1
SSL_cipher_list: ALL
SSL_use_cert: 1
Making as SSL connection using socket IO::Socket::INET=GLOBAL(0x1e8f0f4).
SSL connection to agent.
DEBUG: .../IO/Socket/SSL.pm:1387: new ctx 80662848
DEBUG: .../IO/Socket/SSL.pm:880: dont start handshake: IO::Socket::SSL=GLOBAL(0x1e8f0f4)
DEBUG: .../IO/Socket/SSL.pm:284: ssl handshake not started
DEBUG: .../IO/Socket/SSL.pm:327: Net::SSLay::connect -> 1
DEBUG: .../IO/Socket/SSL.pm:382: ssl handshake done
Socket is of type: ref(IO::Socket::SSL=GLOBAL(0x1e8f0f4))
ReadyLine: 202 HELLO TLS - BuildForge Agent v_VERSION_
.
Storing Agent Version [999.999.999.999-999-9999] for [08974C8E-6C3B-1014-972D-D9B2901D9F42]
cmd ping username pbirk
encpass c1713f4a31af3f1300f7b2414a24559c4d6097e07310cf9c412e
goSending agent request...
```

## 성공적인 SSL 연결을 위한 예제 에이전트 디버그 출력

정상적으로 실행되는 에이전트에서 SSL 연결이 확립되는 경우 다음과 같은 출력이 생성됩니다.

```
[2256] main.c : 409: === NEW AGENT ===
[2256] io.c : 264: In start_SSL
[2256] io.c : 89: Key location: buildForgeKey.pem
[2256] bfpwdlocloader.c: 134: Looking for password locator: ssl_key_password_locator
[2256] bfpwdlocloader.c: 244: Looking for password for prop
ssl_key_password from bfaagent.conf.
```

```

[2256] bfcryptloader.c : 202: Loading password encryption module.
[2256] bfcryptloader.c : 276: Password encryption property
      password_encrypt_module is not configured.
[2256] bfcryptloader.c : 539: Password decoded.
[2256] io.c : 98: Cert location: buildForgeCert.pem
[2256] bfpwdlocloader.c: 134: Looking for password locator:
      ssl_cert_password_locator
[2256] bfpwdlocloader.c: 244: Looking for password for prop
      ssl_cert_password from bfagent.conf.
[2256] io.c : 153: Setting key password in default userdata.
[2256] io.c : 160: Getting private key from PEM.
[2256] io.c : 166: Checking private key from PEM.
[2256] io.c : 172: Getting CA store information.
[2256] bfpwdlocloader.c: 134: Looking for password locator:
      ssl_ca_password_locator
[2256] bfpwdlocloader.c: 244: Looking for password for prop
      ssl_ca_password from bfagent.conf.
[2256] io.c : 178: CA location: buildForgeCert.pem
[2256] io.c : 184: Checking the CA store.
[2256] io.c : 230: Returning from init_CTX.
[2256] io.c : 281: Calling SSL_new
[2256] io.c : 294: Calling SSL_accept.
[2256] io.c : 346: Cipher chosen: AES256-SHA
[2256] io.c : 367: ssl_state = SS_CERTIFIED

```

## 에이전트에서 잘못된 키 저장소 비밀번호의 예제 출력

에이전트 측에 구성된 키 저장소 비밀번호가 잘못된 경우 엔진과 에이전트 출력에 모두 표시됩니다.

엔진 출력(예제):

```

SSL_use_cert: 1
Making an SSL connection using socket IO::Socket::INET=GLOBAL(0x1e8f0f4).
SSL connection to agent.
DEBUG: .../IO/Socket/SSL.pm:1387: new ctx 80662848
DEBUG: .../IO/Socket/SSL.pm:880: dont start handshake: IO::Socket::SSL=GLOBAL(0x1e8f0f4)
DEBUG: .../IO/Socket/SSL.pm:284: ssl handshake not started
DEBUG: .../IO/Socket/SSL.pm:327: Net::SSL::connect -> -1
DEBUG: .../IO/Socket/SSL.pm:1135: SSL connect attempt failed with unknown error
error:1408F10B:SSL routines:SSL3_GET_RECORD:wrong version number

DEBUG: .../IO/Socket/SSL.pm:333: fatal SSL error: SSL connect attempt failed with unknown error
error:1408F10B:SSL routines:SSL3_GET_RECORD:wrong version number
DEBUG: .../IO/Socket/SSL.pm:1422: free ctx 80662848 open=80662848 80566656
DEBUG: .../IO/Socket/SSL.pm:1425: OK free ctx 80662848

```

에이전트 출력(예제):

```

[ 5272] io.c : 98: Cert location: buildForgeCert.pem
[ 5272] bfpwdlocloader.c: 134: Looking for password locator: ssl_cert_password_locator
[ 5272] bfpwdlocloader.c: 244: Looking for password for prop ssl_cert_password from bfagent.conf.
[ 5272] io.c : 153: Setting key password in default userdata.
[ 5272] io.c : 160: Getting private key from PEM.
[ 5272] io.c : 218: Failure reason: SSL_ERROR_BAD_KEY_FILE
[ 5272] io.c : 221: OpenSSL error string: error:00000000:lib(0):func(0):reason(0)
[ 5272] io.c : 281: Calling SSL_new
[ 5272] platform.c :2693: platform_release_credentials
[ 5272] main.c : 412: --- EXITING ---

```

## 에이전트 출력에서 오류 코드

이 목록에는 발생할 수 있는 몇 가지 기타 오류 코드와 원인이 포함되어 있습니다.

- **SSLERorBadCA:** buildForgeCA.pem에서 서명자를 로드하는 데 문제점이 있습니다. 서명자를 추가한 방법으로 인해 파일 형식에 문제점이 발생할 수 있습니다.
- **SSLERorBadCert:** buildForgeCert.pem에서 인증을 로드하는 데 문제점이 있습니다. 인증서가 개인용 키와 일치하지 않거나 PEM에서 손상되었습니다.
- **SSLERorBadPKeyFile:** BFAgent.conf에서 ssl\_key\_password 특성에 대해 지정된 buildForgeKey.pem 비밀번호에 문제점이 있습니다.
- **SSLERorBadPKey:** SSL\_CTX\_check\_private\_key에서 1이 아닌 값을 리턴했습니다. 개인용 키 형식이 올바르지 않거나 인증서와 일치하지 않습니다.
- **SSLERorFIPSEnablement:** FIPS 사용 중에 오류가 발생했습니다. 일반적으로 이는 FIPS 자가 진단 중에 발생한 문제 때문입니다. 이 오류는 내부 오류일 수 있습니다.
- **SSLERorInvalidCipher:** 암호 스펙이 허용되는 OpenSSL과 일치하지 않습니다. BFAgent.conf에서 ssl\_cipher\_group 또는 ssl\_cipher\_override 특성에 지정된 암호를 확인하십시오.
- **SSLERorNoCtx:** 새 SSL\_CTX 오브젝트 작성에 문제점이 있습니다. 이 오류는 내부 오류일 수 있습니다.

## 인증서 관리

설치 후 Build Forge에서 사용되는 인증서 및 키 저장소를 수정할 수 있습니다.

설치하는 동안 사용할 인증서(자체 인증서 또는 Build Forge에서 생성된 인증서) 및 키 저장소 비밀번호를 지정할 수 있습니다. 이 절에서는 다음에 대한 프로시저에 대해 설명합니다.

- PEM 키 저장소를 Build Forge에 필요한 OpenSSL 및 JSSE 키 저장소로 변환
- 키 저장소 비밀번호 변경 및 새 비밀번호를 사용하도록 Build Forge 수정
- 새 자가 서명 인증서 작성
- 클라이언트 인증서를 사용하여 로그인
- 강력하게 암호화된 인증서를 사용하도록 제한되지 않은 정책 파일 다운로드

### PEM 키 저장소를 Build Forge 키 저장소로 변환:

CA(Certificate Authority)에서 받은 PEM 키 저장소를 Build Forge에서 사용할 수 있는 키 저장소로 변환할 수 있습니다.



## 시작하기 전에

SDK에 대한 제한되지 않은 정책 파일을 다운로드합니다. 이 전제조건은 제한된 정책 파일에 대한 키 크기가 너무 큰 경우에만 적용됩니다. [https://www14.software.ibm.com/webapp/iwm/web/reg/signup.do?source=jcesdk&lang=en\\_US&S\\_PKG=142ww](https://www14.software.ibm.com/webapp/iwm/web/reg/signup.do?source=jcesdk&lang=en_US&S_PKG=142ww)에서 파일을 다운로드하십시오.

참고: IBM에서 제공하는 keytool 유틸리티를 사용해야 합니다.

## 이 태스크 정보

인증 기관으로부터 받은 PEM 파일 세트가 있는 경우 해당 파일을 사용하여 Build Forge에 적합한 OpenSSL 및 JSSE 키 저장소 세트를 작성해야 합니다.

## 프로시저

1. 경로에 Build Forge 도구 디렉토리를 포함하십시오.

- `<bfinstall>/openssl`
- `<bfinstall>/ibmjdk/bin`(Windows의 경우)
- `<bfinstall>server/ibmjdk/bin`(UNIX 또는 Linux의 경우)

UNIX 및 Linux의 경우 LD\_LIBRARY\_PATH에 다음 디렉토리를 포함시키십시오.

`<bfinstall>/openssl`

2. PEM 파일을 PKCS12 키 저장소로 변환하십시오.

다음 명령을 사용하십시오.

```
openssl pkcs12
    -export
    -name "buildforge"
    -out buildForgeKeyStore.p12
    -inkey <key.pem>
    -passin pass:<pempassword>
    -in <crt.pem>
    -password pass:<bfpasword>
```

3. 인증서가 추가되었으며 읽을 수 있는지 확인하십시오.

```
keytool -v
    -list
    -keystore buildForgeKeyStore.p12
    -storepass <bfpasword>
    -storetype pkcs12
```

올바르지 않은 키 크기에 대한 오류가 발생하는 경우 제한사항이 없는 정책 파일을 다운로드하십시오. 이 절의 시작 부분에 있는 지시사항을 사용하십시오.

4. 공용 인증서를 내보내십시오.



명령 창에서 `<bfinstall>/keystore`로 이동하고 이 명령을 실행하십시오.

```
keytool -export
        -alias buildforge
        -file cert.der
        -keystore buildForgeKeyStore.p12
        -storepass <bfpasword>
        -storetype pkcs12
```

- 인증서가 cert.der 파일에 저장됩니다.
- 설치 중에 키 저장소에 지정된 동일한 `<bfpasword>`를 사용하십시오. 그렇지 않으면 구성을 변경해야 합니다.

5. 신뢰 저장소를 작성하고 공용 인증서를 가져오십시오.

명령 창에서 `<bfinstall>keystore`로 이동한 후 다음 명령을 실행하십시오.

```
keytool -import
        -noprompt -trustcacerts
        -alias buildforge
        -file cert.der
        -keystore buildForgeTrustStore.p12
        -storepass <bfpasword>
        -storetype pkcs12
```

6. buildForgeCert.pem에 공용 클라이언트 인증서를 배치하십시오.

명령 창에서 `<bfinstall>/keystore`로 이동하고 이 명령을 실행하십시오.

```
openssl pkcs12 -clcerts -nokeys
        -in buildForgeKeyStore.p12
        -passin: pass:<bfpasword>
        -out buildForgeCert.pem
```

7. buildForgeKey.pem에 인증서 및 키를 배치하십시오.

명령 창에서 `<bfinstall>/keystore`로 이동하고 이 명령을 실행하십시오.

```
openssl pkcs12
        -in buildForgeKeyStore.p12
        -passin pass:<bfpasword>
        -passout pass:<bfpasword>
        -out buildForgeKey.pem
```

8. PEM CA(Certificate Authority) buildForgeCA.pem을 작성하십시오.

- a. `<bfinstall>/keystore`에 CA 루트 인증서를 다운로드하십시오. 인증서 이름은 CARootCert.crt입니다. 이 인증서를 PEM 키 저장소에 추가해야 하며, buildForgeTrustStore.p12로 가져올 수 있습니다.

- b. 명령 창에서 `<bfinstall>/keystore`로 이동하고 이 명령을 실행하십시오.

```
cat CARootCert.crt > buildForgeCA.pem
keytool -import -noprompt -v -trustcacerts
        -alias "CA Root"
        -file CARootCert.crt
        -keystore buildForgeTrustStore.p12
        -storepass <bfpasword>
        -storetype pkcs12
```

## 결과

Build Forge에서 비밀번호로 보호된 PEM 키 저장소인 `buildForgeKey.pem`을 사용합니다. Apache 서버에서 시작 중에 비밀번호를 프롬프트합니다.

시작 중에 비밀번호에 대한 프롬프트를 표시하지 않으려면 비밀번호로 보호되지 않는 PEM 키 저장소를 생성하고 Apache 서버에서 해당 키 저장소를 사용하도록 합니다. 다음 명령을 예로 들 수 있습니다.

```
openssl rsa -in buildForgeKey.pem
            -passin pass:<password>
            -out buildForgeKeyForApache.pem
```

Build Forge가 실행되는 프로세스의 ID에 액세스해야 하는 모든 사용자가 보호되지 않는 PEM 키 저장소를 읽을 수 있어야 합니다.

**키 저장소 비밀번호 변경:** 설치하는 동안 키 저장소 비밀번호를 지정할 수 있습니다. 비밀번호를 변경하려면 다음을 수행해야 합니다.

1. 새 비밀번호를 사용하도록 Build Forge 키 저장소를 수정하십시오.
2. 새 비밀번호를 사용하도록 Build Forge 구성을 수정하십시오.

### 키 저장소 비밀번호 수정:

이 태스크 정보

Build Forge에는 비밀번호로 보호된 세 가지 기본 키 저장소가 있으며, 해당 키 저장소는 모두 Build Forge 엔진을 설치한 호스트인 `<bfinstall>/keystore`에 설치되어 있습니다.

- `buildForgeKey.pem` - OpenSSL에서 사용되며, 비밀번호를 변경하려면 `openssl` 도구가 필요합니다.
- `buildForgeKeyStore.p12` - JSSE에서 사용되며, 비밀번호를 변경하려면 `ibmjdk` 도구가 필요합니다.
- `buildForgeTrustStore.p12` - JSSE에서 사용되며, 비밀번호를 변경하려면 `ibmjdk` 도구가 필요합니다.

도구는 Build Forge 소프트웨어에 포함되어 있습니다.

**참고:** 예제 명령에서는 정확성을 위해 행 바꾸기가 사용됩니다. 명령에는 행 바꾸기를 사용하지 마십시오. 하나의 문자열로 입력하거나 행 연속 문자(Windows의 경우 `^`, UNIX 또는 Linux의 경우 `\`)를 사용합니다.

**중요사항:** 모든 키 저장소에 동일한 비밀번호가 사용됩니다. 이는 예제에서 `newpassword`로 표시됩니다.

## 프로시저

1. Build Forge 엔진이 설치되어 있는 호스트에 로그인하십시오.
2. 경로에 도구 디렉토리를 배치하십시오.
  - `<bfinstall>/openssl`
  - `<bfinstall>/ibmjdk/bin`
3. SSL을 사용하지 않도록 설정하십시오. 콘솔에서 **관리 > 보안**으로 이동하십시오. **SSL 사용**을 **아니오**로 설정하십시오.
4. **저장**을 클릭하십시오.
5. 마스터 **BFClient.conf** 업데이트를 클릭하십시오.
6. Build Forge 엔진을 중지하십시오.
7. 기존 키 저장소를 백업하십시오. 기존 Build Forge 키 저장소를 임시 디렉토리에 복사하십시오. 수정된 파일이 손상된 경우 백업된 키 저장소를 사용할 수 있습니다.
8. buildForgeKey.pem을 수정하십시오. `<bfinstall>/keystore` 디렉토리에서 이 명령을 실행하십시오.

```
openssl rsa
-in buildForgeKey.pem
-passin pass:oldpassword
-out buildForgeKey.pem
-passout pass:newpassword -aes128
```

9. buildForgeKeyStore.p12를 수정하십시오. `<bfinstall>/keystore` 디렉토리에서 이 명령을 실행하십시오.

```
keytool -storepasswd -all
-new newpassword
-keystore buildForgeKeyStore.p12
-storepass oldpassword
-storetype pkcs12
```
10. buildForgeTrustStore.p12를 수정하십시오. `<bfinstall>/keystore` 디렉토리에서 이 명령을 실행하십시오.

```
keytool -storepasswd -all
-new newpassword
-keystore buildForgeTrustStore.p12
-storepass oldpassword
-storetype pkcs12
```

## 결과

비밀번호가 변경되면 새 비밀번호를 사용하도록 Build Forge 구성을 수정해야 합니다.

### 새 비밀번호를 사용하도록 Build Forge 구성 수정:

변경된 키 저장소 비밀번호를 사용하도록 Build Forge 구성을 변경해야 합니다.

## 시작하기 전에

### 전제조건:

- SSL을 끄고 Build Forge 엔진을 중지하고 키 저장소 비밀번호를 수정하여 Build Forge 엔진이 시작되지 않았습니다.

### 이 태스크 정보

Apache Tomcat 애플리케이션 서버의 `server.xml` 구성 파일에 키 저장소 비밀번호가 포함되어 있습니다. 이러한 비밀번호는 일반 텍스트로 저장됩니다. Apache Tomcat에서는 이 설정에 인코딩되거나 암호화된 비밀번호를 지원하지 않습니다. 다음 프로시저에서 Build Forge 콘솔의 보안 특성 및 `server.xml`을 수정합니다.

### 프로시저

1. Tomcat 구성에 새 비밀번호를 입력하십시오. `<bfinstall>/Apache/tomcat/conf/server.xml`을 편집하십시오. SSL의 커넥터 명령문은 `<!-- Define a SSL HTTP/1.1 Connector on port 8443 -->` 주석 바로 아래에 있습니다.

```
<Connector port="8443" maxHttpHeaderSize="8192" algorithm="IbmX509"
maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
enableLookups="false" disableUploadTimeout="true"
acceptCount="100" scheme="https" secure="true"
clientAuth="false" sslProtocol="TLS"
keystoreFile="C:\Program Files\IBM\Build Forge\keystore\buildForgeTrustStore.p12"
keystorePass="newpassword"
keystoreType="PKCS12"
truststoreFile="C:\Program Files\IBM\Build Forge\keystore\buildForgeTrustStore.p12"
truststorePass="newpassword"
truststoreType="PKCS12"/>
```

2. Build Forge를 시작하십시오.
3. 콘솔에 로그인하십시오. 보안 액세스 역할이 있는 루트 또는 로그인을 사용합니다.
4. 키 저장소에 대한 새 비밀번호를 입력하십시오. 관리 > 보안 > 키 저장소에서 다음 항목을 편집하여 비밀번호 특성에서 `newpassword`를 사용하십시오.
  - 기본 JSSE 키 저장소
  - 기본 JSSE 신뢰 저장소
  - 기본 OpenSSL 키 저장소
5. SSL을 사용하도록 설정하십시오.
  - a. 콘솔에서 관리 > 보안으로 이동하십시오.
  - b. **SSL** 사용을 예로 설정하십시오.
  - c. 저장을 클릭하십시오.
6. `BFClient.conf`에 대한 변경사항을 내보내십시오. 마스터 **BFClient.conf** 업데이트를 클릭하십시오.
7. Build Forge 엔진을 시작하십시오.

## 새 자가 서명 인증서 작성:

제공된 도구를 사용하여 새 자가 서명 인증서를 작성합니다.

### 시작하기 전에

설치 중에 키 저장소에 대해 지정한 비밀번호가 필요합니다. 모르는 경우 `bfinstall/Apache/tomcat/conf/server.xml`에서 찾아보십시오. SSL/HTTP 커넥터의 `keystorePass` 속성에 비밀번호가 포함되어 있습니다.

### 이 태스크 정보

이 프로시저에서는 Build Forge를 설치하는 동안 자동으로 작성된 인증서를 바꾸는 방법에 대해 설명합니다. 다음 특성이 포함된 인증서가 작성됩니다.

- 키 저장소: `buildForgeKeyStore.p12`
- 만기: 15년(5475일로 설정)
- 주제 DN: `CN=hostname`, 여기서 `hostname`은 완전한 호스트 이름입니다.

`openssl` 및 `ibmjdk` 도구를 사용하여 인증서를 작성합니다. 도구는 Build Forge 소프트웨어에 포함되어 있습니다.

다음과 같은 다섯 개의 키 저장소가 필요합니다.

- `buildForgeKeyStore.p12` - 인증서 및 키의 컨테이너인 키 저장소
- `buildForgeTrustStore.p12` - 인증서 및 키의 컨테이너인 신뢰 저장소
- `buildForgeKey.pem` - PEM 키 저장소
- `buildForgeCert.pem` - 공용 인증서
- `buildForgeCA.pem` - PEM CA(Certificate Authority)

**참고:** 예제 명령에서는 정확성을 위해 행 바꾸기가 사용됩니다. 명령에는 행 바꾸기를 사용하지 마십시오. 하나의 문자열로 입력하거나 행 연속 문자(Windows의 경우 `^`, UNIX 또는 Linux의 경우 `\`)를 사용합니다.

**중요사항:** 모든 키 저장소에 동일한 비밀번호가 사용됩니다. 예제에서는 `password`로 표시됩니다.

### 프로시저

1. Build Forge 엔진이 설치되어 있는 호스트에 로그인하십시오.
2. 경로에 도구 디렉토리를 배치하십시오.
  - `<bfinstall>/openssl`
  - `<bfinstall>/ibmjdk/bin`
3. `LD_LIBRARY_PATH`에 `openssl` 디렉토리를 배치하십시오.

- `<bfinstall>/openssl`
4. buildForgeKeyStore.p12 키 저장소, 인증서 및 공용-개인 키 쌍을 작성하십시오.
    - a. 임시 디렉토리에서 keytool을 실행하여 키 저장소를 작성하십시오.
 

```
keytool -genkey -alias buildforge
-keyalg RSA -keysize 1024 -validity 5475 -dname "CN=hostname"
-keystore buildForgeKeyStore.p12
-storepass password
-storetype pkcs12
```
    - b. 키 저장소 파일(buildForgeKeyStore.p12)을 `<bfinstall>/keystore`로 복사하십시오. 이는 기존 파일을 겹쳐씁니다.
  5. 공용 인증서를 내보내십시오. `<bfinstall>/keystore` 디렉토리에서 이 명령을 실행하십시오.
 

```
keytool -export -alias buildforge
-file cert.der -keystore buildForgeKeyStore.p12
-storepass password
-storetype pkcs12
```
  6. 신뢰 저장소를 작성하십시오.
    - a. 임시 디렉토리에서 keytool을 실행하여 신뢰 저장소를 작성하십시오.
 

```
cd /temp
keytool -import -noprompt -trustcacerts -alias buildforge
-file cert.der -keystore buildForgeTrustStore.p12
-storepass password
-storetype pkcs12
```
    - b. 신뢰 저장소 파일(buildForgeTrustStore.p12)을 `<bfinstall>/keystore`에 복사하십시오. 이는 기존 파일을 겹쳐씁니다.
  7. buildForgeCert.pem에 공용 클라이언트 인증서를 배치하십시오. `<bfinstall>/keystore` 디렉토리에서 이 명령을 실행하십시오.
 

```
openssl pkcs12 -clcerts -nokeys
-in buildForgeKeyStore.p12 -passin pass:password
-out buildForgeCert.pem
```
  8. buildForgeKey.pem에 인증서 및 키를 배치하십시오. `<bfinstall>/keystore` 디렉토리에서 이 명령을 실행하십시오.
 

```
openssl pkcs12
-in buildForgeKeyStore.p12 -passin pass:password
-passout pass:password -out buildForgeKey.pem
```
  9. PEM CA(Certificate Authority) buildForgeCA.pem을 작성하십시오. 이는 buildForgeKey.pem의 사본입니다. `<bfinstall>/keystore` 디렉토리에서 이 명령을 실행하십시오.
 

```
cat buildForgeCert.pem > buildForgeCA.pem
```

## 다음에 수행할 작업

buildForgeKey.pem은 비밀번호로 보호된 PEM 키 저장소입니다. Apache 서버에서 시작 중에 비밀번호를 프롬프트합니다. 시작 중에 비밀번호에 대한 프롬프트를 표시하지 않으려면 Apache 서버에서 사용할 비밀번호로 보호되지 않는 PEM 키 저장소를 생성합니다.

개인 키에서 비밀번호를 제거하려면 다음 단계를 수행할 수 있습니다. Build Forge를 실행하는 프로세스의 ID에 액세스해야 하는 사용자가 buildForgeKeyForApache.pem 파일을 읽을 수 있는지 확인해야 합니다.

```
openssl rsa -in buildForgeKey.pem -passin pass:password  
-out buildForgeKeyForApache.pem
```

## 클라이언트 인증서를 사용하여 로그인:

클라이언트 인증서를 사용하여 로그인하는 기능을 구성할 때 두 가지 옵션이 제공됩니다.

### **WebSphere Application Server를 사용하여 Build Forge를 호스트하는 경우 로그인:**

#### 이 태스크 정보

WebSphere Application Server를 사용하여 Build Forge를 호스트하는 경우 몇 가지 수정사항을 사용할 수 있는 옵션이 제공됩니다. 이 옵션을 사용하려면 WebSphere가 인증을 수행한 후 인증된 프린시펄을 얻는 WebSphereSSOInterceptor가 필요합니다.

이 옵션을 사용하려면 다음 프로시저를 수정하십시오.

#### 프로시저

1. WAS 컨테이너 관리 인증에서 rbf-services에 대한 요청을 인증하도록 Build Forge WAR 파일을 보호하십시오. 이 보호를 설정하는 방법에 대한 정보는 118 페이지의 『사용자 정의 인터셉터를 사용하여 WAS 보안과 통합』의 내용을 참조하십시오.
2. rbf-services 웹 애플리케이션의 클라이언트 인증서 인증 매핑을 지원하도록 WebSphere Application Server를 구성하십시오. 다음 항목을 구성하여 이 지원을 설정하십시오.
  - 클라이언트 인증서에 적합한 SSL
  - 클라이언트 인증서에 적합한 rbf-services 웹 애플리케이션
  - 클라이언트 인증서에 적합한 웹 서버
  - 클라이언트 인증서에 적합한 LDAP 서버 매핑

WebSphere Application Server 구성 시 지침이 될 수 있는 WebSphere Portal의 클라이언트 인증서 인증 구성에 대한 정보는 <http://publib.boulder.ibm.com/infocenter/wpdoc/v6r0/index.jsp?topic=/com.ibm.wp.ent.doc/wpf/certauth.html>을 참조하십시오.



오. 1 - 3 단계를 수행하십시오. 해당 프로시저에서는 118 페이지의 『사용자 정의 인터셉터를 사용하여 WAS 보안과 통합』에서 수정하는 web.xml 파일을 참조합니다. `${WAS_INSTALL_ROOT}/profiles/${PROFILE_NAME}/installedApps/${CELL_NAME}/rbf-services_war.ear/rbf-services.war/WEB-INF/web.xml`에 있는 파일을 수정하십시오.

### 사용자 정의 **Build Forge SSO** 인터셉터를 기본으로 하여 로그인: 이 태스크 정보

이 옵션을 사용하려면 107 페이지의 『싱글 사인은 프레임워크 정보』의 설명대로 사용자 정의 SSO 인터셉터를 추가해야 합니다.

#### 프로시저

1. 사용자 정의 SSO 인터셉터가 다음과 같이 요청 속성에서 X509 인증서를 받는지 확인하십시오.

```
X509Certificate[]  
certs = (X509Certificate[])request.getAttribute("javax.net.ssl.peer_certificates");
```

2. 사용자 정의 SSO 인터셉터가 주제 DN의 일부를 LDAP 속성에 매핑하는지 확인하십시오. 일반적으로 공통 이름(CN)을 인증서에서 LDAP 디렉토리의 사용자 ID에 매핑합니다. Build Forge SSO 인터셉터의 `authenticateRequest` 메소드로 결과를 리턴하십시오.

```
responseAttributes.setStatus(HttpServletResponse.SC_OK);principal =  
mapCert(certs); /* Custom method to map from certificate to Build Forge LDAP principal. */  
result = new Result(Result.UseridOnlyOID, domain, principal);  
/* Specify the domain name of the LDAP server the principal exists in. */  
return result;
```

### 제한되지 않은 정책 파일 다운로드:

강력하게 암호화된 인증서를 사용하려면 먼저 JCE에 대한 제한되지 않은 정책 파일을 다운로드해야 합니다.

내보내기 및 가져오기 규정으로 인해 Installation Manager 및 Build Forge와 함께 제공된 기본 JCE 관할 정책 파일은 강력하면서도 제한된 암호만을 사용할 수 있도록 지원합니다. 내보내기에 승인된 암호화 강도를 초과하는 암호화가 포함된 인증서를 사용하려면 제한되지 않은 JCE 정책 파일을 다운로드하여 Installation Manager 및 Rational Build Forge용 JVM을 업데이트해야 합니다.

설치 중 사용할 인증서를 제공하려고 할 때 다음과 같은 오류가 발생하면 JCE 정책 파일을 업데이트해야 할 수 있습니다.

이 인증서를 확인할 수 없습니다. 인증서 없이 설치한 후 인증서를 추가하십시오.

**경고:** SDK(Software Development Kit) 업데이트가 포함된 Build Forge 업그레이드가 정책 파일을 겹쳐쓴 것 같습니다. 이 경우 업그레이드 후 제한되지 않은 정책 파일을 SDK에 다시 적용하십시오.



**중요사항:** 사용자의 출신 국가에 암호화 소프트웨어의 가져오기, 소유, 사용 또는 다른 국가로의 다시 내보내기에 대한 제약사항이 있을 수 있습니다. 제한되지 않은 정책 파일을 다운로드하거나 사용하기 전에 먼저 암호화 소프트웨어 가져오기, 소유, 사용 및 다시 내보내기와 관련한 해당 국가의 법률, 규정 및 정책을 확인하여 작업이 허용되는지 판단해야 합니다.

1. SDK 버전 1.4.2 이상의 제한되지 않은 JCE 정책 파일을 다운로드하십시오.
2. 파일의 압축을 푸십시오.
3. 압축을 푼 local\_policy.jar 파일을 해당 Build Forge 설치 디렉토리에 복사하십시오. 이렇게 하면 기존 파일을 겹쳐씹니다. 운영 체제 디렉토리를 사용하십시오.

- Windows 시스템

```
<bfinstall>\ibmjdk\jre\lib\security
```

- UNIX 및 Linux 시스템

```
<bfinstall>/Platform/ibmjdk/jre/lib/security
```

4. 다음 Build Forge 설치 시 인증서를 확인할 수 있도록 허용하려면 Installation Manager를 업데이트하십시오. 압축을 푼 local\_policy.jar 파일을 Installation Manager가 설치된 IBM JRE에 복사하십시오. 이렇게 하면 기존 파일을 겹쳐씹니다.

- Windows 시스템

```
<Minstall>\eclipse\jre_version\jre\lib\security
```

- UNIX 및 Linux 시스템

```
<Minstall>/eclipse/jre_version/jre/lib/security
```

## 비밀번호 암호화 사용

비밀번호 암호화를 사용하도록 Build Forge 시스템을 구성하면 시스템 보안이 강화됩니다.

또한 시스템에 대한 권한이 없는 액세스를 방지할 수 있도록 실제 보안이 강화됩니다.

**참고:** WebSphere 컴포넌트와 통합하는 경우 비밀번호 암호화를 구성하기 전에 먼저 비밀번호 암호화의 전제조건이 충족되는지 확인하십시오.

- WebSphere Application Server와의 통합은 615 페이지의 『Apache Tomcat 대신 WebSphere Application Server 사용』을 참조하십시오.
- IHS(IBM HTTP Server)와의 통합은 621 페이지의 『Apache HTTP Server 대신 IBM HTTP Server 사용』의 내용을 참조하십시오.

## Build Forge에서 비밀번호 보안 정보

Build Forge 시스템에서는 기본적으로 인코딩된 비밀번호가 사용되지만 추가 보안을 위해 암호화된 비밀번호를 사용할 수 있습니다.

비밀번호 암호화가 사용 가능한 경우 대칭 키 비밀번호 스키마로 사용할 수 있습니다. 비밀번호를 사용하는 클라이언트와 액세스하는 서비스 모두에 동일한 키를 사용해야 합니다.

- Build Forge 엔진 및 Build Forge 에이전트
- Build Forge 서비스 계층 및 Build Forge에서 사용되는 데이터베이스

Build Forge 시스템에서는 키가 `bfpwcrypt.conf` 파일에 보관됩니다. 파일은 에이전트 및 Build Forge의 설치 디렉토리(엔진의 경우)에 있습니다.

비밀번호 암호화에서는 대칭 키를 사용합니다. 공통 데이터베이스 비밀번호의 암호를 해독해야 하는 모든 시스템에 동일한 키가 필요합니다. 또한 엔진에서 암호화된 비밀번호를 받은 모든 에이전트에 엔진의 키가 필요합니다. 다중 엔진을 실행하는 경우(중복 구성) 에이전트에 각 엔진의 키가 필요합니다.

단순히 하나의 호스트에 하나의 Build Forge 관리 콘솔을 설치하고 다른 호스트에 하나의 에이전트를 설치하는 경우 비밀번호 암호화를 사용하려면 다음 프로시저를 수행해야 합니다.

- Build Forge 콘솔에서 비밀번호 암호화를 사용하도록 설정합니다(관리 > 보안).
- 현재 키를 파일로 내보냅니다. (이 키는 에이전트에서 사용되며, 다른 엔진에서도 사용됩니다.)
- 에이전트에 대한 새 키를 생성합니다. 해당 키를 파일로 내보냅니다. 이 키는 에이전트에서 해당 키 저장소 비밀번호를 암호화하기 위해 사용됩니다.
- 두 키를 모두 사용하여 에이전트의 `bfpwcrypt.conf`를 업데이트하십시오. 새 키를 마지막에 배치합니다.
- Build Forge 콘솔에서 에이전트를 사용하는 모든 서버 정의에 비밀번호 암호화를 사용하도록 설정합니다(서버 패널).
- Build Forge 호스트에서 `bfpwencrypt` 유틸리티를 사용하여 Build Forge에서 데이터베이스에 액세스하는 데 사용하는 비밀번호를 암호화합니다. `buildforge.conf`에서 인코딩된 현재 비밀번호를 암호화된 비밀번호로 바꿉니다.
- `buildforge.conf`의 서비스 계층 사본을 업데이트합니다. 99 페이지의 『Build Forge 구성 파일(buildforge.conf)』의 내용을 참조하십시오.

## 관리 콘솔에 비밀번호 암호화 사용

기타 모든 컴포넌트에서 비밀번호 암호화를 사용하려면 관리 콘솔에서 비밀번호 암호화를 사용해야 합니다.

- 중복 관리 콘솔에서는 첫 번째 관리 콘솔에서 `bfpwcrypt.conf` 파일에 포함할 비밀번호 키를 내보내야 합니다. 모든 관리 콘솔에서 동일한 키를 사용해야 합니다. 일반적으로 해당 키를 사용하여 데이터베이스 비밀번호를 암호화합니다.
- 에이전트에서는 엔진이 에이전트에 암호화된 비밀번호를 보내는 경우 비밀번호 키를 내보내야 합니다. 엔진과 에이전트 둘 다 동일한 키를 사용해야 합니다. 관리 콘솔에서 내보낸 키를 사용하여 암호화된 서버 인증 비밀번호를 암호 해독할 수 있습니다.

에이전트에서 암호화된 비밀번호가 수신되지 않지만 `BFAgent.conf` 파일에서 사용할 암호화된 비밀번호를 생성해야 하는 경우 각 에이전트에서 다른 비밀번호 키를 사용해야 합니다. 관리 콘솔에서 키를 각각 개별적으로 생성합니다.

관리 콘솔에서 비밀번호 암호화를 사용하려면 다음을 수행하십시오.

1. 콘솔에서 **관리 > 보안**으로 이동하십시오.
2. **비밀번호 암호화 사용**을 예로 설정하십시오.
3. **저장**을 클릭하십시오. 이 단계에서 Build Forge 데이터베이스에 구성이 저장됩니다.
4. 마스터 **BFClient.conf** 업데이트를 클릭하십시오. 이 단계에서 `bfclient.conf` 파일에 구성이 저장됩니다.
5. Build Forge를 다시 시작하십시오. 실행 중인 Build Forge에서 `bfclient.conf` 파일의 새 설정을 사용하려면 이 단계를 수행해야 합니다.

구성에서 **비밀번호 암호화 사용**이 예로 설정되어 있으면 다음을 수행할 수 있습니다.

- 비밀번호 키를 파일로 내보내기
- 새 비밀번호 키 생성
- `bfpwencrypt` 및 `bfagent` 명령을 실행하여 콘솔 및 에이전트 구성 파일에 포함할 암호화된 비밀번호 작성

## 비밀번호 키 관리

Build Forge 시스템에서는 기본적으로 인코드된 비밀번호가 사용되지만 추가 보안을 위해 암호화된 비밀번호를 사용할 수 있습니다.

### 비밀번호 암호화 키 파일:

`bfpwcrypt.conf` 파일에 비밀번호 암호화 구성 특성이 포함되어 있습니다.

Build Forge를 처음 시작하면 `bfclient.conf` 파일과 동일한 위치에 `bfpwcrypt.conf` 파일이 자동으로 생성됩니다.

- Windows: `<bfinstall>`
- UNIX 또는 Linux: `<bfinstall>/Platform`

**중요사항:** 이 파일의 이름을 바꾸지 마십시오. 파일 이름은 항상 `bfpwcrypt.conf`여야 합니다.

파일에 다음과 같은 특성이 포함되어 있습니다.

#### **bfpwcrypt\_key\_alias**

암호화 키의 별명입니다. 별명은 이 키로 암호화된 비밀번호의 일부입니다. 시스템에서 이를 사용하여 사용할 키를 판별합니다. 이 특성에 대한 다중 정의가 있을 수 있습니다(각 키에 하나). 마지막 정의가 비밀번호를 암호화하는 데 사용됩니다. 기타 모든 정의는 암호화된 비밀번호를 읽을 때 해당 비밀번호를 암호 해독하는 데 사용됩니다. 구성 파일, 데이터베이스 또는 에이전트와 엔진 간의 통신에 암호화된 비밀번호가 발견될 수 있습니다.

#### **bfpwcrypt\_key**

암호화된 마스터 키로, 128비트 AES 암호화를 사용하여 암호화됩니다. 이 키는 비밀번호를 암호화하는 데 사용됩니다.

#### **bfpwcrypt\_key\_password**

`bfpwcrypt_key`를 암호 해독하는 데 필요한 비밀번호입니다. 이 비밀번호는 인코딩되어 있습니다.

예제 `bfpwcrypt.conf` 파일:

```
***** Password Encryption Configuration Properties *****
bf_pwcrypt_key=MKuoiwD+MsWBFGl/2xeG0TEtpY+hAzXQu21fBcofM0M=
bf_pwcrypt_key_alias=8a679d430c401000b55e00007d1a7d1a
bf_pwcrypt_key_password=Tq0eDXc4G/bdaWeatKTYUx6Sw4S3i6wX
# Creation date=Thu Nov 20 03:44:48 CST 2008
# Origination host=myhost.mycompany.com
```

#### **비밀번호 키 내보내기:**

비밀번호 키를 다른 위치에 배치하려면 Build Forge 콘솔에서 해당 비밀번호 키를 내보냅니다.

Build Forge 엔진에서 사용된 비밀번호 키를 내보내어 다음 위치에 이를 배치하십시오.

- 동일한 데이터베이스에 액세스하는 기타 Build Forge 콘솔
- 비밀번호 암호화가 사용 가능한 서버 정의에서 식별된 에이전트를 호스팅하는 컴퓨터

비밀번호 키를 내보내려면 다음을 수행하십시오.

1. 콘솔에서 **관리 > 보안**으로 이동하십시오.

**참고:** 비밀번호 암호화 사용이 이미 예로 설정되어 있고, 마스터 `BFCClient.conf` 파일에서 설정을 저장하고 업데이트해야 합니다.

2. 키 파일 내보내기를 클릭하십시오. 파일을 저장할 위치를 묻는 메시지가 표시됩니다.
3. 위치를 지정한 다음 저장을 클릭하십시오.

키의 비밀번호를 암호 해독(다른 중복 관리 콘솔 및 에이전트)하거나 비밀번호를 암호화(에이전트)해야 하는 호스트로 파일을 이동하십시오. 파일의 콘텐츠(세 가지 모든 특성 설정)를 해당 호스트의 `bfpwcrypt.conf` 파일에 추가하십시오. 이 표에서는 콘텐츠의 배치에서 키를 사용하는 시점을 판별하는 방법을 설명합니다.

<code>bfpwcrypt.conf</code> 파일에 배치	이 경우 사용된 비밀번호 키
끝	<ul style="list-style-type: none"> <li>• 별명을 포함하는 비밀번호 복호화</li> <li>• 새 비밀번호 암호화</li> </ul>
입력의 위치	<ul style="list-style-type: none"> <li>• 별명을 포함하는 비밀번호 복호화</li> </ul>

**참고:** 새 키를 생성하면 `bfpwcrypt.conf` 파일의 끝 부분에 추가 키가 배치됩니다. 키를 내보낼 때 최신 키만 내보냅니다.

**새 비밀번호 키 생성:** 관리 콘솔에서 새 키를 생성하여 시스템 전체 키를 변경할 수 있습니다. 우수한 시스템 보안 상태를 유지보수하기 위해 이를 정기적으로 수행해야 합니다. 모든 엔진과 에이전트에서 새 키를 업데이트해야 합니다.

새 비밀번호 키를 생성하면 다음과 같이 새 키가 `bfpwcrypt.conf` 파일에 추가됩니다.

- 사용자 또는 콘솔 비밀번호가 생성되거나 이전 키를 사용하여 저장된 경우 해당 비밀번호가 계속해서 작동됩니다.
- **중요:** 이전 키를 내보내고 에이전트의 `bfpwcrypt.conf` 파일에 통합한 경우, 새로 생성된 키가 추가될 때까지 에이전트와 통신되지 않습니다.
- Build Forge에 저장되거나 `bfpwencrypt`로 생성된 모든 새 비밀번호에 새 키가 사용됩니다.
- 키 파일 내보내기를 수행하면 가장 최신 키만 내보냅니다.

새 비밀번호 키를 생성하려면 다음을 수행하십시오.

1. 콘솔에서 **관리 > 보안**으로 이동하십시오.

**참고:** 비밀번호 암호화 사용이 이미 예로 설정되어 있고, 마스터 `BFCClient.conf` 파일에서 설정을 저장하고 업데이트해야 합니다.

2. 새 키 생성을 클릭하십시오. 확인 메시지가 표시됩니다.
3. 예를 클릭하십시오.

## 에이전트에 비밀번호 암호화 사용

에이전트의 구성 파일에서 에이전트에 비밀번호 암호화를 사용할 수 있습니다.

에이전트에 비밀번호 암호화를 사용하려면 다음을 수행하십시오.

1. 에이전트가 실행 중인 경우 해당 에이전트를 중지하십시오.
2. 에이전트가 설치되어 있는 디렉토리로 이동하십시오.
3. `bfaagent.conf`를 편집하여 암호화 설정을 커십시오.

```
password_encrypt_module ./bfcrypt.dll;./bfpwcrypt.conf
```

에이전트가 설치되어 있는 위치가 아닌 디렉토리에서 에이전트가 실행되는 경우 파일을 직접 참조하도록 이 경로를 변경하십시오.

4. 관리 콘솔 암호화 키를 가져오십시오. 해당 키를 파일에 내보내십시오. 이 키는 에이전트에서 암호화된 서버 인증 비밀번호를 암호 해독하기 위해 필요합니다. 또한 로컬 키 저장소 비밀번호를 암호화하는 데에도 사용됩니다.
5. 해당 키를 에이전트의 `bfpwcrypt.conf` 파일에 배치하십시오. 에이전트에 대해 생성된 키를 파일에 마지막으로 배치하십시오. 비밀번호를 수동으로 암호화할 때 파일의 마지막 항목이 사용됩니다.
6. 에이전트를 시작하십시오.
7. 관리 콘솔에서 서버로 이동하십시오. 이 에이전트를 사용하는 모든 서버 정의에 대해 비밀번호 암호화 구성 특성을 사용으로 설정하십시오.
8. 연결 테스트를 클릭하여 암호화된 비밀번호로 연결이 작동되는지 확인하십시오.

## buildforge.conf 및 bfaagent.conf에서 비밀번호 암호화

내보낸 비밀번호를 사용하여 `buildforge.conf` 및 `bfaagent.conf`에서 사용할 암호화된 비밀번호를 빌드합니다.

`buildforge.conf` 파일에 Build Forge에서 데이터베이스에 액세스할 때 사용하는 사용자 이름 및 비밀번호(`db_password`)가 포함되어 있습니다. 일반적으로 해당 비밀번호가 인코드되어 있지만 암호화할 수 있습니다. 관리 콘솔의 비밀번호를 암호화하려면 다음을 수행하십시오.

1. 관리 콘솔 루트 디렉토리로 이동하십시오.
  - Windows: `<bfinstall>`
  - UNIX 또는 Linux: `<bfinstall>/Platform`
2. 다음 명령을 실행하십시오.

```
bfpwencrypt -e password
```

비밀번호에 대해 암호화할 일반 텍스트 비밀번호를 사용하십시오.

암호화된 비밀번호가 `stdout`에 보내집니다.

bfagent.conf 파일에 에이전트에서 키 저장소에 액세스할 때 사용하는 비밀번호 키 (ssl\_key\_password)가 포함되어 있습니다. 일반적으로 해당 비밀번호는 일반 텍스트이지만 암호화할 수 있습니다. 에이전트의 비밀번호를 암호화하려면 다음을 수행하십시오.

1. 에이전트 루트 디렉토리로 이동하십시오.
2. 다음 명령을 실행하십시오.

```
bfagent -e password
```

비밀번호에 대해 암호화할 일반 텍스트 비밀번호를 사용하십시오. 암호화된 비밀번호가 stdout에 보내집니다.

**참고:** AIX를 사용하고 GCC 라이브러리가 /lib 또는 /usr/lib에 없는 경우 "모듈 /usr/local/bin/bfcrypt.dll을 로드할 수 없음"을 알리는 오류가 발생할 수 있습니다. inetd.conf를 업데이트하여 이 문제를 해결할 수 있습니다.

오류를 정정하려면 다음을 수행하십시오.

- a. 다음 행을 찾으십시오.

```
bfagent stream tcp nowait root /usr/local/bin/bfagent bfagent
```

- b. 행을 다음 행으로 변경하십시오.

```
bfagent stream tcp nowait root /usr/bin/env env LIBPATH=path /usr/local/bin/bfagent
```

*path*는 GCC 라이브러리의 위치입니다.

- c. 명령 `refresh -s inetd`로 `inetd.conf`를 다시 로드하십시오.

암호화된 비밀번호는 `bfcrypt:` 문자열로 시작되고 비밀번호 키 별명이 중괄호로 묶이며, 그 뒤에 암호화(AES 128비트)되고 인코드(Base63)된 비밀번호가 표시됩니다. 인코드되고 암호화된 비밀번호의 예제는 다음과 같습니다.

Encoded:

```
dd8b42eed5cc051500f5bffe2b82b1aa6a67baee028a85d0cefa
```

Encrypted:

```
{bfcrypt:7427ab360c4010008f9d000049664966}drAIT1zLDGX/xRcvw65+B8aFpTqvmAdbmnh6FpwkHjU=
```

## 비밀번호 암호화에 대한 문제점 디버깅

암호화된 비밀번호에 문제점이 있는 경우 다음 절에 해당 문제점을 디버깅하는 방법에 대해 설명되어 있습니다.

### 콘솔에서 비밀번호 암호화 문제점 디버깅:

서비스 계층, 웹 인터페이스 및 엔진은 동일한 키 파일을 공유합니다.

세 가지 컴포넌트가 모두 동일한 호스트에 설치되어 있으면 해당 컴포넌트에서 동일한 키 파일을 사용합니다.

- Windows:



<bfinstall>/bfpwcrypt.conf

- UNIX 또는 Linux:

<bfinstall>/Platform/bfpwcrypt.conf

비밀번호 암호화를 사용하도록 설정한 후 문제점이 발생하면 다음 문제를 확인하십시오.

- 비밀번호 암호화를 사용하도록 설정한 후 Build Forge를 다시 시작했는지 확인하십시오. 모든 프로세스가 중지되고 올바르게 다시 시작되었는지 확인하십시오(Apache, Apache Tomcat 및 엔진).
- 중복 콘솔: 동일한 데이터베이스를 사용하여 다중 관리 콘솔을 설치한 경우 해당 콘솔에서 모두 동일한 bfpwcrypt.conf 파일을 사용해야 합니다. 가장 안전한 방법은 네트워크를 사용하지 않고 수동으로 분배하는 것입니다.
- 서버 정의: 콘솔에서 연결 테스트에 실패함 경우 키를 내보내고 bfaagent.conf 파일에 올바르게 배치해야 합니다. 비밀번호에 문제가 있는지 확인하려면 비밀번호 암호화를 사용하지 않도록 설정하고 연결 테스트를 시도해보십시오.
- 로그인: 비밀번호 암호화를 사용하도록 설정한 후 로그인할 수 없는 경우 Build Forge에서 bfclient.conf 및 buildforge.conf에 모두 올바른 bfpwcrypt.conf 키를 사용하고 있는지 확인하십시오. <bfinstall> 디렉토리 및 디렉토리의 서비스 계층 사본에서 buildforge.conf를 업데이트해야 합니다. 99 페이지의 『Build Forge 구성 파일(buildforge.conf)』의 내용을 참조하십시오.

이러한 확인을 모두 완료했지만 문제점이 지속되는 경우 추적을 사용하고 출력 로그를 검사해보십시오.

- 웹 인터페이스(UI): 환경 변수 BFDEBUG\_SECURITY=1을 설정합니다.

웹 인터페이스: 파일에 출력이 표시됩니다.

- Windows:

<bfinstall>/Apache/logs/php\_error.log

- UNIX 또는 Linux:

– <bfinstall>/server/apache/logs/php\_error.log

- 엔진: 포그라운드로 엔진을 시작합니다. 설치 디렉토리에서 bfengine -d를 실행하십시오. UNIX 또는 Linux의 경우 bfengine -d 2>&1 | tee out.txt를 사용하여 파일에 이를 파이프할 수 있습니다. Windows의 경우 tee 유틸리티를 가져오는 경우와 동일하게 수행할 수 있습니다.
- 서비스: 다음을 수행하십시오.
  1. Build Forge를 중지하십시오.
  2. 편집기에서 로그 파일을 여십시오.
    - Windows:



<bfinstall>/Apache/tomcat/common/classes/logging.properties

– UNIX 또는 Linux:

<bfinstall>/server/apache/tomcat/common/classes/logging.properties

3. 파일의 끝 부분에 다음 행을 추가하십시오.

com.buildforge.services.common.security.level=ALL

4. Build Forge를 시작하십시오.

5. 출력을 검사하십시오.

– Windows:

<bfinstall>/Apache/tomcat/logs/catalina.out

– UNIX 또는 Linux:

<bfinstall>/server/apache/tomcat/logs/catalina.out

### 에이전트에서 비밀번호 암호화 문제점 디버깅:

에이전트 통신 디버깅은 에이전트 및 엔진 컴포넌트와 관련되어 있습니다.

비밀번호 암호화를 사용하도록 설정한 후 문제점이 발생하면 다음 문제를 확인하십시오.

- bfaagent.conf를 확인하십시오. 다음 행의 주석이 해제되어야 합니다.

password\_encrypt\_module ./bfcrypt.dll;./bfpcrypt.conf

- 에이전트가 실행되는 디렉토리에 bfpcrypt.conf가 있는지 확인합니다. 하나 이상의 키 항목이 포함되어 있어야 합니다.

- 엔진의 bfpcrypt.conf에 있는 최종 항목이 에이전트의 bfpcrypt.conf에 있는지 확인합니다.

이러한 확인을 모두 완료했지만 문제점이 지속되는 경우 추적을 사용하고 출력 로그를 검사해보십시오. 추적을 사용하려면 다음을 수행하십시오.

1. bfaagent.conf를 여십시오.

2. 다음 행의 주석을 해제하십시오.

activity\_log bfaagent.log

bfaagent.log 대신 다른 경로를 지정할 수 있습니다.

### bfcrypt.dll의 경로 문제

bfaagent.conf에서 password\_encrypt\_module 특성을 bfcrypt.dll에 대한 올바른 경로에 지정해야 합니다. 예를 들어 다음과 같습니다.

password\_encrypt\_module /opt/buildforge/bfcrypt.dll

추적을 켜면 이 경로의 문제점이 다음과 같은 출력으로 표시됩니다.

```
[ 8928] bfcryptloader.c : 208: Loading password encryption module.
[ 8928] bfcryptloader.c : 223: module: bfcrypt
[ 8928] bfcryptloader.c : 232: Loading module: C:/BuildForge71.181.Agent/bfcrypt.dll
[ 8928] bfcryptloader.c : 262: Failed loading DLL, error code = 0
```

성공적으로 로드되면 다음과 같은 출력이 생성됩니다.

```
[ 12248] bfpwdlocloader.c: 134: Looking for password locator: ssl_key_password_locator
[ 12248] bfpwdlocloader.c: 244: Looking for password for prop ssl_key_password from bfagent.conf.
[ 12248] bfcryptloader.c : 208: Loading password encryption module.
[ 12248] bfcryptloader.c : 223: module: bfcrypt
[ 12248] bfcryptloader.c : 232: Loading module: ./bfcrypt.dll
[ 12248] bfcryptloader.c : 269: Loading procedure bfcrypt_init.
```

## 비밀번호 암호 해독 실패

잘못된 키 또는 다른 이유로 인해 비밀번호의 암호 해독에 실패한 경우 로그에 다음과 같은 행이 포함됩니다.

```
[ 4912] agent.c : 237: AUTH failed
```

비밀번호가 올바른 경우 문제점을 자세하게 진단할 수 있습니다. bfcrypt.dll 모듈에 디버깅을 사용합니다. 디버깅을 사용하려면 다음 환경 변수를 설정하십시오.

BFDEBUG\_SECURITY=1

에이전트가 서비스로 실행되는 경우 글로벌로 설정해야 합니다.

에이전트가 실행되는 디렉토리의 bfcrypt.txt에 디버그 출력이 배치됩니다.

다음 출력은 에이전트의 bfpwcrypt.conf에 올바른 키가 없음을 나타냅니다.

```
load_keys_from_file: Parsed 1 key configurations.decrypt: Looking for key matching info:
922492fe0c4010008304c3670e1e0e1e, length=32
decrypt: Comparing against: 4d553f110c401000ac08000051f651f6, length=32
decrypt: Warning! No matching key found.
```

다음 출력은 올바른 키 일치율을 나타냅니다.

```
load_keys_from_file: Parsed 2 key configurations.decrypt:
Looking for key matching info: 922492fe0c4010008304c3670e1e0e1e, length=32
decrypt: Comparing against: 7427ab360c4010008f9d000049664966, length=32
decrypt: Comparing against: 922492fe0c4010008304c3670e1e0e1e, length=32
decrypt: Found match: 922492fe0c4010008304c3670e1e0e1e, length=32
```

## bfclient.conf 참조

bfclient.conf 파일에 Build Forge 보안에 대한 설정이 저장됩니다. 파일은 Build Forge 설치 루트 디렉토리에 있습니다.

bfclient.conf 파일에 보안 통신(SSL) 및 비밀번호 암호화를 사용하는 데 사용된 설정이 포함되어 있습니다. 이 파일에는 다음 섹션이 포함되어 있습니다.

- 연결 특성
- 로그인 특성

- OpenSSL과 JSSE에서 모두 사용되는 SSL 특성
- OpenSSL에서만 사용되는 SSL 특성
- JSSE에서만 사용되는 SSL 특성
- 키 저장소 특성(bf\_keystore\_\*)
- 암호화 특성

표 1. 연결 특성

속성 이름	기본값	가능한 값	필수	설명
bf_services_hostname	설치 중에 지정	n/a	예	Build Forge 서비스 계층이 위치한 호스트 이름
bf_services_tcp_port	설치 중에 지정	n/a	예	Build Forge 서비스에 연결하기 위한 TCP 포트. SSL이 지정된 경우 사용됩니다.
bf_services_ssl_port	설치 중에 지정	n/a	예	Build Forge 서비스에 안전하게 연결할 수 있는 SSL 포트
bf_services_preferred_protocol	tcp	ssl, tcp	예	Perl 또는 PHP 클라이언트의 경우 연결할 SSL 또는 TCP를 지정합니다. Java 클라이언트의 경우 SecureAPIClientConnection 오브젝트가 SSL을 지정하고 APIClientConnection이 TCP를 지정합니다.

표 2. 로그인 특성

속성 이름	기본값	가능한 값	필수	설명
bf_login_user	없음	Build Forge 사용자 목록에서 UserID	아니오	로그인 ID로 사용됩니다. 클라이언트 프로그램 내에 로그인 ID를 지정할 수도 있습니다.
bf_login_password	없음	bf_login_user의 비밀번호	예, bf_login_user가 사용되는 경우	bf_login_user의 비밀번호
bf_login_realm	없음	LDAP 도메인 이름	아니오	사용자가 사용자 테이블에 없는 경우 조회할 LDAP 도메인

표 3. OpenSSL과 JSSE에서 모두 사용되는 SSL 특성

속성 이름	기본값	가능한 값	필수	설명
bf_ssl_usage	없음	jsse, openssl	예	SSL 구현을 선택합니다. 선택사항에 따라 다른 특성을 사용할 수 있습니다.
bf_ssl_cipher_group	모두	ALL, HIGH, MEDIUM, LOW	아니오	SSL 데이터 교환을 수행하는 동안 제공할 암호 그룹을 지정합니다. HIGH는 가장 안전하고, LOW는 최상의 성능을 제공하며, ALL은 가장 상호운용 가능합니다.
bf_ssl_cipher_override	없음	제공하는 암호 스위트	아니오	bf_ssl_cipher_group을 겹쳐씹니다. SSL 데이터 교환을 수행하는 동안 더 작은 암호 세트를 선택하는 데 사용될 수 있습니다.
bf_ssl_protocol	TLSv1	TLSv1, SSLv3. 구현에 따라 다를 수 있습니다.	아니오	SSL에서 사용되는 데이터 교환 프로토콜입니다. 선호하는 프로토콜은 TLSv1입니다.

표 3. OpenSSL과 JSSE에서 모두 사용되는 SSL 특성 (계속)

속성 이름	기본값	가능한 값	필수	설명
bf_ssl_cert_alias	없음	구성된 키 저장소에 있는 올바른 인증 별명	아니오	사용할 인증서를 지정합니다. 동일한 키 저장소에 다중 인증서가 있는 경우 이를 수행할 수 있습니다.

표 4. openssl에서만 사용되는 SSL 특성

속성 이름	기본값	가능한 값	필수	설명
bf_ssl_key_ref	openssl_key	개인용 키가 포함되어 있는 모든 올바른 PEM 키 저장소 참조	아니오	서버에 연결할 때 사용할 클라이언트의 개인용 키가 포함되어 있는 키 저장소 구성에 대한 참조입니다. 사용하는 경우 bf_ssl_cert_ref에서 이 개인용 키에 대한 올바른 인증서를 지정해야 합니다. 서버가 개인 인증서를 요청하도록 설정되어 있는 경우에만 사용됩니다.
bf_ssl_cert_ref	openssl_cert	지정된 개인 키에 대한 인증서가 포함되어 모든 올바른 PEM 키 저장소 참조	아니오	위의 개인용 키에 대한 인증서가 포함되어 있는 키 저장소 구성에 대한 참조입니다. 서버가 개인 인증서를 요청하도록 설정되어 있는 경우에만 사용됩니다.
bf_ssl_ca_ref	openssl_ca	이 클라이언트가 연결되는 서버 인증을 유효성 검증하는 데 사용되는 하나 이상의 인증서가 포함되어 있는 모든 올바른 PEM 키 저장소 참조	예	SSL 데이터 교환을 수행하는 동안 서버 인증을 유효성 검증하는 데 사용되는 하나 이상의 서명자 인증서가 포함되어 있는 키 저장소 구성에 대한 참조입니다. 인증은 CA 루트, 중간 또는 자가 서명될 수 있습니다.

표 5. JSSE에서만 사용되는 SSL 특성

속성 이름	기본값	가능한 값	필수	설명
bf_ssl_keystore_ref	jsse_keystore	keyEntry(개인 키 및 인증서)가 포함되어 있는 모든 올바른 PKCS12, JKS 또는 JCEKS 키 저장소 참조	아니오	서버에 연결할 때 사용할 클라이언트의 개인 인증서(개인용 키 및 연관된 인증서)가 포함되어 있는 키 저장소 구성에 대한 참조입니다. 이는 서버에서 클라이언트 인증을 위해 개인 인증서를 요청하는 경우에만 필요합니다.
bf_ssl_truststore_ref	jsse_truststore	keyEntry(개인 키 및 인증서)가 포함되어 있는 모든 올바른 PKCS12, JKS 또는 JCEKS 키 저장소 참조	예	SSL 데이터 교환을 수행하는 동안 서버 인증을 유효성 검증하는 데 사용되는 서명자 인증서가 포함되어 있는 키 저장소 구성에 대한 참조입니다. 키 저장소에 다른 인증 서명을 유효성 검증하는 데 사용되는 인증서인 하나 이상의 trustedCertEntries가 포함되어 있습니다.

표 6. 키 저장소 특성

속성 이름	기본값	가능한 값	필수	설명
bf_keystore_alias	다양	문자열	예	이는 SSL 구성에서 키 저장소 구성을 참조하는 데 사용하는 이름입니다.
bf_keystore_location	다양	지정된 유형의 키 저장소에 대한 상대 또는 완전한 경로	예	이는 지정된 유형의 키 저장소에 대한 경로 및 위치입니다. 경로는 상대 경로일 수 있지만, 시작 디렉토리에 대해 올바른 경로여야 합니다.

표 6. 키 저장소 특성 (계속)

속성 이름	기본값	가능한 값	필수	설명
bf_keystore_type	openssl의 경우 PEM 및 jsse의 경우 PKCS12	OpenSSL의 경우 PEM이고, "jsse"의 경우 PKCS12, JCEKS 또는 JKS입니다.	예	키 저장소의 유형입니다. bf_keystore_location 특성에서 참조되는 실제 키 저장소 유형과 일치해야 합니다.
bf_keystore_password	설치 중에 지정	키 저장소 유형에서 지원되는 문자열입니다. 일부 키 저장소에서는 비ASCII 문자열이 지원되지 않습니다.	아니오	키 저장소에 액세스할 수 있는 비밀번호입니다. OpenSSL의 경우 일반적으로 공용 키만 포함되어 있는 Cert 및 CA 키 저장소에 대한 비밀번호가 필요하지 않습니다.

표 7. 암호화 특성

속성 이름	기본값	가능한 값	필수	설명
bf_pw_crypt_enabled	false	true, false	아니오	비밀번호가 인코드(false)되거나 암호화(true)되는지 여부를 지정합니다. 사용할 경우, 비밀번호 암호화 구현에서 bfclient.conf와 동일한 디렉토리에 있는 bfpwcrypt.conf라는 파일을 사용합니다.



---

## 제 10 장 에이전트 설치

이 절에서는 에이전트의 설치, 실행, 구성 및 문제점 해결 방법에 대해 설명합니다.

관리 콘솔에서 서버 자원으로 사용할 각 호스트에 에이전트를 설치합니다. 에이전트는 관리 콘솔에서 프로젝트 및 단계 실행 요청을 수신하는 서비스입니다.

---

### Windows 플랫폼에 에이전트 설치

Windows 플랫폼에 에이전트를 설치하려면 다음을 수행하십시오.

1. 설치 매체에서 에이전트 설치 프로그램을 찾아 시작하십시오. 설치 프로그램의 파일 이름은 `win-bfagent-version.exe`입니다.

**팁: Rational Build Forge 에이전트 설치를 선택하면 런치패드가 이 설치 프로세스를 시작합니다.**

2. 설치 프로그램이 기존 버전의 에이전트를 감지하면 기존 버전을 겹쳐 쓸 것인지 프롬프트합니다. **확인**을 클릭하십시오. 기본값: **확인**
3. 시작 메시지가 열리면 다음을 클릭하십시오.
4. 라이선스 계약에 동의하는 경우 **동의함**을 클릭하십시오.
5. 설치 위치 선택 창에서 대상 폴더를 설정하고 다음을 클릭하십시오. 파일을 쉽게 찾을 수 있도록 기본 위치인 `C:\Program Files\IBM\Build Forge\Agent`를 사용하십시오.
6. 구성 창에서 원하는 에이전트 옵션을 선택한 후 설치를 클릭하십시오.
7. 다음 설치 방법 중 하나를 선택하십시오.
  - 서비스로 설치
  - 사용자 모드 에이전트 설치

에이전트에서 GUI 애플리케이션을 실행할 수 있는 경우에만 사용자 모드 에이전트를 선택합니다.

8. 선택사항: **Cygwin** 지원 사용을 클릭하십시오.

**팁:** Cygwin Linux<sup>®</sup> 에뮬레이션 환경을 사용하는 경우, 에이전트를 설치할 때 Cygwin 지원을 설치하도록 선택할 수 있습니다. Cygwin 지원을 설치하는 경우, 다음 단계를 수행하십시오.

- a. Cygwin을 설치하는 동안 **DOS/text** 행 끝을 선택하십시오.
- b. 프로젝트에서 명령에 **UNIX<sup>®</sup>** 스타일 구문을 사용하십시오.

**중요사항:**

Cygwin은 US ASCII로만 작동합니다. UTF-8을 지원하지 않으므로 다른 시스템에서 사용할 수 없습니다.

9. 에이전트에서 관리 콘솔과 통신하기 위해 사용하는 포트를 지정하십시오. 기본 포트는 5555입니다.

10. 설정 완료 패널에서 완료를 클릭하십시오.

**참고:** 설치하는 동안 팝업 창을 닫지 마십시오. 설치를 실행하는 동안 팝업 창이 표시되었다가 없어지도록 두십시오.

## 사용자 모드 에이전트

사용자 모드로 설치된 에이전트에서는 프로젝트에 의해 실행된 애플리케이션과 상호작용할 수 있습니다.

사용자 모드는 Windows 에이전트용으로만 제공되는 옵션입니다. 이 옵션은 설치 중에 설정됩니다. 설치 후에는 이 옵션을 구성할 수 없습니다. 사용자 모드에는 다음 애플리케이션이 있습니다.

- 작업 실행 시 GUI 애플리케이션을 통해 입력을 수동으로 수집합니다. 그러므로, 사용자 입력에 따라 작업이 실행됩니다.
- 프로젝트 및 단계의 문제점을 해결합니다. 서비스 모드 오퍼레이션 중에 숨겨진 출력은 사용자 모드에서 볼 수 있습니다. 각 단계는 단계 실행 중에 콘솔 창을 생성합니다.

**중요사항:** Windows 7 시스템. Windows 7 시스템에서 사용자 모드 에이전트를 설치하거나 다시 설치하는 경우 운영 체제를 다시 부팅해야 합니다. 운영 체제를 다시 부팅하지 않으면 에이전트가 작동하지 않습니다.

사용자 모드에서 컴퓨터를 사용하도록 프로젝트를 설정할 때 다음 차이점을 명심하십시오.

- 사용자 모드 에이전트는 시스템에 현재 로그인한 사용자로 작동합니다. 에이전트는 이 사용자가 컴퓨터에 로그인한 경우에만 활성화됩니다. 사용자가 로그아웃한 경우에는 사용자 모드 에이전트를 실행하는 서버를 사용할 수 없습니다.
- 사용자 모드의 시스템에서 실행되는 단계는 컴퓨터의 모든 사용자가 볼 수 있습니다.
  - 각 단계는 에이전트가 실행 중인 Windows 컴퓨터에서 콘솔 창을 엽니다. 관리 콘솔에서 명령 활동을 표시합니다.
  - 단계에서 GUI 애플리케이션을 실행하면, 에이전트가 실행 중인 Windows 컴퓨터에 애플리케이션 창이 표시됩니다. 관리 콘솔은 단계를 계속하기 전에 애플리케이션이 종료될 때까지 기다립니다.



대안: 기다리지 않고 작업을 계속하려면 시작 명령을 사용하십시오.

- `_USE_BFCREDS` 변수를 사용하지 마십시오. 이 변수를 사용하는 단계는 실패합니다.
- 사용자는 다음 특권을 가져야 합니다. 이 특권은 일반적으로 기본적으로 사용할 수 없습니다. 특권은 명시적으로 추가되어야 합니다.

```
SeInteractiveLogonRight
SeAssignPrimaryTokenPrivilege
SeImpersonatePrivilege
SeIncreaseQuotaPrivilege
SeTcbPrivilege
```

## Windows 운영 체제에서 자동 에이전트 설치 수행

Windows에서 에이전트를 자동 설치하려면 `/S`(대문자 S) 옵션을 사용합니다. 예를 들어, 명령 프롬프트에 다음 명령을 입력합니다. 옵션은 대소문자를 구분합니다.

```
win-bfagent-7.0.1.2305.exe /S
```

자동 설치의 다음 설정을 사용합니다. 이 설정은 수정할 수 없습니다.

- 기존 설치 겹쳐쓰기: yes
- 설치 위치: `C:\Program Files\IBM\Build Forge\Agent`
- 서비스로 설치: yes
- Cygwin 지원: no
- 포트: 5555

---

## UNIX 및 Linux 시스템에 에이전트 설치

플랫폼의 설치 지시사항을 따르십시오.

### • AIX

1. `aix5-bfagent-<version>.tar.gz` 파일 또는 `aix5np-bfagent-<version>.tar.gz` 파일을 사용하십시오.

`aix5np` 파일은 PAM 인증에 대한 지원을 포함하지 않습니다.

#### 중요사항:

루트로 실행되도록 `aixnp` 에이전트를 설치하는 경우, 런타임 시 에이전트는 지정된 서버 인증 신임을 사용하여 인증하기 위해 AIX 인증 호출을 사용합니다. 에이전트를 루트로 실행되지 않도록 설치하는 경우 `bfagent.conf` 파일의 `magic_login` 설정을 사용하여 액세스 권한을 제한해야 합니다.

2. 다음 명령을 입력하여 파일을 추출하십시오.

```
gzip -d gzipfilename.gz
```

3. 다음 명령을 입력하여 파일을 추출하십시오.

```
tar xvf tarfilename.tar
```

4. 다음 명령을 입력하여 에이전트를 설치하십시오.

```
cd extracted-agent-directory
./install.sh
```

**중요사항:** 에이전트가 구성 옵션 `--without-pam`을 사용하여 AIX에 대해 컴파일되는 경우, 에이전트에 대한 인증이 해제됩니다.

에이전트가 루트 특권으로 설치되는 경우, 지정한 비밀번호와 상관없이 모든 사용자가 올바른 사용자로 연결됩니다.

PAM을 사용하지 않는 AIX 시스템에서 실행되도록 에이전트를 컴파일해야 하는 경우, 에이전트에 대한 전용 계정을 사용하여 해당 사용자로 실행하도록 이를 설치하고 `bfagent.conf` 파일의 `magic_login` 설정을 사용하여 액세스 권한을 제한하십시오.

#### • HP-UX

1. 사용자가 프로젝트에서 지정하는 모든 운영 체제 명령을 Rational® Build Forge® 에이전트가 찾을 수 있는지 확인하십시오. 다음 옵션 중 하나를 사용하여 이 테스트를 완수할 수 있습니다.

- 모든 필수 디렉토리를 포함하도록 시스템의 `/etc/PATH` 파일을 편집하십시오.
- HP-UX 시스템의 `bfagent.conf` 파일에서 `nologonshell` 설정을 지정하십시오. 이 설정은 Build Forge®가 `/etc/profile`을 처리하지 못하도록 합니다. Build Forge® 환경의 `/etc/profile`에서 `PATH`와 각 프로젝트에 적용하는 기타 필수 설정에 대한 값을 지정해야 합니다.

2. 설치 매체에서 `hpux11-bfagent-<version>.tar.gz` 파일을 가져오십시오. 이 파일을 에이전트가 설치된 디렉토리에 넣으십시오.
3. 파일을 추출하십시오.

```
gzip -d gzipfilename.gz
```

4. tar 파일에서 추출하십시오.

```
tar xvf tarfilename.tar
```

5. 에이전트를 설치하십시오.

```
cd extracted-agent-directory
./install.sh
```

6. 에이전트가 비대화형 로그인 셸에서 명령을 실행할 수 있도록 하려면 `/etc/profile`의 다음 행을 수정하십시오.

```
if [ ! $VUE]
```

행을 다음으로 변경하십시오.

```
if [ -z "$VUE" -a -n "$PS1" ]
```

- **Mac OS**

1. 설치 매체에서 `mac-bfagent-<version>.dmg` 파일을 가져와서 원하는 위치에 놓으십시오.
2. 파일을 두 번 클릭하여 컴포넌트를 추출하십시오.

- **Red Hat Linux 및 SuSE Linux**

1. 설치 매체에서 `rh9-bfagent-<version>.rpm`(Red Hat Enterprise Linux 4 또는 SuSE Linux 10) 또는 `rhel5-bfagent-<version>.rpm` 파일(Red Hat Enterprise Linux 5)을 가져오십시오.

2. 파일을 설치하려면 RPM 유틸리티를 사용하십시오.

```
rpm -iUvh rh9-bfagent-version.rpm
```

- **Solaris**

1. Sparc의 Solaris 9 또는 Solaris 10의 경우 `sol9-bfagent-<version>-sparc-opt.gz` 파일을 사용하십시오.

2. 패키지 압축 풀기:

```
gzip -d solN-bfagent-<version>-platform-opt.gz
```

3. 명령 프롬프트에 다음 명령을 입력하십시오.

```
pkgadd -d ./unzipped-package
```

- **기타 플랫폼 - 소스에서 컴파일**

다른 플랫폼의 에이전트가 필요한 경우, `src-bfagent-<version>.tar.gz` 파일을 사용하여 소스에서 에이전트를 컴파일하십시오.

1. 다운로드한 tar 파일을 추출하십시오.
2. src 디렉토리에 있는 구성 스크립트를 실행하십시오.

SSL을 구성하려면 옵션은 다음과 같습니다.

```
./configure --with-ssl=/usr/include/openssl
```

SSL 및 비밀번호 암호화 둘 다에 대해 구성하려면 옵션은 다음과 같습니다.

```
./configure --with-ssl=/usr/include/openssl --enable-bfcrypt-dll
```

3. src 디렉토리에서 **make** 명령을 실행하십시오.

소스 팩에 GNU C 컴파일러 또는 시스템의 C 컴파일러가 필요합니다. 로컬 컴퓨터의 설치 프로그램이 포함되어 있지 않은 소스 팩 및 사전 빌드된 에이전트에 컴퓨터의 시스템 `inetd/xinetd` 구성 파일에 에이전트를 설치할 수 있는 설치 프로그램이 포함됩니다.

---

## System i 플랫폼에 에이전트 설치

다음 지시사항을 사용하여 System i에 에이전트를 수동으로 설치합니다.

7단계에서의 명령 스크립트가 시작 시 실행되는 작업 설명을 작성하고 \*ALLOBJ 특수 권한이 있는 BAGENT 사용자로 에이전트를 시작합니다.

- 관리 콘솔에서 지정한 서버 인증 신임을 사용하여 \*ALLOBJ 특수 권한이 있는 사용자 또는 QSECOFR 사용자를 인증할 수 있습니다.
- 이러한 특권이 없는 사용자를 인증하려면 bfaagent.conf 파일에 magic\_login 설정을 구성해야 합니다. 세부사항은 196 페이지의 『bfaagent.conf 참조』를 참조하십시오.

System i 플랫폼에 에이전트를 설치하려면 다음을 수행하십시오.

1. 제품 설치 매체 또는 다운로드 이미지를 사용하여 iseries-bfaagent-*<version>*.tar.gz 파일을 찾으십시오.
2. 다음 명령을 입력하여 아카이브에서 tar 파일을 추출하십시오.  

```
gzip -d iseries-bfaagent-<version>.tar.gz
```
3. tar 파일에서 파일을 추출하십시오.  

```
tar xvf iseries-bfaagent-<version>.tar
```
4. iSeries 서버에서 bfaagent 실행 파일을 에이전트 설치 디렉토리에 배치하십시오(예: /bin).
5. iSeries 서버에서 bfaagent.conf 파일을 /etc에 배치하십시오.
6. 다음 예제에 표시된 대로 bfaagent.conf 파일에서 셸 옵션의 주석을 해제하고 PASE에 대한 기본 셸을 지정하거나 원하는 셸을 지정하십시오.  

```
shell /bin/sh
```
7. 시작 시 BAGENT 사용자로 실행되도록 System i를 구성하십시오.

다음 명령을 입력하여 \*ALLOBJ 특수 권한이 있는 BAGENT 사용자를 작성하고 시작 시 BAGENT 사용자로 실행되는 작업 설명을 작성하십시오. 다음 예제에서는 bfaagent 실행 파일이 /bin에 설치되어 있습니다.

CRTLIB BAGENT

CRTSBSD SBSD(BAGENT/BAGENT) POOLS((1 \*BASE)) TEXT('Build Forge Agent subsystem')

CRTJOBQ JOBQ(BAGENT/BFAJOBQ) TEXT('Build Forge Agent job queue')

CRTUSRPRF USRPRF(BAGENT) PASSWORD(\*NONE) INLMNU(\*SIGNOFF) LMTCPB(\*YES) SPCAUT(\*ALLOBJ)  
TEXT('Build Forge Agent user profile')

CRTJOBQ JOBQ(BAGENT/BFAJOBQ) JOBQ(BAGENT/BFAJOBQ)  
TEXT('Build Forge Agent autostart')USER(BAGENT) RQSDTA('CALL PGM(QP2SHELL)  
PARM('/bin/bfaagent' ' ' -s'))

CRTCLS CLS(BAGENT/BFACLS) TEXT('Build Forge Agent job class')

ADDRTGE SBSD(BAGENT/BAGENT) SEQNBR(1) CMPVAL(\*ANY) PGM(QCMD) CLS(BAGENT/BFACLS)

ADDJOBQE SBSD(BFAGENT/BFAGENT) JOBQ(BFAGENT/BFAJOBQ) MAXACT(\*NOMAX) SEQNBR(10)

ADDAJE SBSD(BFAGENT/BFAGENT) JOB(BFAGENT) JOBD(BFAGENT/BFAJOBQ)

---

## System z 플랫폼에 에이전트 설치 및 실행

다음 지시사항을 수행하여 System z에서 Build Forge 에이전트 소스 코드를 수동으로 추출하고 컴파일하십시오. z/OS에 대한 에이전트 소스 코드는 컴파일되지 않은 소스만 제공됩니다. 2진 분배는 사용할 수 없습니다.

다음 소프트웨어와 프로그램이 필요합니다.

- c89 컴파일러 및 Unix 헤더 파일. z/OS 시스템의 경우 Unix 시스템 서비스(USS) 환경에서 에이전트가 실행됩니다.
- z/OS UNIX 셸 인터페이스. 설치 중에 z/OS UNIX 셸의 z/OS에서 모든 명령을 실행합니다.
- gzip 유틸리티.

참고: z/OS 시스템에서 gzip을 사용할 수 있는 경우, 소스 팩을 z/OS에 전송한 후 z/OS 시스템에서 tar 파일을 추출할 수 있습니다. 그렇지 않은 경우에는 우선 비z/OS 컴퓨터에서 파일을 추출한 후 z/OS 시스템으로 전송해야 합니다.

- z/OS용 Build Forge 에이전트 소스 팩은 src-bfagent-*<version>*.tar.gz입니다.

### 전제조건

- 클라이언트 및 내부 통신에 SSL을 사용하도록 설정하십시오. 129 페이지의 『클라이언트 및 내부 연결에 SSL 사용』의 내용을 참조하십시오.
- 시스템 SSL 라이브러리가 설치되어야 합니다.
- 에이전트 코드를 컴파일할 경우 GSKit의 최신 버전을 사용하십시오.

System z 플랫폼에 에이전트를 설치하려면 다음을 수행하십시오.

1. 제품 설치 매체 또는 다운로드 제품 이미지를 사용하여 에이전트 소스 팩의 파일인 src-bfagent-*<version>*.tar.gz를 찾으십시오.

비z/OS 컴퓨터에서 소스 팩을 디렉토리에 복사하거나 다운로드하십시오.

2. 비z/OS 컴퓨터의 셸 프롬프트에서 다음 명령을 입력하여 에이전트 소스 팩에서 tar 파일을 추출하십시오.

```
gzip -d src-bfagent-<version>tar.gz
```

3. ftp 또는 다른 전송 방법을 사용하여 tar 파일을 2진 이미지로 z/OS 시스템에 전송하고 해당 파일을 전용 HFS 서브디렉토리에 배치하십시오(일반적으로 사용자 계정의 USS 홈 디렉토리).
4. z/OS 시스템에서 다음 명령을 실행하여 에이전트 소스 코드를 빌드하십시오.

- a. 첫 번째 단계: `.configure-zos`. `--with-system-ssl` 사용을 확인합니다. 시스템 SSL에 대한 경로를 제공합니다. GSKit의 최신 버전을 사용하십시오.

```
pax -rf src-bfagent-<version>.tar -ofrom=ISO8859-1,to=IBM-1047
cd bfagent-<version>/src
./configure-zos --with-system-ssl=<path>
```

- b. `.configure-zos` 스크립트를 완료한 후 다음 명령을 실행하십시오.
- ```
./build-zos
```

참고: `.build-zos` 스크립트를 실행한 후 오류 메시지가 표시되면 169 페이지의 『z/OS에서 에이전트 설치 문제점 해결』의 내용을 참조하십시오.

5. z/OS 시스템에서 `bfagent.conf` 파일을 `/etc`에 배치하십시오.

`bfagent.conf`가 `/etc`에 없는 경우 `-f` 옵션을 사용하여 에이전트를 시작해야 합니다. 193 페이지의 『bfagent 참조』의 내용을 참조하십시오.

6. z/OS 시스템에서 `bfagent` 실행 파일을 적절한 위치에 배치하십시오(예: `/usr/bin` or `/usr/local/bin`).

7. z/OS 시스템에서 다음 명령을 루트로 실행하십시오.

```
# extattr +p -s bfagent
```

8. 시스템 SSL을 사용하는 경우 시스템 SSL 라이브러리를 PROGRAM CONTROL에 추가하십시오. 다음 명령을 사용하십시오.

```
ralter program * addmem('SYS1.SIEALNKE'//nopadchk) uacc(read)
```

9. z/OS 시스템에서 루트로 로그인하고 `-s` 옵션을 사용하여 에이전트를 수동으로 시작하십시오.

```
bfagent -s
```

보안 정책 때문에 루트로 로그인할 수 없는 경우 196 페이지의 『bfagent.conf 참조』 및 `bfagent.conf`의 `magic_login` 설정에 대한 지시사항을 참조하십시오.

에이전트가 독립형 디먼으로 실행되고 기본 에이전트 포트 5555를 사용합니다. 기본 포트를 변경하려면 `bfagent.conf`에서 포트 설정을 사용하십시오. 193 페이지의 『bfagent 참조』의 내용을 참조하십시오.

참고: z/OS 시스템에 Unix TCP/IP 디먼(`inetd` 또는 `xinetd`)이 설치되고 활성 상태인 경우 Build Forge 에이전트를 서비스로 실행되도록 설정하고 자동으로 시작할 수 있습니다. 190 페이지의 『UNIX, Linux 및 MacOS에서 에이전트 실행』의 내용을 참조하십시오.

10. z/OS 시스템에서 Telnet 명령을 사용하여 연결을 테스트하십시오. 208 페이지의 『연결 테스트』의 내용을 참조하십시오.

## z/OS에서 에이전트 설치 문제점 해결

z/OS에 에이전트 소스 코드를 빌드한 후 오류 메시지가 표시될 수 있습니다. 이 주제에서는 몇 가지 일반적인 오류의 수정사항에 대해 설명합니다.

configure-zos 스크립트는 몇 가지 공통 값을 설정하고 몇 가지 기본 검사를 수행하여 시스템에서 사용할 수 있는 헤더 및 기능을 식별합니다.

z/OS 시스템 구성의 변형으로 인해 ./configure-zos 스크립트를 오류 없이 실행할 수 있으나 ./build-zos 스크립트 실행 시 다음과 같은 오류가 발생할 수 있습니다.

**CEE3501S CCNDRVR** 모듈을 찾을 수 없습니다.

FSUM3066 COMPILE 단계가 다음 리턴 코드와 함께 종료되었습니다.

-1: EDC5083I 모듈을 스토리지에 로드하는 중에 오류가 발생했습니다.

이 오류는 컴파일러에서 필수 동적 라이브러리를 로드할 수 없음을 나타냅니다.

다음 명령을 실행하십시오. % export STEPLIB="SYS1.SCCNCMP"

./build-zos 명령을 다시 실행하십시오. 명령이 다시 실패하면 시스템 관리자에게 필수 라이브러리를 찾을 수 있도록 도움을 요청하십시오.

**IKJ56228I** 데이터 세트 **CEE.SCEE0BJ**가 카탈로그에 없거나 카탈로그에 액세스할 수 없음

FSUM3066 COMPILE 단계가 다음 리턴 코드와 함께 종료되었습니다.

FSUM3052 데이터 정의 이름 C8961을 해석할 수 없습니다. 데이터 세트를 찾을 수 없습니다. 데이터 세트 이름 CEE.SCEE0BJ가 올바르게 지정되어 있는지 확인하십시오.

이 오류는 링커가 컴파일을 완료해야 하는 시스템 라이브러리를 찾을 수 없음을 나타냅니다. 다음 명령을 실행하십시오.

% export \_C89\_LSYSLIB=SYS1.SCEELKED:SYS1.SCEELKEX

% export \_C89\_PSYSLIB=SYS1.SCEE0BJ

./build-zos 명령을 다시 실행하십시오. 명령이 다시 실패하면 시스템 관리자에게 필수 라이브러리를 찾을 수 있도록 도움을 요청하십시오.

**IEW2456E 9207 xxx** 기호가 해석되지 않음

해석되지 않은 기호 오류는 빌드에서 실제로 없는 system C 라이브러리에 의해 기호가 정의되는 것으로 간주함을 나타냅니다. 대부분의 경우 이는 다른 시스템에서도 종종 누락되는 기호이며, config.h에 문제점을 해결할 수 있는 설정이 있습니다.



예를 들어, 시스템에서 `unsetenv` 기능을 정의하지 않을 수 있습니다. 일반적으로 `configure-zos` 스크립트에서 이를 감지해야 합니다. 감지하지 못하는 경우, 다음과 같이 에이전트 소스 팩과 함께 제공되는 `config.h` 파일을 편집합니다.

```
#define HAVE_UNSETENV 1을 #undef HAVE_UNSETENV로 변경하십시오.  
./build-zos 명령을 다시 실행하여 문제점을 정정하십시오.
```

참고: 기타 기능에 대한 유사 `#define` 문이 있습니다.

---

## 에이전트 업데이트 및 배치 유틸리티

에이전트 업데이트 및 배치 유틸리티(**agentupdate**)는 다중 컴퓨터에서의 IBM Rational Build Forge 에이전트 설치 및 업데이트를 단순화합니다.

Rational Build Forge 에이전트는 Rational Build Forge가 태스크를 자동화하는 모든 컴퓨터에 필요합니다. 여러 컴퓨터에 에이전트를 설치하거나 업데이트하면 시간이 많이 소요될 수 있습니다. 이 유틸리티는 동시에 모든 대상 컴퓨터에서 에이전트를 설치 또는 업데이트할 수 있습니다. 명령행에서 또는 Rational Build Forge가 제공하는 프로젝트를 통해 유틸리티를 시작할 수 있습니다. 유틸리티는 사용자가 제공하는 XML 파일을 사용하여 대상 컴퓨터, 해당 컴퓨터의 로그인 신임 정보, 연결 방법 및 관련 정보를 판별합니다.

**중요사항:** 도구는 단방향 설치 또는 업그레이드 조작을 지원합니다.

- 도구는 설치 또는 업그레이드에서 롤백을 지원하지 않습니다.
- 도구는 설치할 버전에 대해 기존 에이전트의 버전을 확인하지 않습니다. 설치에 대해 지정하는 버전이 이미 설치되어 있는 버전보다 낮은 경우 낮은 버전이 설치됩니다.

유틸리티를 시작할 때 각 대상 컴퓨터에 대한 조치가 완료됩니다.

1. 컴퓨터에 로그인
2. 운영 체제 판별
3. 실행 중인 모든 에이전트 중지

유틸리티와 함께 사용하는 XML 파일에서 **StopCommand** 매개변수를 사용하여 Rational Build Forge 에이전트를 중지하는 사용자 정의된 명령을 사용할 수 있습니다.

4. 컴퓨터에 사용 가능한 임시 공간이 충분한지 여부 판별
5. 대상 컴퓨터에 운영 체제의 에이전트 설치 프로그램 복사
6. 설치 프로그램을 실행하여 컴퓨터에서 에이전트 업데이트 또는 설치



이 프로세스는 에이전트가 기본 위치에 있는 경우 BFAgent.conf 파일을 유지합니다. 프로세스는 또한 사용자가 유틸리티와 함께 사용하는 XML 파일의 **ExistingAgentPath** 매개변수를 사용하여 지정하는 다른 위치에 에이전트가 있는 경우 파일을 유지합니다.

#### 7. 임시 공간에서 설치 프로그램 제거

**참고:** 유틸리티는 또한 로그 파일을 작성하여 해당 조작의 히스토리를 캡처합니다. 이 로그 파일은 유틸리티가 실행되는 컴퓨터에 저장됩니다.

## 지원되는 운영 체제

에이전트 업데이트 및 배치 유틸리티는 두 개의 운영 체제에서 작동합니다.

유틸리티가 해당 운영 체제에서 실행됩니다.

- Windows
- Linux

#### 노트:

- 유틸리티를 실행하는 컴퓨터에는 IBM Rational Build Forge가 포함하는 IBM Java가 설치되어 있어야 합니다.
- 유틸리티에는 IBM Tivoli Remote Execution and Access가 필요하며, 이는 Rational Build Forge에 포함되어 있습니다.

유틸리티는 다음 운영 체제에서 Rational Build Forge 에이전트를 설치 및 업데이트할 수 있습니다.

- Windows
- Linux
- IBM AIX
- HP-UX
- Solaris

## 설치

IBM Rational Build Forge 설치 프로그램은 다음 파일에 에이전트 업데이트 및 배치 유틸리티를 설치합니다.

- Windows 시스템  
`<bfinstall>\java\agentupdate.jar`  
`<bfinstall>\agentupdate.bat`
- UNIX 및 Linux 시스템  
`<bfinstall>/Platform/java/agentupdate.jar`

## UNIX 또는 Linux를 실행하는 대상 컴퓨터의 연결 설정

IBM Tivoli Remote Execution and Access를 사용하여 액세스할 수 있도록 UNIX 또는 Linux를 실행하는 대상 컴퓨터를 구성해야 합니다.

Tivoli Remote Execution and Access를 통해 액세스하려면 대상 컴퓨터가 다음 요구사항을 충족해야 합니다.

**SSH:** 에이전트 업데이트 및 배치 유틸리티가 액세스하는 UNIX 또는 Linux 대상 컴퓨터에서 SSH가 사용 가능한지 확인하십시오.

## Windows를 실행하는 대상 컴퓨터의 연결 설정

IBM Tivoli Remote Execution and Access의 발견 기능을 사용할 수 있으려면 대상 컴퓨터가 Windows를 실행하도록 구성해야 합니다.

Tivoli Remote Execution and Access를 통해 액세스하려면 두 가지 옵션이 있습니다.

- **SSH/RSR:** 에이전트 업데이트 및 배치 유틸리티가 액세스하는 대상 컴퓨터에서 SSH/RSR가 사용 가능한지 확인하십시오.

Windows 운영 체제를 실행하는 대상 컴퓨터에 SSH 또는 RSR가 있는 경우 나머지 구성 단계를 건너뛸 수 있습니다.

- 다음 절의 Tivoli Remote Execution and Access 요구사항을 충족할 수 있도록 Windows 운영 체제를 실행하는 대상 컴퓨터를 구성하십시오. 첫 번째 절은 모든 Windows 운영 체제에 적합합니다. 이후 절은 표시된 Windows 운영 체제에 대한 것입니다.

**참고:** 이 주제의 Windows 정보는 다음 링크를 통해 사용 가능한 정보에서 파생됩니다. 최신 정보는 Windows 운영 체제에서 IBM Tivoli Remote Execution and Access 구성의 해당 링크를 참조하십시오.

- [http://publib.boulder.ibm.com/infocenter/tivihelp/v11r1/index.jsp?topic=/com.ibm.tivoli.tpm.scenario.doc/sdi/rept\\_prereq\\_win.html](http://publib.boulder.ibm.com/infocenter/tivihelp/v11r1/index.jsp?topic=/com.ibm.tivoli.tpm.scenario.doc/sdi/rept_prereq_win.html)
- [http://publib.boulder.ibm.com/infocenter/rafwhelp/v7r1/index.jsp?topic=/com.ibm.rafw.install.doc/topics/r\\_network\\_connection\\_reqs\\_Windows.html](http://publib.boulder.ibm.com/infocenter/rafwhelp/v7r1/index.jsp?topic=/com.ibm.rafw.install.doc/topics/r_network_connection_reqs_Windows.html)

### 모든 Windows 운영 체제

- 원격 레지스트리 관리를 사용하도록 설정해야 하며, 이는 대상 컴퓨터에서 기본 구성입니다. 다음과 같이 서비스를 여십시오.
  - Windows XP, Windows Vista: 시작 > 제어판 > 관리 도구 > 서비스를 선택하십시오.
  - Windows Server 2003, Windows Server 2008: 시작 > 설정 > 제어판 > 관리 도구 > 서비스를 선택하십시오.

원격 레지스트리 서비스에서 서비스의 상태가 시작되었는지 확인하십시오.

- Windows를 실행하는 대상 컴퓨터에 연결하는 데 Tivoli Remote Execution and Access를 사용하려면 관리 공유를 사용하도록 설정해야 합니다. 기본 관리 디스크 공유의 예제는 C\$ 및 D\$입니다. 공유를 사용하지 않도록 설정하는 경우 Tivoli Remote Execution and Access는 드라이브 내에 있는 디렉토리가 숨겨져 있다고 간주합니다. 이 경우 다음 메시지가 표시됩니다.

XCIM0009E: 원격 대상 <host\_name>에 연결하는 중에 오류가 발생했습니다.

예외: java.io.FileNotFoundException:

CTGRI0003E 지정된 원격 경로 이름을 찾을 수 없습니다. <file\_or\_directory\_path>

원인: com.starla.smb.SMBException: 네트워크 이름이 올바르지 않습니다.

관리 공유를 사용 가능하게 하려면 다음 단계를 따르십시오.

1. 내 컴퓨터를 클릭하십시오.
  2. 관리 공유에 사용하도록 설정하려는 디스크 드라이브를 마우스 오른쪽 단추로 클릭하십시오.
  3. 공유 및 보안을 클릭하십시오.
  4. 이 폴더 공유를 선택하십시오.
  5. C\$ 또는 D\$ 같은 공유 이름을 지정하고 확인을 클릭하십시오.
- WSH(Windows Scripting Host) 또는 WMI 서비스가 대상에서 사용하지 않도록 설정되어 있거나 VBScript가 사용하지 않도록 설정되어 있는 경우 일부 Windows 프로토콜 방법이 작동하지 않습니다.

### Windows XP Professional

- Windows XP 운영 체제를 실행하는 대상 컴퓨터에서 Tivoli Remote Execution and Access를 사용하려면 단순 파일 공유를 사용하지 않도록 설정해야 합니다. 단순 네트워킹은 모든 로그인 사용자가 게스트로 인증되도록 합니다. 게스트 로그인에는 Tivoli Remote Execution and Access가 기능하는 데 필요한 권한이 없습니다. 단순 파일 공유를 사용하지 않도록 설정하려면 다음을 수행하십시오.
  1. Windows 탐색기 창에서 도구 > 폴더 옵션을 클릭한 후 보기 탭을 클릭하십시오.
  2. 고급 설정 목록에서 단순 파일 공유 사용 선택란을 선택 취소하십시오.
  3. 적용을 클릭하고 확인을 클릭하십시오.
- Tivoli Remote Execution and Access를 사용하여 성공적으로 통신할 수 있도록 하려면 135(RPC) 및 445(TCP) 포트 모두 대상 컴퓨터에서 사용하도록 설정해야 합니다. Tivoli Remote Execution and Access가 445 포트가 사용되지 않도록 설정되어 있음을 인식하는 경우 139 포트를 사용합니다(TCP/IP를 통한 NetBIOS).
- 방화벽: Windows XP 운영 체제에는 기본적으로 사용하지 않도록 설정된 인터넷 연결 방화벽이라는 내장 방화벽이 포함되어 있습니다. 또한 Windows XP SP2(Service Pack 2)에는 Windows 방화벽이 포함되어 있으며 이는 기본적으로 사용하도록 설

정되어 있습니다. Windows XP를 실행하는 대상 컴퓨터에 방화벽이 사용되는 경우 방화벽은 Tivoli Remote Execution and Access가 액세스할 수 없도록 합니다.

Windows XP SP2에서 Windows 방화벽 구성의 예외 페이지에서 파일 및 프린터 공유 선택란을 선택하여 액세스를 허용할 수 있습니다.

### Windows Vista

- Guest 또는 Everyone 계정에 대해 파일 공유를 사용하도록 설정해야 하며 비밀번호 보호 공유를 사용하지 않도록 해야 합니다.

비밀번호로 보호되는 공유를 사용하지 않도록 설정하려면 다음을 수행하십시오.

1. 시작 > 제어판을 클릭하십시오.
  2. 네트워킹 및 인터넷을 클릭한 후 공유 및 발견을 클릭하십시오.
  3. 비밀번호로 보호되는 공유 옆에 있는 아래로 화살표를 클릭하십시오.
  4. 비밀번호로 보호되는 공유 끄기를 클릭하십시오.
  5. 적용을 클릭하고 제어판을 종료하십시오.
- 레지스트리 항목을 수정해야 할 수 있습니다.
    1. 왼쪽 아래 코너의 시작 > 검색 시작 필드에서 regedit를 입력한 후 Enter를 누르십시오.
    2. 왼쪽 분할창에서 HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\policies\system 폴더를 찾으십시오.
    3. 오른쪽 분할창에서 공백 영역을 마우스 오른쪽 단추로 클릭하십시오.
    4. 새로 작성을 클릭하십시오.
    5. **DWORD** 값을 클릭하십시오.
    6. LocalAccountTokenFilterPolicy를 입력하십시오.
    7. 작성한 항목을 두 번 클릭하십시오.
    8. 상자에 1을 입력하십시오.
    9. 확인을 클릭하십시오.
    10. 컴퓨터를 다시 시작하십시오.
  - 방화벽: Windows Vista에는 기본적으로 사용하지 않도록 설정된 인터넷 연결 방화벽이라는 내장 방화벽이 포함되어 있습니다. Windows Vista 운영 체제를 실행하는 대상 컴퓨터에 방화벽이 사용되는 경우 방화벽은 Tivoli Remote Execution and Access가 액세스할 수 없도록 합니다.

### Windows Server 2003

- Tivoli Remote Execution and Access를 사용하여 성공적으로 통신할 수 있도록 하려면 135(RPC) 및 445(TCP) 포트 모두 대상 컴퓨터에서 사용하도록 설정해야 함

니다. Tivoli Remote Execution and Access가 445 포트가 사용되지 않도록 설정되어 있음을 인식하는 경우 139 포트를 사용합니다(TCP/IP를 통한 NetBIOS).

### Windows Server 2008

- Guest 또는 Everyone 계정에 대해 파일 공유를 사용하도록 설정해야 하며 비밀번호 호 보호 공유를 사용하지 않도록 해야 합니다.

비밀번호로 보호되는 공유를 사용하지 않도록 설정하려면 다음을 수행하십시오.

1. 시작 > 제어판을 클릭하십시오.
  2. 네트워킹 및 인터넷을 클릭한 후 공유 및 발견을 클릭하십시오.
  3. 비밀번호로 보호되는 공유 옆에 있는 아래로 화살표를 클릭하십시오.
  4. 비밀번호로 보호되는 공유 끄기를 클릭하십시오.
  5. 적용을 클릭하고 제어판을 종료하십시오.
- 레지스트리 항목을 수정해야 할 수 있습니다.
    1. 시작 > 실행을 클릭하십시오.
    2. regedit를 입력하고 Enter를 누르십시오.
    3. 왼쪽 분할창에서 HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\policies\system 폴더를 찾으십시오.
    4. 오른쪽 분할창에서 공백 영역을 마우스 오른쪽 단추로 클릭하십시오.
    5. 새로 작성을 클릭하십시오.
    6. **DWORD 32비트 값**을 클릭하십시오.
    7. LocalAccountTokenFilterPolicy를 입력하십시오.
    8. 작성한 항목을 두 번 클릭하십시오.
    9. 상자에 1을 입력하십시오.
    10. 확인을 클릭하십시오.
    11. 컴퓨터를 다시 시작하십시오.
  - 방화벽: Windows Server 2008에는 기본적으로 사용하지 않도록 설정된 인터넷 연결 방화벽이라는 내장 방화벽이 포함되어 있습니다. Windows Server 2008 운영 체제를 실행하는 대상 컴퓨터에 방화벽이 사용되는 경우 방화벽은 Tivoli Remote Execution and Access가 액세스할 수 없도록 합니다.

### 사용법 개요

에이전트 업데이트 및 배치(agentupdate)를 사용하려면 유틸리티에 세 단계가 필요합니다.

다중 컴퓨터에서 IBM Rational Build Forge 에이전트를 설치 또는 업데이트하려면 다음을 수행하십시오.

1. **agentupdate** 유틸리티를 실행하는 컴퓨터의 디렉토리에 에이전트 설치 프로그램을 배치하십시오.

Jazz.net에 있는 Jazz 커뮤니티 사이트의 Rational Build Forge 프로젝트 영역 및 설치 매체에서 설치 프로그램을 얻을 수 있습니다.

2. XML 파일을 작성하여 Rational Build Forge 에이전트를 설치 또는 업데이트할 컴퓨터를 지정하십시오.

**참고:** XML 파일에 로그인 신임 정보를 포함시킬 때 신임 정보를 일반 텍스트로 지정하십시오. 비밀번호 암호화에 대한 정보는 180 페이지의 『에이전트 업데이트 유틸리티 실행』의 내용을 참조하십시오.

3. 명령행에서 또는 제공된 Rational Build Forge 프로젝트를 통해 **agentupdate** 명령을 실행하십시오.

XML 파일에 대한 자세한 정보는 『XML 파일에서 대상 시스템 지정』의 내용을 참조하십시오.

유틸리티를 실행하는 방법에 대한 자세한 정보는 180 페이지의 『에이전트 업데이트 유틸리티 실행』의 내용을 참조하십시오.

## XML 파일에서 대상 시스템 지정

XML 파일을 사용하여 에이전트를 설치 또는 업데이트할 대상 시스템을 지정하십시오.

### XML 파일 개요

에이전트 업데이트 및 배치 유틸리티에는 에이전트를 업데이트하거나 설치할 컴퓨터를 판별하기 위해 XML 파일이 필요합니다. 이 파일은 또한 유틸리티에 필요한 로그인 신임 정보 및 기타 항목을 지정해야 합니다.

파일에는 다음 요구사항이 있습니다.

- 기타 요소를 포함할 하나의 <AgentDeploymentAndUpdate> 요소
- 각 대상 컴퓨터를 지정할 하나 이상의 <Target> 요소
- <Globals> 요소가 사용되면 하나만 사용할 수 있습니다. 모든 대상 컴퓨터에 적용되는 항목일 경우에 사용하십시오.

다음 예제에는 하나의 <Target> 요소 및 <Globals> 요소가 표시됩니다. <Globals> 요소에 <Target> 요소 하나만을 사용하는 경우에는 해당 요소가 유용하지 않습니다. <Globals> 요소는 여러 개의 <Target> 요소를 사용하는 경우에 유용합니다.

```
<?xml version="1.0" encoding="UTF-8"?>
<AgentDeploymentAndUpdate>

  <Target id="9.184.112.152">
    <Parameter Name="ConnectionType" Value="ssh"/>
  </Target>
</AgentDeploymentAndUpdate>
```

```

<Parameter Name="UserName" Value="root"/>
<Parameter Name="Password" Value="passWord"/>
<Parameter Name="ExistingAgentPath" Value="/usr/local/bin/bfagent"/>
<Parameter Name="GZIPPath" Value="/usr/contrib/bin/" />
</Target>

<Globals>
<Parameter Name="KeyFile" Value="..\test\keyfile.txt"/>
</Globals>

</AgentDeploymentAndUpdate>

```

## XML 파일 구문

각 <Parameter> 요소는 <Target> 및 <Globals> 요소에서 모두 유효합니다. 이러한 유연성으로 인해 매개변수가 모든 대상 컴퓨터에 적용되는 경우 <Target> 요소를 사용하여 ID 속성만을 지정하고 각 <Parameter> 요소를 <Globals> 요소에 배치할 수 있습니다.

매개변수를 Value=""로 지정하는 것은 매개변수를 지정하지 않는 것과 동일합니다.

다음 표에서는 파일 구문을 설명합니다.

대상 속성	값
ID	대상 컴퓨터의 이름 또는 IP 주소입니다.

매개변수	값
<b>AgentInstallLocation</b>	IBM AIX, HP-UX 및 Windows 시스템의 경우에만 기타 플랫폼에 영향을 미치지 않습니다. 새 에이전트를 배치할 위치입니다. 지정되지 않은 경우 시스템 기본 임시 디렉토리가 사용됩니다.
<b>BFAgentPort</b>	IBM AIX 및 HP-UX 시스템의 경우에만 사용할 에이전트의 포트입니다. 응답 파일 생성 시 사용됩니다.
<b>BFSolarisAdminFile</b>	Solaris 시스템의 경우에만 자동 설치를 위해 응답 파일을 저장할 대상 시스템의 위치입니다. 지정되지 않은 경우 시스템 기본 임시 디렉토리가 사용됩니다.
<b>BFSolarisResponseFile</b>	Solaris 시스템의 경우에만 응답 파일을 저장할 대상 시스템의 위치입니다. 지정되지 않은 경우 시스템 기본 임시 디렉토리가 사용됩니다.
<b>ConnectionType</b>	UNIX 또는 Linux: SSH 또는 RSH  Windows: SSH, RSH, SMB 또는 CIFS  (RSH는 원격 셸입니다. SMB는 서버 메시지 블록입니다. CIFS는 공통 인터넷 파일 시스템입니다.)
<b>ConnectionPort</b>	(선택사항) 대상이 기본 포트를 사용하지 않는 경우 <b>ConnectionType</b> 매개변수로 지정된 연결 유형에 사용할 포트 번호입니다. 대상이 기본 포트를 사용하는 경우 이 매개변수를 지정할 필요가 없습니다.



매개변수	값
<b>ExistingAgentPath</b>	IBM AIX, HP-UX 및 Windows 시스템의 경우에 필수.  대상 컴퓨터에서의 기존 에이전트 위치입니다. 유틸리티는 사용자가 지정하는 위치에서 BFAgent.conf 파일의 백업을 작성하고 설치 또는 업그레이드가 완료된 후 이를 복원합니다. 기타 시스템의 경우 기본 시스템 설치 위치가 사용됩니다.
<b>GZIPPPath</b>	IBM AIX, HP-UX 및 Solaris 시스템.  에이전트 설치 프로그램이 <b>gzip</b> 명령(설치 프로그램의 압축을 푸는 명령)을 찾을 수 있는 디렉토리입니다.
<b>IsSudoEnabled</b>	sudo가 대상 시스템에서 사용되고 이를 사용하려는 경우 예로 설정하십시오. <b>SudoPassword</b> 를 올바른 sudo 비밀번호로 설정하십시오. 올바르게 설정되지 않으면 유틸리티는 sudo 로그인에 대해 <b>Password</b> 를 사용하려고 시도합니다.
<b>KeyFile</b>	비밀번호의 암호화 및 복호화에 사용되는 키를 저장하는 유틸리티를 실행하는 컴퓨터에 있는 파일입니다.
<b>LocalTempLocation</b>	IBM AIX, HP-UX 및 Windows의 경우에만.  응답 파일을 저장할 대상 시스템의 위치입니다. 지정되지 않은 경우 시스템 기본 임시 디렉토리가 사용됩니다. Solaris 시스템의 경우 BFASolarisResponseFile 또는 BFASolarisAdminFile을 사용하십시오.
<b>ManagedScriptsDir</b>	유틸리티가 실행되는 디렉토리로 설정하십시오. 이 디렉토리는 <b>agentupdate</b> 유틸리티를 실행하는 컴퓨터에 있습니다.
<b>SourceDirectory</b>	대상 컴퓨터의 설치 프로그램을 .exe, .rpm 및 .gz 양식으로 저장하는 디렉토리입니다. 이 디렉토리는 <b>agentupdate</b> 유틸리티를 실행하는 컴퓨터에 있습니다.
<b>StartCommand</b>	(선택사항) 대상 컴퓨터에서 Rational Build Forge 에이전트를 시작하기 위한 사용자 정의된 명령이 있습니다.  명령이 PATH/path 변수에 포함되는지 확인하거나 XML 파일에 전체 경로를 제공하십시오.
<b>StopCommand</b>	(선택사항) 대상 컴퓨터에서 Rational Build Forge 에이전트를 중지하기 위한 사용자 정의된 명령이 있습니다.  명령이 PATH/path 변수에 포함되는지 확인하거나 XML 파일에 전체 경로를 제공하십시오.
<b>SudoPassword</b>	sudo에 사용할 비밀번호입니다. <b>IsSudoEnabled</b> 를 참조하십시오.
<b>Password</b>	대상 컴퓨터에 로그인할 때 <b>UserName</b> 에 사용할 비밀번호입니다. <b>중요사항:</b> 처음에 XML 파일의 일반 텍스트에서 <b>Password</b> 매개변수를 정의하십시오. 180 페이지의 『에이전트 업데이트 유틸리티 실행』에서 설명한 것처럼 <b>-encrypt</b> 명령행 옵션을 사용하여 비밀번호를 암호화하십시오.
<b>TemporaryLocation</b>	유틸리티를 실행하기 전에 설치 프로그램을 저장할 대상 컴퓨터의 위치입니다.
<b>UseManagedScripts</b>	예 또는 아니오로 설정하십시오. 예로 설정하는 경우 유틸리티는 <b>ManagedScriptsDir</b> 디렉토리에서 <b>StartCommand</b> 및 <b>StopCommand</b> 를 찾으려고 시도합니다.
<b>UserName</b>	대상 컴퓨터에 로그인할 때 사용할 사용자 이름입니다.

## 샘플 XML 파일

해당 샘플은 XML 파일을 사용할 수 있는 방법을 설명합니다.

- 샘플 1

이 샘플에는 하나의 대상만 있습니다. 대상 IP 주소는 9.164.259.143입니다. <Target> 요소는 선택적 매개변수 **SourceDirectory** 또는 **TemporaryLocation**을 포함하지



않습니다. 그러나 <Globals> 요소에는 **SourceDirectory** 매개변수가 있습니다. 유틸리티는 대상에 대해 해당 매개변수의 값을 사용합니다.

```
<?xml version="1.0" encoding="UTF-8"?>
<AgentDeploymentAndUpdate>

  <Globals>
    <Parameter Name="SourceDirectory" Value="..\test"/>
    <Parameter Name="KeyFile" Value="..\test\keyfile.txt"/>
  </Globals>

  <Target id="9.164.259.143">
    <Parameter Name="ConnectionType" Value="ssh"/>
    <Parameter Name="UserName" Value="root"/>
    <Parameter Name="Password" Value="toor"/>
    <Parameter Name="ExistingAgentPath" Value="/usr/local/bin/bfagent"/>
    <Parameter Name="AgentInstallLocation" Value="/usr/local/bin/bfagent"/>
  </Target>

</AgentDeploymentAndUpdate>
```

- 샘플 2

이 샘플에는 네 개의 대상이 있습니다. 그러나 대상에는 여기에 특정한 매개변수가 필요하지 않습니다. 그러므로 모든 매개변수는 **UserName**, **Password** 및 **ConnectionType**을 포함하여 <Globals> 섹션에 지정됩니다.

```
<?xml version="1.0" encoding="UTF-8"?>
<AgentDeploymentAndUpdate>

  <Globals>
    <Parameter Name="SourceDirectory" Value="..\test"/>
    <Parameter Name="ConnectionType" Value="ssh"/>
    <Parameter Name="UserName" Value="root"/>
    <Parameter Name="Password" Value="toor"/>
    <Parameter Name="TemporaryLocation" Value="/tmp"/>
    <Parameter Name="ExistingAgentPath" Value="/usr/local/bin/bfagent"/>
    <Parameter Name="AgentInstallLocation" Value="/usr/local/bin/bfagent"/>
    <Parameter Name="KeyFile" Value="..\test\keyfile.txt"/>
  </Globals>

  <Target id="9.164.259.143">
  </Target>

  <Target id="9.164.102.169">
  </Target>

  <Target id="9.164.102.196">
  </Target>

  <Target id="9.106.189.157">
  </Target>

</AgentDeploymentAndUpdate>
```

- 샘플 3

대상 시스템에 대해 IP 주소의 범위를 지정하십시오. 다음 예제는 각 시스템에서 동일한 로그인 신임 정보를 사용하며 IP 주소 9.184.112.152 - 9.184.112.160에서 작동합니다(로그인: root, 비밀번호: password).

```
<?xml version="1.0" encoding="UTF-8"?>
<AgentDeploymentAndUpdate>
<Target id="9.184.112.152-160">
<Parameter Name="ConnectionType" Value="ssh">
<Parameter Name="ConnectionPort" Value=""/>
<Parameter Name="UserName" Value="root">
<Parameter Name="Password" Value="passWord">
</Target>
</AgentDeploymentAndUpdate>
```

- 샘플 4

모든 대상에 동일한 특성(연결 유형, 연결 포트, 사용자 이름, 비밀번호)이 있는 경우 해당 특성은 파일의 글로벌 섹션에서 지정할 수 있습니다.

```
<?xml version="1.0" encoding="UTF-8"?>
<AgentDeploymentAndUpdate>
<Globals>
<Parameter Name="ConnectionType" Value="ssh">
<Parameter Name="ConnectionPort" Value=""/>
<Parameter Name="UserName" Value="root">
<Parameter Name="Password" Value="passWord">
</Globals>

<Target id="9.184.112.152-160">
</Target>

<Target id="9.184.80.12-16">
</Target>
</AgentDeploymentAndUpdate>
```

## 에이전트 업데이트 유틸리티 실행

XML 파일을 작성한 후에 명령행에서 또는 제공된 BuildForgeAgentUpdateUtilityProject를 통해 에이전트 업데이트 및 배치 유틸리티를 실행할 수 있습니다.

### 설치 고려사항

도구는 단방향 설치 또는 업그레이드 조작을 지원합니다.

- 도구는 설치 또는 업그레이드에서 롤백을 지원하지 않습니다.
- 도구는 설치할 버전에 대해 기존 에이전트의 버전을 확인하지 않습니다. 설치에 대해 지정하는 버전이 이미 설치되어 있는 버전보다 낮은 경우 낮은 버전이 설치됩니다.

### 명령행에서 유틸리티 실행

다음 예제에서는 명령행에서 유틸리티를 실행할 수 있는 방법을 설명합니다. 유틸리티 구문은 182 페이지의 『agentupdate 유틸리티 참조』의 내용을 참조하십시오.

다음 명령은 **-preview** 옵션을 사용합니다. 이 옵션을 사용하여 유틸리티를 대상 컴퓨터에 연결하고 실제로 명령을 실행하지 않고 해당 대상에서 실행하는 명령을 인쇄합니다.

```
java -jar agentupdate.jar -preview -targets agent_deploy_config.xml  
-keyfile mykeyfile.txt
```

다음 명령은 입력 XML 파일의 모든 일반 텍스트 비밀번호를 암호화합니다. 명령은 원격 컴퓨터에 연결하지 않습니다.

```
java -jar agentupdate.jar -encrypt -targets agent_deploy_config.xml  
-keyfile mykeyfile.txt -logfile LogFile.txt
```

이 명령은 대상 컴퓨터에 IBM Rational Build Forge 에이전트를 설치하거나 업데이트합니다. **-sourceDir** 옵션은 모든 대상의 에이전트 설치 프로그램을 포함하는 디렉토리를 표시합니다. 유틸리티를 실행할 때 이 옵션을 포함시키지 않는 경우 XML 파일에 해당 매개변수를 포함시키십시오. 설치 프로그램이 XML 파일에서 모든 대상의 공통 디렉토리에 있는 경우 XML 파일에 있는 <Globals> 요소에 **SourceDirectory** 매개변수를 지정하십시오. 그러나 대상 중 하나의 소스 디렉토리가 다른 경우 해당 컴퓨터의 <Target> 요소에서 **SourceDirectory** 매개변수를 정의하십시오. 예제는 178 페이지의 『샘플 XML 파일』의 샘플 2를 참조하십시오.

```
java -jar agentupdate.jar -sourceDir .\resources -targets  
agent_deploy_config.xml -logfile LogFile.txt
```

## BuildForgeAgentUpdateUtilityProject를 통해 유틸리티 실행

IBM Rational Build Forge는 유틸리티를 사용할 수 있는 방법을 설명하는 `bfagentupdate.xml` 샘플 프로젝트를 제공합니다.

샘플 프로젝트를 사용하려면 관리 콘솔에서 가져오기 기능을 사용하거나 **bfimport** 명령을 사용하여 프로젝트를 가져오십시오. `C:\Program Files\IBM\Build Forge\samples\agentupdate\BFproject\Windows` 또는 `/opt/buildforge/samples/agentupdate/BFproject/Linux`에서 프로젝트를 가져오십시오.

이 프로젝트에는 미리보기, 암호화 및 실행의 세 단계가 필요합니다. 샘플 프로젝트에는 미리보기 및 암호화 단계가 포함되지 않습니다.

- 미리보기 단계: 빌드 시스템은 대상 컴퓨터에 연결하고 실제로 명령을 실행하지 않고 각 대상에서 에이전트를 업데이트하거나 설치하도록 실행할 수 있는 명령을 표시합니다.
- 암호화 단계: 빌드 시스템이 대상 컴퓨터의 에이전트를 업데이트 또는 설치하지 않고 모든 비밀번호를 암호화합니다. 이 단계는 입력 XML 파일의 일반 텍스트를 암호화된 비밀번호로 대체합니다.

- 실행 단계: 빌드 시스템은 대상 컴퓨터에 연결하고, 해당 운영 체제를 식별하며 실행 중인 모든 에이전트를 중지하고 설치 프로그램을 위해 임시 공간을 확인하며 설치 프로그램을 대상 컴퓨터에 복사하고 업데이트 또는 설치를 완료한 후 설치 프로그램을 임시 공간에서 제거합니다.

사용자의 요구사항에 따라 단계를 사용하도록 설정하고 프로젝트를 실행하십시오.

## agentupdate 유틸리티 참조

유틸리티는 해당 동작을 제어하기 위한 여러 명령행 옵션 및 환경 변수를 제공합니다.

이 표는 명령행 옵션을 알파벳순으로 나타냅니다.

옵션	필수	설명
<b>-debug</b>	아니오	확장 디버깅 출력을 사용합니다.  또한 DEBUG 환경 변수를 값으로 설정하여 디버깅을 사용할 수도 있습니다. 우선순위에 대한 정보는 183 페이지의 『XML 파일 매개변수, 명령행 옵션 및 환경 변수의 우선순위』의 내용을 참조하십시오.
<b>-encrypt</b>	아니오	XML 파일의 일반 텍스트 비밀번호는 암호화된 양식의 비밀번호로 대체됩니다.
<b>-genkey &lt;key_file&gt;</b>	아니오	비밀번호의 암호화 및 복호화를 위해 비밀 키를 생성합니다.  <b>-genkey</b> 옵션을 지정할 때 <b>-keyfile &lt;key_file&gt;</b> 옵션도 지정하여 지정된 파일에 키를 저장해야 합니다.
<b>-keyfile &lt;key_file&gt;</b>	아니오	비밀번호의 암호화 및 복호화를 위해 키를 보유합니다.  또한 <b>KeyFile</b> XML 파일 매개변수를 설정하여 이 파일을 지정할 수도 있습니다. 우선순위에 대한 정보는 183 페이지의 『XML 파일 매개변수, 명령행 옵션 및 환경 변수의 우선순위』의 내용을 참조하십시오.
<b>-logFile &lt;log_file&gt;</b>	아니오	유틸리티가 메시지를 로그하는 파일을 지정합니다.  기본적으로 이름은 RemoteAgentDeployerUpdater_Log.txt입니다.
<b>-maxThreads</b>	아니오	실행될 때 유틸리티가 사용하는 최대 스레드 수를 설정합니다. 기본적으로 호스트 하드웨어의 프로세서 코어 수의 두 배로 설정됩니다.
<b>-preview</b>	아니오	각 대상에 대해 유틸리티가 실행하는 명령을 실행하지 않고 나열합니다.
<b>-sourceDir &lt;path&gt;</b>	아니오	여러 플랫폼의 에이전트 설치 프로그램이 .exe, .rpm 및 .gz 양식으로 저장되는 디렉토리를 지정합니다.  <b>agentupdate</b> 유틸리티는 각 대상의 운영 체제를 판별한 후 이 디렉토리에서 해당 에이전트 설치 프로그램을 사용합니다. 참고: 이 유틸리티는 에이전트 설치 프로그램을 사용합니다. 그러나 에이전트 소스 코드는 사용하지 않습니다.  또한 <b>SourceDirectory</b> XML 파일 매개변수 또는 <b>SOURCEDIR</b> 환경 변수를 설정하여 이 디렉토리를 지정할 수도 있습니다. 우선순위에 대한 정보는 183 페이지의 『XML 파일 매개변수, 명령행 옵션 및 환경 변수의 우선순위』의 내용을 참조하십시오.
<b>-targets &lt;XML_file&gt;</b>	예	IBM Rational Build Forge 에이전트를 업데이트하거나 설치할 대상 컴퓨터를 표시하도록 사용자가 작성한 XML 파일에 대한 경로와 파일 이름을 지정합니다.

이 표에서는 환경 변수를 설명합니다.

환경 변수	필수	설명
디버그	아니오	확장 디버깅 출력을 사용합니다.  또한 <b>-debug</b> 명령행 옵션을 지정하여 디버깅을 사용할 수도 있습니다. 우선순위에 대한 정보는 『XML 파일 매개변수, 명령행 옵션 및 환경 변수의 우선순위』의 내용을 참조하십시오.
SOURCEDIR	아니오	여러 플랫폼의 에이전트 설치 프로그램이 .exe, .rpm 및 .gz 양식으로 저장되는 디렉토리를 지정합니다.  <b>agentupdate</b> 유틸리티는 각 대상의 운영 체제를 판별한 후 이 디렉토리에서 해당 에이전트 설치 프로그램을 사용합니다. 참고: 이 유틸리티는 에이전트 설치 프로그램을 사용합니다. 그러나 에이전트 소스 코드는 사용하지 않습니다.  또한 <b>SourceDirectory</b> XML 파일 매개변수 또는 <b>-sourceDir</b> 명령행 옵션을 설정하여 이 디렉토리를 지정할 수도 있습니다. 우선순위에 대한 정보는 『XML 파일 매개변수, 명령행 옵션 및 환경 변수의 우선순위』의 내용을 참조하십시오.

## XML 파일 매개변수, 명령행 옵션 및 환경 변수의 우선순위

여러 방법으로 유틸리티를 제어할 수 있습니다. 우선순위는 유틸리티에 실제로 영향을 미치는 방법을 판별합니다.

에이전트 업데이트 및 배치 유틸리티는 해당 동작을 제어하기 위해 우선순위대로 다음 방법을 제공합니다.

1. <Target> 요소의 매개변수
2. 명령행 옵션
3. <Globals> 요소의 매개변수
4. 환경 변수

유틸리티는 각 방법을 순서대로 고려합니다. 방법이 올바른 값으로 정의되면 유틸리티는 해당 값을 사용하고 다음 방법을 모두 무시합니다.

### 예제

이 예제는 유틸리티가 설치 프로그램을 찾는 방법을 설명합니다. <Target> 및 <Globals> 요소 모두가 **SourceDirectory** 매개변수를 정의했습니다. 이 매개변수는 설치 프로그램의 가능한 위치를 표시합니다. 그러나 유틸리티에는 확인할 여러 위치가 있습니다. 유틸리티가 설치 프로그램을 찾는 즉시 유틸리티는 설치 프로그램을 사용합니다.

1. 유틸리티는 먼저 <Target> 요소의 매개변수를 확인합니다. `..\test\wind` 디렉토리가 있는 경우 유틸리티는 해당 디렉토리의 설치 프로그램을 확인합니다. 설치 프로그램이 해당 디렉토리에 있는 경우 유틸리티는 해당 설치 프로그램을 사용합니다.

2. 디렉토리가 없거나 설치 프로그램이 포함되지 않은 경우 유틸리티는 **-sourceDir** 명령 옵션이 지정되었는지 여부를 확인합니다. 옵션이 올바른 디렉토리로 지정된 경우 유틸리티는 해당 디렉토리를 확인하고 해당 디렉토리의 설치 프로그램을 사용합니다.
3. **-sourceDir**이 지정되지 않았거나 디렉토리가 올바르지 않거나 디렉토리에 설치 프로그램이 포함되지 않은 경우 유틸리티는 <Globals> 요소의 매개변수를 확인합니다. 이제 유틸리티가 **..\test** 디렉토리를 확인합니다.
4. **..\test** 디렉토리가 설치 프로그램을 생성하지 않는 경우 유틸리티는 **SOURCEDIR** 환경 변수를 확인합니다. 해당 디렉토리가 없거나 설치 프로그램이 포함되지 않은 경우 유틸리티는 오류를 생성합니다.

```
<?xml version="1.0" encoding="UTF-8"?>
<AgentDeploymentAndUpdate>

<Globals>
<Parameter Name="SourceDirectory" Value="..\test"/>
<Parameter Name="KeyFile" Value="..\test\keyfile.txt"/>
</Globals>

<Target id="9.164.102.169">
<Parameter Name="SourceDirectory" Value="..\test\wind"/>
<Parameter Name="ConnectionType" Value="ssh"/>
<Parameter Name="UserName" Value="root"/>
<Parameter Name="Password" Value="toor123"/>
<Parameter Name="ExistingAgentPath" Value="/usr/local/bin/bfagent"/>
<Parameter Name="AgentInstallLocation" Value="/usr/local/bin/bfagent"/>
</Target>

</AgentDeploymentAndUpdate>
```

## 로그 파일을 통해 유틸리티 조치 보기

로그 파일은 유틸리티 조치의 히스토리를 제공합니다.

**-logFile** 옵션으로 로그 메시지를 저장할 파일 이름을 지정할 수 있습니다. **-logFile** 옵션을 지정하지 않는 경우 유틸리티는 메시지를 **RemoteAgentDeployerUpdater\_Log.txt**에 로그합니다. 이 파일은 **agentupdate** 유틸리티를 저장하는 디렉토리에 작성됩니다.

로그 파일은 형식은 [MSGTYPE] [IP address/MachineName] [Msg]입니다.

모든 대상의 메시지가 동일한 로그 파일에 저장됩니다.

## 문제점 해결

**agentupdate** 유틸리티는 문제점 해결에 도움이 되는 정보를 표시할 수 있는 여러 방법을 제공합니다.

기본적으로, 이 유틸리티는 연결 문제점, 충분하지 못한 디스크 공간 등의 일반적인 문제를 파악할 수 있는 충분한 정보를 제공합니다. 그러나 추가 디버깅 정보가 필요한 경우 **-debug** 명령행 옵션을 사용하십시오. 또는 임의 값으로 **DEBUG** 환경 변수를 정의하여 도구에서 추가 로그 메시지 또는 추가 디버깅 정보를 사용 가능하게할 수 있습니다.

### 에이전트 업데이트의 메시지 참조

**agentupdate** 유틸리티는 조작 동안 오류 및 정보 메시지를 제공합니다.

다음 주제에서는 모든 **agentupdate** 메시지에 대한 참조를 제공합니다.

**CRRBU0001I** 에이전트 업데이트 및 배치 유틸리티가 시작되었습니다.

**CRRBU0002I** 디버그 모드가 켜졌습니다.

**CRRBU0003I** 비밀번호 암호화가 완료되었습니다.

**CRRBU0004I** XML 구성 파일이 구문 분석되었습니다. 대상을 처리하는 중입니다...

**CRRBU0005I** 처리 대상 [{0}]

**CRRBU0006I** [{0}] 프로토콜 = {1} 사용자 = {2}

**CRRBU0007I** [{0}] 암호화된 비밀번호 = {1}

**CRRBU0008I** 관리 대상 스크립트 디렉토리가 작성되었습니다: {0}.

**CRRBU0009I** I[{0}] 복호화된 비밀번호 = #{1}#

**CRRBU0010I** 매개변수 키 파일 = {0}.

**CRRBU0011I** 하위 수: {0}

**CRRBU0012I** 매개변수 genkey = {0}.

**CRRBU0013I** [{0}] 의사 암호화 비밀번호 = {1}.

**CRRBU0014I** 텍스트 노드

**CRRBU0015I** 주석 노드

**CRRBU0016I** 속성 노드

**CRRBU0017I** 문서 노드

**CRRBU0018I** 문서 유형 노드

**CRRBU0019I** 요소 노드

**CRRBU0020I** 알 수 없는 노드

**CRRBU0021I** 총 대상 수: {0}

**CRRBU0022I** 요소 개수 = {0}

**CRRBU0023I** userid={1}인 {0}에 대해 Windows 프로토콜을 시도 중입니다.

**CRRBU0024I** RXA 인터페이스를 사용하여 대상 시스템의 여유 공간을 확인 중입니다.

**CRRBU0025I** 대상 시스템의 여유 공간 = {0} KB.

**CRRBU0026I** OS {0} 설치 위치 {1}에서 에이전트를 확인 중입니다.

**CRRBU0027I** 다음 위치에 에이전트가 설치된 것을 발견했습니다: {0}.

**CRRBU0028I** [프로세스 중지 중]

**CRRBU0029I** 프로세스 이름 = {0}, 프로그램 이름 = {1}



---

**CRRBU0030I** 스테이지 구문 분석 중: 요소: {0} = {1}

---

**CRRBU0031I** 대상의 에이전트 프로세스(bfagent)를 중지할 수 없습니다.

---

**CRRBU0032I** 대상 {0}에서 실행 중인 에이전트를 중지하는 중입니다.

---

**CRRBU0033I** Windows에서 {0} 서비스를 중지하려고 시도 중입니다.

---

**CRRBU0034I** RXA 인터페이스를 사용하여 대상에서 에이전트 서비스를 중지하는 중입니다.

---

**CRRBU0035I** 대상 {0}에서 에이전트 서비스를 중지하는 중입니다.

---

**CRRBU0036I** 대상에서 {0} 임시 디렉토리를 사용 중입니다.

---

**CRRBU0037I** 디렉토리에 접두부가 일치하는 것이 있는지 확인 중입니다. {0}

---

**CRRBU0038I** </0>에서 설치 프로그램을 찾는 중입니다.

---

**CRRBU0039I** RXA 인터페이스를 사용하여 </0> 설치 프로그램을 대상으로 복사하는 중입니다.

---

**CRRBU0040I** 키 파일을 작성했습니다: {0}

---

**CRRBU0041I** 사용자 구성 파일의 백업: {0}

---

**CRRBU0042I** 대상에서 {0}을(를) 정리하는 중입니다.

---

**CRRBU0043I** [{0}] 디렉토리를 제거하는 중입니다.

---

**CRRBU0044I** 에이전트 설치 위치 {0}

---



---

**CRRBU0045I** {0}에서 설치를 진행 중입니다.

---

**CRRBU0046I** 설치/업그레이드 대상: {0}

---

**CRRBU0047I** 미리보기 모드가 사용 가능합니다.

---

**CRRBU0048I** {0} 디렉토리에서 response.txt가 작성되었습니다.

---

**CRRBU0049I** 사용자 구성 파일 {0}을(를) 복원하는 중입니다.

---

**CRRBU0050I** {0} 명령을 실행하는 중입니다.

---

**CRRBU0051I** 현재 작업 디렉토리는 [{0}]입니다.

---

**CRRBU0052I** {0}(으)로 복사하는 중입니다.

---

**CRRBU0053I** 대상 {0}에서 실행 중인 bfdispatch 프로세스를 강제 종료하는 중입니다.

---

**CRRBU0054I** 대상에 대해 새 스레드가 시작되었습니다.

---

**CRRBU0055I** 이제 대상에 에이전트가 설치되었습니다.

---

**CRRBU0056I** 원격 액세스 문제점: 미리보기 모드를 계속할 수 없습니다.

---

**CRRBU0057I** {0}을(를) 사용하여 대상에서 실행 중인 에이전트를 중지하는 중입니다.

---

**CRRBU0058I** 대상에서 에이전트가 중지되었습니다.

---

**CRRBU0059I** 에이전트가 대상에 설치되어 있지 않습니다.

---

**CRRBU0060I** 대상의 임시 스토리지 위치는 {0}입니다.

---



---

**CRRBU0061I** 모든 비밀번호를 암호화하십시오.

---

**CRRBU0062I** {2}의 {1}에 {0} 설치 프로그램이 사용  
됩니다.

---

**CRRBU0063I** {0}이(가) 임시 위치에 복사되었습니다.

---

**CRRBU0064I** 구성 파일이 백업되었습니다.

---

**CRRBU0065I** 기존 에이전트 설치 위치: {0}

---

**CRRBU0066I** {0}을(를) 사용하여 대상에서 실행 중인  
에이전트를 시작합니다.

---

**CRRBU0067I** 스레드를 종료하는 중입니다.

---

**CRRBU0068I** 이름이 {0}인 관리 파일을 작성했습니  
다.

---

**CRRBU0069I** 노드: {0}

---

**CRRBU0070I** 노드 속성: {0}

---

**CRRBU0071I** 매개변수 로그 파일 = {0}

---

**CRRBU0072I** 매개변수 sourceDir = {0}

---

**CRRBU0073I** 매개변수 소스 파일 = {0}

---

**CRRBU0074I** 매개변수 대상 = {0}

---

**CRRBU0075I** getOS: [{0}][{1}]

---

**CRRBU0084I** 대상 정보를 포함하는 파일의 경로입니  
다(필수).

---

**CRRBU0085I** {0}에서 키가 작성되었습니다.

---

**CRRBU0088I** 최대 스레드 설정값: {0}

---



---

**CRRBU1000W** [{0}] 프로토콜이 지정되지 않았습니  
다.

---

**CRRBU1001W** [{0}] 대상 프로토콜이 지원되지 않습  
니다. {1}

---

**CRRBU1002W** XML 파일에 글로벌 섹션이 없습니  
다.

---

**CRRBU1003W** XML 파일에 대상 섹션이 없습니다.

---

**CRRBU1004W** 올바르지 않은 대상 항목: XML 파  
일에 대상 섹션의 ID가 없습니다.

---

**CRRBU1005W** [stderr] {0}

---

**CRRBU1006W** [stdout] {0}

---

**CRRBU1007W** 대상의 임시 위치가 널로 설정되어 있  
습니다.

---

**CRRBU1008W** 지원되지 않음: Sun Solaris 버전  
{0}

---

**CRRBU1009W** XML 파일의 대상 섹션에 지정된 소  
스 디렉토리에서 설치 프로그램을 찾을 수  
없습니다.

---

**CRRBU1010W** 명령행에 설치 프로그램 위치가 지정  
되지 않았습니다.

---

**CRRBU1011W** XML 파일의 글로벌 섹션에서 설치  
프로그램 위치가 지정되지 않았습니다.

---

**CRRBU1012W** 환경에서 설치 프로그램 위치가 지정  
되지 않았습니다.

---

**CRRBU1013W** 대상 오류: 임시 관리 파일을 생성할  
수 없습니다.

---

**CRRBU1014W** 대상 오류: 임시 응답 파일을 작성할  
수 없습니다.

---

---

**CRRBU1015W** [리턴 코드] {0}

---

**CRRBU2000E** 예외가 발견되었습니다.

---

**CRRBU2001E** 로컬 또는 원격 위치에 중지 스크립트가 없습니다.

---

**CRRBU2002E** 관리 대상 스크립트 디렉토리를 작성할 수 없습니다: {0}

---

**CRRBU2003E** 누락된 필수 인수입니다. {0}

---

**CRRBU2004E** **TargetInfoTable** 데이터 구조 사용법이 올바르지 않습니다.

---

**CRRBU2005E** XML 파일에 여러 개의 **Globals** 태그가 있습니다. 하나만 지정하십시오.

---

**CRRBU2006E** 정의되지 않은 매개변수 **ManagedScriptsDir**입니다.

---

**CRRBU2007E** 시작 명령 <{0}>을(를) 실행하는 데 문제가 발생했습니다.

---

**CRRBU2008E** [{0}] 일반 텍스트 비밀번호 및 키 작업이 지정되지 않았습니다.

---

**CRRBU2009E** 시작/중지 스크립트용 파일 오브젝트를 작성하는 중에 오류가 발생했습니다.

---

**CRRBU2010E** 올바르지 않은 대상 항목: XML 파일에서 대상에 널 호스트 이름 또는 주소를 지정했습니다.

---

**CRRBU2011E** 로컬 또는 원격 위치에 중지 스크립트가 없습니다.

---

**CRRBU2012E** XML 파일에 쓰는 중에 오류가 발생했습니다.

---

**CRRBU2013E** 중복 대상 항목 무시: XML 파일의 대상 {0}.

---



---

**CRRBU2014E** 사용 가능한 공간이 충분하지 않습니다. 사용 가능한 여유 공간 = {0} 필요한 여유 공간 = {1}(위치: {2})

---

**CRRBU2015E** 연결 예외가 발견되었습니다. {0}

---

**CRRBU2016E** 파일을 찾을 수 없음 예외 발생: {0}

---

**CRRBU2017E** I/O 예외가 발견되었습니다. {0}

---

**CRRBU2018E** 대상에서 에이전트 프로세스(bfagent)를 중지하는 중에 오류가 발생했습니다.

---

**CRRBU2019E** 대상에서 에이전트 서비스를 중지하는 중에 오류가 발생했습니다.

---

**CRRBU2020E** 디렉토리가 올바르지 않거나 설치 프로그램이 없습니다. {0}

---

**CRRBU2021E** 디렉토리에 접두부와 일치하는 파일이 너무 많습니다. {0}

---

**CRRBU2022E** 디렉토리에 접두부와 일치하는 파일이 없습니다. {0}

---

**CRRBU2023E** 설치 프로그램 소스 디렉토리가 지정되지 않았거나 설치 프로그램을 지정된 디렉토리에서 찾을 수 없습니다.

---

**CRRBU2024E** 오류: 대상에서 bfdispatch 프로세스를 강제 종료할 수 없습니다.

---

**CRRBU2025E** 중지 명령 <{0}>을(를) 실행하는 데 문제가 발생했습니다.

---

**CRRBU2026E** 중지 명령을 완료할 수 없습니다. {0}에서 작업을 계속할 수 없습니다.

---

**CRRBU2027E** 대상에서 에이전트를 중지하는 중에 문제가 발생했습니다.

---

**CRRBU2028E** 중지 명령을 완료할 수 없습니다. 작업을 계속할 수 없습니다.

**CRRBU2029E** 대상 {0}에서 충분하지 않은 디스크 공간

**CRRBU2030E** 설치 프로그램을 찾을 수 없습니다.

**CRRBU2031E** {0}(을/를) 임시 위치에 복사하는 데 실패했습니다.

**CRRBU2032E** 구성 파일의 백업에 실패했습니다.

**CRRBU2033E** 구성 파일의 복원에 실패했습니다.

**CRRBU2034E** {0}에 대한 인증 오류로 인해 원격 액세스에 실패했습니다.

**CRRBU2035E** 지원되지 않는 운영 체제입니다.

**CRRBU2036E** 시작 명령을 완료할 수 없습니다. {0}에서 작업을 계속할 수 없습니다.

**CRRBU2037E** 정의되지 않은 매개변수 ManagedScriptsDir입니다.

**CRRBU2038E** 지원되지 않는 플랫폼입니다.

**CRRBU2039E** StopScript용 로컬 파일이 없거나 접두부가 StopScript\_인 항목이 여러 개 있습니다.

**CRRBU2040E** StartScript용 로컬 파일이 없거나 접두부가 StartScript\_인 항목이 여러 개 있습니다.

**CRRBU2041E** 서버에서 인증할 수 없습니다.

**CRRBU2042E** {0} 키 파일을 작성할 수 없습니다.

**CRRBU2043E** 암호화 알고리즘 오류입니다. 사용자 환경에서 DES가 사용 가능한지 확인하십시오. 키 파일을 작성하는 데 실패했습니다.

**CRRBU2044E** {0} 키 파일은 올바른 키 파일이 아닙니다. 키 파일을 설정하는 데 실패했습니다.

**CRRBU2045E** 암호화 알고리즘 오류입니다. 사용자 환경에서 DES가 사용 가능한지 확인하십시오. 키 파일을 설정하는 데 실패했습니다.

**CRRBU2046E** {0} 키 파일에 액세스할 수 없습니다. 키 파일을 설정하는 데 실패했습니다.

**CRRBU2047E** 암호화 알고리즘 오류입니다. 사용자 환경에서 DES가 사용 가능한지 확인하십시오. 암호화에 실패했습니다.

**CRRBU2048E** 환경에서 DES를 사용한 암호화용 채우기가 사용 불가능합니다. 암호화에 실패했습니다.

**CRRBU2049E** 키 파일에서 생성된 키가 올바르지 않습니다. 올바른 키 파일을 사용 중인지 확인하십시오. 복호화에 실패했습니다.

**CRRBU2050E** 비밀번호에 대한 문자 인코딩이 지원되지 않습니다. 암호화에 실패했습니다.

**CRRBU2051E** 암호에 제공된 데이터 길이가 올바르지 않습니다. 암호화에 실패했습니다.

**CRRBU2052E** 예상된 데이터 채우기가 없습니다. 암호화에 실패했습니다.

**CRRBU2053E** 암호화 알고리즘 오류입니다. 사용자 환경에서 DES가 사용 가능한지 확인하십시오. 복호화에 실패했습니다.

**CRRBU2054E** 사용자 환경에서 DES를 사용한 암호화용 채우기가 사용 불가능합니다. 복호화에 실패했습니다.

**CRRBU2055E** 키 파일에서 생성된 키가 올바르지 않습니다. 올바른 키 파일을 사용 중인지 확인하십시오. 복호화에 실패했습니다.

CRRBU2056E 암호에 제공된 데이터 길이가 올바르지 않습니다. 복호화에 실패했습니다.

CRRBU2057E 예상된 데이터 채우기가 없습니다. 복호화에 실패했습니다.

CRRBU2058E 비밀번호에 대한 문자 인코딩이 지원되지 않습니다. 복호화에 실패했습니다.

CRRBU2059E 키 파일이 지정되지 않았습니다.

CRRBU2060E /O/은(는) 올바른 응답이 아닙니다.

## 에이전트 실행

이 절에서는 실행할 에이전트를 설정하는 방법에 대해 설명합니다. 에이전트는 보통 자동 시작 서비스 또는 디먼으로 실행됩니다.

### Windows에서 에이전트 실행

일반적으로 에이전트는 서비스로 설치되며 컴퓨터를 켤 때 시작되도록 자동으로 설정됩니다. 에이전트를 시작 및 중지하려면 에이전트가 설치된 컴퓨터에 로그인해야 합니다.

에이전트를 시작하고 중지하기 위해 다음과 같이 시작 메뉴를 사용할 수 있습니다.

- 에이전트를 시작하려면 시작 > 프로그램 > IBM Rational Build Forge > 에이전트 서비스 시작을 클릭하십시오.
- 에이전트를 중지하려면 시작 > 프로그램 > IBM Rational Build Forge > 에이전트 서비스 중지를 클릭하십시오.

명령 프롬프트에서 다음 명령을 사용할 수도 있습니다.

- net start bfagent
- net stop bfagent

### UNIX, Linux 및 MacOS에서 에이전트 실행

에이전트는 서비스로 실행되고 시스템 다시 시작 시 자동으로 다시 시작되어야 합니다.

bfagent 항목을 inetd 또는 xinetd의 구성에 적절하게 추가합니다. 다음 예제는 에이전트가 /usr/local/bin에 설치된 Linux 시스템의 xinetd.d에 있는 bfagent 항목입니다.

```
# description: The IBM Rational Build Forge Agent serves build requests
#   from the IBM Rational Build Forge Management Console.
service bfagent
{
    disable           = no
    flags             = REUSE
    socket_type       = stream
    wait              = no
    user              = root
    server             = /usr/local/bin/bfagent
    log_on_failure    += USERID
}
```

필요할 경우, 에이전트를 inetd/xinetd 환경 외부에서 실행할 수 있습니다. 독립형 디먼으로 실행하려면 -s 옵션을 사용하십시오.

```
bfagent -s
```

이 옵션을 사용하면, 에이전트가 백그라운드로 이동되어 연결 청취를 시작합니다. 컴퓨터를 시작할 때 에이전트가 자동으로 시작되도록 시작 스크립트에 이 명령을 배치합니다.

## System i에서 에이전트 실행

System i 플랫폼에서 에이전트를 실행하려는 경우 이 주제에 있는 정보를 검토합니다.

### 에이전트 포트 번호가 고유한지 확인

표준 Build Forge 에이전트 포트인 포트 5555가 System i 서버에서 다른 에이전트에 이미 지정되어 있을 수 있습니다. 이런 경우, 에이전트를 시작하기 전에 Build Forge 에이전트 포트를 지정되지 않은 포트 중 하나로 변경하십시오. 이를 수행하려면 bfagent.conf 파일을 직접 편집하십시오. 자세한 사항은 194 페이지의 『에이전트 포트 변경』의 내용을 참조하십시오.

### 수동으로 에이전트 시작

설치 지시사항 165 페이지의 『System i 플랫폼에 에이전트 설치』에서 7단계를 완료한 경우 System i를 시작하면 에이전트가 BAGENT 사용자로 시작됩니다.

또는, 다음 명령을 사용하여 System i에서 에이전트를 수동으로 시작할 수 있습니다.

```
bfagent -s
```

참고: bfagent.conf 파일이 /etc(기본 위치)에 설치되어 있지 않은 경우 -f 옵션을 사용하여 bfagent.conf 위치를 지정합니다.

bfagent 명령을 실행하고 에이전트를 수동으로 시작하면 에이전트가 에이전트를 시작하는 사용자로 시작됩니다.

- QSECOFR 사용자 또는 \*ALLOBJ 특수 권한이 있는 사용자가 에이전트를 시작하면 관리 콘솔에 지정한 서버 인증을 통해 사용자가 인증됩니다.
- 다른 사용자가 에이전트를 시작하면 bfagent.conf 파일에 magic\_login 설정을 구성하여 사용자를 인증합니다. 196 페이지의 『bfagent.conf 참조』의 내용을 참조하십시오.

### i5/OS PASE 프로그램이 설치되어 있는지 확인

에이전트는 i5/OS PASE(Portable Application Solution Environment) 프로그램으로 실행됩니다. PASE는 i5/OS에 포함되어 있으며 AIX 2진 및 명령을 실행할 수 있습니다. 일반적으로 PASE는 기본적으로 설치됩니다.

PASE가 설치되어 있는지 여부를 판별하려면 명령행에서 DSPSFWRSC를 실행합니다.

PASE 프로그램이 설치되어 있지 않은 경우 설치 CD에서 해당 프로그램을 로드합니다.

## PASE에서 에이전트 사용

PASE 환경에서 i5/OS에 애플리케이션을 빌드하는 데 필요한 대부분의 태스크에 액세스할 수 있습니다. iSeries 플랫폼을 대상으로 하는 프로세스의 자동화를 계획하고 정의하는 경우 이러한 사실을 명심해야 합니다.

단계의 명령은 PASE 셸에 의해 해석됩니다. 다음 구문을 사용하여 원시 명령을 실행할 수도 있습니다.

```
system -bi0E "<native commands>"
```

**중요사항:** 단계에서 각 시스템 명령은 고유 프로세스를 실행합니다. 이는 명령이 명령 자체의 프로세스 내에서만 실행됨을 의미합니다.

예를 들어, 단계 세트의 라이브러리 목록을 설정하려는 경우:

- CHGSYSLIBL 또는 ADDLIBLE 명령은 원시 명령(PASE에서 인식되지 않음)이므로 이 명령을 단계 명령으로 사용할 수 없습니다.
- 단계에서 지원되는 원시 명령 구문(예: `system -bi0E "ADDLIBLE FLGHT400"`)을 사용할 수 없습니다. 이는 이러한 명령 구문이 명령 자체 프로세스에 대해서만 라이브러리 목록을 변경하기 때문입니다. 후속 명령 및 단계는 변경에 의해 영향을 받지 않습니다.

단계, 단계 세트 또는 프로젝트의 라이브러리 목록을 설정할 수 없는 경우에도 시작 명령 스크립트에서 BFAAGENT 사용자에게 해당 목록을 설정할 수 있습니다. 165 페이지의 『System i 플랫폼에 에이전트 설치』에서 예제 시작 스크립트를 참조하십시오. 시작 명령 스크립트에서 라이브러리 목록을 설정하면 예제에서 BFAAGENT 사용자로 실행되는 모든 프로젝트 및 단계에 대한 라이브러리 목록이 설정됩니다. 프로젝트 및 단계를 실행하는 사용자에게 필수 라이브러리에 대한 액세스 권한이 있어야 합니다.

라이브러리 목록을 설정하려면 필수 라이브러리를 나열하는 에이전트에 대한 작업 설명을 추가합니다. 다음 예제 작업 설명에는 라이브러리 FLGHT400 및 FLGHT400M이 포함되어 있습니다.

```
10    UTLIB
20    QGPL
30    QTEMP
40    FLGHT400
50    FLGHT400M
```

에이전트는 시작 루틴에서 이 작업 설명을 지정합니다. 예를 들어, 작업 설명이 BFAJOBID인 경우 시스템 시작 루틴의 행은 다음과 같습니다.



ADDAJE SBSD(BFAGENT/BFAGENT) JOB(BFAGENT) JOBD(BFAGENT/BFAJOBBD)

이 솔루션은 이 에이전트와 연관된 System i 서버에서 실행되는 단계 및 프로젝트의 모든 명령에 영향을 미칩니다.

## bfagent 참조

bfagent executable 실행 파일은 Build Forge 에이전트를 시작합니다. 동일한 디렉토리의 BFAgent.conf 파일에서 구성을 읽습니다.

명령 구문은 다음과 같습니다.

**bfagent** [-f configfile | -s]

옵션

### -f configfile

BFAgent.conf가 아닌 configfile의 구성 파일을 사용하여 실행합니다. 이 옵션은 UNIX 또는 Linux의 런타임 옵션입니다. Windows에서 에이전트를 수동으로 실행하는 경우에는 디버깅 옵션입니다. 이 옵션은 Windows에서 서비스를 시작하는 데 사용할 수 없습니다.

**-s** 독립형 서비스로 시작합니다. 이 옵션은 UNIX 또는 Linux에서만 사용할 수 있습니다. 이와 같이 실행하여 **inetd** 또는 **xinetd**를 통해 bfagent를 시작할 수도 있습니다.

---

## 에이전트 구성

이 절에서는 설치 후 에이전트를 구성하는 방법에 대해 설명합니다.

### 에이전트 구성 파일 찾기

에이전트 구성 파일 BFAgent.conf는 에이전트 오퍼레이션의 런타임 구성을 제공합니다. 이 파일에는 가능한 모든 옵션에 대해 설명하는 주석이 들어 있습니다. 파일은 에이전트 설치 디렉토리에 있습니다.

- Windows 기본값: C:\Program Files\IBM\Build Forge\Agent\BFAgent.conf
- UNIX 및 Linux 기본값: /etc/bfagent.conf

**중요사항:** 설치 디렉토리에서 BFAgent.conf 파일을 변경하는 경우, 이후에 에이전트를 다시 설치하거나 업그레이드한 후에도 변경사항을 반복해야 합니다. 설치할 때마다 구성 파일을 겹쳐씹니다.

대체 구성 파일을 지정할 수 있습니다.

- UNIX 또는 Linux 시스템의 경우, 에이전트 설치 디렉토리 외부의 구성 파일을 사용하여 에이전트 구성을 보존할 수 있습니다. 이를 수행할 때, 에이전트를 시작하는 명령에서 -f 명령행 옵션을 사용하십시오. 예를 들어 다음과 같습니다.

```
bfagent -f /opt/bfagent.conf
```

- Windows 시스템에서는 이 옵션을 사용하여 서비스를 시작할 수 없습니다. 에이전트를 수동으로 실행하는 경우에만 서비스를 사용할 수 있습니다. 이는 디버깅용 도구입니다.

## 에이전트 포트 변경

포트 5555가 이미 사용 중인 서버에 에이전트를 설치하는 경우, 설치 후 에이전트 포트를 변경할 수 있습니다.

Windows 운영 체제에서 포트를 변경하려면 다음을 수행하십시오.

1. 레지스트리 편집기를 여십시오. 시작 > 실행을 클릭한 다음 regedit를 입력하십시오.
2. HKEY\_LOCAL\_MACHINE\SOFTWARE\BuildForge Agent로 이동하십시오.
3. AgentPort 값을 원하는 포트 번호로 변경하십시오.

UNIX, Linux 및 Macintosh 운영 체제의 경우 다음을 수행하십시오.

1. /etc/services 파일을 여십시오.
2. BuildForge Agent Port 값을 원하는 포트 번호로 변경하십시오.

## 다른 셸 구성

BFAgent.conf 파일의 매개변수를 편집하여 기본 셸이 아닌 다른 셸을 사용하도록 에이전트를 구성할 수 있습니다.

예를 들어, Windows 시스템이 MKSTools에서 제공되는 Korn 셸을 사용하도록 변경하려면 다음 명령을 사용하여 셸 매개변수를 변경할 수 있습니다.

```
shell C:\MKSTools\mksnt\ksh.exe -L -c \"%s\"
```

시스템이 서버로 명령을 전송하면 이 명령의 %이 단계 명령으로 바뀝니다. 이 경우, 백슬래시 이스케이프 문자를 사용하여 명령에서 따옴표를 리터럴로 포함합니다.

## 네트워크 공유 파일 시스템에서 에이전트 명령 실행(Windows)

Build Forge 에이전트는 최초 시작 시 Windows 시스템 계정의 신임을 사용합니다. 명령을 실행하기 위해 에이전트에서 나중에 Build Forge 서버 인증 신임을 사용하여 Windows를 인증합니다.

서버 인증 신임은 로컬 명령에 대해서는 문제 없이 승인되지만 에이전트가 외부의 네트워크 공유 드라이브에서 실행해야 하는 일부 명령에 대해서는 실패할 수도 있습니다. 예를 들어 ClearCase의 동적 보기에서 파일을 수정하려면, 에이전트는 네트워크 공유 드라이브 상의 ClearCase 파일에 접근해야 합니다.



외부 파일 시스템은 에이전트의 서버 인증 신임을 무시하고 에이전트의 초기 시스템 계정 신임만 인식하기 때문에 해당 명령은 실패합니다.

네트워크 공유 드라이브에서 명령을 실행하는 중에 문제점이 발생하는 경우 다음 조치를 수행하십시오.

#### 서버 인증 신임을 사용하여 명령 실행

공유 네트워크에 액세스할 수 있는 권한이 있는 Build Forge의 서버 인증 신임을 사용하여 명령을 실행하려면 BFAgent.conf 파일에 win\_reexec\_after\_auth 설정을 추가하십시오.

Build Forge의 서버 인증 신임을 사용하여 공유 네트워크에 대한 액세스를 설정하려면 이 설정을 추가해야 합니다.

win\_reexec\_after\_auth를 설정하면 에이전트는 Windows 인증 후 새 프로세스를 시작하게 됩니다. 새 프로세스를 통해 에이전트에서 사용자 신임을 변경했다는 것을 공유 파일 시스템에서 인식하게 됩니다.

win\_reexec\_after\_auth가 설정되면, 에이전트는 하나의 서비스로 실행되며 공유 네트워크에 액세스하기 위한 명령과 그렇지 않은 명령을 구별하지 않으므로 성능에 영향을 미칠 수 있습니다.

#### 단일 사용자 모드에서 에이전트 실행

에이전트를 설치하는 동안 Build Forge의 서버 인증 신임을 사용하지 않고 단일 사용자 모드에서 명령을 실행하도록 에이전트를 설정하십시오. **Install User Mode Agent** 옵션을 선택하십시오.

지정된 사용자가 관리자 그룹의 멤버인 경우 해당 사용자의 신임은 서버 인증 신임을 사용하여 지정되어야 합니다.

사용자가 관리자가 아닌 경우 에이전트로에 대한 권한이 없는 접근을 방지하기 위해서는 BFAgent.conf에서 magic\_login 설정을 사용하십시오.

사용자가 관리 콘솔에 로그인하면, 에이전트가 기동되며 사용자가 제공한 사용자 이름으로 수행됩니다. 이 때 해당 사용자의 신임 정보를 사용하여 공유 네트워크 접근 권한이 즉시 부여됩니다.

#### 전용 사용자 계정을 가진 서비스로 에이전트를 수행

전용 사용자 계정이 있는 Windows 서비스로 수행되도록 에이전트를 설정합니다. 이 옵션에서 에이전트는 단일 사용자 계정으로 실행되도록 제한됩니다. 그러나 다시 인증하기 위해 에이전트에서 새 프로세스를 시작할 필요는 없습니다. 따라서 성능에 영향을 미치지 않습니다.

전용 사용자 계정이 있는 서비스로 에이전트를 실행하려면 다음을 수행하십시오.

1. Build Forge 서버에서 **관리 도구 > 서비스**를 클릭하여 Windows 제어판을 여십시오. 서비스 목록이 열립니다.

2. IBM Rational Build Forge Agent의 서비스를 여십시오.
3. 에이전트 명령을 실행할 사용자에게 대한 사용자 계정 정보를 제공하십시오.  
예를 들어, ClearCase 관리 사용어나 ClearCase 동적 보기 및 VOB에 대한 액세스 권한이 있는 다른 사용자에게 대한 정보를 제공합니다.

## 에이전트 간 직접 파일 전송 구성

`_XSTREAM_PROTOCOL` 트리거 변수를 사용하면 작업 환경에 표시된 에이전트 간에 직접 파일 전송이 가능합니다. 이 경우 보내는 에이전트, 받는 에이전트 및 엔진이 모두 직접 파일 전송이 가능해야 합니다. 매개변수의 수는 인코딩, 네트워크 및 파일 전송 특성을 제어합니다.

일부 운영 체제의 에이전트의 경우 직접 파일 전송에 대한 지원이 제한되어 있습니다.

- System z: 직접 파일 전송이 지원되지 않습니다.
- System i: PRNG 또는 PLAIN의 `_XSTREAM_PROTOCOL` 값에 대해서만 직접 파일 전송이 지원됩니다. 파일 전송에 SSL을 사용하는 것이 지원되지 않습니다 (AES-CBC의 `_XSTREAM_PROTOCOL` 값).

『bfagent.conf 참조』 참조에서 337 페이지의 『트리거 변수 참조』 및 `xstream_*` 매개변수를 참조하십시오.

## 트리거 변수 및 에이전트 성능

두 개의 트리거 변수가 단계 로그에 대해 생성되는 메시지 수를 줄여서 전체 에이전트 성능에 영향을 줄 수 있습니다.

- `_SUPPRESS_ENV_OUTPUT`: 설정하는 경우 단계 로그에서 ENV 메시지 인쇄를 억제합니다.
- `_SUPPRESS_LOG_OUTPUT`: 설정하는 경우 단계 로그에서 거의 모든 메시지의 인쇄를 억제합니다.

환경에 대한 작업에서 트리거 변수 참조사항을 참조하십시오. 트리거 변수는 프로젝트 환경이나 단계 환경에 포함되는 경우에 작업에 영향을 줍니다.

## bfagent.conf 참조

`bfagent.conf` 파일은 Build Forge 에이전트의 실행 방법에 대한 설정을 저장합니다. 이 파일은 `bfagent` 실행 파일과 동일한 디렉토리에 있습니다.

파일에 모든 설정 및 내부 기본값이 나열되어 있습니다. 비활성 설정은 주석 처리됩니다.

설정

### activity\_log 경로

활동 로깅을 켭니다. 경로에서 지원되는 파일에 정보가 추가됩니다. 경로가 있어야 하며 에이전트 사용자에게 해당 경로에 대한 쓰기 권한이 있어야 합니다.

**참고:** 경로가 없는 경우 또는 경로를 파일에 쓸 수 없는 경우 에이전트에서 오류를 보고하지 않습니다.

**중요사항:** 파일 크기에 대한 한계가 없습니다. 파일을 수동으로 삭제해야 합니다. 이 설정은 에이전트를 디버깅할 때 임시로 사용하기 위한 것이며, 작업 에이전트에 대한 영구 로그로 사용되지 않습니다.

### IP-address-or-range [...] 허용

다음 조건인 경우에만 이 설정을 사용합니다.

- Windows에서 실행 중인 에이전트
- 시작 시 -s 옵션이 사용되는 경우 UNIX 또는 Linux에서 독립형 모드로 실행 중인 에이전트

이 설정은 에이전트에 대한 연결을 제한합니다. IP-address-or-range와 일치하는 IP 주소에서만 연결할 수 있습니다. 기본적으로 연결은 모든 주소에서 허용됩니다.

다음 중 하나 또는 두 항목을 모두 지정하십시오.

- **IP 주소:** 완전한 IPv4 또는 IPv6 주소입니다. 예를 들어, IPv4의 경우, 255.192.192.003. 특정 IP 주소가 허용됩니다.
- **IP 주소 범위:** 부분적으로 규정된 IPv4 또는 IPv6 주소입니다. 예를 들어, IPv4의 경우 192.168 또는 192.168.63입니다. 이 규정과 일치하는 모든 IP 주소가 허용됩니다.

**참고:** 에이전트를 inetd 또는 xinetd와 같은 슈퍼서버에서 실행하는 경우, 다른 메소드를 사용하여 액세스를 제어할 수 있습니다. 방화벽, TCP 래퍼 (hosts.allow and hosts.deny) 또는 xinetd의 내장 필터링 기능을 사용할 수도 있습니다.

**바인드** 이 설정을 사용하면 사용자가 에이전트에 대한 명시적 바인드 주소를 지정할 수 있습니다. 이는 "포트" 설정과 함께, -s 명령행 옵션으로 시작되는 경우 에이전트에서 연결을 청취하는 방법을 결정합니다. bfagent.conf 파일에 지정된 값은 에이전트가 IPv4 localhost 주소에 바인드되도록 하므로, 에이전트에서 동일한 컴퓨터에 있는 콘솔에서의 연결만 수신하게 됩니다(예: bind 255.192.192.003).

**참고:** inetd, xinetd 또는 launchd와 같이 시스템의 서비스 아키텍처에서 시작되는 Windows 또는 Unix 에이전트에는 영향을 주지 않습니다.

### **ccviewroot root-path**

이 설정은 이 호스트의 기본 보기 루트를 지정합니다. 자세한 정보는 init에 대한 ClearCase 문서를 참조하십시오. 내부 기본값은 다음과 같습니다.

- Windows: ccviewroot M:
- UNIX 또는 Linux: ccviewroot /view

### **cc\_suppress\_server\_root**

이를 설정하면 보기 경로가 ccviewroot로 설정되는 경로가 됩니다. 설정하지 않을 경우에는 서버 정의에 설정된 경로가 ccviewroot로 설정되는 경로에 추가됩니다. 이 설정에는 값이 필요 없습니다. 이 설정이 bfagent.conf에 있는 경우에는 설정된 것입니다.

### **command\_output\_cache 크기**

이 설정을 사용하면 지정된 크기(바이트)에 도달할 때까지 에이전트가 출력을 캐시합니다. 내부 기본값은 캐시를 사용하지 않는 것입니다. 캐시를 사용하면 에이전트 성능이 크게 향상되고 네트워크 오버헤드가 감소합니다. 캐시 크기는 명령이 생성하는 출력의 양에 따라 다릅니다.

최소값: 2048. 설정값이 이보다 작으면 2048이 내부적으로 사용됩니다.

### **cygwin**

이 설정은 Windows의 에이전트에만 사용됩니다.

이 설정을 사용하면 에이전트가 Linux와 유사한 환경인 Cygwin을 사용하는 Windows 호스트에서 작업할 수 있습니다. Cygwin을 사용하면 에이전트가 여러 Linux 도구를 사용할 수 있습니다.

이 설정을 사용하는 경우 cygwin\_script\_magic 및 셸 설정을 설정해야 할 수 있습니다. 다음 예제에서는 이러한 설정을 구성할 수 있는 한 가지 방법을 보여줍니다.

```
cygwin
shell C:\cygwin\bin\bash.exe --login -c "%s"
cygwin_script_magic #!/bin/bash
```

셸 설정은 Cygwin 설치와 일치해야 합니다.

### **cygwin\_script\_magic**

이 설정은 **cygwin**이 설정되어 있는 경우 Windows의 에이전트에만 사용됩니다.

이 설정은 #!를 지정합니다. 지정하십시오. 기본값은 #!/bin/bash입니다.

### **default\_logon\_domain**

인증 요청에 도메인이 포함되지 않은 경우 사용할 도메인을 지정합니다. 지정되지 않은 경우, 에이전트 컴퓨터의 도메인이 사용됩니다.

### **disable\_telnet\_support**

최상의 결과를 얻으려면 에이전트 연결 테스트 시 Telnet을 사용합니다.

에이전트의 경우, Telnet 제어 시퀀스의 올바른 처리와 관련하여 고유의 처리 오버헤드가 몇 가지 있습니다.

이 설정을 사용하면 에이전트가 특수 Telnet 문자 코드를 처리할 수 없도록 하여 성능을 다소 향상시킬 수 있습니다. 프로덕션 환경에서 이 설정을 사용하여 향상된 성능으로부터의 이점을 얻을 수 있습니다.

#### **disable\_transcode**

운영 체제가 UTF-8 인코딩을 사용하지 않는 경우 다국적 데이터를 변환하기 위해 에이전트가 수행하는 처리 기능을 끕니다. 혼합 인코딩 및 데이터 손상을 방지하려면 에이전트 운영 체제에 UTF-8을 사용합니다.

운영 체제가 UTF-8 인코딩을 사용하지 않는 경우, 운영 체제의 로케일 설정에 알맞은 인코딩으로 데이터를 변환해야 합니다.

UTF-8을 사용하지 않는 경우, 에이전트의 성능을 향상시키고 최상의 결과를 얻으려면 이 설정을 사용합니다.

#### **enable\_agent\_dll**

이 설정을 사용하면 디버깅 도구인 DLL 프로세스 추적을 사용할 수 있습니다.

#### **env\_recursion\_limit number-of-recursions**

사전 구문 분석에 대한 변수-대체 순환 한계를 설정합니다. 설정하지 않는 경우, 한계는 32입니다.

#### **extensions**

이 설정은 함수의 외부 라이브러리에 대한 경로를 지정합니다. 함수는 단계에서 점 명령으로 사용할 수 있습니다. 이 설정이 지정되어 있지 않으면 외부 라이브러리가 로드되지 않습니다.

구문 분석 중에 단계 명령의 첫 번째 토큰은 함수 이름으로 사용됩니다. 두 번째 토큰은 문자열이고 세 번째는 정수 제한시간 값(초)입니다.

요구사항: 운영 체제에서 동적 로더가 지원되어야 합니다. 예를 들어, UNIX 또는 Linux에서는 /usr/include/dlfcn.h가 필요합니다. 이러한 기본값은 내부적으로 사용됩니다.

- UNIX 또는 Linux: /usr/local/bin/bfextensions.so
- Windows: C:\program files\ibm\build forge\agent\bfextensions.dll

#### **getaddrinfo\_using\_addrconfig**

이 설정은 UNIX 또는 Linux 운영 체제에서 에이전트를 독립형 서비스로 실행하는 경우에만 사용됩니다(bfagent -s). 이 설정으로 청취 인터페이스를 선택하기 위해 getaddrinfo()를 호출할 때 에이전트는 AI\_ADDRCONFIG를 사용합니다. 기본적으로 AI\_ADDRCONFIG는 사용되지 않습니다.

이 설정을 사용하는 경우 에이전트는 올바르게 구성된 주소가 없는 인터페이스를 무시합니다. 올바르게 구성된 주소가 있는 인터페이스만 청취합니다.

**gsk\_ssl\_key\_location** [ <kdb\_path> | <SAF\_specification> ]

kdb 파일의 전체 경로 또는 SAF 키 링 스펙 중 하나를 지정합니다.

**gsk\_ssl\_kdb\_password** <password>

kdb 파일의 비밀번호입니다. 일반 텍스트이거나 암호화된 텍스트일 수 있습니다. SAF 키 링을 사용할 경우 NULL을 사용하십시오. 일반 텍스트에서 암호화된 비밀번호를 작성하려면 **bfagent -e <plaintext>**를 사용하십시오.

**gsk\_ssl\_protocol** <protocol>

사용할 프로토콜(ALL(기본값), SSLV2, SSLV3, TLSV1 또는 TLSV1\_1 중 하나)입니다.

**gsk\_ssl\_cipher\_v2** <seed>

시스템 SSL 버전 2(SSLV2)에 사용할 암호 스위트입니다. 기본값은 6321이며 대부분 애플리케이션에 사용됩니다. 자세한 정보는 System z 문서를 참조하십시오.

**gsk\_ssl\_cipher\_v3** <seed>

시스템 SSL 버전 3(SSLV3)에 사용할 암호 스위트입니다. 기본값은 0906030201이며 대부분 애플리케이션에 사용됩니다. 자세한 정보는 System z 문서를 참조하십시오.

**gsk\_keyring\_label** <label>

kdb 파일의 키 레이블입니다.

**gsk\_password\_encrypt** [true | false]

암호화된 비밀번호를 참조하기 위해 사용됩니다. true로 설정된 경우 **bfagent -e <plaintext>**를 사용하여 암호화된 값을 작성하고 **gsk\_ssl\_kdb\_password**를 설정하십시오. 기본적으로 false로 설정되어 있습니다.

**gsk\_ssl\_client\_authentication** [true | false]

클라이언트 인증서의 유효성을 검증할지 여부를 지정합니다. 기본값은 false입니다.

**lang** lang-code

관리 콘솔에서 올바른 언어를 제공하지 않는 경우에만 이 설정을 사용합니다.

이 설정은 메시지 및 명령 출력을 쓰기 위해 에이전트가 사용하는 언어를 지정합니다. 일반적으로 이는 에이전트가 관리 콘솔에서 지정한 언어를 사용하기 때문에 명시적으로 설정되지 않습니다. 그러나 언어를 설정하면 사용하려는 로케일이 컴퓨터에서 사용할 수 없는 경우에 유용할 수 있습니다. 또한 이 설정은 관리 콘솔이 언어와의 통신에 실패하거나 올바르지 않은 언어와 통신하는 경우에 백업으로 유용할 수 있습니다.

다음과 같이 명시적으로 설정된 것처럼 내부 기본값은 en입니다.

lang en



### **leave\_tmp\_file**

문제점을 해결하는 동안에만 이 설정을 사용합니다.

이 설정을 사용하면 명령 실행 후 임시 파일을 삭제되지 않고 단계 명령을 저장하는 데 사용되는 임시 파일이 유지됩니다. 문제점 해결에서 파일은 관리 콘솔에 표시되는 것처럼 단계와 비교될 수 있습니다.

참고: 일반 오퍼레이션에서는 이 설정을 사용하지 마십시오.

### **locale locale-code.charset-code**

이 설정은 UNIX 및 Linux 운영 체제에서만 사용됩니다. Windows에서는 로케일을 다른 방식으로 처리합니다.

이 설정은 현지화된 애플리케이션이 사용하는 언어와 다중바이트 문자 세트를 지정합니다. 이 설정은 에이전트 컨텍스트에 LANG 환경 변수를 설정하여 적용됩니다.

에이전트가 명령 출력을 미국 영어 UTF-8로 다루도록 설정하려면 운영 체제에 대해 UTF-8 로케일을 사용하십시오. 예를 들어, Linux에서는 다음 표시를 사용하십시오.

```
locale en_US.UTF-8
```

운영 체제에 대한 UTF-8 로케일의 올바른 표시를 판별하려면 **locale -a** 명령을 실행하십시오.

이 설정이 지정되지 않은 경우 에이전트가 운영 체제의 로케일을 사용합니다. 이 설정은 편리합니다. 특히 이 설정은 운영 체제의 기본 로케일이 에이전트가 사용할 로케일이 아닌 경우에 유용합니다. 특히 이 설정은 에이전트 요구사항에 맞추기 위한 시스템 로케일 변경이 효과적이지 않은 경우에 유용합니다.

### **magic\_login user:encoded-password**

일반적으로 에이전트는 관리 특권(예: root 또는 admin)을 사용하여 운영 체제에 로그인합니다. magic\_login 설정은 표준 시스템 인증에 대한 대체 방법입니다. 이 설정을 통해 시스템은 단일 사용자 이름 및 비밀번호를 사용하여 사용자 로그인을 인증할 수 있습니다.

에이전트가 root 또는 admin 사용자로 실행되는 경우, 이 설정은 무시되며 일반 인증이 시도됩니다.

에이전트는 로그인하는 데 사용한 사용자 이름이 아니라 에이전트를 시작한 사용자의 권한을 사용하여 모든 명령을 실행합니다.

이 설정은 다음과 같은 경우에만 사용됩니다.

- 관리 특권으로 에이전트를 실행할 수 없는 경우(예: PAM과 작동하지 않는 UNIX 시스템에서 이 설정 사용)
- 보안 정책으로 인해 관리 특권으로 에이전트를 실행할 수 없는 경우

에이전트에 대한 로그인을 구성하려면 다음을 수행하십시오.

1. 사용자 이름 및 비밀번호를 사용하는 서버 인증을 작성하십시오. 관리 콘솔에서 **서버 > 서버 인증**을 클릭하십시오.
2. 이 예제의 경우 사용자 이름은 **build**이고, 비밀번호는 **MySecretPassword**입니다.
3. 에이전트를 사용하는 서버를 작성하십시오. 인증 필드에서 서버 인증을 이 서버와 연관시키십시오.
4. 에이전트에 대한 인코딩된 비밀번호를 생성하십시오. 에이전트의 설치 디렉토리에서 선택한 비밀번호로 **bfagent -P**를 실행하십시오.

다음과 같이 SMD5 해시 인코드 비밀번호가 리턴됩니다.

```
bfagent -P "MySecretPassword"  
eca0b7f2f4fbf110f7df570c70df844e1658744a4871934a
```

5. BFAgent.conf에서 magic\_login이 원하는 사용자 이름 및 인코딩된 비밀번호를 사용하도록 설정하십시오.  
  
`magic_login build:eca0b7f2f4fbf110f7df570c70df844e1658744a4871934a`
6. 에이전트를 시작하십시오.
7. 서버 연결을 테스트하십시오. 서버에서 서버를 선택한 후 **서버 테스트**를 클릭하십시오.

#### **map** *drive-and-user-spec*[; ...]

이 설정은 매핑된 드라이브를 지정합니다. 일부 시스템에서는 드라이브 매핑이 필요할 수 있습니다. 예를 들어, 셸이 공유 드라이버에서 실행되므로 드라이브 매핑이 필요할 수 있습니다. 에이전트에 지정된 매핑은 관리 콘솔에서 **\_MAP** 환경 변수로 지정된 매핑 전에 수행됩니다. 다음 예제는 두 가지 드라이브 매핑을 보여줍니다.

```
map X:>//host1/share;Z:>//host2/share(username,password)
```

#### **map\_drive\_is\_failure**

이 설정이 지정된 경우 단계 실행 전에 매핑되지 않은 드라이브 스펙이 발견되면 단계에 실패합니다. 이 설정이 지정되지 않은 경우, 단계가 드라이브 실패를 무시하고 단계를 실행합니다. 이 경우 실패로 인해 의미 있는 오류 메시지가 생성되는지 확인하십시오.

#### **no\_preparse\_command**

이 설정은 셸로 명령을 전달하기 전에 일반적으로 에이전트가 명령에서 수행하는 변수-확장 구문 분석을 사용 불가능하게 합니다. 단일 프로젝트 또는 단계에 사용할 수 있는 **\_NO\_PREPARSE\_COMMAND** 환경 변수도 참조하십시오.



## **no\_pty**

이 설정은 UNIX 또는 Linux 시스템에서 실행되는 에이전트에서만 사용됩니다.

이 설정은 셸이 에이전트의 의사 터미널과 상호 작용할 때 시스템 셸이 잠기는 것을 방지하는 데 사용될 수 있습니다. 일반적으로 이 설정은 HP/UX 및 z/OS에 사용됩니다. 또한 두 가지 다른 메소드를 사용하여 이러한 유형의 잠금을 방지할 수 있습니다.

- 대체 셸 사용
- nologonshell 설정 사용

**no\_pty** 설정은 의사 터미널을 할당할 수 없도록 합니다.

**참고:** **no\_pty**를 사용하면 일부 명령에 영향을 미칩니다. 예를 들어, **ls** 명령이 세 개가 아닌 하나의 열로 출력을 리턴합니다. 이 설정을 사용하는 경우, 작업을 프로덕션 환경에 배치하기 전에 철저히 테스트하십시오.

## **nologonshell**

UNIX 또는 Linux에서 실행되는 에이전트에서만 이 설정을 사용합니다.

이 설정은 에이전트가 실행하는 셸이 로그인 셸이 아닌 일반 셸이 되도록 합니다. 이 설정은 다음과 같은 경우에도 종종 유용합니다.

- 로그인 셸이 자세한 출력을 제공합니다.
- 로그인 셸이 원치 않는 방식으로 환경 설정을 변경합니다.
- 로그인 셸이 사용자와 대화식으로 통신하려고 합니다.

설정된 경우 로그인 셸이 아닌 일반 셸을 요청하는 표준 메소드가 사용됩니다. 이는 일부 플랫폼에서 작동하지 않을 수도 있으며, 이 경우 **shellflag** 설정을 사용하여 셸에 플래그를 전달하여 해당 동작을 수정할 수도 있습니다.

이러한 동작은 대화식 사용자가 아닌 사용자로 실행되므로 에이전트에 바람직하지 않습니다.

**참고:** Mac OS X 10.5 시스템은 **nologonshell**에 응답하지 않는 **/bin/bash**를 사용합니다. **shellflag -l**를 사용하십시오.

**참고:** z/OS 운영 체제에서는 항상 로그인 셸 및 비로그인 셸에 **/etc/profile** 스크립트를 사용합니다. 동작이 에이전트에서 제대로 작동되지 않으면 스크립트의 콘텐츠를 변경하거나 다른 셸을 사용해야 할 수 있습니다.

**shellflag** 설정도 참조하십시오. 플래그를 사용하여 로그인 스크립트 동작을 변경할 수 있습니다.

## **password\_encrypt\_module dll\_path;conf\_path**

에이전트에서 SSL을 사용 가능하도록 설정해야 합니다. 이는 구성 파일 및 DLL에 대한 경로를 지정합니다.

- *dll\_path*는 bfcrypt.dll에 대한 경로입니다(일반적으로 ./bfcrypt.dll).
- *conf\_path*는 bfpwcrypt.conf에 대한 경로입니다(일반적으로 ./bfcrypt.conf).

#### **port** *port-number-or-range* [...]

이 설정은 시작 시 **-s** 옵션을 실행하는 경우 UNIX 또는 Linux에서 독립형 모드로 실행 중인 에이전트에서만 사용됩니다.

이 설정은 에이전트가 관리 콘솔과의 연결을 청취하는 데 사용하는 포트를 지정합니다.

에이전트가 관리 콘솔과의 연결을 청취하는 데 사용하는 포트를 지정합니다.

**참고:** 기본적으로, 포트는 5555로 설정됩니다. UNIX 또는 Linux의 경우 /etc/services에서 설정됩니다.

#### **read\_timeout**

에이전트가 연결이 끊길 때까지 요청을 대기하는 시간(초)입니다. 기본값은 1800초(30분)입니다. 제한시간을 사용하지 않으려면 값을 0으로 설정하십시오. 지시문은 적당한 엔진 요청이 수신되지 않는 경우 클라이언트 연결 담당자가 포트를 열어 두지 않도록 예방하는 데 도움이 됩니다. 일부 네트워크 포트 스캐닝 소프트웨어는 이 방식으로 작동합니다.

이 지시문에 대해 아주 작은 값을 설정하지 마십시오. 일반 엔진 동작에서 요청 사이에 몇 분의 차를 둘 수 있습니다.

#### **shell** *shell\_name* [*options*]

이 설정은 기본 셸을 지정합니다. 내부 기본값은 다음과 같습니다.

- Windows: shell cmd.exe /q /c "%s". 다음 설정이 사용되지 않을 때.
  - cygwin 설정이 사용되면 기본값은 shell C:\cygwin\bin\bash.exe --login -c "%s"입니다.
  - cygwin 설정이 사용되지 않는 경우 기본값은 shell cmd.exe /u /q /c "%s"입니다.
- UNIX 또는 Linux: 사용자 계정에 설정된 셸, 또는 사용자 셸을 판별할 수 없는 경우 /bin/sh입니다. 이 설정에서는 매개변수를 지정할 수 없지만 패스하시 위해 shellflag 설정을 사용할 수 있습니다. 에이전트는 하이픈을 삽입하여 기본값이 로그인 셸이 되도록 지정합니다. 예를 들어/bin/ksh는 -ksh로 전송됩니다. shell을 명시적으로 설정하면, nologonshell이 내재적으로 설정됩니다. nologonshell을 참조하십시오.
- System i: 셸 값을 /bin/sh로 설정하십시오.

단계 내에서 이 설정을 겹쳐쓸 수 있습니다. #!을 포함하는 행으로 시작되는 단계가 셸 설정을 겹쳐쓰고 nologonshell 설정을 사용하여 단계 명령을 실행합니다.

### **shell\_compatible\_undef\_vars**

이 설정은 정의되지 않은 변수가 빈 문자열로 표시되도록 합니다. 설정하지 않으면, \$VAR, \${VAR} 또는 %VAR% 형식의 변수에 대한 변수 이름과 \${VAR}에 대한 빈 문자열로 표시됩니다.

### **shellarg**

이 설정은 UNIX 또는 Linux에서 실행되는 에이전트에서만 사용됩니다.

명령이 뒤섞이면 이 설정을 사용하십시오. Red Hat Linux Enterprise의 일부 셸에 이 설정이 필요합니다.

설정된 명령 스크립트를 셸에 전달하는 방법을 변경합니다. 대개, 스크립트는 표준 입력을 통해 전달됩니다.

```
/bin/sh < /tmp/bfshellscript.sh
```

이 설정을 사용하면 스크립트를 매개변수로 전달하여 실행합니다.

```
/bin/sh /tmp/bfshellscript.sh
```

### **shellflag *flag***

이 설정은 UNIX 또는 Linux에서 실행되는 에이전트에서만 사용됩니다.

이 설정은 셸이 실행될 때 플래그를 추가합니다. 하나의 플래그만 지정할 수 있습니다. 이는 일반적으로 출력 또는 원하지 않는 처리를 줄이기 위해 **rc** 스크립트 처리를 사용하지 않도록 설정하는 데 사용됩니다. 예제:

- **csh** 및 파생물: **rc** 스크립트 처리를 사용 불가능하게 하려면 **shellflag -f**를 사용하십시오.
- **bash**: use **shellflag --noprofile**을 사용하여 프로파일 스크립트 처리를 사용 불가능하게 하십시오.

### **ssl\_ca\_location *path***

CA(Certificate Authority)가 포함되어 있는 키 저장소 파일을 지정합니다. 에이전트가 서비스로 실행되는 경우 절대 경로를 사용합니다.

### **ssl\_cert\_location *path***

개인용 인증이 포함되어 있는 키 저장소를 지정합니다. 에이전트가 서비스로 실행되는 경우 절대 경로를 사용합니다.

### **ssl\_client\_authentication [true | false]**

에이전트에 연결할 때 클라이언트 인증이 필요한 경우 **true**로 설정합니다. **true**인 경우 에이전트의 CA(Certificate Authority) 키 저장소에 Build Forge 엔진의 인증을 추가해야 합니다.

### **ssl\_cipher\_group [*grouplist* | ALL]**

사용할 개별 암호 그룹을 지정합니다. **ALL**로 설정할 수 있습니다.

### **ssl\_cipher\_override *cyphers***

암호 그룹을 겹쳐씹니다. 사용할 암호를 지정합니다.

**ssl\_key\_location** *path*

키가 포함되어 있는 키 저장소 파일을 지정합니다. 에이전트가 서비스로 실행되는 경우 절대 경로를 사용합니다.

**ssl\_key\_password** *password*

키의 비밀번호입니다. 기본적으로 이 특성은 일반 텍스트에 저장됩니다. 자체 키 또는 Build Forge 서버의 키를 사용하여 이 비밀번호를 암호화하도록 에이전트를 구성할 수 있습니다.

**ssl\_protocol** *protocol*

사용할 SSL 데이터 교환 프로토콜(SSL, SSLv2, SSLv3, SSL-TLS, TLS 및 TLSv1 중 하나)입니다. 프로토콜은 Build Forge 서버에서 사용되는 프로토콜과 일치해야 합니다.

**update\_path** *path*

이 설정은 Build Forge 에이전트 실행 파일의 전체 경로를 식별합니다. 설정은 설치 중에 자동으로 설정됩니다. 디렉토리는 운영 체제 또는 지정하는 설치 디렉토리의 기본 디렉토리입니다.

참고: Windows 에이전트에서는 이 설정이 무시됩니다. 업데이트 경로는 레지스트리 키에서 가져옵니다. 키는 에이전트 설치 중에 설정됩니다.

**win\_reexec\_after\_auth**

Build Forge 서버 인증 신임을 사용하여 네트워크 공유 파일 시스템에서 에이전트 명령을 실행해야 하는 경우 이 설정을 추가하십시오. 예를 들어 ClearCase의 동적 보기에서 파일을 수정하려면, 에이전트는 네트워크 공유 파일 시스템상의 ClearCase 파일에 접근해야 합니다.

Build Forge 에이전트는 최초 기동 시 Windows 시스템 계정의 신임 정보를 사용합니다. 명령을 실행하기 위해 에이전트에서 나중에 Build Forge 서버 인증 신임을 사용하여 Windows를 인증합니다.

이 설정이 없으면 공유 네트워크에서 초기 Windows 시스템 계정 신임만 인식하며 네트워크 공유 파일 시스템에 파일을 쓰거나 액세스할 때 필요한 이후 서버 인증 신임은 무시합니다.

win\_reexec\_after\_auth는 서버 인증 신임을 사용하여 Windows에서 다시 인증한 후 새 프로세스를 시작하며, 공유 파일 시스템에서 변경된 신임을 인식하도록 합니다.

win\_reexec\_after\_auth 설정을 사용하면 에이전트가 하나의 서비스로 실행되며, 네트워크 공유 파일에 액세스하는 명령과 그렇지 않은 명령을 구별하지 않으므로 성능에 영향을 미칠 수 있습니다.

**xstream\_allow\_ssl\_mismatch**

OpenSSL로 컴파일된 에이전트와 OpenSSL 없이 컴파일된 에이전트 사이에 파

일을 전송해야 할 경우 필수입니다. 기본적으로 OpenSSL로 컴파일된 에이전트에는 AES\_CBC 암호화 파일 전송이 필요합니다. 이 설정이 사용되지 않는 경우 PLAIN 또는 PRNG 인코딩을 사용하여 요청된 파일 전송을 거부합니다.

#### **xstream\_bind ip\_address**

직접 파일 전송에만 사용할 IP 주소를 지정합니다. 주소는 파일을 수신하는 에이전트에서 액세스할 수 있어야 합니다. 기본적으로 에이전트는 모든 네트워크 인터페이스에서 청취합니다. **bind**도 참조하십시오.

#### **xstream\_conn\_timeout 초**

에이전트가 연결을 대기하는 시간(초)입니다. 엔진은 연결 요청을 수신 에이전트로 전달해야 하고, 수신 에이전트는 이 시간 이내에 전송 에이전트와 연결을 설정해야 합니다. 기본적으로 20초로 설정됩니다.

#### **xstream\_listen\_range 포트 범위**

에이전트가 연결에 대해 청취하는 포트 범위입니다. 이 설정은 연결 호스트 간 방화벽이 있을 때 유용합니다. 방화벽 관리는 연결에 허용되는 포트를 허용하도록 방화벽을 구성할 수 있습니다(예: 22880-22889). 기본 포트 범위는 16384-32767입니다. 그러나 xstream\_bind가 사용되고 xstream\_listen\_randomize가 사용되지 않는 경우, 에이전트가 범위를 지정하지 않고 운영 체제가 사용할 포트를 판별합니다.

#### **xstream\_listen\_randomize**

xstream\_port\_range 내에서 포트를 랜덤 선택합니다. 지정되지 않은 경우, 에이전트는 가장 낮은 포트 번호부터 검사를 시작합니다. 이 설정은 보안 기준으로 권장됩니다.

#### **xstream\_recv\_timeout 초**

파일 전송을 대기하는 시간입니다. 파일 전송 중에 이 기간이 지나도 수신 에이전트가 전송 에이전트에서 데이터를 가져오지 못하면 전송이 실패하고 연결이 닫힙니다. 기본값은 20초입니다.

#### **xstream\_send\_timeout 초**

파일 전송을 대기하는 시간입니다. 파일 전송 중에 이 기간이 지나도 전송 에이전트가 데이터를 수신 에이전트에 쓰지 못하면 전송이 실패하고 연결이 닫힙니다. 기본값은 20초입니다.

---

## **에이전트 문제점 해결**

이 절에서는 올바르게 작동하지 않는 에이전트의 문제점을 해결하기 위해 사용할 수 있는 프로시저에 대해 설명합니다. 프로시저를 순서대로 수행하십시오. 이 프로시저를 사용해도 에이전트가 작동되지 않는 경우 지원 부서에 문의하십시오.

## 호스트 이름 해석 테스트

관리 콘솔 호스트에서 에이전트 호스트에 연결할 수 있는지 확인하십시오. 관리 콘솔 호스트에서 ping 유틸리티를 사용하여 에이전트 호스트를 테스트하십시오.

```
ping hostname
```

다음 예제 세션은 관리 콘솔과 에이전트가 모두 설치된 Windows에서 실행됩니다.

```
C:\> ping localhost
```

```
Pinging somehost.city.company.com [127.0.0.1] with 32 bytes of data:
```

```
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
```

```
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
```

```
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
```

```
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
```

```
Ping statistics for 127.0.0.1:
```

```
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
```

```
Approximate round trip times in milli-seconds:
```

```
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

다음과 유사한 메시지는 문제점을 나타냅니다.

```
Unknown host
```

이 문제점은 관리 콘솔 호스트의 네트워크 구성에 있습니다. 네트워크 관리자에게 문의하십시오.

## 연결 테스트

Telnet을 사용하거나 관리 콘솔에서 테스트하여 에이전트에 대한 연결을 테스트할 수 있습니다.

명령행에서 연결을 테스트하려면 다음을 수행하십시오.

1. 에이전트를 Telnet 명령과 연결하십시오. 에이전트가 실행 중인 호스트에 로그인한 경우, 호스트 이름으로 localhost를 사용할 수 있습니다.

```
telnet hostname 5555
```

다음 응답은 연결에 성공했음을 나타냅니다.

```
200 HELLO - BuildForge Agent v7.0.1.buildnumber
```

2. 로그인 신임을 사용하여 다음 명령을 실행하고 인증을 확인하십시오.

```
telnet localhost 5555
```

```
username user name
```

```
password password
```

```
cmd ping go
```

다음 메시지는 성공을 나타냅니다.

```
AUTH: set user account to <user name>
```

위 테스트가 작동하지만 작업에 실패하고 서버 테스트에서 사용자 인증 오류가 표시되면, PAM(Pluggable Authentication Modules) 구성을 확인하십시오. 다음과 같은 메시지가 표시되면 다음 단계를 계속하십시오.

```
AUTH: unable to set user account to <user name>: unknown account (1)
```

3. 다음 명령을 입력하십시오.

```
cmd ping go
```

일반적으로 다음과 같은 Telnet 세션의 출력이 표시됩니다. 출력의 마지막 부분에 있는 RESULT 0은 성공을 나타냅니다. 이 테스트는 Windows에서 실행 중인 에이전트에 실행됩니다.

```
300 DATA s 67
AUTH: Running as: [SYSTEM] in domain [NT AUTHORITY] SID Type [User]
300 DATA s 52
AUTH: Running with Privilege: [Lock pages in memory]
300 DATA s 66
AUTH: Running with Privilege: [Adjust memory quotas for a process]
300 DATA s 63
AUTH: Running with Privilege: [Create permanent shared objects]
300 DATA s 46
AUTH: Running with Privilege: [Debug programs]
300 DATA s 56
AUTH: Running with Privilege: [Bypass traverse checking]
300 DATA s 61
AUTH: Running with Privilege: [Back up files and directories]
300 DATA s 54
AUTH: Running with Privilege: [Change the system time]
300 DATA s 68
AUTH: Running with Privilege: [Remove computer from docking station]
300 DATA s 73
AUTH: Running with Privilege: [Impersonate a client after authentication]
300 HEARTBEAT 1
300 DATA s 16
PLAT: Windows XP
250 RESULT 0
PING: internal loopback test complete
260 EOR
```

관리 콘솔에서 연결을 테스트하려면 다음을 수행하십시오.

**참고:** 서버가 에이전트를 사용하도록 설정된 경우에만 이 메소드를 사용하십시오.

1. 서버로 이동하십시오.
2. 서버 목록에서 테스트할 서버를 클릭하십시오.
3. 연결 테스트를 클릭하십시오.

테스트 후, 결과는 테스트 결과 탭에서 볼 수 있습니다.

다음 출력과 유사한 메시지는 문제점을 나타냅니다.

```
Could not open a connection to host on port 5555
```

관리 콘솔이 호스트에 연결할 수 없거나 에이전트가 기능하지 않습니다.



## Windows에서 에이전트의 문제점 해결

Windows에서 에이전트의 문제점을 해결하려면 다음을 수행하십시오.

1. 설치된 실행 파일을 점검하십시오. 에이전트 설치 디렉토리에 다음 파일이 있는지 확인하십시오.

- `bfagent.exe`
- `bfdispatch.exe`

참고: 계속하려면 먼저 사용자 정의 구성이 있는지 판별하십시오. 사용자 정의 구성이 있는 경우, `BFAgent.conf`를 설치 디렉토리 외부에 저장하고 설치 후 복원하십시오.

2. 에이전트를 다시 설치하십시오. 에이전트를 다시 설치하면 대부분의 Windows 에이전트 문제점을 해결할 수 있습니다. 다시 설치를 통해 실행 파일이 업데이트되고 레지스트리 키가 복원됩니다.

## UNIX, Linux 또는 MacOS에서 에이전트 문제점 해결

UNIX, Linux 또는 MacOS에서 에이전트의 문제점을 해결하려면 다음 프로시저를 수행하십시오.

- 셸에서 **bfagent**를 실행하십시오. 올바른 응답은 다음 메시지와 유사합니다.

```
200 HELLO - Build Forge Agent v7.0.1.122
```

예제와 유사한 메시지가 표시되고 공유 라이브러리 문제점이 있는 경우, 해당 문제점에 대한 메시지가 표시됩니다. 경로를 올바르게 설정하면 대부분의 공유 라이브러리 문제점을 해결할 수 있습니다.

- 에이전트가 청취 중인지 확인하십시오. 다음 명령을 사용하십시오(기본 포트 5555라고 가정).

```
telnet localhost 5555
```

200 HELLO 응답은 에이전트가 청취 중임을 나타냅니다. 이 응답이 표시되지 않는 경우 시스템 네트워크 구성을 확인하십시오. **inetd** 구성이 올바른지 확인하거나, Linux 또는 UNIX 시스템 관리자에게 문의하십시오.

- 인증을 확인하십시오. 로그인 신임을 사용하여 다음 명령을 실행하십시오.

```
telnet localhost 5555
username <user name>
password <password>
cmd ping go
```

다음 메시지와 유사한 메시지는 인증이 올바르게 작동되고 있음을 나타냅니다.

```
AUTH: set user account to <user name>
```



이전 테스트가 작동하지만 빌드에 실패하고 서버 테스트에서 사용자 인증 오류가 표시되면 PAM(Pluggable Authentication Modules) 구성을 확인하십시오. 다음 메시지와 유사한 메시지가 표시되면 다음 프로시저를 계속해서 수행하십시오.

AUTH: unable to set user account to *user name*: unknown account (1)

- PAM 구성을 확인하십시오. PAM 구성의 문제점은 AIX 플랫폼에서의 일반적인 문제입니다. 운영 체제에 따라 PAM은 다음 두 가지 방법 중 하나로 구성됩니다. pam.conf 파일의 한 행으로 또는 pam.d 디렉토리의 파일로 구성됩니다.

팁: Solaris 10은 다음 프로시저에서 예외입니다. pam\_dial\_auth 모듈을 지정하는 행을 삭제하십시오(예: pam\_dial\_auth.so.1). 이 모듈이 포함되면 에이전트 인증이 작동하지 않습니다.

1. /etc/pam.conf가 있는지 확인하십시오. 없는 경우 이 주제의 뒷부분에 있는 pam.d에 대한 지시사항으로 이동하십시오. 파일이 있으면 계속해서 다음 단계를 수행하십시오.
2. 파일에 bfaagent에 대한 항목을 작성하십시오.
3. 다른 애플리케이션(예: sshd 또는 로그인)에 대한 행을 복사하고 [application] 필드 대신 bfaagent를 사용하십시오.

[application] [when] [mode] [module]

필드는 다음과 같습니다.

- [application]은 사용자를 인증해야 하는 애플리케이션의 이름입니다.
- [when]은 인증 요청의 유형입니다.
- [mode]는 인증 요청의 모드입니다.
- [module]은 호출할 인증 모듈입니다. 다음 예제는 로그인에서 bfaagent로 복사된 항목을 보여줍니다. 모듈 이름은 시스템마다 다를 수 있습니다.

bfaagent auth requisite	pam_authok_get.so.1
bfaagent auth required	pam_dhkeys.so.1
bfaagent auth required	pam_unix_cred.so.1
bfaagent auth required	pam_unix_auth.so.1

4. PAM 항목을 설정한 후 3단계에 설명된 대로 다시 로그인하십시오.
  5. 자세한 정보는 <http://www.sun.com/software/solaris/pam>에서 PAM 문서를 참조하십시오.
- pam.d에 구성되어 있는 PAM의 문제점을 해결하려면 다음을 수행하십시오.
    1. /etc/pam.d 디렉토리를 찾고 애플리케이션에 대해 각각 이름이 지정된 여러 개의 파일이 포함되어 있는지 확인해야 합니다. 각 파일 내에서 각 행은 다음과 같이 형식화됩니다.

[when] [mode] [module]
    2. 다른 애플리케이션(예: sshd 또는 로그인)에서 파일을 복사하면 이름이 bfaagent로 바뀝니다.

3. PAM 항목을 설정한 후 3단계에 설명된 대로 다시 로그인하십시오.
4. 자세한 정보는 <http://www.sun.com/software/solaris/pam/>에서 PAM 문서를 참조하십시오.

---

## 제 11 장 설치 후 태스크

이 절에서는 설치를 완료한 후 수행할 태스크에 대해 설명합니다.

다음 주제에 대해 설명합니다.

- 엔진 시작 및 중지
- 사용자 설정
- 설치 확인
- 공통 문제점 해결

---

### 엔진 시작 및 중지

다음 절에서는 엔진 시작 및 중지 방법에 대해 설명합니다.

#### Windows에서 엔진 시작 및 중지

Windows의 경우 다음을 수행하십시오.

- 시작 > 프로그램 > **IBM Rational Build Forge** 관리 콘솔에서 다음 중 하나를 선택하십시오.
  - 엔진 서비스 시작
  - 엔진 서비스 중지

제어판: 관리 도구 > 서비스 제어판을 사용하여 **IBM Rational Build Forge** 관리 콘솔 서비스를 시작하거나 중지할 수도 있습니다.

포그라운드로 실행: 엔진을 실행하는 데 문제점이 있으면, 엔진이 생성하는 상태 및 오류 메시지를 볼 수 있도록 포그라운드로 실행하십시오(시작 > 프로그램 > **IBM Rational Build Forge** 관리 콘솔 > 엔진 시작(포그라운드)). 관리 콘솔이 실행될 때 로그 출력이 콘솔 창에 표시됩니다. 이 모드에서 엔진을 중지하려면 콘솔 창에서 Ctrl-C를 누르십시오.

#### UNIX 또는 Linux 시스템에서 엔진 시작 및 중지

rc 파일이 설치되어 있는 경우 다음 명령을 사용하여 제품을 시작 및 중지하십시오.

```
$ /opt/buildforge/rc/buildforge start
```

```
$ /opt/buildforge/rc/buildforge stop
```

rc 파일이 설치되지 않은 경우, 다음을 사용하여 제품을 시작하십시오.

```
$ /<bfinstall>/Platform/buildforge &
```

프로세스 ID를 판별한 후 kill 명령을 실행하여 중지하십시오.

```
$ ps aux | grep buildforge
$ kill ${<PID>}
```

---

## 사용자 설정

이 절에서는 Build Forge® 시스템에서 사용자 계정을 설정하는 방법에 대해 설명합니다.

다음 주제에 대해 설명합니다.

- 루트 사용자
- 사용자 계정 추가
- 보고서 읽기 전용 사용자

### 루트 사용자

로그인 이름이 "root"인 루트 사용자에게는 특수 특성이 있습니다.

- 설치 시 작성됨: 루트 사용자는 설치 프로그램에서 작성되는 유일한 기본 사용자입니다. 기본 비밀번호는 "root"입니다. 설치 후 비밀번호를 즉시 변경해야 합니다.
- 라이선스 필요 없음: 루트 사용자는 사용자 라이선스를 사용하지 않습니다. 로그인한 사용자 수에 상관없이 항상 루트 사용자로 로그인할 수 있습니다(임의의 사용자가 루트로 로그인하면 루트로 이미 로그인한 다른 사용자가 로그오프됨).
- 시스템 시간대: 루트 사용자의 시간대는 관리 콘솔의 기본 시간대입니다. 기본적으로, 다른 사용자(시스템 내 및 LDAP 사용자)는 루트 사용자의 시간대를 사용합니다. 사용자는 로그인 후 시간대를 설정할 수 있습니다. 시스템에서 보고되는 모든 시간과 로그는 사용자의 시간대로 표현됩니다.
- 모든 권한: 루트 사용자는 사용 가능한 모든 권한을 가지고 있으며, 다른 사용자의 특성을 편집할 수 있습니다. 루트 사용자의 액세스 특권을 제거할 수 없습니다. 루트 사용자가 액세스 그룹의 구성원이 아니더라도 루트 사용자는 시스템의 모든 데이터 오브젝트를 보거나, 편집 또는 사용할 수 있습니다.
- 우선순위: 루트 사용자는 항상 우선순위 사용자입니다.
- 현재 사용자 로그아웃: 루트 사용자는 사용자 로그아웃을 클릭하여 사용자를 로그아웃할 수 있습니다. (관리 > 사용자를 클릭한 후 사용자 이름을 클릭하십시오.)

### 사용자 작성 및 편집

관리 > 사용자를 사용하여 사용자를 작성하고 특성을 지정할 수 있습니다. 시스템을 LDAP/Active Directory 데이터베이스에 연결하여 사용자 정보를 가져올 수도 있습니다. 사용자를 그룹에 지정하여 사용자 보안 권한을 관리하므로 보안 기능을 테스트하려면 일부 사용자를 작성해야 합니다.

현재 사용자의 목록을 표시하려면 **관리 > 사용자**를 클릭하십시오. 목록 뒤에 사용자 패널이 표시됩니다. 시스템에서 이름, 로그인, 이메일, 한계, 활동(마지막 사용자 활동 이후 경과된 시간) 및 각 사용자의 시간대를 표시합니다.

- 사용자를 편집하려면 사용자 이름을 클릭하고 사용자 패널에서 특성을 편집한 후 **저장**을 클릭하십시오.
- 새 사용자를 작성하려면 사용자가 선택되지 않았을 때 사용자 패널에서 특성을 지정하십시오. 사용자가 선택되면, **사용자 추가**를 클릭하여 패널을 지우십시오. 사용자 정보 편집을 완료하면 **저장**을 클릭하십시오.
- 사용자를 로그아웃하려면 사용자 이름을 클릭한 후 **사용자 로그아웃**을 클릭하십시오.
- 고정 라이선스 보유를 사용하려면 먼저 루트로 로그인하십시오. 사용자 이름을 클릭한 후 **라이선스 보유 제거**를 클릭하십시오. 콘솔은 고정 라이선스 세트에 해당되는 ID 목록에서 사용자를 제거합니다. 또한 해당 사용자가 로그인되어 있는 경우 로그 오프됩니다. 고정 라이선스의 경우 콘솔은 로그인한 경험이 있는 사용자의 수를 계산합니다. 한계에 도달하면 새로운 사용자는 라이선스를 취득할 수 없습니다. 다른 사용자가 고정 라이선스를 사용할 수 있도록 기존 사용자를 삭제하거나 제거해야 합니다. 라이선스 보유를 제거해도 콘솔에서 사용자가 삭제되지는 않습니다. 사용자가 다시 로그인하면 고정 라이선스 수가 증가합니다. Floating 라이선스 사용자에서 **라이선스 보유 제거**를 사용하는 경우 이 조치는 **사용자 로그아웃**과 동일한 효과를 나타냅니다.
- 사용자 계정을 삭제하려면 사용자 이름을 클릭한 후 **삭제**를 클릭하십시오.

삭제를 사용할 수 없는 경우 사용자 계정에 스케줄된 작업이 있습니다. 스케줄된 작업이 있는 사용자 계정을 삭제하려면 먼저 스케줄된 작업을 삭제하십시오.

사용자 레코드는 사용자가 시스템과 상호작용할 수 있는 방법에 대한 기본 특성을 설정하고 사용자의 로그인 이름, 비밀번호 및 비밀번호 만기를 제어합니다. 관리 콘솔을 통해 사용자 레코드에 대한 데이터를 시스템에 제공하거나 LDAP/Active Directory 데이터베이스에서 데이터를 파생할 수 있습니다.

**참고:** 레코드가 LDAP 데이터베이스에서 파생된 사용자를 편집하는 경우, 사용자 페이지의 많은 필드가 사용 불가능합니다. 소스 데이터베이스에서 이러한 특성을 변경합니다.

새 사용자를 추가하려면 **사용자 추가**를 클릭하고 패널을 편집한 후 **업데이트**를 클릭하십시오.

사용자 레코드를 표시하면 세 개의 탭이 사용 가능합니다.

- **세부사항:** 편집할 수 있는 사용자 특성을 표시합니다. 사용 가능한 특성은 이 주제의 뒷부분에 설명되어 있습니다.
- **현재 그룹:** 사용자가 직접 또는 다른 그룹의 구성원으로 속해 있는 그룹을 통해 구성원으로 속해 있는 액세스 그룹을 표시합니다.

- **그룹 변경:** 사용자가 직접 구성원으로 속해 있는 그룹을 표시하며, 그룹에 사용자를 추가하거나 그룹에서 사용자를 제거할 수 있습니다.

각 사용자에 대해 **세부사항** 탭에서 다음 특성을 설정할 수 있습니다.

**이름** 사용자의 표시 이름 및 레이블을 지정합니다.

**이메일** 시스템이 이 사용자에게 이메일 알림을 보낼 수 있는 이메일 주소를 지정하십시오.

**참고:** 이메일은 알림에 대해 명시적으로 선택된 사용자에게만 전송됩니다.

**사용자 이름**

사용자가 관리 콘솔에 로그인할 때 입력하는 이름을 지정합니다.

**비밀번호**

사용자가 관리 콘솔에 로그인할 때 제공하는 비밀번호를 입력합니다. 로그인한 사용자에게는 이 필드가 표시되지 않습니다. 이 필드를 사용하여 새 비밀번호를 입력하거나 기존 비밀번호를 변경하십시오. 확인 필드에 동일한 비밀번호를 입력하십시오.

**한계** 사용자가 하루에 실행할 수 있는 최대 작업 수를 지정합니다. 한계에 도달하면, 사용자의 실행 할당량이 초과되었음을 나타내는 메시지가 표시됩니다.

**시간대** 사용자의 시간대를 지정합니다. 시스템은 루트 사용자의 시간대를 게시된 모든 시간의 기본 시간대로 사용합니다.

기본적으로, 시스템 내 사용자와 LDAP 사용자에게는 루트 사용자와 동일한 시간대가 지정됩니다. 사용자가 지정된 시간대를 편집할 수 있습니다.

**참고:** 이전 버전에서 Build Forge 7.1로 업그레이드하는 경우 루트 사용자의 시간대를 수동으로 재설정해야 합니다.

**확인** 비밀번호를 다시 입력하여 올바르게 입력했는지 확인합니다.

**우선순위 로그인**

사용자가 우선순위 사용자인지 여부를 지정합니다. 우선순위 사용자는 항상 시스템에 로그인할 수 있습니다. 사용 가능한 사용자 라이선스가 더 이상 없는 경우 우선순위 사용자가 로그인할 수 있도록 세션이 가장 오래된 사용자가 로그아웃됩니다. 루트 사용자는 항상 우선순위 사용자입니다.

**날짜 형식**

사용자가 선호하는 날짜 형식을 설정합니다.

**언어** 사용자의 기본 언어를 설정합니다.

### 비밀번호 만기

사용자의 비밀번호가 만기됨을 지정합니다. 이 옵션이 선택된 경우, 비밀번호 만기일 수 시스템 설정에 지정된 대로 일 수가 경과하면 사용자의 비밀번호가 만기됩니다.

### 스크린 리더 사용

동적 강조표시 및 초점과 같은 스크린 리더 기능을 지원하는 인터페이스를 사용합니다.

### 달력 시작 요일

스케줄 달력이 맨 먼저 표시하는 요일을 선택하십시오. 기본값은 일요일입니다.

---

## 설치 확인

이 절에서는 설치 및 구성된 Build Forge® 시스템을 테스트하는 방법에 대해 설명합니다.

다음 주제에 대해 설명합니다.

- 서버 구성
- 테스트 프로젝트 작성
- 프로젝트 실행

## 서버 인증

서버 인증을 사용하여 서버에 로그인 신임을 연관시키십시오. 서버 인증 세트 관리를 통해 여러 서버에 동일한 신임을 사용하고 신임을 글로벌로 업데이트할 수 있습니다.

서버 인증은 로그인 이름과 비밀번호를 하나 이상의 서버와 연관시킬 수 있는 하나의 이름 지정된 **오브젝트**로 저장합니다. 서버 인증 페이지를 사용하여 서버 인증을 작성하고 편집하십시오.

## 서버 구성

시스템의 동적 서버 선택사항을 최대한 사용하려면 여러 데이터 오브젝트를 특정 순서대로 설정해야 합니다. 이 주제에서는 서버 사용에 필요한 최소 요구사항을 요약 설명합니다.

### 1. 서버 인증을 작성하십시오.

서버 인증은 서버의 로그인 이름 및 비밀번호를 제공합니다. 둘 이상의 서버에 서버 인증을 적용할 수 있으므로 서버마다 고유한 로그인을 사용할 필요가 없고 로그인을 변경할 때 일련의 시스템에 대해 변경할 수 있습니다.

### 2. 서버 그룹의 콜렉터를 작성하십시오.



컬렉터는 서버로부터 특성을 수집하기도 하고 서버에 특성을 지정하기도 합니다. 컬렉터 없이 서버를 사용할 수 있습니다. BF\_NAME과 같은 기본 특성에 기초하여 서버를 선택해야 합니다.

### 3. 선택기를 작성하십시오.

다음 선택기 유형 작성을 고려하십시오.

- 이름 기반: 각 서버에 대해 호스트 이름을 기반으로 서버를 선택하는 하나의 선택기를 작성하십시오. 그런 다음, 이름별로 서버를 선택할 수 있습니다.
- 운영 체제 기반: 환경의 각 운영 체제 유형에 대해 하나의 선택기를 작성하여 프로젝트가 운영 체제별로 서버를 선택할 수 있도록 하십시오.
- 용량 기반: 사용 가능한 RAM 또는 하드 디스크 공간을 기반으로 서버를 선택하여 보다 특정하게 할 수 있습니다.

### 4. Build Forge 서버로 사용할 각 컴퓨터에 에이전트를 설치하십시오.

### 5. Build Forge에서 사용할 각 컴퓨터에 대해 관리 콘솔에서 서버를 작성하십시오.

### 6. 서버를 테스트하십시오.

연결 테스트를 클릭하여 서버에 대한 연결을 테스트하십시오. Manifest를 검토하여 예상한 특성이 있는지 확인하십시오.

## 테스트 프로젝트 작성

Build Forge® 시스템이 제대로 기능 중인지 확인하려면 11 페이지의 『Hello world 프로젝트 작성』에서 설명하는 대로 단순 프로젝트를 작성하고 실행하십시오.


## 프로젝트 실행

여러 가지 방법으로 프로젝트를 실행할 수 있습니다.

### 시작하기 전에

이 태스크에서는 사용자가 선택기, 서버 및 프로젝트를 이미 작성했다고 가정합니다.

### 프로시저

- 프로젝트 목록이 표시되면 프로젝트가 즉시 시작되도록 프로젝트의 앞에 있는 빠른 시작 아이콘()을 클릭하십시오. 프로젝트에 단계가 없거나 변경 필수 대상 프로젝트 조치가 있는 환경 변수가 있으면 이 방법을 사용할 수 없습니다. 이와 같이 프로젝트를 실행하면 선택기, 클래스, 태그 및 환경 변수의 기본값이 사용됩니다.
- 프로젝트의 단계를 보면서 프로젝트 시작을 클릭하십시오. 이 메소드는 프로젝트의 프로젝트 시작 페이지를 표시하며, 이 페이지에서 프로젝트 매개변수, 환경 변수 값을 변경하고 실행에서 제외시킬 단계를 선택할 수 있습니다.
  - 프로젝트 매개변수의 새 값을 선택하십시오.

- 프로젝트 태그 변수 값을 편집하십시오.
- 프로젝트 환경 변수 값을 편집하십시오. 변경사항을 이러한 변수의 새 기본값으로 저장하려면 **환경 저장** 선택란을 클릭하십시오.
- 작업 단계 탭을 선택하여 프로젝트 단계 목록을 표시하십시오. 이 실행에서만 제외시킬 개별 단계를 선택할 수 있습니다.

선택이 완료되면, **실행**을 클릭하여 프로젝트를 시작하십시오.

- **작업 > 시작**을 선택한 후 프로젝트 이름을 클릭하십시오. 프로젝트 시작을 사용하는 경우와 마찬가지로 이 방법은 프로젝트 시작 페이지를 표시합니다.

작업 세부사항

작업 단계

프로젝트 매개변수

프로젝트 환경

환경 저장 ☐

스냅샷:

기본 스냅샷

선택기:

Selector1

클래스:

스크래치

태그 형식:

BUILD\_\$B

태그 예제:

BUILD\_1

프로젝트 태그

☐ 편집 가능 태그

B

1

## 결과

프로젝트 실행 중에 **작업 > 실행 중** 페이지를 검토하여 프로젝트 상태를 확인하십시오.

작업 결과를 보려면 **작업 > 완료**를 선택하여 완료된 작업을 표시하십시오. 태그 이름을 클릭하여 작업 결과 보기 옵션에 액세스하십시오.

## 로그 위치 및 설정

Rational Build Forge는 여러 개의 컴포넌트로 구성되어 있습니다. 각 컴포넌트는 자체 로그를 가집니다.

### Build Forge

<bfinstall>/db.log

### Apache

<bfinstall>/Apache/logs

## Apache Tomcat

<bfinstall>/Apache/tomcat/logs. Tomcat은 서비스 계층(rbf-services.war) 및 도움말(BuildForgeHelp.war)을 실행하는 데 사용되는 애플리케이션 서버입니다.

**PHP** <bfinstall>/db.log. PHP는 사용자 인터페이스를 실행하는 데 사용됩니다.

## 로그 순환

일부 조건에서는 컴포넌트가 많은 양의 데이터를 씁니다. Tomcat에서 작성된 catalina 로그 파일의 경우가 이에 해당합니다. 디자인상으로는 Tomcat out 로그가 순환되지 않습니다. 특정한 조건에서는 out 로그가 예상보다 훨씬 커질 수 있습니다.

이 문제를 해결할 수 있는 방법은 두 가지입니다.

- 서버의 컨텍스트에서 swallowOutput 속성을 사용하십시오. 출력의 경로가 순환되는 웹 애플리케이션 로그로 재지정됩니다. 세부사항은 Apache Tomcat Configuration Reference를 참조하십시오.
- Catalina 시작 스크립트를 수정하십시오. 사양은 환경 설정에 따라 달라지지만 다음에서는 Catalina 출력을 파이프하고 cronolog 기능을 사용하여 이를 순환시킵니다. 다음 또는 이와 유사한 명령을 포함하도록 <bfinstall>/Apache/tomcat/bin/catalina.sh를 수정하십시오.

```
start >> catalina.out 2>&1 & --replaced with
```

```
start |cronolog /tomcat/logs/%Y-%M-%D.catalina.out >> /dev/null 2>&1 &
```

---

## 공통 문제점 해결

**Solaris에서 에이전트를 업그레이드하려면 pkgrm 명령을 수행해야 함**

pkgadd 명령을 실행하기 전에 pkgrm BFAgent 명령을 사용하여 기존 Solaris Build Forge 에이전트를 제거합니다.

**7.0의 알림 템플릿에 대한 URL이 이후 버전에서는 작동되지 않을 수 있음**

알림 이메일에서 URL 링크를 클릭하면 알림 템플릿 URL에 Build Forge 작업 보고서가 열립니다.

버전 7.0.1 및 7.1에서 알림 템플릿의 URL이 변경되어, 7.0에서 이후 버전으로 업그레이드하면 URL이 작동되지 않을 수 있습니다.

링크 오류가 발생하면 다음 단계를 완료하여 알림 템플릿을 수동으로 편집하십시오.

1. 프로젝트 > 템플릿을 선택하십시오.

2. 알림 템플릿 이름을 클릭하여 세부사항 탭에 해당 특성을 표시하십시오.
3. 본문 필드에서 템플릿에 대한 URL을 찾으십시오. URL은 다음 예와 유사한 형태입니다.

`http://${CONSOLEHOST}:${CONSOLEPORT}/fullcontrol/index.php?mod=projectruns&action=edit&bfid=${PID}&bfid=${BID}&bfid=${UID}`

4. 다음 URL 요소를 적절한 7.0.1 이상 URL 요소로 바꾸십시오.

URL 요소	7.0.1(및 이후 버전) URL 요소
projectruns	jobs
&amp;	&
action=edit	action=build.view
&bfid=\${PID}   &bfid=\${BID}   &bfid=\${UID}	&bf_id=\${BID}

## 제품 응답 없음

Rational Build Forge에서 응답하지 않는 경우 `<bfinstall>/Apache/tomcat/logs/(UNIX 또는 Linux)` 또는 `<bfinstall>/Apache\tomcat\logs\ (Windows)`의 로그에서 다음과 유사한 메시지가 있는지 확인하십시오.

```
Services: 20075: CRRBF1381I: Established connection to Build Forge Services.
DBD::DB2::st execute failed: [IBM][CLI Driver] SQL30081N A communication error has been detected.
Communication protocol being used: "TCP/IP".
Communication API being used: "SOCKETS".
Location where the error was detected: "XXX.XXX.XXX.XXX".
Communication function detecting the error: "recv".
Protocol specific error code(s): "131", "*", "0". SQLSTATE=08001
August 17, 2010 7:18:14 AM EDT
Database: 20075: CRRBFEEEE: DBD::DB2::st execute failed:
[IBM][CLI Driver] SQL30081N A communication error has been detected.
Communication protocol being used: "TCP/IP".
Communication API being used: "SOCKETS".
Location where the error was detected: "XXX.XXX.XXX.XXX".
Communication function detecting the error: "recv".
Protocol specific error code(s): "131", "*", "0". SQLSTATE=08001

Database: 20075: CRRBF0551I: StackTrace from Process id [20075] called from [
BuildForge::DB::db2(./PerlApp/BuildForge/DB/db2.pm:65)
BuildForge::Utilities::SysParams(./PerlApp/BuildForge/Utilities/SysParams.pm:57)
main(.buildforge.pl:305)
main(.buildforge.pl:213)
]
Database: 20075: CRRBF0556I: Trying to call [execute] on a non-existent database handle
DBD::DB2::st fetchrow_hashref failed: no statement executing at /PerlApp/BuildForge/DB/db2.pm
line 78, <$sock> line 1054.
20075: CRRBF0555E: Problem performing Database Operation [fetchrow_hashref] : DBD::DB2::st
fetchrow_hashref failed: no statement executing at /PerlApp/BuildForge/DB/db2.pm
line 78, <$sock> line 1054.

Database: 20075: CRRBFEEEE: DBD::DB2::st fetchrow_hashref failed: no statement executing
at /PerlApp/BuildForge/DB/db2.pm line 78, <$sock> line 1054.

Database: 20075: CRRBF0551I: StackTrace from Process id [20075] called from [
BuildForge::Utilities::SysParams(./PerlApp/BuildForge/Utilities/SysParams.pm:59)
main(.buildforge.pl:305)
main(.buildforge.pl:213)
]
[IBM][CLI Driver] CLI0106E Connection is closed. SQLSTATE=08003 at /PerlApp/BuildForge/DB/Handle.pm
```

```
line 385, <$sock> line 1054.  
Database: 20075: CRRBF0555E: Problem performing Database Operation [prepare] :  
query [SELECT * FROM bf_jobcount WHERE bf_engine_id=?]  
DBD::DB2::db prepare failed: [IBM][CLI Driver] CLI0106E Connection is closed.  
SQLSTATE=08003 at /PerlApp/BuildForge/DB/Handle.pm line 385, <$sock> line 1054.  
  
Database: 20075: CRRBFEEEE: DBD::DB2::db prepare failed: [IBM][CLI Driver] CLI0106E  
Connection is closed. SQLSTATE=08003 at /PerlApp/BuildForge/DB/Handle.pm  
line 385, <$sock> line 1054.
```

여기서, XXX.XXX.XXX.XXX는 데이터베이스를 호스트하는 서버입니다.

이 메시지는 Rational Build Forge 및 데이터베이스 간 연결이 없음을 표시합니다.

연결을 다시 설정하려면 다음을 수행하십시오.

1. 286 페이지의 『엔진 시작 및 중지』에서 설명한 대로 Rational Build Forge를 종료하십시오.
2. 데이터베이스가 실행 중인지 확인하십시오.
3. 286 페이지의 『엔진 시작 및 중지』에서 설명한 대로 Rational Build Forge를 시작하십시오.

---

## 제 12 장 이전 버전에서 업그레이드

다음 주제에서는 이전 버전에서 최신 버전으로 컴포넌트를 업그레이드하는 방법에 대해 설명합니다.

- 다음은 업그레이드를 위한 필수 소프트웨어입니다.

**중요사항:** 필수 소프트웨어를 확인하지 않으면 시스템이 손상되거나 시스템을 사용하지 못하게 될 수 있습니다.

- 관리 콘솔 업그레이드
  - 버전 7.1.x 콘솔 업그레이드
  - 버전 7.0.2.x 콘솔 업그레이드
- 에이전트 업그레이드

---

### 업그레이드를 위한 필수 소프트웨어

업그레이드 설치를 수행하기 전에 먼저 다음을 수행해야 합니다.

- 데이터베이스를 백업하십시오. 37 페이지의 『데이터베이스 백업(필수)』의 내용도 참조하십시오.
- 데이터베이스 및 데이터베이스 클라이언트 버전을 업그레이드하려면 먼저 해당 업그레이드를 수행하십시오. 새 버전을 사용하도록 설치된 시스템은 변경할 수 없습니다. 이 시스템은 설치 프로그램에 지정한 데이터베이스 및 데이터베이스 클라이언트 파일을 사용합니다.
- Apache 및 tomcat용으로 사용자 정의된 구성 파일을 백업하십시오.

```
<bfinsastall>/Apache/conf  
<bfinsastall>/Apache/tomcat/conf
```

httpd.conf 및 php.ini 파일은 일반적으로 Apache용으로 사용자 정의된 파일입니다.

- 시스템이 HTTPS 및 SSL을 사용하도록 구성된 경우 인증서 디렉토리를 백업하십시오. 업그레이드로 인해 HTTPS 및 SSL의 사용에 영향을 미쳐서는 안 됩니다.
- 현재 시스템이 RAFW(Rational Automation Framework for Websphere)와 함께 설치된 경우에는 RAF(Rational Automation Framework)로 마이그레이션해야 합니다. Build Forge 업그레이드 설치 중에 RAFW 데이터는 압축 파일로 저장되어 <bfinstall>/Manager/rafw로 이동됩니다. RAF로 데이터를 이동하는 데 대한 정보는 RAF 문서를 참조하십시오.

---

## 버전 7.1 콘솔 업그레이드

7.1.x 버전에서 현재 버전으로 업그레이드하려면 이 절을 사용하십시오.

업그레이드를 시작하기 전에 223 페이지의 『업그레이드를 위한 필수 소프트웨어』의 내용을 확인하십시오.

업그레이드하려면 업데이트 설치를 수행하십시오.

**참고:** 버전 7.1 콘솔을 업그레이드할 때, 성능 개선사항을 사용할 수 있도록 수행하는 추가 태스크가 있습니다. 세부사항은 jazz.net의 릴리스 페이지를 참조하십시오.

### 업데이트 설치 수행 프로시저

업데이트 설치를 수행하려면 다음을 수행하십시오.

1. Build Forge가 실행 중인 경우 중지하십시오. 그런 다음 Windows에서 실행 중인 경우에는 설치 디렉토리에서 bfengine.pid라는 파일이 있는지 확인하십시오. 파일이 있으면 이를 제거하십시오. 이 파일이 있는 경우 Installation Manager가 업데이트를 수행할 수 없습니다. 포그라운드에서 관리 콘솔을 실행한 경우 해당 작업이 있을 가능성이 높습니다.
2. IBM Installation Manager를 시작하십시오.
3. 업데이트하기 전에 저장소 URL을 업데이트된 저장소로 설정해야 합니다. 지시사항은 87 페이지의 『저장소 URL 지정』의 내용을 참조하십시오.
4. Installation Manager에서 업데이트를 클릭하십시오.
5. 업데이트 패키지에서 업데이트를 찾을 패키지 그룹을 선택한 후 다음을 클릭하십시오.
6. 설치할 업데이트 또는 수정사항을 선택하십시오. 목록에서 설치를 선택한 후 다음을 클릭하십시오.
7. 업데이트할 수 있는 버전 목록이 표시됩니다. 목록에서 설치를 선택한 후 다음을 클릭하십시오.
8. 라이선스 조항에 동의합니다 선택란을 선택한 후 다음을 클릭하십시오.
9. 업데이트 패키지 페이지가 표시됩니다. 기능이 선택되어 있습니다. 다음을 클릭하십시오.
10. 데이터베이스를 구성하고 화면 맨 아래 근처에 있는 연결 테스트를 클릭하여 구성을 테스트하십시오. 테스트가 패스한 후 다음을 클릭하십시오.
11. 다음 페이지에는 buildforge.conf 파일에서 편집한 정보가 표시됩니다. 다음을 묻습니다.



- a. 설치 프로그램에서 필수 데이터베이스 수정을 수행하시겠습니까? 이 선택란을 체크하십시오. 설치 프로그램이 스키마에 대한 필수 변경을 수행하도록 합니다. 이 선택란이 선택되지 않은 경우 스키마 변경사항을 적용하려면 설치 후 `bfschema -u`를 수동으로 실행해야 합니다.
- b. 업그레이드 후 콘솔을 시작하시겠습니까? 설치가 완료된 후 자동으로 콘솔이 시작되도록 하려면 이 선택란을 체크하십시오.

12. 다음을 클릭하십시오.

13. 설치하도록 선택한 기능이 나열됩니다.

14. 설치를 시작하려면 업데이트를 클릭하십시오.

15. 업데이트가 수행되면 완료를 클릭하십시오.

16. Installation Manager를 닫으십시오. Build Forge를 시작하십시오. 버전 7.1.1.x에서 업그레이드하는 경우 CLOB(Character Large Object) 마이그레이션이 완료될 때까지 빌드의 단계 로그가 표시되지 않습니다. Build Forge를 시작할 때 마이그레이션이 시작됩니다. 마이그레이션이 완료되면 Build Forge에서 알려줍니다.

### 버전 7.1.3.1로 업그레이드 후에 어댑터 복사

7.1.1.x에서 7.1.3.1 이상으로 업그레이드한 후에 어댑터를 복사할 경우 문제점이 발생합니다. 일부 어댑터가 버전 7.1.3.1용으로 업데이트되었기 때문입니다. 업그레이드 이전에 작성된 어댑터를 버전 7.1.3.1로 업그레이드한 후에 복사하려고 하면 오류가 보고됩니다.

CRRBF0789I: 올바른 어댑터 템플릿 ID를 지정해야 합니다.

해결 방법: 어댑터 복사를 시도하기 전에 다음을 수행하십시오. 어댑터를 열고 이름을 바꾼 후 저장하십시오. 또한 어댑터를 열고 어댑터의 콘텐츠를 복사한 후 새 어댑터를 작성할 수도 있습니다.

---

## 버전 7.0.2.x 콘솔 업그레이드

버전 7.0.2.x 시스템을 현재 버전으로 업그레이드하려는 경우 이 절을 사용하십시오.

업그레이드하려면 다음을 수행하십시오.

1. 223 페이지의 『업그레이드를 위한 필수 소프트웨어』의 내용을 확인하십시오.
2. 버전 7.0.2.x에서 버전 7.1.1.3으로 업그레이드하십시오.

IBM 지원 센터 사이트에서 버전 7.1.1.3 패키지를 다운로드한 후, *IBM Rational Build Forge* 버전 7.1.1.3 설치 안내서의 지시사항을 따르십시오. 데이터베이스 스키마 변경으로 인해 버전 7.0.2.x에서 버전 7.1.1.3으로의 업그레이드를 위해 사용자가 구성 데이터(프로젝트) 및 선택적으로 히스토리 데이터(작업의 로그)를 마이그레이션해야 합니다.

**중요사항:** 모든 항목이 마이그레이션되지는 않습니다. 특히 사용자 정의된 알림 템플릿(프로젝트 > 템플릿의 항목)가 마이그레이션되지 않습니다.

3. 버전 7.1.1.3에서 업데이트 설치를 수행하여 현재 버전으로 업데이트하십시오.

**참고:** 버전 7.1 콘솔을 업그레이드할 때, 성능 개선사항을 사용할 수 있도록 수행하는 추가 태스크가 있습니다. 세부사항은 [jazz.net](http://jazz.net)의 릴리스 페이지를 참조하십시오.

---

## 에이전트 업그레이드

이전 에이전트에 새 에이전트를 설치하여 에이전트를 업그레이드하십시오.

이전 에이전트 위에 새 에이전트를 설치하여 에이전트를 업그레이드하십시오. 에이전트 지시사항에 따라 에이전트를 설치하십시오.

## 제 13 장 제품 컴포넌트 설치 제거

IBM Installation Manager를 사용하여 Installation Manager를 통해 설치된 제품 컴포넌트를 설치 제거합니다. Build Forge Agent 소프트웨어를 설치 제거하려면 이 절에 설명되어 있는 명령 및 운영 체제 도구를 사용합니다.

### Installation Manager를 사용하여 제품 설치 제거

이 시나리오를 사용하여 Installation Manager를 사용하여 설치한 Build Forge 제품 컴포넌트를 설치 제거합니다.

**참고:** 이 프로시저는 Build Forge용으로 작성된 데이터베이스 또는 데이터베이스 오브젝트를 설치 제거하지 않습니다. 따라서 데이터베이스용 도구를 사용하여 개별적으로 제거해야 합니다. Windows에 설치된 데이터베이스의 경우 제어판 > 프로그램 추가/제거를 사용할 수도 있습니다. 지시사항은 데이터베이스 문서를 참조하십시오. Build Forge 제품 컴포넌트를 제거한 후에 데이터베이스 또는 데이터베이스 오브젝트를 제거하십시오.

제품 컴포넌트를 설치 제거하려면 다음 단계를 완료하십시오.

1. 제품 패키지를 설치할 때 사용한 동일한 사용자 계정을 사용하여 운영 체제에 로그인하십시오.

Installation Manager를 사용하여 설치한 모든 실행 중인 프로그램을 닫으십시오.

2. Build Forge 엔진 및 서비스가 실행 중인 경우 다음과 같이 중지하십시오.

Windows	<ul style="list-style-type: none"><li>• 엔진이 포그라운드로 실행 중인 경우: <b>Ctrl + c</b>를 눌러 엔진 및 Build Forge 서비스를 중지합니다.</li></ul> <p><b>중요사항:</b> 가장 신뢰할 수 있는 엔진 및 서비스 중지 방법은 Ctrl + c를 사용하는 것입니다. 설치 제거할 때 엔진 또는 Build Forge 서비스가 계속해서 실행 중인 경우 설치 제거에 실패합니다.</p> <ul style="list-style-type: none"><li>• 엔진이 포그라운드로 실행 중인 경우: 명령 창을 열고 다음 명령을 입력하여 엔진 및 서비스를 중지합니다.</li></ul> <pre>net stop bfengine71</pre> <p>이러한 명령은 Build Forge 엔진, Apache HTTP Server 및 Apache Tomcat 애플리케이션 서버를 중지합니다.</p>
---------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

UNIX/Linux	1. 명령 창에서 Build Forge rc 디렉토리로 이동합니다. <code>cd /opt/buildforge/rc</code> 2. 엔진을 중지합니다. <code>./buildforge stop</code>  이 명령은 Build Forge 엔진, Apache HTTP Server 및 Apache Tomcat 애플리케이션 서버를 중지합니다.
------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

3. IBM Installation Manager를 시작하십시오.
4. 시작 페이지에서 설치 제거를 클릭하십시오.
5. 패키지 설치 제거 페이지의 설치 패키지 목록에서 설치 제거할 제품 패키지를 선택하고 다음을 클릭하십시오.
6. 요약 페이지에서 설치 제거할 패키지 목록을 검토하고 설치 제거를 클릭하십시오. 패키지가 제거되면 완료 페이지가 표시됩니다.
7. 완료를 클릭하십시오.
8. Installation Manager 창을 닫고 Installation Manager를 종료하십시오.

Build Forge 파일을 정리하거나 DB2 Express 데이터베이스를 선택적으로 제거하거나 제품을 다시 설치하기 전에 Installation Manager를 종료해야 합니다.

## Installation Manager 설치에 실패한 경우 수동으로 제품 설치 제거

Build Forge 엔진과 서비스를 중지하기 전에 IBM Installation Manager를 시작하면 설치 제거에 실패합니다.

Installation Manager에 실패한 후 제품 컴포넌트를 수동으로 설치 제거하려면 다음을 수행하십시오.

1. 명령 창에서 Build Forge 설치 디렉토리에 있는 관리 프로그램 디렉토리로 변경하십시오. 예를 들어, 다음과 같습니다.

Windows	C:\Program Files\IBM\Build Forge\Manager
UNIX/Linux	/opt/buildforge/Manager

2. 명령 프롬프트에 `main.exe uninstall main.res`를 입력하십시오.
3. 명령의 실행이 완료되면 Build Forge 설치 디렉토리를 삭제하십시오.
4. Installation Manager를 사용하여 제품을 다시 설치하십시오.

## Build Forge 파일의 설치 후 제거 정리

설치 제거 후에 몇 가지 수동 정리 작업이 필요합니다.

Installation Manager를 통해 제품 컴포넌트를 설치 제거한 후 다음 디렉토리를 수동으로 삭제합니다. 이러한 디렉토리가 있는 경우 Installation Manager를 사용하여 Build Forge를 다시 설치할 수 없습니다.

디렉토리	기본 위치
Build Forge 설치 디렉토리	<b>Windows:</b> C:\Program Files\IBM\Build Forge <b>UNIX/Linux:</b> /opt/buildforge
공유 파일 디렉토리	<b>Windows:</b> C:\Program Files\IBM\SDPShared <b>UNIX/Linux:</b> /opt/IBM/SDPShared

## Installation Manager를 사용하여 제품 다시 설치 및 기존 DB2 Express 사용

다른 제품 컴포넌트를 설치 제거할 때 제거하지 않은 경우 제품을 다시 설치하고 기존 DB2 Express 데이터베이스를 사용할 수 있습니다. 이 시나리오에서는 기존 Build Forge 프로젝트 및 작업 로그에 액세스할 수 있습니다.

Installation Manager를 시작하기 전에 다음 단계에 있는 검사 목록을 사용하여 필수 DB2 Express 정보를 얻습니다.

**중요사항:** HTTPS 및 SSL을 사용하도록 시스템을 구성한 경우 `<bfinstall>/keystore`와 `<bfinstall>/bfclient.conf`(인증서 비밀번호가 저장되는 위치)에 인증서를 백업해 두어야 합니다. UNIX 및 Linux 시스템의 경우 해당 디렉토리 위치는 `<bfinstall>/Platform`입니다.

기존 DB2 Express 데이터베이스를 다시 설치하고 사용하려면 다음을 수행하십시오.

1. Installation Manager를 시작하십시오.
2. 시작 페이지에서 설치를 클릭하십시오.
3. Installation Manager 마법사의 지시사항에 따라 제품 컴포넌트를 다시 설치하십시오.
4. 데이터베이스 구성 페이지에서 다음과 같이 선택하십시오.
  - a. DB2 Express 설치에 대해 **아니오**를 선택하십시오.
  - b. 설치 시 이 데이터베이스를 채우시겠습니까?에서 **아니오**를 선택하십시오.
  - c. 나머지 필드를 완료하려면 다음 표에 있는 정보를 사용합니다.

### 점검 목록: DB2 Express 구성 정보

✓	필드	설명
	데이터베이스 호스트	제공된 DB2 Express 데이터베이스는 로컬 호스트 이름 (127.0.0.1)에 설치되어 있습니다.
	데이터베이스 이름	데이터베이스 이름은 BUILD(모두 대문자)입니다.

필드	설명
데이터베이스 스키마 이름	데이터베이스 스키마 이름은 BUILD(모두 대문자)입니다.
데이터베이스 사용자 이름	DB2 Express 데이터베이스를 작성할 때 제공한 데이터베이스 사용자 이름입니다.
데이터베이스 비밀번호	데이터베이스 사용자 이름에 대해 제공한 비밀번호입니다.
DB2 클라이언트 라이브러리의 경로	DB2 Express에 대한 DB2 클라이언트 라이브러리 (db2cli.dll)의 경로는 C:\Program Files\IBM\SQLLIB\bin입니다.
JDBC 드라이버 위치	DB2 Express에 대한 JDBC 드라이버(db2jcc.jar)의 경로는 C:\Program Files\IBM\SQLLIB\java입니다.

5. 연결 테스트를 클릭하십시오.
6. 다음을 클릭하여 계속해서 설치를 수행하고 완료하십시오.

## Windows Build Forge 에이전트 설치 제거

Windows의 프로그램 추가/제거 도구를 사용하여 Build Forge 에이전트 소프트웨어를 제거합니다.

**참고:** 설치 제거하는 경우 동일한 버전의 에이전트 소프트웨어를 겹쳐쓸 수 있습니다.

1. Windows에서 프로그램 추가/제거를 찾으십시오. 예를 들어, 시작 → 모든 프로그램 → 제어판 → 프로그램 추가/제거를 선택합니다.
2. 현재 설치된 프로그램 목록에서 IBM Rational Build Forge Agent를 찾으십시오.
3. 변경/제거를 클릭하십시오.
4. 지시사항에 따라 에이전트를 설치 제거하십시오.

## UNIX 또는 Linux Build Forge 에이전트 설치 제거

다음 지시사항에 따라 UNIX 또는 Linux 플랫폼에서 에이전트 소프트웨어를 설치 제거할 수 있습니다.

### Linux 에이전트

rpm 패키지를 사용하여 설치된 에이전트 소프트웨어를 제거하려면 다음을 수행하십시오.

1. 에이전트 소프트웨어를 찾고 패키지 이름과 버전을 나열하십시오.  

```
rpm -qa | grep bfaagent
```
2. 에이전트 소프트웨어를 삭제하십시오.  

```
rpm -e bfaagent-<version_number>
```

## Solaris 에이전트

pkgadd 프로그램을 사용하여 설치된 에이전트 소프트웨어를 제거하려면 다음 명령을 실행하십시오.

```
pkgrm BFAgent
```

## 기타 에이전트

기타 플랫폼의 경우 설치 제거 프로세스가 수동으로 수행되며, 플랫폼에 따라 다릅니다. 플랫폼 및 수퍼서버 구현에 적용되는 지시사항에 따라 수행하십시오.

**참고:** 대부분의 명령을 실행하려면 루트 액세스가 필요하며, /sbin and /usr/sbin 디렉토리가 현재 PATH 환경 변수에 설정되어 있어야 합니다.

1. 에이전트 서비스 디먼 bfaagent를 제거하십시오. 플랫폼에 적용되는 수퍼서버(inetd, xinetd, launchd 또는 SMF) 구현에 대한 지시사항을 사용하십시오.

수퍼서버	프로시저
inetd(이전 UNIX 시스템에 공통)	<ol style="list-style-type: none"><li>1. /etc/inetd.conf 파일을 편집하고 bfaagent에 대한 행을 제거하십시오.</li><li>2. inetd에 대한 프로세스 ID를 찾으십시오.  ps -ef   grep [i]netd  FreeBSD 및 Mac OS/X 10.4 이전과 같은 BSD 파생 시스템의 경우 ps -ef 대신 ps auwwwx를 사용하십시오.</li><li>3. 업데이트된 inetd.conf를 읽고 inetd를 시작하십시오.  kill -HUP &lt;PID&gt;</li></ol>
xinetd(새 UNIX 시스템에 공통)	<ol style="list-style-type: none"><li>1. 에이전트 서비스를 제거하려면 다음 명령을 실행하십시오.  rm /etc/xinetd.d/bfaagent</li><li>2. inetd에 대한 프로세스 ID를 찾으십시오.  ps -ef   grep [i]netd  FreeBSD 및 Mac OS/X 10.4 이전과 같은 BSD 파생 시스템의 경우 ps -ef 대신 ps auwwwx를 사용하십시오.</li><li>3. 업데이트된 inetd.conf를 읽고 inetd를 시작하십시오.  kill -HUP &lt;PID&gt;</li></ol>

수퍼서버	프로시저
launchd(Mac OS/X 및 OpenBSD 시스템의 경우)	<ol style="list-style-type: none"> <li>1. launchctl을 실행하십시오.</li> <li>2. stop com.ibm.rational.bfagent를 편집하십시오.</li> <li>3. 다음 명령을 입력하십시오. unload /Library/LaunchDaemons/com.ibm.rational.bfagent.plist</li> <li>4. quit를 입력하십시오.</li> <li>5. 다음 명령을 실행하십시오. rm Library/LaunchDaemons/com.ibm.rational.bfagent.plist</li> </ol>
Solaris 10의 Solaris SMF(System Management Facility)	<ol style="list-style-type: none"> <li>1. inetadm -d network /bfagent/tcp를 실행하십시오.</li> <li>2. svccfg delete -f network/bfagent/tcp를 실행하십시오.</li> </ol>

2. PAM 인터페이스에서 에이전트 서비스를 제거하십시오.
  - a. /etc/pam.conf를 편집하고 bfagent로 시작하는 모든 행을 제거하십시오.
  - b. rm /etc/pam.d/bfagent를 실행하십시오.
3. etc/services 파일에서 프로토콜 항목을 제거하십시오.  
 /etc/services를 편집하고 bfagent에 대한 행을 제거하십시오.
4. 에이전트에서 설치된 다음 파일을 제거하십시오.  
 /etc/bfagent.conf  
 /etc/bfagent.conf-example  
 /usr/local/bin/bfagent/usr/local/bin/bfcrypt.dll



---

## 제 14 장 릴리스 정보

이 절에는 제품의 현재 릴리스에 대한 릴리스 정보가 들어 있습니다.

---

### 릴리스 정보 - IBM Rational Build Forge 버전 7.1.3.5

*Rational Build Forge* 버전 7.1.3.5를 사용할 수 있습니다. 호환성, 설치 및 기타 시작하기 문제를 다룹니다.

© Copyright International Business Machines Corporation 2003, 2013. All rights reserved.

US Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

#### 컨텐츠

- 『설명』
- 『신규 및 변경된 기능』
- 234 페이지의 『이 릴리스에서 수정된 APAR』
- 234 페이지의 『공지사항』
- 234 페이지의 『이전 버전과의 호환성 문제』
- 234 페이지의 『시스템 요구사항』
- 234 페이지의 『설치』
- 235 페이지의 『알려진 문제점』

#### 설명

*IBM® Rational® Build Forge®*는 소프트웨어 개발의 어셈블리 라인 안에서 각 핸드오프 사이의 모든 프로세스를 자동화, 조직, 관리 및 추적하여 자동화된 소프트웨어 팩토리를 작성하는 적응 프로세스 실행 프레임워크입니다. IBM Rational Build Forge는 사용자의 현재 환경에 통합되며 주요 개발 언어, 스크립트, 도구 및 플랫폼을 지원합니다. 이를 사용하여 기존의 투자를 계속 사용하면서 프로세스 자동화, 가속화, 알림 및 스케줄링에 유용한 기능을 추가할 수 있습니다.

#### 신규 및 변경된 기능

다음 목록은 이 릴리스의 신규 및 변경된 기능을 표시합니다. 최신 기능 목록의 경우 Jazz.net(계정이 필요함)에서 GA 릴리스에 대한 New & Noteworthy 페이지를 참조하십시오.

## 이 릴리스에서 수정된 APAR

이 릴리스에서 수정된 APAR의 최신 목록에 대해서는 Jazz.net(계정이 필요함)에서 GA 릴리스에 대한 릴리스 정보 페이지를 참조하십시오.

## 공지사항

이 버전에 대한 릴리스 공지사항은 [www.ibm.com/common/ssi/index.wss](http://www.ibm.com/common/ssi/index.wss)에 있습니다. 다음 정보에 대해서는 공지사항을 참조하십시오.

- 새 기능의 설명을 포함한 상세한 제품 설명
- 하드웨어 및 소프트웨어 요구사항
- 소프트웨어 서비스 및 지원 가용성
- 패키징 및 순서 지정 세부사항
- 국제적 호환성 정보

## 이전 버전과의 호환성 문제

데이터베이스 테이블 아키텍처가 버전 7.1에서 변경되었습니다. Build Forge 구성 데이터 및 히스토리 데이터가 이전 데이터베이스 테이블 아키텍처로부터 마이그레이션되어야 합니다.

버전 7.1.3.5로 업그레이드하기 위한 요구사항은 현재 버전에 따라 다릅니다.

- 7.1.0.x, 7.1.1.x 및 7.1.2.x: 이들 버전 중 하나를 실행 중인 경우 Installation Manager의 업데이트 기능을 사용하여 업그레이드할 수 있습니다.
- 7.0.2.x: 이들 버전 중 하나를 실행 중인 경우 먼저 버전 7.1.1.3으로 업그레이드한 후 버전 7.1.3.5로 업그레이드해야 합니다.

## 시스템 요구사항

하드웨어 및 소프트웨어 요구사항은 특정 제품에 대한 세부 시스템 요구사항 사이트에서 조회를 통해 보고서에서 확인할 수 있습니다. 사용 중인 제품, 제품 에디션, 버전 및 운영 체제를 제공합니다. 하드웨어, 운영 체제, 데이터베이스, 에이전트 호스트 및 웹 클라이언트에 대한 요구사항이 규정되어 있습니다.

## 설치

단계별 설치 지시사항에 대해서는 Information Center에 있는 설치 섹션을 참조하십시오.


## 알려진 문제점

알려진 문제점은 지원 지식 기반 데이터베이스(IBM Rational Build Forge 지원 사이트)에 개별 TechNotes의 양식으로 문서화되어 있습니다. 문제점이 발견되고 해결되면 IBM 지원 센터 팀이 지식 기반을 업데이트합니다. 지식 기반을 검색하여 해결책이나 솔루션을 빨리 찾을 수 있습니다.

사용자 고유의 조회를 실행하여 추가 유형의 정보를 요청할 수 있습니다.

- 다운로드
- 플래시(경보)
- 뉴스
- 제품 정보 및 출판물
- 문제점 해결

관련 정보:

 IBM 지원 센터에 문의



---

## 제 15 장 대시보드 탭

대시보드 탭은 시스템 상태 및 활동에 대한 개요를 제공합니다.

### 작업 성공

지난 4주간 패스되거나 실패한 작업의 수입니다(주별로 표시). 모든 프로젝트 또는 단일 프로젝트의 상태동향을 보도록 선택할 수 있습니다.

**스케줄** 스케줄된 실행 시간별로 나열되는 나중에 실행하도록 스케줄된 작업입니다. 목록을 통해 페이지 및 목록(모든 또는 제한된 수)을 표시하는 방법을 선택할 수 있습니다.

### 서버 상태

서버 및 해당 상태: 사용 가능 또는 작동 중지. 모든 서버의 상태 또는 사용자가 선택하는 콜렉터와 연관된 서버의 상태만 보도록 선택할 수 있습니다. 목록을 통해 페이지 및 목록(모든 또는 제한된 수)을 표시하는 방법을 선택할 수 있습니다.

### 작업 큐

실행 중인 작업 및 대기 중인 작업 수입니다. 차트는 시간이 경과하면 자동으로 업데이트됩니다. 최대 동시 작업 시스템 설정은 차트의 임계값으로 표시됩니다.

### 제거 큐

실행 중인 제거 및 대기 중인 제거 수입니다. 차트는 시간이 경과하면 자동으로 업데이트됩니다. 최대 동시 제거 시스템 설정은 차트의 임계값으로 표시됩니다.

### 라이선스 시트

사용 중인 라이선스 시트의 수입니다. 표시기 다이얼에 표시되며 "(X)/(총계)개의 라이선스 시트가 사용됨"이라는 캡션이 있습니다.

**참고:** 브라우저 제한사항으로 인해 Internet Explorer를 사용할 때 다이얼에 있는 수는 표시되지 않습니다. 캡션에 있는 수는 올바릅니다.



## 제 16 장 관리

이 주제는 Build Forge® 시스템의 관리 오퍼레이션에 대해 설명합니다.

### 관리 정보

관리 패넌을 사용하여 구성 및 환경 설정을 관리할 수 있습니다.

관리 패넌에서 사용자, 보안 특권, 알림 설정 및 시스템 설정에 대한 작업을 수행하여 관리 콘솔을 구성할 수 있습니다.

관리 패넌에 액세스하려면 왼쪽 메뉴에서 관리를 클릭하십시오.

The screenshot displays the IBM Build Forge management console. At the top, there are tabs for 'IBM', 'UI 구성', '콘솔', '보고서', and '로그아웃: Root User'. The main content area is titled '시스템' (System) and shows a list of system settings. The settings are as follows:

설정명	값
XML 피드에 대한 인증 사용 안함	아니오
감사 로그 자동 정리일 수	365
강제 실행 한계	아니오
경고 로그 자동 정리일 수	120
경보 이메일 제한	10/30

Below the table, there are buttons for '경보 이메일 제한', '저장', and '기본값으로 되돌리기'. A section titled '세부사항' (Details) shows the '경보 이메일 제한' (Warning Email Limit) set to '10/30'. Below this, it states: '값 입력: 시간당 최대 경고 이메일 메시지 수(메시지 수/분). 무제한은 0/0입니다.' (Value input: Maximum number of warning email messages per hour (messages/minute). Unlimited is 0/0). The '기본값' (Default) is also listed as '10/30'.

### 액세스 그룹

액세스 그룹은 시스템이 권한을 제어하는 데 사용하는 사용자 컬렉션입니다.

새 액세스 그룹을 작성하고 사용자를 추가 또는 제거하며 그룹 특성을 수정하려면 관리 > 액세스 그룹 패넌을 사용하십시오. 이 패넌은 기존 액세스 그룹 목록을 표시합니다. 그룹 이름을 클릭하여 그룹을 선택하고 패넌의 하단 분할창에 해당 특성을 표시하십시오.

IBM UI 구성 콘솔 보고서 로그아웃: Root User

액세스 그룹
그룹 추가
도움말 ?

필터
1 표시 중 - 4/6 모두 표시
페이지 1/2

액세스 그룹 이름	새 사용자 기본값	LDAP 그룹 DN
<a href="#">개발자</a>	예	*
<a href="#">게스트</a>	예	*
<a href="#">보안</a>	아니오	
<a href="#">빌드 엔지니어</a>	아니오	

새 그룹
저장
복사
삭제

세부사항
사용자
서브그룹
권한

☐ 기본값

이름:

소유자:

보안

LDAP 그룹 DN:

- 새 그룹을 작성하려면 그룹 추가를 클릭하여 패널의 하단 부분에 있는 필드를 지우십시오.(필요한 경우). 그런 다음, 그룹에 이름을 제공하고 소유자 그룹을 선택하십시오. (소유자 그룹은 새 그룹으로의 액세스를 제어합니다. 그룹을 편집하려면 사용자가 소유자 그룹의 구성원이어야 합니다.) LDAP 인증을 사용 중인 경우, LDAP 그룹 DN 필드를 채워 사용자의 그룹에 맵핑할 LDAP 그룹을 시스템에 알려십시오. 예를 들어, 개발자 그룹을 SoftwareEngineers라는 LDAP 그룹에 맵핑한 후 개발자 그룹에 권한을 지정하여 소프트웨어 엔지니어가 갖게 될 액세스 유형을 제공할 수 있습니다.
- LDAP 그룹 DN 필드에, 구성원이 이 액세스 그룹과 연관된 관리 콘솔 보안 특권을 수신해야 하는 모든 LDAP 그룹의 식별 이름을 나열하십시오.
- 여러 개의 LDAP 그룹을 임의의 액세스 그룹에 맵핑할 수 있습니다. 이 특성에 별표(\*) 문자를 사용하여 모든 LDAP 사용자에게 이 액세스 그룹의 멤버십을 제공할 수 있습니다. 여러 개의 LDAP 그룹을 나열하고 세미콜론으로 그룹을 분리할 수 있습니다.
- 액세스 그룹을 삭제하려면 그룹을 선택한 후 삭제를 클릭하십시오.

**참고:** 액세스 특성으로 다른 곳에 지정된 액세스 그룹은 삭제할 수 없습니다. 예를 들어, 액세스 그룹을 작성하고 이를 사용하도록 프로젝트의 액세스 특성을 설정하는 경우 그룹을 삭제할 수 없습니다. 먼저 프로젝트를 편집하여 다른 액세스 그룹을 사용해야 합니다.



- 사용자를 추가하거나 제거하려면 그룹을 선택한 후 사용자 탭을 클릭하십시오. 시스템이 왼쪽에는 비구성원 목록을, 오른쪽에는 구성원 목록을 표시합니다. 사용자를 선택하고 추가 및 제거를 사용하여 한 목록에서 다른 목록으로 사용자를 이동하십시오.

참고: "액세스 그룹 맵핑 맵"을 "yes"로 설정한 LDAP 사용자는 사용자 탭에 표시되지 않습니다. 이 동작으로 "액세스 그룹 맵핑 맵" 설정은 실제로 Build Forge 수동 설정에 의해 모순되지 않고 그룹 맵핑을 수행할 수 있습니다.

- 그룹을 중첩하려면 다른 그룹의 서브그룹으로 그룹을 추가하십시오. 이를 수행할 때 포함 그룹에 적용되는 모든 권한이 구성원 그룹의 모든 사용자에게도 적용됩니다. 한 그룹이 다른 그룹의 서브그룹이 되도록 하려면 원하는 상위 그룹을 선택한 후 패널의 하단 부분에서 서브그룹 탭을 클릭하십시오. 서브그룹으로 만들 그룹을 선택한 후 추가를 클릭하십시오. 그룹을 재귀적으로 중첩시킬 수 있습니다. 예를 들어, 그룹 A에 그룹 A를 포함하는 그룹 B가 포함되도록 하위에 상위를 추가할 수 있습니다. 이를 수행하는 경우, 시스템은 그룹 A의 모든 구성원을 그룹 B의 구성원으로 간주하고 그룹 B의 모든 구성원을 그룹 A의 구성원으로 간주합니다.
- 그룹의 권한을 관리하려면, 그룹을 선택한 후 권한 탭을 클릭하십시오. 그룹의 현재 권한을 보고 권한을 추가 또는 제거할 수 있습니다.

## 액세스 개요

시스템은 해당 데이터베이스에서 사용자를 관리합니다. 사용자를 지정하는 그룹을 통해 사용자 특권을 제어합니다. 그룹에 특권을 지정한 후 각 사용자가 해당 그룹의 구성원이 되도록 하십시오.

이는 역할 기반 시스템입니다. 그룹은 조직에서 사용자가 가질 수 있는 역할을 표시합니다. 역할에는 특권이 있습니다. 사용자의 특권은 사용자가 속하는 그룹의 합계입니다. 개별 사용자에게 직접 특권을 지정할 수는 없으며 그룹에만 지정할 수 있습니다.

또한 시스템은 알림에 액세스 그룹을 사용합니다. 시스템이 알림 메시지를 보내도록 구성하는 경우 메시지 대상은 액세스 그룹이어야 합니다. 373 페이지의 『알림 설정』의 내용을 참조하십시오.

보안 특권 또는 권한은 그룹이 수행 및 확인할 수 있는 사항을 정의합니다. 이 특권은 그룹의 시스템 경험에 대한 필터로 작동할 수 있습니다. 예를 들어, 게스트 그룹의 구성원이고 다른 그룹의 구성원이 아닌 사용자는 게스트 그룹이 액세스 특성으로 지정된 프로젝트만 볼 수 있습니다. 해당 사용자는 게스트 액세스 권한으로만 프로젝트를 실행할 수 있습니다. 사용자가 개발자 그룹의 구성원이기도 한 경우, 액세스 특성이 게스트 또는 개발자인 모든 프로젝트를 볼 수 있습니다.

참고: 사용자 인증에 데이터베이스 대신 기존의 LDAP 데이터베이스를 사용할 수 있습니다. LDAP을 사용하는 경우, 시스템에 사용자를 정의하는 대신 LDAP 데이

터베이스의 일부 또는 모든 사용자가 시스템에 액세스하도록 허용합니다. 또한 LDAP 그룹에 액세스 그룹을 맵핑할 수 있습니다. LDAP 설정에 대한 세부사항은 251 페이지의 『LDAP 통합 정보』를 참조하십시오.

액세스 그룹을 사용하여 제어할 수 있는 활동 및 자원은 권한, 서버, 프로젝트, 단계 및 액세스 그룹입니다.

- 자원(서버, 프로젝트 또는 단계)에 대한 액세스를 특정 그룹으로 확장하려면 해당 자원을 선택하고 액세스 필드를 해당 그룹의 이름으로 변경하십시오. 예를 들어, 개발자 그룹에 Win234라는 특정 서버에 대한 액세스를 제공하려면 서버의 액세스 특성을 개발자로 설정하십시오.

**참고:** 서버 또는 프로젝트의 액세스 그룹 구성원이 아닌 사용자는 서버 또는 프로젝트 목록 페이지에 나열된 해당 오브젝트를 보지 못합니다. 단계의 액세스 그룹 구성원이 아닌 사용자는 프로젝트 목록에서 단계를 볼 수 있지만 단계를 편집하거나 실행할 수는 없습니다. 사용자가 단계가 포함된 프로젝트를 실행하는 경우 시스템은 사용자에게 액세스 권한이 없는 단계를 건너뛸 것입니다.

- 한 액세스 그룹의 구성원이 다른 액세스 그룹을 편집할 수 있게 하려면 한 그룹을 다른 그룹의 제어 그룹으로 설정하십시오. 예를 들어, 그룹 A 구성원이 그룹 B에 구성원을 추가할 수 있게 하려면 그룹 A가 그룹 B의 제어 그룹이 되게 하십시오.
- 글로벌 특권을 그룹으로 확장하려면 관리 > 권한 페이지를 사용하여 해당 그룹에 특정 권한을 사용 가능하게 하십시오.

이 유연한 모델을 사용하여 한 특권(예: 빌드 실행 기능)을 일부 유형의 사용자에게 안전하게 제공하는 반면에, 다른 특권(예: 프로젝트 편집 또는 특정 서버 사용 권한)을 제한할 수 있습니다.

## 액세스 예제: 작업을 실행할 수 있는 기능을 그룹에 제공

보안 기능을 사용하여 특정 작업을 실행할 수 있는 기능을 액세스 그룹 중 하나로 확장할 수 있습니다. 예를 들어, 다른 작업으로 시스템 보기를 어렵게 하지 않고 사용자 자신의 작업에 관련되는 작업을 실행할 수 있도록 할 디바이스 드라이버 프로그래머 그룹이 있을 수 있습니다. 그러나 프로그래머가 작업을 편집하지 못하도록 하려고 합니다. 이 시나리오를 작성하려면 다음을 수행하십시오.

1. 조직에서 이 역할에 대해 액세스 그룹을 작성하십시오(예: DeviceDriverDevs).
2. 새 액세스 그룹을 사용자가 실행할 수 있게 할 모든 프로젝트의 액세스 특성으로 지정하십시오.
3. 프로젝트의 단계에 해당 액세스 특성이 있는지도 확인하십시오. 작업이 실행될 때 사용자에게 액세스 권한이 없는 단계는 건너뛸 것입니다.
4. 그룹에 빌드 실행 권한을 지정하십시오.

5. 이 빌드를 실행해야 하는 모든 사용자를 새 DeviceDriverDevs 그룹에 추가하십시오. 시스템 관리자가 그룹의 구성원이 되게 해야 할 수도 있습니다. 프로젝트의 액세스 특성을 변경할 때 DeviceDriverDevs 그룹의 구성원이 아닌 사용자는 프로젝트를 보거나 실행 또는 편집할 수 있는 기능을 상실합니다.

사용자는 많은 그룹의 구성원일 수 있으며, 권한은 누적됨에 유의하십시오. 다른 프로젝트 팀(예: PlatformDevs)에 대한 그룹을 가질 수 있으며 두 그룹 모두의 구성원인 사용자는 두 그룹 중 하나가 액세스 특성으로 설정된 프로젝트를 보고 실행할 수 있습니다.

## 팀 및 프로젝트 보안 계획

서로 다른 프로젝트에 대해 작업하는 사용자가 많은 경우, 다음과 같은 일반 계획은 개별 사용자가 필요한 권한을 가져오지만 상호작용해야 하는 프로젝트 및 기타 자원만 볼 수 있도록 이를 관리할 수 있는 기능을 제공합니다.

- 사용자가 수행하는 다양한 활동에 대해 역할 기반의 액세스 그룹을 작성하십시오. 예를 들어, 빌드 관리자 및 개발자 그룹을 작성하십시오. 작업에 맞게 이 그룹에 권한을 지정하십시오. 빌드 관리자는 사용 가능한 대부분의 권한을 갖는 반면, 개발자는 작업 실행과 관련된 권한만 가질 수 있습니다.
- 조직의 각 교차 기능 팀에 대해 추가 그룹을 작성하십시오. IDE 팀, PrinterDriver 팀 및 기타 팀을 가질 수 있습니다.
- 프로젝트, 서버 및 기타 자원의 액세스 특성을 팀 그룹으로 설정하십시오. PrinterDriver 팀과 관련된 모든 프로젝트는 PrinterDriver 액세스 그룹을 액세스 특성으로 가져야 합니다.
- 시스템에 사용자를 추가할 때 사용자를 모든 해당 액세스 그룹에 지정하십시오. 최소 하나의 역할 그룹과 최소 하나의 팀 그룹에 모든 사용자를 지정해야 합니다.

이 가이드라인을 따를 경우, 사용자는 자신과 관련된 프로젝트만 볼 수 있고 해당 프로젝트에서 자신의 역할에 적합한 권한을 갖습니다. 또한 사용자 권한을 조직 변경에서 해당 작업으로 쉽게 변경할 수 있습니다.

## 액세스 특성 관리

사용자는 사용자의 액세스 그룹을 기반으로 액세스 특성을 설정합니다. 프로젝트, 서버 또는 단계와 같은 데이터 오브젝트를 해당 액세스 그룹에 지정하려면 사용자가 액세스 그룹의 구성원이어야 합니다.

예를 들어, 관리 그룹의 구성원이 아닌 경우 해당 그룹에 프로젝트를 지정할 수 없습니다.

액세스 그룹 목록은 사용자가 구성원인 그룹으로 제한됩니다.

단계는 상위 프로젝트로부터 액세스 그룹 특성을 상속합니다. 단계에 프로젝트가 아닌 다른 액세스 그룹 특성이 있을 수 있도록 단계 작성기가 단계의 액세스 그룹 특성을 변경할 수 있습니다. 단계에 지정된 액세스 그룹의 구성원이 아닌 사용자는 단계를 실행할 수 없습니다. 따라서 사용자가 프로젝트에서 특정 단계를 실행하지 못하게 할 수 있습니다.

## 사용자

이 주제는 시스템의 사용자를 관리하는 방법을 설명합니다.

### 사용자 특성

관리 > 사용자를 사용하여 사용자를 작성하여 특성을 지정할 수 있습니다. 시스템을 LDAP/Active Directory 데이터베이스에 연결하여 사용자 정보를 가져올 수도 있습니다. 사용자를 그룹에 지정하여 사용자 보안 권한을 관리하므로 보안 기능을 테스트하려면 일부 사용자를 작성해야 합니다.

사용자 레코드를 표시하면 세 개의 탭이 사용 가능합니다.

- 세부사항: 대부분의 사용자 특성을 편집하려면 이 탭을 사용하십시오. 사용 가능한 특성은 아래에 설명되어 있습니다.
- 현재 그룹: 사용자가 직접 구성원으로 속해 있거나 직접 그룹을 통해 다른 그룹의 구성원으로 속해 있는 액세스 그룹을 표시합니다.
- 그룹 변경: 사용자가 직접 구성원으로 속해 있는 그룹을 표시하며, 그룹에 사용자를 추가하거나 그룹에서 사용자를 제거할 수 있습니다.

The screenshot shows the '세부사항' (Details) tab for user management. The form includes the following fields and options:

- 사용자 이름:** [Text input field]
- 이메일:** [Text input field]
- 이름:** [Text input field]
- 시간대:** [Dropdown menu, currently showing US/Central]
- 날짜 형식:** [Dropdown menu, currently showing 03. 1. 1. 오후 2:30]
- 화면 판독기 사용:** [Dropdown menu, currently showing 마니오]
- 사용자 유형:** [Dropdown menu, currently showing 일반]
- 비밀번호:** [Text input field]
- 확인:** [Text input field]
- 언어:** [Dropdown menu, currently showing 한국어]
- 달력 시작 요일:** [Dropdown menu, currently showing 일요일]
- 한계:** [Dropdown menu, currently showing 무제한]
- 우선순위 로그인:** [Dropdown menu, currently showing 마니오]
- 비밀번호 만기:** [Dropdown menu, currently showing 예]
- 절단:** [Dropdown menu, currently showing 기본값]
- 검색:** [Dropdown menu, currently showing 대소문자 구분]
- 단계 로그 초기 보기:** [Dropdown menu, currently showing 첫 번째 페이지]

각 사용자에 대해 세부사항 탭에서 다음 특성을 설정할 수 있습니다.

**이름** 사용자의 표시 이름 및 레이블

**이메일** 시스템이 이 사용자에게 이메일 알림을 보낼 수 있는 이메일 주소.

**참고:** 이메일은 알림에 대해 명시적으로 선택된 사용자에게만 전송됩니다. 다음 이메일 주소 형식이 지원됩니다. *Username*은 이메일 사용자 이름입니다. *Name*은 임의의 문자열로, 대개 일부 양식에 지정된 이름입니다.

```
username@host.com
name username@host.com
name "<username@host.com>"
name <username@host.com>
name "username@host.com"
```

## 사용자 이름

관리 콘솔에 로그인하는 데 사용되는 이름

## 비밀번호

관리 콘솔에 로그인하는 데 사용되는 비밀번호. 현재 로그인한 사용자에게 대해서는 이 필드가 표시되지 않습니다. 이 필드를 사용하여 새 비밀번호를 입력하거나 기존 비밀번호를 변경하십시오. 확인 필드에 동일한 비밀번호를 입력하십시오.

**한계** 사용자가 하루에 실행할 수 있는 최대 작업 수를 설정합니다. 한계에 도달하면, 시스템은 사용자의 실행 할당량이 초과되었음을 나타내는 메시지를 표시합니다. 값이 0이면, 시스템은 사용자의 빌드 실행 수에 제한을 두지 않습니다.

**시간대** 사용자의 시간대. 시스템은 루트 사용자의 시간대를 게시된 모든 시간의 기본 시간대로 사용합니다. 게시된 시간에서 서버 호스트 시계를 사용합니다.

기본적으로, 시스템 내 사용자와 LDAP 사용자에게는 루트 사용자와 동일한 시간대가 지정됩니다. 사용자는 지정된 시간대를 편집할 수 있습니다.

**확인** 이 필드에 비밀번호를 다시 입력하여 확인하십시오.

## 우선순위 로그인

사용자가 우선순위 사용자가 되도록 이 옵션을 지정하십시오. 우선순위 사용자는 항상 시스템에 로그인할 수 있습니다. 사용 가능한 사용자 라이선스가 더 이상 없는 경우 우선순위 사용자가 로그인할 수 있도록 세션이 가장 오래된 사용자가 로그아웃됩니다. 루트 사용자는 항상 로그인할 수 있으며 라이선스를 이용하지 않습니다.

## 날짜 형식

사용자가 선호하는 표시 형식을 선택합니다.

**언어** 사용자의 언어를 선택합니다.

## 비밀번호 만기

이 옵션이 선택된 경우, 비밀번호 만기일 수 시스템 설정에 지정된 대로 일 수가 경과하면 사용자의 비밀번호가 만기됩니다.

## 스크린 리더 사용

예로 설정하면, 인터페이스에서 시각 장애가 있는 사용자를 위해 동적 강조표시 및 초점 맞추기 등과 같은 스크린 리더 기능이 지원됩니다.

## 사용자 유형

사용자는 다음 세 유형 중 하나에 속할 수 있습니다.

- 일반: 웹 클라이언트를 사용하여 콘솔에 액세스할 수 있습니다. 이 사용자는 해당 액세스 그룹 멤버십 및 권한을 기반으로 시스템에 대한 변경사항을 작성할 수 있습니다. 사용자별로 세션이 하나씩만 허용됩니다.
- 읽기 전용: 웹 클라이언트를 사용하여 콘솔에 액세스할 수 있습니다. 이 사용자는 비밀번호 및 환경 설정과 같은 개인 정보 외의 변경사항을 작성할 수 없습니다. 사용자별로 세션이 하나씩만 허용됩니다.
- API: 웹 클라이언트를 사용하여 콘솔에 액세스할 수 없습니다. 이 유형의 사용자는 Java API 또는 Perl API로 개발된 클라이언트를 통해 시스템에 액세스합니다. 동일한 API 사용자에게는 둘 이상의 로그인 세션이 있을 수 있습니다. 세션 정보는 자동으로 공유됩니다.

## 달력 시작 요일

스케줄 달력이 맨 먼저 표시하는 요일을 선택하십시오. 기본값은 일요일입니다.

## 잘라내기

목록 및 폴다운 메뉴에 표시되는 문자 수를 제어합니다. 예를 들어 20으로 설정되면, 프로젝트 이름은 이름의 처음 20자만 표시합니다.

**검색** 제품을 통해 필터 필드가 검색을 수행하는 방법을 결정합니다. 값은 다음과 같습니다.

- 대소문자 구분(기본값) - 검색은 각 문자의 대소문자를 구분합니다.
- 대소문자 구분 안함 - 검색은 대소문자를 구분하지 않습니다.

## 단계 로그 초기 보기

단계 로그가 작업 결과에서 열릴 때 볼 수 있도록 위치지정되는 방법을 지정합니다. 첫 번째 페이지(기본값)로 설정되는 경우 보기는 로그의 첫 번째 페이지에서 위치지정됩니다. 마지막 페이지로 설정되는 경우 보기는 로그의 마지막 페이지에서 위치지정됩니다.

## 사용자 관리

이 주제는 사용자에 대한 기본 관리 조작을 설명합니다.

**관리 > 사용자**를 선택하여 현재 사용자 목록을 표시하고 이 목록 아래에 사용자 패널을 표시하십시오. 시스템에서 이름, 로그인, 이메일, 한계, 활동(서버 호스트 시계를 사용하여 마지막 사용자 로그인 이후 경과된 시간) 및 각 사용자의 시간대를 표시합니다.

- 새 사용자를 작성하려면 사용자가 선택되지 않았을 때 사용자 패널에서 특성 입력을 시작하십시오. 사용자가 선택되면, **사용자** 추가를 클릭하여 패널을 지우십시오. 사용자 정보 편집을 완료하면 **저장**을 클릭하십시오.
- 사용자를 편집하려면 사용자 이름을 클릭하고 사용자 패널에서 특성을 편집한 후 **저장**을 클릭하십시오.



- 사용자를 로그아웃하려면 사용자 이름을 클릭한 후 **사용자 로그아웃**을 클릭하십시오.
- 비밀번호를 사용하지 않고 사용자로 로그인 하려면 먼저 루트로 로그인하십시오. 사용자 이름을 클릭하고 **사용자로 전환**을 클릭하십시오. 로트 사용자는 비밀번호를 사용하지 않고 사용자로 로그인할 수 있습니다. 현재 비루트 사용자로 로그인되어 있고 다른 비루트 사용자로 로그인하려는 경우 먼저 루트로 로그인해야 합니다.
- 고정 라이선스 보유를 비우려면 먼저 루트로 로그인하십시오. 사용자 이름을 클릭한 후 **라이선스 보유 제거**를 클릭하십시오. 콘솔은 고정 라이선스 세트에 해당되는 ID 목록에서 사용자를 제거합니다. 또한 해당 사용자가 로그인되어 있는 경우 로그오프됩니다. 고정 라이선스의 경우 콘솔은 로그인한 경험이 있는 사용자의 수를 계산합니다. 한계에 도달하면 새로운 사용자는 라이선스를 취득할 수 없습니다. 다른 사용자가 라이선스를 취득하려면 기존 사용자를 삭제하거나 제거해야 합니다. 라이선스 보유를 제거해도 콘솔에서 사용자가 삭제되지는 않습니다. 사용자가 다시 로그인하면 고정 라이선스 수가 증가합니다. Floating 라이선스 사용자를 사용하는 경우, 라이선스 보유 제거는 사용자 로그아웃과 동일한 효과를 가집니다.
- 사용자를 복사하려면 사용자 이름을 클릭한 후 **복사**를 클릭하십시오. "복사" 단어에 추가된 새 사용자의 이름이 목록에 나타납니다.

**참고:** 복사된 사용자의 비밀번호는 password로 재설정됩니다. 비밀번호를 변경하려면 루트 또는 복사된 사용자로 로그인하십시오.

- 사용자 계정을 삭제하려면 사용자 이름을 클릭한 후 **삭제**를 클릭하십시오.

삭제가 사용 불가능한 경우, 사용자 계정에 스케줄된 작업이 있으며 사용자 계정을 삭제할 수 없습니다. 예약된 작업이 있는 사용자 계정을 삭제하려면, 우선 그 작업부터 삭제해야 합니다.

사용자 레코드는 사용자의 시스템 사용 경험에 대한 기본 특성을 설정하고 사용자의 로그인 이름, 비밀번호 및 비밀번호 만기를 제어합니다. 사용자 레코드 데이터는 관리 콘솔을 통해 시스템에 입력되거나 LDAP/Active Directory 데이터베이스에서 파생될 수 있습니다.

**참고:** 레코드가 LDAP 데이터베이스에서 파생된 사용자를 편집하는 경우, 사용자 페이지의 많은 필드가 사용 불가능합니다. 소스 데이터베이스에서 이러한 특성을 변경해야 합니다.

## 사용자 유형을 사용하여 액세스 제어

사용자 유형 특성의 설정은 사용자가 보유하는 액세스 유형을 판별합니다.

- **일반:** 웹 클라이언트를 사용하여 콘솔에 액세스할 수 있습니다. 이 사용자는 해당 액세스 그룹 멤버십 및 권한을 기반으로 시스템에 대한 변경사항을 작성할 수 있습니다.

- **읽기 전용:** 웹 클라이언트를 사용하여 콘솔에 액세스할 수 있습니다. 이 사용자는 비밀번호 및 환경 설정과 같은 개인 정보 외의 변경사항을 작성할 수 없습니다.
- **API:** 웹 클라이언트를 사용하여 콘솔에 액세스할 수 없습니다. 이 유형의 사용자는 Java API 또는 Perl API로 개발된 클라이언트를 통해 작업합니다. 동일한 API 사용자에게는 둘 이상의 로그인 세션이 있을 수 있습니다. 세션 정보는 자동으로 공유됩니다.

LDAP을 통해 로그인하는 사용자는 쓰기 액세스 그룹 DN 특성의 설정에 따라 사용자 유형 일반 또는 읽기 전용으로 판별될 수 있습니다. 251 페이지의 『LDAP 통합』의 내용을 참조하십시오.

## 계정 잠금

시스템 설정을 사용하여 잠금 정책을 강제 실행할 수 있습니다.

잠금 정책은 로그인 시도가 여러 번 실패하면 사용자 로그인이 막히게 되는 조건을 판별합니다. 조건은 연속으로 실패한 최대 로그인 시도 수로 설정됩니다. 타이머는 로그인 세트가 연속적으로 시도되었다고 간주할지 여부를 결정합니다. 시스템은 타이머를 기반으로 실패한 로그인을 "지울 수" 있습니다.

마지막으로 로그인에 성공한 이후로 사용자 계정 로그인에 실패하면 다음 로그인 성공 시 실패한 로그인 시도 횟수를 보여 주는 팝업이 표시됩니다. 팝업은 잠금 조건이 충족되지 않은 경우에만 표시됩니다.

잠금 조건이 충족되면 사용자는 잠금 상태를 입력합니다. 사용자는 다음 중 두 개를 실행해야 할 수 있습니다.

- 다른 로그인 시도가 허용될 때까지 사전 판별된 시간 동안 대기하십시오.
- 계정을 잠금 해제하려면 시스템 관리자에게 문의하십시오.

시스템 설정이 잠금 정책을 판별합니다.

- **계정 잠금 최대 시도 수:** 0인 경우 계정 잠금이 사용 불가능합니다. 이는 잠금을 트리거하는 데 필요한 연속으로 실패한 시도 수로 설정할 수 있습니다. 기본값은 3입니다.
- **계정 잠금 재설정 타이머:** 음수로 설정되는 경우 로그인을 다시 시도하기 전에 해당 계정을 잠금 해제하려면 관리자에게 문의해야 합니다. 사용자가 다시 로그인할 수 있는 데 필요한 분 수로 설정할 수 있습니다. 기본값은 120분입니다.
- **계정 잠금 해제:** 음수로 설정한 경우 실패한 시도 수는 지워지지 않습니다. 분 수로 설정할 수도 있는데, 이 시간이 지나면 마지막으로 실패한 로그인 시도가 지워지며 최대 실패 시도 수로 계수되지 않습니다. 기본값은 60분입니다.

관리 > 시스템을 참조하십시오.



잠긴 계정을 재설정하기 위해 루트 사용자는 `COMMAND_USER_RESET_LOCKOUT` 명령을 사용합니다.

## 루트 사용자

루트 사용자(로그인 이름이 **root**인 사용자)는 시스템 내에서 특별한 특성을 갖습니다.

- 설치 시 작성됨: 루트 사용자는 설치 프로그램이 작성하는 유일한 기본 사용자입니다. 기본 비밀번호는 **root**입니다(설치 후 비밀번호를 즉시 변경해야 함).
- 라이선스 필요 없음: 루트 사용자는 사용자 라이선스를 사용하지 않습니다. 로그인한 사용자 수에 상관없이 항상 루트 사용자로 로그인할 수 있습니다(누군가 루트로 로그인하면 루트로 이미 로그인한 다른 사용자가 로그오프됨).
- 시스템 시간대: 루트 사용자의 시간대는 관리 콘솔의 기본 시간대입니다. 기본적으로, 다른 사용자(시스템 내 및 LDAP 사용자)는 루트 사용자의 시간대를 사용합니다. 사용자는 로그인 후 시간대를 설정할 수 있습니다. 시스템에서 보고되는 모든 시간과 로그는 사용자의 시간대로 표현됩니다.
- 모든 권한: 루트 사용자는 사용 가능한 모든 권한을 갖고 있으며, 다른 사용자의 특성을 편집할 수 있습니다. 루트 사용자의 액세스 특권을 제거할 수 없습니다. 루트 사용자가 액세스 그룹의 구성원이 아니더라도 루트 사용자는 시스템의 모든 데이터 오브젝트를 보거나, 편집 또는 사용할 수 있습니다.
- 우선순위: 루트 사용자는 항상 우선순위 사용자입니다.
- 사용자 로그인: 루트 사용자는 관리 > 사용자 > <사용자 이름> 페이지에서 사용자 로 전환을 클릭하여 비밀번호 사용없이 사용자로 로그인할 수 있습니다.
- 현재 사용자 로그아웃: 루트 사용자는 관리 > 사용자 > <UserName> 페이지에서 사용자 로그아웃을 클릭하여 사용자를 로그아웃할 수 있습니다.
- 기본적으로, LDAP 사용자에게는 루트 사용자와 동일한 시간대가 지정됩니다. 그러나 한 번 로그인한 후에는 LDAP 사용자 자신에게 지정된 시간대를 편집할 수 있습니다. 시스템은 새 환경 설정을 기억합니다.

## API 사용자

Java API 또는 Perl API로 개발된 클라이언트를 통해 콘솔에 액세스하는 사용자는 사용자 유형=API로 설정되어야 합니다. 이 유형의 사용자는 웹 클라이언트를 사용하여 콘솔에 액세스할 수 없지만, 둘 이상의 로그인 세션을 사용할 수 있습니다. 세션 정보는 세션 간에 자동으로 공유됩니다.

동일한 API 사용자에게 의해 열린 다중 세션은 하나의 라이선스만 이용합니다. 라이선스를 비우려면 모든 세션이 로그아웃되어야 합니다. 라이선스를 비우기 위해 루트 계정을 사용할 수도 있습니다. 246 페이지의 『사용자 관리』의 내용을 참조하십시오.

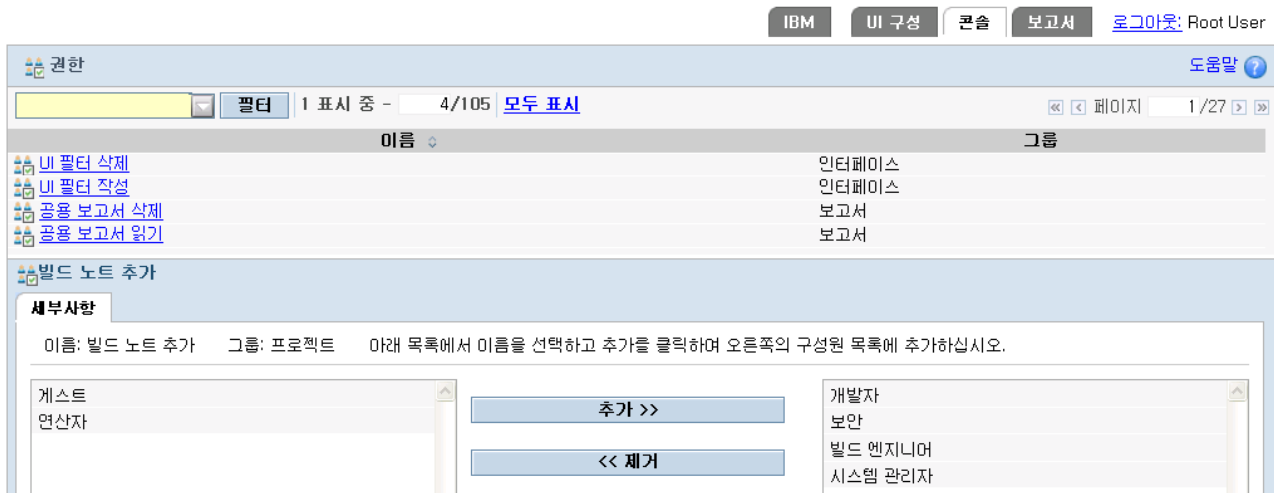
웹 클라이언트와 API 클라이언트를 통해 콘솔에 액세스해야 하는 경우에는 해당 사용자에게 대한 별도의 사용자 ID를 설정하십시오. 웹 콘솔에 액세스하는 사용자의 사용자

유형은 일반 또는 읽기 전용입니다. 이러한 사용자 유형은 여러 세션을 사용할 수 없습니다. 동일한 사용자가 두 번째로 로그인하면 첫 번째 로그인 세션이 종료됩니다.

## 권한

권한은 사용자가 시스템에서 수행할 수 있는 작업을 정의합니다. 액세스 그룹에 권한을 지정하고 사용자에게 직접 권한을 지정하지는 마십시오.

권한에 대해 작업하려면 **관리 > 권한**을 선택하십시오.



그룹에 권한을 지정하려면 권한을 선택하고 원하는 그룹이 해당 권한을 갖는 것으로 나열되도록 하십시오. **관리 > 권한**을 선택하고 목록에서 권한을 선택한 후 패널의 하단 부분에서 작업하십시오. 왼쪽의 그룹에는 권한이 없고 오른쪽의 그룹에는 권한이 있습니다. 그룹에 권한을 제공하려면 왼쪽의 그룹에서 권한을 선택한 후 **추가** 단추를 클릭하십시오.

## 권한 부여 연습

이 예제에서는 새 액세스 그룹에 해당 권한을 제공하고 해당 그룹에 사용자를 지정하여 사용자 Jane Doe에게 서버를 추가 및 편집할 수 있는 독점 권한을 제공합니다.

1. 루트로 로그인하십시오.
2. Server Admin이라는 새 액세스 그룹을 작성하십시오.
3. 새 액세스 그룹에 Jane Doe를 추가하십시오.
4. **관리 > 권한**으로 이동하십시오.
5. 새 서버 추가 권한으로 화면 이동하여 클릭하십시오.
6. 세부사항 탭에서 **추가** 및 **제거**를 사용하여 Server Admin을 이 권한을 가진 유일한 액세스 그룹이 되게 하십시오.

7. 업데이트를 클릭하십시오. 이제 Jane Doe만 시스템에 서버를 추가할 수 있는 권한을 갖습니다.

**참고:** 루트 사용자는 항상 모든 권한을 갖습니다. 루트 사용자의 액세스 특권을 제거할 수 없습니다.

Jane만 서버를 추가할 수 있음에 유의하십시오. 그러나 Jane은 서버의 로그인 정보를 입력할 수는 없습니다. 서버 인증 편집 권한에 Server Admin 그룹을 추가하면 이를 수행할 수 있습니다.

---

## LDAP 통합

Build Forge가 LDAP 서버와 함께 작동하여 사용자를 로그인하도록 설정할 수 있습니다. LDAP을 사용하여, 사용자는 조직 내 다른 위치에서 사용하는 동일한 로그인 이름과 비밀번호를 사용하여 Build Forge에 로그인할 수 있습니다. LDAP를 사용하는 경우 Build Forge에서 수동으로 사용자를 작성하지 않아도 됩니다. 각 사용자는 Build Forge에 처음 로그인할 때 Build Forge에서 작성됩니다.

LDAP 그룹을 Build Forge 액세스 그룹으로 매핑하는 옵션이 있습니다. 이 통합을 사용하면 LDAP에서 그룹을 관리할 수 있으며 사용자가 로그인할 때 사용자 권한이 자동으로 업데이트됩니다.

Build Forge 시스템에 있는 사용자도 계속 수동으로 작성하고 유지보수할 수 있습니다. 그러한 사용자의 액세스는 수동으로 관리해야 합니다.

이 통합을 사용하려면 **관리 > LDAP**에서 항목을 작성합니다.

**참고:** 루트 사용자만 Build Forge 사용자 인터페이스에서 LDAP 도메인 항목을 작성하고 편집할 수 있습니다.

## LDAP 통합 정보

사용자가 LDAP 신임을 사용하여 Build Forge에 처음 로그인하면 해당 사용자가 Build Forge에서 다음과 같이 인증 및 설정됩니다.

**중요사항:** 그룹 매핑을 사용하려면 사용자 로그인 전에 LDAP 그룹 매핑을 사용해야 합니다.

그룹 매핑을 사용하지 않고 사용자가 로그인한 후 나중에 그룹 매핑을 사용하면 기존 사용자에게 매핑이 수행되지 않습니다. 사용자가 로그인한 후에 LDAP 그룹 매핑을 사용하는 경우에는 Build Forge 사용자 목록에서 사용자를 삭제하고 다시 로그인하도록 하십시오. Build Forge 액세스 그룹에 대한 사용자의 멤버십은 작성하는 수동 변경사항이 아니라 LDAP 그룹 매핑을 기반으로 합니다.

1. 사용자는 로그인 패널의 도메인 필드를 봅니다. 여러 도메인이 구성되는 경우 필드는 풀다운 목록입니다. 사용자는 도메인을 선택하고 로그인합니다.

참고: 둘 이상의 도메인을 구성하는 경우, 개별 고유 사용자 ID는 도메인 전체에서 고유해야 합니다. 시스템에서는 고유 사용자당 하나의 로그인만 허용됩니다. 한 사용자가 로그인한 후 다른 사용자가 동일한 고유 사용자 ID를 사용하여 로그인하는 경우 첫 번째 사용자 세션이 닫힙니다. 사용자 세션에 대한 자세한 정보는 7 페이지의 『콘솔 액세스 및 사용』의 내용을 참조하십시오.

2. Build Forge는 LDAP 서버의 계정을 확인합니다. 일반 사용자 또는 관리 사용자를 사용하여 확인을 수행하도록 Build Forge를 구성할 수 있습니다.
3. 사용자 이름이 있으면 Build Forge가 사용자가 Build Forge 로그인 패널에서(또는 API 클라이언트를 사용하는 프로그램의 로그인에서) 제공한 신임을 사용하여 LDAP 로그인을 시도합니다.
  - 신임이 일치하지 않거나 사용자 이름이 없으면 로그인이 실패합니다.
  - 신임이 일치하면 로그인이 진행됩니다.
4. 사용자가 이전에 로그인하지 않은 경우, Build Forge는 자동으로 사용자 목록에 사용자를 작성합니다. LDAP을 통해 로그인하는 사용자는 사용자 이름, 비밀번호, 로그인, 확인 및 이메일 필드를 사용할 수 없습니다. 이러한 정보는 LDAP이 제공하기 때문입니다.

참고: 시스템은 LDAP에서 시간대 정보를 가져오지 않으므로 처음 로그인할 때 루트 사용자의 시간대에 LDAP 사용자를 지정합니다. 이후에는 시간대를 수동으로 설정할 수 있습니다.

5. Build Forge는 사용자에게 액세스 그룹을 적용합니다.
  - LDAP 그룹 매핑을 사용하는 경우 지정된 액세스 그룹이 적용됩니다. 기본 Build Forge 액세스 그룹 또한 적용됩니다. 그룹 매핑을 사용하려면 Build Forge LDAP 도메인 특성의 구성이 필요합니다.

참고: 그룹 매핑은 사용자가 로그인할 때마다 수행됩니다. 이를 통해 Build Forge가 LDAP의 그룹 멤버십 변경사항과 동기화됩니다.

- LDAP 그룹 매핑을 사용하지 않는 경우에는 Build Forge 기본 액세스 그룹이 적용됩니다. 이 경우 액세스 그룹 멤버십을 수동으로 관리할 수 있습니다.

## LDAP 도메인 특성

작성된 LDAP 도메인의 특성을 편집하려면 다음을 수행하십시오.

1. 관리 > LDAP<도메인 이름>을 선택하십시오.
2. 편집할 도메인을 선택하십시오. LDAP 도메인 특성 패널에 특성이 표시됩니다.

새 LDAP 도메인 저장 복사 삭제 기본값으로 설정 LDAP 도메인 테스트

세부사항

이름: 관리 DN: cn=Administrator,cn=users,dc=example,dc=com 액세스 그룹 매핑: 관리자

호스트: ldap.example.com:389 표시 이름: displayname 권한 부여 그룹 DN: 쓰기 액세스 그룹 DN: 검색 기준: cn=users,dc=example,dc=com 그룹 검색 기준: 그룹 고유 ID:

비밀번호: 식별 이름: distinguishedname

확인: 그룹 이름: memberof

사용자 계정 바인드: 메일 이름: mail

프로토콜: LDAP 고유 ID: sAMAccountName=%

3. 필드 값을 편집한 후 저장을 클릭하십시오. 다음 필드는 필수입니다.

- 이름
- 호스트
- 바인드 사용자 계정
- 프로토콜
- 표시 이름
- 식별 이름
- 메일 이름
- 고유 ID

4. 도메인을 기본값으로 사용하려면 기본값으로 설정을 클릭하십시오.

**이름** 필수. Build Forge 내에서 LDAP 도메인의 이름입니다. 최소 하나 이상의 LDAP 도메인이 구성되어 있는 경우, Build Forge 로그인 양식은 이 이름을 기준으로 해당 도메인을 나열합니다.

#### 관리 DN

LDAP 서버 데이터베이스에 검색 액세스를 제공하기 위해 사용하는 계정입니다. 서버가 데이터베이스 검색에 익명의 바인드를 허용하는 경우, 이 필드를 공백으로 두십시오.

일부 LDAP 서버에서 데이터베이스를 검색하려면 관리 바인드가 필요합니다. 이 설정을 사용하여 다음 예제에 표시된 대로 관리자 계정의 DN을 지정할 수 있습니다.

cn=Administrator,cn=users,dc=example,dc=com

**비밀번호 및 비밀번호 확인** 필드에 관리자 DN 계정의 비밀번호를 지정하십시오.

#### 액세스 그룹 매핑

LDAP 서버의 그룹 정보를 관리 콘솔의 액세스 그룹에 매핑할지 여부를 판별

합니다. 기본값은 아니오입니다. Build Forge의 각 액세스 그룹에는 해당 LDAP 그룹 DN 특성이 LDAP의 올바른 그룹 이름으로 설정되어 있어야 합니다.

- 아니오인 경우, LDAP 그룹이 Build Forge 액세스 그룹에 맵핑되지 않습니다. 사용자는 최소한 한 번 로그인한 후에만 Build Forge의 액세스 그룹에 지정될 수 있습니다. 이 옵션을 사용하면 Build Forge 내에서 사용자의 액세스 그룹을 관리할 수 있습니다. 사용자가 처음 로그인하고 Build Forge에서 작성된 사용자 이름을 가지고 있을 때 기본 액세스 그룹이 적용됩니다.
- 예인 경우, Build Forge는 사용자가 Build Forge에 로그인할 때마다 사용자의 LDAP 서버에서 그룹 멤버십 정보를 새로 고칩니다. 마지막 로그인 이후로 Build Forge 내에서 작성된 사용자에 대한 액세스 그룹 멤버십의 모든 변경사항을 겹쳐씹니다. 이 옵션을 사용하면 LDAP의 모든 그룹 멤버십을 관리할 수 있습니다. LDAP 그룹 멤버십은 Build Forge의 액세스 그룹에 자동으로 맵핑(추가 또는 제거)됩니다. 사용자의 그룹 멤버십을 판별하기 위해 다음과 같이 그룹 특성을 사용합니다.
  1. 그룹 이름이 공백이 아닌 경우 지정된 키워드의 값에 대해 조회하십시오. 사용자의 그룹으로 리턴된 값을 사용하십시오.
  2. 그룹 이름이 공백이거나 해당 조회가 값을 리턴하지 않으면, 그룹 검색 기본 및 그룹 고유 ID를 사용하여 사용자가 속한 LDAP 그룹에 대해 조회하십시오.
  3. (1) 및 (2)에서 그룹 정보가 리턴되지 않으면, 사용자는 로그인하도록 허용되고 새로운 사용자의 기본 액세스 그룹으로 지정되는 액세스 그룹의 멤버십이 지정됩니다.

**호스트** 필수. LDAP 서버의 호스트 이름 및 포트입니다. 예제:

ldapservers.mycompanyname.comldap.mycompany.com:9000

**비밀번호**

관리자 DN 계정의 비밀번호입니다. 관리자 DN이 지정되어 있는 경우 필수입니다.

**확인** 관리자 DN 비밀번호의 반복 항목입니다.

**바인드 사용자 계정**

필수. Build Forge가 로그인 시 LDAP에 대하여 사용자 신임의 유효성 검증을 시도하는지 여부를 판별합니다. 기본값은 예입니다.

- 예인 경우, Build Forge는 LDAP 서버에 로그인 시 제공된 사용자 이름 및 비밀번호를 확인합니다.
- 아니오인 경우, Build Forge는 유효성 검증 없이 사용자 이름을 승인합니다. Build Forge에 대한 외부 비밀번호 유효성 검증(예: SSO(단일 사인온))이 구현되면 이 설정이 사용됩니다.

## 프로토콜

필수. Build Forge 사용자 인증을 목적으로 Build Forge가 디렉토리 서비스에  
서 데이터를 읽고 쓰는 데 사용하는 프로토콜을 식별합니다. 기본값은 LDAP  
입니다. SSL(LDAPS)을 통해 LDAP를 사용하는 경우 LDAPS를 입력하십시  
오. 이 옵션에 대한 추가 설정은 필수입니다. 보안 LDAP(LDAPS) 사용을 참  
조하십시오.

## 표시 이름

필수. 사용자의 전체 이름을 지정하는 키 이름을 입력하십시오.

## 식별 이름

필수. 사용자 계정의 식별 이름을 지정하는 키 이름을 입력하십시오.

## 메일 이름

필수. 사용자의 이메일 주소를 지정하는 키 이름을 입력하십시오.

## 그룹 이름

LDAP 스키마에 사용자가 구성원인 그룹의 목록을 보유한 키 이름을 입력하십  
시오. 액세스 그룹 매핑이 예이거나 권한 부여된 그룹 DN이 사용되는 경우에  
만 사용됩니다.

## 권한 부여된 그룹 DN

LDAP 그룹의 식별 이름입니다. 설정되어 있으면, 지정된 그룹의 구성원만 로  
그인할 수 있습니다. 공백이면, 모든 올바른 LDAP 사용자가 콘솔에 로그인할  
수 있습니다.

## 쓰기 액세스 그룹 DN

사용자에게 일반 또는 읽기 전용 액세스 권한이 있는지 여부를 판별합니다. 값  
은 다음 중 하나입니다.

- 공백 - 새 로그인인 경우 사용자 유형은 일반으로 설정됩니다. 기존 사용자는 지정된 사용자 유형(일반, 읽기 전용 또는 API)을 유지합니다. 유형은 관리 > 사용자에서 설정됩니다.
- \*(별표) - 모든 로그인에 대해 사용자 유형이 일반으로 설정됩니다.
- LDAP 그룹 이름 - 사용자가 그룹에 속하는 경우 사용자 유형은 일반으로 설정됩니다. 사용자가 그룹에 속하지 않는 경우 사용자 유형은 읽기 전용으로 설정됩니다.
- 기타 - 모든 사용자를 강제로 읽기 전용으로 설정하려면 다른 값을 사용하십시오, 예제: RO.

## 검색 기준

필수. 사용자의 LDAP 레코드를 조회하는 데 사용되는 검색 문자열입니다. 예를 들어 다음과 같습니다.

cn=users,dc=buildforge,dc=com



## 고유 ID

필수. 사용자가 로그인 시 입력하는 사용자 이름과 비교해야 하는 LDAP 데이터베이스의 필드를 식별합니다. 사용자가 입력한 로그인 이름에 % 문자를 사용하십시오. 예를 들어 다음과 같습니다.

```
(sAMAccountName=%)
```

## 그룹 검색 기준

그룹 고유 ID를 필요로 합니다. 액세스 그룹 매핑이 에이저나 권한 부여된 그룹 DN이 사용되는 경우에만 사용됩니다. 그룹 데이터에 대한 LDAP 레코드를 조회하는 데 사용된 검색 문자열입니다. LDAP 데이터베이스가 사용자 레코드를 저장하는 데 사용된 데이터베이스와 구별되는 데이터베이스에 그룹 멤버십을 저장하는 경우에 필요합니다. 예를 들어 다음과 같습니다.

```
cn=groups,dc=buildforge,dc=com
```

## 그룹 고유 ID

그룹 검색 기본을 필요로 합니다. 액세스 그룹 매핑이 에이저나 권한 부여된 그룹 DN이 사용되는 경우에만 사용됩니다. LDAP 사용자 데이터베이스에서 그룹 멤버십 정보를 얻기 위해 사용할 필드를 식별합니다. 필터는 사용자 계정의 임의 데이터 필드를 그룹 테이블의 키로 사용할 수 있습니다. %fieldname% 구문을 사용하여 필드를 식별하십시오. 그룹 테이블이 sAMAccountname 필드를 사용자의 키로 사용하는 경우 다음 예제가 작동합니다.

```
sAMAccountName=%sAMAccountname%
```

## 태스크

이 주제에서는 LDAP 도메인에 대한 작업의 태스크를 식별합니다.

### LDAP 도메인 항목 작성

원하는 수의 LDAP 도메인 항목을 작성할 수 있습니다. 사용자가 로그인을 시도하는 경우 도메인이 지정되어야 합니다. API 클라이언트 로그인인 경우 로그인 호출에서 도메인을 지정하거나 사용할 도메인을 bfcclient.conf에 구성해야 합니다.

도메인 항목을 추가하려면 다음을 수행하십시오.

1. **관리 > LDAP**를 선택하십시오.
2. **LDAP 도메인 추가**를 클릭하십시오.
3. 도메인의 특성을 채우거나 변경하십시오. 이름 특성은 Build Forge의 내부입니다. 기본적으로 제공되는 값은 LDAP 또는 표준 Microsoft® Active Directory 서버에 대한 작업을 수행하도록 설계되어 있습니다.
4. **저장**을 클릭하십시오.



## LDAP 도메인 항목 테스트

LDAP 도메인이 올바르게 설정되었는지 확인하려면 다음을 수행하십시오.

1. 관리 > LDAP를 선택하십시오.
2. 목록에서 도메인을 선택하십시오.
3. LDAP 도메인 테스트를 클릭하십시오.

시스템은 LDAP 도메인 항목의 특성을 사용하여 LDAP 서버를 조회합니다.

## 보안 LDAP(LDAP) 사용

LDAP 서버가 SSL을 통한 LDAP(LDAP)를 지원하는 경우 Build Forge LDAP 도메인 항목 또한 LDAP를 사용하도록 구성할 수 있습니다. 기본적으로 엄격한 SSL이 구성됩니다. 엄격한 SSL에는 서버 인증이 필요합니다.

1. Build Forge에서 LDAP 도메인 항목을 작성하십시오.
2. 프로토콜 특성을 LDAP로 설정하십시오. 이 경우 LDAP의 암호화 전용 방법을 사용합니다.
3. 호스트를 LDAP 서버의 완전한 도메인 이름과 SSL 포트로 설정하십시오. 포트 636은 엄격한 보안 LDAP에 정의된 기본값입니다(예: myldap.mycompany.com:636).
4. LDAP 서버에서 서명자 인증서를 가져와 Build Forge 신뢰 저장소에 추가하십시오. 아웃바운드 LDAP는 기본적으로 관리 > 보안의 다음 설정을 사용하도록 구성됩니다.
  - SSL 패널: 기본 JSSE 아웃바운드 SSL
  - 키 저장소 패널: 기본 JSSE 신뢰 저장소. 이 신뢰 저장소는 기본적으로 `<bfinstall>/keystore/buildForgeTrustStore.p12`를 사용하도록 설정됩니다. 여기에 서명자 인증서를 저장하십시오.
5. Build Forge를 다시 시작하십시오.
6. 관리 > 보안으로 이동하여 보안 LDAP 구성을 선택하십시오.
7. 연결 테스트를 클릭하십시오.

참고: 엄격한 LDAP SSL은 Build Forge에서 기본적으로 설정됩니다. 엄격한 구성에서는 서버 인증서 유효성 검증이 필요합니다. 엄격한 LDAP를 사용하지 않으려면 다음을 수행하십시오.

1. JAVA\_OPTS 환경 변수에 Tomcat 시스템 특성 `-Dcom.buildforge.services.server.ldap.strict=false`를 설정하십시오. Tomcat 스크립트는 이 변수를 읽고 Tomcat 프로세스에 지정된 시스템 특성을 적용합니다.
2. Build Forge를 다시 시작하십시오.

이 구성에서는 LDAP 서버 인증서를 Build Forge 신뢰 저장소에 추가하지 않아도 됩니다. 그러나 이 구성은 SSL 프로토콜 디자인의 취약한 구현입니다. Build Forge는 통신 중에 LDAP 서버의 ID를 확인하지 않습니다.

## LDAPS SSL 구성 변경

아웃바운드 LDAP 요청이 사용하는 SSL 구성은 기본적으로 설정됩니다. 해당 측면 중 다음 두 가지를 변경할 수 있습니다.

- SSL 구성. LDAP 서버가 기본 프로토콜 또는 데이터 교환을 사용하여 Build Forge와 통신할 수 없는 경우 이를 수행해야 합니다.
- 키 저장소 구성. 엄격한 SSL에서는 클라이언트(Build Forge)가 LDAP 서버와의 안전한 통신을 위해 사용하는 서명자 인증서를 신뢰 저장소에 저장해야 합니다. 다른 신뢰 저장소를 사용하거나 다른 위치에 저장하려면 Build Forge에 신뢰 저장소의 새 키 저장소 구성을 작성해야 합니다.

이러한 지시사항은 Build Forge에 대한 보안 LDAPS를 이미 사용 가능하게 했고 Build Forge에 대한 SSL 컴포넌트는 사용 가능하게 하지 않은 것으로 가정합니다.

LDAPS SSL 구성을 변경하려면 다음을 수행하십시오.

1. 신뢰 저장소의 위치 또는 이름을 변경하는 경우 Build Forge 호스트에서 원하는 위치에 배치하십시오. LDAP 서버의 서명자 인증서를 추가하십시오.
2. 필요한 경우 **관리 > 보안 > 키 저장소**에서 신뢰 저장소 구성을 작성하십시오. 신뢰 저장소 구성에는 신뢰 저장소의 위치와 이름에 대한 특성이 포함됩니다.
3. 필요한 경우 **관리 > 보안 > SSL**에서 SSL 구성을 작성하십시오. 새로 작성된 신뢰 저장소 구성을 사용하도록 구성하십시오(새로 작성한 경우). 필요에 구성에 다른 조정 작업을 수행하십시오.
4. 아직 설정되지 않은 경우 **관리 > 보안**에서 **SSL 사용**을 예로 설정하십시오. 추가 필드가 나타납니다.
5. **아웃바운드 LDAP** 목록에서 작성한 SSL 구성을 선택하십시오. 다른 설정은 변경하지 마십시오.
6. **저장**을 클릭하십시오.
7. 마스터 **BFClient.conf** 업데이트를 클릭하십시오.
8. 이전에 SSL을 사용하지 않은 경우 다음을 수행하십시오.
  - a. **SSL 사용**은 아니오를 클릭하십시오.
  - b. **저장**을 클릭하십시오.
  - c. 마스터 **BFClient.conf** 업데이트를 클릭하십시오.
9. Build Forge를 다시 시작하십시오.
10. **관리 > LDAP**에서 LDAP 구성을 선택하십시오.
11. **연결 테스트**를 클릭하십시오.

## LDAP/Active Directory 지원 끄기

LDAP 또는 Active Directory에 대한 인증을 중단하려면 다음을 수행하십시오.

1. 관리 > LDAP를 선택하십시오.
2. 모든 도메인 항목을 삭제하십시오. 삭제할 도메인 항목 옆에 있는 휴지통 아이콘을 클릭하십시오.

도메인이 없는 경우에는 수동으로 추가된 사용자만 Build Forge이 로그인할 수 있습니다.

## 시스템 구성 설정

다양한 설정을 사용하여 관리 콘솔을 구성할 수 있습니다. 관리 > 시스템 페이지에서 이 설정을 찾을 수 있습니다.

관리 > 시스템을 클릭하면 시스템이 설정 목록을 표시합니다. 시스템 설정의 이름을 클릭하여 설정의 편집 패널을 표시하십시오.

**참고:** 숫자 값을 사용하는 시스템 설정의 경우, 관리 콘솔은 하나 이상의 정수(0 - 9)로 구성된 값을 승인합니다. 숫자 그룹화 문자(예: 쉼표(.), 10진수(.) 및 기타 정수가 아닌 구분 기호)는 지원되지 않습니다.

패널에는 다음 단추가 포함되어 있습니다.




- **저장:** 설정 값 변경사항을 저장합니다.
- **기본값으로 되돌리기:** 기본값으로 설정을 다시 설정합니다.

다음 테이블은 사용 가능한 설정에 대해 설명합니다.

설정	설명
계정 잠금 해제	기본값은 60분입니다. 이 시간이 지나면 시스템은 마지막 실패 로그인 시도를 버립니다. 음수 값으로 설정한 경우 실패한 로그인은 버려지지 않습니다.
계정 잠금 최대 시도 수	기본값: 3. 로그인 시도가 연속해서 실패하는 수이며 이 수를 초과하면 계정이 잠깁니다. 0인 경우 계정 잠금이 사용 불가능합니다.
계정 잠금 재설정 타이머	기본값은 120분입니다. 최소값은 5분입니다. 잠겨진 이후 사용자가 다시 로그인을 시도할 수 있을 때까지 필요한 시간의 양입니다. 음수로 설정한 경우 잠금은 만기되지 않습니다. 사용자가 다시 로그인할 수 있도록 하려면 관리자가 사용자를 재설정해야 합니다.
경보 이메일 한계	시스템이 지정된 시간(분 수) 동안 보내는 최대 경보 이메일 수를 설정합니다. 예를 들어, 10/60 값은 시간당 최대 10개의 메시지를 설정합니다. 기본값인 0/0은 시스템이 메시지에 제한이 없음으로 해석합니다.
인라인 단계 컨테이너 환경 적용	기본값: 아니오. 예인 경우, 인라인 단계를 포함하는 프로젝트 또는 라이브러리의 환경을 적용합니다.

설정	설명
서버 환경 최종 적용	기본값: 아니오. 예인 경우, 단계의 서버 환경을 마지막으로 적용합니다. 이 환경을 지정한 경우 단계 환경 또는 프로젝트 환경 다음에 서버 환경이 적용됩니다.
감사 최대 문자열 길이	기본값은 128자입니다. DETAILED 변경에 대해 레코딩할 수 있는 정보의 양을 제어합니다. 문자열 값이 이 문자 수를 초과하는 경우 스토리지 공간을 보존하기 위해 원래 값 대신 암호화 해시가 저장됩니다. 비밀번호와 같이 민감한 정보를 포함하는 필드의 경우 문자열 길이에 상관없이 암호화 해시가 사용됩니다.
감사 정책	<p>감사 로깅이 완료된 수를 판별합니다.</p> <p><b>없음</b>      레코딩된 감사 정보가 없습니다.</p> <p><b>보안</b>      보안 관련만 레코딩됩니다. (예: 액세스 그룹 작성 및 사용자 세션 활동)</p> <p><b>기본</b>      이는 기본 설정입니다. 이 설정은 올바르게 않은 감사 정책 값이 있는 경우에도 사용됩니다. 사용자, 오브젝트 유형, UUID 및 조치와 같은 최소 정보만 레코딩됩니다.</p> <p><b>세부사항</b>    이전 값과 새 값을 포함하여 변경된 특성에 대한 자세한 정보를 기록합니다. 모든 오브젝트가 이 정보를 제공할 수 있는 것은 아닙니다. 정보를 제공할 수 없는 경우 BRIEF 및 DETAILED 정책 간에 차이가 없습니다. 최대 문자열 길이 감사 구성 설정을 사용하여 저장되는 데이터 양을 제한할 수 있습니다.</p>
자동 로그오프 시간(분)	시스템은 대기 상태인 사용자를 자동으로 로그오프합니다. 이 설정은 시스템이 사용자를 로그오프하기 전에 경과해야 하는 대기 시간(분 단위)을 지정합니다. 설정이 0이면, 시스템이 사용자를 자동으로 로그오프하지 않습니다.
감사 로그 자동 정리일 수 오류 로그 자동 정리일 수 정보 로그 자동 정리일 수 경고 로그 자동 정리일 수	이 값은 항목의 각 범주가 감사 로그에 남아 있는 최대 일 수를 설정합니다. 이전 항목은 자동으로 삭제됩니다. 값이 0이면, 시스템은 해당 범주의 항목을 절대 삭제하지 않습니다. 문자열 값은 정수 0으로 평가되므로 0 대신 『Never』 같은 값을 사용할 수 있습니다.
빌드 취소 확인 빈도	시스템이 빌드 취소 요청을 확인하는 빈도(확인 간격)를 초 단위로 지정합니다.
콘솔 포트	웹 서버가 Build Forge 요청에 대해 청취하기 위해 사용하는 포트 번호입니다.
콘솔 URL	<p>웹 서버가 Build Forge 요청에 대해 청취하기 위해 사용하는 URL입니다.</p> <p>80 이외의 포트에서 콘솔이 실행 중인 경우 설정해야 합니다. 설정되면, 기본 콘솔 URL을 해당 값으로 대체합니다. &lt;protocol&gt;://&lt;hostname&gt;[:&lt;port&gt;] 양식을 사용합니다. (예: http://myHost:81)</p>

설정	설명
실패 시 단계 파트 계속	동일한 단계에서 다중 명령의 실행 동작을 판별하십시오. 기본값인 아니오로 설정되는 경우 단계의 실패한 명령으로 인해 단계가 실패할 수 있습니다. 예로 설정되면 명령이 실패해도 단계는 실패하지 않습니다. 단계에서 기타 명령이 실행됩니다.
누락된 경로 작성	기본값: 아니오. 예인 경우, 경로가 아직 없으면 프로젝트 경로를 작성합니다.
데이터베이스 크기 임계값	콘솔이 통지를 보낼 데이터베이스 크기의 임계값입니다. 기본값: 2G  임계값에 가까워지면 UI 성능이 느려집니다. 데이터베이스 정리를 수행하거나 임계값 크기를 늘리면 성능이 복원됩니다.
데이터베이스 크기 임계값 알림	데이터베이스 크기 임계값에 도달한 경우 이메일을 보낼 사용자 이름 또는 알림 그룹입니다. 사용자 이름이 사용되는 경우, 사용자의 이메일 필드에 있는 주소가 사용됩니다. 관리 > 사용자를 참조하십시오.
날짜 한 번	.date 명령 값이 작업당 한 번 또는 단계당 한 번 설정되는지 여부를 판별합니다. 동작은 다음과 같이 지정된 값에 따라 다릅니다. <ul style="list-style-type: none"> <li>기본값: 아니오. .date 명령 값이 단계당 한 번 설정됩니다. 각 단계는 에이전트 시스템의 시간대를 사용하여 .date 값을 다시 평가합니다. 에이전트는 이 값을 제공합니다. 다시 시작해도 .date 값을 유지합니다. 이는 초기 실행과 같이 다시 평가됩니다. 동일한 변수는 값이 사용되는 각 단계에 대해 다른 값을 가질 수 있는데, 이는 각 단계에 대해 .date 값이 재해석되기 때문입니다.</li> <li>예: 현재 .date 명령 값이 작업당 한 번 설정됩니다. 날짜 값은 에이전트의 시간대가 아닌 엔진 호스트의 시간대를 사용하여 생성됩니다. 다시 시작해도 .date 값을 유지합니다.</li> </ul> .email 명령이나 알림 템플릿과 같은 이메일 알림에 .date를 포함하는 환경 변수를 사용할 때 예를 사용하십시오. 그러면 이메일에 사용된 환경 변수가 엔진의 시간대를 사용합니다.
기본 에이전트 포트	에이전트에 연결하는 데 사용되는 기본 포트 번호를 설정합니다.
기본 가져오기 클래스	가져온 프로젝트에 클래스가 정의되어 있지 않거나 존재하지 않는 클래스가 정의된 경우 사용할 클래스입니다. 기본값은 프로젝트입니다.
XML 피드에 대한 인증 사용 안함	RSS 데이터 피드가 인증되는지 여부를 판별합니다.

설정	설명
빠른 시작 사용	<p>기본값은 아니오입니다. 모든 프로젝트는  아이콘을 표시합니다. 프로젝트를 클릭하면 프로젝트에 포함된 모든 변수가 변경 필수 유형의 변수로 선택됩니다. 프로젝트에 변경 필수 변수가 포함되지 않은 경우에 프로젝트가 시작됩니다. 프로젝트에 변경 필수 변수가 포함된 경우, 프로젝트는 시작되지 않으며 이유를 설명하는 대화 상자가 표시되고 아이콘이  으로 변경됩니다.</p> <p>참고: 프로젝트에 대해 정의된 변수만 선택됩니다. 프로젝트의 단계에 정의된 변수는 선택되지 않습니다. 323 페이지의 『작업 시작 시 프로젝트 변수 변경 내용』의 내용을 참조하십시오.</p> <p>예로 설정되면 프로젝트 페이지는 페이지의 모든 프로젝트에 대한 모든 환경을 확인하고 변경 필수 유형의 변수가 있는지 판별합니다. 즉시 시작할 수 있는 프로젝트에는  아이콘이 표시됩니다. 이는 버전 7.1.1.1까지 기본 동작이었습니다.</p>
강제 실행 한계	<p>기본값: 아니오. 예이면 실행이 프로젝트의 실행 한계 설정을 위반하지 않는 경우 시스템은 프로젝트의 스케줄된 빌드를 실행합니다. 아니오이면 시스템은 스케줄된 빌드에 대해 프로젝트의 실행 한계 설정을 무시합니다.</p>
기본 보안 액세스 그룹 가져오기	<p>보안 액세스로 가져오기 설정이 Y로 설정된 경우 가져온 프로젝트의 기본 액세스 그룹을 지정합니다.</p>
비보안 기본 액세스 그룹 가져오기	<p>보안 액세스로 가져오기 설정이 N으로 설정된 경우 가져온 프로젝트의 기본 액세스 그룹을 지정합니다. 기본 그룹은 가져온 파일에 액세스 그룹이 없는 경우에만 사용됩니다.</p>
보안 액세스로 가져오기	<p>Y로 설정된 경우, 시스템은 가져온 데이터 오브젝트에 이전 설정에 나열된 기본 액세스 그룹을 지정합니다. 이는 가져오고 있는 XML 파일에 지정된 액세스 그룹을 대체하므로 사용자는 데이터를 가져와 보안을 대체할 수 없습니다. N으로 설정된 경우, 시스템은 가져온 파일의 액세스 그룹 설정을 사용합니다.</p>
태그 상속	<p>예로 설정되면, 체인을 통해 실행된 작업이 호출자와 동일한 작업 태그를 사용하게 됩니다. MasterProject 프로젝트의 BUILD_15가 ComponentProject 프로젝트를 호출하는 경우, 해당 ComponentProject 실행에 대한 작업 태그(및 작업 디렉토리 이름)는 BUILD_15가 됩니다.</p> <p>참고: 호출된 프로젝트는 항상 호출자의 원래 태그를 상속합니다. 예를 들어 .retag 명령의 결과로 호출자의 태그가 실행 중에 변경되면 호출된 프로젝트는 호출자가 시작한 태그를 여전히 가져옵니다.</p>
올바르지 않은 상대 디렉토리 문자	<p>프로젝트 이름에서 사용되는 경우 시스템이 밑줄로 변경할 문자를 설정합니다.</p>
LASTRUN 형식	<p>.date 명령에 정의된 날짜 형식 문자를 사용하여 BF_LASTRUN 환경 변수 형식의 값을 입력하십시오(425 페이지의 『.date』 참조).</p>

설정	설명
라이선스 서버	라이선스 서버 호스트 이름입니다. 이 이름은 설치 중에 설정됩니다. 예제: myhost.mycompany.com. 이 값에는 포트 이름이 포함될 수 있습니다. 예제: myhost.mycompany.com:80. 라이선스 서버를 변경하려면 24 페이지의 『관리 콘솔의 라이선스 서버 변경』을 참조하십시오.
링크 디버그 모드	예로 설정되면 어댑터 링크가 정의된 작업은 연관된 프로젝트를 실행하는 대신 링크 테스트를 실행합니다. 작업 출력에는 어댑터의 출력이 포함된 단일 단계가 있습니다. 데이터는 어댑터 인터페이스의 문제점 해결 시 유용합니다. 참고: 어댑터 링크 상태를 디버그로 설정하여 개별적인 어댑터 링크에 대한 디버깅을 설정할 수 있습니다. 상태는 링크 디버그 모드 설정보다 우선합니다.
수동 작업 링크	프로젝트를 스케줄러에서 실행하지 않고 수동으로 빠른 시작하는 경우 시스템이 어댑터 링크를 통해 어댑터를 실행하는지 여부를 판별합니다.  링크는 작업 BOM에 추가 출력을 생성할 수 있습니다.  N으로 설정된 경우에는 링크가 확인되지 않거나 작업이 실행될 때 실행됩니다.
최대 콘솔 프로세스 수	콘솔이 동시에 실행하는 최대 프로세스 수를 설정합니다. 콘솔 활동에 대한 일반 스로틀로 사용하십시오. 시스템은 데이터베이스에 각 프로세스의 ID를 저장하고 새 외부 프로세스를 실행하기 전에 총계를 확인하여 프로세스를 관리합니다. 이 값은 최소 5 이상 실행 큐 크기 설정보다 커야 합니다. 그렇지 않으면, 시스템이 실행 큐를 지원하기에 충분한 프로세스를 실행할 수 없습니다.
최대 인라인 깊이	프로젝트가 무한정 중첩될 수 없도록 시스템이 프로젝트 인라인화에 허용하는 레벨 수를 제어합니다. 기본값은 32입니다. 값이 0으로 설정되면, 시스템은 32를 사용합니다. 시스템이 인라인 한계에 도달하면 한계를 초과하는 인라인 프로젝트가 실행되지 않으며, 단계가 포함 프로젝트에 삽입되지 않습니다. "인라인을 포기했습니다"라는 메시지가 시스템 메시지 목록에 기록됩니다.
최대 동시 서버 테스트 수	동시에 실행할 수 있는 서버 테스트 수를 지정합니다. 시스템 자원에 따라, 동시에 서버 테스트를 너무 많이 실행하면 속도가 몹시 느려지거나 콘솔이 잠길 수 있습니다.
최대 동시 제거 수	동시에 실행할 수 있는 제거 수를 제어합니다. 원하는 만큼 빌드를 제거할 수 있지만, 최대 동시 제거 수의 값보다 많은 빌드는 동시에 삭제되지 않습니다. 기본값: 20
최대 새로 고치기 수	페이지를 자동으로 새로 고치는 최대 횟수입니다. 기본값은 50입니다.
체인화 시 클래스 대체	시스템이 체인 프로젝트의 클래스를 호출자의 클래스로 바꾸는지 여부를 판별합니다. 기본값이 Y이면 시스템이 체인 프로젝트의 클래스를 대체하고 호출자의 클래스를 대신 사용합니다.

설정	설명
비밀번호 만기일 수	비밀번호 만기가 설정된 사용자가 비밀번호를 변경해야 하기 전의 일 수를 설정합니다. 이 시간이 만기되면 관련 사용자는 다음 로그인 시에 비밀번호를 변경해야 합니다.



설정	설명
비밀번호 형식	<p>마침표로 분리된 6개 필드의 형식 문자열을 사용하여 사용자 비밀번호에 대한 요구사항을 지정합니다.</p> <p><i>length.char_types.upper.lower.numeric.special</i></p> <p>예를 들어, 5.2.u1.l1.n1.s1을 지정할 수 있습니다. 이 예제는 아래에서 설명됩니다.</p> <p>처음 두 필드는 다음을 지정합니다.</p> <ul style="list-style-type: none"> <li>• 최소 비밀번호 길이(문자 수)</li> <li>• 나머지 네 개의 필드에서 사용할 최소 문자 유형 수(1에서 4까지의 정수)</li> </ul> <p>나머지 필드는 문자 유형 및 빈도 요구사항을 지정합니다. 각 필드는 유형과 숫자를 포함합니다.</p> <ul style="list-style-type: none"> <li>• 유형: u(대문자), l(소문자), n(숫자) 또는 s(특수) 중 하나. 대문자(U, L, N, S)는 해당 문자가 필수임을 표시합니다. 소문자(u, l, n, s)는 문자가 선택적임을 표시합니다.</li> <li>• 숫자: 필수인 경우, 이 유형의 필수 문자 수를 표시합니다. 선택적인 경우, 임의의 문자가 사용되는 경우 이 유형의 필수 문자 수를 표시합니다.</li> </ul> <p>유형은 다음과 같습니다.</p> <ul style="list-style-type: none"> <li>• U 또는 u는 대문자를 표시합니다. 여기에는 각 로케일에서 문자로 간주되는 모든 문자 중 소문자가 아닌 문자가 포함됩니다. 여기에는 특히 대문자, 제목 문자 또는 대소문자 구분이 없는 언어(예: 중국어)의 문자가 포함됩니다.</li> <li>• L 또는 l는 소문자를 표시합니다. 여기에는 각 로케일에서 문자로 간주되는 모든 문자가 포함됩니다.</li> <li>• N 또는 n은 숫자 문자를 표시합니다. 여기에는 각 로케일에서 숫자로 간주되는 모든 문자가 포함됩니다.</li> <li>• S 또는 s는 특수 문자를 표시합니다. 위의 세 카테고리에 맞지 않는 모든 문자입니다. 여기에는 문자도 아니고 숫자도 아닌 모든 문자가 포함됩니다.</li> </ul> <p>예제: 5.2.u1.l1.n1.s1 문자열은 다음 비밀번호 요구사항을 표시합니다.</p> <ul style="list-style-type: none"> <li>• 최소 5자여야 함</li> <li>• 네 개 범주(대문자, 소문자, 숫자, 특수) 중 최소 두 개 범주의 문자를 포함해야 함.</li> <li>• 각 유형에 대해 유형의 한 문자는 요구사항을 계수하기 위한 일치로 규정됨.</li> </ul> <p>abC1x 및 Abc2% 같은 비밀번호는 요구사항을 만족합니다.</p>

설정	설명
Build Forge 엔진 일시정지	Y로 설정되면, 시스템이 현재 작업을 완료한 후 엔진을 일시정지합니다. 정상 오퍼레이션으로 리턴하려면 N으로 설정하십시오.
공용 호스트 이름	설정되면, 시스템이 알림 템플릿에 있는 CONSOLEHOST 변수의 서버 호스트 이름을 이 설정 값으로 대체합니다.
제거 확인 시간	시스템이 제거할 작업을 확인하는 빈도(확인 간격)를 분 단위로 설정합니다.
QuickReport 공용 디렉토리	공용 보고서 디자인의 파일 시스템 위치입니다.  7.1에서는 이 시스템 설정을 사용하여 공용 보고서에 대한 완전한 위치를 지정합니다. 보고서 디자인을 자동으로 마이그레이션하려면 해당 보고서 디자인이 이 디렉토리에 있어야 합니다.  이전 릴리스에서는 기본 파일 위치(../reports/public)가 애플리케이션 서버 설치 디렉토리(예: <bf-install>/Apache/tomcat/webapps/quickReport)와 관련되어 있습니다.
QuickReport 임시 디렉토리	7.1에서는 이 디렉토리를 사용하여 서비스 계층 컴포넌트와 동일한 호스트에 완전한 디렉토리를 지정합니다. 서비스 계층은 이 작업 디렉토리를 사용하여 데이터베이스에 성공적으로 마이그레이션된 보고서 디자인을 나열합니다.  이전 릴리스에서는 파일 시스템의 공용 또는 개인용 디렉토리에 저장되기 전에 Quick Report 보고서 디자인을 임시로 저장하기 위해 이 디렉토리가 사용되었습니다.
QuickReport 사용자 디렉토리	개인용 보고서 디자인의 파일 시스템 위치입니다.  7.1에서는 이 시스템 설정을 사용하여 개인용 보고서에 대한 완전한 위치를 지정합니다. 자동으로 마이그레이션하려면 보고서 디자인이 이 디렉토리에 있어야 합니다.  이전 릴리스에서 개인용 보고서에 대한 기본 파일 위치(../reports/users)는 애플리케이션 서버 설치 디렉토리에 상대적입니다(예: <bfinstall>/Apache/tomcat/webapps/quickReport).
언어 팩 다시 로드	기본값: 아니오. 예로 설정되면, 다시 시작할 때 콘솔이 언어 팩을 다시 로드하고 이 값을 아니오로 다시 설정합니다. 더 이상 버전 7.0.1에서 시작할 필요가 없습니다.
어댑터 템플릿 재설정	업데이트로 인한 변경사항을 구성에 복사하기 위해 어댑터 템플릿을 다시 설정하려면 이 설정을 사용하십시오. 이 설정을 사용하면, 값을 『예』로 설정한 후 1분간 기다리십시오. 시스템이 템플릿을 다시 설정한 후 값을 다시 아니오로 설정합니다.
서버 작업 계수 다시 설정	모든 서버의 작업 수(BF_JOBS)를 동시에 0으로 다시 설정하려면 이 설정(예)을 사용하십시오. Manifest 검사 간격이 실행될 때 다시 설정됩니다. (기본 시간은 매 10초입니다.)  모든 서버에 대해 BF_JOBS가 다시 설정되면 서버 작업 계수 다시 설정 값은 아니오(기본값)로 되돌려집니다.
보고서 마이그레이션 다시 시작	기본값: 아니오. 7.1에서 서비스 계층 컴포넌트를 다시 시작하지 않고 마이그레이션을 시작하려면 이 값을 예로 설정하십시오.
체인 링크 실행	실행된 체인 프로젝트가 접속된 어댑터 링크도 실행하는지 여부를 제어합니다.

설정	설명
실행 큐 크기	이 값은 시스템이 동시에 실행하려고 시도하는 작업 수를 제한합니다. 큐의 실행 수가 이 값과 같거나 이 값을 초과하면, 이 값 아래로 작업 수가 줄어들 때까지 시스템이 큐에서 실행 큐로 실행 이동을 중지합니다. 실행 큐 크기를 변경하는 경우, 최대 콘솔 프로세스 수 설정(실행 큐 크기보다 최소 5는 커야 함)을 확인하십시오.
시작 환경 저장	프로젝트의 수동 시작 페이지에 있는 『환경 저장』 기본값 선택란을 제어합니다. 이 설정이 Y이면, 상자가 기본적으로 선택되고 그렇지 않으면 상자가 기본적으로 선택 취소됩니다. 『환경 저장』 선택란이 선택되면, 시작 페이지에서 변경한 환경 변수가 데이터베이스의 환경 레코드에 저장되어 향후 실행은 기본적으로 해당 값으로 설정됩니다.
체인 전의 서버 환경	시스템이 단계에서 체인 프로젝트의 프로젝트 환경 설정 이전(Y) 또는 이후(N) 중 언제 단계의 서버 환경을 설정하는지 여부를 결정합니다. 처리된 두 번째 환경의 변수가 첫 번째 환경의 변수보다 우선합니다. 기본값은 Y이며, 이는 체인 프로젝트의 환경이 두 번째로 처리되고 단계의 서버 환경을 대체함을 표시합니다.
서버 재시도 수	모든 선택기 조건과 일치하는 서버를 발견할 때 시스템이 단계를 종료하고 실패로 처리하기 전에 서버에 단계를 할당하려고 시도하는 횟수를 설정합니다.

설정	설명
서버 테스트 빈도	<p>서버에 대한 Manifest 데이터를 테스트하고 새로 고치는 빈도를 판별하기 위해 사용 가능한 서버 수와 함께 사용됩니다. 기본값은 120분(2시간)입니다. 값 0은 서버를 검사하지 않음을 의미합니다.</p> <p>이 확인 동안, 시스템은 사용 가능한 모든 서버에 접속하여 다음을 확인합니다.</p> <ul style="list-style-type: none"> <li>• 서버가 아직 접속 가능함</li> <li>• 서버에 대한 로그인 정보가 올바른</li> <li>• 서버의 Manifest 데이터가 최신 상태임</li> </ul> <p>서버 테스트는 분당 최소 하나의 서버에서 수행됩니다. 이 비율은 서버 수가 이 설정보다 많을 경우에 증가합니다.</p> <p>시스템이 간격에 대해 균등하게 테스트를 분배합니다. 예제:</p> <ul style="list-style-type: none"> <li>• 120개의 서버를 가지고 있고 간격은 120분으로 설정됩니다. 시스템은 분당 하나의 서버를 테스트하려고 합니다(120개 서버 / 120분).</li> <li>• 12,000개의 서버를 가지고 있고 간격은 1200분으로 설정됩니다. 시스템은 분당 10개의 서버를 테스트하려고 합니다.</li> <li>• 10개의 서버를 가지고 있고 간격은 120분으로 설정됩니다. 분당 하나의 서버가 확인됩니다. 서버는 이 시나리오에서 새로 고치기 간격 동안 12배 확인됩니다.</li> </ul> <p>콘솔에서 시작된 수동 서버 테스트는 이 자동화된 테스트보다 우선합니다.</p> <p>서버 콜렉터의 복잡도는 처리량에 영향을 줄 수 있습니다. 데이터를 수집하기 위해 많은 수동 명령을 수행하는 콜렉터는 완료하는데 1분보다 많은 시간이 필요할 수 있습니다.</p>
서버 사용 연결 제한시간	<p>관리 콘솔이 연결이 열리기를 기다리는 에이전트 연결을 작성한 후 단계 실패 이전의 시간(초)을 설정합니다.</p> <p>에이전트에 대한 기존 연결의 제한시간 값을 설정합니다. 에이전트는 15초마다 관리 콘솔에 접속하도록 설계되었습니다. 제한시간 기간 중에 접속되지 않으면, 에이전트가 중지되거나 네트워크 통신 문제가 발생할 수 있습니다. 이 값을 초과하면 단계가 실패합니다.</p>
서비스 계층 인증 서블릿 URL	<p>설정되는 경우, 서비스 계층 인증 서블릿에 대해 프로그램으로 구축된 URL을 겹쳐줍니다.</p> <p>별명 또는 비기본 포트를 사용 중인 경우, 다음 형식 (<a href="http://server:port/rbf-services/AuthServlet">http://server:port/rbf-services/AuthServlet</a>)을 사용하여 이 설정을 업데이트해야 합니다.</p>
SMTP 서버	<p>이메일 알림을 보낼 때 SMTP 서버로 사용할 컴퓨터를 설정합니다.</p> <p>기본 설정은 localhost입니다.</p>

설정	설명
SSO 원격 사용자	기본값: 아니오. 예인 경우, 표준 웹 서버 인증을 통한 단일 사인은 원격 사용자 연결이 허용됩니다.
BuildForge 환경 변수 스택	시스템은 일반적으로 체인 프로젝트로 전달되는 BF_ 변수의 이름을 BF_CALLER_로 변경합니다. 이 설정은 두 레벨 이상 체인화할 때 시스템이 이름 지정을 스택하는지 여부를 판별합니다. 기본값은 N입니다. 설정이 Y로 변경되면, 두 레벨 간의 호출 프로젝트에서 파생된 BF_TAG 변수는 BF_CALLER_CALLER_TAG 이름을 수신합니다.
단계당 최대 재시도	단계에서 첫 번째 시도가 실패하는 경우 에이전트에 연결하기 위해 시도하는 횟수를 제어합니다. 지정된 재시도 횟수 안에 연결하지 못하는 경우 단계는 실패합니다.
사용자 인증을 로컬로 저장	시스템이 LDAP/Active Directory 사용자 인증 정보를 암호화된 양식으로 캐시하는지 여부를 판별합니다. 기본값은 예입니다. LDAP/Active Directory 인증을 사용하는 경우에만 시스템이 적절합니다. 설정이 예인 경우, 시스템은 암호화된 양식의 사용자 인증 정보를 캐시하여 이를 _USE_BFCREDS 및 _USE_BFCREDS_DOMAIN 특수 변수(서버에 대한 사용자 인증에 적용되는)와 함께 사용합니다. 값을 아니오로 변경하여 캐싱을 끌 수 있습니다. 그러나 이 경우 시스템은 서버에 로그인할 때 _USE_BFCREDS 및 _USE_BFCREDS_DOMAIN 특수 변수를 사용하여 사용자의 신임 정보를 사용할 수 없습니다.
시스템 정보 이메일	<p>시스템은 이 설정으로 정의된 주소로 정보 이메일 메시지를 보냅니다.</p> <p>기본값은 root@localhost입니다. 다음 이메일 주소 형식이 지원됩니다. Username은 이메일 사용자 이름입니다. Name은 임의의 문자열로, 대개 일부 양식에 지정된 이름입니다.</p> <pre>username@host.com name username@host.com name "&lt;username@host.com&gt;" name &lt;username@host.com&gt; name "username@host.com"</pre> <p>여러 주소로 발송하려면 쉼표(,) 또는 세미콜론(;)으로 주소를 구분하십시오.</p>
시스템 정보 소스	<p>시스템은 정보 이메일 메시지를 보낼 때 이 설정에 정의된 주소로 보내는 사람으로 사용합니다.</p> <p>기본값은 root@localhost입니다. 다음 이메일 주소 형식이 지원됩니다. Username은 이메일 사용자 이름입니다. Name은 임의의 문자열로, 대개 일부 양식에 지정된 이름입니다.</p> <pre>username@host.com name username@host.com name "&lt;username@host.com&gt;" name &lt;username@host.com&gt; name "username@host.com"</pre>
시스템 전반 로그인 메시지	로그인 양식 위에 표시될 메시지를 정의할 수 있게 합니다.
시스템 전반 사용자 메시지	각 페이지 맨 위에(탐색 단추 바로 아래에) 표시될 메시지를 정의할 수 있게 합니다.

설정	설명
태그: 날짜 형식	BF_D 태그 변수에 날짜를 표시하는 데 사용되는 형식을 정의합니다. 문자 y, m 및 d를 각각 년, 월 일에 대한 변수로 사용하고 또한 원하는 특정 문자를 분리 문자로 사용하여 원하는 형식을 표시합니다. 예를 들어 2005년 9월 21일은 다음과 같이 표시됩니다.  형식 문자열....출력  ymd...050921  m/d/y...09/21/05
태그: 시간 형식	BF_T 태그 변수에 날짜를 표시하는 데 사용되는 형식을 정의합니다. 태그: 날짜 형식 설정과 마찬가지로 작동하지만 h, m 및 s 문자를 사용하여 시, 분 및 초를 나타냅니다. h:m:s 설정은 12:53:42와 같은 출력을 생성합니다.
메일 템플릿의 로그 행 수	알림 템플릿에서 TAILNORMALLOG 변수가 사용될 때 알림에 표시되는 로그 끝에서부터의 행 수를 설정합니다.
스레드 종료	스레드된 단계가 실패하는 경우 동일한 프로젝트에 있는 다른 모든 활성 스레드 블록이 중지되는지 여부를 판별합니다.

## 기타 시스템 설정

시스템 외부에서 설정되는 구성 설정이 있습니다.

### I/O 제한시간

I/O 제한시간은 기본적으로 300초입니다. 운영 체제 변수로 BF\_SOCKET\_SELECT\_TIMEOUT을 작성하고 여기에 값(초)을 지정하여 해당 값을 조정할 수 있습니다. 이 값은 60초 미만으로 설정해서는 안 됩니다. 처음 시작할 때 이 값은 읽기 전용입니다. 이 값을 변경하려면 시스템을 중지한 후 다시 시작해야 합니다. 값이 너무 낮으면 대형 병렬 로드(예: 여러 작업을 한 번에 시작) 시에 시스템이 정지한 것처럼 보일 수 있습니다. 기본값은 모든 상황에 적합합니다.

## 감사 로깅

감사 로깅은 감사 로깅을 관리하고 감사 로그를 보도록 명령행 인터페이스를 제공합니다. 감사 로그는 시스템의 오브젝트에 대한 변경사항의 히스토리를 제공합니다. 이는 전체 텍스트 색인에 저장됩니다.

추적하는 오브젝트에 대해 감사 로깅이 사용되면 해당 로깅은 관리 > 메시지에서 볼 수 있는 메시징을 대체합니다.

**중요사항:** 감사 로깅은 실행 중인 시스템에서 직접 수행되는 조작에 대해서만 작업합니다. 이는 시스템으로 가져오는 데이터에 대해서 작업하지 않습니다. `bfimport` 기능은 데이터베이스를 직접 업데이트하며 감사 로깅 기능과 상호작용하지 않습니다.

## 감사 정책 관리

감사 정책은 감사가 완료되는 수를 제어합니다. 이는 감사 정책 시스템 구성 설정의 설정으로 판별됩니다.

**없음** 레코드된 감사 정보가 없습니다.

**보안** 보안 관련만 레코드됩니다. (예: 액세스 그룹 작성 및 사용자 세션 활동)

**기본** 이는 기본 설정입니다. 이 설정은 올바르게 않은 감사 정책 값이 있는 경우에도 사용됩니다. 사용자, 오브젝트 유형, `uuid` 및 조치와 같은 최소 정보만 레코드됩니다.

### 세부사항

이전 값과 새 값을 포함하여 변경된 특성에 대한 자세한 정보를 기록합니다. 모든 오브젝트가 이 정보를 제공할 수 있는 것은 아닙니다. 정보를 제공할 수 없는 경우 **BRIEF** 및 **DETAILED** 정책 간에 차이가 없습니다. 최대 문자열 길이 감사 구성 설정을 사용하여 저장되는 데이터 양을 제한할 수 있습니다.

최대 문자열 길이 감사 시스템 구성 설정은 문자열 데이터 필드의 길이를 제한하여 **DETAILED** 변경에 대해 레코드할 수 있는 정보의 양을 제어합니다. 감사 항목이 이 문자 수를 초과하는 경우 문자열 값 대신 **SHA** 메시지 요약이 저장됩니다. 비밀번호와 같이 민감한 정보를 포함하는 필드의 경우 문자열 길이에 상관없이 메시지 요약이 사용됩니다.

## 감사 정보 표시

명령행 인터페이스로 표시할 정보와 이를 형식화할 방법을 지정할 수 있습니다.

모든 감사 로그 보기 권한이 필요합니다. 이는 기본적으로 보안 및 빌드 엔지니어 액세스 그룹에 지정됩니다.

사용법: `bfauditlog [connection-options][command-options][query]`

리터럴 실행 파일은 다음과 같습니다.

- Windows 시스템: `bfauditlog.bat`
- UNIX 및 Linux 시스템: `bfauditlog.sh`

### 연결 옵션

`-H hostname` | `--hostname=hostname`

연결에 대한 원격 호스트 이름입니다. 기본값은 `localhost`입니다.

**-P port | --port=port**

연결에 대한 통신 포트입니다. 기본값은 3966입니다.

**-d domain | --domain=domain**

인증에 대한 도메인 또는 영역입니다. 기본값은 없음입니다.

**-u login | --user=login**

인증에 사용할 사용자 이름입니다. 기본값은 root입니다.

**-p password | --password=password**

인증 사용자의 비밀번호입니다. 기본값은 root입니다.

**-E encoding | --encoding=encoding**

사용할 출력 문자 세트의 인코딩입니다. 올바르지 않은 인코딩을 지정하면 사용 가능한 모든 인코딩이 나열되고 프로그램이 종료됩니다. 요청된 인코딩에 텍스트를 표시하는 기능은 클라이언트에 사용된 터미널 소프트웨어의 기능에 따라 다릅니다. 기본값은 운영 체제와 지원되는 JVM에 따라 결정됩니다.

## 명령 옵션

**-h**

도움말을 표시합니다.

**-b | --bare**

출력에 정보를 단순하게 표시합니다(기본). 정확한 영향은 출력 형식에 따라 다릅니다(-t 참조). 예를 들어, XML 출력에서 열기 태그, 닫기 태그 및 행 개수는 생략됩니다. CSV에서 행 개수 및 열 이름은 생략됩니다.

**-c auditClass[,auditClass ...] | --audit.class=class-list**

조회에서 사용할 감사 클래스의 목록을 지정합니다. 다중 클래스 이름은 쉼표로 분리하십시오. 여러 번 지정하면 지정된 모든 감사 클래스의 유니온이 사용됩니다. 옵션을 지정하지 않으면 사용자가 지정할 수 있는 모든 감사 클래스가 사용됩니다. -L 옵션을 사용하여 사용 가능한 감사 클래스를 나열하십시오. 기본값은 모든 감사 클래스로, 사용자가 볼 수 있도록 권한이 부여됩니다.

**-f field[,field ...] | --field=field-list**

출력에 포함할 필드를 지정합니다. 다중 필드 이름은 쉼표로 분리하십시오. -L 옵션을 사용하여 사용 가능한 필드를 나열하십시오.

**-s field[+|-][,field[+|-] ...] | --field=field-list**

출력 정렬을 위해 사용할 필드를 나열합니다. 기본적으로 출력은 audit.stamp의 순서로 내림차순으로 정렬됩니다. + 마커는 오름차순을 지정합니다. - 마커는 내림차순을 지정합니다. 다중 필드 이름은 쉼표로 분리하십시오. 기본값은 -audit.stamp입니다.

**-l [offset,]count | --limit=[offset,]count**

결과 페이지의 번호를 매기는 데 사용됩니다. 오프셋 및 개수는 항목 수로 표시됨



니다. 예를 들어, "-l 100,50"은 100번째 항목 다음에 50개의 항목을 표시함을 의미합니다. 비어 있지 않은 출력 형식에는 출력의 일치하는 총 항목 수가 포함됩니다.

**-a** *user[,user ...]* | **--audit.user**=*user-list*

출력은 지정된 사용자와 연관된 항목으로 제한합니다. 다중 사용자 이름은 쉼표로 분리하십시오. 여러 개의 -a 옵션을 사용하면 명령에서 옵션에 지정된 모든 사용자의 유니온을 사용합니다. 사용자는 로그인 ID 또는 UUID로 지정할 수 있습니다. 사용자 인수 없이 --audit-user를 지정하면 인증에 사용되는 사용자가 사용됩니다.

**-t** *format* | **--format**=*format*

출력에 사용할 형식은 CSV, XML, JSON 또는 TEXT 중 하나입니다. 기본값은 TEXT입니다.

**-L** | **--list**

사용자에 대해 사용 가능한 감사 클래스를 나열합니다. 별표(\*)로 표시된 감사 클래스는 사용자가 액세스할 수 있습니다. -c(감사 클래스 목록)와 함께 사용하면 각 클래스에 사용 가능한 필드 목록이 대신 표시됩니다. 이 옵션을 지정해도 다른 명령 옵션에는 영향을 주지 않습니다. ViewAllAudits 권한이 있는 사용자는 모든 클래스의 감사 로그를 볼 수 있습니다. 다른 사용자의 경우 -a 옵션을 사용하여 자신의 감사 로그만 볼 수 있습니다.

## 조회

조회는 웹 검색 엔진이 일반적으로 사용하는 형식을 사용하는 문자열입니다: 공백으로 분리된 스펙 목록. 스펙은 '필드:값' 양식이며 지정된 필드에서 값을 검색합니다. 조회를 지정하지 않았지만 최소 하나 이상의 옵션을 지정하면 일치하는 모든 감사 로그가 표시됩니다. 조회에는 감사 사용자와 감사 클래스를 지정할 수 없습니다. 감사 사용자와 감사 클래스를 지정하려면 명령 옵션을 사용하십시오.

**참고:** Windows 시스템에서는 큰따옴표를 사용하고 UNIX 및 Linux 시스템에서는 작은따옴표를 사용하십시오.

## 예제

1. 이 예제는 프로젝트 및 단계 감사 클래스의 모든 알려진 조회 필드를 나열합니다.

- Windows

```
$ bfauditlog.bat -c Project,Step -L
```

- UNIX 및 Linux

```
$ bfauditlog.sh -c Project,Step -L
```

2. 이 예제는 쉽표로 구분된 변수 출력 형식을 사용하여 컬럼 표제나 총 행 개수 없이 프로젝트 작성에 사용된 최근 10개의 감사 로그를 보여 줍니다. 각 레코드에는 프로젝트의 고유 ID와 이름만 지정됩니다.

- Windows

```
$ bfauditlog.bat -c Project -t CSV -f audit.target,
  sxProjectName --bare -l 10 "audit.type:Create"
```

- UNIX 및 Linux

```
$ bfauditlog.sh -c Project -t CSV -f audit.target,
  sxProjectName --bare -l 10 'audit.type:Create'
```

3. 이 예제는 제공된 도메인, 사용자 이름 및 비밀번호로 로그인한 후에 'buildforge.example.com' 시스템에 설치 작업을 수행할 때 사용된 최근 100개의 감사 로그를 검색합니다.

- Windows

```
$ bfauditlog.bat -H buildforge.example.com
  -d ExampleLDAP -u apiuser -p "My$ecret" -l 100
```

- UNIX 및 Linux

```
$ bfauditlog.sh -H buildforge.example.com
  -d ExampleLDAP -u apiuser -p 'My$ecret' -l 100
```

4. 이 예제는 연결을 인증한 사용자의 시간대 설정에 따라 2011년 5월 23일부터 2011년 5월 27일까지 기록된 감사 로그를 검색합니다.

- Windows

```
$ bfauditlog.bat "audit.stamp:[2011/05/23 TO 2011/05/27]"
```

- UNIX 및 Linux

```
$ bfauditlog.sh 'audit.stamp:[2011/05/23 TO 2011/05/27]'
```

5. 이 예제는 해당 액세스 그룹 레벨이 이전 값인 4, 5, 6에서 새 값인 3으로 변경된 프로젝트의 감사 로그를 검색합니다. 대괄호는 포함 범위를 나타내는 데 사용되므로 '[4 TO 6]'은 값 4와 6이 일치함을 의미합니다. 또한 감사 정책이 'DETAILED'로 설정된 경우에는 이전 레벨 값과 새 레벨 값만 검색할 수 있습니다.

- Windows

```
$ bfauditlog.bat -c Project "ioLevel:[4 TO 6] AND inLevel:3"
```

- UNIX 및 Linux

```
$ bfauditlog.sh -c Project 'ioLevel:[4 TO 6] AND inLevel:3'
```

6. 이 예제는 해당 액세스 그룹 레벨이 이전 값인 4 또는 5에서 새 값인 2 또는 3으로 변경된 프로젝트의 감사 로그를 검색합니다. 중괄호는 제외 범위를 나타내는 데 사용되므로 '{3 TO 6}'은 값 3과 6이 일치하지 않음을 의미합니다. 또한 감사 정책이 'DETAILED'로 설정된 경우에는 이전 레벨 값과 새 레벨 값만 검색할 수 있습니다.

- Windows

```
$ bfauditlog.bat -c Project "ioLevel:{3 TO 6} AND inLevel:(2 OR 3)"
```

- UNIX 및 Linux

```
$ bfauditlog.sh -c Project 'ioLevel:{3 TO 6} AND inLevel:(2 OR 3)'
```

7. 이 예제는 사용자 root 및 sv 감사에 대한 결과 조회를 요구하며 CSV 파일에서 출력을 생성합니다. 이는 프로젝트 및 단계 클래스에 대한 데이터를 리턴하며 클래스, 유형, 대상, 사용자 및 시간소인에 대한 정보를 포함합니다.

- Windows

```
$ bfauditlog.bat -c Project,Step -f audit.class,audit.type,audit.target,audit.user,audit.stamp  
-a root,sv -l 10,5 -t CSV
```

- UNIX 및 Linux

```
$ bfauditlog.sh -c Project,Step -f audit.class,audit.type,audit.target,audit.user,audit.stamp  
-a root,sv -l 10,5 -t CSV
```

## 클래스

클래스는 감사 로그 항목에 대한 레이블입니다. -L 옵션을 사용하여 사용 가능한 클래스를 나열하십시오.

## 필드

필드는 각 감사 로그 레코드에 지정된 정보를 나타냅니다.

- audit.class: 프로젝트(문자열)
- audit.stamp: 시간소인(정수)
- audit.stamp.localized: audit.stamp 정수 값을 나타내는 현지화된 문자열
- audit.type: 업데이트(문자열)
- audit.target: 프로젝트의 UUID(문자열)
- audit.user: 업데이트를 수행한 사용자의 UUID

감사 로그 레코드의 일부 필드는 해당 상태(이전의 경우 o, 새로 작성된 경우 n, 확장된 경우 x) 및 해당 데이터 유형(문자열의 경우 s, 정수의 경우 i)을 반영합니다. 예를 들어 다음과 같습니다.

```
soDescription: 이전 프로젝트 설명(문자열)  
snDescription: 새 프로젝트 설명(문자열)  
soSelectorUuid: 이전 선택기의 UUID(문자열)  
snSelectorUuid: 새 선택기의 UUID(문자열)  
ioRunLimit: 이전 실행 한계(정수)  
inRunLimit: 새 실행 한계(정수)
```

## 메시지

시스템 메시지 페이지는 사용자가 검토할 수 있는 사용자 조치 및 시스템 로그를 표시합니다.

관리 > 메시지를 선택하여 메시지 목록을 표시하십시오. 이 메시지는 조치에 대한 감사 추적을 구성합니다.

IBM UI 구성 콘솔 보고서 [로그아웃](#) Root User

메시지			도움말 ?
	심각도: 감사	마지막: 모두	필터 1 표시 중 - 3/3 << < 페이지 1/1 > >>
시간소인	사용자	메시지	
10. 9. 26. 오전 1:46	시스템	CRRBF0547: 스케줄 항목 'TestSchedule'이(가) 작성되었습니다.	
10. 9. 26. 오전 1:42	시스템	CRRBF0117: root 인증	

감사 로깅을 사용하여 로그되지 않는 항목에 대해서만 메시지가 표시됩니다. 270 페이지의 『감사 로깅』의 내용을 참조하십시오. 메시지를 보려면 감사 이외의 심각도를 사용하십시오.

제공된 제어 중 하나 이상을 사용한 후 필터를 클릭하여 목록을 필터링할 수 있습니다.

- 텍스트 상자: 검색할 문자열을 입력하십시오.
- 심각도: 다음 중 하나를 선택하십시오.
  - 모두(기본값)
  - 오류
  - 정보
  - 경고
  - 감사
- 기간: 다음 중 하나를 선택하십시오.
  - 모두(기본값)
  - 12시간
  - 24시간
  - 2일
  - 3일
  - 4일
  - 5일

목록 업데이트 시간은 메시지의 수 및 필터 기준에 따라 다릅니다. 필터가 좁을수록 목록 업데이트가 빨라집니다. 모든 심각도 및 모든 기간에서 문자열을 검색하면 처리하는데 시간이 걸립니다.

각 메시지에 대해 세 개의 필드가 표시됩니다.

- 시간소인
- 사용자
- 메시지

관리 > 시스템에는 로그 크기를 제어하는 설정이 제공됩니다.

- 감사 로그 자동 정리일 수
- 오류 로그 자동 정리일 수
- 정보 로그 자동 정리일 수
- 경고 로그 자동 정리일 수

자세한 정보는 259 페이지의 『시스템 구성 설정』의 내용을 참조하십시오.

## 번역된 메시지

영어가 아닌 메시지는 동적으로 번역됩니다.

en\_US가 아닌 언어로 Build Forge를 사용 중인 경우 메시지는 동적으로 번역됩니다. 관리 > 메시지를 선택하여 메시지 목록을 표시하십시오. 메시지는 15초마다 250의 일괄처리로 번역됩니다.

메시지가 아직 번역되지 않은 경우 메시지 필드에는 Message Not Yet Translated가 표시됩니다.

**중요사항:** 현재 설정과 다른 언어를 사용하는 새 사용자가 추가되면 새 사용자의 언어가 메시지에 사용됩니다. 기존 메시지는 이전 언어나 새 사용자 언어로 표시되지 않습니다.

## 작업 상태에 대한 RSS 데이터 공급에 등록

이제 RSS 데이터 공급을 사용하여 개별 작업의 상태를 추적하고 필터할 수 있습니다. 작업에 대한 Build Forge RSS 데이터 공급은 Build Forge 관리 콘솔에서 서버 상태와 동일한 정보를 표시합니다.

작업 상태에 대한 RSS 데이터 공급에 등록하려면 다음을 수행하십시오.

1. Build Forge 관리 콘솔에서 작업을 선택하십시오.

웹 브라우저가 RSS 데이터 공급을 발견하고 브라우저 주소 표시줄에 RSS 아이콘을 표시합니다.

2. RSS 집계기 도구에서 Build Forge RSS 데이터 공급을 로드하십시오.

예를 들어, URL을 복사하여 RSS 데이터 공급 목록에 추가하거나 RSS 아이콘을 끌어서 놓아 RSS 데이터 공급 목록에 URL을 추가하십시오.

3. RSS 데이터 공급에 등록하여 URL을 저장하고 업데이트가 발생할 때 알림을 받으십시오.

**참고:**

- URL 로드 및 RSS 데이터 공급 등록에 대한 정보는 RSS 집계기 도구 문서를 참조하십시오.
- 영어가 아닌 다른 언어로 RSS 데이터 공급을 통해 Build Forge 작업 상태, 시스템 메시지 또는 서버 상태를 보려면 RSS 집계기 도구가 UTF-8 멀티바이트 문자 인코딩을 지원해야 합니다.
- RSS 피드는 활동 보고를 위해 시간소인을 사용합니다. 시간소인은 Build Forge 사용자 계정에 대한 시간대 설정을 기반으로 합니다. 사용자 브라우저가 다른 시간대에 있는 경우 정확한 시간소인을 얻으려면 Build Forge 계정과 동일한 시간대로 이를 설정해야 합니다.

## 시스템 메시지에 대한 RSS 데이터 공급 필터링

RSS 데이터 공급에 표시되는 메시지를 필터할 수 있습니다. 이를 수행하려면 관리 콘솔의 **관리 > 메시지** 페이지에 있는 필터 특성을 사용하십시오.

예를 들어, 심각도 특성을 감사로 변경하거나 최신 특성을 12시간으로 변경할 수 있습니다. 필터 특성을 변경할 때 메시지 페이지의 RSS 아이콘 URL이 자동으로 업데이트됩니다.

업데이트된 URL을 RSS 집계기 도구로 로드하려면 메시지 페이지에서 RSS 아이콘을 마우스 오른쪽 단추로 클릭하여 링크를 복사하거나 업데이트된 RSS 아이콘을 도구의 RSS 데이터 공급 목록으로 끌어서 놓으십시오.

**참고:**

- URL 로드 및 RSS 데이터 공급 등록에 대한 세부사항은 RSS 집계기 도구 문서를 참조하십시오.
- 영어가 아닌 다른 언어로 RSS 데이터 공급을 통해 Build Forge 시스템 메시지 또는 서버 상태를 보려면 RSS 집계기 도구가 UTF-8 멀티바이트 문자 인코딩을 지원해야 합니다.
- RSS 피드는 활동 보고를 위해 시간소인을 사용합니다. 시간소인은 Build Forge 사용자 계정에 대한 시간대 설정을 기반으로 합니다. 사용자 브라우저가 다른 시간대에 있는 경우 정확한 시간소인을 얻으려면 Build Forge 계정과 동일한 시간대로 이를 설정해야 합니다.

---

## 보안 패널

보안 패널에서는 보안 서비스를 사용 가능하게 할 수 있습니다.

- **SSL:** 이 패널에서 SSL을 사용 가능하게 하는 것은 시스템 전체에서 SSL 보안 기능을 사용 가능하게 하는 작업의 일부일 뿐입니다. SSL을 켜려면 추가 작업이 필요합니다.
- **비밀번호 암호화:** 비밀번호 암호화를 사용 가능하게 하는 것은 시스템 전체에서 비밀번호 암호화를 사용 가능하게 하는 작업의 일부일 뿐입니다.

보안 패널을 보려면 **관리 > 보안**을 선택하십시오.

The screenshot shows the IBM Security Manager console interface. At the top, there are tabs for 'IBM', 'UI 구성', '콘솔', '보고서', and a '로그아웃: Root User' link. Below these is a navigation bar with buttons for '보안' (Security), '저장' (Save), '마스터 BFClient.conf 업데이트' (Update Master BFClient.conf), '키 파일 내보내기' (Export Key File), '새 키 파일' (New Key File), and '키 생성' (Generate Key). The main content area has a dropdown menu for '액세스:' (Access) set to '보안' (Security). Below this, there are three sections: 'SSL 사용:' (SSL Use) with a dropdown set to '아니오' (No), '비밀번호 암호화 사용:' (Password Encryption Use) with a dropdown set to '아니오' (No), and 'FIPS 사용:' (FIPS Use) with a dropdown set to '아니오' (No).

**중요사항:** 보안 패널에서는 일부 설정만 수행됩니다. 추가 정보는 106 페이지의 『보안 기능』을 참조하십시오.

## SSL 사용

**설치 전제조건:** 설치 중에 SSL 구성에서 사용하는 다음 두 가지를 지정합니다.

- **SSL 포트:** 웹 및 애플리케이션 서버 패널에서 지정됩니다. 이 포트는 아래에서 선택하는 구성에 지정된 포트와 일치해야 합니다. 설치 중과 구성에서의 기본값은 포트 8443입니다. 이 포트는 Apache Tomcat의 인증 서블릿이 로그인 중에 사용자 로그인 신임을 인코드 또는 암호화하기 위해 사용합니다.
- **인증서:** 웹 및 애플리케이션 서버 패널에서 지정됩니다. 자체 인증서를 제공했거나 설치 프로그램에서 자체 서명 인증서를 작성할 수 있도록 허용했습니다. 인증서는 기본 키 저장소에 저장됩니다. 키 저장소 위치는 이름 지정된 SSL 구성에서 정의됩니다.
- **SSL 사용을 예로 설정하십시오.** 다음과 같은 추가 특성이 표시됩니다.
  - **LDAP 아웃바운드:** LDAP를 통한 아웃바운드 통신에 사용되는 구성을 지정합니다. 기본값은 기본 JSSE 아웃바운드 SSL입니다.
  - **엔진 대 에이전트 기본 아웃바운드:** 엔진 컴포넌트에서 에이전트로의 통신에 사용되는 구성을 지정합니다. 기본값은 기본 OpenSSL 아웃바운드 SSL입니다.
  - **서비스 계층 인바운드:** 서비스 계층 컴포넌트가 웹 인터페이스 컴포넌트와 엔진 컴포넌트로부터의 통신을 승인하기 위해 사용하는 구성을 지정합니다. 기본값은 기본 JSSE 인바운드 SSL입니다.

- 서비스 계층 아웃바운드(JSSE): 엔진 컴포넌트와 웹 인터페이스 컴포넌트 서비스 계층 컴포넌트가 데이터베이스와 통신하기 위해 사용하는 JSSE 구성을 지정합니다. 기본값은 기본 JSSE 아웃바운드 SSL입니다.
- 서비스 계층 아웃바운드(OpenSSL): 엔진 컴포넌트와 웹 인터페이스 컴포넌트 서비스 계층 컴포넌트가 데이터베이스와 통신하기 위해 사용하는 OpenSSL 구성을 지정합니다. 기본값은 기본 OpenSSL 아웃바운드 SSL입니다.
- 저장을 클릭하십시오.
- 마스터 **BFClient.conf** 업데이트를 클릭하십시오. 이 단계는 이러한 특성 설정을 사용하여 BFClient.conf 파일을 편집합니다.
- Build Forge를 다시 시작하십시오. 보안 통신은 시스템이 이러한 설정을 사용하기 시작할 때까지 유효하지 않습니다.

선택하는 구성은 **SSL** 패널에서 정의됩니다.

SSL 이후 요구사항은 이 패널에서 사용됩니다.

- 인증서 분배: 인증서는 에이전트 호스트의 키 저장소, 데이터베이스 호스트 및 실행 중인 모든 추가 Build Forge 설치에 설치해야 합니다(예비 구성).
- 에이전트 SSL 사용: 엔진 컴포넌트와 에이전트 간의 통신에 SSL을 사용하려면 각 에이전트가 SSL을 사용하도록 구성해야 합니다.
- API 클라이언트 사용: 서비스 계층 컴포넌트와 통신하려면 모든 API 클라이언트가 SSL을 구성해야 합니다.

## 비밀번호 암호화 사용

전제조건:

- **SSL 포트:** 웹 및 애플리케이션 서버 패널에서 지정됩니다. 이 포트는 아래에서 선택하는 구성에 지정된 포트와 일치해야 합니다. 설치 중과 구성에서의 기본값은 포트 8443입니다. 이 포트는 Apache Tomcat의 인증 서블릿이 로그인 중에 사용자 로그인 신임을 인코드 또는 암호화하기 위해 사용합니다.
- **인증서:** 웹 및 애플리케이션 서버 패널에서 지정됩니다. 자체 인증서를 제공했거나 설치 프로그램에서 자체 서명 인증서를 작성할 수 있도록 허용했습니다. 인증서는 기본 키 저장소에 저장됩니다. 키 저장소 위치는 이름 지정된 SSL 구성에서 정의됩니다.
- **비밀번호 암호화 사용**을 예로 설정하십시오.
- 저장을 클릭하십시오.
- 마스터 **BFClient.conf** 업데이트를 클릭하십시오. 이 단계는 이러한 특성 설정을 사용하여 BFClient.conf 파일을 편집합니다.
- Build Forge를 다시 시작하십시오. 비밀번호 암호화는 시스템이 이러한 설정을 사용하기 시작할 때까지 유효하지 않습니다.



- 사용 가능한 경우, 서버 인증 비밀번호 및 콘솔에서 작성된 사용자의 사용자 비밀번호를 포함하여 콘솔에서 입력된 모든 새 비밀번호가 암호화됩니다.

추가 요구사항:

암호화를 사용하도록 설정한 후에는 다음을 수행해야 합니다.

- 모든 에이전트에서 암호화를 사용하십시오. 키를 내보내고 각 에이전트의 구성에서 서버 인증 비밀번호를 업데이트하는 데 사용하십시오. 비밀번호는 BFAgent.conf에서 수동으로 업데이트되어야 합니다.
- 데이터베이스 액세스에 암호화된 비밀번호를 사용하십시오. 키를 내보내고 Build Forge가 데이터베이스에 로그인하기 위해 사용하는 데이터베이스 비밀번호를 업데이트하는 데 사용합니다. 비밀번호는 buildforge.conf에서 수동으로 업데이트되어야 합니다.

## 키 저장소 패널

키 저장소 패널에는 개별 키 저장소에 대한 구성이 포함됩니다. 관리 > 보안 > SSL 패널에서 SSL 구성을 편집하는 경우 이러한 개별 구성을 SSL 구성의 일부로 선택할 수 있습니다.

구성을 작성하거나 제공된 구성을 사용할 수 있습니다.

- 기본 JSSE 키 저장소
- 기본 JSSE 신뢰 저장소
- 기본 OpenSSL CA 저장소(인증 권한)
- 기본 OpenSSL Cert 저장소(인증서용)
- 기본 OpenSSL 키 저장소

각 키 저장소는 다음과 같은 특성을 갖습니다.

### 이름

**액세스** 이 키 저장소를 편집 또는 삭제할 수 있는 사용자를 정의하는 액세스 그룹.

**위치** 키 저장소 파일의 위치. 기본 키 저장소는 모두 Build Forge 키 저장소의 기본 위치(<bfinstall>/keystore)를 사용합니다. 제공된 Tomcat 애플리케이션 서버가 아니라 WebSphere Application Server를 애플리케이션 서버로 사용 중인 경우 절대 경로를 지정하십시오.

### 키 저장소 유형

키 저장소는 다음 유형 중 하나여야 합니다.

- JKS
- PKCS12
- JCEKS
- PEM

## 비밀번호

키 저장소에 액세스할 때 사용해야 하는 비밀번호를 지정합니다.

**확인** 비밀번호 확인을 위해 여기서 다시 지정하십시오.

## 키 저장소 및 WebSphere Application Server

제공된 Tomcat 애플리케이션 서버가 아니라 WebSphere Application Server를 사용하는 경우, 키 저장소 구성에 추가 요구사항이 적용됩니다.

- 위치 필드: 상대 경로가 아니라 절대 경로를 제공해야 합니다.
- 다중 서비스 컴포넌트: 다중 Build Forge 서비스 컴포넌트를 설치하는 경우, 다른 호스트에 설치됩니다. 각 서비스 구성요소의 보안을 구성해야 합니다. 위치 필드에 지정된 키 저장소 경로는 각 서비스 컴포넌트에 대해 동일해야 합니다.

## SSL 패널

SSL 패널에는 SSL의 개별 구성이 포함됩니다. 관리 > 보안 패널에서 **SSL 사용**을 예로 선택하면 이러한 개별 구성을 SSL 사용의 일부로 선택할 수 있습니다.

고유한 구성을 작성하거나 제공된 구성을 사용할 수 있습니다.

- 기본 JSSE 인바운드 SSL
- 기본 JSSE 아웃바운드 SSL
- 기본 OpenSSL 인바운드 SSL
- 기본 OpenSSL 아웃바운드 SSL

각 구성은 다음과 같은 특성을 갖습니다.

**이름** 이 구성의 이름

**액세스** 이 키 저장소를 편집 또는 삭제할 수 있는 사용자를 정의하는 액세스 그룹.

**유형** JSSE 또는 OpenSSL을 선택하십시오.

### 클라이언트 인증

다음 중 하나를 선택하십시오.

- 수행 안함
- 지원됨
- 필수

### 서버 인증 별명

서버 인증 별명을 입력하십시오.

### 클라이언트 인증 별명

클라이언트 인증 별명을 입력하십시오.

### 키 저장소 구성

키 저장소 구성 중 하나를 선택하십시오. 이 구성은 키 저장소 패널에서 구성됩니다.

### 신뢰 저장소 구성

키 저장소 구성 중 하나를 선택하십시오. 이 구성은 키 저장소 패널에서 구성됩니다.

### 데이터 교환 프로토콜

다음 중 하나:

- SSLv2
- SSLv3
- SSL
- TLSv1
- TLS
- SSL\_TLS

### 암호 스위트 그룹

높음, 중간, 낮음 또는 모두 중 하나. 순위가 높은 암호가 보다 안전하지만 성능이 저하됩니다.

### 암호 대체 목록

## SSO 패널

SSO 패널에는 SSO(Single Sign-On)에 대한 구성이 포함됩니다. 새 구성을 작성하거나 제공된 구성을 사용할 수 있습니다.

- 양식 SSO 인터셉터
- SPNEGO SSO 인터셉터

단일 사인은 서비스를 사용하려면 추가 단계가 필요하며 이 패널에서 완전히 수행되지는 않습니다. 107 페이지의 『싱글 사인온 구현』의 내용을 참조하십시오.

각 구성은 다음과 같은 특성을 갖습니다.

**이름** SSO 구성의 이름

**활성** 이 인터셉터를 활성화하려면 예를 선택하십시오.

**액세스** 이 구성을 편집 또는 삭제할 수 있는 사용자를 정의하는 액세스 그룹.

### Java 클래스

이 SSO 구성을 구현하는 Java 클래스. 다음 두 가지 클래스가 제공됩니다.

- 양식 인터셉터:

`com.buildforge.services.server.sso.form.FORMSSOInterceptor`

- SPNEGO 인터셉터:

`com.buildforge.services.server.sso.spnego.SPNEGOSSOInterceptor`

#### 키 저장소 유형

키 저장소는 다음 유형 중 하나여야 합니다.

- JKS
- PKCS12
- JCEKS
- PEM

#### 비밀번호

키 저장소에 액세스할 때 사용해야 하는 비밀번호를 지정합니다.

**확인** 비밀번호 확인을 위해 여기서 다시 지정하십시오.

---

## 라이선스 관리

사용자를 작성할 때 시스템은 사용자에 대한 정보를 데이터베이스에 저장합니다. 사용자가 시스템에 로그인한 후에 시스템은 사용자에게 라이선스를 지정합니다.

#### 권한 부여된 사용자

라이선스는 사용자에게 지정됩니다. 사용자는 항상 로그인할 수 있습니다. 로그인한 라이선스를 이용합니다.

#### 유동 사용자

Floating 라이선스 사용자는 사용 가능한 라이선스가 있는 경우에만 로그인할 수 있습니다. 모든 라이선스가 사용중인 경우에는 로그인할 수 없습니다.

#### 토큰 라이선스

다중 Rational 제품에 대한 라이선스를 보유한 고객은 토큰 라이선스 사용을 선택할 수 있습니다. 이들 고객은 이러한 제품에 사용할 수 있는 토큰 라이선스의 풀을 구입합니다. 토큰 라이선스는 로그인뿐만 아니라 특정 기능 사용 시 이용할 수 있습니다.

토큰 라이선스 지원은 유동 서버, 유동 사용자, 어댑터 및 QuickReport 라이선스에 사용할 수 있습니다. 고정 라이선스(권한 부여된 서버 또는 권한 부여된 사용자)에는 사용할 수 없습니다. 서버, 어댑터 및 QuickReport 라이선스는 체크아웃된 경우 Build Forge®를 종료할 때까지 릴리스되지 않습니다.

토큰 라이선스 지원은 Rational Build Forge 버전 7.1.3.4 이상에 사용할 수 있습니다. 토큰 라이선스 지원에는 Rational Common Licensing 버전 8.1.2 이상도 필요합니다.

토큰 라이선싱에 대한 자세한 정보는 한국 IBM 담당자에게 문의하십시오.

소유하고 있는 라이선스 수보다 많은 사용자를 작성하는 경우, 라이선스를 추가로 구입하거나 루트 사용자 계정을 사용하여 일부 사용자에서 라이선스를 제거(또는 시스템에서 일부 사용자를 삭제)해야 새 사용자가 로그인할 수 있습니다.

사용자는 한 번에 한 명만 로그인할 수 있습니다. 한 컴퓨터에 로그인한 상태에서 동일한 이름으로 다른 컴퓨터에서 로그인하는 경우, 원래의 세션이 무효화됩니다.

**참고:** 버전 3.8 이전의 시스템을 구입한 경우, 라이선스 체계가 여기 설명된 것과 다를 수 있습니다. 라이선싱에 대한 문의사항이 있으면 고객 지원에 문의하십시오.

## 새 라이선스 키 입력

라이선스 키를 변경하려면 **관리 > 시스템**을 선택한 다음 설정 목록에서 라이선스 서버 설정을 찾으십시오. 필터 상자에 "라이선스"를 입력하여 목록을 페이지화하지 않고 이 설정을 신속하게 표시하십시오.

목록에서 라이선스 서버 항목을 클릭하면 시스템이 콘텐츠 패널 맨 아래에 편집 가능한 라이선스 서버 필드가 있는 탭을 표시합니다. 필드를 편집한 다음 저장 단추를 클릭하십시오.

## 토큰 라이선스 로그

Rational Common Licensing 구성 파일 RCL\_Log\_Config.txt는 Build Forge 설치 디렉토리에 있습니다. 기본적으로 이 설치 디렉토리의 위치는 다음과 같습니다.

- Windows 시스템  
C:\Program Files\IBM\Build Forge
- UNIX 및 Linux 시스템  
/opt/buildforge/Platform

기본적으로 Rational Common Licensing 로그 파일은 설치 디렉토리 아래의 temp/rc1 서브디렉토리에 작성됩니다.

구성 파일과 로그 파일을 모두 동일한 위치에 저장하려면 다음 단계를 수행하십시오.

1. Build Forge를 중지하십시오.
2. RCL\_Log\_Config.txt 파일을 <bfinstall>/token\_libs/rc1로 이동시키십시오.
3. Build Forge를 다시 시작하십시오. 모든 토큰 로그가 이제 <bfinstall>/token\_libs/rc1에 생성됩니다.

---

## 엔진 관리

이 주제에서는 엔진(bfengine.exe / bfengine.pl)을 일시정지, 시작 또는 중지하는 방법에 대해 설명합니다.

**참고:** 엔진을 실행시키는 데 문제점이 있는 경우 엔진이 생성하는 상태 및 오류 메시지를 볼 수 있도록 포그라운드로 실행하십시오. Windows에서 시작 > 모든 프로그램 > **IBM Rational Build Forge** > 엔진 시작(포그라운드)을 선택하여 엔진을 포그라운드로 실행하십시오. 그러면, 콘솔 창에 출력이 표시됩니다. 엔진이 포그라운드로 실행되면 콘솔 창을 닫아 엔진을 중지할 수 있습니다.

## 엔진 일시정지

새 작업의 처리를 임시로 지연하려는 경우 엔진을 일시정지해야 합니다.

**참고:** 데이터베이스에 대해 직접 작업하려면(예: 데이터베이스를 백업하거나 복원하려는 경우) 먼저 실행 중인 모든 Build Forge 서비스 및 프로세스를 중지해야 합니다. 엔진을 일시정지하는 것은 이 경우에 충분하지 않습니다. 불일치 및/또는 손상이 발생합니다.

엔진을 일시정지하려면 관리 > 시스템을 선택하고 **Build Forge** 엔진 일시정지 특성을 찾으십시오(필터 상자 사용). 시스템이 **Build Forge** 엔진 일시정지 설정을 편집하기 위한 패널을 표시합니다. 특성을 Y로 변경하고 구성 저장 단추를 클릭하십시오.

시스템이 엔진을 일시정지하지만 현재 실행 중인 프로젝트는 계속됩니다. 데이터베이스에 데이터가 기록되지 않도록 현재 활성 작업이 완료되기를 기다려야 합니다.

엔진을 다시 활성화하려면 특성의 값을 다시 N으로 변경하고 구성 저장을 다시 클릭하십시오.

## 엔진 시작 및 중지

다음 절은 Windows®, Linux® 또는 Solaris에서 엔진을 시작하고 중지하는 방법에 대해 설명합니다.

### Windows에서 엔진 시작 및 중지

Windows의 경우 다음을 수행하십시오.

- 시작 > 프로그램 > **IBM Rational Build Forge** 관리 콘솔에서 다음 중 하나를 선택하십시오.
  - 엔진 서비스 시작
  - 엔진 서비스 중지

제어판: 관리 도구 > 서비스 제어판을 사용하여 **IBM Rational Build Forge** 관리 콘솔 서비스를 시작하거나 중지할 수도 있습니다.

포그라운드로 실행: 엔진을 실행하는 데 문제점이 있으면, 엔진이 생성하는 상태 및 오류 메시지를 볼 수 있도록 포그라운드로 실행하십시오(시작 > 프로그램 > **IBM**

**Rational Build Forge** 관리 콘솔 > 엔진 시작(포그라운드)). 관리 콘솔이 실행될 때 로그 출력이 콘솔 창에 표시됩니다. 이 모드에서 엔진을 중지하려면 콘솔 창을 닫으십시오.

### Linux 또는 Solaris 시스템에서 엔진 시작 및 중지

rc 파일이 설치되어 있는 경우 다음 명령을 사용하여 제품을 시작 및 중지하십시오.

```
$ /opt/buildforge/rc/buildforge start
```

```
$ /opt/buildforge/rc/buildforge stop
```

rc 파일이 설치되지 않은 경우, 다음을 사용하여 제품을 시작하십시오.

```
$ /<bfinstall>/Platform/buildforge &
```

프로세스 ID를 판별한 후 kill 명령을 실행하여 중지하십시오.

```
$ ps aux | grep buildforge
```

```
$ kill ${<PID>}
```

---

## 데이터베이스 관리

이 주제는 특히 기본 구성을 변경하려는 경우 관리 콘솔 데이터베이스를 설정할 때 중요한 문제에 대해 설명합니다.

### 데이터베이스 로그 파일 삭제

정기적으로 데이터베이스 로그 파일을 삭제하십시오.

시스템이 관리 콘솔 설치 디렉토리의 db.log 파일에 데이터베이스 디버깅 정보를 저장합니다. 매달 이 파일의 크기를 확인하고 콘솔 컴퓨터에 여유 공간이 필요한 경우 해당 파일을 삭제해야 합니다.

---

## 오류 메시지

이 주제는 관리 콘솔을 사용하는 동안 발생할 수 있는 오류 메시지에 대해 설명합니다.

### 활성 단계 없음

이 메시지가 스케줄 항목의 다음 실행 열에 표시되면 연관된 프로젝트의 모든 단계가 사용 불가능합니다. 프로젝트의 단계 목록을 표시하고 사용 불가능한 각 단계 옆의 빨간색 원을 클릭하여 일부 단계를 사용 가능하도록 하십시오.

### 라이선스 키가 올바르지 않거나 또는 Build Forge 라이선스 키가 손상 또는 누락되었습니다.

라이선스 키가 설치한 제품 버전에 올바르지 않거나 만기되었습니다. 새 라이선스 키를 입력하십시오. 285 페이지의 『새 라이선스 키 입력』의 내용을 참조하십시오.

## 데이터베이스 라이선스 필요

LicMgr: 5140: A database license is required.

위의 메시지가 콘솔 출력에 표시되면(포그라운드로 콘솔을 실행할 때 표시됨. 또는 Linux<sup>®</sup>나 UNIX<sup>®</sup> 시스템에서 콘솔 로그를 보는 경우) 엔터프라이즈 라이선스의 이점 없이 고급 데이터베이스 기능을 사용하려고 시도한 것입니다. 예를 들어, 비DB2<sup>®</sup> 데이터베이스로 시스템을 사용하려고 할 수도 있습니다. 라이선스를 업그레이드하려면 지원에 문의하십시오.



## 제 17 장 서버

이 주제는 관리 콘솔에서 서버 자원을 설정 및 관리하는 방법에 대해 설명합니다.

### 서버 자원 정보

콘솔의 서버 자원은 프로젝트 또는 단계를 실행할 수 있는 호스트를 나타냅니다.

- 서버에는 *Manifest*가 있습니다. *Manifest*는 서버 특성 목록입니다. 콜렉터가 실행될 때 *Manifest*가 채워집니다. 서버에 지정된 콜렉터가 없는 경우, 서버의 *Manifest*에서 몇 가지 특성이 자동으로 채워집니다.
- 콜렉터가 *Manifest*를 채웁니다. 콜렉터는 서버에 지정됩니다. 콜렉터는 특성 값을 설정하기도 하고 서버의 에이전트에서 특성 값을 수집하기도 합니다.
- 선택기는 *Manifest*에서 서버 특성을 읽습니다. 프로젝트는 선택기를 사용하여 단계를 실행할 서버를 판별할 수 있습니다.

시스템을 설정하는 관리자로서 먼저 서버를 작성하십시오. 그런 다음, 서버에 지정할 수 있는 콜렉터를 작성하십시오. 콜렉터를 실행하여 서버 *Manifest*를 채우십시오. 해당 태스크가 완료되면, 빌드 엔지니어가 선택기를 사용하는 프로젝트를 작성하여 프로젝트 단계가 실행되는 위치를 판별할 수 있습니다.

각 서버 자원은 에이전트가 설치되어 있는 호스트를 지시합니다. 서버 자원을 추가할 때 관리 콘솔이 특정 호스트를 액세스 및 사용하는 방법을 설명하게 됩니다.

서버를 작성하기 전에 종속된 기타 데이터 오브젝트가 이미 있는지 확인하십시오. 서버에 다음 항목을 지정하십시오.

항목	필수 또는 선택사항	설명
서버 인증	필수	서버에 사용할 로그인 이름과 비밀번호를 지정합니다.
콜렉터	선택사항입니다.	기본 특성은 물론 시스템이 서버에서 수집하는 특성을 정의합니다.
환경	선택사항입니다.	서버에서 프로젝트가 실행될 때마다 지정될 환경 변수를 지정합니다.

**참고:** 원하는 경우, 관리 콘솔에서 단일 실제 서버에 대해 두 개 이상의 서버 오브젝트를 작성할 수도 있습니다. 이 서버 오브젝트는 논리 서버라고 부릅니다. 프로젝트가 다른 특성을 가진 동일한 하드웨어에 액세스하기 위해 일반적으로 논리 서버를 사용합니다. 예를 들어, 두 개의 논리 서버가 다른 경로 또는 다른 환경을 사용할 수 있습니다.

- 경로가 다른 두 개의 논리 서버는 동일한 서버에 별개의 작업 디렉토리를 작성합니다. 모든 출력이 다른 디렉토리에 저장되기 때문에 한 서버를 사용하여 수행된 작업을 다른 서버를 사용하여 수행된 작업과 식별할 수 있습니다.
- 환경이 다른 두 개의 논리 서버는 서로 다른 시작 환경 값을 사용하여 단계를 실행합니다.

## 서버 패널 정보

프로젝트를 실행할 서버를 지정하려면 서버 패널을 사용하십시오.

관리 콘솔에서 서버를 관리하려면 왼쪽 메뉴의 서버를 클릭하십시오.

The screenshot shows the IBM Rational Build Forge server configuration interface. At the top, there are tabs for 'IBM', 'UI 구성', '콘솔', '보고서', and '로그아웃: Root User'. Below these is a '서버' (Server) section with a '서버 추가' (Add Server) button. A table lists servers with columns: 이름 (Name), 액세스 (Access), 콜렉터 (Collector), 호스트 (Host), and 경로 (Path). The first entry is 'HelloWorldServer' with access '개발자' (Developer), collector '9.125.75.117', and path 'C:'. Below the table are buttons for '(새 서버) 저장' (Save), '복사' (Copy), '삭제' (Delete), '작업 계수 다시 설정' (Reset Job Count), 'Manifest 새로 고침' (Refresh Manifest), and '연결 테스트' (Test Connection). The '세부사항' (Details) section has tabs for 'Manifest', '테스트 결과' (Test Results), and '노트 (0)' (Notes). The 'Manifest' tab is active, showing fields for '사용 안함' (Not Used), '이름' (Name), '경로' (Path), '호스트' (Host), '인증' (Authentication), '액세스' (Access), '최대 작업' (Max Jobs), '콜렉터' (Collector), '비밀번호 암호화' (Password Encryption), 'SSL 사용' (SSL Usage), '환경' (Environment), '파일' (File), and '초기 작업' (Initial Job).

## 세부사항 탭

세부사항 탭에 다음 필드가 있습니다.

**호스트** Rational Build Forge 에이전트를 실행 중인 호스트의 이름

**콜렉터** 서버와 함께 사용할 콜렉터

**환경** 서버와 함께 사용할 환경

**인증** 사용할 서버 인증(로그인 및 비밀번호)

**비밀번호 암호화**

에이전트 비밀번호가 암호화되는지 여부를 판별합니다.

**파일** .get 및 .put 명령과 함께 사용할 파일 전송 유형

**액세스** 서버를 사용할 수 있는 사용자의 액세스 그룹

**SSL 사용**

Rational Build Forge 시스템이 SSL을 사용하도록 구성했는지 여부를 표시합니다.

## 최대 작업 수

동시에 실행할 가장 많은 작업 수

## Manifest 탭

Manifest 탭은 다양한 서버 특성을 표시합니다. 기본 특성 세트를 제공합니다. 컬렉터를 사용하여 추가 특성을 포함할 수 있습니다.

## 테스트 결과 탭

테스트 결과 탭은 관리 콘솔과 서버 간 연결의 마지막 테스트 상태를 표시합니다.

## 노트 탭

노트 탭은 서버에 관한 주석을 저장 및 공유하기 편리한 위치를 제공합니다.

# 서버 작성

서버 자원은 프로젝트 및 단계를 실행할 수 있는 호스트를 나타냅니다.

서버를 작성하려면 다음을 수행하십시오.

1. 왼쪽 메뉴에서 서버를 선택하십시오. 시스템이 기본 콘텐츠 패널의 맨 아래에 새 서버 패널을 표시합니다. 기존 서버를 선택한 경우, 새 서버를 추가할 수 있도록 서버 추가를 클릭하여 패널을 지우십시오.
2. 서버 세부사항을 제공하십시오.

새 서버 탭: 저장, 복사, 삭제, 작업 개수 다시 설정, Manifest 새로 고침, 연결 테스트

세부사항: Manifest, 테스트 결과, 노트 (0)

☐ 사용 안함 | 이름: | 경로: |

호스트: | 포트: | 환경: |

인증: | 비밀번호 암호화: | 파일: |

액세스: | 사용 불가능 | |

개발자: | SSL 사용: |

최대 작업: 3 | 마니오

- 이름: 서버에 이름을 제공하십시오. 이 이름은 서버의 BF\_NAME 특성입니다. 선택기에서 이 이름을 참조하여 이름으로 서버를 지정합니다.
- 경로: 프로젝트 및 작업 디렉토리를 작성할 때 서버가 사용하는 디렉토리(예: c:/buildforgeprojects)를 지정하십시오. 시스템은 빌드 디렉토리를 사용할 때 이 경로 값을 시작점으로 사용합니다.

팁: 시스템은 서버 경로를 작성하지 않습니다. 빌드가 서버에 액세스를 시도하기 전에 경로가 존재해야 합니다. 경로가 존재하지 않으면 빌드가 실패합니다.

- 호스트: 에이전트를 실행 중인 실제 컴퓨터의 호스트 이름을 제공하십시오. 관리 콘솔 컴퓨터를 서버로 정의하고 있는 경우 localhost 값을 사용하십시오(에이전트도 관리 콘솔에 설치되어야 함).

참고: 호스트 이름과 함께 포트 번호를 포함시킬 수 있습니다(예: *host\_name:port\_number*). 호스트 이름과 함께 포트 번호를 지정하는 경우, 이 번호는 기본 에이전트 포트 시스템 설정으로 정의된 포트 번호를 대체합니다(관리 > 시스템 > 기본 에이전트 포트를 클릭하십시오.)

참고: 호스트 이름 앞에 프로토콜이 오지 않게 하십시오. 예를 들어, *http://*를 사용하지 마십시오.

- 인증: 이 서버에 사용할 서버 인증을 선택하십시오.
- 액세스: 이 서버를 사용할 수 있는 사용자의 액세스 그룹을 선택하십시오.
- 콜렉터: 이 서버에 사용할 콜렉터를 선택하십시오.
- 환경: 프로젝트를 실행하는 데 이 서버가 사용될 때마다 적용될 환경 변수 그룹을 선택하십시오. 이러한 변수는 다른 모든 변수보다 먼저 적용됩니다. 또한, 서버에 특정한 매개변수를 설정합니다.
- 파일: 이 서버에 대해 *.get* 및 *.put* 명령과 함께 사용할 수 있는 파일 전송 유형을 정의하십시오. 전송 안함(없음), 파일 읽기(*.get*), 파일 쓰기(*.put*) 또는 둘 다(*.get* 및 *.put*)를 선택할 수 있습니다.
- 최대 작업 수: 동시에 실행될 최대 작업 수를 입력하십시오. 기본값은 3입니다.
- 비밀번호 암호화 구성: 에이전트 비밀번호가 암호화되게 하려면 사용을 선택하십시오. 기본값은 사용 불가능입니다.
- SSL 사용: Build Forge 시스템이 SSL을 사용하도록 구성했고 이 서버 자원이 SSL을 통해 에이전트와 통신하게 하려면 예를 선택하십시오. 기본값은 아니오입니다.

참고: 예를 선택하더라도 Build Forge가 SSL에 대해 올바르게 구성되지 않은 경우, 이 서버 자원은 에이전트와 전혀 통신할 수 없습니다.

3. 저장을 클릭하십시오. 새 서버가 콘텐츠 패널 맨 위의 서버 목록에 표시됩니다. 서버를 올바르게 구성했는지 확인하려면 목록에서 서버를 선택한 후 연결 테스트를 클릭하십시오. 서버와 통신할 수 없는 경우 시스템이 오류를 보고합니다.

---

## 서버 테스트

서버에서 진단 테스트 세트를 수행할 수 있습니다.

진단 테스트를 수행하려면 다음을 수행하십시오.

1. 서버-> *server\_name*을 클릭하십시오.
2. 연결 테스트를 클릭하십시오.

서버 연결 테스트가 수행되고 서버 Manifest가 업데이트됩니다. 새로 고치기가 완료될 때까지 테스트 결과 탭에 진행 표시줄이 표시됩니다.

테스트 결과 탭의 결과를 보십시오. 추가 정보를 보려면 테스트 세부사항 보기를 클릭하십시오.

서버가 테스트에 실패하면 다음 조치 중 하나를 시도하십시오.

- 서버에 사용하는 서버 인증의 사용자 이름과 비밀번호가 올바른지 확인하십시오.
- 올바른 호스트 이름을 사용하고 있는지 확인하십시오.
- 서버에 에이전트를 다시 설치하거나 설치되어 있는지 확인하십시오.

---

## 서버 변경

필요한 경우 서버의 특성을 편집할 수 있습니다.

**중요사항:** 다음 조건이 true인 경우 콘솔을 다시 시작해야 합니다.

- 호스트 매개변수는 에이전트 호스트의 호스트 이름을 사용함
- 에이전트 호스트의 IP 주소 변경

DNS의 IP 주소는 Build Forge 서비스 컴포넌트를 호스트하는 JVM에서 캐시됩니다. 캐시를 지우고 서비스 컴포넌트가 새 IP 주소에서 에이전트를 찾도록 하려면 다시 시작해야 합니다.

호스트 매개변수에서 IP 주소를 사용하는 경우 다시 시작해야 합니다. 에이전트 호스트 IP 주소가 변경되면 호스트 매개변수를 새 주소로 변경하고 저장하십시오.

---

## 서버 사용 및 사용 안함

서버를 일시적으로 사용 불가능하게 할 수 있습니다. 사용 불가능한 서버 자원에 대해서는 작업을 실행할 수 없습니다.

서버를 사용 안함 또는 사용하려면 다음을 수행하십시오.

1. 서버를 클릭해서 서버 목록을 표시하십시오.
2. 수정할 서버 자원의 이름을 클릭하십시오. 해당 서버에 대한 세부사항이 표시됩니다.
3. 기본 패널 맨 아래에 있는 세부사항 탭에서 사용 안함을 선택하십시오.
4. 서버 저장을 클릭하십시오.

---

## 서버의 동시 작업 수 제한

최대 작업 수 서버 특성을 사용하여 특정 서버에서 동시에 실행할 수 있는 최대 작업 수를 지정할 수 있습니다.

## 이 태스크 정보

시스템은 특정 서버에서의 실행을 시도하는 프로세스 수를 제한합니다. 최대 작업 수 특성은 프로세스 수의 한계를 지정합니다. 이 특성의 기본값은 3입니다.

참고: 기타 프로그램은 서버에서 실행할 수 있습니다. 시스템 한계는 Build Forge 작업에만 적용됩니다.

참고: 기본 변수 BF\_RESERVE 및 BF\_EXCLUSIVE가 서버의 작업 슬롯 중 하나 또는 전부의 예약을 제어합니다. 자세한 정보는 313 페이지의 『사전 설정 특성』을 참조하십시오.

서버에 비기본값을 지정하려면 다음을 수행하십시오.

### 프로시저

1. 서버를 클릭하십시오.
2. 기존 서버를 선택하거나 새 서버를 작성하십시오.
3. 최대 작업 수 특성을 원하는 값으로 설정하십시오.
4. 저장을 클릭하십시오.

---

## 작업 계수 재설정

### 서버에서 작업 계수를 0으로 재설정

서버 페이지에서 작업 계수 다시 설정을 사용하여 선택된 서버의 작업 계수(BF\_JOBS)를 0으로 다시 설정하십시오. BF\_JOBS는 서버에서 현재 실행 중인 단계 또는 작업의 숫자입니다. 이 숫자가 너무 낮게 설정되면 작업이 작업 경합에서 실패할 수 있습니다.

이 선택사항으로 작업이 완료, 실패 또는 취소될 때 BF\_JOBS이 올바르게 재설정되지 않는 경우 이를 재설정할 수 있습니다.

예를 들어 여러 개의 작업을 취소할 때 가끔 BF\_JOBS를 재설정에서 실패합니다. BF\_JOBS를 다시 설정하지 않으면 Max Jobs(최대 작업 수) 설정의 한계에 도달하여 단계 또는 작업이 실행되지 않을 수 있습니다.

1. 서버를 선택하여 서버 목록을 표시하십시오.
2. 서버를 선택하십시오.
3. 작업 계수 다시 설정을 클릭하십시오.

## 모든 서버에서 작업 계수를 0으로 재설정

서버 작업 계수 다시 설정 페이지를 사용하여 모든 서버의 작업 계수(BF\_JOBS)를 동시에 0으로 다시 설정할 수 있습니다. BF\_JOBS 특성은 서버에서 실행 중인 단계 또는 작업의 숫자입니다. 이 숫자가 너무 낮게 설정되면 작업이 작업 경험에서 실패할 수 있습니다.

모든 서버의 BF\_JOBS 값을 다시 설정하면, 서버 작업 계수 다시 설정 값은 아니오(기본값 설정)로 되돌려집니다.

예를 들어, 여러 작업을 취소하면 간혹 빌드 시스템이 BF\_JOBS 값을 다시 설정하지 못하는 경우도 생깁니다. BF\_JOBS 값이 다시 설정되지 않으면, 값이 기본 Max Jobs(최대 작업 수) 시스템 설정의 한계에 도달하여 단계 또는 작업이 실행되지 않을 수 있습니다.

1. 시스템 구성 설정의 목록을 표시하려면 **관리 > 시스템**을 클릭하십시오.
2. 서버 작업 계수 다시 설정을 찾으십시오.
3. 서버 작업 계수 다시 설정을 클릭하십시오.
4. 세부사항 탭을 클릭하고 값으로 **예**를 클릭하십시오.
5. 저장을 클릭하십시오.

---

## 서버 상태에 대한 RSS 데이터 피드

관리 콘솔은 서버 상태 검사를 실행하여 서버가 기능 테스트를 패스하고 에이전트가 로그인할 수 있는지 확인합니다. 테스트 결과 탭은 상태 검사 결과를 표시합니다. 관리 콘솔은 서버가 작성되거나 편집될 때마다 자동으로 서버 상태를 검사합니다. 예를 들어, 프로젝트를 실행하기 전에 언제든지 서버 상태 검사를 시작할 수 있습니다.

서버 상태에 대한 Build Forge RSS 데이터 공급은 Build Forge 관리 콘솔의 테스트 결과 탭과 동일한 정보를 표시합니다.

서버 상태에 대한 RSS 데이터 공급에 등록하려면 다음을 수행하십시오.

1. Build Forge 관리 콘솔에서 서버를 선택하십시오.

웹 브라우저가 RSS 데이터 공급을 발견하고 브라우저 주소 표시줄에 RSS 아이콘을 표시합니다.

2. RSS 집계기 도구에서 Build Forge RSS 데이터 공급을 로드하십시오.

예를 들어, URL을 복사하여 RSS 데이터 공급 목록에 추가하거나 RSS 아이콘을 끌어서 놓아 RSS 데이터 공급 목록에 URL을 추가하십시오.

3. RSS 데이터 공급에 등록하여 URL을 저장하고 업데이트가 발생할 때 알림을 받으십시오.

#### 참고:

- URL 로드 및 RSS 데이터 공급 등록에 대한 정보는 RSS 집계기 도구 문서를 참조하십시오.
- 영어가 아닌 다른 언어로 RSS 데이터 공급을 통해 Build Forge 시스템 메시지 또는 서버 상태를 보려면 RSS 집계기 도구가 UTF-8 멀티바이트 문자 인코딩을 지원해야 합니다.
- RSS 피드는 활동 보고를 위해 시간소인을 사용합니다. 시간소인은 Build Forge 사용자 계정에 대한 시간대 설정을 기반으로 합니다. 사용자 브라우저가 다른 시간대에 있는 경우 정확한 시간소인을 얻으려면 Build Forge 계정과 동일한 시간대로 이를 설정해야 합니다.

---

## 서버 인증

이 주제는 서버에서 실행할 프로젝트에 대한 액세스를 부여하는 방법에 대해 설명합니다.

### 서버 인증 정보

서버 인증을 사용하여 서버에 로그인 신임을 연관시키십시오.

서버 인증은 로그인 이름과 비밀번호를 하나 이상의 서버와 연관시킬 수 있는 하나의 이름 지정된 오브젝트로 저장합니다. 서버 인증 페이지를 사용하여 서버 인증을 작성하고 편집하십시오.

서버 인증 세트 관리를 통해 여러 서버에 동일한 신임을 사용하고 신임을 글로벌로 업데이트할 수 있습니다.

### 서버 인증 패널 정보

서버 인증을 작성하고 관리하려면 서버 인증 패널을 사용하십시오.

패널을 보려면 서버 > 서버 인증을 선택하십시오.





2. 이름에 인증 이름, 논리 이름을 입력하여 시스템에서 서버 인증을 식별하십시오.
3. 로그인에 서버 로그인 이름을 지정하십시오.

참고: 로그인 이름의 출처가 도메인 사용자인 경우, 이 필드에 도메인을 포함시키십시오. 예를 들어, MYDOMAIN/joeuser를 입력하십시오.

4. 비밀번호에 비밀번호를 입력하십시오.
5. 확인에 비밀번호를 다시 입력하십시오.
6. 서버 인증 저장을 클릭하십시오. 시스템이 사용자가 선택한 이름을 사용하여 새 서버 인증을 저장합니다.

## 결과

빌드 시스템이 사용자가 선택한 이름을 사용하여 새 서버 인증을 저장합니다.

## 서버 인증 대체

특수 환경 변수를 사용하여 서버가 서버에 지정된 서버 인증 대신 관리 콘솔 로그인 신임 정보를 사용하도록 강제 실행할 수 있습니다. 지정된 인증을 대체하려면 프로젝트 또는 단계에서 사용하는 환경에 값이 1인 `_USE_BFCREDS` 변수를 추가하십시오. 프로젝트 환경에 변수를 추가하는 경우 빌드 시스템은 프로젝트의 모든 단계에서 대체를 사용합니다.

빌드 시스템이 `_USE_BFCREDS=1`이 포함된 환경에서 단계 실행을 시도하면 시스템은 프로젝트를 시작한 사용자의 콘솔 로그인 신임 정보를 사용하여 단계 명령을 실행합니다.

참고: LDAP/Active Directory 인증을 사용하는 경우, `_USE_BFCREDS` 변수가 작동하려면 사용자 인증을 로컬로 저장 시스템 설정을 예(기본값)로 설정해야 합니다. 설정이 예이면 시스템이 사용자 인증 정보를 암호화된 양식으로 캐시한 다음 `_USE_BFCREDS`로 사용하도록 사용자 인증 정보에 액세스할 수 있습니다. 그렇지 않으면 시스템이 LDAP 정보를 저장하지 않아 사용할 수 없습니다.

팁: Windows에서는 `_USE_BFCREDS_DOMAIN` 변수도 설정할 것을 고려하십시오. 이 변수를 값 1로 설정하면 사용자의 도메인이 포함됩니다.

## 제한된 서버 인증 사용 허용

사용자는 액세스 불가능 서버 인증 실행 권한을 사용하여 사용자에게 액세스 권한이 없는 서버 인증으로 서버에서 단계를 실행할 수 있습니다.

전제조건으로, 사용자는 서버에 대한 액세스 권한이 이미 있거나 부여된 상태여야 합니다. (서버에 대한 액세스를 부여하려면 **서버 > 액세스**를 클릭하십시오.)

서버 인증을 사용하여 서버에 로그인 및 액세스합니다. 서버 인증은 액세스 그룹과 연관됩니다(서버 > 서버 인증 > 액세스).

서버에 대한 다음 서버 인증을 작성할 수 있습니다.

- dev/dev 서버 인증. 이 서버 인증은 서버에 대한 개발자 액세스 그룹과 연관시키십시오.
- qa/qa 서버 인증. 이 서버 인증은 QA 액세스 그룹과 연관시키십시오.
- prod/prod 서버 인증. 이 서버 인증은 빌드 액세스 그룹과 연관시키십시오.

qa/qa 서버 인증에 대한 액세스 권한이 있는 사용자가 prod/prod 서버 인증으로 단계를 실행할 수 있도록 하려면 액세스 불가능 서버 인증 실행 권한을 QA 액세스 그룹에 추가하십시오.

**참고:** 사용자가 서버에 대한 액세스 권한은 있지만 액세스 불가능 서버 인증 실행 권한을 통한 서버 인증에 대한 액세스 권한은 없는 경우 환경 변수 `_USE_BFCREDS`가 서버 인증을 대체하도록 설정된 경우에만 단계가 계속 실행됩니다.

---

## 선택기

이 주제는 선택기에 대해 설명합니다.

### 선택기 정보

선택기는 프로젝트 또는 단계를 실행할 서버 자원을 선택합니다.

선택기에는 변수 목록이 포함됩니다. 각 변수마다 값과 비교식을 지정합니다. 예를 들어, 특성 `CompilerVersion = 1.1`로 지정하면 해당 특성을 갖는 서버 자원만 선택할 수 있습니다. 또한 `CompilerVersion >= 1.1`을 지정하여 버전 1.1, 1.3, 2 또는 2.0을 갖는 서버 자원을 선택할 수 있습니다.

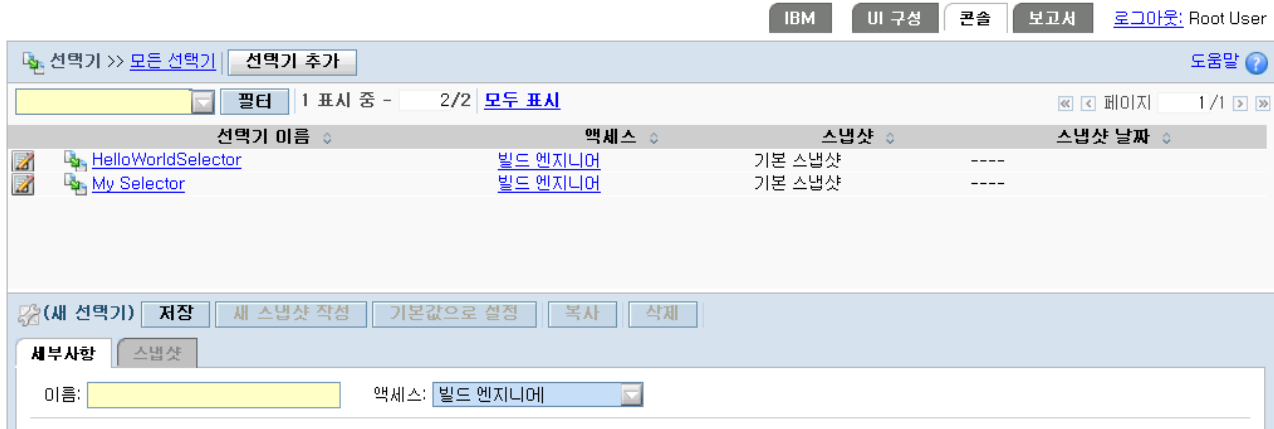
프로젝트 또는 단계가 실행될 때 지정된 선택기가 프로젝트 또는 단계가 실행되는 서버 자원을 결정합니다.

- 정적 선택기는 `BF_NAME` 변수를 사용하여 이름으로 서버 자원을 식별합니다.
- 동적 선택기는 선택기의 변수로 지정된 기준을 사용하여 시스템의 모든 서버 자원에서 서버 자원을 선택합니다.

### 선택기 패널 정보

프로젝트 또는 단계를 실행할 서버 자원을 선택하는 선택기를 작성 및 관리하려면 선택기 패널을 사용하십시오.

패널을 보려면 서버 > 선택기를 선택하십시오.



**세부사항 탭:** 세부사항 탭에 다음 필드가 있습니다.

**이름** 선택기의 이름입니다.

**액세스** 이 선택기를 사용할 수 있는 사용자의 액세스 그룹입니다.

**스냅샷 탭:** 선택기 스냅샷은 선택기의 인스턴스입니다. 스냅샷 탭에서는 스냅샷 이름을 제공합니다. 스냅샷 이름 및 스냅샷에 대한 주석을 보거나 변경하려면 이 탭을 사용하십시오.

## 선택기 설정 방법

선택기는 프로젝트 또는 단계에 적합한 서버 유형을 설명합니다. 선택기는 이름 또는 콜렉터가 수집하여 Manifest에 저장하는 특성으로 서버를 지정할 수 있으며 필수 및 선택적 특성을 포함하여 여러 특성을 지정할 수 있습니다.

선택기 설정에는 다음과 같은 여러 가지 방법을 사용할 수 있습니다.

- 서버 자원을 이름으로 선택하십시오. 선택기를 작성하고 서버 자원에 따라 선택기 이름을 지정하십시오. 선택기는 시스템에서 사용되는 고유 이름인 해당 BF\_NAME 값으로 서버 자원을 지정합니다. 프로젝트 또는 단계를 실행할 서버 자원을 지정하려면 이들 선택기 중 하나를 사용하십시오.
- 서버 풀로 서버를 선택하십시오. 서버를 이름 지정된 풀로 분류하고 각 풀마다 콜렉터를 작성할 수 있습니다. 풀 이름을 콜렉터의 특성(세트값 특성)으로 정의합니다. 그런 다음 각 풀 이름에 대한 선택기를 작성합니다. 프로젝트 또는 단계의 서버 자원은 현재 로드에서 선택됩니다.
- 서버 속성으로 서버를 선택하십시오. 기능 특성(예: 사용 가능한 하드 디스크 공간, 운영 체제 또는 CPU 수)을 기반으로 서버를 선택할 수 있습니다. 동적 선택을 구현하려면 다음을 수행하십시오.
  1. 적합한 특성을 수집하고 지정하는 콜렉터를 작성하십시오.
  2. 적합한 서버에 콜렉터를 지정하십시오.

3. 서버 선택사항을 나타내는 특성 세트 또는 각 특성에 대해 선택기를 작성하십시오.

예를 들어, 선택기를 작성하여 다음 기준에 따라 선택할 수 있습니다.

- "Windows®"를 포함하는 운영 체제가 있는 서버 자원
- CPU개 여러 개인 서버 자원
- 지정된 로드 미만으로 실행 중인 서버 자원
- 중첩된 콜렉터로 서버 자원을 선택하십시오. 유형 특성 포함을 사용하여 다른 콜렉터를 가리키는 콜렉터를 작성합니다. 콜렉터는 콜렉터 포인터 세트 구성될 수 있습니다. 예를 들어, 각 서버가 사용자가 지정하는 일부 고유 특성을 가질 수 있도록 각 서버마다 개별 콜렉터를 작성할 수 있습니다. 포함 유형을 사용하여 유틸리티 콜렉터를 가리킬 수 있습니다. 예를 들어, 환경 내 키 자원(예: Perl 및 Java)의 버전 번호를 지정하는 버전이라는 콜렉터를 작성할 수 있습니다.

## 선택기 변수 유형

선택기는 런타임에서 프로젝트 또는 단계에 대한 서버 자원을 선택하는 방법을 정의합니다.

선택기를 정의하기 위해 사용할 수 있는 변수 유형은 다음과 같습니다.

- 표준 특성: 이 유형의 변수에는 다음을 지정합니다.
  - 이름: 사용할 특성의 이름. 모든 기본 특성과 콜렉터에서 정의된 세트-값 특성으로 구성되는 자동 생성 목록에서 선택합니다.

**참고:** 세트-값 특성에서 선택기를 빌드하려면 먼저 하나 이상의 콜렉터에서 세트-값 특성을 정의해야 합니다.

- 연산자: 다음 비교 연산자 중 하나입니다.
  - EQ - 같음. 값이 정확히 일치해야 합니다. 값은 숫자 또는 문자열입니다.
  - NE - 같지 않음. 값은 지정된 값과 같지 않아야 합니다. 값은 숫자 또는 문자열입니다.
  - GT - 보다 큼.
  - GE - 크거나 같음.
  - LT - 보다 작음.
  - LE - 작거나 같음.
  - 포함.
- 값: 연산자가 비교하기 위해 사용하는 값.
- 필수: 선택기가 이 변수와 일치해야 하는 경우에는 예, 선택사항인 경우에는 아니오.

- 포함: 포함 유형을 사용하면 중첩을 통해 단순 선택기에서 빌드된 복잡한 선택기를 정의할 수 있습니다. 다음을 지정합니다.
  - 선택기: 포함할 선택기를 선택하십시오. 선택기로 지정된 모든 특성이 포함됩니다. 여러 단순 선택기를 포함하여 복잡한 선택기를 빌드할 수 있습니다.

## 선택기 변수 비교 규칙

시스템은 프로젝트나 단계에 사용할 서버 자원을 선택할 때 선택기 변수 값과 같은 이름의 Manifest 특성 값을 비교합니다.

시스템은 두 값 모두 다음 숫자 기준과 일치하지 않으면 문자열 비교를 수행합니다.

- 값이 숫자로 시작하고 뒤에 최소 하나의 숫자가 오는 숫자와 소수점만 포함하는 경우, 시스템은 숫자 비교를 수행합니다.
  - 5, 5.5, 0.5, 5.0 및 5.5.5는 숫자로 간주됩니다.
  - 5., .5, 5., 5..5, 5.4.6\_05 및 5.6i5는 모두 문자열로 간주됩니다.
- 숫자 값에 두 개 이상의 소수점이 포함된 경우 하위 버전 숫자 비교를 수행하는데, 이 경우 시스템은 소수점으로 분리된 각 필드를 비교합니다. 5.21은 5.3보다 낮은 반면(일반 숫자 비교), 5.21.0은 5.3보다 높습니다(하위 버전 숫자 비교).

**참고:** Contains 연산자에 대해서 시스템은 항상 대소문자 구분 문자열 비교를 수행합니다.

다음 테이블에서는 비교 규칙 적용 방법의 예제를 표시합니다.

특성 이름	Manifest 특성 값	연산자	선택기 변수 값	비교 유형	일치 여부
PerlVersion	v5.8.4	>=	5.2.1	문자열	예
PerlVersion	v5.8.4	>=	v.5.2.1	문자열	예
PerlVersion	v5.8.4	>=	v5.22.1	문자열	예
OS_VERSION	1.15	>=	1.1	숫자	예
OS_VERSION	1.10	>=	1.1.0	하위 버전 숫자	예
BF_NAME	WinServer1	contains	win	문자열	예
BF_NAME	Server123	=	123	문자열	아니오

## 올바른 서버 자원에 대한 선택기 평가

시스템은 프로젝트 또는 단계의 서버 자원을 선택하기 위해 다음과 같이 선택기를 사용하여 모든 올바른 서버 자원을 평가합니다.

1. 시스템은 선택기의 모든 필수 변수를 포함하는 서버의 목록을 컴파일합니다. 서버 자원이 필수 선택기 기준과 일치하는 경우 프로젝트 또는 단계가 실패하고 시스템에서 노트가 작성됩니다.

2. 둘 이상의 서버 자원이 필요한 기준을 충족시키는 경우 시스템은 이런 서버 자원 각각의 등급을 지정하고 다음과 같이 점수를 부여합니다.
  - 일치하는 각 선택적 변수에는 1점이 부여됩니다. 선택기에 동일한 변수의 여러 사본이 포함되는 경우에는 각 사본마다 1점이 부여됩니다.
  - 가장 낮은 BF\_LOADRATIO 값을 갖는 서버 자원에는 1점이 부여됩니다.
3. 시스템은 가장 많은 점수를 받은 서버를 선택합니다. 둘 이상의 서버 자원이 가장 많은 점수를 받은 경우 시스템이 그 중에서 선택합니다.

선택기에서 선택적 변수를 반복할 수 있습니다. 이렇게 하면 일치하는 서버의 범위가 증가합니다. 예를 들어, 1GB 메모리가 필요하지만 2GB 이상이 강력하게 권장되는 경우 다음과 같이 선택기 변수를 설정할 수 있습니다.

- 필수 MEM\_TOTAL GE 1GB를 지정하십시오.
- 선택적 MEM\_TOTAL GE 2GB를 세 번 지정하십시오.

---

## 컬렉터

### 컬렉터 정보

컬렉터는 수집되고 서버 자원에 지정되는 특성을 정의합니다.

서버 패널의 컬렉터 섹션은 사용 가능한 컬렉터를 나열하고 새 컬렉터를 작성할 수 있게 합니다.

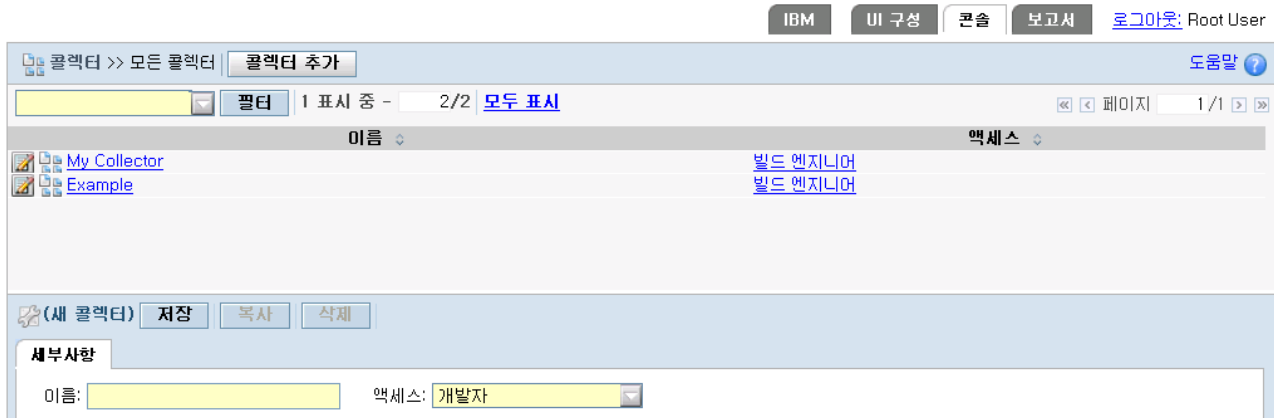
서버 패널에 대한 컬렉터 섹션을 열려면 **서버 > 컬렉터**를 클릭하십시오.

컬렉터는 하나 이상의 특성으로 구성됩니다. 각 특성은 Manifest에 포함시킬 정보를 지정합니다.

**참고:** 컬렉터가 있으면 동적 선택기를 사용할 수 있습니다. 컬렉터를 사용하려면 Enterprise Edition이 필요합니다.

### 컬렉터 패널 정보

컬렉터를 작성, 편집 및 삭제하려면 컬렉터 패널을 사용하십시오. 패널을 보려면 **서버 > 컬렉터**를 선택하십시오.



패널에 다음 필드가 있습니다.

**이름** 컬렉터의 이름입니다.

**액세스** 이 컬렉터를 사용할 수 있는 사용자의 액세스 그룹입니다.

## 컬렉터 작성

프로젝트에 대한 서버 자원의 동적 선택을 사용 가능하게 하려면 컬렉터를 작성하십시오.

### 프로시저

1. 서버 > 컬렉터를 선택하십시오.
2. 컬렉터의 이름을 지정하십시오.
3. 컬렉터의 액세스 권한을 지정하십시오.
4. 저장을 클릭하십시오. 패널은 수집할 데이터를 판별하는 변수를 입력할 수 있도록 변경됩니다.
5. 하나 이상의 변수를 작성하고 저장하십시오. 변수 특성에 대한 정보는 『컬렉터 변수 특성 유형』을 참조하십시오.

### 컬렉터 변수 특성 유형

컬렉터 변수 특성은 서버 Manifest에 대한 정보 수집 방법을 지정합니다.

컬렉터에 다음 유형의 특성을 정의할 수 있습니다.

#### 값 설정

이 특성은 이름 지정된 정적 값을 서버에 지정합니다. 특성 이름과 값을 지정합니다.

사전 정의된 응답을 가져오기 위해 값에서 특수 값을 사용할 수 있습니다. 특수 값은 밑줄(\_) 문자로 시작합니다. 314 페이지의 『컬렉터의 특수 특성』의 내용을 참조하십시오.



**빌트인** 이 특성은 서버 자원에 지정된 호스트에 대한 정보를 리턴합니다. 빌트인 특성 목록은 309 페이지의 『빌트인 특성』의 내용을 참조하십시오.

#### 실행 명령

이 유형의 특성은 시스템이 실행할 명령을 지정합니다. 명령 실행의 결과물에 특성 값이 설정됩니다. 기본적으로 출력의 처음 255자가 사용됩니다. 정규 표현식을 사용하여 출력에서 지정된 부분을 추출할 수 있습니다.

- **특성:** 특성의 이름
- **명령:** 서버 자원에 지정된 호스트에서 실행할 명령
- **정규 표현식:** 선택적이며, 출력을 필터하기 위해 사용하는 정규 표현식입니다. 지정된 경우, 빌드 시스템은 명령 출력의 각 행과 정규 표현식을 일치시키려고 시도합니다. 행이 처음 일치할 때 시스템은 \$1 값(Perl 규약)을 검색하고 이를 해당 특성 값으로 사용합니다. 정규 표현식은 값을 리턴하도록 최소 한 세트의 괄호를 포함해야 합니다. Perl 정규식 생성에 대한 자세한 정보는 Perl 문서를 참조하십시오.

**포함** 이 유형의 특성은 콜렉터 목록을 지정합니다. 콜렉터를 중첩할 수 있습니다. 포함 유형의 콜렉터를 작성할 때 다른 콜렉터의 이름을 값으로 지정합니다. 빌드 시스템이 Manifest를 작성하거나 업데이트할 때 시스템은 참조된 콜렉터에서 특성을 삽입합니다.

**팁:** 시스템은 콜렉터에 나열된 순서로 콜렉터 변수 특성을 적용하여 동일한 이름의 나중 특성이 이전 특성을 대체하도록 합니다. 다른 콜렉터에 한 콜렉터를 포함시키는 경우 이 기능을 사용하십시오. 콜렉터의 변수 특성을 전체가 아닌 일부만 사용하려는 경우 사용하지 않으려는 변수 특성을 대체하십시오.

또한 시스템은 몇 가지 특성(예: 서버의 논리 이름을 포함하는 BF\_NAME 특성)을 자동으로 적용합니다. 이러한 특성은 특수 Manifest 특성의 일부로 고려합니다. 313 페이지의 『사전 설정 특성』의 내용을 참조하십시오.

---

## Manifest 및 동적 서버 선택

콜렉터, Manifest 및 선택기를 함께 사용하여 런타임에서 프로젝트 또는 단계에 대한 서버 자원을 선택할 수 있습니다.

다음 세 개의 서로 다른 데이터 오브젝트는 시스템이 동적으로 서버를 선택할 수 있게 합니다.

- **콜렉터**는 시스템이 서버 자원에서 수집하거나 서버 자원에 지정하는 특성 세트를 정의하는 오브젝트입니다. 시스템은 서버 자원의 특성을 확인할 때 콜렉터를 실행합니다. 수집된 특성 값은 Manifest에 저장됩니다.
- **Manifest**는 특정 서버의 특성 목록입니다. Manifest에는 콜렉터 실행 결과가 포함됩니다.

- 선택기는 특성과 비교식의 목록입니다(예: MEM\_TOTAL = 512). 시스템은 선택기 특성과 Manifest를 비교하여 서버가 특정 선택기 요구사항을 충족하는지 여부를 확인할 수 있습니다. 프로젝트와 단계는 선택기를 해당 특성 중 하나로 지정합니다. 프로젝트 또는 단계가 실행될 때 실행될 서버 자원을 선택하기 위해 정의된 모든 서버 자원의 Manifest와 선택기가 비교됩니다.

다음 예제는 단순 선택기 작성 및 사용 방법을 보여줍니다.

1. Mercury라는 서버 자원을 작성하고 에이전트와 연관시키십시오.
2. Mercury라는 선택기를 작성하십시오. BF\_NAME = Mercury로 서버를 선택하도록 설정하십시오.
3. Lincoln이라는 프로젝트를 작성하십시오. 프로젝트에 Mercury 선택기를 지정하십시오.

Lincoln 프로젝트를 실행하면 시스템에서 Mercury라는 서버 자원을 선택합니다. 해당 서버 자원을 사용할 수 없으면 프로젝트가 실패합니다.

다음 예제는 서버 세트에서 동적 서버 선택사항을 설정하는 방법을 보여줍니다.

1. RAMSIZE라는 콜렉터를 작성하십시오. 빌트인 특성 MEM\_TOTAL을 수집하도록 콜렉터를 설정하십시오.
2. 호스트와 연관시킬 서버 자원을 작성하십시오. 각 자원이 RAMSIZE 콜렉터를 사용하도록 설정하십시오.
  - Mercury(RAM에 512MB인 호스트)
  - Mars(RAM이 1GB인 호스트)
  - Jupiter(RAM이 3GB인 호스트)
3. BigRam이라는 선택기를 작성하십시오. 표준 특성, 특성=MEM\_TOTAL, 연산자=GE(크거나 같음) 및 값=2048을 선택하도록 설정하십시오. MEM\_TOTAL은 MB 단위로 표시됩니다. 이 선택기는 RAM이 2GB 이상인 호스트만 선택합니다.
4. SmallRam이라는 선택기를 작성하십시오. 표준 특성인 property=MEM\_TOTAL, Operator=GE(크거나 같음) 및 Value=256을 선택하도록 설정하십시오. 이 선택기는 256MB 이상의 RAM을 갖는 호스트만 선택합니다.
5. 다음 두 프로젝트를 작성하십시오.
  - HighMaint: BigRam 선택기를 사용하려면 이 프로젝트를 설정하십시오.
  - LowMaint: SmallRam 선택기를 사용하려면 이 프로젝트를 설정하십시오.

HighMaint를 실행하면 시스템은 Jupiter 서버를 선택합니다. 이 서버는 2GB 이상의 RAM을 갖는다는 선택기 요구사항을 충족시키는 유일한 서버이기 때문입니다.

LowMaint를 실행하면 시스템에서 사용 가능한 세 개 서버 자원을 선택합니다.

나중에 2GB의 RAM을 갖는 호스트에 대해 Neptune이라는 서버 자원을 추가한 후 다음 HighMaint 프로젝트가 실행되면 Neptune 또는 Jupiter 중 하나가 프로젝트에 대해 선택됩니다. 어떠한 이유로 Jupiter 작동이 중지되면 선택기에 적합한 유일한 서버인 Neptune이 선택됩니다.

## Manifest 보기

서버에 대한 Manifest를 보려면 다음을 수행하십시오.

1. 서버를 클릭하십시오.
2. 서버 자원의 이름을 클릭하십시오.
3. **Manifest** 탭을 클릭하십시오.

서버의 Manifest를 직접 변경할 수는 없습니다. Manifest 콘텐츠는 콜렉터가 정의합니다.

Manifest는 서버 특성 또는 시스템 설정에서 결정되는 간격에 따라 자동으로 새로 고쳐집니다. 간격 사이에서 새로 고치려면 큐 **Manifest** 새로 고치기를 클릭하십시오.

## Manifest를 수동으로 새로 고치기

Manifest를 수동으로 새로 고치려면 다음을 수행하십시오.

1. 서버를 클릭해서 서버 목록을 표시하십시오.
2. 서버를 선택하십시오.
3. 큐 **Manifest** 새로 고치기를 클릭하십시오.

서버 연결 테스트가 수행되고 서버 Manifest가 업데이트됩니다. 새로 고치기가 완료될 때까지 **Manifest** 새로 고치기 진행 표시줄이 **Manifest** 탭에 표시됩니다.

자동 서버 Manifest 새로 고치기 간격은 서버 테스트 빈도 시스템 설정에 의해 설정됩니다. 큐에서 수동 Manifest 새로 고치기는 큐의 자동 Manifest 새로 고치기 전에 수행됩니다.

## 서버 Manifest의 업데이트 빈도 설정

시스템 설정을 사용하여 시스템이 서버에서 Manifest 특성을 조회하는 빈도를 제어할 수 있습니다.

### 이 태스크 정보

기본적으로 시스템은 명령 실행 특성보다 내장된 특성과 값 설정 특성을 더 자주 확인합니다.

콜렉터에서 이름 `_AGE`를 갖는 변수를 정의하여 해당 콜렉터를 사용하는 서버가 Manifest 특성에 대해 조회되는 빈도를 제어할 수 있습니다. 기본값은 86400으로, 하

루에 한 번 업데이트를 제공합니다. 명령 실행 특성을 통해 검색한 값이 자주 변경되지 않는 경우에는 이 설정보다 낮은 값을 설정하지 마십시오. 이 변수를 작성하려면 다음을 수행하십시오.

## 프로시저

1. 서버 > 콜렉터를 클릭하십시오.
2. 기존 콜렉터를 선택하거나 새 콜렉터를 작성하십시오.
3. 콜렉터에 변수를 추가하십시오.
  - a. 값 설정 유형을 선택하십시오.
  - b. 변수 이름 \_AGE를 입력하십시오.
  - c. 값을 초 단위로 지정하십시오.
  - d. 저장 단추를 클릭하십시오.
4. 필요한 경우, 두 개 이상의 서버가 방금 편집한 콜렉터와 일치하도록 콜렉터 특성을 변경하십시오.

## 정적 및 동적 서버 선택 설정 예제

다음 예제는 단순 정적 선택기를 작성하고 사용하는 방법을 보여줍니다.

1. Mercury라는 서버 자원을 작성하고 에이전트와 연관시키십시오.
2. Mercury라는 선택기를 작성하십시오. 표준 특성인 property=BF\_NAME, Operator=EQ(같음), Value=Mercury 및 Required를 선택하도록 설정하십시오.
3. Lincoln이라는 프로젝트를 작성하십시오. 프로젝트에 Mercury 선택기를 지정하십시오.

Lincoln 프로젝트를 실행하면 시스템에서 Mercury라는 서버 자원을 선택합니다. 해당 서버 자원을 사용할 수 없으면 프로젝트가 실패합니다.

다음 예제는 서버 세트에서 동적 서버 선택사항을 설정하는 방법을 보여줍니다.

1. RAMSIZE라는 콜렉터를 작성하십시오. 빌트인 특성 MEM\_TOTAL을 수집하도록 콜렉터를 설정하십시오.
2. 호스트와 연관시킬 서버 자원을 작성하십시오. 각 자원이 RAMSIZE 콜렉터를 사용하도록 설정하십시오.
  - Mercury(RAM에 512MB인 호스트)
  - Mars(RAM이 1GB인 호스트)
  - Jupiter(RAM이 3GB인 호스트)
3. BigRam이라는 선택기를 작성하십시오. 표준 특성인 property=MEM\_TOTAL, Operator=GE(크거나 같음), Value=2048 및 Required를 선택하도록 설정하십시오. MEM\_TOTAL은 MB 단위로 표시됩니다. 이 선택기는 RAM이 2GB 이상인 호스트만 선택합니다.

4. SmallRam이라는 선택기를 작성하십시오. 표준 특성인 property=MEM\_TOTAL, Operator=GE(크거나 같음) 및 Value=256을 선택하도록 설정하십시오. 이 선택기는 256MB 이상의 RAM을 갖는 호스트만 선택합니다.
5. 다음 두 프로젝트를 작성하십시오.
  - HighMaint: BigRam 선택기를 사용하려면 이 프로젝트를 설정하십시오.
  - LowMaint: SmallRam 선택기를 사용하려면 이 프로젝트를 설정하십시오.

HighMaint를 실행하면 시스템은 Jupiter 서버를 선택합니다. 이 서버는 2GB 이상의 RAM을 갖는다는 선택기 요구사항을 충족시키는 유일한 서버이기 때문입니다.

LowMaint를 실행하면 시스템에서 사용 가능한 세 개 서버 자원을 선택합니다.

나중에 2GB의 RAM을 갖는 호스트에 대해 Neptune이라는 서버 자원을 추가한 후 다음 HighMaint 프로젝트가 실행되면 Neptune 또는 Jupiter 중 하나가 프로젝트에 대해 선택됩니다. 어떠한 이유로 Jupiter 작동이 중지되면 선택기에 적합한 유일한 서버인 Neptune이 선택됩니다.

## 특성 참조

다음 주제는 Manifest에 영향을 주는 특성에 대해 설명합니다.

- 『빌트인 특성』
- 313 페이지의 『사전 설정 특성』
- 314 페이지의 『컬렉터의 특수 특성』

### 빌트인 특성

관리 콘솔은 서버에서 빌트인 특성을 수집하여 서버 Manifest에 값을 지정합니다.

### 빌트인 특성

빌드 시스템의 서로 다른 여러 데이터 오브젝트에서 빌트인 특성을 사용합니다.

- 선택기는 빌트인 특성을 선택기 변수로 사용하여 서버를 해당 특성의 특정 값과 일치시킬 수 있습니다.
- 컬렉터는 빌트인 특성을 사용하여 서버에서 데이터를 수집합니다.
- **Manifest**는 빌트인 특성이 수집되면 빌트인 특성 값을 저장합니다.

빌트인 특성은 자동으로 추가되지 않습니다. 빌트인 특성이 Manifest에 표시되게 하려면 빌트인 특성을 컬렉터에 추가해야 합니다. 이 테이블에서는 빌트인 특성을 나열하고 설명합니다.

**참고:** 특성의 가용성은 플랫폼에 따라 다릅니다.

표 8. 콜렉터 및 Manifest의 빌트인 특성

특성	설명
CPU_ARCH	리턴된 값은 아키텍처 이름의 label이며 다음과 같이 표시됩니다. <ul style="list-style-type: none"> <li>• HP-PA: HP Precision Architecture</li> <li>• IA-64: Intel Itanium</li> <li>• MVS: IBM S/390</li> <li>• PPC: PowerPC</li> <li>• PPC-64: PowerPC 64</li> <li>• SPARC: Sun SPARC</li> <li>• X86: Intel, AMD, Cyrix 및 기타에서 사용되는 x86 호환 아키텍처</li> </ul>
CPU_LOAD(Windows만 해당)	CPU 로드 또는 CPU 사용량이 용량의 백분율(0-100 사이)로 표시됩니다.
CPU_LOAD1	리턴된 값은 실행 중이거나 지난 1분간 실행 대기 중인 평균 프로세스 수(로드 평균)를 보고합니다.  CPU_LOAD1은 CPU 활동의 측정값입니다. 대기 컴퓨터의 로드 수는 0입니다. CPU를 사용 중이거나 대기 중인 각 프로세스는 1씩 로드 수에 추가됩니다. 참고: Windows의 경우 활성 또는 비활성의 모든 프로세스가 로드 수에 추가됩니다. 또한 이 정보는 bfdispach 프로세스에서 수집되며, 공유 메모리 세그먼트를 사용하여 에이전트에 공개됩니다. 에이전트에 연결하기 위해 사용되는 사용자 신임이 특권이 없으면 이러한 통계가 사용 가능하지 않습니다.
CPU_LOAD5	리턴된 값은 지난 5분간 실행 대기 중인 평균 프로세스 수(로드 평균)를 보고합니다.  CPU_LOAD5는 CPU 활동의 측정값입니다. 대기 컴퓨터의 로드 수는 0입니다. CPU를 사용하고 있거나 실행될 각 프로세스는 1씩 로드 수에 추가됩니다. 참고: Windows의 경우 활성 또는 비활성의 모든 프로세스가 로드 수에 추가됩니다. 또한 이 정보는 bfdispach 프로세스에서 수집되며, 공유 메모리 세그먼트를 사용하여 에이전트에 공개됩니다. 에이전트에 연결하기 위해 사용되는 사용자 신임이 특권이 없으면 이러한 통계가 사용 가능하지 않습니다.
CPU_LOAD15	지난 15분간 실행 대기 중인 평균 프로세스 수(로드 평균)입니다.  CPU_LOAD15는 CPU 활동의 측정값입니다. 대기 컴퓨터의 로드 수는 0입니다. CPU를 사용하고 있거나 실행될 각 프로세스는 1씩 로드 수에 추가됩니다. 참고: Windows의 경우 활성 또는 비활성의 모든 프로세스가 로드 수에 추가됩니다. 또한 이 정보는 bfdispach 프로세스에서 수집되며, 공유 메모리 세그먼트를 사용하여 에이전트에 공개됩니다. 에이전트에 연결하기 위해 사용되는 사용자 신임이 특권이 없으면 이러한 통계가 사용 가능하지 않습니다.
CPU_MHZ	이 특성은 프로세서 속도(MHz)를 보고합니다. 이 특성 정보를 입력하려면 특정 조건이 충족되어야 합니다. <ul style="list-style-type: none"> <li>• Linux: 주파수 조정이 사용 가능해야 합니다.</li> <li>• Windows: ~MHz 레지스트리 항목이 존재하고 정보가 입력되어 있어야 합니다.</li> <li>• x86 및 x86-64 프로세서: 인라인 어셈블리가 작동해야 합니다.</li> </ul>

표 8. 콜렉터 및 Manifest의 빌트인 특성 (계속)

특성	설명
CPU_MANUFACTURER	<p>이 특성은 프로세서 제조업체의 회사 이름을 리턴합니다. 정보를 직접 사용할 수 없는 경우 이름은 아키텍처를 기본으로 가정됩니다. 사용 가능한 프로세서 정보가 충분하지 않은 경우 값이 리턴되지 않습니다. 다음 값이 지원됩니다.</p> <ul style="list-style-type: none"> <li>• AMD: 해당 x86 및 AMD64 프로세서의 경우</li> <li>• Cyrix: 해당 x86 호환 프로세서의 경우</li> <li>• DEC: Alpha 및 VAX의 경우</li> <li>• HP: Hewlett-Packard Precision Architecture</li> <li>• IBM: IBM S/390 및 PowerPC G5</li> <li>• Intel: Intel x86(Intel64 포함), IA-64 Itanium</li> <li>• Motorola: PowerPC G4</li> <li>• NexGen: x86 호환 프로세서</li> <li>• National: National Semiconductor x86 호환 프로세서</li> <li>• Rise: Rise x86 호환 프로세서</li> <li>• Sis: Sis x86 호환 프로세서</li> <li>• Sun: Sun Microsystems SPARC</li> <li>• TransMeta: TransMeta x86 호환 프로세서</li> <li>• UMC: UMC x86 호환 프로세서</li> <li>• VIA: VIA Technologies x86 호환 프로세서</li> </ul>
CPU_MODEL	<p>이 특성은 제조업체 특정 CPU 모델 번호를 리턴합니다. 이 값은 다음과 같이 보고됩니다.</p> <ul style="list-style-type: none"> <li>• x86 아키텍처 <ul style="list-style-type: none"> <li>– 386</li> <li>– 486</li> <li>– 586</li> <li>– 686</li> <li>– X86_64</li> </ul> </li> <li>• PowerPC 아키텍처 <ul style="list-style-type: none"> <li>– 6xx</li> <li>– POWER</li> <li>– RS64</li> <li>– G3</li> <li>– G4</li> <li>– G5</li> <li>– Cell</li> </ul> </li> </ul>



표 8. 콜렉터 및 Manifest의 빌트인 특성 (계속)

특성	설명
CPU_SERIAL	이 특성은 CPU 또는 컴퓨터의 일련 번호를 리턴합니다. 현재 이 기능은 다음 아키텍처로 제한됩니다. <ul style="list-style-type: none"> <li>x86: Intel 또는 Transmeta 일련 번호만 해당합니다. 참고: 대부분의 x86 프로세서는 일련 번호를 보고하지 않습니다. 해당 경우에는 값이 리턴되지 않습니다.</li> <li>MacOS/X: 지정된 일련 번호가 I/O 레지스트리에서 검색됩니다. 특성을 보고하려면 CoreFoundation 및 IOKit 프레임워크를 찾을 수 있어야 합니다.</li> </ul>
DISK_FREE	UNIX 및 Linux의 경우, 이 특성은 서버 경로가 지정하는 파일 시스템상의 여유 공간(MB)의 양을 리턴합니다.  Windows의 경우, 서버 경로에서 지정하는 드라이브상의 디스크 여유 공간(MB)입니다.  예를 들어, 4GB의 여유 공간을 가진 디스크는 4096MB로 보고됩니다.
DISK_TOTAL	이 특성은 사용 가능한 총 디스크 여유 공간을 리턴합니다. 이 값은 남아 있는 전체 디스크 또는 파티션보다 작은 독립된 할당이 있을 수 있는 에이전트의 기본 경로에 대해 보고됩니다. 디스크 공간 관리는 운영 체제에 따라 다릅니다.
MEM_LOAD(UNIX/Linux만 해당)	UNIX 및 Linux의 경우, 현재 사용 중인 시스템 메모리 또는 RAM의 양이 전체 실제 메모리의 백분율(0 - 100 사이)로 표시됩니다.
MEM_FREE	이 특성은 RAM 또는 시스템 메모리의 양(MB)을 리턴합니다. 예를 들어, 2GB의 사용 가능 RAM은 2048MB로 보고됩니다.
MEM_PAGESIZE	이 특성은 RAM 또는 시스템 메모리 페이지 크기(MB)를 리턴합니다. 이 그림은 호스트 시스템의 표준 페이지 크기를 나타냅니다. 예를 들어, 페이지 크기가 4KB인 호스트 시스템은 4096MB로 보고됩니다.
MEM_TOTAL	이 특성은 전체 RAM 또는 시스템 메모리(MB)를 리턴합니다.  예를 들어, 2GB의 RAM을 가진 컴퓨터는 2048MB로 보고됩니다.
NET_FQDN	이 특성은 에이전트를 실행 중인 컴퓨터의 완전한 도메인 이름(FQDN)을 리턴합니다. FQDN은 에이전트가 통신하는 데 사용하는 주소를 기반으로 보고됩니다. 리턴되는 주소는 실제로 사용 중인 주소에 따라 IPv4 또는 IPv6일 수 있습니다. NET_IPV, NET_IPV4 및 NET_IPV6도 참조하십시오.
NET_HWADDR	이 특성은 NET_IFACE에서 보고된 인터페이스의 하드웨어 주소를 리턴합니다.
NET_IFACE	이 특성은 에이전트가 통신하기 위해 사용하는 인터페이스의 이름을 리턴합니다. <ul style="list-style-type: none"> <li>Windows: 보고되는 이름은 ipconfig 명령이 리턴하는 이름입니다(예: Intel(R) PRO/100 VE Network Connection - Packet Scheduler Miniport).</li> <li>기타 운영 체제에서 보고되는 이름은 ifconfig가 리턴하는 이름입니다(예: en0 또는 eth0 또는 OSA1).</li> </ul>
NET_IPV	이 특성은 에이전트와 통신하기 위해 사용되는 IP 연결의 유형(IPv4의 경우 4이고 IPv6의 경우 6)을 리턴합니다.
NET_IPV4	이 특성은 에이전트가 통신하기 위해 사용하는 IPv4 주소를 리턴합니다. IPv6를 통한 연결에서 에이전트가 동일한 인터페이스에 대해 IPv4 주소를 식별할 수 있는 경우, 해당 주소가 보고됩니다.
NET_IPV6	이 특성은 에이전트가 통신하기 위해 사용하는 IPv6 주소를 리턴합니다.
NET_SPEED(Windows만 해당)	이 특성은 인터페이스의 속도(Mb/초)(예: Gigabit Ethernet의 경우 1000)를 리턴합니다.
NUM_CPU	이 특성은 컴퓨터의 CPU 수를 리턴합니다.
OS_HOSTID	이 특성은 gethostid() 시스템 호출의 결과를 리턴합니다. 이 결과는 일반적으로 시스템 관리자가 /etc/hostid를 정보 값으로 설정한 경우를 제외하고는 정보를 제공하지 않습니다.



표 8. 콜렉터 및 Manifest의 빌트인 특성 (계속)

특성	설명
OS_SYSNAME	이 특성은 서버의 운영 체제 이름을 리턴합니다. 예제: Microsoft Windows XP, AIX, Macintosh OS.
OS_RELEASE	이 특성은 서버의 운영 체제 릴리스 레벨을 리턴합니다.  예를 들어 서버 운영 체제가 Microsoft XP 버전 5.1.2600이면 이 리턴값은 5입니다.
OS_VERSION	이 특성은 서버의 운영 체제 버전을 리턴합니다.  예를 들어 서버 운영 체제가 Microsoft XP 버전 5.1.2600이면 이 리턴값은 1입니다.
WIN_SERVICEPACK(Windows 만 해당)	이 특성은 서버에 설치된 Windows 서비스 팩의 버전 번호를 리턴합니다. 예를 들어, 서비스 팩 2의 경우 이 값은 2입니다.

## 사전 설정 특성

일부 Manifest 특성은 자동으로 설정됩니다.

다음 Manifest 특성은 자동으로 설정됩니다. 빌트인 특성과 달리, 이들 특성은 데이터를 채우기 위해 콜렉터에 추가하지 않아도 됩니다.

테이블 설명에 선택기로 표시된 특성을 선택기에서 사용할 수 있습니다. 기타 특성은 Manifest에 정보를 제공할 경우에만 사용됩니다.

표 9. 자동으로 설정된 Manifest 특성

특성	설명
BF_AGENT_VERSION	선택기. 서버에 설치되어 있는 에이전트의 버전 번호입니다.
BF_EXCLUSIVE	선택기. 연산자나 값이 없는 플래그입니다. 선택기에 이 특성이 포함된 경우 선택된 서버의 모든 슬롯이 작업 지속 기간 동안 예약됩니다.  작업의 단계에서 실행할 다른 서버를 지정하는 경우 현재 서버의 모든 현재 슬롯은 다른 서버에서 단계가 실행되는 동안 계속 예약된 상태입니다.
BF_JOBS	선택기. 서버 자원이 동시에 실행 중인 작업(단계) 수입니다. 이 값은 기타 Manifest 특성 업데이트와 관계없이 콘솔이 서버에 단계를 지정할 때마다 업데이트됩니다.
BF_LAST_REFRESH	Manifest에 빌트인 특성의 마지막 업데이트 날짜의 시간입니다. 값은 UNIX <sup>®</sup> 시간소인: 1970년 1월 1일 이후 시간(초)으로 보고됩니다.
BF_LASTJOBS	Manifest를 마지막으로 새로 고쳤을 때 서버에서 실행되던 작업 수입니다.
BF_LAST_UPDATE	Manifest에 명령 실행 특성의 마지막 업데이트 날짜의 시간입니다. 값은 UNIX <sup>®</sup> 시간소인과 같이 1970년 1월 1일 이후 시간(초)로 보고됩니다.
BF_LOADRATIO	선택기. 계산 값은 작업 수(BF_JOBS)를 서버에 허용되는 최대 작업 수(Max Jobs 설정)로 나눈 비율로 보고됩니다. 1개 작업을 실행 중이고 최대 작업 수가 4인 서버는 로드 비율이 .25입니다.

표 9. 자동으로 설정된 Manifest 특성 (계속)

특성	설명
BF_NAME	선택기. 실행할 서버 자원을 지정합니다. 값은 서버 자원의 이름입니다. BF_NAME 특성은 Manifest 목록에 표시되지 않습니다.
BF_RESERVE	<p>선택기. 연산자나 값이 없는 플래그입니다. 선택기에 이 특성이 포함되어 있으면, 선택된 서버에서 슬롯이 작업 지속 기간 동안 예약됩니다.</p> <ul style="list-style-type: none"> <li>작업의 단계가 실행할 다른 서버를 지정하는 경우, 기타 서버가 단계를 실행하는 동안 선택된 서버에서 슬롯이 계속 예약됩니다.</li> <li>단계가 선택된 서버를 명시적으로 지정하는 경우, 서버는 해당 단계에 예약된 슬롯을 사용합니다.</li> </ul> <p>이 플래그는 하나 이상의 단계가 다른 서버에서 실행될 때 프로젝트가 서버에서 해당 슬롯을 유실하는 경우에 발생하는 프로젝트 지연을 방지합니다.</p>

## 컬렉터의 특수 특성

일부 값 설정 특성은 컬렉터에서 이름 지정될 때 시스템에서 특정 동작을 일으킵니다. 이 특성은 밑줄 문자로 시작합니다. 빌드 시스템은 이 특성 값을 사용하여 컬렉터에서 이 특성을 획득하는 서버에 동작을 적용합니다.

**참고:** 문자열 "BF\_"로 시작하는 이름은 시스템이 사용하도록 예약되어 있기 때문에 그런 이름의 특성을 작성할 수 없습니다.

표 10. 컬렉터 및 Manifest의 특수 값 설정 특성

특성	설명
_AGE	이 특성은 Manifest를 새로 고쳐야 하는 빈도를 초 단위로 정의합니다. 기본값 86400은 하루에 한 번 새로 고치기를 제공합니다. 값 3600을 사용하면 시스템이 매시간 Manifest를 업데이트하게 됩니다.

## 스냅샷을 사용하여 선택기의 새 인스턴스 작성

선택기 스냅샷을 찍어 변경 또는 수정할 선택기의 새 인스턴스를 빠르게 작성합니다.

### 선택기 스냅샷 개요

이 주제를 검토하면 선택기 스냅샷에 대해 알게 되며 환경 스냅샷 사용 방법을 이해할 수 있습니다.

### 선택기 스냅샷 유스 케이스

다음 예제는 선택기 스냅샷의 일부 공통 유스 케이스에 대해 설명합니다.


- 선택기 스냅샷을 찍어 선택기 구성을 변경하거나 기존 선택기와 함께 작업을 계속 실행하면서 새 도구 또는 스크립트 테스트를 수행합니다.
- 선택기의 스냅샷을 임시 백업으로 또는 공식 아카이브의 일부로 저장하십시오.
- 선택기 스냅샷을 찍어 외부 또는 내부 릴리스와 같은 마일스톤과 일치하는 적시 선택기 구성을 캡처합니다.

## 선택기 스냅샷 개념 및 용어

스냅샷은 선택기에 대한 작업을 위한 몇 가지 새 개념과 용어를 소개합니다.

**선택기 스냅샷:** 스냅샷은 기존 선택기의 새 인스턴스입니다. 스냅샷에 대해 기억해야 할 몇 가지 중요 사항은 다음과 같습니다.


- 스냅샷은 별도 선택기 오브젝트입니다. 스냅샷 세트의 특정 스냅샷을 변경해도 세트의 다른 스냅샷에 영향을 주지 않습니다.
- 스냅샷은 사본이 아닙니다. 선택기와 연관된 오브젝트의 스냅샷을 찍으면 스냅샷이 오브젝트의 별도 인스턴스를 작성합니다. 사본은 오브젝트 간 관계를 맵핑하며 새 오브젝트를 작성하지 않습니다.
- 스냅샷은 선택기의 변경내용이 아닙니다.
  - 스냅샷은 두 선택기 스냅샷 간의 변경사항 비교를 지원하지 않습니다.
  - 선택기 스냅샷 변경사항은 소스 제어 시스템에서와 같이 버전 번호로 추적 또는 식별되지 않습니다. 그러나 버전 번호를 포함하는 스냅샷 이름 지정 구조(예: 7.5.0, 3.4.01)를 사용하여 선택기 스냅샷과 마일스톤을 상관시킬 수 있습니다.

**스냅샷 세트:** 스냅샷 세트는 특정 기본 스냅샷의 하위인 모든 선택기 스냅샷의 세트입니다. 최소한 세트에는 기본 또는 상위 스냅샷과 하위 스냅샷이 포함됩니다. UI에서 선택기 이름 옆의 스냅샷 아이콘()은 선택기에 대한 스냅샷 세트가 존재함을 나타냅니다.


**기본 스냅샷:** 처음에는 모든 선택기의 스냅샷 이름이 기본 스냅샷입니다. 기본 스냅샷을 다른 이름으로 변경할 수 있습니다. 기본 스냅샷은 스냅샷 세트의 상위입니다.

**기본 선택기 스냅샷:** 기본 선택기 스냅샷은 현재 실행 중인 선택기입니다. 세트에서 하나의 스냅샷만 기본값이 될 수 있습니다. 기본 스냅샷을 지정하지 않는 경우 기본값은 기본 스냅샷입니다.

- 콘솔에서 기본 스냅샷이 선택기 목록의 최상위 레벨에 표시됩니다. 선택기 목록을 표시하려면 서버 > 선택기를 선택하십시오.
- 스냅샷이 있는 선택기를 선택하면 목록 상자에서 다른 선택기 스냅샷을 선택하지 않는 한 기본 선택기 스냅샷이 사용됩니다.

- 선택기 스냅샷 세트의 다른 스냅샷에 액세스하여 작업을 수행하려면 스냅샷 아이콘()을 클릭해야 합니다.

## 선택기 스냅샷 보기

스냅샷 아이콘()을 선택하여 스냅샷 보기를 표시합니다. 스냅샷 보기는 세트에 있는 스냅샷의 계층 구조를 보여줍니다.

- 기본 스냅샷은 최상위 레벨이며 고유 이름을 지정하지 않는 경우 기본 스냅샷이라고 합니다.
- 모든 선택기 스냅샷은 기본 스냅샷의 하위입니다. 기본 스냅샷이 같은 하위는 스냅샷 열의 동일한 레벨에서 들어옵니다.
- 하위 스냅샷에서 작성된 선택기 스냅샷은 하위 스냅샷의 하위가 되며 스냅샷 열의 다음 레벨에서 들어옵니다.

## 선택기 스냅샷 계획

기본 선택기 스냅샷을 선택하고 선택기 스냅샷의 이름을 지정하는 몇 가지 우수 사례를 검토하십시오.

- 세트의 기본 스냅샷 선택 전략

UI는 스냅샷 세트에 대해 하나의 기본 또는 현재 선택기 스냅샷만 인식합니다. 기본 스냅샷 선택을 위해서는 다음과 같은 동일 전략을 사용하십시오.

- 기본 스냅샷을 기본 스냅샷으로 사용

스냅샷을 백업으로 사용합니다. 기본 스냅샷만 변경합니다. 백업된 선택기 스냅샷은 변경하지 마십시오.

- 최신 스냅샷을 기본 스냅샷으로 사용

선택기 스냅샷을 새로 작성할 때마다 새 기본 선택기 스냅샷이 되게 합니다. 기본 스냅샷 또는 이전 선택기 스냅샷은 변경하지 않습니다.

- 세트의 스냅샷 이름 지정 규칙 식별

선택기 스냅샷 이름은 선택기 스냅샷 세트에서 고유해야 합니다.

환경 스냅샷 이름을 작성하려면 다음 기준을 사용하십시오.

- 이름은 스냅샷 사용법 또는 용도를 설명해야 합니다.
- 이름 지정 규칙은 정의된 표준을 따라야 합니다. 스냅샷 탭의 주석 상자를 사용하여 이름 지정 규칙을 설명할 수 있습니다.

- 세트에 단일 선택기 이름 사용


선택기 스냅샷을 작성한 후 선택기의 이름을 변경할 수 있습니다. 선택기 이름을 변경하는 경우 모든 선택기 스냅샷에 대해 업데이트됩니다.



## 기존 선택기 또는 선택기 스냅샷에서 선택기 스냅샷 작성

선택기 스냅샷을 작성하면 선택기의 새 인스턴스가 작성됩니다. 이 스냅샷은 사본이 아니라 새로운 선택기입니다.

**팁:** 스냅샷을 작성하고 기본 스냅샷을 변경하려면 사용자에게 필수 권한이 있어야 합니다. 385 페이지의 『스냅샷 권한의 액세스 그룹 확인 및 편집』의 내용을 참조하십시오.

1. 스냅샷을 찍을 선택기 또는 선택기 스냅샷 옆에 있는 **편집** 아이콘을 클릭하십시오.

- 기본 선택기 스냅샷을 찍으려면 선택기 목록(서버 > 선택기)에서 최상위 레벨 스냅샷 옆에 있는 편집 아이콘()을 클릭하십시오.

- 비기본 선택기 스냅샷을 찍으려면 스냅샷 아이콘()을 클릭하십시오. 스냅샷 보기에는 세트의 선택기 스냅샷이 표시됩니다. 비기본 선택기 스냅샷 옆에 있는 편집 아이콘()을 클릭하십시오.

2. 새 스냅샷 작성을 클릭하십시오.

3. 스냅샷 탭의 이름에 스냅샷 이름을 입력하십시오. 이름은 선택기 스냅샷을 찍는 모든 오브젝트에 지정됩니다.

이름은 선택기 스냅샷 세트에서 고유해야 합니다.

4. 선택기 스냅샷을 작성할 때 스냅샷을 찍을 Build Forge 오브젝트를 선택하십시오. 선택할 수 있는 오브젝트는 다음 표에 설명되어 있습니다.

오브젝트	설명
기본값	UI에서 기본 스냅샷은 선택기 목록의 최상위 레벨에 표시됩니다. 선택기 목록을 표시하려면 서버 > 선택기를 선택하십시오.
다음 선택기 포함	선택기가 포함 특성 유형을 사용하여 기타 선택기를 포함하는 경우 해당 선택기에 대한 스냅샷이 작성됩니다. 참고: 포함 변수 유형은 이전 릴리스에서 제공된 .include 기능을 바꿉니다.



5. 저장을 클릭해서 선택기 스냅샷을 저장하십시오.

## 기본 선택기 스냅샷 변경

기본 선택기 스냅샷은 선택기 스냅샷 세트의 최상위 레벨 스냅샷이며 선택기 목록(서버 > 선택기)에 표시됩니다.

**팁:** 스냅샷을 작성하고 기본 스냅샷을 변경하려면 사용자에게 필수 권한이 있어야 합니다. 385 페이지의 『스냅샷 권한의 액세스 그룹 확인 및 편집』의 내용을 참조하십시오.

기본 선택기 스냅샷을 변경하려면 새 기본값으로 설정할 스냅샷의 스냅샷 정의를 편집하십시오.

1. 서버 > 선택기를 선택하십시오.
2. 선택기 목록에서 기본 선택기 스냅샷에 대한 스냅샷 아이콘()을 클릭하십시오.
3. 스냅샷 목록에서 새 기본값이 될 선택기 스냅샷에 대한 편집 아이콘()을 클릭하십시오.
4. 기본값으로 설정을 클릭하십시오.
5. 중요: 팝업에서 확인 또는 취소를 선택하십시오.



확인	참조 업데이트: 오브젝트가 이전 기본 선택기를 참조하는 경우 새 기본 선택기를 사용하여 오브젝트를 업데이트하십시오.
취소	참조를 업데이트하지 않음: 이전 기본값을 참조하는 오브젝트에 대해 참조를 새 기본 선택기 스냅샷으로 업데이트하지 않습니다.

## 선택기 스냅샷의 스냅샷 이름 변경

선택기 스냅샷의 스냅샷 이름을 변경할 수 있으며 선택기 스냅샷을 작성할 때 스냅샷을 찍기로 선택한 오브젝트의 스냅샷 이름도 변경할 수 있습니다.

기본 스냅샷의 경우 이 옵션을 사용하여 기본 이름인 기본 스냅샷을 단일 선택기 스냅샷 전용 또는 현재와 미래의 모든 선택기에 대한 다른 스냅샷 이름으로 변경할 수 있습니다.

스냅샷 이름을 변경하려면 다음을 수행하십시오.

1. 서버 > 선택기를 선택하십시오.
2. 선택기 목록에서 기본 선택기 스냅샷에 대한 스냅샷 아이콘()을 클릭하십시오.
3. 스냅샷 목록에서 선택기 스냅샷에 대한 편집 아이콘()을 클릭하십시오.
4. 스냅샷 탭을 선택하십시오.
5. 이름에 새 이름을 입력하십시오.
6. 선택적: 주석에 주석을 입력하십시오.
7. 중요: 팝업에서 확인 또는 취소를 선택하십시오.

확인	<p>선택기 스냅샷 이름과 기타 스냅샷 오브젝트 이름 변경: 선택기 스냅샷을 작성할 때 스냅샷을 찍기로 선택한 오브젝트의 경우 이러한 오브젝트와 선택기 스냅샷의 이름을 변경합니다.</p> <p>기본 스냅샷의 경우: 모든 현재 선택기 스냅샷과 모든 미래 선택기 스냅샷의 이름인 기본 스냅샷을 변경합니다.</p>
취소	<p>선택기 스냅샷 이름은 변경하고 기타 스냅샷 오브젝트 이름은 변경하지 않음: 선택기 스냅샷을 작성할 때 스냅샷을 찍기로 선택한 오브젝트의 경우 이러한 오브젝트의 이름을 변경하지 마십시오. 선택기 스냅샷 이름만 변경하십시오.</p> <p>기본 스냅샷의 경우: 모든 현재 선택기 스냅샷과 모든 미래 선택기 스냅샷의 이름인 기본 스냅샷을 유지합니다.</p>

## 선택기 스냅샷 세트의 스냅샷 액세스 및 보기

선택기 스냅샷을 찍어 변경 또는 수정할 선택기의 새 인스턴스를 빠르게 작성합니다.

선택기 스냅샷을 작성하면 두 개 이상의 선택기(기준 선택기와 새 선택기 스냅샷)를 포함하는 스냅샷 세트가 작성됩니다.



스냅샷 세트의 모든 선택기 스냅샷을 보려면 다음을 수행하십시오.

### 1. 서버 > 선택기를 선택하십시오.

선택기 목록에는 선택기와 선택기 스냅샷의 목록이 표시됩니다. 최상위 레벨 스냅샷이 기본 선택기 스냅샷입니다.

### 2. 스냅샷 세트에 있는 선택기 스냅샷을 표시하려면 스냅샷 아이콘()을 클릭하십시오.

스냅샷 보기에서 다음을 수행할 수 있습니다.

- 새 선택기 스냅샷을 작성하십시오. 시작하려면 편집 아이콘()을 클릭하십시오.
- 선택기의 기본 스냅샷을 변경하십시오. 편집 아이콘()을 클릭하고 기본값으로 설정을 클릭하십시오.
- 표준 선택기의 경우와 같이 선택기 스냅샷 정의를 편집하십시오.



## 선택기 스냅샷 삭제

삭제 옵션을 사용하여 선택기 스냅샷을 삭제할 수 있습니다.

다른 오브젝트가 사용하는 선택기는 삭제할 수 없습니다. 예를 들어, 다른 선택기에 포함되거나 프로젝트, 단계 또는 스케줄에서 사용하는 선택기는 삭제할 수 없습니다.

선택기 스냅샷을 삭제하려면 다음을 수행하십시오.

### 1. 서버 > 선택기를 선택하십시오.

2. 선택기 목록에서 기본 스냅샷에 대한 스냅샷 아이콘()을 클릭하십시오.  
스냅샷 보기에는 세트의 선택기 스냅샷이 표시됩니다.
3. 삭제될 선택기 스냅샷 옆에 있는 편집 아이콘()을 클릭하십시오.
4. 삭제를 클릭하십시오.



---

## 제 18 장 환경에 대한 작업

이 절에서는 환경을 설정 및 관리하는 방법에 대해 설명합니다.

---

### 환경 정보

환경은 이름 지정된 변수 세트입니다.

정의된 환경은 다음을 수행하는 데 사용할 수 있습니다.

- 작업의 단계에서 사용할 변수를 설정하십시오. 환경은 서버, 프로젝트 및 단계에 지정할 수 있습니다. 실행 단계는 작업 실행 중에 다음 세 가지 환경 모두에서 변수 값을 상속합니다.
  - 작업이 실행 중인 서버와 연관된 서버 환경
  - 단계가 정의된 프로젝트와 연관된 프로젝트 환경
  - 단계와 명시적으로 연관된 단계 환경
- 예약 작업에서 사용할 변수를 설정하십시오. 예약 작업의 환경 세트는 프로젝트에 지정된 환경을 바꿉니다.
- 어댑터가 사용할 변수를 설정하십시오. 환경은 어댑터 링크에 지정할 수 있습니다. 환경은 프로젝트의 초기 어댑터 단계에서 사용합니다.

### 환경 상속

시스템은 단계를 실행하기 전에 단계 환경을 작성합니다. 단계 환경은 단계에 적용되는 모든 변수로 구성됩니다. 변수는 서버 환경, 프로젝트 환경 및 단계 환경에서 순서대로 상속됩니다. 기본 사례는 다음과 같습니다.

1. 서버 환경: 서버 환경 변수가 단계 환경에 복사됩니다.
2. 프로젝트 환경: 프로젝트 환경 변수가 단계 환경에 적용됩니다. 프로젝트 환경에 서버 환경의 변수와 이름이 같은 변수가 포함되는 경우 해당 값은 두 변수 정의 모두에서 변수 조치에 따라 업데이트됩니다.
3. 단계 환경: 단계 환경 변수가 단계 환경에 적용됩니다. 단계 환경의 변수가 서버 및 프로젝트 환경에서 상속된 변수와 이름이 같은 경우 변수 조치에 따라 값이 업데이트됩니다.

변수에 대한 변수 조치는 상속 시 값 적용 방법에 직접 영향을 줍니다. 예를 들어, 다음과 같습니다.

- 사례 1: 변수 조치가 설정인 경우 상속을 통해 값을 겹쳐씹습니다.
  - Server environment: X = 1, action: Set

- Project environment: X = 100, action: Set
- Step environment: X = 3, action: Set
- Final value during step execution: X = 3

변수 X는 1, 100, 3으로 차례로 설정됩니다. 설정 변수 조치는 새 값이 적용될 때마다 변수 값을 바꿉니다.

- 사례 2: 설정되지 않은 경우 설정 변수 조치로 인해 값이 상속되었습니다.

- Server environment: Y = 1, action: Set
- Project environment: Y = 100, action: Set
- Step environment: Y = 3, action: **Set if Not Set**
- Final value during step execution: x = 100

변수 Y는 서버 및 프로젝트 환경에서 Y에 대한 설정 조치로 인해 1, 100으로 설정됩니다. Y는 단계 환경에서 설정되지 않은 경우 설정 변수 조치를 사용하므로 프로젝트 환경에서 설정된 값이 상속됩니다.

## 특수 상속 사례

다음 사례는 상속에 영향을 줍니다.

### 인라인 프로젝트

단계는 단계의 인라인 특성에 프로젝트를 지정하여 프로젝트를 인라인합니다. 단계가 프로젝트를 인라인하면 호출된 프로젝트의 서버 환경과 프로젝트 환경이 사용되지 않습니다. 상속은 다음 순서로 수행됩니다.

1. 호출 단계에 대한 서버 환경.
2. 호출 단계에 대한 프로젝트 환경.
3. 호출 단계에 대한 단계 환경.
4. 호출된 프로젝트의 각 단계에 대한 단계 환경(지정된 경우).

### 체인 프로젝트

프로젝트 또는 단계는 프로젝트를 체인 패스 또는 체인 실패로 지정할 수 있습니다. 프로젝트가 이러한 방식으로 호출되며 고유 환경에서 실행됩니다. 또한 호출 프로젝트 또는 호출 단계에서 모든 변수에 액세스합니다. 그러한 변수는 접두부 BF\_CALLER\_를 사용하여 새 이름에 복사됩니다. 예를 들어, 호출 프로젝트 또는 단계의 BF\_NAME 변수는 호출된 프로젝트와 단계에서 BF\_CALLER\_BF\_NAME으로 사용할 수 있습니다.

### 예약 작업

스케줄에서 프로젝트가 스케줄되는 경우, 기본적으로 설정된 환경이 아닌 다른 환경을 프로젝트에 적용하도록 선택할 수 있습니다. 환경이 지정되면 환경 탭을 사용하여 해당 환경의 변수에 대한 값을 설정할 수 있습니다. 변수는 해당 프

로젝트 특성 설정에 따라 환경 탭에 나타나며 예약되지 않은 작업 시작에 대해 나타나는 경우와 동일한 규칙을 따릅니다.

#### 상속 순서 대체

상속 순서를 대체하려면 서버 환경을 마지막으로 적용 시스템 설정을 사용합니다. 값이 예인 경우 상속 순서는 다음과 같이 설정됩니다.

1. 프로젝트 환경
2. 단계 환경
3. 서버 환경

### 작업 시작 시 프로젝트 변수 변경 내용

사용자가 작업을 시작할 때 프로젝트 변수를 변경하여 프로젝트 환경에서 설정된 변수 값을 대체할 수 있습니다.

사용자가 작업을 시작하면 프로젝트 환경의 변수가 작업 시작 페이지에 표시됩니다. 각 변수의 프로젝트 특성에 따라 사용자가 제공된 값을 변경할 수 있습니다.

작업 시작 시 변경사항에는 프로젝트 환경에 정의된 변수와 동일한 상속 규칙이 적용됩니다.

예를 들어, 다음과 같습니다.

1. 초기값을 갖는 변수가 있는 JavaEnv 환경을 정의합니다(JavaVersion = 1.4).
2. JavaEnv 환경을 사용하는 MyBuild 프로젝트를 정의합니다.
3. MyBuild 프로젝트를 실행하는 작업을 실행합니다. 작업 시작 패널에서 JavaVersion 값을 1.5로 변경합니다.

결과:

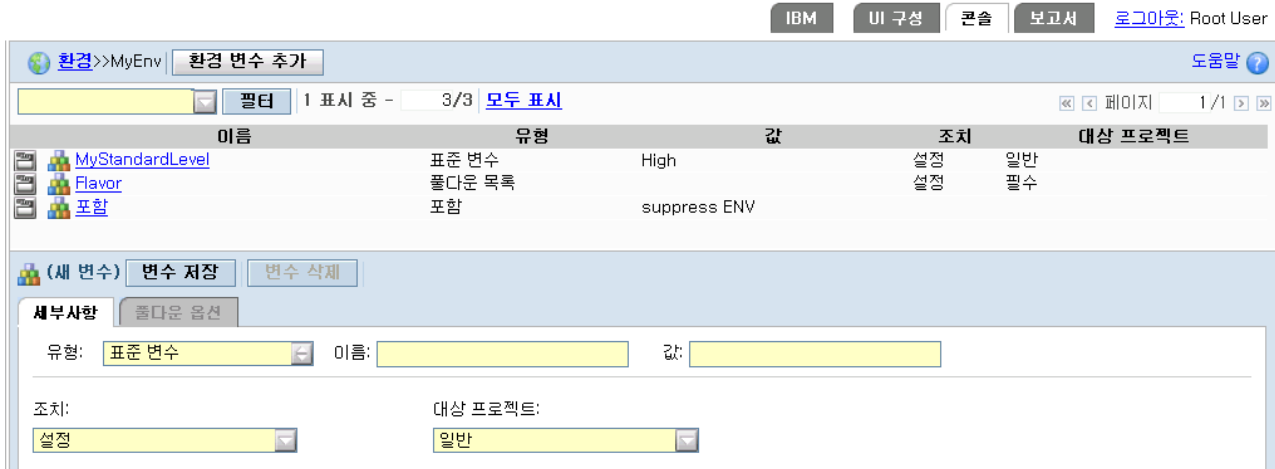
- 프로젝트 환경(JavaEnv)을 대체하지 않는 단계는 수정된 JavaVersion 값 1.5를 상속합니다.
- JavaEnv 환경을 단계 환경으로 명시적으로 사용하는 단계에서는 프로젝트 환경에 정의된 JavaVersion 값 1.4를 사용합니다.

---

## 변수 정보

변수는 환경 내에서 정의됩니다.

값 외에 변수에도 해석될 때 동작을 결정하는 추가 특성이 있습니다.



변수를 새로 작성하려면 다음을 수행하십시오.

1. 왼쪽 메뉴에서 환경을 클릭하십시오.
2. 환경 변수 추가를 클릭하십시오.
3. 변수의 이름을 지정하십시오.
4. 원하는 대로 변수에 대한 기타 특성을 지정하십시오. 자세한 정보는 326 페이지의 『환경 특성』의 내용을 참조하십시오.
5. 변수 저장을 클릭하십시오.

변수가 작성되면, 변수를 클릭하여 편집할 수 있습니다.

## 단계에서 변수 해석

단계 명령 또는 환경 변수 정의에서 UNIX® 스타일 또는 Windows® 스타일 변수 구문을 사용할 수 있습니다.

시스템은 프리프로세서를 사용하여 UNIX 스타일(\$VAR) 또는 Windows 스타일(%VAR%) 구문 둘 다 단계가 실행되는 서버에 적합한 형식으로 해석합니다. 사전 구문 분석을 통해 Windows 기반 서버나 Unix 기반 서버에서 단계를 실행할 수 있습니다.

예제:

- 다음 두 지정 명령문이 단계에서 동일합니다.

```
echo %fooVar%      # Windows syntax
echo $fooVar       # UNIX or Linux syntax
```

- 변수 지정은 사전 처리되지 않습니다. 따라서 특히 운영 체제 제한 없이 서버를 선택할 수 있는 시나리오에서는 명령행에서 직접 지정을 사용하지 마십시오. 환경의 변수를 사용하십시오.

```
set fooVar=100     # Windows
fooVar=200         # UNIX or Linux syntax
```

변수 구문 분석 방법은 다음과 같습니다.

1. 사전 구문 분석기가 변수 지정을 평가합니다. 특수 문자는 백슬래시 문자가 이스케이프하는 경우를 제외하고 사용됩니다(\$, %, {, }, ", '). 사전 구문 분석이 해제되어 있으면 모든 문자가 전달됩니다.
2. 대상 환경에서 양쪽 변수 지정을 평가합니다.
3. 평가된 변수 지정이 실행됩니다.

사전 구문 분석기, Windows 환경 및 여러 UNIX 및 Linux 셸은 특수 문자를 다르게 해석합니다. 특수 문자 및 백슬래시 이스케이프 문자를 사용할 때 주의하십시오.

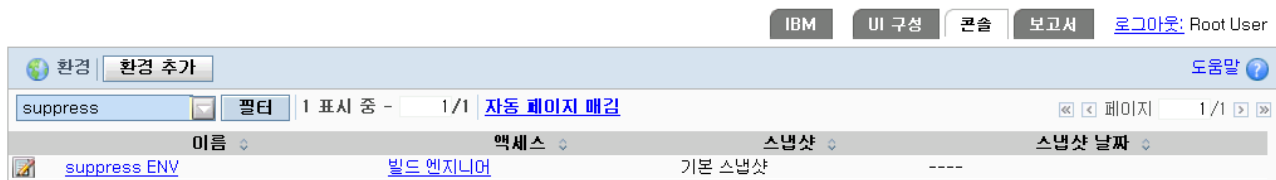
## 선언되지 않은 변수 해석

단계에서 변수가 호출되지만 정의되지 않은 경우, 리턴된 값이 사전 구문 분석이 작동(기본값)되는지 작동되지 않는지 여부에 따라 다릅니다. 사전 구문 분석 동작은 `bfagent.conf` 파일의 `no_preparse` 명령 또는 `_NO_PREPARSE_COMMAND` 환경 변수를 편집하여 설정됩니다. 337 페이지의 『트리거 변수 참조』의 내용을 참조하십시오.

변수 형식	리턴된 값 - 사전 구문 분석 작동(기본값)	리턴된 값 - 구문 분석 작동 안함
<code>echo %foo%</code>	foo	Windows: %foo% UNIX 또는 Linux: 공백
<code>echo \$foo</code>	foo	Windows: \$foo UNIX 또는 Linux: 공백
<code>echo \${foo}</code>	foo	Windows: \${foo} UNIX 또는 Linux: 공백
<code>echo \$[foo]</code>	빈 문자열	Windows: \$[foo] UNIX 또는 Linux: 시스템 오류

## 환경 패널 정보

환경 패널을 사용하여 프로젝트에 대한 환경을 작성 및 관리합니다. 패널을 보려면 환경을 선택하십시오.



환경 패널에서 환경을 클릭해서 해당 변수를 표시하고 편집할 수도 있습니다.

## 세부사항 탭

### 환경 특성

다음 환경 특성을 지정하십시오.

이름 환경 이름입니다.

액세스 이 환경을 사용할 사용자의 액세스 그룹입니다.

### 변수 특성

환경 패널에서 변수를 선택하여 변수 패널을 봅니다. 변수 패널을 사용하여 다음 특성을 지정하십시오.

이름 변수 이름에는 영숫자 문자(a-z, A-Z, 0-9) 및 밑줄 문자(\_)만 사용할 수 있습니다. 최대 길이는 255바이트입니다.

- 1바이트 문자 세트: 255자
- 2바이트 문자 세트: 127자

값 변수값은 다음 특성을 가지고 있습니다.

- 길이: 값은 모든 길이(길이가 있는 경우, 최대 운영 체제 한계까지)가 될 수 있습니다.
- 특수 문자: %, \$, [, ], {, }, \, " 및 ' 문자는 운영 체제 평가 이전의 사전 구문 분석기 및 운영 체제상의 평가에 대해 특수 의미를 가지고 있습니다. 특수 문자를 사용하지 마십시오. 특수 문자를 패스하려면 백슬래시(\)로 이스케이프하십시오. 324 페이지의 『단계에서 변수 해석』의 내용을 참조하십시오.

값이 작은따옴표 안에 있는 경우 변수 이름은 해석되지 않고 리터럴로 전달됩니다. 예를 들어, 변수 지정 `MyEnv='$env'`는 문자열 `$env`에 변수 값 `$env`가 아니라 `MyEnv`가 지정됩니다.

- 값의 변수: 변수값에 변수가 있으면, 해당 변수는 이를 사용하는 단계가 실행될 때 해석됩니다.
- 폴다운 값: 폴다운 변수의 항목을 지정하려면, 변수 유형을 폴다운 목록으로 설정하고 변수를 저장한 다음 편집하십시오. 폴다운 옵션 탭을 클릭하여 폴다운에 항목을 추가하십시오. 330 페이지의 『변수의 폴다운 작성』의 내용을 참조하십시오.
- 점 명령을 값으로: 일부 점 명령을 환경 변수 값으로 사용할 수 있습니다. 이 경우, 시스템은 점 명령을 다른 값으로 바꿉니다. 334 페이지의 『변수에서 점 명령 사용』의 내용을 참조하십시오.
- 캐리지 리턴: 변수는 캐리지 리턴을 저장하지 않습니다. 다음 예제에 표시된 대로 변수에 다중 행 값을 지정할 수 있습니다. `text.txt` 파일의 콘텐츠가 `test` 변수에 지정됩니다.

```
.bset env "test = `type text.txt`"
```

파일의 행이 연결됩니다. 예를 들어, 파일 콘텐츠가 다음과 같다고 가정하십시오.

```
A first line  
And a second line
```

변수 값은 다음과 같이 됩니다.

```
A first lineAnd a second line
```

**유형** 변수는 다음 유형 중 하나로 지정됩니다.

- 표준 - 기본값입니다. 변수에는 값과 조치가 지정되어 있을 수 있습니다.
- 포함 - 변수값은 다른 환경을 포함하도록 지시합니다. 해당 환경에 모든 변수가 포함됩니다.

참고: 포함 변수 유형이 이전 릴리스에서 제공된 `.include` 기능과 바뀔니다.

- 풀다운 목록 - 변수는 사용자가 선택할 수 있는 값 세트를 포함합니다. 풀다운 목록 변수가 작성되면, 해당 변수를 선택하고 풀다운 옵션 탭을 클릭하여 값을 추가할 수 있습니다.

## 조치

다음 중 하나:

- 설정: 기본 옵션입니다. 지정된 값이 변수에 지정됩니다. 변수가 작성됩니다 (존재하지 않는 경우).
- 설정되지 않은 경우 설정: 이 조치는 변수에 아직 값이 없는 경우에만 변수에 값을 지정합니다. 323 페이지의 『변수 정보』의 내용을 참조하십시오.
- 추가: 값이 변수의 현재 값에 추가됩니다. 운영 체제 특성의 PATH 구분 문자가 다음 값 사이에 추가됩니다.
  - Windows: 세미콜론(;)
  - UNIX 또는 Linux: 콜론(:)
- 선행부: 값이 현재 값 앞에 삽입됩니다. 운영 체제 특성의 PATH 구분 문자가 다음 값 사이에 추가됩니다.
  - Windows: 세미콜론(;)
  - UNIX 또는 Linux: 콜론(:)
- 지우기: 값이 빈 문자열로 설정됩니다. 값 특성이 값을 포함하는 경우 해당 값이 사용되지 않습니다.
- 삭제/언셋: 현재 적용된 환경에서 변수가 삭제됩니다. 값 특성이 값을 포함하는 경우 해당 값이 사용되지 않습니다.

- 지정 숨김: 시스템이 변수를 지정하지만, 로그에 값을 『\*\*\*\*\*』로 표시하여 숨깁니다. 이 옵션을 사용하여 프로젝트를 실행하는 사용자로부터 변수값을 숨깁니다.

**중요사항:** 숨겨진 변수에서 보안 정보를 저장하는 경우 주의하십시오. API 클라이언트를 사용하거나 데이터베이스를 검사하여 숨겨진 변수의 값을 읽을 수 있습니다. 비밀번호와 같이 보안 정보에 대해서는 안전하지 않습니다.

지정 숨김 변수는 기본적으로 편집하지 못하도록 보호되어 있습니다. 숨겨진 태그 편집 권한이 없는 사용자는 이 값을 편집할 수 없습니다. 그러나 해당 권한이 있는 경우 이 값을 삭제하고 다시 작성할 수 있습니다. 숨겨진 태그 편집 권한이 있는 사용자는 이 값을 직접 편집할 수 있지만 사용할 값을 다시 입력해야 합니다. 이 동작은 해당 유형을 설정으로 변경하여 사용자가 지정 숨김 변수 값을 표시하지 못하게 합니다.

**참고:** 단계에서 변수가 숨겨진 항목 지정으로 설정된 경우 해당 단계의 모든 변수는 숨겨진 항목 지정으로 설정된 경우와 같이 처리됩니다.

**참고:** 숨겨진 항목 지정 변수는 프로젝트에만 적용됩니다. 숨겨진 항목 지정 변수가 어댑터로 전달되면 더 이상 숨겨진 상태가 아닙니다.

일반적으로 시스템은 명령행의 변수 구문을 운영 체제에 적합한 양식으로 변경합니다(Windows®의 경우 %VAR% Linux® 및 UNIX® 시스템의 경우 \$VAR). 숨겨진 변수에 대해서는 이를 수행하지 않습니다. 변수는 직접 서버로 전달되며 서버의 운영 체제 환경이 변수를 해석합니다.

그러므로 운영 체제에서 구문 분석할 수 없는 변수 이름을 사용하지 마십시오. 예제:

- \$name: 이 변수가 지정 숨김으로 설정되면 UNIX 또는 Linux 시스템에서 해석될 때 운영 체제는 Build Forge 환경이 아니라 해당 환경에서 해석하려고 시도합니다.
- %address%. 이 변수가 지정 숨김으로 설정되면 Windows 시스템에서 해석될 때 운영 체제는 Build Forge 환경이 아니라 해당 환경에서 해석하려고 시도합니다.

### 프로젝트 중

수동으로 작업을 시작할 때 변수를 사용하는 방법을 정의합니다. 이 특성은 프로젝트에 지정된 환경에서 사용되는 변수에만 영향을 줍니다. 작업이 실행 중이면 특성이 변수에 영향을 주지 않습니다. 값은 다음 중 하나일 수 있습니다.

- 일반: 프로젝트에 지정될 때 변수가 일반적으로 동작합니다.



- 필수: 변수 값이 있어야 합니다. 이 특성을 가진 변수가 시작 패널에서 강조 표시됩니다. 변수 정의에 정의된 값이 충분합니다. 값이 정의되지 않은 경우 작업을 빨리 시작하거나 시작할 수 없습니다.

사용자가 아니라 스케줄러가 필수 변수를 포함하는 작업을 시작한 경우, 현재 값이 있는 경우에는 변수가 변경되지 않고 남아 있으며 값이 없는 경우에는 공백입니다.

- 읽기 전용: 값을 변경할 수 없습니다.
- 표시 억제: 작업 시작 페이지에 변수가 표시되지 않습니다. 그러나 변수는 여전히 존재하고 단계에서 사용할 수 있습니다.
- 필수 변경: 변수값을 변경해야 합니다. 이 특성의 최상위 레벨 변수가 시작 패널에서 강조 표시됩니다. 새 값이 입력되지 않은 경우 작업을 빨리 시작하거나 시작할 수 없습니다.

**참고:** 사용자가 아니라 스케줄러가 변경 필수 변수를 포함하는 작업을 시작한 경우, 변수 값은 변경되지 않습니다. 또한 변경 필수 변수를 포함하는 작업이 클래스 특성으로 시작된 경우 변수 값은 변경되지 않습니다. 클래스 특성은 제거 시 시작, 입력 시 시작 및 종료 시 시작을 포함합니다.

## 스냅샷 탭

환경 스냅샷은 환경의 인스턴스입니다. 스냅샷 탭에서는 스냅샷 이름을 제공합니다. 스냅샷 이름 및 스냅샷에 대한 주석을 보거나 변경하려면 이 탭을 사용하십시오.

---

## 환경 작성

### 프로시저

1. 왼쪽 메뉴에서 환경을 클릭하십시오.
2. 환경 추가를 클릭하십시오.
3. 환경의 이름을 지정하십시오.
4. 멤버가 환경을 사용할 수 있어야 하는 액세스 그룹을 지정하십시오.
5. 저장을 클릭하십시오. 환경에 변수를 추가할 수 있는 변수 패널이 표시됩니다.

---

## 변수 사용

다음 절에서는 변수를 포함하는 공통 태스크를 수행하는 프로시저를 설명합니다.

## 변수의 풀다운 작성

### 이 태스크 정보

하나의 변수에 가능한 여러 값을 정의할 수 있습니다. 사용자가 제공하는 값은 풀다운에서 선택 가능한 옵션으로 표시됩니다.


환경 변수의 풀다운을 작성하려면 다음을 수행하십시오.

1. 프로젝트 > 환경을 선택하십시오.
2. 목록에서 환경을 선택하십시오.
3. 환경 변수 추가를 클릭하십시오. 다음과 같이 특성을 채우십시오.
  - 이름: 환경 변수 이름을 입력하십시오.
  - 유형: 풀다운 목록을 선택하십시오.
  - 조치: 조치를 선택하십시오.
  - 프로젝트에서: 특성을 선택하십시오.
4. 저장을 클릭하십시오.
5. 변수를 클릭하십시오.
6. 풀다운 옵션 탭을 클릭하십시오.
7. 다음과 같이 풀다운 값을 추가하십시오.
  - a. 풀다운 옵션의 이름을 지정하십시오. 사용자에게는 풀다운의 선택 사항 이름으로 표시됩니다. 풀다운 이름은 영숫자 문자(a-z, A-Z, 0-9)와 밑줄 문자(\_)만 사용할 수 있습니다. 최대 길이는 255바이트입니다.
    - 1바이트 문자 세트: 255자
    - 2바이트 문자 세트: 127자
  - b. 풀다운 옵션의 값을 지정하십시오. 사용자가 사용 중인 값을 볼 수 있도록 하려면 변수 이름과 같은 값을 설정할 수 있습니다. 값에서 변수가 사용되는 경우 단계가 실행되는 운영 체제에서 해석됩니다. 이는 실행 전에 처리되거나 평가되지 않습니다.
  - c. 작성을 클릭하십시오. 목록에 옵션이 추가됩니다.
  - d. 원하는 값 각각에 대해 반복하십시오.
8. 변수 저장을 클릭하십시오.
9. 제공된 기본 옵션을 설정하십시오. 풀다운 옵션을 채우고 변수를 저장한 후 이를 수행하십시오. 환경을 클릭하십시오. 세부사항 탭에서 기본 옵션 특성을 --없음--에서 기본적으로 나타낼 값의 이름으로 변경하십시오.

기본 옵션이 설정되지 않은 경우(값이 --없음--인 경우) 변수의 세부사항 탭에서 첫 번째 항목을 보면 풀다운 옵션 목록의 첫 번째 항목이 기본적으로 값 옆에 표시됩니다.

## 10. 변수 저장을 클릭하십시오.

다음과 같이 추가 옵션 작업을 수행할 수 있습니다.

- 목록 내 옵션 위치를 지정하거나 삭제하려면 각 옵션 이름 왼쪽에 있는 편집 아이콘  을 사용하십시오.
- 편집할 옵션을 클릭하십시오. 이름 및 값 필드를 모두 편집할 수 있습니다. 완료되면 저장을 클릭하십시오.
- 이름 및 값 필드를 지우려면 지우기를 클릭하십시오. 일반적으로 이 작업은 새 옵션을 작성하기 위해 기존 옵션을 본 후에 수행합니다.

## 다른 환경 포함

포함 변수 유형을 사용하여 다른 기존 환경의 모든 변수를 포함할 수 있습니다.

1. 프로젝트 > 환경을 선택하십시오.
2. 새 환경을 작성한 다음 저장을 클릭하십시오.
3. 환경 이름을 클릭하십시오.
4. 환경 변수 추가를 클릭하십시오. 다음과 같이 특성을 채우십시오.
  - 이름: 환경 변수 이름을 입력하십시오.
  - 유형: 포함을 선택하십시오. 사용자 인터페이스가 변경되어 환경 포함 폴다운이 표시됩니다. 조치 및 프로젝트 특성은 제거됩니다.
  - 환경 포함: 목록에서 포함할 환경을 선택하십시오.
5. 변수 저장을 클릭하십시오.

## 단계 실행 중 변수 값 변경

단계, 프로젝트 또는 영구 범위 실행 중에 변수를 변경할 수 있습니다.

- 단계 범위: 단계에서 명령을 사용하면 명시적 지정을 사용하여 변수 값을 대체할 수 있습니다. 해당 값은 현재 단계에서만 유효합니다.
- 프로젝트 범위: 단계에서 .bset 명령을 사용하면 실행 중 작업의 범위에 대한 변수 값이 변경됩니다. .bset를 사용하여 새 변수를 작성할 수 있습니다. 새 변수는 나머지 작업에 대한 범위에 속합니다. .bset를 사용한 변경사항은 .bset가 사용된 단계 이후 단계에서 유효합니다.
- 영구 범위: 단계에서 .set 명령을 사용하면 환경 변수 정의가 변경됩니다. 변수는 서버 환경, 프로젝트 환경 및 단계 환경에서 정의됩니다. .set를 사용하여 서버 변수 또는 프로젝트 변수를 변경해도 현재 작업의 변수 사본은 변경되지 않습니다. 현재 작업 이후에 실행된 작업은 변경된 변수를 가져옵니다. 그러나 .set 명령이 단계 환경 변수를 변경하는 경우 단계 환경을 사용하는 다음 단계는 변경된 변수를 가져옵니다. .set 명령은 새 변수를 작성할 수 없습니다.

예를 들어, JavaVersion = 1.4 변수를 포함하는 Java라는 프로젝트 환경에서 프로젝트를 실행하고 .bset를 사용하여 값을 1.5로 변경하는 경우, 해당 프로젝트 환경을 상속하는 단계는 값 1.5를 가져오지만 Java 환경을 참조하는 단계는 특별히 원래 값 1.4를 가져옵니다.

시스템은 작업을 시작할 때 프로젝트 환경 변수를 작업을 위해 남겨둔 데이터베이스 레코드에 복사하고 그 후 프로젝트 기본값을 가져올 때 이 작업 환경을 참조함에 유의하십시오. 사용자가 작업을 시작할 때 프로젝트 변수의 시작 값을 수정하는 경우 해당 값은 작업 레코드에 기록됩니다.

## Windows 드라이브 매핑

Microsoft Windows® 운영 체제는 매핑된 드라이브를 다르게 관리합니다. 에이전트는 사용자 계정에 대해 기억된 연결을 다시 매핑하려고 시도하지만 실행시에 매핑을 완료하지 못할 수도 있습니다. 특수 환경 변수(\_MAP 변수)를 사용하여 Windows에서 드라이브 매핑을 지원할 수 있습니다. 이 변수를 설정하면, Windows 에이전트는 단계를 실행하기 전에 드라이브를 매핑합니다.

\_MAP 변수 사용 시 일반적인 관행은 프로젝트 환경에서 해당 변수를 지정하는 것이므로 동일한 드라이브 매핑이 환경 변수 상속을 통해 모든 단계 환경으로 전달됩니다. 또한 단계 환경에서 \_MAP 변수를 정의하는 경우 특정 단계에 대해 하나의 \_MAP 값만 정의할 수 있으므로 단계 환경의 값이 프로젝트 환경을 대체함에 유의하십시오.

Windows 환경용이긴 하지만 슬래시를 사용하여 \_MAP 변수에서 디렉토리 경로 이름을 분리하십시오. 경로가 사용될 때 에이전트는 필요에 따라 이를 자동으로 조정합니다.

예를 들어, \_MAP를 다음과 같이 설정하면

```
X:=//server/share
```

X: 드라이브를 Windows UNC 경로 이름 \\server\\share에 연결하는 런타임 매핑을 정의합니다.

\_MAP 변수에 추가 매핑 스펙을 제공하여 여러 개의 드라이브를 매핑할 수 있습니다 (세미콜론을 사용하여 분리).

```
X:=//server/share;Y:=//server/share2
```

기본적으로, Windows에서의 드라이브 매핑은 논리 서버에 대해 정의된 것과 동일한 사용자 이름/비밀번호를 사용하여 수행됩니다. 다음 예제와 같이 매핑 후에 괄호 안에 사용자 이름과 비밀번호를 추가하여 다른 사용자 이름에 대해 드라이브를 매핑할 수 있습니다.

```
X:=//server/share(alternateusername,password)
```

**참고:** 비밀번호에 \$ 문자가 포함되어 있으면 또 하나의 \$ 문자로 이스케이프하십시오.  
예제: 비밀번호 pas\$word를 pas\$\$word와 같이 입력하십시오. 비밀번호에 다음  
특수 문자(% , [ , ] , { , } , " 또는 ')를 사용하지 마십시오.

\_MAP 변수를 통해 맵핑된 드라이브는 명령 완료 시 맵핑 해제됩니다.

드라이브가 성공적으로 맵핑되더라도 Windows에서의 드라이브 맵핑은 시스템의 콘솔  
에 로그인한 사용자가 문제의 드라이브를 사용 중이거나 공유하는 경우에는 여전히 액  
세스할 수 없습니다.

### 사용 가능한 다음 드라이브 이름을 사용하여 Windows 드라이브 맵핑

사용 가능한 다음 드라이브 이름을 시스템이 선택하도록 할 수 있습니다. 다음 구문을  
사용하십시오.

```
<driveletter>?=//<directory path>
```

예를 들어, \_MAP를 다음과 같이 설정할 수 있습니다.

```
X?=//server/share
```

이 경우, 시스템은 드라이브를 X에 맵핑하지 않습니다. 대신, 사용 가능한 다음 드라이브  
에 맵핑하고 선택한 드라이브 이름을 \_MAP\_X라는 변수에 저장합니다. 선택된 드라이브가 F인 경우, \_MAP\_X 변수 값은 『F:』입니다. 변수를 사용하여 맵핑에 액세스할 수 있습니다.

다음과 같이 다중 맵핑을 사용할 수 있습니다.

```
X?=//server/share; Y?=//server/public
```

원하는 모든 문자를 사용할 수 있으며, 다음 예제와 같이 다중 맵핑을 사용할 수도 있습니다.

- F:는 //server/share에 맵핑됨
- G:는 //server/public에 맵핑됨

또한 이 예제는 다음 변수를 작성합니다.

- 값이 『F:』인 \_MAP\_X
- 값이 『G:』인 \_MAP\_Y

**참고:** Cygwin을 사용하는 Windows® 시스템을 대상으로 할 때 사용 가능한 다음 드라이브 구문을 사용하는 경우, 다음과 같이 백슬래시로 물음표를 이스케이프해야 합니다.

```
Y\?=//server/share
```

## 에이전트 기반 드라이브 맵핑

에이전트의 구성 매개변수를 사용하여 드라이브를 맵핑할 수 있습니다. *map* 매개변수는 BFAgent.conf 파일에 추가될 때 *\_MAP* 변수와 동일한 구문을 사용합니다. 이 매개변수를 사용하여 특정 서버의 드라이브 맵핑을 작성할 수 있습니다. *\_MAP* 변수도 사용하는 경우, 해당 맵핑은 에이전트 맵핑을 대체합니다.

## 변수에서 점 명령 사용

일부 점 명령은 환경 변수의 이름 또는 값으로 사용될 수 있습니다.

### .source로 명령 전에 스크립트 실행

시스템은 .source라는 특수 환경 변수를 정의하여 시스템이 명령을 실행하기 전에 서버에서 스크립트를 실행할 수 있는 기능을 제공합니다. 따라서 서버의 소스 파일에서 환경 변수 세트를 로드하거나 사용자 정의 준비 명령을 실행할 수 있습니다.

이 기능을 시도하려면 다음을 수행하십시오.

1. 일부 문장을 보여주는 mybatch.bat라는 일괄처리 파일을 시스템에 작성하십시오. 일괄처리 파일을 C:\temp에 저장하십시오.
2. Step Variables라는 새 환경을 작성하십시오.
3. 값이 C:\temp\mybatch.bat인 .source라는 변수를 추가하십시오.
4. 단계의 환경을 새로 작성된 Step Variables 환경으로 설정하십시오.
5. 프로젝트를 실행하고 단계의 로그 출력을 검사하십시오.

단계 명령 전에 mybatch.bat 파일이 실행되었음을 표시하는 추가 로그 데이터에 주의하십시오. .source에 대한 몇 가지 중요한 노트는 다음과 같습니다.

- 지정된 경로는 인수를 가질 수 없습니다.
- Windows® 플랫폼에서 스크립트는 호출을 통해 호출됩니다.
- UNIX® 플랫폼에서 스크립트는 실행 중인 셸에서 소싱될 때 원시 셸 구문으로 되어 있어야 합니다.

### .date 명령으로 변수에 날짜 또는 시간 저장

.date 환경 점(.) 명령을 사용하여 변수에 현재 날짜 또는 시간을 제공할 수 있습니다. 명령을 변수 값으로 사용하십시오. 시스템은 변수를 사용하는 프로젝트를 실행할 때 시스템이 .date 명령의 결과로 변수를 업데이트합니다.

예를 들어, 값이 .date %B인 MONTH라는 변수(프로젝트에 포함될 때)는 5월 중에 값이 『May』인 작업에 제공됩니다.

날짜 형식 코드 목록을 포함하여 이 명령 사용에 대한 자세한 정보는 425 페이지의 『.date』를 참조하십시오.

## 시스템 변수 참조

변수에 사용하기 위해 시스템에 정의된 변수가 사용 가능합니다.

시스템은 작업의 각 단계에서 자동으로 다음 변수의 값을 설정합니다. 이 변수는 읽기 전용입니다. 작업에 대한 변수 값은 단계 로그의 ENV 행에 나열됩니다. 처음 네 개는 프로젝트 레벨의 알림입니다. 기타 모든 BF\_ 변수는 단계 레벨에서 사용됩니다.

프로젝트 레벨 변수	값
BF_D	날짜. 태그에서 사용될 수 있습니다. 형식은 태그: 날짜 형식 시스템 설정으로 판별됩니다.
BF_J	연중 날짜. 태그에서 사용될 수 있습니다.
BF_T	시간. 태그에서 사용될 수 있습니다. 형식은 태그: 시간 형식 시스템 설정으로 판별됩니다.
BF_W	0(일요일)부터 6(토요일)까지의 숫자로 표시된 요일입니다.

단계 레벨 변수	값
BF_D	날짜. 태그에서 사용될 수 있습니다. 형식은 태그: 날짜 형식 시스템 설정으로 판별됩니다.
BF_J	연중 날짜. 태그에서 사용될 수 있습니다.
BF_T	시간. 태그에서 사용될 수 있습니다. 형식은 태그: 시간 형식 시스템 설정으로 판별됩니다.
BF_W	0(일요일)부터 6(토요일)까지의 숫자로 표시된 요일입니다.
BF_AGENT_PLATFORM	에이전트가 실행 중인 운영 체제 플랫폼을 식별하는 문자열입니다.
BF_AGENT_VERSION	현재 서버의 에이전트 버전 이름입니다.
B	기본 태그 변수(1부터 시작하고 모든 작업에 대해 증분됨)입니다. 태그에서 사용될 수 있습니다(BF_TAG 로 표시됨).
BF_BID	동일한 프로젝트의 작업에 고유한 작업 ID 번호입니다.
BF_CALLER_	호출하는 프로젝트에서 체인 프로젝트로 전달된 변수에 적용된 접두어입니다.
BF_CLASS	프로젝트의 Build Forge 클래스
BF_ENGINE	고유하게 엔진을 식별하는 문자열입니다. 또한 이 값은 설치 디렉토리의 파일(engine.id)에도 저장됩니다. 예제: D8531015-6C07-1014-8CA0-BD58317220B3.
BF_HOST	논리 서버의 호스트 이름(TCP/IP 호스트 이름)입니다. (이 변수는 서버 환경의 일부입니다.)
BF_ITERATION	While 루프에서 단계가 시작된 횟수입니다. 단계의 조건이 true로 평가되면 증분됩니다. 작업 다시 시작은 이 변수 값을 다시 시작하기 위한 반복으로 사용됩니다.



단계 레벨 변수	값
BF_ITERATION_MAX	While 루프를 실행할 수 있는 최대 횟수입니다. 단계 특성에서 설정됩니다. 이 반복 수에 도달하면, BF_ITERATION_MAX_REACHED가 예로 설정됩니다.
BF_ITERATION_MAX_REACHED	기본적으로 작성되거나 설정되지 않습니다. While 루프 단계의 단계 고유 ID(BF_SSID)는 While 루프 반복이 BF_ITERATION_MAX에 도달하면 이 변수에 추가됩니다. 프로젝트의 다중 While 루프 단계가 해당 BF_ITERATION_MAX에 도달하면, 이 변수는 최대 반복 수에 도달하는 각 단계에 하나씩 여러 개의 값을 포함합니다.
BF_LASTGOODRUN	동일한 프로젝트의 마지막 패스 작업 날짜 또는 패스 작업이 없는 경우 마지막 작업입니다.
BF_LASTGOODTAG	마지막 패스 작업의 태그(또는 동일한 프로젝트의 패스 작업이 저장되지 않은 경우 마지막 작업)
BF_LASTGOODUNIX	BF_LASTGOODRUN과 같지만 UNIX® 형식으로 날짜를 표시합니다.
BF_LASTRUN	현재 작업의 이전 실행 날짜입니다.
BF_LASTTAG	같은 프로젝트의 이전 작업에 사용하는 태그 문자열입니다.
BF_LASTUNIX	BF_LASTRUN과 같지만 UNIX 형식으로 날짜를 표시합니다.
BF_ONFAIL	단계의 정지/계속 플래그입니다.
BF_PID	프로젝트 ID 번호입니다.
BF_PROJECTNAME	이 작업의 프로젝트 이름입니다.
BF_PROJECTNAME_PHYS	프로젝트 디렉토리를 작성하는 데 사용되는 프로젝트 이름. 시스템은 올바른지 않은 상태 디렉토리 문자 시스템 설정에 지정된 문자를 밑줄 문자로 변경하여 프로젝트 디렉토리를 작성합니다. 예를 들어, 설정에 공백이 포함된 경우 My Project라는 프로젝트는 My_Project라는 프로젝트 디렉토리를 수신합니다.
BF_ROOT	작업 특성에서 가져온, 작업의 기본 작업 디렉토리입니다. BF_STEP_ROOT를 참조하십시오.
BF_SERVER	현재 작업이 실행 중인 서버 이름(이 변수는 서버 환경의 일부임)입니다.
BF_SERVER_ROOT	서버 특성에서 논리 서버에 지정된 경로(이 변수는 서버 환경의 일부임)입니다.
BF_SID	빌드 내 단계 결과의 순서 번호입니다.
BF_SPID	다른 작업이 현재 작업을 호출한 경우 호출하는 프로젝트 ID를 포함합니다. 그렇지 않으면, 값은 BF_PID와 같습니다.
BF_STEPNAME	단계 이름입니다. 단계 특성에서 설정됩니다.
BF_STEP_ROOT	단계 특성에서 가져온 단계의 기본 작업 디렉토리입니다. BF_ROOT를 참조하십시오.



단계 레벨 변수	값
BF_SSID	프로젝트에서 현재 단계를 고유하게 식별하는 단계 ID
BF_TAG	작업의 태그입니다. 태그 정의는 변수를 포함할 수 있습니다. 이 변수는 작업이 시작될 때 해당 변수를 해석한 결과로 얻은 값을 포함합니다.
BF_TAG_PHYS	작업의 태그입니다(BF_TAG 값에 있는 모든 간격이 밑줄로 바뀜). 단계에서 절대 옵션이 선택된 경우, BF_TAG_PHYS는 BF_TAG와 동일합니다.
BF_USER	작업 소유자의 사용자 이름입니다.

## 트리거 변수 참조

시스템은 다음 변수 이름을 감시합니다. 단계의 환경에 이런 변수 중 하나가 포함된 경우(명확하게 또는 프로젝트 또는 서버에서 상속됨), 조치가 수행됩니다.

변수	컨텐츠
_CI_BUILD_DELETE	작업이 실행된 후에 빌드 및 연관된 빌드 데이터를 삭제하려면 이 변수를 임의의 값으로 설정하십시오 (실행된 기타 프로젝트 빌드가 없으면 태그 변수는 삭제된 빌드 이전의 초기 값으로 다시 설정됨).
_CI_BUILD_KEEP	작업이 실행된 후에 빌드 및 연관된 빌드 데이터를 보존하려면 이 변수를 임의의 값으로 설정하십시오. 예를 들어, 작업에 어댑터 링크가 포함되어 있고 어댑터 단계가 실패하는 경우 기타 프로젝트 단계는 실행되지 않습니다. 예를 들어, 디버깅을 위해 작업에 대한 빌드 레코드 사본을 보존할 수 있습니다.
CLEARCASE_VIEW	지정된 ClearCase 보기를 시작합니다. 이 변수에 지정된 보기는 존재해야 하며 이 변수를 사용하는 단계는 "absolute"로 설정되어야 합니다. Microsoft Windows를 실행하는 시스템에서 이 변수는 bfagent.conf의 에이전트에 대한 cc_suppress_server_root 구성 옵션과 함께 사용해야 합니다.
_CLEARCASE_VIEWS	명령 실행 전에 시작할 ClearCase 보기 목록을 지정합니다. 값을 쉼표로 분리된 보기 목록으로 설정하십시오(예: 『View1,View2,View3』).
_CLEARCASE_VOBS	명령 실행 전에 마운트할 ClearCase VOB 목록을 지정합니다. 값을 쉼표로 분리된 VOB 목록으로 설정하십시오(예: 『\Vob1,\Vob2,\Vob3』).

변수	컨텐츠
_CONTEXT_LOG_RANGE	로그 출력을 필터 일치 가까이에 있는 행으로 제한하려면 이 변수를 사용하십시오. 이 변수는 양의 정수 값을 가지며, 시스템이 크기가 변수 값과 같은 각 필터 문자열 히트 근처의 행 범위를 제외하고 로그 출력을 생략하게 합니다. 예를 들어, 변수를 5로 설정한 경우 로그는 필터 일치가 있는 행과 이전 5개의 행 및 해당 일치 다음의 5개 행만 표시합니다.
_ERROR_THRESHOLD	<p>허용되는 최대 오류 수(사용자가 정의한 실패 설정 필터로 포착됨)를 설정합니다. 이 변수를 사용하여 개별 단계 또는 프로젝트의 실패 또는 메시지 임계값을 설정할 수 있습니다.</p> <p>다음 형식 중 하나를 사용하십시오.</p> <ul style="list-style-type: none"> <li>값 5 또는 F5는 5개를 초과하는 오류가 발생하는 경우 작업이 실패하게 됨을 표시합니다.</li> <li>값 N7은 7개를 초과하는 오류가 발생하는 경우 시스템에 작업 노트에 메시지를 추가함을 표시합니다. 메시지는 이 임계값이 충족되었음을 표시합니다.</li> </ul> <p>단계에서 변수를 사용할 때 시스템은 개별 단계에서 오류를 계수합니다. 다음과 같은 추가 양식을 사용할 수 있습니다.</p> <ul style="list-style-type: none"> <li>W9 같은 값은 필터가 포착하는 향후 오류에 관계없이 9개의 오류 후에는 단계가 경고 상태에 놓이게 됨을 표시합니다.</li> <li>C8 같은 값은 8개의 오류 후에 단계가 실패 상태로 설정되지만 실패 지우기 필터가 실패를 지울 수 있음을 표시합니다.</li> </ul> <p>참고: 이 변수가 계수하는 오류는 필터를 실패 설정 조치와 일치시키고 프로젝트의 단계에 지정된 문자열로 정의됩니다. 필터에 의해 실패로 식별된 각 문자열은 단계 총계에 하나의 오류로 계수되고 프로젝트 총계에 하나의 오류로 계수됩니다.</p>
_EXITCODE_MAP	시스템이 단계 성공 표시기로 승인해야 하는 숫자 목록(쉼표, 공백, 세미콜론 또는 콜론으로 분리)을 지정합니다. 기본적으로 종료 코드 0은 성공을 표시합니다. 이 변수가 지정된 경우, 변수에 나열된 모든 값도 성공을 표시합니다.

변수	컨텐츠
<code>_InterfaceLoggingLevel</code>	<p>Build Forge가 어댑터 단계를 실행할 때 로깅하는 로그 데이터의 양을 제어합니다. <code>_InterfaceLoggingLevel</code> 이름을 사용하여 어댑터 환경에 환경 변수를 작성하십시오. 0에서 8까지의 정수 값을 지정하십시오. 로깅 레벨은 포괄적입니다. 예를 들어, 레벨 2는 레벨 1과 레벨 0의 정보를 포함합니다.</p> <ul style="list-style-type: none"> <li>• 0: Exec 행 + 서버 연결 오류 또는 취소 알림, 그 밖엔 없음</li> <li>• 1: 구문 분석된 명령(서버로 보내질 명령)</li> <li>• 2: 구문 분석되지 않은 명령(로컬 변수가 설정되기 전의 명령)</li> <li>• 3: 빌드 및 환경 변수 SET 행</li> <li>• 4: 임시 및 내부 변수 SET 행</li> <li>• 5: 환경 평가, 이메일 그룹 추가, BOM 텍스트 로깅 행</li> <li>• 6: 블록 및 하위 블록 시작/끝 행</li> <li>• 7: (기본 로깅 레벨) 일치 패턴과 비교하여 확인되는 에이전트 출력 + 패턴과 일치한 행</li> <li>• 8: 모든 에이전트 출력</li> </ul>
<code>_LOG</code>	<p>Build Forge 에이전트의 원시 출력을 포함하는 로그 파일을 작성할 경로 이름을 지정합니다.</p> <p><b>참고:</b> 이 로그는 <code>_LOG_TIMESTAMP</code>도 지정되지 않는 한, 시간소인을 포함하지 않습니다. 이 파일의 로그 데이터는 일반적으로 에이전트 코드, 로그 버킷 및 메시지와 같이 형식화됩니다.</p> <p>서버에 작업 로그 사본을 저장하려면 이 변수를 사용하십시오. 파일이 있으면, 시스템은 여기에 추가합니다.</p>
<code>_LOG_TIMESTAMP</code>	<p><code>_LOG</code>의 각 출력 행에 시간소인을 접두부로 추가합니다. 이 변수 값은 <code>.date</code> 및 <code>.gmdate</code> 환경 명령에서 사용한 동일한 <code>strftime</code> 구문의 형식 문자열이어야 합니다.</p> <p><b>참고:</b> <code>_LOG</code>가 필요합니다.</p>
<code>_MAP</code>	<p>이 변수 사용 방법에 대한 설명은 332 페이지의 『Windows 드라이브 맵핑』의 내용을 참조하십시오.</p>

변수	컨텐츠
_NO_PREPARSE_COMMAND	일반적으로 시스템은 에이전트로 명령을 보내기 전에 환경 변수 값을 분석하려고 시도합니다. _NO_PREPARSE_COMMAND 변수가 임의의 값을 사용하여 정의된 경우, 시스템은 변수를 분석하지 않고 에이전트로 보냅니다. 운영 체제 셸이 변수를 처리하도록 하려면 이 변수를 사용하십시오.
_PRISM_DIR_POSTCMD	IDE용 플러그인과 함께 사용됨. 프로젝트 단계가 실행된 후 디렉토리에서 실행할 명령을 지정합니다. 588 페이지의 『테스트 프로젝트의 특수 변수』의 내용을 참조하십시오.
_PRISM_DIR_PRECMD	IDE용 플러그인과 함께 사용됨. 프로젝트 단계 중에 서버로 복사되기 전에 디렉토리에서 실행할 명령을 지정합니다. 588 페이지의 『테스트 프로젝트의 특수 변수』의 내용을 참조하십시오.
_PRISM_FILE_POSTCMD	IDE용 플러그인과 함께 사용됨. 프로젝트 단계가 실행된 후 파일에서 실행할 명령을 지정합니다. 588 페이지의 『테스트 프로젝트의 특수 변수』의 내용을 참조하십시오.
_PRISM_FILE_PRECMD	IDE용 플러그인과 함께 사용됨. 프로젝트 단계 중에 서버로 복사되기 전에 파일에서 실행할 명령을 지정합니다. 588 페이지의 『테스트 프로젝트의 특수 변수』의 내용을 참조하십시오.
_SUPPRESS_ENV_OUTPUT	시스템이 로그에서 환경 메시지를 생략하도록 지정합니다. 기본적으로 이 변수는 설정되지 않으며 단계 명령을 실행하기 전에 환경의 모든 변수 값이 인쇄됩니다. 값은 단계 로그에서 ENV 항목으로 나타납니다. 변수는 다음 값으로 설정할 수 있습니다. <ul style="list-style-type: none"> <li>• ALWAYS: 항상 ENV 메시지를 생략합니다.</li> <li>• 기타 값: ENV 메시지를 생략합니다. 그러나 이 명령에 실패하면 명령 메시지 다음에 ENV 메시지가 인쇄됩니다. 이 정보는 명령 실행 실패를 디버깅하는 경우에 유용합니다.</li> </ul>
_SUPPRESS_AGENT_LOG_OUTPUT	1로 설정하면 에이전트가 로그 데이터를 엔진으로 전송하지 못하게 합니다. _SUPPRESS_LOG_OUTPUT과 비교하십시오. 여기서 로그 데이터는 에이전트에서 전송되지만 엔진에 의해 삭제됩니다. 참고: 이 변수를 사용하면 필터 일치를 사용하지 못합니다.
_SUPPRESS_LOG_OUTPUT	1로 설정하면 엔진이 에이전트에서 수신된 로그 출력을 거의 모두 삭제합니다. 일부 콘솔 로그 메시지는 남아 있습니다. 필터 일치사항이 표시됩니다.
_TIMEOUT	프로젝트의 모든 또는 한 단계에 대해 제한시간 특성을 대체하는 값입니다.

변수	컨텐츠
_TRAP	현재 단계가 실패하면 실행될 문자열입니다. 실행 파일 또는 명령 이름으로 문자열을 설정할 수 있습니다. 참고: 단계가 실패하면 콘솔과 에이전트 간의 연결이 단히므로 명령의 출력이 콘솔에 리턴되지 않습니다. _TRAP를 통해 실행된 명령의 출력을 보유하려면, 나중 검색을 위해 명령이 파일에 출력을 기록하게 하십시오.
_USE_BFCREDS	1로 설정되면, 시스템은 서버에 접속된 서버 권한에 저장된 신임을 사용하지 않고 <b>사용자의 로그인 신임</b> 을 사용하여 서버에 로그인합니다. 시스템은 프로젝트에서 명령을 실행하기 위해 프로젝트를 시작한 사용자의 관리 콘솔 로그인 신임 정보를 사용합니다. 단일 단계에 대해 또는 전체 프로젝트에 대해 이 변수를 설정할 수 있습니다. 참고: LDAP/Active Directory 인증을 사용 중인 경우, _USE_BFCREDS 함수가 작동하려면 <b>사용자 인증을 로컬로 저장</b> 시스템 설정을 예(기본값)로 설정해야 합니다. 설정이 예이면 시스템이 사용자 인증 정보를 암호화된 양식으로 캐시한 다음 _USE_BFCREDS로 사용하도록 사용자 인증 정보에 액세스할 수 있습니다. 팁: Windows에서는 _USE_BFCREDS_DOMAIN 변수도 설정할 것을 고려하십시오.
_USE_BFCREDS_DOMAIN(Windows만 해당)	1로 설정되면 시스템은 _USE_BFCREDS가 서버에 로그인하는 데 사용하는 로그인 신임 정보에 추가로 <b>사용자의 도메인</b> 을 사용합니다.

변수	컨텐츠
<code>_XSTREAM_PROTOCOL</code> 유형	<p>에이전트 간 직접 파일 전송을 사용합니다.</p> <p><b>중요사항:</b> 일부 운영 체제의 에이전트의 경우 직접 파일 전송에 대한 지원이 제한되거나 지원되지 않습니다. 196 페이지의 『에이전트 간 직접 파일 전송 구성』의 내용을 참조하십시오.</p> <p>엔진, 전송 에이전트 및 수신 에이전트는 모두 직접 파일 전송을 지원합니다. 해당사항이 없는 경우, <code>_XSTREAM_PROTOCOL</code>은 경고 없이 무시되며 일반 파일 전송 메소드가 사용됩니다.</p> <p>수신 에이전트는 전송 에이전트 호스트에서 TCP 연결을 작성할 수 있어야 합니다. 해당 연결이 존재하는 경우 연결을 허용하도록 방화벽을 구성해야 합니다.</p> <p>프로토콜 유형은 인코딩 데이터의 메소드를 판별하며 다음 중 하나입니다.</p> <p><b>AES-CBC</b></p> <p>암호화적으로 강한 알고리즘은 데이터를 인코딩하는 데 사용됩니다. 두 에이전트는 모두 OpenSSL을 사용하여 컴파일되고 엔진과의 통신에서 SSL을 사용해야 합니다. 암호화 키는 엔진에서 얻습니다.</p> <p><b>PRNG</b> 의사 난수 생성기는 파일 콘텐츠를 알 수 없게 하기 위해 사용됩니다.</p> <p><b>PLAIN</b> 파일은 인코딩하지 않고 현상태대로 전송됩니다.</p>

## 환경 스냅샷

환경 스냅샷을 찍어 변경 또는 수정할 환경의 새 인스턴스를 빠르게 작성합니다.

### 환경 스냅샷 개요

이 주제를 검토하면 환경 스냅샷에 대해 알게 되며 환경 스냅샷 사용 방법을 이해할 수 있습니다.

### 환경 스냅샷 유스 케이스

다음 예제는 환경 스냅샷의 일부 공통 유스 케이스에 대해 설명합니다.

- 환경 스냅샷을 찍어 환경 구성을 변경하거나 새 도구 또는 스크립트 테스트를 수행합니다.
- 환경 스냅샷을 임시 백업으로 또는 공식 아카이브의 일부로 저장하십시오.

- 환경 스냅샷을 찍어 외부 또는 내부 릴리스와 같은 마일스톤과 일치하는 적시 환경 구성을 캡처합니다.

## 환경 스냅샷 개념 및 용어

UI에서 스냅샷은 환경에 대한 작업을 위한 몇 가지 새 개념과 용어를 소개합니다.


**환경 스냅샷:** 스냅샷은 기존 환경의 새 인스턴스입니다. 스냅샷에 대해 기억해야 할 몇 가지 중요 사항은 다음과 같습니다.

- 스냅샷은 별도 환경 오브젝트입니다. 스냅샷 세트의 특정 스냅샷을 변경해도 세트의 다른 스냅샷에 영향을 주지 않습니다.
- 스냅샷은 사본이 아닙니다.

환경과 연관된 오브젝트의 스냅샷을 찍으면 스냅샷이 오브젝트의 별도 인스턴스를 작성합니다. 사본은 오브젝트 간 관계를 맵핑하며 새 오브젝트를 작성하지 않습니다.


- 스냅샷은 환경의 변경내용이 아닙니다.
  - 스냅샷은 두 환경 스냅샷 간의 변경사항 비교를 지원하지 않습니다.
  - 환경 스냅샷 변경사항은 소스 제어 시스템에서와 같이 버전 번호로 추적 또는 식별되지 않습니다. 그러나 버전 번호를 포함하는 스냅샷 이름 지정 구조(예: 7.5.0, 3.4.01)를 사용하여 환경 스냅샷과 마일스톤을 상관시킬 수 있습니다.

**스냅샷 세트:** 스냅샷 세트는 특정 기본 스냅샷의 하위인 모든 환경 스냅샷의 세트입니다. 최소한 세트에는 기본 또는 상위 스냅샷과 하위 스냅샷이 포함됩니다. UI에서 환경


이름 옆의 스냅샷 아이콘()은 환경에 대한 스냅샷 세트가 존재함을 나타냅니다.

**기본 스냅샷:** 처음에는 모든 환경의 스냅샷 이름이 기본 스냅샷입니다. 기본 스냅샷을 다른 이름으로 변경할 수 있습니다. 기본 스냅샷은 스냅샷 세트의 상위입니다.

**기본 환경 스냅샷:** 기본 환경 스냅샷은 현재 실행 중인 환경입니다. 세트에서 하나의 스냅샷만 기본값이 될 수 있습니다. 기본 스냅샷을 지정하지 않는 경우 기본값은 기본 스냅샷입니다.

- UI에서 기본 스냅샷은 환경 목록의 최상위 레벨에 표시됩니다. 환경 목록을 표시하려면 환경을 선택하십시오.
- 스냅샷이 있는 환경을 선택하면 목록 상자에서 다른 환경 스냅샷을 선택하지 않는 한 기본 환경 스냅샷이 사용됩니다.
- 환경 스냅샷 세트의 다른 스냅샷에 액세스하고 이에 대한 작업을 수행하려면 스냅샷 아이콘()을 클릭해야 합니다.

## 환경 스냅샷 보기

스냅샷 아이콘  을 선택하여 스냅샷 보기를 표시합니다. UI에서 스냅샷 보기는 다음과 같은 세트 내 스냅샷의 계층 구조를 보여줍니다.

- 기본 스냅샷은 최상위 레벨이며 고유 이름을 지정하지 않는 경우 기본 스냅샷이라고 합니다.
- 모든 환경 스냅샷은 기본 스냅샷의 하위입니다. 기본 스냅샷이 같은 하위는 스냅샷 열의 동일한 레벨에서 들어옵니다.
- 하위 스냅샷에서 작성된 환경 스냅샷은 하위 스냅샷의 하위가 되며 스냅샷 열의 다음 레벨에서 들어옵니다.

## 환경 스냅샷 계획

기본 환경 스냅샷을 선택하고 환경 스냅샷의 이름을 지정하는 몇 가지 우수 사례를 검토하십시오.

- 세트의 기본 스냅샷 선택 전략

UI는 스냅샷 세트에 대해 하나의 기본 또는 현재 환경 스냅샷만 인식합니다. 기본 스냅샷 선택을 위해서는 다음과 같은 동일 전략을 사용하십시오.

- 기본 스냅샷을 기본 스냅샷으로 사용

이 전략을 사용하여 스냅샷을 적시 백업으로 작성하고 백업된 환경 스냅샷을 변경하지 않습니다. 기본 스냅샷을 변경합니다.

- 최신 스냅샷을 기본 스냅샷으로 사용

이 전략을 사용하는 경우 새 환경을 작성하는 이유는 새 환경을 새 기본 환경 스냅샷으로 설정하기 위해서입니다. 기본 스냅샷 또는 이전 환경 스냅샷은 변경하지 않습니다.

- 세트의 스냅샷 이름 지정 규칙 식별

환경 스냅샷 이름은 환경 스냅샷 세트에서 고유해야 합니다.

환경 스냅샷 이름을 작성하려면 다음 기준을 사용하십시오.

- 이름은 스냅샷 사용법 또는 용도를 설명해야 합니다.
- 이름 지정 규칙은 정의된 표준을 따라야 합니다. 스냅샷 탭의 주석 상자를 사용하여 이름 지정 규칙을 설명할 수 있습니다.

- 세트에 단일 환경 이름 사용

환경 스냅샷을 작성한 후 환경의 이름을 변경할 옵션이 있습니다. 환경 이름을 변경하는 경우 모든 환경 스냅샷에 대해 업데이트됩니다.






## 환경 스냅샷 작성

환경 스냅샷을 작성하면 환경의 새 인스턴스가 작성됩니다. 스냅샷은 사본이 아닌, 새 환경입니다.

환경 스냅샷은 환경 또는 환경 스냅샷에서 작성할 수 있습니다.

**팁:** 스냅샷을 작성하고 기본 스냅샷을 변경하려면 사용자에게 필수 권한이 있어야 합니다. 385 페이지의 『스냅샷 권한의 액세스 그룹 확인 및 편집』의 내용을 참조하십시오.

- 스냅샷을 찍을 환경 또는 환경 스냅샷 옆에 있는 **편집** 아이콘을 클릭하십시오.
  - 기본 환경 스냅샷을 찍으려면 환경 목록(환경)에서 최상위 레벨 스냅샷 옆에 있는 편집 아이콘  을 클릭하십시오.
  - 기본이 아닌 환경 스냅샷을 찍으려면 스냅샷 아이콘  을 클릭하십시오. 스냅샷 보기에는 세트의 환경 스냅샷이 표시됩니다. 기본이 아닌 환경 스냅샷 옆에 있는 편집 아이콘  을 클릭하십시오.
- 새 스냅샷 작성을 클릭하십시오.
- 스냅샷 탭에 이름을 입력하십시오. 이름은 환경 스냅샷을 찍는 모든 오브젝트에 지정됩니다.

이름은 환경 스냅샷 세트에서 고유해야 합니다.

- 환경 스냅샷을 작성할 때 스냅샷을 찍을 Build Forge 오브젝트를 선택하십시오. 선택할 수 있는 오브젝트는 다음 표에 설명되어 있습니다.

오브젝트	설명
기본값	UI에서 기본 스냅샷은 환경 목록의 최상위 레벨에 표시됩니다. 환경 목록을 표시하려면 환경을 선택하십시오.
다음 환경 포함	다음 환경 포함 유형을 사용하여 환경에 포함되는 환경의 스냅샷을 찍습니다.

- 저장을 클릭해서 환경 스냅샷을 저장하십시오.

## 기본 환경 스냅샷 변경

기본 환경 스냅샷은 스냅샷 세트의 최상위 레벨 스냅샷이며 환경 목록(환경)에 표시됩니다.

**팁:** 스냅샷을 작성하고 기본 스냅샷을 변경하려면 사용자에게 필수 권한이 있어야 합니다. 385 페이지의 『스냅샷 권한의 액세스 그룹 확인 및 편집』의 내용을 참조하십시오.

기본 환경 스냅샷을 변경하려면 새 기본값으로 설정할 스냅샷의 스냅샷 정의를 편집하십시오.

1. 환경을 선택하십시오.
2. 환경 목록에서 기본 환경 스냅샷의 스냅샷 아이콘을 클릭하십시오.
3. 스냅샷 목록에서 새 기본값으로 설정할 환경 스냅샷의 편집 아이콘을 클릭하십시오.
4. 기본값으로 설정을 클릭하십시오.
5. 중요: 팝업에서 확인 또는 취소를 선택하십시오.



확인	참조 업데이트: 이전 기본값을 참조하는 오브젝트에 대해 이전 기본 환경 스냅샷의 참조를 새 기본값으로 업데이트합니다.
취소	참조를 업데이트하지 않음: 이전 기본값을 참조하는 오브젝트에 대해 참조를 새 기본 환경 스냅샷으로 업데이트하지 않습니다.

## 환경 스냅샷 이름 변경

환경 스냅샷의 스냅샷 이름을 변경할 수 있으며 환경 스냅샷을 작성할 때 스냅샷을 찍기로 선택한 오브젝트의 스냅샷 이름도 변경할 수 있습니다.

기본 스냅샷의 경우 이 옵션을 사용하여 기본 이름인 기본 스냅샷을 단일 환경 스냅샷 전용 또는 현재와 미래의 모든 환경에 대한 다른 스냅샷 이름으로 변경할 수 있습니다.

스냅샷 이름을 변경하려면 다음을 수행하십시오.

1. 환경을 선택하십시오.
2. 환경 목록에서 기본 환경 스냅샷에 대한 스냅샷 아이콘  을 클릭하십시오.
3. 스냅샷 목록에서 환경 스냅샷에 대한 편집 아이콘  을 클릭하십시오.
4. 스냅샷 탭을 선택하십시오.
5. 이름에 새 이름을 입력하십시오.
6. 선택적: 주석에 주석을 입력하십시오.
7. 중요: 팝업에서 확인 또는 취소를 선택하십시오.

확인	<p>환경 스냅샷 이름과 기타 스냅샷 오브젝트 이름 변경: 환경 스냅샷을 작성할 때 스냅샷을 찍기로 선택한 오브젝트의 경우 이러한 오브젝트와 환경 스냅샷의 이름을 변경합니다.</p> <p>기본 스냅샷의 경우: 모든 현재 환경 스냅샷과 모든 미래 환경 스냅샷의 이름인 기본 스냅샷을 변경합니다.</p>
취소	<p>환경 스냅샷 이름은 변경하고 기타 스냅샷 오브젝트 이름은 변경하지 않음: 환경 스냅샷을 작성할 때 스냅샷을 찍기로 선택한 오브젝트의 경우 이러한 오브젝트의 이름을 변경하지 마십시오. 환경 스냅샷 이름만 변경하십시오.</p> <p>기본 스냅샷의 경우: 모든 현재 환경 스냅샷과 모든 미래 환경 스냅샷의 이름인 기본 스냅샷을 유지합니다.</p>


## 스냅샷 액세스 및 보기

환경 스냅샷을 작성하면 두 개 이상의 환경(기본 스냅샷 환경과 새 환경 스냅샷)을 포함하는 스냅샷 세트가 작성됩니다.



스냅샷 세트의 모든 환경 스냅샷을 보려면 다음을 수행하십시오.

1. 환경을 선택하십시오.

환경 목록에는 환경과 환경 스냅샷의 목록이 표시됩니다. 최상위 레벨 스냅샷이 기본 환경 스냅샷입니다.

2. 스냅샷 아이콘  을 클릭하여 스냅샷 세트에서 환경 스냅샷을 표시하십시오.

스냅샷 보기에서 다음을 수행할 수 있습니다.


- 새 환경 스냅샷을 작성하십시오. 시작하려면 편집 아이콘  을 클릭하십시오.
- 환경의 기본 스냅샷을 변경하십시오. 새 기본 스냅샷에 대해 편집 아이콘  을 클릭하고 기본값으로 설정을 클릭하십시오.
- 표준 환경의 경우와 같이 환경 스냅샷 정의를 편집하십시오.

## 환경 스냅샷 삭제


환경 삭제 옵션을 사용하여 환경 스냅샷을 삭제할 수 있습니다.

다른 오브젝트가 사용하는 환경은 삭제할 수 없습니다(예를 들어, 다른 환경에 포함되거나 프로젝트, 단계, 스케줄 또는 서버에서 사용하는 경우).

환경 스냅샷을 삭제하려면 다음을 수행하십시오.

1. 환경을 선택하십시오.
2. 환경 목록에서 기본 스냅샷에 대한 스냅샷 아이콘  을 클릭하십시오.

스냅샷 보기에는 세트의 환경 스냅샷이 표시됩니다.

3. 삭제할 환경 스냅샷 옆에 있는 편집 아이콘  을 클릭하십시오.
4. 환경 삭제를 클릭하십시오.



---

## 제 19 장 프로젝트에 대한 작업

이 주제는 관리 콘솔에서 프로젝트를 작성 및 관리하는 방법에 대해 설명합니다.

---

### 프로젝트 정보

프로젝트는 고유 환경 그룹과 서버 특성이 있는 단계의 실행 파일 세트입니다.

프로젝트 이외에도 라이브러리를 작성할 수 있습니다. 라이브러리는 프로젝트와 유사하지만 단계가 실행하는 위치를 판별하기 위한 선택기를 갖지 않습니다.

### 프로젝트 패널 정보

새 프로젝트를 작성하고 기존 프로젝트를 보거나 편집하려면 프로젝트 패널을 사용하십시오. 프로젝트 패널에 액세스하려면 왼쪽 메뉴에서 프로젝트를 클릭하십시오.

프로젝트 패널은 다음 탭을 제공합니다.

- 『세부사항 탭』
- 352 페이지의 『태그 탭』
- 352 페이지의 『레지스터 탭』
- 353 페이지의 『노트 탭』
- 353 페이지의 『스냅샷 탭』

### 세부사항 탭

세부사항 탭에서 프로젝트에 관한 다양한 특성을 지정할 수 있습니다.

IBM UI 구성 콘솔 보고서 로그아웃: Root User

프로젝트
프로젝트 추가
도움말

필터
1 표시 중 - 2/2
모두 표시

프로젝트	스냅샷	태그	클래스	환경	선택기	액세스
HelloWorld	기본 스냅샷	BUILD_\$B	스크래치		HelloWorldSelector	----- 개발자
Say hi	기본 스냅샷	BUILD_\$B	스크래치		My Selector	----- 개발자

프로젝트: 프로젝트 추가
저장
새 스냅샷 작성
기본값으로 설정
프로젝트 복사
프로젝트 삭제
정리

프로젝트 세부사항
태그
레지스터
노트 (0)
스냅샷

이름:

최대 스레드 수:

실행 한계:

패스 체인:

실패 체인:

클래스:

선택기:

환경:

고정:

알림 시작:

패스 알림:

실패 알림:

액세스:

☐ 사용 안함

**이름** 프로젝트의 이름입니다. 시스템이 이 이름을 사용하여 목록 및 데이터베이스에서 프로젝트를 참조합니다.

프로젝트 이름은 프로젝트가 실행될 때 프로젝트 디렉토리를 생성하는 데 사용됩니다. 프로젝트는 다른 운영 체제에서 실행되는 단계를 포함할 수 있으므로 프로젝트 이름에 특수 문자와 기호를 사용하지 마십시오. 프로젝트 이름에 올바른 디렉토리 이름을 생성하는 문자를 사용해야 하는 경우 해당 문자를 올바른 상대 디렉토리 문자 시스템 구성 설정에 추가하십시오. 이 설정에 나열되는 문자는 프로젝트 디렉토리를 작성할 때 밑줄(\_)로 변환됩니다. 앰퍼샌드(&)를 사용하면 의도하지 않게 HTML 엔티티가 프로젝트 이름의 파트로 생성될 수 있습니다.

**액세스** 프로젝트를 보고 사용하도록 허용된 액세스 그룹입니다. 액세스 특성은 권한과 함께 사용되어 사용자가 수행할 수 있는 작업을 판별합니다. 예를 들어, 작업을 실행하려면 프로젝트에 지정된 액세스 그룹의 구성원이어야 하며 작업 실행 권한을 가진 그룹의 구성원이기도 해야 합니다. 액세스 그룹에 대한 자세한 정보는 241 페이지의 『액세스 개요』를 참조하십시오.

### 사용 안함

프로젝트를 사용 안함 설정하려면 이 선택란을 선택하십시오. 사용자가 사용 안함 설정된 프로젝트를 실행하려고 하면 Rational Build Forge가 해당 프로젝트가 비활성임을 나타내는 메시지를 표시하고 프로젝트를 실행하지 않습니다.

### 최대 스레드 수

프로젝트가 실행하도록 허용된 최대 병렬 프로세스 수입니다. 이 필드를 사용

하여 프로젝트가 너무 많은 시스템 자원을 사용하지 않도록 하십시오. 각 스프레드 사용 단계 및 스프레드 사용 단계를 자체적으로 실행할 수 있는 인라인 프로젝트로 인해 병렬 프로세스가 발생할 수 있지만 모든 해당 프로세스는 상위 프로젝트의 최대값과 비교하여 계수됩니다. 시스템은 최대 스프레드 값에 도달하면 새 병렬 프로세스 실행을 중지하고 프로젝트의 병렬 프로세스 수가 최대 스프레드 값 아래로 떨어질 때까지 기다렸다가 계속합니다. 스프레딩에 대한 자세한 정보는 403 페이지의 『스레딩: 병렬로 단계 실행』의 내용을 참조하십시오.

#### 실행 한계

실행 한계 특성은 동시에 허용되는 최대 프로젝트 작업 수를 설정합니다.

- 프로젝트를 실행하는 경우 활성 작업 수가 실행 한계와 같으면 새 작업은 하나 이상의 작업이 완료될 때까지 대기 큐에 있습니다.
- 스케줄이 프로젝트를 실행하려고 시도할 때 활성 작업 수가 실행 한계와 같으며 하드 실행 한계 시스템 구성 설정의 값이 예이면 시스템은 새 작업을 실행하지 않습니다. 하드 실행 한계가 아니므로 설정되면 시스템은 스케줄된 빌드의 실행 한계 설정을 무시합니다.
- 인라인을 통해 실행되는 프로젝트는 원래 프로젝트의 인스턴스로 간주되지 않으며 실행 한계 계수에 포함되지 않습니다.

**클래스** 각 프로젝트는 한 클래스에 지정되어야 하며, 이 클래스는 작업 그룹에 글로벌 특성을 지정합니다. 자세한 정보는 370 페이지의 『클래스』의 내용을 참조하십시오.

**선택기** 프로젝트의 서버를 선택할 때 사용할 선택기의 이름입니다. 시스템은 이 선택기를 자체 선택기를 지정하지 않는 프로젝트의 모든 단계에 기본값으로 사용합니다. 299 페이지의 『선택기』의 내용을 참조하십시오. 선택기를 지정하지 않은 경우, 프로젝트는 프로젝트 패널 대신 라이브러리 패널에 추가됩니다. 라이브러리는 호출 단계의 선택기를 사용합니다. 즉, 해당 단계가 선택기를 갖지 않는 경우 라이브러리는 호출 프로젝트의 선택기를 사용합니다.

#### 패스 체인, 실패 체인

프로젝트 빌드가 패스 또는 실패할 때 실행되는 프로젝트를 선택하십시오. 프로젝트 레벨에서 패스/실패 체인을 설정하면 프로젝트의 패스/실패 상태를 기반으로 별도의 패스/실패 조치를 호출할 수 있습니다. 이 기능은 프로젝트의 단계 레벨에서 패스/실패 조치를 설정하는 것과 유사합니다. 프로젝트 레벨에서 패스/실패 조치는 단계 상태가 아닌 프로젝트 실행 상태로 트리거됩니다.

**환경** 서버 환경 이후 및 단계 환경 이전에 적용할 환경입니다. 환경이 함께 작동하는 방법에 대한 자세한 정보는 323 페이지의 『변수 정보』를 참조하십시오.

**고정** 기본 프로젝트 선택기를 사용하는 프로젝트의 모든 단계가 동일한 서버에 있게

하고, 사용 중인 경우 사용 가능하게 될 때까지 기다리게 하려면 고정 선택란을 사용하십시오. 이 옵션에 대한 자세한 정보는 356 페이지의 『단계를 서버에 고정』를 참조하십시오.

#### 시작 알림, 패스 알림, 실패 알림

이들 필드 중 하나 또는 모두에서 액세스 그룹을 선택하여 시스템이 프로젝트 시작, 패스 및/또는 실패 시 이메일로 알림을 보내게 하려면 이 필드를 사용하십시오.

#### 태그 탭

프로젝트의 빌드 태그를 관리하려면 태그 탭을 사용하십시오.

#### 태그 형식

일반 텍스트 및 태그 변수 참조를 사용하여 프로젝트의 태그를 정의하는 문자열입니다. 태그 형식에 대한 자세한 정보는 413 페이지의 『작업 중에 빌드 태그 변경』을 참조하십시오.

#### 태그 동기화

두 프로젝트의 태그 변수를 동기화하십시오. 현재 프로젝트와 태그 변수를 동기화할 프로젝트를 선택하십시오. 두 프로젝트가 동기화되면, 동일한 풀에서 변수를 가져오므로 프로젝트가 차례로 실행될 때 한 프로젝트는 값 1을 가져오고 다음 프로젝트는 값 2를 가져옵니다. 자세한 정보는 360 페이지의 『태그 동기화』의 내용을 참조하십시오.

#### 태그 이름

변수 이름입니다. 태그 형식에서 태그 변수를 사용할 때 `$<태그 이름>` 양식을 사용하여 이름을 참조하십시오. 예를 들어, MainVer 및 B 변수를 사용하는 태그를 작성하려면 "Build\_\$MainVer.\$B" 태그 형식을 사용하여 Build\_005.1 같은 태그를 가져오십시오.

**초기값** 태그 변수 값을 설정합니다. 자동 증분 옵션을 사용하지 않는 경우, 변수는 사용자가 변경할 때까지 이 값을 보유합니다.

**채우기** 채우기 값을 없음 이외의 값으로 선택하는 경우, 시스템은 필요하면 태그에서 사용될 때 변수 값에 선행 제로를 추가하여 숫자의 수가 채우기 값과 같아지도록 합니다. 예를 들어, 변수가 현재 2이고 채우기가 3인 경우 시스템은 값을 002로 렌더링합니다. 채우기는 1 - 8 범위일 수 있습니다.

#### 자동 증가

예로 설정된 경우, 시스템은 모든 프로젝트 작업에 대해 변수 값을 1씩 증가합니다.

#### 레지스터 탭

빌드 사이에 지속되는 정보를 저장하려면 프로젝트 레지스터를 사용하십시오.



## 레지스터

프로젝트 레지스터의 이름입니다.

컨텐츠 레지스터의 값입니다.

추가 정보는 416 페이지의 『프로젝트 레지스터』를 참조하십시오.

## 노트 탭

프로젝트에 관한 관심 항목을 저장하려면 노트 탭을 사용하십시오.

각 노트에 대해 탭은 다음을 표시합니다.

- 사용자가 노트를 작성한 날짜 및 시간
- 노트를 입력한 사용자
- 노트


## 스냅샷 탭

프로젝트 스냅샷은 프로젝트의 인스턴스입니다. 스냅샷 탭에서는 스냅샷 이름을 제공합니다.

스냅샷 이름 및 스냅샷에 대한 주석을 보거나 변경하려면 이 탭을 사용하십시오.

---

## 프로젝트 특성 변경

프로젝트 레벨 특성을 변경하려면 프로젝트를 선택한 후, 원하는 프로젝트의 이름 옆에 있는 편집 아이콘()을 클릭하십시오.

### 프로젝트 이름

프로젝트의 이름입니다. 시스템이 이 이름을 사용하여 목록 및 데이터베이스에서 프로젝트를 참조합니다.

프로젝트 이름은 프로젝트가 실행될 때 프로젝트 디렉토리를 생성하는 데 사용됩니다. 프로젝트는 다른 운영 체제에서 실행되는 단계를 포함할 수 있으므로 프로젝트 이름에 특수 문자와 기호를 사용하지 마십시오. 프로젝트 이름에 올바른 디렉토리 이름을 생성하는 문자를 사용해야 하는 경우 해당 문자를 올바른 디렉토리 문자 시스템 구성 설정에 추가하십시오. 이 설정에 나열되는 문자는 프로젝트 디렉토리를 작성할 때 밑줄(\_)로 변환됩니다. 앰퍼샌드(&)를 사용하면 의도하지 않게 HTML 엔티티가 프로젝트 이름의 파트로 생성될 수 있습니다.

**액세스** 프로젝트를 보고 사용하도록 허용된 액세스 그룹입니다. 액세스 특성은 권한과 함께 사용되어 사용자가 수행할 수 있는 작업을 판별합니다. 예를 들어, 작업을 실행하려면 프로젝트에 지정된 액세스 그룹의 구성원이어야 하며 작업 실행

권한을 가진 그룹의 구성원이기도 해야 합니다. 액세스 그룹에 대한 자세한 정보는 241 페이지의 『액세스 개요』를 참조하십시오.

### 태그 형식

일반 텍스트 및 태그 변수 참조를 사용하여 프로젝트의 태그를 정의하는 문자열입니다. 태그 형식에 대한 자세한 정보는 413 페이지의 『작업 중에 빌드 태그 변경』을 참조하십시오.

### 태그 동기화

두 프로젝트의 태그 변수를 동기화하십시오. 현재 프로젝트와 태그 변수를 동기화할 프로젝트를 선택하십시오. 두 프로젝트가 동기화되면, 동일한 폴에서 변수를 가져오므로 프로젝트가 차례로 실행될 때 한 프로젝트는 값 1을 가져오고 다음 프로젝트는 값 2를 가져옵니다. 자세한 정보는 360 페이지의 『태그 동기화』의 내용을 참조하십시오.

### 최대 스레드 수

프로젝트가 실행하도록 허용된 최대 병렬 프로세스 수입니다. 이 필드를 사용하여 프로젝트가 너무 많은 시스템 자원을 사용하지 않도록 하십시오. 각 스레드 사용 단계 및 스레드 사용 단계를 자체적으로 실행할 수 있는 인라인 프로젝트로 인해 병렬 프로세스가 발생할 수 있지만 모든 해당 프로세스는 상위 프로젝트의 최대값과 비교하여 계수됩니다. 시스템은 최대 스레드 값에 도달하면 새 병렬 프로세스 실행을 중지하고 프로젝트의 병렬 프로세스 수가 최대 스레드 값 아래로 떨어질 때까지 기다렸다가 계속합니다. 스레딩에 대한 자세한 정보는 403 페이지의 『스레딩: 병렬로 단계 실행』의 내용을 참조하십시오.

### 실행 한계

실행 한계 특성은 동시에 허용되는 최대 프로젝트 작업 수를 설정합니다.

- 프로젝트를 실행하는 경우 활성 작업 수가 실행 한계와 같으면 새 작업은 하나 이상의 작업이 완료될 때까지 대기 큐에 있습니다.
- 스케줄이 프로젝트를 실행하려고 시도할 때 활성 작업 수가 실행 한계와 같으며 하드 실행 한계 시스템 구성 설정의 값이 예이면 시스템은 새 작업을 실행하지 않습니다. 하드 실행 한계가 아니므로 설정되면 시스템은 스케줄된 빌드의 실행 한계 설정을 무시합니다.
- 인라인을 통해 실행되는 프로젝트는 원래 프로젝트의 인스턴스로 간주되지 않으며 실행 한계 계수에 포함되지 않습니다.

**클래스** 각 프로젝트는 클래스에 지정되어야 하며, 클래스는 프로젝트 그룹에 글로벌 특성을 지정합니다. 자세한 정보는 370 페이지의 『클래스』의 내용을 참조하십시오.

**선택기** 프로젝트의 서버를 선택할 때 사용할 선택기의 이름입니다. 시스템은 이 선택기를 자체 선택기를 지정하지 않는 프로젝트의 모든 단계에 기본값으로 사용합니다. 299 페이지의 『선택기』의 내용을 참조하십시오. 선택기를 지정하지 않은

경우, 프로젝트는 프로젝트 패널 대신 라이브러리 패널에 추가됩니다. 라이브러리는 호출 단계의 선택기를 사용합니다. 즉, 해당 단계가 선택기를 갖지 않는 경우 라이브러리는 호출 프로젝트의 선택기를 사용합니다.

#### 패스/실패 체인

프로젝트 빌드가 패스 또는 실패할 때 실행되는 프로젝트를 선택하십시오. 프로젝트 레벨에서 패스/실패 체인을 설정하면 프로젝트의 패스/실패 상태를 기반으로 별도의 패스/실패 조치를 호출할 수 있습니다. 이 기능은 프로젝트의 단계 레벨에서 패스/실패 조치를 설정하는 것과 유사합니다. 프로젝트 레벨에서 패스/실패 조치는 단계 상태가 아닌 프로젝트 실행 상태로 트리거됩니다.

**환경** 서버 환경 이후 및 단계 환경 이전에 적용할 환경입니다. 환경이 함께 작동하는 방법에 대한 자세한 정보는 321 페이지의 『환경 상속』를 참조하십시오.

**고정** 기본 프로젝트 선택기를 사용하는 프로젝트의 모든 단계가 동일한 서버에 있게 하고, 사용 중인 경우 사용 가능하게 될 때까지 기다리게 하려면 고정 선택란을 사용하십시오. 이 옵션에 대한 자세한 정보는 356 페이지의 『단계를 서버에 고정』를 참조하십시오.


#### 시작 알림, 패스 알림, 실패 알림

이들 필드 중 하나 또는 모두에서 액세스 그룹을 선택하여 시스템이 프로젝트 시작, 패스 및/또는 실패 시 이메일로 알림을 보내게 하려면 이 필드를 사용하십시오.

---

## 프로젝트 복사

기존 프로젝트의 사본을 작성하려면 다음 단계를 완료하십시오.

1. 프로젝트를 선택하십시오.
2. 복사하려는 프로젝트 옆에 있는 편집 아이콘()을 선택하십시오.
3. 프로젝트 복사를 클릭하십시오.

프로젝트를 복사할 때 시스템은 기존 프로젝트의 다음 참조사항을 새 프로젝트에 복사합니다.

- 단계 및 단계의 세부사항 탭에 나열되는 모든 특성
- 프로젝트의 세부사항 탭에 나열되는 모든 프로젝트 특성(예: 프로젝트의 클래스, 선택기 및 기타 특성)
- 프로젝트의 태그 형식(프로젝트 특성 패널의 태그 탭에 있음)

시스템은 다음 특성을 복사하지 않습니다.

- 태그 변수(프로젝트 특성 패널의 태그 탭에 있음). 태그 형식에서 사용되는 태그 변수가 복사됨
- 프로젝트 레지스터(프로젝트 특성 패널의 레지스터 탭에 있음)

- 단계 노트(단계 특성 패널의 노트 탭에 있음)
- 프로젝트 노트(프로젝트 특성 패널의 노트 탭에 있음)

## 프로젝트 삭제


프로젝트에 연관된 작업이 있는지 여부에 따라 프로젝트를 삭제하는 두 가지 방법이 있습니다.

프로젝트를 삭제하려면 다음 옵션 중 하나를 선택하십시오.

### 프로젝트 삭제 단추

**참고:** 프로젝트 삭제는 실행 취소할 수 없습니다.


프로젝트 삭제 단추는 작업이 없는 프로젝트를 삭제합니다. 이 단추로 프로젝트를 삭제하려면 먼저 프로젝트의 모든 작업을 삭제해야 합니다. 이 단추는 프로젝트 특성 편집 페이지 및 프로젝트 단계 목록에 있습니다. 프로젝트 특성 편집 페이지를 보려면 다음 단계를 완료하십시오.

1. 왼쪽 메뉴에서 프로젝트를 선택하십시오.
2. 원하는 프로젝트 이름 옆에 있는 편집 아이콘()을 클릭하십시오.

### 정리 단추


**참고:** 정리 조작은 프로젝트가 잠겨있는 경우에도 프로젝트를 삭제합니다. 또한 프로젝트의 정리를 실행 취소할 수 없습니다.

정리 단추는 프로젝트 및 연관된 모든 작업을 Build Forge 데이터베이스에서 삭제합니다. 시스템은 프로젝트를 정리(clobbering)하기 전에 확인을 요구합니다. 이 단추는 프로젝트 특성 편집 페이지에 있습니다. 프로젝트 특성 편집 페이지를 보려면 다음 단계를 완료하십시오.

1. 왼쪽 메뉴에서 프로젝트를 선택하십시오.
2. 원하는 프로젝트 이름 옆에 있는 편집 아이콘()을 클릭하십시오.

## 단계를 서버에 고정

선택기가 허용하는 경우 프로젝트의 단계를 서로 다른 서버에서 실행할 수 있습니다. 그러나 해당 서버를 미리 지정하는지 여부에 관계없이 모든 또는 대부분의 프로젝트 단계를 동일한 서버에서 실행할 수 있습니다. 프로젝트 레벨 고정 특성이 해당 옵션을 제공합니다.

프로젝트 레벨 특성을 보려면 프로젝트를 선택한 후, 원하는 프로젝트의 이름 옆에 있는 편집 아이콘()을 클릭하십시오.

고정(Sticky) 특성은 프로젝트에서 자체 선택기를 지정하지 않는 단계에만 적용됩니다. 단계에 기본값 이외의 다른 선택기 옵션이 있는 경우, 선택기가 프로젝트의 선택기와 동일하더라도 시스템은 해당 선택기를 사용하여 단계의 서버를 선택합니다.

고정 특성이 설정된 경우, 프로젝트는 선택기 필드가 기본값으로 설정된 모든 단계에 동일한 서버를 사용합니다. 이 특성은 프로젝트가 다시 시작해도 지속됩니다.

시스템은 인라인 프로젝트를 시작할 때 인라인 프로젝트의 선택기를 인라인 단계에 대한 기본 선택기로 사용합니다. 호출 프로젝트의 고정 특성은 인라인 프로젝트에 영향을 주지 않으며, 인라인 프로젝트는 자체 고정 특성이 설정된 경우 이를 준수합니다.

시스템은 인라인 라이브러리를 시작할 때 다음 규칙을 준수합니다.

- 인라인 라이브러리(고정 특성이 선택되지 않음)는 호출 단계의 선택기를 인라인 단계의 기본 선택기로 사용합니다.
- 인라인 라이브러리(고정 특성이 선택됨)는 호출 단계의 서버를 인라인 단계의 기본 서버로 사용합니다.

참고: `.bset server` 명령을 사용하여 작업 중에 프로젝트의 기본 서버를 변경할 수 있습니다. `.bset` 명령 다음에 발생하는 단계는 해당 명령으로 설정된 새 기본값을 사용하고 새 서버와 결합됩니다.

---

## 체인: 다른 프로젝트 또는 라이브러리의 조건부 실행

프로젝트 레벨에서 호출할 수 있는 다음 두 가지 유형의 체인이 있습니다.

- 패스 체인: 프로젝트가 패스할 때 실행할 프로젝트나 라이브러리를 지정합니다.
- 실패 체인: 프로젝트가 실패할 때 실행할 프로젝트나 라이브러리를 지정합니다.

이 기능은 다음과 같은 사용법을 갖습니다.

- 프로젝트 레벨에서 조건부 실행을 채택합니다. 기타 플로우 제어 기능은 단계 레벨에서 제공됩니다. 또한 한 단계가 고유한 패스 체인과 실패 체인을 가질 수 있습니다. 393 페이지의 제 20 장 『단계에 대한 작업』의 399 페이지의 『실행 플로우 제어』를 참조하십시오.
- 자주 사용되는 단계 그룹을 그에 의존하는 프로젝트와 분리하여 유지보수합니다. 라이브러리도 이 목적으로 사용할 수 있습니다.
- 프로젝트가 패스 또는 실패한 후 파일을 정리합니다.
- 소프트웨어 빌드 프로젝트가 패스할 때 자동화된 테스트 및 배치 프로젝트를 호출합니다.

## 호출 프로젝트 또는 라이브러리로부터 체인 상속

체인 형식으로 연결된 프로젝트나 라이브러리는 호출 프로젝트로부터 다음의 일부 특성을 상속합니다.

- 체인 형식으로 연결된 프로젝트는 기본적으로 호출 프로젝트의 클래스를 상속합니다. 관리 > 시스템에서 체인화 시 클래스 대체를 아니오로 설정하여 이 동작을 변경할 수 있습니다.
- 라이브러리는 고유한 선택기를 갖지 않기 때문에 체인 형식으로 연결된 라이브러리는 호출 단계의 선택기를 상속합니다. 해당 단계가 선택기를 갖지 않는 경우 라이브러리는 빌드의 선택기를 상속합니다. 체인 형식으로 연결된 라이브러리의 단계는 명시 선택기를 갖지 않는 경우에 상속된 선택기를 사용합니다.

그렇지 않은 경우 체인 형식으로 연결된 프로젝트나 라이브러리는 다음의 자체 특성으로 실행됩니다.

- 서버는 자체 선택기에 의해 지정됩니다.
- 자체 알림 설정 및 체인 설정을 포함하여 자체 특성을 사용합니다.
- 해당 환경이 호출 프로젝트의 환경 뒤에 적용됩니다. 호출 프로젝트의 변수가 이름이 바뀌어 호출된 프로젝트에서 사용할 수 있습니다. 『체인 프로젝트의 환경 변수 상속』의 내용을 참조하십시오.

## 체인 중첩

프로젝트를 체인화할 때 호출된 프로젝트가 호출 프로젝트에 중첩됩니다. 최대 중첩 레벨은 32레벨입니다. 중첩 레벨은 관리 콘솔을 실행하는 호스트에서 사용 가능한 메모리에 따라서 한계가 발생할 수도 있습니다.

## 체인 실행 및 인터럽트

체인 형식으로 연결된 프로젝트 내에서 .break 명령을 사용하는 경우 시스템이 체인 형식으로 연결된 프로젝트를 중지하지만 제어를 호출 프로젝트로 리턴하며, 호출 프로젝트가 계속됩니다. 423 페이지의 『.break』의 내용을 참조하십시오.

## 체인 프로젝트의 환경 변수 상속

프로젝트가 패스/실패 체인을 통해 실행될 때 시스템은 호출 프로젝트의 환경 변수를 적용합니다. 호출된 프로젝트는 호출 프로젝트의 환경 및 자체 환경에서 다음 순서로 변수를 설정합니다.

1. 호출된 프로젝트 서버 환경
2. 호출 프로젝트의 변수(세트에서 "BF\_" 변수 이름이 "BF\_CALLER\_"로 변경됨)
3. 호출된 프로젝트 서버 환경(호출자의 변수로 수정된 경우 두 번째 적용됨)
4. 호출된 프로젝트 환경

5. 실행되는 단계 환경(지정된 경우).

## 대기 사용 시 체인 프로젝트 취소

일반적으로 시스템은 체인 프로젝트를 취소하지 않습니다. 대기 패스 또는 대기 실패 속성을 예로 설정하면 시스템에서 체인 패스 또는 체인 실패에 대해 호출된 프로젝트를 자동으로 취소합니다. 시스템은 호출 프로젝트 또는 호출 단계가 취소될 때 호출된 프로젝트를 취소합니다.

---

## 태그 정의

시스템은 태그를 사용하여 프로젝트의 특정 작업을 식별하고 기본적으로 프로세스 활동이 발생하는 작업 디렉토리의 이름을 생성합니다. 시스템은 프로젝트의 태그 형식 특성(숫자 태그 변수는 물론 정적 텍스트를 포함할 수 있음)에서 작업 태그를 작성합니다.

프로젝트의 기본 태그 형식은 BUILD\_\$B로, 모든 프로젝트에 대해 시스템이 정의한 자동 증분 값인 기본 태그 변수 B를 사용합니다. 이 기본 태그 형식으로 인해 다음과 같이 빌드 태그 스트림이 발생합니다.

BUILD\_1

BUILD\_2

BUILD\_3

그러나 이러한 태그로 제한되지는 않습니다. 자체 태그 변수를 정의하거나 자체 태그 형식을 설정하여 다양한 태그 유형을 생성할 수 있습니다. 또한 작업 중에 .retag 명령을 사용하여 태그를 임의의 문자열로 변경할 수 있습니다. (자세한 정보는 439 페이지의 『.retag』의 내용을 참조하십시오.)

현재 작업의 태그를 작업 중에 시스템이 정의한 환경 변수(BF\_TAG)로 사용할 수 있으므로 이를 액세스 및 사용하여 소스 저장소를 레이블링하거나 기타 추적 및 레이블링 목적으로 사용할 수 있습니다. (이 변수에 대한 자세한 정보는 335 페이지의 『시스템 변수 참조』를 참조하십시오.)

두 프로젝트의 태그 변수를 동기화할 수 있습니다. 이렇게 하면 둘 중 하나를 실행할 때 동일한 태그 변수 값이 사용되는 링크가 작성됩니다. (자세한 정보는 360 페이지의 『태그 동기화』를 참조하십시오.)

이 절의 주제에서는 사용자가 원하는 값을 반영하는 동적 태그를 생성하기 위한 태그 형식 및 태그 변수 설정 방법을 설명합니다.



## 프로젝트의 태그 형식 편집

태그 형식은 시스템의 태그 생성 방법을 정의합니다. 태그 형식은 \$ 기호가 표시하는 변수 참조 및 일반 텍스트로 구성됩니다. 태그 형식에서 사용하는 변수는 이전 섹션의 시스템 정의 태그 변수 목록에서 가져오거나 프로젝트를 실행하기 전에 사용자가 프로젝트에 대해 이 변수를 정의해야 합니다. 정의되지 않은 변수는 정적 텍스트로 간주됩니다.

태그 형식은 프로젝트 특성입니다. 이를 편집하려면 프로젝트 단추를 클릭하여 프로젝트 목록을 표시한 후 편집할 프로젝트의 프로젝트 이름을 클릭하십시오. 시스템이 프로젝트의 단계 목록을 표시합니다. 페이지 맨 위의 프로젝트 이름을 클릭하여 프로젝트 특성을 표시하십시오.

태그 형식에서 \$ 기호를 사용하여 태그 변수 시작을 표시하십시오. 필요할 경우 여러 개의 태그 변수를 포함시킬 수 있습니다. 예를 들어, 프로젝트의 주 개정판에 대해 비증분 변수(\$MAJ)를 정의하고 부 개정판에 대해 증분 변수(\$MIN)를 정의한 후 프로젝트의 버전 번호를 반영하는 태그 형식(예: Version\$MAJ.\$MIN)을 가질 수 있습니다. 이렇게 하면 주 버전 번호를 수동으로 제어할 수 있지만 모든 릴리스에 부 버전 번호를 자동으로 증분하게 되어 다음과 같은 태그가 생성됩니다.

Version1.1

Version1.2

## 태그 동기화

프로젝트 레벨의 태그 동기화 특성을 사용하여 둘 이상의 프로젝트가 동일한 변수 값을 사용할 수 있도록 여러 프로젝트에서 태그를 동기화할 수 있습니다. Project B의 태그 동기화 특성을 Project A와 같게 설정하는 경우, Project A(상위)와 Project B(하위) 사이에 상위-하위 관계를 설정합니다.

태그 동기화 특성을 사용하여 프로젝트를 실행하는 경우, 시스템은 하위 프로젝트에 상위 프로젝트의 태그 변수와 일치하는 태그 변수가 있는지 확인합니다. 있으면 하위 프로젝트의 변수는 상위 프로젝트가 마지막으로 사용한 값으로 설정됩니다.

하위 프로젝트의 태그 형식으로 된 변수가 상위 프로젝트의 태그 형식으로 된 변수와 일치하지 않는 경우, 태그 동기화 특성은 영향을 주지 않습니다.

동기화는 태그 변수의 값에 대해서만 작동합니다. 자동 증가 및 채우기 특성은 동기화되지 않습니다.

태그의 변수만이 동기화되므로 여전히 서로 다른 프로젝트를 구별할 수 있습니다.

예를 들어, 다음 표에 나타난 것과 같은 두 프로젝트가 정의되어 있다고 간주합니다.



프로젝트	태그 형식	자동 증가	태그 동기화
Project A	Project_A_\$B	예	-- 없음 --
Project B	Project_B_\$B	예	Project A

그런 다음 프로젝트를 교대로 실행하는 경우(프로젝트 A부터 시작), 완료된 작업 목록에는 다음과 같은 태그가 표시됩니다. 마지막 실행이 먼저 표시되고 동일한 방식으로 완료된 작업 목록에 작업이 표시됩니다.

프로젝트	태그
Project B	Project_B_4
Project A	Project_A_3
Project B	Project_B_2
Project A	Project_A_1

상위 프로젝트에서만 자동 증가 특성을 예로 설정하는 경우에는 결과가 다릅니다. 프로젝트는 다음과 같이 설정됩니다.

프로젝트	태그 형식	자동 증가	태그 동기화
Project A	Project_A_\$B	예	-- 없음 --
Project B	Project_B_\$B	아니오	Project A

그런 다음 프로젝트를 교대로 실행하는 경우(프로젝트 A부터 시작), 완료된 작업 목록에는 다음과 같은 태그가 표시됩니다. 마지막 실행이 먼저 표시되고 동일한 방식으로 완료된 작업 목록에 작업이 표시됩니다.

프로젝트	태그
Project B	Project_B_2
Project A	Project_A_2
Project B	Project_B_1
Project A	Project_A_1

## 태그의 시스템 정의 변수

작업 태그에서 사전 정의된 다음 변수를 사용할 수 있습니다.

변수	값
B	작업 번호: 1로 시작하고 모든 새 작업으로 증분되는 정수 값
BF_D	태그: 날짜 형식 시스템 설정으로 설정된 형식의 날짜. 실행 시 평가되므로 프로젝트 시작 페이지의 태그 예제에서 평가할 수 없습니다. 이 날짜를 사용할 경우 { }로 묶으십시오.
BF_J	연중 날짜. 실행 시 평가되므로 프로젝트 시작 페이지의 태그 예제에서 평가할 수 없습니다. 이 날짜를 사용할 경우 { }로 묶으십시오.
BF_T	태그: 시간 형식 시스템 설정으로 설정된 형식의 시간. 실행 시 평가되므로 프로젝트 시작 페이지의 태그 예제에서 평가할 수 없습니다. 이 날짜를 사용할 경우 { }로 묶으십시오.
BF_W	요일(0에서 6까지의 숫자 값) 실행 시 평가되므로 프로젝트 시작 페이지의 태그 예제에서 평가할 수 없습니다. 이 요일을 사용할 경우 { }로 묶으십시오.

## 시스템 정의 변수 및 시작 페이지의 태그 예제

일부 변수는 작업 시작 시 평가되기 때문에 시작 페이지의 태그 예제에서 평가할 수 없습니다. 이 변수를 사용할 경우 { }로 묶으십시오.


예제:

```
BUILD_$B.{BF_T}
BUILD_$B.{BF_D}.{BF_T}
```

## 태그 변수 작성 또는 편집

자체 태그 변수를 정의하여 태그 형식에 포함할 수 있습니다. 태그 변수는 숫자 값을 가지며, 원하면 각 작업과 함께 시스템에 의해 자동으로 증분될 수 있습니다.

새 변수를 추가하거나 프로젝트에 대한 기존 변수를 편집하려면 프로젝트를 선택한 후,

원하는 프로젝트의 이름 옆에 있는 편집 아이콘()을 클릭하십시오. 프로젝트 특성이 패널의 맨 아래에 나타납니다. 프로젝트의 태그 변수를 표시하려면 태그 탭을 클릭하십시오.


시스템이 프로젝트의 태그 변수 목록을 표시합니다.

프로젝트: TestProject    저장    새 스냅샷 작성    기본값으로 설정    프로젝트 복사    프로젝트 삭제    정리

프로젝트 세부사항    태그    레지스터    노트 (0)    스냅샷

태그 형식: BUILD\_\$\$B    태그 동기화: -- 없음 --

옵션	태그 이름	초기값	채우기	자동 증분
작성		0	-- 없음 --	아니오
지우기				
	B	1	0	예

- 태그 변수를 편집하려면 해당 변수 이름 옆에 있는 편집 아이콘()을 클릭하십시오. 시스템이 태그 변수의 값으로 패널을 채우고 작성을 저장 단추로 변경합니다. 값을 변경하고 저장을 클릭하여 변경사항을 저장하십시오.
- 태그 변수를 삭제하려면 이름 옆에 있는 휴지통 아이콘을 클릭하십시오.
- 새 변수를 추가하려면 변수의 특성을 입력하고 작성 단추를 클릭하십시오.

태그 이름, 초기값, 채우기 및 자동 증분 특성에 대한 정보는 352 페이지의 『태그 탭』을 참조하십시오.

## 라이브러리

라이브러리는 선택기 특성이 없음으로 설정되는 프로젝트입니다. 라이브러리는 다른 프로젝트 내에서 실행되며 해당 라이브러리를 호출하는 단계의 서버 자원에서 실행됩니다.

선택기가 없음인 프로젝트를 저장하면 프로젝트가 라이브러리로 저장된다는 경고 메시지가 표시됩니다. 라이브러리는 라이브러리 패널에 나열됩니다.

단계에서 라이브러리를 호출하려면 단계의 인라인, 체인 패스 또는 체인 실패 특성에서 선택하십시오.

### 라이브러리 정보

라이브러리 패널은 지정된 선택기가 없는 프로젝트인 라이브러리를 포함합니다.

라이브러리는 이를 호출하는 단계의 선택기를 사용합니다. 호출 단계가 선택기를 갖지 않는 경우 라이브러리는 단계 프로젝트의 선택기를 사용합니다. 라이브러리는 일반적으로 기타 프로젝트에서 단계의 인라인으로, 또는 단계의 패스 체인이나 실패 체인으로 호출합니다.

라이브러리 패널에서 라이브러리를 보기, 편집, 작성 또는 실행할 수 있습니다. 라이브러리를 홀로 실행할 수 있지만 이를 수행할 때 선택기를 지정해야 합니다.

**참고:** 라이브러리를 실행할 때 목록의 첫 번째 선택기가 기본적으로 지정됩니다. 다른 기본 선택기를 지정하려면 프로젝트로 작성하십시오.

프로젝트를 편집하고 사용할 선택기를 선택하여 라이브러리를 프로젝트로 변경할 수 있습니다. 라이브러리를 선택기와 함께 저장하면 라이브러리가 프로젝트가 되며 라이브러리 목록에 표시되지 않습니다.


선택기가 없다는 것을 제외하고 라이브러리는 다른 프로젝트와 마찬가지로 취급됩니다.

라이브러리 패널에 액세스하려면 왼쪽 메뉴에서 라이브러리를 클릭하십시오.



## 라이브러리 복사

기존 라이브러리의 사본을 작성하려면 다음 단계를 완료하십시오.

1. 라이브러리를 선택하십시오.
2. 복사하려는 라이브러리 옆에 있는 편집 아이콘()을 선택하십시오.
3. 라이브러리 복사를 클릭하십시오.

라이브러리를 복사할 때 시스템은 기존 라이브러리의 다음 양상을 새 라이브러리에 복사합니다.

- 단계 및 단계의 세부사항 탭에 나열되는 모든 특성
- 라이브러리의 세부사항 탭에 나열된 모든 라이브러리 특성(예: 클래스 및 기타 특성)
- 라이브러리의 태그 형식(라이브러리 특성 패널의 태그 탭에 있음)

시스템은 다음 특성을 복사하지 않습니다.

- 태그 변수(라이브러리 특성 패널의 태그 탭에 있음). 태그 형식에서 사용되는 태그 변수가 복사됨
- 라이브러리 레지스터(라이브러리 특성 패널의 레지스터 탭에 있음)
- 단계 노트(단계 특성 패널의 노트 탭에 있음)
- 라이브러리 노트(라이브러리 특성 패널의 노트 탭에 있음)

## 로그 필터

이 주제는 로그 필터 작성 및 사용 방법에 대해 설명합니다.

## 로그 필터 정보

로그 필터를 사용하여 단계의 성공 기준을 지정하십시오. 필터는 하나 이상의 정규식을 저장합니다.

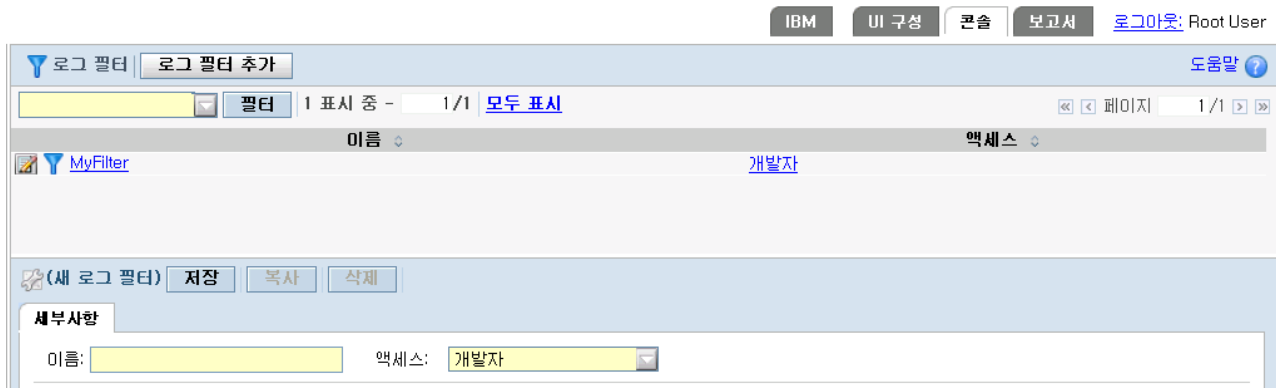
필터링이 설정되지 않은 경우, Build Forge는 명령의 종료 상태로 단계 명령의 성공 또는 실패를 판별하는데, 0은 성공이고 1은 실패입니다. 명령 특성에 다중 명령이 사용되는 경우, 마지막으로 실행된 명령의 종료 상태만 단계 결과 상태에 영향을 줍니다. 일부 명령은 항상 종료 상태 0을 리턴합니다. `net use` 같은 보고 명령은 맵핑된 네트워크 드라이브 목록을 인쇄합니다. 목록에 원하는 드라이브가 포함되어 있지 않더라도 이 명령은 항상 성공합니다.

로그 필터를 사용하면 종료 상태가 아니고 단계에서 명령의 출력을 평가할 수 있습니다.

예를 들어 `net use` 명령의 경우, 로그 필터를 사용하여 특정 드라이브를 찾고 드라이브를 찾으면 단계를 성공으로 표시할 수 있습니다.

## 로그 필터 패널 정보

로그 필터를 작성, 편집 및 삭제하려면 로그 필터 패널을 사용하십시오. 패널을 보려면 프로젝트 > 로그 필터를 선택하십시오.



패널에 다음 필드가 있습니다.

이름 로그 필터의 이름입니다.

액세스 이 필터를 사용할 수 있는 사용자의 액세스 그룹입니다.

## 로그 필터 작성

로그 필터는 하나 이상의 필터 패턴을 포함할 수 있습니다. 각 필터 패턴은 조치와 연관되며, 알림을 위해 액세스 그룹과 선택적으로 연관됩니다. 먼저 로그 필터를 정의한 후 로그 필터를 프로젝트의 단계와 연관시키십시오.

로그 필터를 작성하려면 다음을 수행하십시오.

1. 프로젝트 > 로그 필터를 선택하십시오. 관리 콘솔이 로그 필터 목록과 새 로그 필터 패널을 표시합니다.
2. 이름에 로그 필터 이름을 입력한 후 저장을 클릭하십시오. 관리 콘솔이 로그 필터를 저장하고 새 패턴 패널을 표시합니다.
3. 로그 필터에 대해 정의하는 각 필터 패턴에 대해 다음을 수행하십시오.
  - a. 패턴 필드에 정규 표현식을 입력하십시오(정규 표현식은 Perl 호환 표현식이어야 함). 프로젝트가 실행될 때 Build Forge가 단계 출력에서 패턴을 검색합니다. 자세한 사항은 『필터 패턴』의 내용을 참조하십시오.
  - b. 조치에서, 필터 패턴을 찾으려면 취할 필터 조치를 선택하십시오. 기본 특성(실패 설정)은 단계 상태를 실패로 설정합니다. 자세한 사항은 368 페이지의 『필터 조치』의 내용을 참조하십시오.
  - c. 알림에서, 필터가 활성화되면 구성원에게 알림 이메일을 발송할 액세스 그룹을 선택하십시오.
  - d. 저장을 클릭하십시오.

로그 필터를 사용하려면 프로젝트 단계를 선택하고 단계의 결과 특성을 새 로그 필터로 설정하십시오. 『단계에 로그 필터 지정』의 내용을 참조하십시오.

## 단계에 로그 필터 지정

로그 필터를 사용하려면 단계 결과 특성을 사용하여 프로젝트 단계에 로그 필터를 지정해야 합니다. 단계에 로그 필터를 지정할 때 프로젝트가 실행될 때마다 로그 필터의 필터 패턴이 단계 출력에서 실행됩니다. 그러나 로그 필터가 어댑터 출력에 적용되지 않습니다.

단계에 로그 필터를 지정할 때 로그 필터로 설정된 단계 결과는 단계 성공 또는 실패를 판별하기 위한 모든 다른 기준을 대체함에 유의하십시오. 여기에는 단계 명령의 종료 상태 또는 단계 특성이 포함됩니다. 예를 들어, 단계 실행 시간이 단계 제한시간 특성으로 지정된 시간을 초과하는 경우 단계가 중단됩니다. 그러나 연관된 로그 필터 조치로 인해 단계가 실패로 설정되어야 단계 상태가 실패로 간주됩니다.

단계에 로그 필터를 지정하려면 다음을 수행하십시오.

1. 프로젝트 또는 라이브러리를 선택하여 단계에 액세스하십시오.
2. 단계를 포함하는 프로젝트 또는 라이브러리를 선택하십시오.
3. 단계를 선택하여 단계 세부사항 패널을 여십시오.
4. 결과에서 단계가 실행될 때마다 실행할 로그 필터를 선택하십시오.

## 필터 패턴

필터 패턴은 단계 출력에서 일치시킬 문자열 또는 표현식을 정의합니다. 사용자가 작성하는 각 필터 패턴은 단일 필터 조치와 연관됩니다. 필터 패턴과 조치 둘 다 필터 로그

세트에 정의됩니다. 하나의 로그 필터에 여러 개의 필터 패턴을 포함시키고 단일 단계의 출력에 이를 적용하는 기능을 통해 복잡한 표현식을 생성하지 않고 다중 검색 기준을 사용할 수 있습니다.

로그 필터를 작성하려면 프로젝트 > 로그 필터를 선택하십시오. 자세한 사항은 364 페이지의 『로그 필터』의 내용을 참조하십시오.

## 필터 패턴 구문

필터 패턴 작성을 위한 다음 가이드라인을 검토하십시오.

- 필터 패턴은 정규 표현식으로 정의되며 Perl 호환 구문을 사용해야 합니다. Perl 호환 표현식 생성에 대한 세부사항은 Perl 문서를 참조하십시오.
- 시스템이 구분 슬래시 문자를 추가하므로(<expression>/) 슬래시 구분 기호 안에 넣지 않고 표현식을 지정하십시오(expression).
- 표현식에 메타 문자(예: a /b)가 포함된 경우, 메타 문자 앞에 백슬래시 이스케이프 문자가 와야 합니다(a\b).

몇 가지 표준 정규 표현식 구문이 다음 표에 표시되어 있습니다.

표현식	일치
Production	문자열의 모든 위치에서 <i>Production</i> 과 일치합니다.
^Production	문자열 처음에 <i>Production</i> 과 일치합니다.
Error.*[0-9]\$	문자열 끝에 숫자로 끝나는 문자 세트가 다음에 오는 <i>Error</i> 를 포함하는 행과 일치합니다.
[Ww]arning	<i>Warning</i> 또는 <i>warning</i> 과 일치합니다.
.*	임의의 문자와 0번 이상 일치합니다. 점(.)은 임의의 문자와 일치하며 별표(*)는 0번 이상 임의의 문자와 일치합니다.

## 동일 행에서 다중 패턴 일치

패턴 필터를 생성하려면 시스템의 패턴 일치 검색 방법을 이해하는 것이 중요합니다.

각 출력 행에 대해 시스템은 순서대로 모든 필터 패턴과 비교하여 일치를 확인합니다. 일치를 찾으면 중지하고 다음 패턴으로 이동합니다. 따라서 패턴이 한 행에서 두 번 발생하는 경우, 시스템이 찾지 못할 수도 있습니다. 예를 들어 다음 출력 행을 고려하십시오.

```
exception retrying exception
```

아래 표의 필터 패턴을 사용하여 시스템이 첫 번째 *exception*과 일치하면 단계 결과를 실패로 설정하고, *retrying*과 일치하면 단계 결과를 패스로 설정하며, 두 번째 *exception*과 일치하지 않으면 다음 행으로 이동합니다.

필터 패턴	필터 조치	예제 설명
[[Ee]xception [Rr]etrying	실패 설정 - 실패 지우기 실패 - 패스	이는 Java 프로젝트에 유용하며, 예외에서는 단계가 실패하지만 재시도에서는 실패를 지웁니다. 재시도가 실패하면 새 예외가 생성되므로 명령의 최종 상태는 올바릅니다.

이 문제점을 해결하는 한 가지 방법은 표에 있는 필터 패턴을 다음 필터 패턴으로 바꾸는 것입니다.

`retrying.*exception`

## 필터 조치

필터 조치는 단계 출력에서 필터 패턴을 찾았을 때 취할 조치를 정의합니다. 사용자가 작성하는 각 필터 패턴은 단일 필터 조치와 연관됩니다. 필터 조치와 패턴 둘 다 로그 필터에 정의됩니다.

로그 필터를 작성하려면 **프로젝트 > 로그 필터**를 선택하십시오. 자세한 사항은 364 페이지의 『로그 필터』의 내용을 참조하십시오.

필터 조치	정의	단계 결과
실패 설정(기본값)	시스템은 필터 패턴을 찾으면 단계 결과 상태를 실패로 설정하고 현재 행에서 세트의 필터 패턴을 계속 검색합니다.	실패
실패/정지 설정	시스템은 필터 패턴을 찾으면 단계 결과 상태를 실패로 설정하고 현재 행에서 세트의 필터 패턴 검색을 중지하고 다음 행으로 건너뛴 후, 패턴 검색을 다시 시작합니다.	실패
실패 지우기	시스템은 필터 패턴을 찾으면 단계 결과 상태를 패스로 설정하고 현재 행에서 세트의 필터 패턴을 계속 검색합니다.	패스
실패/정지 지우기	시스템은 필터 패턴을 찾으면 단계 결과 상태를 패스로 설정하고 현재 행에서 세트의 필터 패턴 검색을 중지하고 다음 행으로 건너뛴 후, 패턴 검색을 다시 시작합니다.	패스
정지	시스템은 필터 패턴을 찾으면 현재 행에서 세트의 필터 패턴 검색을 중지하고 다음 행으로 건너뛴 후, 패턴 검색을 다시 시작합니다. 단계 결과 상태를 변경하지 않습니다.	적용할 수 없음
포함	포함을 사용하여 다른 로그 필터에 있는 하나 이상의 로그 필터를 참조할 수 있습니다. 패턴 필드에 포함시킬 로그 필터를 지정하고 조치 필드에서 포함을 선택하십시오.	적용할 수 없음
경고	시스템은 필터 패턴을 찾으면 단계 결과 상태를 경고로 설정하고 현재 행에서 세트의 필터 패턴을 계속 검색합니다. (참고: 경고 상태는 패스 상태입니다. 이 단계에 지정된 모든 패스 체인이 실행됩니다.)	경고
경고 지우기	시스템은 필터 패턴을 찾으면 단계 결과 상태를 패스로 설정하고 현재 행에서 세트의 필터 패턴을 계속 검색합니다.	패스



필터 조치	정의	단계 결과
경고/정지 지우기	시스템은 필터 패턴을 찾으면 단계 결과 상태를 패스로 설정하고 현재 행에서 세트의 필터 패턴 검색을 중지하고 다음 행으로 건너뛰면 후, 패턴 검색을 다시 시작합니다.	패스
변경자에게 알림	<p>변경자에게 알림을 사용하려면 관계 목록을 작성하는 어댑터를 프로젝트에 포함시키고 변경자에게 알림 로그 필터를 포함하는 단계 이전에 어댑터 단계를 실행해야 합니다.</p> <p>어댑터 관계 목록은 사용자와 오브젝트(예: 변경된 파일)를 쌍으로 연결합니다. 세부사항은 어댑터 XML 참조를 참조하십시오.</p> <p>어댑터가 실행되어 관계 목록을 작성한 후, 변경자에게 알림 조치가 있는 로그 필터가 단계 출력 행의 필터 패턴과 일치하는 경우 관계 목록의 오브젝트를 일치시키기 위해 행을 다시 스캔합니다. 오브젝트 일치를 찾은 경우 오브젝트와 쌍이 되는 사용자에게 이메일 알림이 발송됩니다.</p> <p>예를 들어, 다음의 단계 출력 행에서 필터 패턴 Error에 대한 오브젝트 일치하는 MyFile.c입니다. 따라서 관계 목록에서 MyFile.c 오브젝트와 쌍이 되는 사용자에게 오류에 대한 이메일 알림이 발송됩니다.</p> <p>Error: Invalid token on line 55 of MyFile.c</p>	적용할 수 없음
실패 결과로 빌드 중지	시스템은 필터 패턴을 찾으면 작업 결과를 실패로 설정하고 작업을 종료합니다. 단계 결과는 해당 결과에 따라 설정됩니다. 추가 단계가 실행되지 않습니다.	결과
패스 결과로 빌드 중지	시스템은 필터 패턴을 찾으면 작업 결과를 패스로 설정하고 작업을 종료합니다. 단계 결과는 해당 결과에 따라 설정됩니다. 추가 단계가 실행되지 않습니다.	결과
경고 결과로 빌드 중지	시스템은 필터 패턴을 찾으면 작업 결과를 경고로 설정하고 작업을 종료합니다. 단계 결과는 해당 결과에 따라 설정됩니다. 추가 단계가 실행되지 않습니다.	결과

## 필터 알림

로그 필터의 모든 필터 패턴에 대해, 액세스 그룹에 이메일을 발송하여 단계의 패턴 필터가 활성화된 구성원에게 알리도록 선택적으로 알림을 설정할 수 있습니다.

## 오류 임계값

특수 환경 변수, `_ERROR_THRESHOLD`를 사용하여 개별 단계 및/또는 프로젝트의 임계값을 설정할 수 있습니다. 그러면 시스템이 필터 일치 수를 계수하고 임계값이 충족되면 단계나 프로젝트를 실패로 처리하거나 임계값에 도달했다는 사실을 작업 노트에 적어 둡니다.

자세한 정보는 337 페이지의 『트리거 변수 참조』의 내용을 참조하십시오.

## 오류 및 경고 계수

단계의 성공 또는 실패 여부를 판별하도록 필터를 단계와 연관시키는 경우, 시스템은 필터로 포착한 오류 및 경고 수를 표시합니다. 작업 > 완료됨 탭에서 숫자가 작업 결과 뒤에 있는 소괄호 안의 결과 옆에 표시됩니다. 형식은 (<fail matches> / <warning matches>)입니다.

예제: 작업 결과 실패함(1 / 0)은 작업이 실패했으며 1개의 실패 필터가 일치했고 경고 필터는 일치하지 않았음을 표시합니다.

---

## 클래스

이 주제는 클래스 작성 및 사용 방법에 대해 설명합니다.

### 클래스 정보

클래스는 작업 그룹입니다. 각 작업은 오직 한 클래스의 구성원이어야 합니다. 클래스를 사용하여 클래스의 각 작업에 서로 다른 글로벌 관리 동작을 적용할 수 있습니다. 작업은 프로젝트의 특성에서 기본 클래스를 가져옵니다. 또한 작업 > 시작 페이지에서 작업을 시작할 때 수동으로 작업의 다른 클래스를 선택할 수도 있습니다.

클래스에는 다음 활동을 관리하기 위한 특성이 있습니다.

- 자동으로 작업 삭제
- 시스템이 이 클래스의 작업을 제거할 때 또는 기존 작업이 이 클래스로 또는 이 클래스에서 변경될 때 작업 실행

**참고:** 작업이 완료된 후에 작업 클래스를 변경할 수 있습니다. 작업 클래스를 변경하려면 작업 > 완료를 선택하여 작업을 본 후 작업 태그(예: BUILD\_5)를 클릭하십시오. 클래스 필드에서 다른 클래스를 선택하십시오.

### 클래스 패널 정보

클래스를 추가, 편집 및 삭제하려면 클래스 패널을 사용하십시오. 이 패널을 보려면 프로젝트 > 클래스를 선택하십시오. 시스템이 클래스의 목록을 표시합니다. 클래스 특성을 편집하려면 해당 클래스를 선택하십시오.

IBM UI 구성 콘솔 보고서 로그아웃: Root User

클래스
클래스 추가
도움말

필터
1 표시 중 - 2/2
모두 표시

<<
<
페이지
1 / 1
>
>>

클래스 빌드	액세스	삭제	일	계수	선택
<a href="#">스크래치</a>	<a href="#">게스트</a>	전체	5	무제한	임의 빌드
<a href="#">프로덕션</a>	<a href="#">게스트</a>	콘솔 데이터	영구적	무제한	임의 빌드

(새 클래스)
저장
복사
삭제

이름:

액세스: 개발자

파일 삭제:

전체

일:

영구적

계수:

무제한

선택:

임의 빌드

제거 시 시작:

-- 없음 --

입력 시 시작:

-- 없음 --

종료 시 시작:

-- 없음 --

클래스의 액세스 특성은 사용자가 지정하는 액세스 그룹을 기반으로 클래스를 보거나 변경할 수 있는 사용자를 제어합니다.

#### 자동으로 작업 삭제 특성:

대부분의 클래스 특성은 삭제되는 프로젝트 데이터 유형과 삭제되는 조건을 제어합니다.

시스템은 제거 확인 시간 시스템 설정(기본적으로 15분으로 설정됨)으로 정의된 간격으로 삭제할 작업을 확인합니다.

**참고:** 또한 스케줄을 사용하여 제거가 수행되어야 하는 시기를 표시하여, 시스템이 다른 방법으로 사용될 때 제거 실행을 시도하지 않게 할 수 있습니다. 이 기능을 사용하여 예를 들어 밤에 또는 일주일에 한 번만 제거가 발생하도록 할 수 있습니다. 370 페이지의 『클래스』의 내용을 참조하십시오.

제거 작업이 실행되면, 시스템은 작업을 아카이브하고 클래스의 설정에 따라 데이터를 삭제합니다.

#### 파일 삭제

삭제되는 데이터의 유형을 판별합니다. 다음 옵션이 있습니다.

**전체** 데이터베이스에서 작업에 대한 모든 정보를 삭제하고 작업을 실행한 서버에서 작업 디렉토리를 삭제합니다.

#### 콘솔 데이터

데이터베이스에서 작업에 대한 모든 정보를 삭제하고 서버의 작업 디렉토리는 그대로 둡니다.

## 로그 및 파일

작업 디렉토리나 로그를 삭제하지만 작업 > 아카이브 페이지에 단계 패스/실패 정보를 보유합니다.

**로그만** 작업 로그만 삭제합니다.

**파일만** 작업을 실행한 서버에서 작업 디렉토리를 삭제합니다. 로그 및 기타 일부 정보(예: 단계 패스/실패 상태)는 데이터베이스에 남아 있습니다. 작업 레코드는 작업 > 아카이브 페이지로 이동됩니다.

**일\*** 작업을 삭제하기 전의 경과 일 수입니다.

**계수\*** 허용되는 최대 작업 수입니다. 작업 수가 계수 값을 초과하면 시스템은 추가 빌드를 삭제하기 위해 제거 작업의 스케줄을 지정합니다. 기본값(무제한)은 존재하는 작업 수 때문에 시스템이 작업을 제거하지 못하게 합니다.

**선택** 선택 특성은 작업을 삭제하기 전에 충족해야 하는 추가 조건을 설정합니다. 다음 옵션이 있습니다.

### 임의 빌드

이 옵션이 선택된 경우, 선택 특성은 작업 삭제에 영향을 주지 않습니다.

**실패만** 시스템은 실패한 작업만 삭제합니다.

**패스만** 시스템은 패스한 작업만 삭제합니다.

### 1 패스 유지

다른 삭제 기준을 충족하더라도 시스템은 항상 가장 최근에 패스한 작업을 보유합니다.

\* 일 또는 계수 값 중 하나를 초과하면 시스템이 작업을 삭제합니다. 예를 들어, 계수가 10으로 설정되고 일 수가 2로 설정되며 8개의 작업이 있지만 2일이 지난 작업이 3개이면 3개의 작업이 삭제됩니다. 마찬가지로, 12개의 작업이 있고 모두 2일을 넘지 않은 경우, 오래된 작업 두 개가 삭제됩니다.

제거 작업이 실행되면, 시스템은 작업을 아카이브하고 클래스의 설정에 따라 데이터를 삭제합니다.

### 이벤트 시 프로젝트 시작 특성:

클래스와 관련된 특정 이벤트가 발생할 때 프로젝트를 실행(체인)할 수 있습니다. 이 특성을 사용하면 프로세스에서 상태 진행을 모델링할 수 있습니다.

다음 클래스 특성을 사용하여 특정 이벤트가 발생할 때 작업을 실행할 수 있습니다.

### 제거 시 시작

이 특성은 클래스의 작업이 제거될 때(즉, 시스템이 이 클래스의 작업에 대해

제거 작업을 시작할 때마다) 지정된 프로젝트를 실행합니다. 이 특성을 사용하여 제거와 함께 자동으로 삭제되지 않는 일부 특정 파일을 삭제할 수 있습니다.

#### 입력 시 시작

이 특성은 작업의 클래스 특성이 이 클래스로 변경될 때 지정된 프로젝트를 실행합니다. 이 특성을 사용하여 프로세스를 작업 재분류에 연결할 수 있습니다. 예를 들어, 테스트 클래스를 작성하고 작업이 테스트 클래스로 승격되면 일부 표준 테스트를 실행할 수 있습니다.

#### 종료 시 시작

이 특성은 작업의 클래스 특성이 이 클래스에서 다른 클래스로 변경될 때 지정된 프로젝트를 실행합니다.

이러한 특성은 프로젝트를 체인으로 실행합니다.

**참고:** 이들 특성이 변경 필수 변수를 포함하는 작업을 실행하는 경우 변수 값은 변경되지 않습니다.

---

## 알림 설정

시스템은 프로젝트 또는 단계가 패스하거나 실패할 때 또는 특정 기타 이벤트가 발생할 때 이메일 알림을 발송할 수 있습니다. 이 절에서는 이메일 알림 구성 방법과 이메일 알림의 모양을 제어하는 알림 템플릿의 수정 방법을 설명합니다.

알림은 액세스 그룹으로 전송되므로 보안을 염두에 두고 알림에 대한 액세스 그룹을 설정하십시오. 알림은 개별 사용자에게 직접 전송되지 않고 항상 그룹으로 전송되지만 필요하면 한 명의 사용자만 포함하는 그룹을 설정할 수 있습니다.

계층 구조(서브그룹을 포함하는 액세스 그룹)가 될 액세스 그룹을 설정한 경우에는 다음과 같은 알림이 작동합니다.

- 알림의 액세스 그룹이 상위 그룹이고 하나 이상의 사용자를 포함하는 경우, 알림은 상위 그룹의 사용자에게만 송신됩니다.
- 알림의 액세스 그룹이 상위 그룹이고 사용자를 포함하지 않으며 서브그룹만 포함하는 경우, 알림은 모든 서브그룹 사용자에게 송신됩니다.

알림 이벤트를 작성하려면 알림 특성에 대한 액세스 그룹을 선택하십시오.

- 프로젝트의 경우, 시작 알림, 패스 알림 및 실패 알림 그룹을 선택할 수 있습니다.
- 단계의 경우, 패스 알림 및 실패 알림 그룹을 선택할 수 있습니다.

이러한 특성에 선택된 액세스 그룹이 있을 때마다 시스템은 해당 이벤트가 발생할 때 그룹 구성원에게 이메일을 발송합니다. 이러한 이메일 메시지에서 시간에 대한 참조는 알림을 발송하는 관리 콘솔의 시간대를 기반으로 합니다.

프로젝트에 다른 프로젝트가 인라인 프로젝트로 포함된 경우, 인라인 프로젝트의 시작, 패스 및 실패 알림 설정은 무시되지만 단계에 대한 알림 설정은 사용됩니다. 세부사항은 382 페이지의 『인라인 프로젝트에 대한 알림』의 내용을 참조하십시오.

다음은 수행해야 알림을 사용할 수 있습니다.

- 시스템이 이메일을 보내는 데 사용할 SMTP 서버를 알 수 있도록 SMTP 서버 시스템 설정을 구성하십시오. 기본값은 localhost입니다. 또한 시스템 정보 소스의 시스템 설정을 설정해야 할 수도 있습니다. 이 주소가 소스 주소로 사용되며 대다수 SMTP 서버에는 올바른 소스 주소가 필요합니다. 기본값은 root@localhost입니다.
- 하나 이상의 알림 그룹을 작성하고 그룹에 사용자를 지정하십시오.
- 개별 프로젝트 및/또는 단계에 대해 알릴 그룹을 선택하십시오.

또한 알림 템플릿을 편집하여 시스템이 보내는 알림 이메일을 구성할 수 있습니다. 378 페이지의 『알림 템플릿 사용자 정의』의 내용을 참조하십시오.

## 알림 템플릿 정보

알림 템플릿은 사용자에게 시스템에서 발생한 이벤트에 대한 사용자 정의 메시지를 보내는 방법을 제공합니다.

### 제공되는 템플릿 유형

시스템은 다음과 같은 알림 템플릿 유형 세트를 포함합니다.

- 변경 아티팩트 일치 메시지
- 필터 일치 메시지
- 작업 .break 메시지
- 작업 실패 메시지
- 작업 전달 메시지
- 작업 시작 메시지
- 작업 경고 메시지
- 제거 실패 메시지
- 단계 .email 메시지
- 단계 실패 메시지
- 단계 전달 메시지
- 단계 경고 메시지
- 시스템 정보 메시지

각 유형은 변환된 메시지 세트를 포함합니다. 메시지는 변수를 사용하여 컨텍스트를 제공합니다.

### 글로벌 사용법

제공된 템플릿은 다음과 같이 글로벌로 사용할 수 있습니다.

- 프로젝트: 각 프로젝트의 특성으로 프로젝트 시작, 패스 및 실패에 대한 알림 유형을 지정할 수 있습니다. 경고로 완료된 작업의 경우 작업 경고 알림이 제공됩니다.
- 단계: 각 단계의 특성으로 단계 시작, 패스 및 실패에 대한 알림 유형을 지정할 수 있습니다. 단계 실패 후에 계속되는 작업의 경우 단계 경고 알림이 제공됩니다.
- 기타 알림 유형은 다음과 같습니다.
  - 아티팩트 변경 일치 메시지 - 아티팩트가 변경되었으며 사용자와 연관됩니다.
  - 작업 .break 메시지 - 작업에서 .break 명령을 포함하는 단계가 발생함
  - 단계 .email 메시지 - 단계에서 .email 명령이 실행됨
  - 시스템 정보 메시지

### 변환으로 저장된 메시지

사용된 알림 메시지는 템플릿 유형과 연관된 변환으로 저장됩니다. 각 변환은 알림 메시지에 대한 정보를 지정합니다.

- 언어
- 설명
- 제목(알림 이메일의 제목으로 사용됨)
- 본문(알림 이메일에서 메시지 본문으로 사용됨)

제공된 알림 템플릿 및 사용자가 작성하는 알림 템플릿에서 변환을 편집할 수 있습니다.

### 메시지 형식

메시지는 두 개의 메시지 유형 중 하나로 보낼 수 있습니다.

- 일반 텍스트 메시지(기본값)
- MIME - 본문 필드의 특성 HTML 태그를 포함하는 알림은 자동으로 MIME 메시지로 전송됩니다. 379 페이지의 『본문 필드에서 HTML 사용』의 내용을 참조하십시오.

### 템플릿의 변수

시스템은 다수의 변수에 대한 템플릿을 구문 분석합니다(380 페이지의 『환경 변수 및 레지스터 변수 사용』 참조).

### 지정된 프로젝트 및 단계의 사용자 정의 템플릿

제공된 알림 유형을 기반으로 사용자 정의된 알림 템플릿을 작성할 수 있습니다. 사용자 정의된 템플릿은 프로젝트 내의 개별 단계 또는 프로젝트와 연관되어야 합니다. 378 페이지의 『특정 프로젝트 및 단계에 사용할 새 템플릿 작성』의 내용을 참조하십시오.

## 템플리트 패널 정보

이벤트에 관한 알림을 설정하려면 템플리트 패널을 사용하십시오. 이 패널을 보려면 프로젝트 > 템플리트를 선택하십시오.

The screenshot shows the '템플리트' (Template) panel in Rational Build Forge. At the top, there are tabs for 'IBM', 'UI 구성', '콘솔', '보고서', and a '로그아웃: Root User' link. Below the tabs, there's a '템플리트 추가' button and a '도움말' link. The main area has a table with columns: '설명' (Description), '프로젝트' (Project), and '단계' (Stage). The '설명' column lists four templates: '단계 .email 메시지', '단계 경고 메시지', '단계 실패 메시지', and '단계 패스 메시지'. The '프로젝트' column shows 'HelloWorld' and the '단계' column shows '-- 모두 --'. Below the table, there are buttons for '(새 템플리트)', '저장', and '삭제'. At the bottom, there's a '세부사항' (Details) section with fields for '유형:' (Type), '프로젝트:' (Project), and '단계:' (Stage), each with a dropdown menu.

패널은 Rational Build Forge와 함께 제공되는 모든 템플리트를 나열합니다. 새 템플리트를 작성할 때 다음 필드를 지정하십시오.

**유형** 메시지의 유형입니다. 기존 템플리트 유형을 선택하십시오.

**프로젝트**

필수: 메시지를 보내는 대상 프로젝트입니다.

**단계** 선택사항: 메시지를 발송할 지정된 프로젝트의 단계

## 변환 패널 정보

변환 패널을 사용하여 템플리트가 사용하는 메시지를 설정하십시오. 패널을 보려면 프로젝트 > 템플리트를 선택한 후 목록에서 템플리트를 선택하십시오. 템플리트의 변환 목록이 표시됩니다. 목록의 각 항목은 다음 정보를 표시합니다.

**언어** 필수. 메시지의 언어입니다.

**설명** 목록에서 변환의 제목으로 사용됩니다.

**보낸 사람**

선택사항: 메시지의 보낸 사람 필드에서 사용할 이메일 주소입니다. 다음 이메일 주소 형식이 지원됩니다. *Username*은 이메일 사용자 이름입니다. *Name*은 임의의 문자열로, 대개 일부 양식에 지정된 이름입니다.

```
username@host.com
name username@host.com
name "<username@host.com>"
name <username@host.com>
name "username@host.com"
```

**제목** 필수: 메시지의 제목 필드에서 사용할 문자열입니다. 변수를 포함할 수 있습니다.



본문 필수: 메시지의 텍스트입니다. 변수를 포함할 수 있습니다.

## SMTP 서버 구성

SMTP 서버를 구성하려면 **관리 > 시스템 > SMTP 서버**를 선택하십시오. 시스템이 SMTP 서버 값의 편집 패널을 표시합니다. 사이트의 SMTP 서버 이름을 입력하십시오. 기본값은 localhost입니다.

이 주소가 소스 주소로 사용되며 대다수 SMTP 서버에는 올바른 소스 주소가 필요하기 때문에 시스템 정보 소스 매개변수도 설정해야 합니다. 기본값은 root@localhost입니다.

자세한 정보는 259 페이지의 『시스템 구성 설정』의 내용을 참조하십시오.

## 프로젝트 및 단계의 알림 특성 설정

알림의 SMTP 및 그룹 구성이 제 위치에 있는 경우, 프로젝트와 단계를 구성하여 특정 이벤트가 발생할 때 알림을 전송할 수 있습니다.

- 프로젝트의 경우, 시작, 통과 및 실패 알림 특성을 설정할 수 있습니다. 이는 프로젝트 특성입니다.
- 단계의 경우, 통과 또는 실패 알림 특성을 설정할 수 있습니다. 이는 단계 특성입니다.

단일 단계에 여러 알림 설정이 있는 경우, 첫 번째 일치만 이메일 알림이 됩니다.

## 알림 연습

다음 프로시저는 이메일 알림을 설정하고 시험하는 방법을 설명합니다. 연습에는 SMTP 서버와 이메일 계정이 필요합니다.

1. 『SMTP 서버 구성』에 설명된 대로 SMTP 서버를 설정하십시오.

사용자가 액세스할 수 있는 이메일 계정으로 이메일을 발송하는 사용자 계정이 있는지 확인하십시오.

2. **관리 > 액세스 그룹**을 선택하십시오.
3. 선택한 사용자를 초기 구성원으로 하여 이메일 테스트라는 새 액세스 그룹을 작성하십시오.
4. 프로젝트(예: Hello World 프로젝트)를 선택하고 프로젝트 특성을 편집하십시오. 시작 알림, 패스 알림 및 실패 알림 필드에서 이메일 테스트 그룹을 선택하십시오.
5. 프로젝트를 실행하십시오.
6. 두 개의 이메일(프로젝트 시작을 표시하는 이메일과 프로젝트 성공 또는 실패를 표시하는 이메일)을 수신했는지 확인하십시오. 이메일을 수신하지 못하는 경우, 올바른 SMTP 서버 값을 사용했는지 확인하십시오.

## 알림 템플리트 사용자 정의

다음과 같이 알림 템플리트를 사용자 정의할 수 있습니다.

- 프로젝트 및 단계의 사용자 정의 템플리트 작성: 프로젝트 내에서 개별 프로젝트 및 개별 단계의 사용자 정의 템플리트를 작성할 수 있습니다. 제공된 템플리트 유형을 사용하여 사용자 정의 템플리트를 작성하십시오. 사용자가 작성하는 사용자 정의 템플리트 내에서 변환을 작성할 수 있습니다.
- 제공된 템플리트의 변환 편집: 제공된 각 알림 템플리트 내에서 변환을 편집하거나 여기에 추가할 수 있습니다. 제공된 템플리트의 기본 특성을 편집할 수 없습니다.

### 특정 프로젝트 및 단계에 사용할 새 템플리트 작성

시스템은 시스템에서 발생할 수 있는 여러 이벤트에 사용할 템플리트를 수반합니다. 특정 프로젝트 또는 프로젝트 내의 단계에 특정한 새 템플리트를 작성할 수 있습니다. 프로젝트 또는 단계는 사용자 정의 알림을 작성하기 전에 시스템에서 작성해야 합니다.

새 템플리트를 작성하려면 다음을 수행하십시오.

1. 프로젝트 > 템플리트를 선택하십시오. 시스템이 현재 템플리트 목록을 표시합니다.
  2. 템플리트 추가를 클릭하십시오.
  3. 템플리트에 정보를 지정하십시오.
    - 유형: 새 템플리트의 기초로 사용할 기존 알림 유형을 선택하십시오.
    - 프로젝트: 새 템플리트가 적용되는 프로젝트를 선택하십시오. 템플리트는 선택된 프로젝트 실행에 대해 생성되는 알림 메시지에서만 사용됩니다.
- 참고: 프로젝트에 대해 둘 이상의 스냅샷이 정의된 경우 스냅샷 목록이 표시됩니다. 사용할 스냅샷을 선택하십시오. 알림은 해당 스냅샷을 사용하는 프로젝트 실행에만 적용됩니다. 기본 스냅샷을 지정하는 경우 기본으로 지정된 스냅샷이 사용됩니다.
- 단계: 특정 단계를 선택하거나(템플리트가 해당 단계의 알림에만 적용되도록), 선택한 프로젝트의 모든 알림에 템플리트가 적용되도록 이벤트를 선택할 수 있습니다.
4. 저장을 클릭하십시오.
  5. 템플리트의 변환을 작성하십시오.
    - a. 프로젝트 > 템플리트에서 템플리트를 선택하십시오.
    - b. 변환 추가를 선택하십시오.
    - c. 템플리트에 정보를 지정하십시오.
      - 언어: 필수 - 메시지의 언어입니다.
      - 설명: 필수 - 이 필드는 변환의 제목으로 사용됩니다.
      - 보낸 사람: 선택사항 - 메시지의 보낸 사람 필드에서 사용할 이메일 주소입니다.

- **제목:** 필수 - 메시지의 제목 필드에서 사용할 문자열입니다. 변수를 포함할 수 있습니다.
  - **본문:** 필수 - 메시지의 텍스트입니다. 변수를 포함할 수 있습니다. 여기에 특정 HTML 태그가 포함되는 경우 메시지는 일반 텍스트가 아닌 MIME 메시지로 전송됩니다.
- d. **저장**을 클릭하십시오.
- e. 필요한 경우 기타 변환의 프로세스를 반복하십시오.

## 알림 템플릿에서 변환 편집

각 알림 템플릿은 변환 목록을 사용하여 알림의 메시지를 저장합니다.

제공된 알림 템플릿과 개별 프로젝트 및 단계에 대해 사용자가 작성하는 알림 템플릿의 변환을 편집할 수 있습니다.

변환을 편집하려면 다음을 수행하십시오.

1. **프로젝트 > 템플릿**에서 편집할 템플릿을 선택하십시오. 변환 목록이 표시됩니다.
2. 편집할 변환을 클릭하십시오.
3. 템플릿에 정보를 지정하십시오.
  - **언어:** 필수 - 메시지의 언어입니다.
  - **설명:** 필수 - 이 필드는 변환의 제목으로 사용됩니다.
  - **보낸 사람:** 선택사항 - 메시지의 보낸 사람 필드에서 사용할 이메일 주소입니다.
  - **제목:** 필수 - 메시지의 제목 필드에서 사용할 문자열입니다. 변수를 포함할 수 있습니다.
  - **본문:** 필수 - 메시지의 텍스트입니다. 변수를 포함할 수 있습니다. 여기에 특정 HTML 태그가 포함되는 경우 메시지는 일반 텍스트가 아닌 MIME 메시지로 전송됩니다.
4. **저장**을 클릭하십시오.

## 본문 필드에서 HTML 사용

알림은 기본적으로 일반 텍스트로 전송됩니다. 다음 HTML 태그 또는 파트를 사용하는 경우 알림은 MIME 메시지로 전송되며 모든 HTML 태그가 해석됩니다.

```
<html>
<body>
<a>
<b>
<p>
<table>
```

시스템은 변환을 작성할 때 본문 필드를 구문 분석합니다. 해당 시점에서 사용할 형식을 판별합니다. 시스템은 나열된 태그 및 태그 파트만 스캔합니다.

메시지가 스캔되는 요소를 포함하지 않는 경우 메시지는 일반 텍스트로 전송됩니다. 메시지의 HTML 태그는 해석되지 않습니다. 태그는 메시지에 포함됩니다.

## 환경 변수 및 레지스터 변수 사용

`${VAR}` 구문을 사용하는 한 알림 템플릿에서 환경 변수(표준 시스템 변수뿐 아니라 사용자가 정의한 변수)를 참조할 수 있습니다.

또한 알림 템플릿에 프로젝트의 레지스터 변수를 포함시킬 수 있습니다. 빈 레지스터를 참조하는 경우 시스템이 빈 문자열을 리턴합니다.

## 특수 알림 템플릿 변수

다음 표에는 알림 템플릿에 사용 가능한 특수 변수가 나열되어 있습니다. 일부 변수는 컨텍스트에 따라 다르며, 관련된 경우에만 사용할 수 있습니다(예를 들어, STEPNAME 변수는 프로젝트 레벨 알림에 대해서는 설정되지 않고 단계 레벨 알림에 대해서만 설정됨).

변수	포함
ACTION	빌드 제거의 경우 수행된 삭제의 유형을 설명합니다.
BID	작업 ID 번호를 지정합니다. 보고서에 액세스하기 위해 관리 콘솔로 돌아가는 링크를 생성하는 데 사용됩니다.
CONSOLEHOST	관리 콘솔 컴퓨터의 호스트 이름입니다.
CONSOLEPORT	관리 콘솔에서 사용되는 포트 번호입니다. 알림 템플릿에서 올바른 URL을 생성할 수 있게 합니다.
CONTEXTLOGLINKS	『FILT:』로 시작하는 로그의 행을 나열합니다(항목당 세 개의 컨텍스트 행이 있음). 시스템은 메시지에 관리 콘솔 로그 항목으로의 링크를 제공합니다.
DURATION	단계의 경우, 해당 단계 및 단계가 인라인한 모든 단계의 실행 시간을 초 단위로 지정합니다.
EID	환경 ID 번호를 지정합니다. 보고서에 액세스하기 위해 관리 콘솔로 돌아가는 링크를 생성하는 데 사용됩니다.
FULLNORMALLOG	작업의 각 단계에 대한 로그 정보를 표시합니다(자세한 로그에 나타나는 환경 설정 조치는 제외).
LINK	빌드의 경우 링크 이름을 지정합니다.
MESSAGE	실패 또는 경보 메시지에 대한 오류 또는 메시지 텍스트를 포함합니다.

변수	포함
ONFAIL	단계의 경우, 단계의 계속 특성을 보유합니다.
PATH	단계의 경우, 서버나 단계 같은 데이터 항목의 경로를 지정합니다(적절한 경우).
PID	프로젝트 ID 번호를 지정합니다. 보고서에 액세스하기 위해 관리 콘솔로 돌아가는 링크를 생성하는 데 사용됩니다.
PROJECTNAME	프로젝트 이름을 포함합니다.
RUNACTION	이메일 템플릿이 영향을 주는 변수를 지정합니다.
SELECTOR	단계 또는 프로젝트의 선택기 이름을 포함합니다.
SERVER	단계 또는 프로젝트의 선택기 이름을 포함합니다.
SID	단계의 경우 단계 ID 번호를 지정합니다. 보고서에 액세스하기 위해 관리 콘솔로 돌아가는 링크를 생성하는 데 사용됩니다.
SRVRHOST	단계의 경우, 단계에 대한 서버의 TCP/IP 호스트 이름을 포함합니다.
START	작업이 시작된 날짜/시간을 포함합니다.
STEPNAME	단계의 경우, 단계 이름을 포함합니다.
STEPNORMALLOG	단계의 경우, 작업의 현재 단계에 대한 로그 정보를 표시합니다(자세한 로그에 나타나는 환경 설정 조치는 제외).
TAG	작업의 태그 문자열을 포함합니다. \$BF_TAG와 동일한 값입니다.
TAILNORMALLOG	단계의 경우, STEPNORMALLOG 처럼 작동하지만 로그의 끝만 표시합니다.  빌드의 경우 FULLNORMALLOG 처럼 작동하지만 각 단계에 대한 로그의 끝만 표시합니다.  표시되는 행 수는 메일 템플릿의 로그 행 수 시스템 설정에 의해 제어됩니다.
UID	사용자 ID 번호를 지정합니다. 보고서에 액세스하기 위해 관리 콘솔로 돌아가는 링크를 생성하는 데 사용됩니다.
USEREMAIL	작업/이벤트 소유자의 이메일 주소를 포함합니다.
USERNAME	작업/이벤트 소유자의 전체 이름을 포함합니다.

## 인라인 프로젝트에 대한 알림

시스템은 인라인 프로젝트의 단계가 호출 프로젝트에 임베드된 것처럼 인라인 프로젝트에 대한 알림을 처리합니다.

- 프로젝트에 인라인 프로젝트가 포함된 경우, 인라인 프로젝트의 프로젝트 레벨 알림 설정은 무시됩니다. 프로젝트 A가 프로젝트 B를 인라인하는 경우, 프로젝트 B 시작, 패스 또는 실패에 대한 메시지가 전송되지 않습니다.
- 단계 레벨 알림 설정은 영향을 받지 않습니다. 단계에 패스 또는 실패 알림 액세스 그룹이 설정된 경우, 단계가 최상위 레벨 프로젝트 또는 인라인 프로젝트에 있는지 관계없이 해당 메시지가 전송됩니다.
- 인라인 프로젝트는 호출 프로젝트의 성공 또는 실패 여부를 판별하는 데 기여합니다. 예를 들어, 인라인 단계에서의 실패는 호출 프로젝트가 실패하게 하거나, 단계가 실패 시 계속으로 설정된 경우에는 호출 프로젝트의 상태를 "실패했지만 계속됨"으로 변경합니다.

---

## 스냅샷을 사용하여 프로젝트의 새 인스턴스 작성

프로젝트의 스냅샷을 작성하면 변경하거나 수정할 프로젝트의 새 인스턴스를 빠르게 작성할 수 있습니다. 프로젝트 스냅샷은 실행 가능한 별도 프로젝트입니다. 스냅샷을 사용하여 라이브러리의 새 인스턴스를 작성할 수도 있습니다.

### 프로젝트 스냅샷 개요

이 주제를 검토하면 프로젝트 스냅샷에 대해 알게 되며 환경 스냅샷 사용 방법을 이해할 수 있습니다.

### 프로젝트 스냅샷 유스 케이스

다음 예제는 프로젝트 스냅샷의 몇몇 공통 유스 케이스에 대해 설명합니다.

- 프로젝트 스냅샷을 찍어 프로젝트 구성을 변경하거나 기존 프로젝트와 함께 작업을 계속 실행하면서 새 도구 또는 스크립트 테스트를 수행합니다.
- 프로젝트 스냅샷을 임시 백업으로 또는 공식 아카이브의 일부로 저장합니다.
- 프로젝트 스냅샷을 찍어 외부 또는 내부 릴리스와 같은 마일스톤과 일치하는 적시 프로젝트 구성을 캡처합니다.

### 프로젝트 스냅샷 개념 및 용어

UI에서 스냅샷은 프로젝트에 대한 작업을 위한 몇 가지 새 개념과 용어를 소개합니다.


**프로젝트 스냅샷:** 스냅샷은 기존 프로젝트의 새 인스턴스입니다. 스냅샷에 대해 기억해야 할 몇 가지 중요 사항은 다음과 같습니다.

- 스냅샷은 별도 프로젝트입니다. 스냅샷 세트의 특정 스냅샷을 변경해도 세트의 다른 스냅샷에 영향을 주지 않습니다.

- 스냅샷은 실행 가능 프로젝트입니다. 스냅샷은 프로젝트 스냅샷을 작성할 때 스냅샷을 찍도록 선택하는 오브젝트 또는 소스 프로젝트(기본 스냅샷이라고도 함)와 연관된 오브젝트와 함께 실행됩니다.
- 스냅샷은 사본이 아닙니다.


프로젝트와 연관된 오브젝트의 스냅샷을 찍으면 오브젝트의 별도 인스턴스가 작성됩니다. 프로젝트를 복사하면 오브젝트 간 관계가 복사되며 선택기, 환경이나 인라인 또는 체인 프로젝트의 새 인스턴스가 작성되지 않습니다.

- 스냅샷은 프로젝트의 변경내용이 아닙니다.
  - 스냅샷은 두 프로젝트 스냅샷 간의 변경사항 비교를 지원하지 않습니다.
  - 프로젝트 스냅샷 변경사항은 소스 제어 시스템에서와 같이 버전 번호로 추적 또는 식별되지 않습니다. 그러나 버전 번호를 포함하는 스냅샷 이름 지정 구조(예: 7.5.0, 3.4.01)를 사용하여 프로젝트 스냅샷과 마일스톤을 상관시킬 수 있습니다.


**스냅샷 세트:** 스냅샷 세트는 특정 기본 스냅샷의 하위인 모든 프로젝트 스냅샷의 세트입니다. 최소한 세트에는 기본 또는 상위 스냅샷과 하위 스냅샷이 포함됩니다. UI에서 프로젝트 이름 옆에 있는 스냅샷 아이콘()이 프로젝트에 대해 스냅샷 세트가 작성되었음을 표시합니다.

**기본 스냅샷:** 처음에는 모든 프로젝트의 스냅샷 이름이 기본 스냅샷입니다. 기본 스냅샷을 다른 이름으로 변경할 수 있습니다. 기본 스냅샷은 스냅샷 세트의 상위입니다.

**기본 프로젝트 스냅샷:** 기본 프로젝트 스냅샷은 현재 실행 중인 프로젝트입니다. 세트에서 하나의 스냅샷만 기본값이 될 수 있습니다. 기본 스냅샷을 지정하지 않는 경우 기본값은 기본 스냅샷입니다.

- UI에서 기본 스냅샷은 프로젝트 목록의 최상위 레벨에 표시됩니다. 프로젝트 목록을 표시하려면 **프로젝트** 또는 **작업 > 시작**을 선택하십시오.
- 스냅샷이 있는 프로젝트를 인라인 프로젝트 또는 체인 프로젝트로 선택하는 경우 목록 상자에서 다른 프로젝트 스냅샷을 선택하지 않는 한 기본 프로젝트 스냅샷이 사용됩니다.
- 프로젝트 스냅샷 세트의 다른 스냅샷에 액세스하여 작업을 수행하려면 스냅샷 아이콘()을 클릭해야 합니다.

## 프로젝트 스냅샷 보기

스냅샷 보기를 표시하려면 스냅샷 아이콘()을 선택하십시오. UI에서 스냅샷 보기는 다음과 같은 세트 내 스냅샷의 계층 구조를 보여줍니다.

- 기본 스냅샷은 최상위 레벨이며 고유 이름을 지정하지 않는 경우 기본 스냅샷이라고 합니다.



- 모든 프로젝트 스냅샷은 기본 스냅샷의 하위입니다. 기본 스냅샷이 같은 하위는 스냅샷 열의 동일한 레벨에서 들여쓰입니다.
- 하위 스냅샷에서 작성된 프로젝트 스냅샷은 하위 스냅샷의 하위가 되며 스냅샷 열의 다음 레벨에서 들여쓰입니다.

## 프로젝트 스냅샷 계획

기본 프로젝트 스냅샷을 선택하고 프로젝트 스냅샷의 이름을 지정하는 몇 가지 우수 사례를 검토합니다.

- 세트의 기본 스냅샷 선택 전략

UI는 스냅샷 세트에 대해 하나의 기본 또는 현재 프로젝트 스냅샷만 인식합니다. 기본 스냅샷 선택을 위해서는 다음과 같은 동일 전략을 사용하십시오.

- 기본 스냅샷을 기본 스냅샷으로 사용

이 전략을 사용하여 스냅샷을 적시 백업으로 작성하고 백업된 프로젝트 스냅샷을 변경하지 않습니다. 기본 스냅샷을 변경하고 기본 스냅샷 프로젝트만을 사용하여 작업을 계속 실행합니다.

- 최신 스냅샷을 기본 스냅샷으로 사용

이 전략을 사용하는 경우 새 프로젝트를 작성하는 이유는 새 프로젝트를 새 기본 프로젝트 스냅샷으로 설정하기 위해서입니다. 기본 스냅샷 또는 이전 프로젝트 스냅샷으로 변경하지 않습니다. 작업을 실행하려면 최신 스냅샷을 사용합니다.

- 세트의 스냅샷 이름 지정 규칙 식별

프로젝트 스냅샷 이름은 프로젝트 스냅샷 세트에서 고유해야 합니다.

프로젝트 스냅샷 이름을 작성하려면 다음 기준을 사용하십시오.

- 이름은 스냅샷 사용법 또는 용도를 설명해야 합니다.
- 이름 지정 규칙은 정의된 표준을 따라야 합니다. 스냅샷 탭의 주석 상자를 사용하여 이름 지정 규칙을 설명할 수 있습니다.

- 세트에 단일 프로젝트 이름 사용

프로젝트 스냅샷을 작성한 후 프로젝트의 이름을 변경할 옵션이 있습니다. 프로젝트 이름을 변경하는 경우 모든 프로젝트 스냅샷에 대해 업데이트됩니다.

## 프로젝트 스냅샷 옵션

스냅샷을 작성할 때 스냅샷에 포함할 오브젝트를 선택해야 합니다. 다음 표는 Build Forge 오브젝트에 사용 가능한 옵션에 대해 설명합니다. 표에 다음 오브젝트에 대한 열이 있습니다.

- 프로젝트 스냅샷에 자동으로 포함되는 오브젝트.



- 프로젝트 스냅샷을 작성할 때 선택하는 경우 선택적으로 작성되고 포함되는 오브젝트.

이러한 오브젝트의 경우 프로젝트 스냅샷과 스냅샷 이름이 같은 새 오브젝트가 UI에서 작성됩니다. 예를 들어, 프로젝트 스냅샷의 이름이 `release_7.1`인 경우 환경, 선택기, 인라인 프로젝트 또는 라이브러리와 체인 프로젝트 또는 라이브러리의 스냅샷 이름 또한 `release_7.1`입니다.

- 프로젝트 스냅샷에 포함되지 않는 오브젝트. 이러한 오브젝트는 수동으로 작성하여 프로젝트에 추가해야 합니다.

	선택적으로 프로젝트 스냅샷과 함께 포함됨	
프로젝트 스냅샷에 자동으로 포함됨	* 복사 전용; 별도 인스턴스가 작성되지 않음	프로젝트 스냅샷과 함께 포함되지 않음
프로젝트 단계 및 설정(로그 필터, 알림 그룹 등)	프로젝트 및 해당 단계의 환경	프로젝트 노트
프로젝트 태그	스냅샷 환경에 대한 포함 유형의 환경 변수로 추가된 환경	
	인라인 프로젝트 또는 라이브러리와 해당 단계	
	체인 프로젝트 또는 라이브러리와 해당 단계	
	프로젝트 및 해당 단계의 선택기	
	스냅샷 선택기에 대해 포함 유형의 선택기 특성으로 추가된 선택기	
	* 프로젝트 레지스터(복사됨)	
	* 프로젝트 태그 변수 값(복사됨)	
	* 알림 템플릿(복사됨)	
	* 어댑터 링크(복사됨)	

## 스냅샷 권한의 액세스 그룹 확인 및 편집

사용자에게 스냅샷을 작성하고 기본 스냅샷을 설정하는 데 필요한 권한이 있는지 확인합니다. 권한이 없는 경우 액세스 그룹을 사용하여 사용자에게 권한을 지정하십시오.

권한은 액세스 그룹을 통해 사용자에게 지정되며 액세스 그룹은 Build Forge가 제공하거나 Build Forge 관리자가 작성할 수 있습니다.

스냅샷 권한에 지정된 액세스 그룹을 확인하고 편집하려면 다음을 수행하십시오.

1. 관리 > 권한을 선택하십시오.
2. 권한 목록에서 모두 표시를 선택하여 모든 권한을 나열하십시오.
3. 올바른 액세스 그룹 및 사용자에게 다음 스냅샷 권한에 대한 액세스 권한이 있는지 확인하십시오.


스냅샷 작성	프로젝트, 환경 및 선택기의 스냅샷을 작성하는 데 필요한 사용자 권한.
기본 스냅샷 설정	프로젝트, 환경 및 선택기의 기본 스냅샷을 설정 또는 변경하는 데 필요한 사용자 권한.



## 기존 프로젝트 또는 프로젝트 스냅샷에서 프로젝트 스냅샷 작성

프로젝트 스냅샷을 작성하면 프로젝트의 새 인스턴스와 스냅샷을 찍을 오브젝트가 작성됩니다. 스냅샷은 사본이 아니라 프로젝트의 새로운 실행 가능 인스턴스입니다.

**참고:** 관리 콘솔은 처음 2999개의 프로젝트 뒤에 있는 프로젝트에 대한 스냅샷을 표시하지 않습니다.

1. 스냅샷을 찍을 프로젝트 또는 프로젝트 스냅샷 옆에 있는 편집 아이콘을 클릭하십시오.

- 기본 프로젝트 스냅샷을 찍으려면 프로젝트 목록(프로젝트)에서, 최상위 레벨 스냅샷 옆에 있는 편집 아이콘()을 클릭하십시오.

- 비기본 프로젝트 스냅샷을 찍으려면 스냅샷 아이콘()을 클릭하십시오. 스냅샷 보기에는 세트의 프로젝트 스냅샷이 표시됩니다. 비기본 프로젝트 스냅샷 옆에 있는 편집 아이콘()을 클릭하십시오.

2. 새 스냅샷 작성을 클릭하십시오.

3. 스냅샷 탭의 이름에 스냅샷 이름을 입력하십시오.

이름은 프로젝트 스냅샷 세트에서 고유해야 합니다. 이름은 프로젝트 스냅샷을 찍는 모든 오브젝트에 지정됩니다.

4. 프로젝트 스냅샷을 작성할 때 스냅샷을 찍을 Build Forge 오브젝트를 선택하십시오. 선택할 수 있는 오브젝트는 다음 표에 설명되어 있습니다.

오브젝트	설명
기본값	UI에서 기본 프로젝트 스냅샷은 프로젝트 목록의 최상위 레벨에 표시됩니다.  프로젝트 목록을 표시하려면 프로젝트 또는 작업 > 시작을 선택하십시오.
프로젝트 환경 포함	프로젝트의 프로젝트 및 단계 환경의 스냅샷을 찍습니다.
다음 환경 포함	프로젝트 환경 포함이 선택된 경우 포함 유형의 환경 변수로 포함되는 다른 환경의 스냅샷도 찍습니다.
프로젝트 선택기 포함	프로젝트에 포함된 프로젝트 및 단계 선택기의 스냅샷을 찍습니다.
다음 선택기 포함	프로젝트 선택기 포함이 선택된 경우 포함 유형의 선택기 특성으로 포함되는 다른 선택기의 스냅샷도 찍습니다.
프로젝트 어댑터 링크 복제	어댑터 링크를 스냅샷의 일부로 복사합니다.  어댑터 링크는 프로젝트에 어댑터를 추가합니다. 어댑터는 프로젝트에서 첫 번째 단계(단계 0)로 실행됩니다.



오브젝트	설명
프로젝트 레지스터 복제	프로젝트 레지스터를 스냅샷의 일부로 복사합니다.
프로젝트 태그 변수값 복제	프로젝트 태그 변수의 태그 값을 복사합니다. 태그 변수는 자동으로 복사되지만 해당 값은 복사되지 않습니다. 태그 값을 복사하지 않으면 1로 재설정됩니다.
프로젝트 템플릿 복제	프로젝트 레벨과 단계 레벨에서 설정된 패스 및 실패 알림 이벤트의 알림 템플릿을 복사합니다.
체인 프로젝트 포함	프로젝트 레벨 또는 단계 레벨에서 참조되는 체인 프로젝트 또는 라이브러리와 해당 단계의 스냅샷을 찍습니다.  체인은 프로젝트 패스/실패 또는 단계 패스/실패 조건으로 트리거됩니다.
프로젝트 인라인 포함	인라인 프로젝트 또는 라이브러리와 단계 레벨에서 참조되는 해당 단계의 스냅샷을 찍습니다.  인라인은 단계로 트리거되어 단계가 완료된 후 실행됩니다.

5. 저장을 클릭해서 프로젝트 스냅샷을 저장하십시오.

## 기본 프로젝트 스냅샷 변경

기본 프로젝트 스냅샷은 프로젝트 스냅샷 세트의 최상위 레벨 스냅샷이며 프로젝트 목록(프로젝트)에 표시됩니다.

기본 프로젝트 스냅샷을 변경하려면 새 기본값으로 설정할 스냅샷의 스냅샷 정의를 편집하십시오.

1. 프로젝트를 선택하십시오.
2. 프로젝트 목록에서 기본 프로젝트 스냅샷에 대한 스냅샷 아이콘()을 클릭하십시오.
3. 스냅샷 목록에서 새 기본값이 될 프로젝트 스냅샷에 대한 편집 아이콘()을 클릭하십시오.
4. 기본값으로 설정을 클릭하십시오.
5. 중요: 팝업에서 확인 또는 취소를 선택하십시오.



확인	참조 업데이트: 이전 기본값을 참조하는 오브젝트에 대해 이전 기본 프로젝트 스냅샷의 참조를 새 기본값으로 업데이트합니다.
취소	참조를 업데이트하지 않음: 이전 기본값을 참조하는 오브젝트에 대해 참조를 새 기본 프로젝트 스냅샷으로 업데이트하지 않습니다.

## 프로젝트 스냅샷의 스냅샷 이름 변경

프로젝트 스냅샷의 스냅샷 이름을 변경할 수 있으며 프로젝트 스냅샷을 작성할 때 스냅샷을 찍기로 선택한 오브젝트의 스냅샷 이름도 변경할 수 있습니다.

기본 스냅샷의 경우 이 옵션을 사용하여 기본 이름인 기본 스냅샷을 단일 프로젝트 스냅샷 전용 또는 현재와 미래의 모든 프로젝트에 대한 다른 스냅샷 이름으로 변경할 수 있습니다.

스냅샷 이름을 변경하려면 다음을 수행하십시오.

1. 프로젝트를 선택하십시오.
2. 프로젝트 목록에서 기본 프로젝트 스냅샷에 대한 스냅샷 아이콘()을 클릭하십시오.
3. 스냅샷 목록에서 프로젝트 스냅샷에 대한 편집 아이콘()을 클릭하십시오.
4. 스냅샷 탭을 선택하십시오.
5. 이름에 새 이름을 입력하십시오.
6. 선택적: 주석에 주석을 입력하십시오.
7. 중요: 팝업에서 확인 또는 취소를 선택하십시오.

확인	<p>프로젝트 스냅샷 이름과 기타 스냅샷 오브젝트 이름 변경: 프로젝트 스냅샷을 작성할 때 스냅샷을 찍기로 선택한 오브젝트의 경우 이러한 오브젝트와 프로젝트 스냅샷의 이름을 변경합니다.</p> <p>기본 스냅샷의 경우: 모든 현재 프로젝트 스냅샷과 모든 미래 프로젝트 스냅샷의 이름인 기본 스냅샷을 변경합니다.</p>
취소	<p>프로젝트 스냅샷 이름은 변경하고 기타 스냅샷 오브젝트 이름은 변경하지 않음: 프로젝트 스냅샷을 작성할 때 스냅샷을 찍기로 선택한 오브젝트의 경우 이러한 오브젝트의 이름을 변경하지 마십시오. 프로젝트 스냅샷 이름만 변경합니다.</p> <p>기본 스냅샷의 경우: 모든 현재 프로젝트 스냅샷과 모든 미래 프로젝트 스냅샷의 이름인 기본 스냅샷을 유지합니다.</p>


## 프로젝트 스냅샷 세트의 스냅샷 액세스 및 보기

프로젝트 스냅샷을 작성하면 두 개 이상의 프로젝트(기본 프로젝트와 새 프로젝트 스냅샷)를 포함하는 스냅샷 세트가 작성됩니다.



스냅샷 세트의 모든 프로젝트 스냅샷을 보려면 다음을 수행하십시오.

1. 프로젝트를 선택하십시오.

프로젝트 목록에는 프로젝트와 프로젝트 스냅샷의 목록이 표시됩니다. 최상위 레벨 스냅샷이 기본 프로젝트 스냅샷입니다.

2. 스냅샷 아이콘()을 클릭하여 스냅샷 세트의 프로젝트 스냅샷을 표시하십시오.


스냅샷 보기에서 다음을 수행할 수 있습니다.

- 새 프로젝트 스냅샷을 작성하십시오. 시작하려면 편집 아이콘  을 클릭하십시오.
- 프로젝트의 기본 스냅샷을 변경하십시오. 편집 아이콘()을 클릭하고 기본값으로 설정을 클릭하십시오.
- 표준 프로젝트의 경우와 같이 프로젝트 스냅샷 정의를 편집하십시오.

## 기본 프로젝트 스냅샷에 대한 작업 시작

기본 프로젝트 스냅샷을 시작하려면 빠른 시작 아이콘 또는 프로젝트 시작 페이지를 사용하십시오.

빠른 시작 아이콘을 사용하여 기본 프로젝트 스냅샷을 시작하려면 다음을 수행하십시오.

1. 프로젝트를 선택하십시오.
2. 프로젝트 목록에서 빠른 시작 아이콘()을 클릭하십시오.

작업은 선택기, 클래스, 태그 형식 및 환경의 기본값을 사용하여 실행됩니다.

프로젝트 시작 페이지를 사용하여 기본 프로젝트 스냅샷을 시작하려면 다음을 수행하십시오.

1. 작업 → 시작을 선택하십시오.
2. 프로젝트 목록에서 기본 스냅샷의 프로젝트 이름을 클릭해서 프로젝트 시작 페이지를 표시하십시오.

프로젝트 시작 페이지에서 환경 변수인 선택기, 클래스 및 태그 형식을 변경할 수 있습니다.

선택기에 선택기 스냅샷이 있는 경우 스냅샷 필드 아래 필드에는 선택할 수 있는 선택기 스냅샷이 나열됩니다.

참고: 기본 선택기 스냅샷을 빠르게 선택하려면 기본 스냅샷을 선택하십시오. 기본 스냅샷은 기본 선택기 스냅샷의 이름으로 매핑됩니다.


3. 실행을 클릭하십시오.

## 비기본 프로젝트 스냅샷에 대한 작업 시작

비기본 프로젝트 스냅샷을 시작하려면 빠른 시작 아이콘 또는 프로젝트 시작 페이지를 사용하십시오.

빠른 시작 아이콘을 사용하여 비기본 프로젝트 스냅샷을 시작하려면 다음을 수행하십시오.

1. 프로젝트를 선택하십시오.

2. 프로젝트 목록에서 기본 스냅샷의 스냅샷 아이콘()을 클릭하십시오.


스냅샷 보기에는 세트의 프로젝트 스냅샷이 표시됩니다.

3. 비기본 프로젝트 스냅샷 옆에 있는 빠른 시작 아이콘()을 클릭하십시오.

작업은 선택기, 클래스, 태그 형식 및 환경의 기본값을 사용하여 실행됩니다.

프로젝트 시작 페이지를 사용하여 비기본 프로젝트 스냅샷을 시작하려면 다음을 수행하십시오.

1. 작업 → 시작을 선택하십시오.

2. 프로젝트 목록에서 기본 스냅샷의 스냅샷 아이콘()을 클릭하십시오.

스냅샷 보기에는 세트의 프로젝트 스냅샷이 표시됩니다.

3. 비기본 프로젝트 스냅샷의 프로젝트 이름을 클릭해서 프로젝트 시작 페이지를 표시하십시오.

프로젝트 시작 페이지에서 환경 변수인 선택기, 클래스 및 태그 형식을 변경할 수 있습니다.

선택기에 선택기 스냅샷이 있는 경우 스냅샷 필드 아래 필드에는 선택할 수 있는 선택기 스냅샷이 나열됩니다.

참고: 기본 선택기 스냅샷을 빠르게 선택하려면 기본 스냅샷을 선택하십시오. 기본 스냅샷은 기본 선택기 스냅샷의 이름으로 매핑됩니다.

4. 실행을 클릭하십시오.


## 프로젝트 스냅샷 삭제

프로젝트 삭제 또는 프로젝트 제거 옵션을 사용하여 프로젝트 스냅샷을 삭제합니다.


- 프로젝트 삭제 옵션은 프로젝트 스냅샷에 대한 작업이 없는 경우 또한 다른 오브젝트(클래스 또는 다른 프로젝트)가 프로젝트 스냅샷을 인라인 또는 체인 프로젝트로 참조하지 않는 경우 사용할 수 있습니다.
- 제거 옵션은 Build Forge 데이터베이스에서 프로젝트와 해당 작업을 삭제하고 다른 오브젝트의 프로젝트에 대한 참조를 제거합니다.

프로젝트 스냅샷을 삭제하려면 다음을 수행하십시오.

1. 프로젝트를 선택하십시오.

2. 프로젝트 목록에서 기본 스냅샷의 스냅샷 아이콘()을 클릭하십시오.

스냅샷 보기에는 세트의 프로젝트 스냅샷이 표시됩니다.

3. 삭제될 프로젝트 스냅샷 옆에 있는 편집 아이콘()을 클릭하십시오.
4. 정리 또는 프로젝트 삭제를 클릭하십시오.





## 제 20 장 단계에 대한 작업

이 주제는 관리 콘솔에서 단계를 작성 및 관리하는 방법에 대해 설명합니다.

### 단계 정보

단계는 프로젝트의 컴포넌트입니다. 프로젝트가 작업으로 실행될 때 각 단계는 순서대로 실행됩니다. 단계에는 하나 이상의 명령이 포함되어 있으며 동작에 영향을 주는 단계 특성이 있습니다.

### 단계 패널 정보

#### 세부사항 탭

단계 특성은 단계 실행 방법, 단계 출력 처리 방법 및 단계 완료 시 수행할 작업을 지정합니다. 또한 단계는 다른 프로젝트나 라이브러리를 실행할 수 있습니다.

단계 특성을 보려면 프로젝트 내에서 단계를 선택하십시오. 세부사항 탭은 기본적으로 표시됩니다. 단계 특성을 표시합니다.

단계 특성이 명시적으로 설정되지 않은 경우, 해당 값은 프로젝트에서 상속됩니다. 단계에 대한 단계 특성 세트는 상속된 값을 대체합니다.

The screenshot shows the '세부사항' (Details) tab for a stage configuration. The interface includes a header with '단계: [ 새 단계 추가 ]' and '단계 저장'. Below the header, there are several configuration fields:

- 이름:** (Name) - Text input field.
- 디렉토리:** (Directory) - Text input field with a slash (/).
- 단계 유형:** (Stage Type) - Dropdown menu with '일반' (General) selected.
- 명령:** (Command) - Large text area for entering commands.
- 활성:** (Active) - Dropdown menu with '사용' (Use) selected.
- 경로:** (Path) - Dropdown menu with '상대' (Relative) selected.
- 인라인:** (Inline) - Dropdown menu with '-- 없음 --' (None) selected.
- 역세스:** (Reverse) - Dropdown menu with '-- 기본값 --' (Default) selected.
- 환경:** (Environment) - Dropdown menu with '-- 없음 --' (None) selected.
- 선택기:** (Selector) - Dropdown menu with '-- 기본값 --' (Default) selected.
- 결과:** (Result) - Dropdown menu with '-- 종료 코드 --' (End Code) selected.
- 패스 체인:** (Pass Chain) - Dropdown menu with '-- 없음 --' (None) selected.
- 실패 체인:** (Fail Chain) - Dropdown menu with '-- 없음 --' (None) selected.
- 제한시간(분):** (Timeout) - Text input field with '5'.
- 스레드:** (Thread) - Dropdown menu with '아니오' (No) selected.
- 패스 알림:** (Pass Notification) - Dropdown menu with '-- 없음 --' (None) selected.
- 실패 알림:** (Fail Notification) - Dropdown menu with '-- 없음 --' (None) selected.
- 브로드캐스트:** (Broadcast) - Dropdown menu with '아니오' (No) selected.
- 실패 시:** (On Fail) - Dropdown menu with '정지' (Stop) selected.
- 패스 대기:** (Pass Wait) - Dropdown menu with '아니오' (No) selected.
- 실패 대기:** (Fail Wait) - Dropdown menu with '아니오' (No) selected.

단계 특성은 다음을 포함합니다.

**이름** 단계의 이름입니다. 시스템 및 로그의 단계에 대한 레이블로 사용됩니다.

**활성** 단계가 실행되는지 여부를 지정합니다. 단계는 기본적으로 사용 가능합니다. 단계의 실행을 방지하려면 사용 불가능을 선택하십시오. 작업에서 실행되는 사용 불가능한 단계를 사용할 수 없습니다.

#### 디렉토리

단계 명령이 실행되는 위치를 설정합니다. 시스템은 모든 작업에 대한 고유 디렉토리를 자동으로 작성합니다. **디렉토리** 필드는 작업 중에 프로젝트가 생성한 디렉토리의 명령을 실행하는 편리한 방법을 제공합니다(Build Forge는 디렉토리 필드에 언급된 디렉토리를 생성하지 않음).

**경로** 디렉토리가 절대 경로인지 상대 경로인지를 지정합니다.

- **상대:** 단계 명령은 서버, 프로젝트, 작업 및 단계 디렉토리를 함께 추가하여 찾은 경로에서 실행됩니다.
- **절대:** 단계 명령이 서버 및 단계 디렉토리를 함께 추가하여 찾은 경로에서 실행됩니다. 이 옵션을 사용하여 프로젝트 디렉토리 구조에 없는 디렉토리에 액세스할 수 있습니다. 예제: 서버에 영구적으로 설치된 애플리케이션을 실행하는 데 사용될 수 있습니다.

#### 단계 유형

단계의 실행 방법을 판별합니다. 이 특성은 명령의 콘텐츠 및 인라인에 지정된 프로젝트(있는 경우)에 영향을 줍니다.

- **일반:** 단계가 한 번 실행됩니다.
- **조건부:** 조건 특성의 표현식이 true로 평가되면 단계가 한 번 실행됩니다. 조건부 선택 시 조건, Else 인라인 및 Else 명령 특성이 표시됩니다. 조건 특성이 false로 평가되면 명령 및 인라인이 실행되지 않습니다. 대신 Else 명령 및 Else 인라인이(지정되어 있는 경우) 실행됩니다.
- **While 루프:** 단계가 여러 번 실행될 수 있습니다. 조건 특성의 표현식이 false로 되거나 최대 반복 수에 도달할 때까지 실행됩니다. While 루프 선택 시 조건 및 최대 반복 수 특성이 표시됩니다.

While 루프의 각 반복에서 선택기가 평가되어 반복에 사용할 서버를 판별합니다.

**인라인** 현재 프로젝트와 함께 인라인으로 실행할 프로젝트나 라이브러리를 지정합니다. 프로젝트 또는 라이브러리의 단계가 현재 프로젝트의 환경 및 대부분의 특성을 사용하여 실행됩니다. 그러나 시스템은 인라인 프로젝트의 선택기를 인라인 단계에 대한 기본 선택기로 사용합니다. 현재 단계 뒤에 지정된 프로젝트의 단계를 복사하는 것과 같은 동작입니다.

**액세스** 액세스 그룹을 선택하여 단계를 사용하도록 허용된 사용자를 정의하십시오. 이 특성을 사용하여 프로젝트의 특정 단계로 액세스를 제한할 수 있습니다. 단계에 대한 액세스 그룹의 구성원이 아닌 사용자가 단계를 포함하는 프로젝트를 실행하는 경우 해당 단계는 건너뛸니다.

프로젝트 기본값을 선택하면 단계가 프로젝트의 액세스 특성을 상속하게 됩니다.

#### 최대 반복 수

단계 유형이 While 루프인 경우에만 표시됩니다. 루프에서 실행할 수 있는 단계의 최대 반복 수를 지정합니다. 시스템에 부과된 기본값은 100입니다. 단계 로그에서 성공적으로 완료된(패스된) 단계가 표시됩니다. 최대치에 도달한 경우 단계 실패를 사용하여 최대 반복 수에 도달하면 단계가 실패하도록 작성하십시오.

작업이 실행 중인 경우, 읽기 전용 변수 BF\_ITERATION은 성공적으로 입력된 반복 수를 포함합니다. 작업이 중지되고 다시 시작되는 경우 BF\_ITERATION의 반복에서 다시 시작됩니다.

#### 최대치에 도달한 경우 단계 실패

예인 경우, 최대 반복 수에 도달하면 While 루프 단계가 실패합니다. 아니오인 경우, 단계가 패스됩니다.

#### Else 인라인

단계 유형이 조건부인 경우에만 표시됩니다. 지정된 조건이 false인 경우 인라인으로 실행할 프로젝트를 지정합니다. 기본값은 아니오입니다.

#### 명령

하나 이상의 명령입니다. 명령은 운영 체제 명령, 점 명령 또는 둘의 조합일 수 있습니다. 400 페이지의 『단계가 실행하는 방법』의 내용을 참조하십시오.

**조건** 조건부 또는 While 루프의 단계 유형을 선택한 경우에만 표시됩니다.

- 조건부: 조건이 true로 평가되면 명령이 실행됩니다.
- While 루프: 조건이 true로 평가되는 한 명령을 여러 번 실행할 수 있습니다. 최대 반복 수를 사용하여 한계를 설정할 수 있습니다.

조건은 선택된 서버 자원에서 실행할 함수 또는 명령일 수 있습니다.

- 함수가 사용되는 경우에는 조건 필드의 처음에 사용되어야 합니다. Build Forge 엔진에서 평가합니다. 서버 자원에 보내지지 않습니다. 함수 및 함수 사용 방법에 관한 지시사항의 목록은 407 페이지의 『조건 함수』를 참조하십시오.
- 명령은 선택된 서버에서 실행됩니다. 여기서 사용된 모든 명령은 에이전트의 셸 환경에서 유효해야 합니다. 실행에서 리턴 코드는 조건의 패스 또는 실패 여부를 판별합니다.

프로젝트에 사용 가능한 Build Forge 변수는 조건 표현식에 사용할 수 있습니다. 변수를 표현할 수 있는 방법과 변수를 평가하는 방법에 대한 자세한 정보는 324 페이지의 『단계에서 변수 해석』을 참조하십시오.

## Else 명령

조건부 단계 유형을 선택한 경우에만 표시됩니다. 조건이 false로 평가되는 경우에 실행할 명령을 지정합니다.

## 환경

명령을 실행하기 전에 적용할 환경을 지정합니다. 이 환경의 값은 서버 환경, 프로젝트 환경 및 단계 변수로부터 상속된 모든 값을 대체합니다.

**선택기** 이 단계의 서버를 선택하기 위해 사용할 선택기를 지정합니다. 기본값을 그대로 둔 경우 프로젝트의 선택기에 의해 판별된 서버에서 단계가 실행됩니다.

## 브로드캐스트

이 상자가 선택된 경우, 현재 선택기(지정된 경우에는 단계 선택기이고, 그렇지 않으면 프로젝트 선택기임)와 일치하는 모든 서버에서 단계를 실행합니다. 실행 시, 시스템은 브로드캐스트 단계를 일련의 단계로 바꾸고(각 서버에 대해 하나 씩), 브로드캐스트 단계의 스프레드 특성에 따라 이를 직렬 또는 병렬로 실행합니다.

**다시 시작 시 브로드캐스트 단계 동작:** 브로드캐스트 단계가 다시 시작되면 브로드캐스트되지 않습니다. 해당 설정은 단계를 새로 시작하는 경우에만 적용됩니다. 다시 시작하면 엔진이 단계에 대해 단일 서버를 랜덤으로 선택합니다.

## 제한시간(분)

현재 명령이 결과물을 생성하기까지 시스템이 대기하는 시간(분)을 지정합니다 (기본값: 5분). 단계가 에이전트에 적절히 연결된 경우 값 0은 단계의 제한시간이 초과되지 않음을 의미합니다. 제한시간 값에 도달하면, 시스템은 단계를 실패로 처리합니다. 단계가 실패 시 계속으로 설정되어 있지 않으면, 프로젝트도 또한 실패합니다.

**결과** 결과 특성은 시스템이 단계가 성공했는지 또는 실패했는지 여부를 판단하는 방법을 판별합니다. 명령 셸이 리턴하는 종료 코드를 기반으로 성공을 판별하려면 종료 코드 기본값을 사용하십시오. 명령 결과물을 검사하는 로그 필터를 선택할 수도 있습니다. 로그 필터를 선택하기 위해서는 먼저 로그 필터를 작성해야 합니다.

## 실패 시

단계가 실패하면 작업을 중지할지 계속할지 여부를 지정합니다. 기본적으로, 시스템이 작업을 중지합니다.

**스레드** 예인 경우, 이 단계를 기타 단계와 병렬로 실행합니다. 이 단계의 스레딩(단계를 다른 단계와 병렬로 실행)을 허용하려면 이 특성을 예로 설정하십시오. 스레딩을 방지하려면 특성을 아니오로 설정하십시오. 스레드 단계 블록을 분리하려면 이 특성을 결합으로 설정하십시오. 첫 번째 단계 세트를 완료해야 결합 단계 다음에 오는 다음 스레드 단계 세트를 시작할 수 있습니다.

### 패스 알림

단계가 패스하면 알림을 받을 액세스 그룹을 지정합니다.

### 패스 체인

현재 단계 패스 시 실행할 프로젝트를 지정합니다. ("경고" 상태를 갖는 단계는 패스로 계수되며 패스 체인을 실행합니다.)

### 패스 대기

이 상자가 선택된 경우, 시스템은 패스 체인 프로젝트가 완료될 때까지 현재 프로젝트를 일시중단합니다. 이 단계(또는 해당 프로젝트)가 취소되면 체인 형식으로 연결된 프로젝트도 취소됩니다. 검사되지 않는 경우 체인 형식으로 연결된 프로젝트가 비동기식으로 시작되며 현재 프로젝트가 다음 단계로 계속됩니다.

### 실패 알림

단계가 실패하면 알림을 받을 액세스 그룹을 지정합니다.

### 실패 체인

현재 단계 실패 시 실행할 프로젝트를 지정합니다. (실패 시 계속으로 설정된 단계는 실패로 계수되며, 단계에 지정된 모든 실패 체인을 실행합니다.)

### 실패 대기

이 상자가 선택된 경우, 시스템은 실패 체인 프로젝트가 완료될 때까지 현재 프로젝트를 일시중단합니다. 이 단계(또는 해당 프로젝트)가 취소되면 체인 형식으로 연결된 프로젝트도 취소됩니다.


## 노트 탭

노트 탭에는 단계에 대해 작성된 노트의 시간소인 목록이 포함됩니다. 노트는 수동으로 작성하며 편집 내용이 단계에 자동으로 기록되지 않습니다. 이 탭은 현재 노트 수(예: **Notes (2))**)를 보여줍니다.

노트를 추가하려면 다음을 수행하십시오.


1. 노트 탭을 클릭하십시오.
2. 텍스트 필드에 새 노트를 작성하십시오.
3. 제출을 클릭하십시오.

노트를 편집하려면 다음을 수행하십시오.

1. 노트 탭을 클릭하십시오.
2. 편집하려는 노트 옆에 있는 편집 아이콘()을 클릭하십시오. 노트를 편집하십시오.
3. 제출을 클릭하십시오.

노트를 삭제하려면 다음을 수행하십시오.

1. 노트 탭을 클릭하십시오.

- 휴지통 아이콘()을 클릭하십시오. 노트 삭제를 확인하는 프롬프트가 표시됩니다.
- 확인을 클릭하십시오.

---

## 단계 추가

### 이 태스크 정보

이 프로시저는 프로젝트 끝에 단계를 추가합니다. 단계를 첫 번째 단계로 또는 기존 단계 사이에 삽입하는 방법에 대해서는 399 페이지의 『추가 단계 조작』을 참조하십시오.

#### 프로시저

- 프로젝트를 선택한 후 프로젝트 이름을 클릭하십시오. 시스템이 선택된 프로젝트의 단계 목록을 표시합니다.
- 기본 패널의 맨 위에 있는 단계 추가를 클릭하십시오. 시스템이 비어 있는 단계 세 부사항 패널을 표시합니다.
- 특성의 값을 입력하십시오. 이름 및 명령은 필수입니다.
- 단계 저장을 클릭하십시오.

---

## 단계 편집


### 프로시저

- 프로젝트를 선택한 후 프로젝트 이름을 클릭하십시오. 시스템이 선택된 프로젝트의 단계 목록을 표시합니다.
- 단계 이름을 클릭하십시오. 시스템이 패널의 아래 부분에 단계의 특성을 표시합니다.
- 특성을 편집하십시오.
- 단계 저장을 클릭하십시오.

---

## 단계 사용 안함

### 프로시저

- 프로젝트를 선택한 후 프로젝트 이름을 클릭하십시오. 시스템이 선택된 프로젝트의 단계 목록을 표시합니다.
- 단계 이름 앞에 있는 선택란을 클릭하십시오. 단계를 선택하면 , 프로젝트에 다음 영향을 미칩니다.
  - 프로젝트를 하나의 작업으로 실행할 때 해당 단계가 실행되지 않습니다.


- 이 단계는 정상적으로 작업을 시작할 때 단계 목록에서 비활성화됩니다. 작업 > 시작을 선택하고, 작업 이름을 클릭한 후 작업 단계를 클릭하십시오. 단계가 표시되지만 시작 중인 작업에 대해 활성화할 수 없습니다.

#### 참고:

단계의 세부사항 탭에서 활성 특성을 사용 불가능으로 설정하여 단계를 사용 불가능하게 할 수도 있습니다. 단계를 저장할 때 선택란을 체크합니다.

## 추가 단계 조작

단계에 대해 작업하려면 프로젝트를 선택한 후 프로젝트 이름을 클릭하십시오. 시스템이 선택된 프로젝트의 단계 목록을 표시합니다.

추가 옵션을 표시하려면 단계 이름 앞에 있는 조치 아이콘()을 클릭하십시오.

- 새 단계 삽입: 단계를 선택된 단계 위에 삽입합니다.
- 단계 복제: 단계 및 모든 해당 특성을 복사하십시오. 이름이 변경되어 단계 끝에 숫자를 추가합니다. Build라는 단계를 복사하면 Build COPY 0이라는 새 단계가 생성됩니다. 숫자는 자동으로 설정됩니다. 다음 위치에 복사할 수 있습니다.
  - 맨 위: 목록의 시작
  - 위: 현재 단계의 바로 앞
  - 아래: 현재 단계의 바로 아래
  - 맨 아래: 목록의 맨 아래
- 단계 이동: 단계를 목록의 다른 위치로 이동합니다. 다음 위치로 이동할 수 있습니다.
  - 맨 위: 목록의 시작
  - 위로: 한 위치 위로 이동
  - 다음으로 이동...: 대화 상자가 단계 번호를 요청합니다. 단계가 해당 위치로 이동되고 필요에 따라 다른 단계 번호가 조정됩니다.
  - 아래로: 한 위치 아래로 이동
  - 맨 아래: 목록의 맨 아래
- 단계 삭제

## 실행 플로우 제어

단계 내에는 프로젝트 내 실행 플로우를 제어하기 위해 사용할 수 있는 여러 가지 기능이 있습니다.



- **인라인:** 프로젝트 또는 라이브러리를 지정하려면 단계의 인라인 특성을 사용하십시오. 프로젝트 또는 라이브러리의 단계는 이 단계의 명령 직후에 인라인으로 실행됩니다. 인라인된 프로젝트 또는 라이브러리에 대한 단계는 단계 로그에서 들여쓰입니다.
- **패스 및 실패 체인:** 단계는 프로젝트에 지정된 체인과 다른 고유한 패스 체인 및 실패 체인을 가질 수 있습니다.
- **스레딩:** 스레드를 위해 표시된 단계를 병렬로 실행할 수 있습니다. 스레드 단계를 표시하려면 단계의 스레드 특성을 사용하십시오.
- **브로드캐스팅:** 여러 서버에서의 브로드캐스트용으로 표시된 단계를 실행할 수 있습니다. 단계의 브로드캐스트 특성을 사용하십시오.
- **조건부:** 조건이 참인 경우에만 실행할 단계를 설정할 수 있습니다. 조건이 참인 경우 실행할 대체 명령 세트와 인라인 프로젝트 또는 라이브러리를 설정할 수 있습니다. 이 기능을 사용하려면 단계 유형 특성을 조건부로 설정하고 관련 조건 및 Else 특성을 사용하십시오.
- **While 루프:** 조건이 참으로 평가될 때마다 루프에서 단계를 실행할 수 있습니다. 이 기능을 사용하려면 단계 유형 특성을 While 루프로 설정하고 관련 특성을 사용하십시오.
- **점 명령:** `.run` 및 `.runwait` 명령은 단계에 대한 명령에서 라이브러리 또는 프로젝트를 실행합니다.

복잡한 실행 플로우의 일반 용도는 필요한 경우에만 단계를 실행하는 작업 최적화입니다.

소프트웨어 빌드 엔지니어링 환경에서 작업 최적화는 전체 애플리케이션이 아닌 필요에 따라 애플리케이션의 일부만 빌드하는 것을 의미할 수 있습니다. 작업은 또한 마지막으로 컴파일된 2진에 대한 소스 상태를 확인할 수 있으며 소스 변경사항이 있는 경우에만 컴파일을 실행할 수 있습니다. 복잡한 애플리케이션의 경우 실행 플로우는 소스 상태와 모듈 종속성에 대해 응답할 수 있습니다.

## 단계가 실행하는 방법

단계 정의에서 명령 특성에 운영 체제 명령, 점 명령 또는 둘의 조합이 들어 있습니다.

개별 단계에서 둘 이상의 명령을 실행할 수 있습니다. 명령을 별도의 행에 배치하여 개별 명령을 분리하십시오.

**참고:** 단계의 결과 특성에 기본 종료 코드 설정을 사용하는 경우, 전체 단계의 성공 또는 실패는 단계의 마지막 명령으로 리턴된 종료 코드에 의해 판별됩니다. 어느 명령이든 실패를 발견하려면 로그 필터를 작성하고 결과 특성에서 해당 필터의 사용을 지정하십시오.

시스템이 단계를 실행하기 전에 단계 환경을 생성합니다. 서버 환경, 프로젝트 환경 및 단계 환경에서 지정되는 값을 사용하여 변수가 설정됩니다. 321 페이지의 『환경 상속』



의 내용을 참조하십시오. 기본적으로 변수가 구문 분석된 후 명령에서 사용할 수 있게 됩니다. 324 페이지의 『단계에서 변수 해석』의 내용을 참조하십시오.

## 셸 지정

#! 지시문을 사용하여 명령을 실행하는 데 사용할 셸을 지정할 수 있습니다. 이는 Linux<sup>®</sup> 및 UNIX<sup>®</sup> 시스템은 물론 Windows<sup>®</sup>에서도 작동합니다(Windows 에이전트는 지정된 해석기로 명령 전달을 처리함). Windows에서 단계의 명령을 C:\perl\bin의 Perl 사본으로 전송하려면 #!C:\perl\bin\perl.exe를 사용하십시오. Windows 에이전트를 Cygwin과 함께 사용하지만 Windows 셸 cmd.exe로 명령을 지정해야 하는 경우, Windows 내재 경로를 이용하는 다음 행을 사용할 수 있습니다.

```
#!cmd.exe /C
```

cmd.exe에는 /C 옵션이 필요하며, 그렇지 않으면 단계 명령이 전달된 후에 추가 명령을 기다림에 유의하십시오. UNIX 또는 Linux 컴퓨터에서는 #!/bin/perl 명령을 사용할 수 있습니다.

**참고:** Linux 또는 UNIX 시스템에서 #! 명령을 사용할 때 시스템은 필수 구문을 예측할 수 없기 때문에 표준 기본 디렉토리(서버 경로, 프로젝트, 이름 및 단계 경로 필드를 조합하여 생성된 경로)로 변경하지 않습니다. 따라서 자체 디렉토리 변경 명령을 포함시켜야 합니다. 이를 수행하려면 시스템이 작성한 특수 환경 변수(예: BF\_SERVER\_ROOT 및 BF\_PROJECTNAME\_PHYS)를 사용하십시오.

## 시스템이 단계를 파트로 분할하는 방법

Rational<sup>®</sup> Build Forge<sup>®</sup>는 단계를 파트로 분할하며, 각 파트는 운영 체제 명령 또는 단일 점 명령의 세트로 형성됩니다. 예를 들어 다음 단계는 6개의 파트를 갖습니다.

```
cmd1          # Part 1
cmd2
.dot_cmd1     # Part 2
cmd3          # Part 3
cmd4
.dot_cmd2     # Part 4
.sleep 30     # Part 5
.dot_cmd3     # Part 6
```

단계 파트는 순차적으로 실행됩니다. 각 파트의 환경이 다음 파트로 패스됩니다. 임의의 파트에서 오류가 발생할 때 처리가 즉시 중지됩니다.

이전 릴리스에서는 위의 단계의 각 파트가 단일 단계여야 했습니다.

**참고:** 파트 간의 참조를 작성하지 마십시오. 한 파트에서 GOTO를 사용하고 다른 파트에서 해당 대상 레이블을 사용하는 것과 같은 파트 사이의 점프를 작성하지 마십시오.

## 인라인: 프로젝트 또는 라이브러리의 단계 포함

지정된 프로젝트 또는 라이브러리의 모든 단계를 포함하려면 단계의 인라인 특성을 사용하십시오.

단계에서 인라인을 사용하려면 인라인 특성을 프로젝트 또는 라이브러리의 이름으로 설정하십시오. 단계가 실행될 때 다음이 발생합니다.

1. 단계가 명령 특성의 명령(들)을 실행합니다.
2. 단계가 인라인 특성에 지정된 프로젝트 또는 라이브러리의 단계를 실행합니다.

인라인된 단계를 실행하려고 하지만 명령 특성을 사용하지 않는 경우 명령 특성에서 `.sleep 0`을 사용하십시오.

## 호출 단계에서의 인라인 상속

호출된 프로젝트 또는 라이브러리의 모든 단계는 호출하는 단계의 컨텍스트에서 실행됩니다. 인라인된 단계는 호출하는 단계의 환경을 상속합니다.

그러나 시스템은 인라인 프로젝트의 선택기를 인라인 단계에 대한 기본 선택기로 사용합니다.

## 호출 단계의 상태에 대한 인라인 단계 상태의 영향

호출하는 단계의 명령이 패스하는 경우 인라인 단계의 실행 상태는 다음과 같이 간주됩니다.

- 단계에 대한 명령이 실패하는 경우 작업은 정상적으로 중지됩니다. 그러나 단계에 대한 실패 시 특성이 계속으로 설정되고 인라인이 지정되는 경우 인라인이 실행됩니다.
- 모든 단계가 실행 상태 패스를 리턴하는 경우, 호출하는 단계는 패스로 표시됩니다. 패스 체인이 지정된 호출 단계는 실행됩니다.
- 실행 상태 실패를 리턴하는 인라인된 단계가 있는 경우, 호출하는 단계는 실패로 표시됩니다. 실패 체인이 지정된 호출 단계는 실행됩니다.

이 동작으로 깊게 중첩된 인라인 및 체인을 통해서 실행 상태를 쉽게 추적할 수 있습니다.

## 인라인 중첩

프로젝트 또는 라이브러리의 단계를 인라인할 때 호출된 단계가 호출 단계에 중첩됩니다.

최대 중첩 레벨은 시스템 설정 최대 인라인 깊이에 의해 설정됩니다. 기본값은 32입니다. 중첩은 작업 시작 시간에 테스트되지 않습니다. 실행 중인 작업이 한계를 초과하는 경우 한계가 초과된 지점에서 실패합니다.

중첩 레벨은 관리 콘솔을 실행하는 호스트에서 사용 가능한 메모리에 따라서 한계가 발생할 수도 있습니다.

## 단계의 패스 체인 및 실패 체인

개별 단계는 패스 체인과 실패 체인을 가질 수 있습니다.

단계의 패스 체인과 실패 체인은 프로젝트의 패스 체인 및 실패 체인과 독립적으로 실행됩니다.

패스 체인 및 실패 체인 특성은 프로젝트 또는 라이브러리의 이름으로 설정됩니다. 프로젝트에 대한 체인 세트와 동일한 방식으로 작업합니다. 359 페이지의 『대기 사용 시 체인 프로젝트 취소』의 내용을 참조하십시오.

## 스레딩: 병렬로 단계 실행

스레딩을 사용하면 같은 서버나 다른 서버에서 단계를 병렬로 실행할 수 있습니다. 스레딩은 단계의 스레드 특성 설정으로 제어합니다. 기본적으로, 스레드 특성은 아니오로 설정됩니다. 스레딩은 서로 독립하여 실행될 수 있는 프로젝트의 부분이 있는 경우 프로젝트 실행 시간을 단축하는 데 도움이 됩니다.

여러 인접 단계의 스레드 특성이 예로 설정된 경우, 시스템은 단계를 병렬로 실행하려고 시도합니다. 이러한 단계는 스레드 사용으로 간주되며, 나머지 작업이 계속되는 동안 각 단계를 별도로 실행할 수 있습니다. 스레딩은 다음 규칙을 준수합니다.

- 스레딩이 발생하려면 차례로 최소 두 단계의 스레드 특성이 예로 설정되어야 합니다. 차례로 스레드된 단계 세트는 스레드 블록이라고 부릅니다. 스레드 블록은 인라인에 속하는 단계로 계속할 수 있습니다. 예를 들어, 프로젝트의 한 단계에 인라인이 있고 인라인의 첫 번째 단계도 스레드되는 경우, 두 단계는 동일한 스레드 블록에 속합니다. 이들은 동시에 실행됩니다. 스레드 블록은 결합 단계 또는 스레드되지 않은 단계가 발생할 때까지 중첩 인라인 단계를 포함하여 스레드된 단계를 따릅니다. 중첩 인라인 단계를 사용하는 경우 경쟁 조건이 발생하지 않도록 주의하십시오. 경쟁 조건은 스레드된 상위 단계의 결과 또는 데이터에 따라 인라인된 스레드 단계에서 발생할 수 있습니다.
- 스레드 블록은 스레드 특성이 결합으로 설정된 단계에 의해 또는 스레드되지 않은 단계가 발생할 때 종료됩니다. 해당 위치에서 단계가 순차적으로 실행됩니다.
- 시스템은 스레드 사용 단계를 발견하면 단계를 시작하려고 시도합니다. 다음 단계도 스레드된 경우, 시스템은 해당 단계를 시작하려고 시도하고 다음 단계를 계속하며, 스레드 가능 단계가 더 이상 없을 때까지 또는 작업 한계에 도달할 때까지 반복합니다. 프로젝트의 선택기가 서버 풀을 지정하는 경우, 작업 한계는 개념적으로 풀에서 서버의 작업 한계의 합계입니다.

**참고:** 스프레드된 단계의 시작 시간은 실행하기로 되어 있는 서버의 사용가능성에 따라 다릅니다. 단계를 시작할 수 없는 경우, 시스템은 기다렸다가 다시 시도합니다. 먼저 시작되는 단계를 명시적으로 제어할 수 없습니다.

- 단계는 한 서버에서(해당 서버의 용량에 따라) 또는 여러 서버에서(선택기와 일치하는 서버 수에 따라) 동시에 실행되어 종료될 수 있습니다.
- 단일 서버에서 모든 단계를 강제 실행하려면 프로젝트의 고정 특성을 사용하십시오.
- 여러 스프레드 블록이 있는 경우, 첫 번째 스프레드 블록은 다음 스프레드 블록이 시작되기 전에 완료되어야 합니다.

다음 예제에서는 단계 2, 3 및 4를 완료해야 단계 5와 6을 시작할 수 있습니다.

프로젝트	단계의 스프레드 특성
단계 1	아니오
단계 2	예
단계 3	예
단계 4	결합
단계 5	예
단계 6	예
단계 7	아니오

- 프로젝트에 최대 스프레드 특성을 사용하여 동시에 실행될 수 있는 스프레드 수를 제한하십시오. 각 스프레드 가능 단계 및 해당 인라인 프로젝트(존재하는 경우)는 병렬 프로세스가 발생할 수 있습니다. 모든 프로세스는 상위 프로젝트의 최대값에 도달할 때까지 계수됩니다. 시스템은 최대 스프레드 값에 도달하면 새 병렬 프로세스 실행을 중지합니다. 프로젝트의 병렬 프로세스 수가 값 아래로 떨어질 때까지 기다렸다가 계속합니다. 프로젝트의 병렬 프로세스 수가 최대 스프레드 값 아래로 떨어질 때까지 기다렸다가 계속합니다.

## 다중 서버에 단계 브로드캐스트

많은 서버에서 유용하게 수행할 수 있는 활동이 있는 경우, 브로드캐스트 기능을 사용하여 많은 서버에서 동일한 단계를 반복할 수 있습니다.

일반적으로 단계는 한 서버에서만 실행됩니다. 그러나 각 단계마다 브로드캐스트 선택란이 있습니다. 단계의 브로드캐스트 상자가 선택된 경우, 실행 시에 시스템은 단계의 선택기와 일치하는 각 서버에 대해 하나씩 단계를 비브로드캐스트 단계 세트로 바꿉니다.

**참고:** 단계의 선택기가 오직 하나의 서버와 일치하는 경우, 단계는 한 번만 실행됩니다.

잠재적으로 브로드캐스트를 사용하는 경우는 다음과 같습니다.

- 서버 그룹 재시동
- 서버 그룹에 대한 테스트 실행

- 동일한 파일 세트를 전체 서버 그룹에 복사
- 동일한 소스 코드 세트를 여러 서버에 체크아웃하고, 관리가 용이한 단일 단계를 사용하여 이후 개별 태스크를 수행하도록 준비

## 브로드캐스트 다시 시작 단계

458 페이지의 『브로드캐스트 다시 시작 단계』의 내용을 참조하십시오.

## 브로드캐스트 단계에서의 스레딩

실행시에 브로드캐스트 단계에 대한 대체 단계를 작성하는 경우, 시스템은 다음과 같이 단계를 스레드합니다.

- 브로드캐스트 단계의 스레드 특성이 **아니오**로 설정된 경우, 대체 단계는 동일한 스레드 값을 가져오므로 모두 연속해서 실행됩니다. 각 단계를 완료해야 다음 단계를 시작할 수 있습니다.
- 브로드캐스트 단계의 스레드 특성이 **예**로 설정된 경우에도 대체 단계는 동일한 스레드 값을 가져옵니다. 따라서 단계 세트가 서로 병렬로 실행되고, 브로드캐스트 단계에 선행하거나 후행하는 임의의 스레드된 단계와 병렬로 실행됩니다.
- 브로드캐스트 단계의 스레드 특성이 **결합**으로 설정된 경우, 시스템은 스레드를 **예**로 설정하여 대체 단계를 작성합니다(결합으로 표시되는 마지막 단계는 제외). 그 결과 단계 세트가 서로 병렬로 실행되고, 단계에 선행하는 스레드된 단계와 병렬로 실행되지만 전체 세트를 완료해야 브로드캐스트 단계 다음에 오는 단계를 시작할 수 있습니다.

## 브로드캐스트 단계에서 기타 프로젝트 실행

인라인 프로젝트를 포함하거나 단계 패스 또는 실패 시 프로젝트를 체인하는(패스 체인/실패 체인) 단계를 브로드캐스트할 수 있습니다. 다른 프로젝트를 실행(체인)하는 단계를 브로드캐스트할 때 브로드캐스트 단계는 실행된 프로젝트의 선택기를 대체하지 않음을 참고하십시오. 일반적으로, 브로드캐스트 단계의 선택기와 일치하는 모든 서버에서 프로젝트를 실행하려면 브로드캐스트 단계에서 프로젝트를 실행할 때 라이브러리(자체 선택기가 없는 프로젝트)를 사용하십시오.

라이브러리를 사용하지 않는 경우, 각 브로드캐스트 단계 사본은 서로 다른 서버에서 실행되지만 인라인 또는 체인 형식으로 연결된 프로젝트는 자체 선택기를 따릅니다(브로드캐스트 단계 사본과 동일한 서버를 선택하지 않을 수 있음). 다른 서버에서 실행 중인 각 브로드캐스트 단계로 종료할 수 있습니다. 반면에, 인라인 프로젝트의 모든 단계는 동일한 서버에서 여러 번 실행됩니다.

**참고:** 브로드캐스트를 사용하여 선택기와 일치하는 각 서버에서 한 번 라이브러리를 실행하려는 경우, 기본 프로젝트 서버를 사용하는 모든 단계가 동일한 서버에서 실행되도록 라이브러리에서 고정 옵션도 설정하십시오.

## 조건부 단계 실행

조건부 실행은 단계에 대한 if-then-else 분기를 구현합니다.

단순 If-Then 실행:

1. 단계 유형을 조건부로 설정하십시오.
2. 조건을 평가할 수 있는 표현식으로 설정하십시오.
3. 실행될 명령을 채우십시오.
4. 원하는 경우 실행할 인라인을 지정하십시오. (인라인이 설정되는 경우 명령을 공백으로 둘 수 있습니다.)

작업이 실행될 때 조건이 true로 평가되면 단계가 실행됩니다. 지정된 경우 인라인 프로젝트 또는 라이브러리 또한 실행됩니다. 표현식이 거짓으로 평가되면 해당 표현식을 건너뛰고 작업 실행이 다음 단계로 진행됩니다.

If-Then-Else 실행:

조건이 거짓을 리턴할 때 다른 명령 및/또는 인라인을 실행하려면 다음과 같이 추가 특성을 채우십시오.

- 실행할 Else 명령을 채우십시오.
- 원하는 경우 실행할 Else 인라인 프로젝트 또는 라이브러리를 지정하십시오 (Else 인라인이 설정된 경우 Else 명령은 공백일 수 있습니다.)

작업 실행 중에 조건이 성공적으로 평가되고 Command 또는 Else 명령의 명령이 성공적으로 실행되면 단계 결과가 패스로 표시됩니다. 해당 경로를 판별하려면 로그를 확인해야 합니다.

407 페이지의 『조건 함수』의 내용을 참조하십시오.

## While 루프 실행

While 루프 실행을 사용하면 조건을 기반으로 단계를 반복할 수 있습니다.

단계를 While 루프로 구현하려면 다음을 수행하십시오.

1. 단계 유형을 While 루프로 설정하십시오.
2. 평가할 수 있는 표현식 또는 명령에 조건을 설정하십시오.
3. 실행될 명령을 채우십시오.
4. 원하는 경우 실행할 인라인을 지정하십시오. (인라인이 설정되는 경우 명령을 공백으로 둘 수 있습니다.)
5. 원하는 경우 최대 반복 횟수를 단계를 실행할 최대 횟수로 설정하십시오. 사용자가 지정하는 조건이 올바르게 적용됨을 확인할 때까지 개발 중에 이 한계를 사용하지하십시오. 기본값은 100입니다.

6. 최대 반복 수에 도달한 경우 단계가 실패하도록 하려면 최대값 도달 시 단계 실패를 예로 설정하십시오. 그렇지 않으면 최대 반복 수에 도달할 때 단계가 패스됩니다.

각 단계 반복은 로그에 기록됩니다. 각각의 반복 결과는 결과 기준에 따라 패스 또는 실패로 설정됩니다.

『조건 함수』의 내용을 참조하십시오.

## 조건 함수

조건 함수는 조건 단계 특성에서 사용되고, 어댑터 XML 요소의 조건 특성에서 사용됩니다.

- 조건 필드의 처음에 다음 함수가 사용되는 경우 해당 함수는 엔진에 의해 평가되며, 조건이 true로 평가되지 않으면 선택된 서버로 아무런 정보도 송신되지 않습니다. 조건이 true로 평가되는 경우 선택된 서버에서 단계가 실행됩니다.

### 중요사항:

- 서버 자원의 쉘 환경에서 설정된 변수에 대해 함수를 사용하려고 시도하지 마십시오. 평가는 Build Forge 엔진에서 발생하므로, 단계에서 Build Forge 환경에 정의된 변수에 대해서만 작동합니다.
- 평가되는 문자열에는 연산자 문자를 사용하지 마십시오. 연산자 문자는 =, <, >, !입니다.
- 어댑터 템플릿의 경우: 조건 속성이 있는 어댑터 XML 요소에 다음 함수를 사용할 수 있습니다. 어댑터가 실행되는 방법을 지정하는 데 사용됩니다. 중요: 조건 함수는 조건 속성에서 condition="condition\_function"과 같이 큰따옴표 안에 넣어야 합니다.

다음 함수가 사용 가능합니다.

### true(표현식)

표현식이 true이면 true를 리턴합니다.

### false(표현식)

표현식이 false이면 true를 리턴합니다.

### contains(a,b)

문자열 *a*에 문자열 *b*가 들어있는 경우 true를 리턴합니다. *a* 및 *b* 매개변수는 리터럴 문자열 또는 변수일 수 있습니다. 리터럴 문자열은 따옴표로 묶지 않아야 합니다. 리터럴 문자열을 따옴표로 묶으면, 따옴표는 평가되는 문자열의 일부가 됩니다.

참고: Rational Build Forge 버전 7.1.2 이전에는 문자열 *a*가 문자열 *b*에 있는 경우에 이 함수가 true를 리턴했습니다.



**hastext(var)**

변수가 비어 있지 않은 경우에 `true`를 리턴합니다. *Var*은 Build Forge 내의 변수 세트입니다.

**isempty(var)**

변수가 비어 있는 경우 `true`를 리턴합니다. *Var*은 Build Forge 내의 변수 세트입니다.

**a eq b**

*a*가 *b*와 동일하면 `true`를 리턴합니다. *a* 및 *b* 매개변수는 Build Forge 내의 변수 세트 또는 리터럴 값일 수 있습니다. 문자 및 숫자 유형을 사용 가능합니다. 매개변수와 연산자 사이에 공백을 사용하십시오.

**a ne b**

*a*가 *b*와 같지 않은 경우 `true`를 리턴합니다. *a*와 *b* 매개변수는 Build Forge 내에 설정된 변수 또는 리터럴 값이 될 수 있습니다. 문자 및 숫자 유형을 사용 가능합니다. 문자 및 숫자 유형을 사용 가능합니다. 매개변수와 연산자 사이에 공백을 사용하십시오.

**a contains b**

*b* 문자열을 *a* 문자열에서 찾으면 `true`를 리턴합니다. 리터럴 문자열은 따옴표로 묶지 않아야 합니다. 리터럴 문자열을 따옴표로 묶으면, 따옴표는 평가되는 문자열의 일부가 됩니다. 문자 및 숫자 유형을 사용 가능합니다. 매개변수와 연산자 사이에 공백을 사용하십시오.

**함수의 표현식**

`true()` 및 `false()` 함수의 *expression* 매개변수는 다음 연산자를 사용할 수 있습니다.

**a==b** 일치를 테스트합니다. 매개변수는 문자열이나 숫자가 가능합니다. Build Forge에서 정의된 리터럴 또는 변수가 매개변수일 수 있습니다.

**a eq b**

일치를 테스트합니다. 매개변수는 문자열이나 숫자가 가능합니다. Build Forge에서 정의된 리터럴 또는 변수가 매개변수일 수 있습니다. 매개변수와 연산자 사이에 공백을 사용하십시오.

**a!=b**

비일치를 테스트합니다. 매개변수는 문자열이나 숫자가 가능합니다. Build Forge에서 정의된 리터럴 또는 변수가 매개변수일 수 있습니다.

**a ne b**

비일치를 테스트합니다. 매개변수는 문자열이나 숫자가 가능합니다. Build Forge에서 정의된 리터럴 또는 변수가 매개변수일 수 있습니다. 매개변수와 연산자 사이에 공백을 사용하십시오.



$a > b$   $a$ 가  $b$ 보다 큰 지를 테스트합니다. 매개변수는 숫자여야 합니다. Build Forge에서 정의된 리터럴 또는 변수가 매개변수일 수 있습니다. 리터럴은 산술 연산자를 사용할 수 있습니다(예:  $2+2$ ).

$a < b$   $a$ 가  $b$ 보다 크지 않은지를 테스트합니다. 매개변수는 숫자여야 합니다. Build Forge에서 정의된 리터럴 또는 변수가 매개변수일 수 있습니다. 리터럴은 산술 연산자를 사용할 수 있습니다(예:  $2+2$ ).

$a \geq b$   $a$ 가  $b$ 보다 크거나 같은지를 테스트합니다. 매개변수는 숫자여야 합니다. Build Forge에서 정의된 리터럴 또는 변수가 매개변수일 수 있습니다. 리터럴은 산술 연산자를 사용할 수 있습니다(예:  $2+2$ ).

$a \leq b$   $a$ 가  $b$ 보다 크거나 같지 않은지를 테스트합니다. 매개변수는 숫자여야 합니다. Build Forge에서 정의된 리터럴 또는 변수가 매개변수일 수 있습니다. 리터럴은 산술 연산자를 사용할 수 있습니다(예:  $2+2$ ).

**$a$  contains  $b$**

$b$  문자열을  $a$  문자열에서 찾는지를 테스트합니다. Build Forge에서 정의된 리터럴 또는 변수가 매개변수일 수 있습니다. 리터럴 문자열은 따옴표로 묶지 않아야 합니다.

## 조건 함수의 예제

아래 표의 예제에서 변수가 다음과 같이 설정됩니다.

- \$AVAL에 값 String이 들어있습니다.
- \$BVAL contains the value 3.

조건	평가	노트
문자열 contains \$AVAL	TRUE	문자열 비교
문자열 contains "문자열"	FALSE	문자열 주위의 따옴표가 비교의 일부가 됩니다.
true(문자열 contains \$AVAL)	TRUE	문자열 비교
\$AVAL contains 문자열	TRUE	문자열 비교
\$AVAL contains "문자열"	FALSE	문자열 주위의 따옴표가 비교의 일부가 됩니다.
contains(문자열,\$AVAL)	TRUE	문자열 비교
true(문자열 contains "\$AVAL")	FALSE	\$AVAL 주위의 따옴표가 비교의 일부가 되므로, "A String"은 "String" 파트 주위에 따옴표가 없습니다.
문자열 != \$AVAL	TRUE	문자열 비교
문자열 ne \$AVAL	TRUE	문자열 비교
false("Not Here" contains \$AVAL)	TRUE	문자열 비교 테스트
true( $2+1 == \$BVAL$ )	TRUE	일치에 대한 숫자 표현식

조건	평가	노트
false(2+2 < \$BVAL)	TRUE	비일치에 대한 숫자 표현식
\$AVAL eq \$AVAL	TRUE	문자열 비교 테스트
true(\$AVAL ne Linus)	TRUE	문자열 비교 테스트
true(\$BVAL > 2+2)	FALSE	(3 > 2+2)는 true가 아님
contains(Not Here, \$AVAL)	FALSE	문자열 비교 테스트

## 단계에서 프로젝트 실행

단계에서 프로젝트를 실행하려면 `.run` 또는 `.runwait` 명령을 사용하십시오.

이러한 방식으로 실행된 프로젝트는 체인처럼 작동합니다. 프로젝트는 고유한 선택기와 환경을 사용하여 실행됩니다. 440 페이지의 『`.run` 및 `.runwait`』의 내용을 참조하십시오.

## 로그 출력 사용자 정의

이 주제는 단계에서 명령 기능을 사용하여 로그 출력을 사용자 정의하는 몇 가지 방법을 보여줍니다.

### 단계의 로그 출력 레이블 지정

단계 출력이 단계 로그의 고유 카테고리에 나열되게 하려면 레이블을 작성하십시오.

#### 시작하기 전에

이 태스크에서는 사용자가 선택기, 서버 및 프로젝트를 이미 작성했다고 가정합니다. 사용자가 `echo` 명령을 허용하는 운영 체제(예: Windows, Linux 또는 UNIX)에서 서버를 사용하고 있다고 가정합니다.

#### 이 태스크 정보

출력 행 맨 처음에 대문자 레이블을 포함시킬 수 있습니다. 레이블은 단계가 종료되거나 새 레이블이 발견될 때까지 계속 사용됩니다.

레이블은 다음 구문을 갖습니다.

- 콜론이 뒤에 오는 대문자만 포함합니다. (레이블은 공백, 구두점, 숫자 또는 소문자를 포함할 수 없습니다. "SPACESHIPS:"는 올바릅니다. "Space Ships:"는 올바르지 않습니다.)
- 셋 이상의 문자를 갖습니다.

단계에서 `echo` 명령을 사용하여 레이블을 작성할 수 있습니다. 시스템은 `echo` 명령에 대한 첫 번째 인수가 레이블 구문을 따르는 경우 레이블로 인식합니다.

**참고:** 임의의 빌드 출력에 있는 행의 시작에 있는 텍스트가 `echo` 명령의 인수가 아닌 경우에도 구문을 따르는 경우 시스템은 레이블로 인식합니다.

예제에서 보는 바와 같이, 기존 출력 레이블 이름과 새 이름으로 레이블을 설정할 수 있습니다.

## 프로시저

1. 프로젝트의 단계를 새로 작성하십시오. 이 예제에서 프로젝트의 이름은 Say\_hi로 지정됩니다.
2. 단계 이름을 LabeledLogOutput으로 지정하십시오.
3. 명령 필드에 다음 텍스트를 입력하십시오.

```
echo SPACESHIPS: Voyager I
echo Voyager II
echo EXEC: You can add text to existing categories as well
```

4. 프로젝트를 실행하십시오.
5. 작업이 완료되면 로그를 보십시오.

## 결과

카테고리 헤더의 SPACESHIPS 선택란을 참고하고 SPACESHIPS로 레이블 지정된 행 354와 355를 출력하십시오.

```
79 7/9/10 10:24 AM EXEC Locale set to 'English_United States.1252'
250 7/9/10 10:24 AM EXEC Locale set to 'English_United States.1252'
349 7/9/10 10:24 AM EXEC Performing variable expansion on command line
353 7/9/10 10:24 AM EXEC start [C:\data\BuildForge\buildforge_projects\Say_hi\BUILD_10RBF-14]
354 7/9/10 10:24 AM SPACESHIPS Voyager I
355 7/9/10 10:24 AM SPACESHIPS Voyager II
356 7/9/10 10:24 AM EXEC You can add text to existing categories as well
357 7/9/10 10:24 AM EXEC end [C:\data\BuildForge\buildforge_projects\Say_hi\BUILD_10RBF-14]
```

참고: SPACESHIPS 선택란을 선택하거나 선택 취소하여 카테고리를 표시하거나 숨길 수 있습니다.

참고: EXEC 카테고리에 텍스트를 추가할 때 반드시 특수 문자(예: 소괄호)를 사용하지 마십시오. 일부 시스템에서는 따옴표로 묶는 경우 실행 가능하지만, 일반적으로는 단순히 일반 텍스트를 사용하는 것이 가장 좋습니다.

## 단계 출력을 색상 또는 활성 링크로 강조표시

로그 뷰어는 출력 텍스트의 [STATUS] 및 [URL] 명령을 인식합니다. 이 명령은 대소문자를 구분하지 않습니다. 이 명령은 echo 명령 또는 출력을 생성하는 다른 명령에서 사용할 수 있습니다. 시작 및 종료 태그는 출력과 같은 행에 나타나야 합니다.

- [STATUS=*condition*] 및 [/STATUS] 태그는 강조표시할 텍스트를 표시합니다. *condition*은 다음과 같이 강조표시 색상을 설정합니다.
  - PASS - 초록색
  - WARN - 노란색
  - FAIL - 붉은색
  - RUN - 파란색

- [URL] 및 [/URL] 태그는 텍스트를 활성 하이퍼링크로 표시합니다.

[status] 사용 예제:

```
echo [STATUS=WARN]Access to source control timed out[/STATUS]
```

[url] 사용 예제:

```
echo See the support forums at [url]http://www.ibm.com[/url]
```

로그에는 텍스트가 표시된 URL의 활성 하이퍼링크로 표시됩니다. 이를 클릭하면 새 브라우저 창 또는 탭이 열리고 페이지가 표시됩니다. 그러나 [URL] 주소의 일부와 일치하는 로그 필터로 인해 링크가 기능하지 않게 됨을 참고하십시오.

---

## 작업 데이터에 대한 작업

이 주제는 명령 기능을 사용하여 프로젝트와 작업을 수정하는 몇 가지 방법을 보여줍니다.

### 프로젝트 파일에 빌드 번호 임베드

.strsub 명령을 사용하여 한 문자열을 파일의 다른 문자열로 교체할 수 있습니다. 일반적인 사용은 표준 토큰을 시스템 변수(예: 현재 작업 번호를 제공하는 \$B 변수)로 바꾸는 것입니다.

.strsub 점 명령을 사용하여 코드 파일에 빌드 또는 버전 번호를 임베드할 수 있습니다. .strsub 명령을 프로젝트에 조기에 배치하면 나중 단계가 업데이트된 정보를 포함하는 파일을 컴파일할 수 있습니다.

예를 들어, 다음 단계는 빌드 번호를 임베드하도록 프로젝트를 설정합니다.

1. 고유 문자열(예: \_BUILD\_)을 프로젝트의 파일에 추가하십시오. 예를 들어, README.TXT 파일을 수정하고 버전 선언을 다음과 같이 변경하십시오.

```
Application version 5.0.123
Application version 5.0._BUILD_
```

2. 프로젝트의 초기 단계는 작업할 파일을 체크아웃해야 합니다. README.TXT가 체크아웃된 후에 \_BUILD\_를 \$B 시스템 변수로 바꾸는 단계를 추가하십시오. 명령에 다음을 사용하십시오.

```
.strsub _BUILD_ $B README.TXT
```

3. 프로젝트를 실행하고 README.TXT 파일에 현재 작업 번호가 포함되어 있는지 확인하십시오. 프로젝트를 세 번째 실행하는 경우, README.TXT 파일은 다음 행을 포함해야 합니다.

```
Application version 5.0.3
```

## 개선사항

다음과 같은 방법으로 이 실습을 개선할 수 있습니다.

- 추가 환경 변수를 사용하십시오. 예를 들어, `$MAJORVERSION` 및 `$MINORVERSION` 변수를 작성하고 다음과 같이 사용하십시오.

```
.strsub _MAJORVERSION_ $MAJORVERSION README.TXT
.strsub _MINORVERSION_ $MINORVERSION README.TXT
```

- 프로젝트를 시작할 때 환경 변수를 업데이트하십시오. **작업 > 시작**을 선택하여 프로젝트를 시작하면 현재 환경 변수를 보고 프로젝트를 실행하기 전에 값을 편집할 수 있습니다. 예를 들어, 작업에 주석을 변수로 포함시킬 수 있습니다. 사용자가 프로젝트를 실행할 때 새 값을 강제로 입력하도록 하려면 주석 변수에서 **변경 필수** 프로젝트 조치를 사용하십시오.

## 작업 중에 빌드 태그 변경

다음 구문이 있는 `.retag` 명령을 사용하여 작업 중에 태그의 값을 완전히 새 값으로 설정할 수 있습니다.

```
.retag <new tag value>
```

다음은 간단한 사용 예제입니다.

```
.retag MyProject
```

다음과 같이 보다 복잡하게 사용할 수도 있습니다.

```
.retag Job_${B}_${BF_D}
```

이 예제는 실행 증가 및 현재 날짜 시스템 변수를 사용하도록 태그를 설정합니다. 서버의 명령 해석기에 명령을 사용하여 결과를 설정할 수 있습니다. 점 명령에서 명령을 사용하려면 명령을 백틱(backtick) 또는 역인용 기호(`) 문자 안에 넣으십시오. .

```
.retag `hostname`
```

이 예제는 단계를 실행 중인 서버에서 호스트 이름 명령 실행 결과로 태그를 설정합니다.

**참고:** 백틱(backtick) 양식과 표준 명령 지정 양식을 혼용하지 마십시오.

## 작업 중에 환경 변수 값 변경

`.set`, `.bset` 및 `.tset` 명령을 사용하여 단계에서 환경 변수를 변경할 수 있습니다. 이 명령은 기존 환경 변수 값을 다음과 같이 변경합니다.

- `.set` 명령을 사용하여 환경의 **마스터 레코드**를 변경하십시오. 시스템은 프로젝트를 실행할 때 마스터 레코드에서 프로젝트 환경 사본을 작성하고 해당 사본을 프로젝트 기본값으로 사용합니다. 따라서 다음과 같은 결과를 초래합니다.

- .set 명령이 프로젝트 환경을 수정하는 경우, 시스템은 마스터 레코드를 다시 참조하지 않기 때문에 기본 환경을 사용하는 이후 단계는 변경사항을 확인하지 못합니다.
- .set 명령을 사용하여 환경을 수정하고 이후 단계가 명시적으로 동일한 환경을 사용하는 경우, 해당 단계는 사용자가 작성한 변경사항을 볼 수 있습니다. 단계에 선택된 특정 환경이 있으면 시스템이 환경의 마스터 레코드로 다시 돌아갑니다. 단계의 환경 설정이 『기본값』이 아니면 이름 지정된 그룹이 프로젝트 기본 그룹과 동일한 경우에도 이렇게 작동합니다.
- .set 명령으로 작성된 변경사항은 작업이 종료된 이후에도 지속됩니다. 향후 작업은 이전에 실행된 .set 명령으로 작성된 값을 사용합니다.

다음 기본 구문을 사용하십시오.

```
.set env <EnvGroupName>[(<SnapshotName>)] "<VariableName>=<DesiredValue>"
```

- .bset 명령을 사용하여 작업 실행 중에 변수값을 추가 및 변경할 수 있습니다. 변경사항은 .bset 단계가 나타난 후의 단계에 적용됩니다. 또한 나머지 작업에도 적용됩니다.

```
.bset env "<VariableName>=<DesiredValue>"
```

참고: .set 명령과 달리 .bset 명령에 지정하는 변수는 설정 시에 없어도 되므로 .bset 명령을 사용하여 작업 중에 새 변수를 작성할 수 있습니다. 변수 값은 현재 작업이 끝나면 지속되지 않습니다.

- .tset 명령을 사용하여 작업 실행 중에 변수값을 추가 및 변경할 수 있습니다. 변경사항은 현재 단계에 적용됩니다. 변경사항은 단계의 기타 모든 명령 및 단계에 대해 지정된 모든 인라인에 적용됩니다. 변수 값은 현재 단계가 끝나면 지속되지 않습니다.

```
.tset env "<VariableName>=<DesiredValue>"
```

참고: .set 명령과 달리 .tset 명령에 지정하는 변수는 설정 시에 없어도 되므로, .tset 명령을 사용하여 작업 중에 새 변수를 작성할 수 있습니다.

## 여러 변수 설정

다음 예제에서와 같이 공백으로 구분된 추가 변수 및 값 쌍을 포함시켜 다음 명령으로 둘 이상의 변수를 동시에 설정할 수 있습니다.

```
.set env MyGroup "X=5" "X2=45"
```

```
.bset env "Y=7" "CompilerVersion=4.511"
```

```
.tset env "Z=9" "Z2=54"
```

## 명령 출력을 사용하여 값 설정

명령을 서버의 명령 해석기로 보내 `.set` 또는 `.bset` 명령의 변수 값을 생성할 수 있습니다. 점 명령에서 명령을 사용하려면 명령을 백틱(backtick) 문자 안에 넣으십시오. 예를 들어, 명령은 다음과 같습니다.

```
.set env SetupGroup "PerlVer=`perl --version`"
```

명령은 `PerlVer` 변수를 `perl --version` 명령의 출력으로 설정합니다.

변수는 256자만 저장할 수 있습니다. 변수에 더 많은 문자가 지정된 경우, 값이 잘립니다.

기본적으로 시스템은 백틱(backtick) 안에 있는 명령의 전체 출력을 변수에 지정하지만 사용자는 대괄호 안의 범위 명령을 사용하여 변수에 지정할 명령 출력의 행을 선택할 수 있습니다. 범위 숫자는 0 색인(첫 번째 행 번호는 0으로 지정되고 두 번째는 1로 지정되며, 이런 식으로 계속됨)을 사용하여 출력의 행을 지정합니다. 다음 예제에서

```
.set env SetupGroup "WindowsIPinfo[0,5-8]=`ipconfig`"
```

`WindowsIPinfo` 변수는 `ipconfig` 명령 출력의 첫 번째 행과 여섯 번째에서 아홉 번째까지의 행을 수신합니다.

다음은 단일 행, 행 그룹 또는 조합을 선택하는 모두 올바른 범위 수정자입니다.

[5]

[4-6]

[1,2,5,8-11]

시스템은 분리 없이 행을 결합합니다. 공백 또는 캐리지 리턴은 추가되지 않습니다.

참고: 백틱(backtick) 양식과 표준 명령 지정 양식을 혼용하지 마십시오.

---

## 레지스터 사용

레지스터는 단계가 지속적 데이터를 저장하는 데 사용할 수 있는 범용 버퍼입니다. 일반 레지스터는 1자 이름을 갖거나 문자로 시작하는 다문자 이름을 가질 수 있습니다.

기초 데이터베이스의 대소문자 구분이 레지스터 이름을 작성할 때 사용할 수 있는 문자 크기를 판별합니다.

알림 템플릿에 레지스터 변수를 포함시킬 수 있습니다. 알림 템플릿에서 레지스터를 참조할 때 `${X}` 중괄호 양식을 사용하십시오. 빈 레지스터를 참조하면 빈 문자열이 리턴됩니다.

.push 및 .pop 점 명령을 사용하여 레지스터에 정보를 저장하고 레지스터에서 정보를 검색하십시오. 현재 작업 태그를 레지스터 콘텐츠와 같게 만드는 .poptag 명령(437 페이지의 『.poptag』)도 참조하십시오.

**참고:** 변수와 유사한 명령에서 레지스터를 사용할 수 없습니다. 먼저 레지스터 값을 파일에 팝해야 사용할 수 있습니다.

표 11. 특수 레지스터

레지스터	포함
!	실패 필터 패턴과 일치한 명령 출력 행을 포함합니다. <b>참고:</b> 이 레지스터는 필터가 적용되는 단계의 범위에서만 볼 수 있습니다. Rational Build Forge가 필터를 갖는 단계를 처리한 후 해당 단계 필터에 의해 잠재적으로 설정된 레지스터의 콘텐츠가 더 이상 보이지 않습니다.
@	패스 필터 패턴과 일치한 명령 출력 행을 포함합니다. <b>참고:</b> 이 레지스터는 필터가 적용되는 단계의 범위에서만 볼 수 있습니다. Rational Build Forge가 필터를 갖는 단계를 처리한 후 해당 단계 필터에 의해 잠재적으로 설정된 레지스터의 콘텐츠가 더 이상 보이지 않습니다.
=	작업의 노트 데이터베이스를 지정합니다. 단계가 파일의 데이터를 작업에 노트로 추가할 수 있게 합니다. 이 레지스터는 다른 레지스터와 다릅니다. <ul style="list-style-type: none"> <li>이 레지스터에는 기록(푸시)만 할 수 있습니다. 이 레지스터에서 읽을 수는 없습니다.</li> <li>이 레지스터에 푸시된 데이터는 이전 데이터를 겹쳐쓰지 않고 항상 레지스터에 추가됩니다.</li> <li>시스템은 추가된 데이터에 시간소인과 사용자 ID를 제공합니다. 따라서 작업 노트에 대한 감사 추적이 보존됩니다.</li> </ul>

## 프로젝트 레지스터

프로젝트 레지스터는 일반 레지스터와 별개입니다. 프로젝트 레지스터는 빌드 전반에서 지속되고 관리 콘솔 인터페이스를 통해 작성하고 볼 수 있으며, 몇 가지 유형의 구성 정보를 저장하는 이상적인 방법입니다.



예를 들어, IBM® Rational® ClearCase® 구성 스펙을 프로젝트 레지스터로 저장한 후 단계가 .pop -p 명령을 사용하여 스펙을 추출하도록 하고 지우기 도구인 setcs 명령과 함께 사용하여 빌드를 구성할 수 있습니다. 따라서 프로젝트와 함께 구성을 관리할 수 있습니다.

ALPHA라는 프로젝트 레지스터가 있는 경우, 콘텐츠가 완전히 다른 ALPHA라는 일반 레지스터도 가질 수 있습니다. 프로젝트 레지스터는 별도의 값 세트입니다.

다음 두 가지 방법으로 프로젝트 레지스터를 작성하고 액세스할 수 있습니다.


- -p 옵션을 추가하여 점 명령(.push 및 .pop)을 통해 -p 옵션을 사용할 때 명령은 일반 레지스터가 아니라 프로젝트 레지스터를 참조합니다.

예를 들어,

```
.push -p ALPHA register.txt
```


명령은 ALPHA라는 프로젝트 레지스터에 register.txt 파일의 콘텐츠를 위치시킵니다.

- 관리 콘솔 인터페이스를 통해. 프로젝트를 선택한 후, 원하는 프로젝트의 이름 옆에

있는 편집 아이콘()을 클릭하십시오. 프로젝트 특성이 패널의 하단부에 나타납니다. 프로젝트의 레지스터를 표시하려면 레지스터 탭을 클릭하십시오. 탭은 레지스터 관리 패널을 제공합니다.

– 새 레지스터를 작성하려면 이름과 콘텐츠를 입력한 후 작성을 클릭하십시오.

– 레지스터를 삭제하려면 패널의 오른쪽에 있는 목록에서 레지스터 이름 옆에 있는 휴지통 아이콘을 클릭하십시오.

– 레지스터를 편집하려면 목록에서 레지스터 이름 옆에 있는 편집 아이콘()을 클릭하십시오. 시스템이 레지스터 패널을 레지스터의 콘텐츠로 채웁니다. 변경사항을 작성한 후 편집된 레지스터 저장 단추를 클릭하십시오.

프로젝트에 대한 액세스 권한을 가진 모든 사용자가 프로젝트 레지스터를 보고 편집할 수 있습니다.

참고: 기초 데이터베이스의 대소문자 구분이 레지스터 이름을 작성할 때 사용할 수 있는 문자 크기를 판별합니다.

---

## 단계에서 서버 자원과의 파일 복사

점 명령을 사용하여 특정 서버 자원에서 다른 서버 자원으로 파일을 복사할 수 있습니다. 이 주제는 .get 및 .put 명령(단일 파일의 경우)과 .rget 및 .rput 명령(전체 디렉토리 트리 복사)을 사용하는 방법에 대해 설명합니다.

**중요사항:** 서버 자원에서는 파일 복사를 사용해야 합니다. 기본적으로는 사용되지 않습니다. 『서버 자원에서 파일 복사 사용』의 내용을 참조하십시오.

## 서버 자원에서 파일 복사 사용

서버의 기본 설정은 점 명령을 사용한 파일 복사를 허용하지 않습니다. 프로젝트에서 파일을 서버 자원에 복사하거나 서버 자원을 파일에 복사할 수 있도록 하려면 서버의 파일 특성을 변경하십시오.

설정을 변경하려면 다음을 수행하십시오.

1. 서버 > <ServerName>을 선택하십시오.
2. 파일 특성에 없음 이외의 값을 선택하십시오. 서버에서 서버로의 파일 복사를 사용할 수 있습니다.

## 서버에서 파일 가져오기

서버에서 파일 사본을 가져와 현재 단계의 작업 디렉토리에 해당하는 대상에 두려면 `.get` 명령을 사용하십시오. 예를 들어, 구성 디렉토리에 `config.txt` 파일이 있는 `winbuildserver1`이라는 서버가 있는 경우, 프로젝트에 다음 단계를 추가하여 파일을 현재 서버의 구성 디렉토리에 복사하십시오.

```
.get winbuildserver1:./config/config.txt ./config/config.txt
```

자세한 정보는 다음을 참조하십시오.

- `.get`에 대한 참조 항목
- 439 페이지의 『`.rget`』에 대한 참조 항목
- 468 페이지의 『작업용 작업 디렉토리』에 있는 단계의 경로가 해석되는 방법에 대한 설명

## 서버에 파일 넣기

현재 서버에서 다른 서버로 파일을 복사하려면 `.put` 명령을 사용하십시오. 다음 예제 단계는 현재 서버의 구성 디렉토리에 현재 경로에서 액세스할 수 있는 `config.txt` 파일이 있다고 가정합니다.

```
.put ./config/config.txt winbuildserver1:./config/config.txt
```

자세한 정보는 다음을 참조하십시오.

- `.put`에 대한 참조 항목
- `.rput`에 대한 참조 항목
- 468 페이지의 『작업용 작업 디렉토리』에 있는 단계의 경로가 해석되는 방법에 대한 설명

## 직접 파일 전송 구성

\_XSTREAM\_PROTOCOL 트리거 변수를 사용하면 작업 환경에 표시된 에이전트 간에 직접 파일 전송이 가능합니다. 이 경우 보내는 에이전트, 받는 에이전트 및 엔진이 모두 직접 파일 전송이 가능해야 합니다. 매개변수의 수는 인코딩, 네트워크 및 파일 전송 특성을 제어합니다.

196 페이지의 『bfagent.conf 참조』 참조에서 337 페이지의 『트리거 변수 참조』 및 xstream\_\* 매개변수를 참조하십시오.

---

## 문제점 해결 단계 처리

단계 처리 관련 문제가 발생하는 경우 이 주제의 정보를 검토하여 허용 가능한 해결 방법 또는 솔루션이 있는지 여부를 확인합니다.

### 작업이 ANT 빌드 명령 다음의 단계 명령을 처리하지 않음

문제점 설명:

ANT **build** 명령 뒤에 있는 단계의 명령은 처리되지 않습니다.

다음 단계 예제에서 **echo** 명령이 실행되지 않습니다.

```
<path to ant bin directory> ant -f  
<path to Java project>\build.xml build  
echo "Ant build complete"
```

설명: ANT 빌드는 ANT 빌드 실패 또는 성공 여부에 관계 없이 오류 코드 1을 리턴합니다.

단계의 명령 특성에서 여러 명령이 사용되는 경우 마지막으로 실행된 명령의 종료 상태만 단계 결과 상태에 영향을 줍니다. 서버가 ANT build 명령이 포함된 단계에 대한 명령 스크립트를 실행하는 경우 오류 상태가 1이면 ANT build 다음의 모든 명령이 실패합니다.

솔루션:

ANT 빌드에서 생성된 단계 출력을 처리하는 단계 로그 필터를 작성하십시오. 단계 로그 필터는 단계 결과를 설정하고 작업의 다음 단계가 처리되는지 확인합니다.

1. ANT 빌드는 단계에서 유일한 명령이거나 단계에서 마지막 명령이어야 합니다.

로그 필터가 없는 경우 ANT 빌드는 오류 코드 1을 리턴하며 단계 결과는 실패로 설정됩니다.

2. 단계 출력에서 적합한 실패 텍스트 문자열(BUILD FAILED)을 검색하여 단계 처리를 효과적으로 제어하는 로그 필터를 작성합니다.

텍스트 문자열이 발견되면 실패 설정 조치를 사용하여 단계 결과를 실패로 설정하십시오. 단계 로그 필터를 사용할 때 텍스트 문자열이 없는 경우 단계 결과가 항상 패스로 설정됩니다.

단계의 로그 필터 설정에 대한 세부사항은 364 페이지의 『로그 필터』의 내용을 참조하십시오.

## Windows 일괄처리 명령 다음의 단계 내 명령이 실행되지 않음

### 문제점 설명:

Windows 일괄처리 명령 다음의 단계 명령이 처리되지 않습니다.

다음 단계 예제에서는 첫 번째 일괄처리 파일 다음의 두 명령이 실행되지 않습니다.

```
C:\script1.bat
C:\script2.bat
echo "Performed both batch commands"
```

**설명:** 단계의 모든 명령이 서버에서 실행될 Windows 일괄처리 파일에 놓입니다. 단계 명령이 일괄처리 파일에 대한 참조를 포함하는 경우, 일괄처리 파일이 실행된 후 단계가 종료됩니다. 일괄처리 파일 참조 다음의 단계 명령이 실행되지 않습니다.

### 솔루션:

호출 명령을 사용하여 단계 내에서 일괄처리 파일을 실행하십시오. 호출 명령은 단계 일괄처리 파일 내에서 실행됩니다.

```
call C:\script1.bat
call C:\script2.bat
echo "Performed both batch commands"
```

---

## 점 명령 참조

시스템은 단계의 명령 필드에 있는 점 명령을 사용할 수 없습니다. 점 명령은 시스템에서 특수 기능과 함수에 대한 액세스를 제공합니다.

단계에서 점(.) 명령을 일반 명령과 혼합할 수 있으며 단일 단계에 여러 점(.) 명령이 있을 수 있습니다. 그러나, 단일 단계에 둘 이상의 .scan 명령을 사용하지 마십시오. 그럴 경우, 시스템이 명령의 결과를 정확히 보고할 수 없기 때문입니다.

환경 변수와 함께 사용할 점 명령의 개별 목록이 있습니다. 334 페이지의 『변수에서 점 명령 사용』의 내용을 참조하십시오.

## 점 명령 구문

구문 스펙: 각 점 명령 설명은 다음 표기법을 사용하여 구문 스펙을 포함합니다.

- 사용자 제공 값은 꺾쇠 괄호(<value>)로 표시됩니다.

- 선택적 텍스트는 대괄호로 표시됩니다(예: [선택적 텍스트]).

특별히 명시된 경우를 제외하고 명령 매개변수에 환경 변수를 사용할 수 있습니다.

점 명령이 매개변수로 환경 변수를 허용하는 경우 다음도 사용할 수 있습니다.

- 명령 실행에 대해 백틱(`)
- 범위를 나타내는 대괄호([ 및 ])(범위 표기법이라고도 함)

다음 .set 예제를 고려하십시오.

```
.set envgroupname "FOO[1,3-5]=`cat foofile`"
```

이 경우 에이전트는 cat foofile을 실행하며, 여기서는 foofile의 콘텐츠를 나열합니다. 범위 표기법을 사용하여 행 1 및 3 - 5가 추출됩니다. 그런 다음 .set 명령은 이러한 행을 사용하여 환경 그룹 envgroupname에서 기존 변수 'FOO'를 업데이트합니다.

## **.bom**

```
.bom addcategory "category"
.bom setcolumn "category" "section" "column" [...]
.bom data category "section" "column=value" [...]
```

.bom 명령은 빌드 명세서(BOM)에 데이터를 추가합니다. 이 명령을 사용하여 카테고리, 섹션 및 데이터를 추가할 수 있습니다.

**범주** 카테고리는 BOM에 인쇄된 헤더입니다. addcategory 옵션을 사용하여 지정하십시오.

```
.bom addcategory "category"
```

**섹션** 섹션에서는 카테고리 내의 데이터 열을 정의합니다. setcolumn 옵션을 사용하여 섹션 및 섹션 내의 열을 지정하십시오. 섹션 이름은 인쇄되지 않습니다. 섹션의 열 헤더 세트는 섹션의 처음에 인쇄됩니다. **-p** 옵션을 사용하고 상위 섹션을 식별하여 섹션을 중첩할 수 있습니다.

섹션 및 열은 data 옵션이 해당 섹션 및 열에 데이터를 추가하려고 시도하기 전에 정의되어야 합니다.

```
.bom setcolumn "category" "section" "columnheader" [...]
.bom setcolumn "category" "section"
-p parentsection"columnheader" [...]
```

**데이터** 섹션에 정의된 열에 데이터가 채워집니다. 섹션 및 열은 setcolumn을 사용하여 이미 정의되어 있어야 합니다.

```
.bom data category "section" "column=value" [...]
```

다음 예제는 카테고리, 섹션, 열 및 데이터를 지정해야 하는 순서를 나타냅니다.

```
.bom addcategory "Spaceships"
.bom setcolumn "Spaceships" "Section1" "ShipName" "WarpSpeed" "Tonnage"
.bom setcolumn "Spaceships" "Subsection1" -p "Section1" "ShippingDate" "ShippingManifest"
.bom data "Spaceships" "Section1" "ShipName=SpaceShipOne" "WarpSpeed=9" "Tonnage=10000"
.bom data "Spaceships" "Subsection1" "ShippingDate=123" "ShippingManifest=456"
.bom data "Spaceships" "Section1" "ShipName=Freighter" "WarpSpeed=6" "Tonnage=20000"
```

이 예제는 다음과 같은 BOM을 나타냅니다.

ShipName	WarpSpeed	Tonnage	ShippingDate	ShippingManifest
SpaceShipOne	9	10000	123	456
Freighter	6	20000		

다른 점 명령의 경우와 마찬가지로 명령에서 환경 변수를 사용할 수 있습니다. 다음과 같은 명령은

```
.bom data "Spaceships" "${SECTION}" "ShipName=${NAME}" "WarpSpeed=${SPEED}"
"Tonnage=${TONNAGE}"
```

이전 명령으로 환경 변수에 로드된 데이터로 BOM을 채웁니다.

얼마든지 열을 작성할 수 있지만 시스템은 마지막 열이 채워질 때까지 BOM에 행을 쓰지 않습니다.

데이터 행에서 열을 생략하면 시스템은 다음 예제에 나타난 것처럼 이전 행의 값을 사용합니다.

```
.bom addcategory "Spaceships"
.bom setcolumn "Spaceships" "Section1" "ShipName" "WarpSpeed" "Tonnage"
.bom data "Spaceships" "Section1" "ShipName=SpaceShipTwo" "WarpSpeed=3" "Tonnage=30000"
.bom data "Spaceships" "Section1" "ShipName=Tanker" "Tonnage=50000"
```

그 결과로 SpaceShipOne의 WarpSpeed 값이 반복됩니다.

ShipName	WarpSpeed	Tonnage
SpaceShipTwo	3	30000
Tanker	3	50000

## **.bomexport**

### **설명**

.bomexport 점 명령은 작업 BOM을 XML 파일로 내보냅니다. BOM 정보를 수집한 후에 .bomexport는 사용자가 지정하는 파일과 위치에 이를 저장합니다.

경로 및 파일 이름은 선택적입니다. 기본적으로 Build Forge는 서버에 있는 단계의 작업 디렉토리에 BOM 보고서를 저장하며, 태그 이름을 파일 이름으로 사용합니다 (<build\_tagname>.xml).

프로젝트의 마지막 단계로 .bomexport 명령을 지정하십시오.

## 구문

```
.bomexport [path_name][file_name]
```

## 옵션

옵션	설명
path_name	선택적 경로 이름입니다. 제공된 경우, 경로는 Build Forge 서버에 있는 단계의 작업 디렉토리와 관련되어야 합니다. 생략된 경우, 파일은 단계의 작업 디렉토리에 저장됩니다.
file_name	선택적 파일 이름입니다. 작업 BOM은 XML 형식으로 파일에 저장됩니다. 파일 이름이 제공되지 않은 경우, 파일 이름은 빌드 태그 이름 및 문자열 _BOM에서 생성됩니다(<build_tagname>_BOM.xml).

## 예제:

```
.bomexport
.bomexport myproj.xml
.bomexport path/to/myproj.xml
.bomexport /path/to/myproj.xml
```

## .break

```
.break [<notification_group_name>]
```

.break 명령을 사용하여 작업을 다시 시작할 때까지 정지 상태가 되도록 하십시오. 시스템이 .break 명령이 있는 단계를 발견하면, 실행이 완료되어 결과적으로 중지됩니다. 다시 시작 아이콘을 사용하여 작업을 다시 시작하고 .break 단계 다음의 단계로 진행 하십시오.

체인 작업에서 .break 명령이 발생하는 경우, 시스템은 체인 작업을 중지시키지만 호출 작업에 제어를 리턴하여 단계를 계속 처리합니다.

액세스 그룹을 명령에 대한 선택적 인수로 포함시킬 수 있습니다. 이렇게 하는 경우, 시스템은 작업을 중지할 때 지정된 액세스 그룹으로 이메일 메시지를 보냅니다.

## .bset

```
.bset env "<VarName>=<Value>" [...]
.bset selector <SelectorName> [<SelectorSnapshotName>]
.bset server <ServerName>
.bset buildserver <ServerName>
```

.bset 명령은 작업 중에 프로젝트 설정을 일시적으로 변경합니다.



참고: `.bset` 명령은 해당 환경을 사용하는 이후 단계에서 기본 빌드 환경에 영향을 줍니다. 동일한 단계에서 이전의 `.bset` 명령에 의존하는 `.bset` 명령을 지정하지 마십시오. 예를 들어 동일한 단계에서 다음 조합과 유사한 단계를 지정하지 마십시오.

```
.bset env "VAR1=VALUE1"
.bset env "VAR2=$VAR1"
```

명령에는 다음 옵션이 있습니다.

- **env**는 실행하는 작업에 대해 하나 이상의 프로젝트 환경 변수 값을 변경합니다. 변경사항은 현재 단계에서 즉시 적용됩니다. 이런 식으로 명령을 사용하는 방법에 관한 자세한 정보는 413 페이지의 『작업 중에 빌드 태그 변경』을 참조하십시오. 아직 존재하지 않는 변수를 설정할 수 있습니다. `.bset` 명령으로 설정된 값은 작업 레코드에 기록됩니다. 프로젝트나 단계 환경에도 정의된 변수에 새 값을 설정하면, 새 값은 작업 중에만 유효합니다. 프로젝트나 단계의 환경은 변경되지 않습니다.
- **selector**는 작업 중에 프로젝트 선택기를 변경합니다. 새 선택기는 다음 단계에 적용됩니다(현재 단계의 인라인 단계가 포함될 수 있음). 새 선택기는 명시적 선택기 설정을 포함하지 않는 단계에만 사용됩니다(단계 선택기는 프로젝트 기본값으로 설정됨). 인라인 단계에서 선택기 옵션을 사용하는 경우 옵션은 인라인 단계에서 인라인하는 단계를 포함한 인라인 프로젝트의 모든 후속 단계에 영향을 줍니다. 호출 단계 또는 호출 단계 이상의 다른 단계에는 영향을 주지 않습니다.

명시적 선택기가 설정되어 있는 단계는 적용되지 않습니다.

선택적 <SelectorSnapshotName>을 사용하여 지정된 선택기의 스냅샷을 지정하십시오.

- **server**는 작업 중에 기본 프로젝트 서버를 변경합니다. 새 서버 설정은 다음 단계에 적용됩니다(현재 단계의 인라인 단계 이후). 단계에서는 지정된 서버만 사용하며, 명시적 선택기 설정이 포함되지 않습니다.

백틱(*Backtick*) 구문: 백틱을 사용하여 서버 이름을 명령 출력으로 설정할 수 있습니다. 예를 들어, 다음 명령은 `SelectAServer.sh` 스크립트를 실행하고 출력을 `.bset server` 명령의 서버 이름으로 제공합니다.

```
.bset server `SelectAServer.sh`
```

- **buildserver**는 작업 중에 기본 프로젝트 서버를 변경합니다. 새 서버 설정은 다음 단계에 적용됩니다(현재 단계의 인라인 단계 이후). 이는 인라인된 단계와 프로젝트 레벨 단계에 적용되므로 **server** 옵션과 다릅니다.

참고: 스레드 단계에서는 여러 개의 `.bset` 명령 사용을 권장하지 않습니다.

## **.buildstatus**

```
.buildstatus <result>
```



이 명령은 빌드 단계 결과에 관계 없이 빌드가 완료된 후 빌드가 지정된 <result>를 포함하도록 강제 실행합니다.

<result>의 올바른 값은 다음과 같습니다.

**P** 결과를 패스됨으로 설정합니다.

**F** 결과를 실패함으로 설정합니다.

**W** 결과를 경고로 설정합니다.

**B** .break 명령이 지정된 경우처럼 결과를 중지됨으로 설정합니다.

## .date

`.date <conversion_specifier>`

.date 명령에 하나 이상의 변환 지정자 문자를 인수로 사용하여 프로젝트가 실행될 때 현재 날짜-시간 정보를 생성하십시오.

.date 명령과 해당 인수는 환경에서 환경 변수로 정의되어야 합니다. 이 경우, 프로젝트 또는 단계에 환경을 지정할 수 있습니다.

예를 들어, 환경에서 환경 변수 DayOfWeek를 정의하고 이 변수의 값으로 .date %A를 지정하십시오. 프로젝트나 단계에 환경을 지정하십시오. 프로젝트가 수요일에 실행되면, 작업은 Wednesday(수요일) 텍스트를 이 환경 변수 DayOfWeek에 지정합니다.

**중요사항:** .date 명령은 단계 명령 필드에서 직접 참조할 수 없습니다.

**참고:** .date 명령은 모든 단계에서 다시 평가됩니다. 특정 시간을 보존하려면 .bset env 와 함께 .date를 사용하십시오. 예를 들어 \$ORIGTIMESTAMP = .date %d-%b-%Y.%H:%M:%S 및 .bset env "TIMESTAMP=`echo \$ORIGTIMESTAMP`"를 이 순서로 사용하십시오.

.date 명령은 POSIX strftime 함수에서 빌드되며 ANSI C89 표준으로 식별된 변환 지정자를 허용합니다. 변환 지정자의 날짜-시간 값은 다음 테이블에서 제공합니다.

**참고:** 모든 변환 지정자가 전체 로케일 및 운영 체제에서 이동할 수 있는 것은 아닙니다. .date 명령을 사용하려는 서버 운영 체제와 로케일에서 해당 명령 결과를 테스트하십시오.

날짜 변환 지정자	설명
%a	현재 로케일에 따라 축약된 요일 이름
%A	현재 로케일에 따른 전체 요일 이름
%b	현재 로케일에 따라 축약된 월 이름

날짜 변환 지정자	설명
%B	현재 로케일에 따른 전체 월 이름
%c	현재 로케일에 대한 날짜 및 시간의 선호하는 표시
%d	10진수로 된 월중 날짜(01 - 31 범위)
%H	10진수로 된 24시간 시계를 사용하는 시간(00 - 23 범위)
%I	10진수로 된 12시간 시계를 사용하는 시간(01 - 12 범위)
%j	10진수로 된 연중 날짜(001 - 366 범위)
%m	10진수로 된 월(01 - 12 범위)
%M	10진수로 된 분(00 - 59 범위)
%p	주어진 시간 값에 따른 『AM』이나 『PM』, 또는 현재 로케일에 해당하는 문자열. 경오는 『pm』으로, 자정은 『am』으로 간주됩니다.
%S	10진수로 된 초(00 - 61 범위)
%U	10진수로 된 현재 연도의 주 번호(첫 번째 일요일을 01 주의 첫 번째 날로 시작하여 00 - 53 범위).
%w	10진수로 된 요일(일요일을 0으로 하여 0 - 6 범위)
%W	10진수로 된 현재 연도의 주 번호(첫 번째 월요일을 01 주의 첫 번째 날로 시작하여 00 - 53 범위).
%y	10진수로 된 세기가 없는 연도(00 - 99 범위)
%Y	10진수로 된 세기를 포함하는 연도
%Z	시간대나 이름 또는 약어
%%	리터럴 『%』 문자

## .defect

**설명**   결함 추적 애플리케이션용 어댑터를 프로젝트 단계에 추가하려면 .defect 명령을 사용하십시오. 결함 어댑터는 Build Forge 오브젝트입니다. 결함 추적 애플리케이션용 어댑터 템플릿을 기반으로 합니다. 프로젝트가 실행될 때 단계에 대한 어댑터 코드가 실행됩니다.

## 구문

```
.defect <adaptor_name> [entry_name]
```

<adaptor\_name>은 필수입니다. 이 이름은 관리 콘솔에서 어댑터에 지정된 이름입니다. <adaptor\_name>의 대소문자는 콘솔에서 사용되는 대소문자와 일치해야 합니다.

어댑터 템플릿에 여러 인터페이스 함수가 있는 경우, *entry\_name* 옵션을 사용하여 실행할 함수를 지정하십시오. *entry\_name*은 어댑터 템플릿의 <interface> 요소에 지정된 이름 속성과 일치해야 합니다. *entry\_name*에 지정된 <interface> 요소가 없거나 찾을 수 없는 경우 대신 기본 <interface> 요소가 실행됩니다. 다음 예제에서 항목 이름은 DefectFunction입니다.

어댑터 링크를 사용하는 경우, 어댑터가 자동으로 호출되고 어댑터 템플릿의 첫 번째 인터페이스 함수가 실행됩니다. 다른 인터페이스를 실행하려면 어댑터 템플릿에서 실행하려는 인터페이스의 기본 속성을 true(default="true")로 설정하십시오.

### 예제:

```
.defect MyClearCaseQuestAdaptor
.defect MyClearCaseQuestAdaptor DefectFunction
```

**노트** 어댑터를 작성하거나 어댑터 목록을 보려면 **프로젝트 > 어댑터**를 선택하십시오. Build Forge 제품과 함께 제공되는 어댑터 템플릿은 다음 위치에 있습니다.

```
<bfinstall>/interface
```

## .drill

```
.drill [through]
<"var1,var2,var3"|${EnvVar}-r[p] Register>
[grouped by] "{}"
[separated by] ","
[exec] "Command $1 $2"
```

.drill 명령을 사용하여 명령을 반복하고 일련 값의 각 구성원에 대해 명령을 한 번 실행할 수 있습니다. 명령행에 값을 지정하거나 환경 변수 또는 레지스터에서 값을 가져올 수 있습니다. 시스템은 .drill 명령을 실행할 때 시스템은 .drill 구문을 사용하여 일련의 명령행을 생성하고 실행을 위해 에이전트로 보냅니다.

예를 들어, .drill "A,B,C,D" "echo value \$1" 명령은 다음 명령을 작성합니다.

```
echo value A
echo value B
echo value C
echo value D
```

## 그룹화

값을 그룹화하고  $n$  구문을 사용하여 각 그룹의 다중 값을 참조할 수 있습니다. \$1은 그룹의 첫 번째 값을 참조하고 \$2는 그룹의 두 번째 값을 참조합니다. 예를 들어, `.drill through "(A,B,C,D,E),(B,C,D,E,F),(C,D,E,F,G)" grouped by "("` separated by "," exec "echo 1[\$1] 2[\$2] 3[\$3] 4[\$4] 5[\$5]"는 다음 명령을 작성합니다.

```
echo 1[A] 2[B] 3[C] 4[D] 5[E]
echo 1[B] 2[C] 3[D] 4[E] 5[F]
echo 1[C] 2[D] 3[E] 4[F] 5[G]
```

**참고:** 기본 그룹화 문자는 없습니다. 기본 분리 문자인 쉼표가 있습니다. grouped by 를 지정하지 않은 경우, 시스템은 제공된 값을 분리 문자로써 분리된 것으로 인식하고 각각을 문자열 단일 값으로 간주합니다. 예를 들어, `.drill "(A,B),(C,D)" "echo $1 $2"` 명령은 다음 명령으로 해석됩니다.

```
echo (A 2
echo B) 2
echo (C 2
echo D) 2
```

## 데이터 원본

`.drill` 명령이 루프 통과 데이터를 가져오는 위치에 대한 여러 옵션이 있습니다. 명령의 첫 번째 매개변수는 데이터 원본입니다. 선택적 명령 단어인 "through"를 포함시켜 데이터 원본을 표시할 수 있습니다.

- 다음 명령에서와 같이 명령행에 one, two, three 값을 반복하는 데이터를 명시적으로 나열할 수 있습니다.

```
.drill through "one,two,three" exec "echo $1"
```

- 환경 변수에서 데이터를 가져올 수 있습니다. 다음 명령은 FILENAMES 환경 변수가 쉼표로 분리된 파일 목록이라고 가정하며 DOS 명령을 사용하여 목록의 파일을 모두 삭제합니다.

```
.drill through $FILENAMES exec "del $1"
```

- 레지스터 또는 프로젝트 레지스터에서 데이터를 가져올 수 있습니다. RegisterA에 쉼표로 분리된 파일 이름 목록이 포함된 경우, Linux 시스템에 실행된 다음 명령은 각 파일의 콘텐츠를 작성합니다.

```
.drill -r RegisterA exec "cat $1"
```

반면에 다음 예제는 동일한 작업을 수행하지만 프로젝트 레지스터를 사용합니다.

```
.drill -rp ProjectRegisterA exec "cat $1"
```

## .edit

```
.edit /<search_expression>/<replace_expression>/  
[<relative_path>/]file [file ...]
```

하나 이상의 파일에서 텍스트 문자열을 검색하고 바꾸려면 .edit 명령을 사용하십시오. .edit 명령은 지정된 각 파일의 각각의 모든 행에서 문자열의 첫 번째 인스턴스 (search\_expression)를 바꿉니다. 상대 경로를 지정하지 않는 한 파일은 단계의 작업 디렉토리에 상주한다고 가정합니다.

.edit 명령은 대체 패턴에 () 부분 문자열 선택 및 \N 대체 사용을 포함하여 일치와 바꾸기를 위해 표준 POSIX 정규 표현식을 구현합니다.

.edit 명령은 기본적으로 POSIX 확장 정규 표현식 구문을 사용합니다. 에이전트가 Perl 호환 가능 정규 표현식 지원으로 컴파일된 경우, PCRE 구문을 지정하는 "p" 문자 다음에 대체 표현식이 올 수 있습니다.

백슬래시를 이스케이프 문자로 사용하려면 에이전트 호스트의 운영 체제에 따라 다른 처리가 필요합니다.

- UNIX 및 Linux에서 표현식은 에이전트 처리에서 두 번 해석되므로 하나의 백슬래시를 표시하려면 4개의 백슬래시를 사용하십시오. 예를 들어 4개의 백슬래시와 마침표를 사용하여 리터럴 마침표 문자와 일치시키십시오.

```
\\\\.
```

- Windows에서 동일한 리터럴 마침표를 일치시키려면 2개의 백슬래시만 필요합니다.

```
\\.
```

사용자는 구분 기호를 이스케이프할 수 없습니다. 이것이 필요하다고 판단되는 경우, 올바른 접근 방법은 대체 구분 문자를 사용하는 것입니다. 예를 들어, abc 문자만 포함하는 abc.txt 파일이 있습니다. 이를 a/c 문자열로 대체하려 합니다.

- 잘못된 - 다음의 경우 문자열은 변경되지 않습니다. 왜냐하면 구분 문자는 이스케이프 문자에 우선하기 때문입니다.

```
.edit /abc/a\\c/ abc.txt  
.edit /abc/a\\/c/ abc.txt  
.edit /abc/a\\\\c/ abc.txt
```

- 올바른 - 구분 문자 변경을 통해 a/c를 대체할 수 있습니다.

```
.edit "abc"a/c" abc.txt
```

**참고:** 와일드카드를 사용하지 않고 하나 이상의 파일 이름을 명시적으로 나열해야 합니다.

예제: 다음 명령은 drivermakefile이라는 파일에서 winXPdriver 및 win2000driver 같은 문자열을 바꿉니다.

```
.edit /win.*driver/linuxdriver/ drivermakefile
```

.edit 명령은 .strsub 명령과 유사하며, 차이점은 다음과 같습니다.

- .strsub 명령은 대형 텍스트 파일 또는 다수의 파일에서 대체를 수행할 때 .edit보다 속도가 빠릅니다.
- .edit 명령은 정규 표현식 검색 및 바꾸기를 수행할 수 있습니다.
- .edit 명령은 각 파일의 각각의 모든 행에서 문자열의 첫 번째 인스턴스만 (search\_expression) 바꿉니다.
- .strsub 명령은 각 파일의 각각의 모든 행에서 문자열의 모든 인스턴스(소스)를 바꿉니다.

## .email

```
.email <recipient>
```

이 명령은 step\_email 메일 템플릿을 사용하여 *recipient*로 이메일을 보냅니다. 지정된 주소는 Build Forge 사용자인 것이 바람직하지만 .email은 모든 올바른 이메일 주소로도 보냅니다.

## .export

```
.export [path_name][file_name]
```

.export 명령은 호출 프로젝트의 프로젝트 정의를 단계의 작업 디렉토리에 있는 XML 파일에 저장합니다. XML 파일은 프로젝트 및 해당 단계에 대해 설명합니다. 또한 서버와 같은 기타 연관 오브젝트를 설명하지는 않습니다.

내보낸 XML 파일은 관리 콘솔에 프로젝트 정의를 가져오는 데 사용될 수 있습니다.

.export 명령은 선택적 경로 및/또는 파일 이름을 가질 수 있습니다. 경로는 상대 경로여야 합니다. 이는 단계의 작업 디렉토리로부터 적용됩니다.

파일 이름이 제공되지 않은 경우, 파일 이름은 현재 프로젝트 태그인 \$BF\_TAG.xml로 생성됩니다.

## .get

```
.get server:[[<relative_path>/]file/]file [[[<relative_path>/]file/]file]
```

한 논리 서버에서 다른 논리 서버로 파일을 전송하려면 .get 명령을 사용하십시오. .get 오퍼레이션은 현재 서버/경로에서 실행되며 지정된 서버/경로에서 파일을 검색합니다. 대상 경로 이름은 단계의 현재 작업 디렉토리 및 연관됩니다. 소스 경로 이름은 지정된 서버의 기본 경로와 관련이 있습니다. 서버는 파일에 대한 .get 오퍼레이션을 허용하는 논리 서버를 지정해야 합니다(418 페이지의 『서버 자원에서 파일 복사 사용』 참조). 단일 파일만 전송할 수 있습니다.

경로 스펙은 환경 변수를 포함할 수 있습니다. 이 기능을 사용하면 특정 작업에 사용된 경로와 관련된 파일을 지정할 수 있습니다. 468 페이지의 『작업용 작업 디렉토리』에서 작업에 사용된 경로에 대한 설명을 참조하십시오.

사용 중인 서버 이름에 공백이 있는 경우, 파일에 대한 경로 및 서버 이름을 따옴표로 묶으십시오. 예를 들어 다음과 같습니다. `.get [<relative_path>/]"file server:[<relative_path>/]file]"`

전송 속도가 빠르지 않으므로 다른 방법을 선택하여 대형 파일을 전송할 수 있습니다. 속도는 초당 40KB를 넘지 않습니다. 따라서 70MB의 파일을 전송하려면 45분에서 1 시간 정도 소요될 수 있습니다.

참고: 대상 파일이 이미 존재하는 경우 경고하지 않고 이를 겹쳐씹습니다.

## **.load**

```
.load [-o] [-e] [-v] [-j] [<relative_path>/]<filename>
.load -r|-p <registername>
.load -s `<command name>`
```

.load 명령은 XML 파일에서 프로젝트를 로드하고 .load 명령을 실행한 단계 다음에 로드된 프로젝트의 단계를 현재 프로젝트에 추가하며, 프로젝트가 실행 시에 단계를 동적으로 작성하고 로드할 수 있게 합니다. 옵션을 사용하면, .load 명령이 레지스터 또는 명령 출력에서 데이터를 가져오게 할 수 있습니다.

.load 명령의 XML 파일을 작성하려면 기존 프로젝트에서 내보내기 파일로 시작하여 적절한 기본 구조를 제공하십시오. 또한 시스템에 프로젝트를 작성한 후 .load 명령에서 이를 사용하도록 내보낼 수 있습니다. 이 주제에는 샘플 XML 코드가 포함되어 있습니다.

.load 명령으로 로드된 단계는 인라인 또는 체인 프로젝트에 대한 참조를 포함할 수 있습니다. 기본적으로 시스템은 XML 파일에서 인라인 프로젝트의 정의를 찾고 단계를 로드합니다. 시스템이 데이터베이스에서 인라인 프로젝트 정의를 가져오도록 하는 방법은 이 주제에서 아래의 -e 옵션을 참조하십시오. 패스 체인 또는 실패 체인 프로젝트의 경우, 시스템은 항상 데이터베이스에서 프로젝트 정의를 찾습니다.

참고: .load에서 실행된 JPO 단계의 경우, else-inline 식별 프로젝트 또는 라이브러리는 시스템 내에서 이미 존재하는 프로젝트나 라이브러리어야 합니다. 그렇지 않으면 인라인이 실행되지 않습니다.

## **XML 파일의 다중 프로젝트**

시스템이 호출 프로젝트와 함께 인라인 프로젝트를 내보내기 때문에 XML 파일은 여러 프로젝트를 포함할 수 있습니다. .load 명령은 파일에서 기본 레이블이 붙은 프로젝



트를 실행합니다. 이 프로젝트는 <project> 요소에 대한 primary="1" 속성을 갖습니다.

## 명령 옵션 및 매개변수

가장 단순한 명령 양식은 .load <filename>입니다. 파일 이름 앞에 선택적 경로 이름(작업 디렉토리와 연관)을 포함시킬 수 있습니다. 예를 들어,

```
.load ../../project.xml
```

명령은 단계의 경로 특성이 "/"(기본값)라고 가정하고 서버 디렉토리(프로젝트 및 작업 디렉토리를 포함하는 디렉토리)에서 project.xml 파일을 로드합니다.

**참고:** 정상 단계가 인라인 프로젝트를 실행할 때 시스템은 데이터베이스로 이동하여 해당 프로젝트의 현재 정의를 가져옵니다. .load 명령으로 가져온 단계가 인라인 프로젝트를 실행할 때 시스템은 XML 파일에서 인라인 프로젝트의 정의를 찾습니다. 이런 상황을 방지하기 위한 방법은 아래 -e 옵션에 대한 설명을 참조하십시오.

**참고:** 7.1 이전의 Build Forge 버전에서 생성된 단계 XML은 지원되지 않으면 해당 요소는 ID 대신 이름으로 참조되므로 실패할 수 있습니다.

**참고:** .load는 가져오기 유틸리티를 사용하며 마찬가지로, 가져오기 유틸리티의 설정에 따라 액세스 그룹 소유권을 적용합니다. 추가 정보는 509 페이지의 『가져온 오브젝트에 액세스 그룹을 지정하는 방법』을 참조하십시오.

명령에는 다음 옵션이 있습니다.

### -r 또는 -p

이 옵션을 사용하면 시스템이 레지스터에서 단계를 로드합니다. 명령행을 이런 옵션과 함께 사용하십시오.

```
.load -r|-p <registername>
```

-r 옵션이 일반 레지스터에서 단계를 로드하는 반면, -p 옵션은 프로젝트 레지스터에서 단계를 로드합니다. 프로젝트의 이전 단계에서 레지스터에 데이터를 빌드한 후 이 명령을 사용하여 레지스터에서 단계를 로드할 수 있습니다.

**-s** 이 옵션을 사용하면 시스템이 명령을 실행하고 해당 명령의 출력을 로드할 데이터로 사용합니다. 다음 명령행을 사용하십시오.

```
.load -s `<command name>`
```

**-e** -e 옵션이 설정된 경우, 시스템은 로드된 XML 파일 대신 데이터베이스에서 인라인 프로젝트를 가져옵니다. chainID 값을 데이터베이스 내 프로젝트 ID에 대한 참조로 간주합니다. 이 옵션을 사용하여 XML 파일이 XML 파일의 프로젝트 대신 인라인 프로젝트의 최신 버전을 참조하게 하거나 XML 파일에 포함되지 않은 프로젝트를 참조하게 할 수 있습니다.



- o XML 파일의 인라인 프로젝트를 사용 불가능하게 하려면 -o 옵션을 사용하십시오. 이 옵션이 사용되는 경우, 시스템은 기본 프로젝트의 모든 인라인 프로젝트를 무시합니다. 인라인 프로젝트에 대한 참조를 포함하는 단계는 명령을 실행하지만 인라인을 무시합니다.
- j XML 파일의 마지막 단계 세트가 스레드되었고 .load 명령 이후의 단계도 스레드된 경우 -j 옵션을 사용하십시오. -j 옵션은 마지막의 스레드된 단계를 결합 단계로 전환합니다. 이 옵션을 사용하지 않으면 스레드된 단계가 .load 명령 이후의 스레드된 단계 블록에 속하게 됩니다.
- v 보려는 표시 터미널(stdout)에 로드되는 XML 파일의 콘텐츠를 보냅니다.

## 샘플 XML

다음 예제는 .load 명령과 함께 사용할 XML 파일을 표시합니다. XML은 HelloWorldPlusInline이라는 프로젝트를 내보내 작성되었습니다.

XML 예제의 다음 세부사항을 참고하십시오.

- XML은 두 개의 <project> 요소를 포함합니다.
- XML의 첫 번째 프로젝트는 1차 프로젝트입니다. 이 프로젝트는 name="HelloWorldPlusInline" 및 primary="1" 속성을 갖습니다.
- XML의 두 번째 프로젝트는 Sleepytime이며 1차가 아님을 표시하기 위한 primary="0" 속성을 갖습니다.
- HelloWorldPlusInline의 첫 번째 단계는 echo 명령과 chainID 속성을 포함하는 EchoHelloWorld라는 단계입니다. chainID 속성 값은 2이며, 시스템이 ID 2(Sleepytime 프로젝트를) 사용하여 프로젝트를 인라인해야 함을 표시합니다.

참고: inline 단계 속성은 무시하십시오. 더 이상 사용되지 않는 권장하지 않는 속성입니다. 모든 단계에서 이 속성 값은 N입니다. 단계에 인라인 프로젝트가 있는지 판별하려면 chainID 속성을 찾으십시오. chainID 값은 프로젝트 ID를 참조합니다. 기본적으로 시스템은 XML 파일에서 인라인 프로젝트를 찾지만, .load 명령에 -e 옵션을 사용하는 경우 시스템은 해당 값을 데이터베이스 내 프로젝트 ID로 간주합니다. 따라서 인라인 프로젝트를 포함시키지 않아도 자체 .load 파일을 작성할 수 있습니다.

참고: Build Forge 버전 7.0.x 이하에서 생성된 XML 파일에서 .load의 사용은 지원되지 않으며 이전 구문이 UUID가 아닌 이름으로 오브젝트를 참조하므로 올바르게 작동하지 않습니다.

- 각 프로젝트에는 id 속성이 있습니다. 이 ID 값은 데이터베이스의 프로젝트 ID와 동일합니다. 설치 디렉토리에서 다음 명령을 실행하여 프로젝트 ID 목록을 가져올 수 있습니다.

```
bfexport -l
```

- commandStore 속성은 특정 문자 XML이 이스케이프 처리된 <command> 요소의 콘텐츠 사본을 포함합니다. 예를 들어 <command>에서 ", ', &, <, > 또는 줄 바꿈 기가 포함될 수 있습니다. 문자는 &quot;, &apos;, &amp;, &lt;, &gt; 및 &#10;으로 각각 commandStore에 표시됩니다.

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<buildforge schema="7.115014" comment="">
  <project access="6" active="Y" name="HelloWorldPlusInline" primary="1"
    selectorId="Choose_local" maxthread="0" increment="Y" tagsync="0"
    buildclass="Production" sticky="N" envId="0" tag="BUILD_$B" id="19" exclusive="0">
    <tagvar autoincrement="Y" name="B" id="1">2</tagvar>
    <step absolute="N" failwait="N" selectorId="" dir="/" broadcast="N"
      timeout="300" id="1" passwait="N" inline="N" threadable="N" chainId="2"
      access="6" active="Y" passnotify="0" description="EchoHelloWorld" onfail=" "
      failnotify="0" envId="0" commandStore="echo Hello World">
      <command>echo Hello World</command>
    </step>
    <step absolute="N" failwait="N" selectorId="" dir="/" broadcast="N"
      timeout="300" id="2" passwait="N" inline="N" threadable="N" access="6"
      active="Y" passnotify="0" description="export proj to build and server folders"
      onfail=" " failnotify="0" envId="0"
      commandStore=".export $BF_PROJECTNAME_PHYS.xml&#10;copy

      /Y $BF_PROJECTNAME_PHYS.xml ..\..\>
      <command>.export $BF_PROJECTNAME_PHYS.xml
copy /Y $BF_PROJECTNAME_PHYS.xml ..\..\</command>
    </step>
  </project>
  <project access="6" active="Y" name="Sleepytime" primary="0"
    selectorId="Choose_local" maxthread="0" increment="Y" tagsync="0"
    buildclass="Production" sticky="N" envId="0" tag="SLEEP_$B" id="2" exclusive="0">
    <tagvar autoincrement="Y" name="B" id="1">21</tagvar>
    <step absolute="N" failwait="N" selectorId="" dir="/" broadcast="N"
      timeout="300" id="1" passwait="N" inline="N" threadable="N" access="6"
      active="Y" passnotify="0" description="Sleep, perchance to dream" onfail=" "
      failnotify="0" envId="0" commandStore=".sleep 0">
      <command>.sleep 0</command>
    </step>
  </project>
  <class maxdays="0" access="1" entranceprojectId="1" name="Production" keepfiles="B"
    deletechangedata="N" purgeprojectId="2" exitProjectId="5" candidates="AnyBuild "
    maxbuilds="0"></class>
  <selector operator="" required="" access="6" value="" name="Choose_local"
    selectorId="" property=""></selector>
</buildforge>
```

## .lock

```
.lock
```

.lock 명령은 시스템이 작업을 완료한 다음 해당 작업을 잠그도록 합니다. 이로써, 작업이 해당 클래스의 특성에 따라 자동으로 삭제되지 않도록 합니다. 또한 잠긴 실행은 작업 > 완료 탭에 나열되지 않으며 대신 잠김 탭에 표시됩니다. 이 명령은 매개변수를 갖지 않습니다. 명령이 사용되는 작업을 잠급니다.

## **.mkdir**

```
.mkdir <relative_path>
```

.mkdir 명령은 디렉토리를 작성합니다. <relative\_path> 매개변수는 현재 단계 디렉토리의 상대 경로로 해석됩니다. 경로 이름 스펙의 디렉토리가 존재하지 않는 경우에는 디렉토리가 작성됩니다. 절대 경로 및 드라이브 이름을 포함하는 경로(예: C:\)는 허용되지 않습니다.

## **.monitor**

```
.monitor [-c] [-w] <interval> [<relative_path>/]<filename>
```

.monitor 명령은 파일 크기 변경이 중지될 때 보려는 파일을 감시하는 동안 시스템이 프로젝트를 정지하게 합니다. 단계가 이 명령을 실행하면 시스템은 표시된 파일을 확인합니다. 그런 다음, <interval>초마다 파일을 다시 확인합니다. 파일 크기가 두 간격 간에 변경되지 못하면 시스템은 다음 단계를 계속합니다.

-c 옵션을 사용하는 경우, 시스템은 변경을 중지시켰음을 판별한 후에 모니터된 파일의 콘텐츠를 단계 로그에 기록한 후 다음 단계를 계속합니다.

파일이 없는 경우, 시스템은 기다리지 않고 첫 번째 간격 직후에 계속합니다. 시스템이 모니터링 프로세스를 시작하기 전에 파일이 작성되기를 강제로 기다리게 하려면 -w 옵션을 사용하십시오.

## **.pack**

**설명**     패키징 애플리케이션용 어댑터를 프로젝트 단계에 추가하려면 .pack 명령을 사용하십시오. 패키징 어댑터는 Build Forge 오브젝트입니다. 패키징 애플리케이션용 어댑터 템플릿을 기반으로 합니다. 프로젝트가 실행될 때 단계에 대한 어댑터 코드가 실행됩니다.

### **구문**

```
.pack <adaptor_name> [entry_name]
```

<adaptor\_name>은 필수입니다. 이 이름은 관리 콘솔에서 어댑터에 지정된 이름입니다. <adaptor\_name>의 대소문자는 콘솔에서 사용되는 대소문자와 일치해야 합니다.

어댑터 템플릿에 여러 인터페이스 함수가 있는 경우, entry\_name 옵션을 사용하여 실행할 함수를 지정하십시오. entry\_name은 어댑터 템플릿의 <interface> 요소에 지정된 이름 속성과 일치해야 합니다. entry\_name에 지정된 <interface> 요소가 없거나 찾을 수 없는 경우 대신 기본 <interface> 요소가 실행됩니다. 다음 예제에서 항목 이름은 PackageFunction입니다.

어댑터 링크를 사용하는 경우, 어댑터가 자동으로 호출되고 어댑터 템플릿의 첫 번째 인터페이스 함수가 실행됩니다. 다른 인터페이스를 실행하려면 어댑터 템플릿에서 실행하려는 인터페이스의 기본 속성을 `true(default="true")`로 설정하십시오.

예제:

```
.pack MyPackagingAdaptor
.pack MyPackagingAdaptor PackageFunction
```

참고: 어댑터를 작성하거나 어댑터 목록을 보려면 **프로젝트 > 어댑터**를 선택하십시오.

Build Forge 제품과 함께 제공되는 어댑터 템플릿은 다음 위치에 있습니다.

```
<bfinstall>/interface
```

## .pop

```
.pop [-p] <register_name> [+][<relative_pathname>|-]
.pop [-p] <register_name> [>|>>]<register_name>
```

파일, 단계 로그 또는 다른 레지스터에 레지스터 콘텐츠를 기록하십시오.

선택적 `-p` 매개변수를 사용하면 명령이 프로젝트 레지스터를 참조합니다. 프로젝트 레지스터는 일반 레지스터와 별개이며, 프로젝트 레지스터는 작업이 종료된 후에도 지속됩니다.

다음 예제는 `.pop` 명령의 다양한 사용법을 보여줍니다.

- `.pop A data.txt`

레지스터 A는 단계의 작업 디렉토리에 있는 `data.txt` 파일에 기록됩니다.

- `.pop ver +data.txt`

레지스터 `ver`의 콘텐츠는 파일 `data.txt`에 추가됩니다.

- `.pop Alpha`

레지스터 Alpha의 콘텐츠는 단계의 로그에 기록됩니다.

- `.pop ALPHA > BETA`

레지스터 BETA의 콘텐츠를 레지스터 ALPHA의 콘텐츠와 동일하게 작성합니다.

- `.pop A >> B`

레지스터 A의 콘텐츠가 레지스터 B에 추가됩니다.

참고: 레지스터를 팝핑해도 레지스터는 비지 않습니다. 레지스터 콘텐츠를 변경하려면 `.push` 명령을 사용하여 레지스터에 새 값을 푸시하십시오.

## .poptag

`.poptag [-p]<registername>`

`.poptag` 명령은 현재 태그를 지정된 레지스터의 콘텐츠로 바꾸어 해당 태그를 변경합니다.

선택적 `-p` 매개변수를 사용하면 명령이 프로젝트 레지스터를 참조합니다. 프로젝트 레지스터는 일반 레지스터와 별개이며, 프로젝트 레지스터는 작업이 종료된 후에도 지속됩니다.

## .purge

`.purge`

`.purge` 명령을 사용하여 작업이 완료되자마자 제거되도록 하는 플래그를 설정하십시오. `.purge` 명령 다음에 실행되는 `.lock` 명령으로 작업이 대신 저장됩니다. 이 명령을 사용하여 모든 단계가 성공적으로 완료되는 경우에만 저장되는 작업을 작성할 수 있습니다. 이러한 작업을 작성하려면 프로젝트의 첫 번째 단계에서 `.purge` 명령을 작성하고 마지막 단계에서 `.lock` 명령을 작성하십시오.

## .push

`.push [-p] [+]<register_name> [<relative_pathname> | -]`

`<relative_pathname>`의 콘텐츠를 `<register_name>` 레지스터에 두십시오. `<register_name>`의 현재 콘텐츠가 바뀝니다.

바꾸지 않고 추가하려면 `<register_name>` 앞에 더하기 부호(+)를 두십시오.

레지스터를 지우려면 `<relative_pathname>` 대신 하이픈을 사용하십시오.

선택적 `-p` 매개변수를 사용하면 명령이 프로젝트 레지스터를 참조합니다. 프로젝트 레지스터는 일반 레지스터와 별개이며, 프로젝트 레지스터는 작업이 종료된 후에도 지속됩니다.

`<relative_pathname>`은 단계의 절대 특성을 사용하지 않는 한 프로젝트 또는 태그 경로와 관련이 있습니다.

다음 예제에서는 절대 특성 단계에 사용되지 않는다고 가정합니다.

- `.push Alpha data.txt`

단계의 작업 디렉토리에 있는 `data.txt` 파일의 콘텐츠는 레지스터 Alpha에 배치됩니다.

- `.push +B ..\newdata.txt`

단계 작업 디렉토리의 상위 디렉토리에 있는 newdata.txt 파일의 콘텐츠는 레지스터 B에 추가됩니다.

- .push ALPHA -

레지스터 ALPHA가 지워집니다.

## .put

```
.put [<relative_path>/]file server:[<relative_path>/]file]
```

한 논리 서버에서 다른 논리 서버로 파일을 전송하려면 .put 명령을 사용하십시오. .put 오퍼레이션은 현재 서버/경로에서 실행되며 지정된 파일을 원격 서버로 보냅니다. 대상 경로 이름은 대상 서버의 기본 경로와 연관됩니다. 소스 경로 이름은 단계의 현재 작업 디렉토리와 연관됩니다. 원격 서버는 파일에 대한 .put 오퍼레이션을 허용하는 논리 서버를 지정해야 합니다(418 페이지의 『서버 자원에서 파일 복사 사용』 참조). 단일 파일만 전송할 수 있습니다.

경로 스펙은 환경 변수를 포함할 수 있습니다. 이 기능을 사용하면 특정 작업에 사용된 경로와 관련된 파일을 지정할 수 있습니다. 468 페이지의 『작업용 작업 디렉토리』에서 작업에 사용된 경로에 대한 설명을 참조하십시오.

사용 중인 서버 이름에 공백이 있는 경우, 이름을 따옴표로 묶으십시오.

전송 속도가 빠르지 않으므로 다른 방법을 선택하여 대형 파일을 전송할 수 있습니다. 속도는 초당 40KB를 넘지 않습니다. 따라서 70MB의 파일을 전송하려면 45분에서 1 시간 정도 소요될 수 있습니다.

참고: 대상 파일이 이미 존재하는 경우 경고하지 않고 이를 겹쳐씹습니다.

## .rem

### 설명

.rem 점 명령은 주석을 위한 것입니다. 단계의 명령에 대한 참고사항을 작성하는 데 유용합니다.

### 예제

```
.rem This is some comment text here  
.rem all text after .rem is ignored.
```

## .report

빠른 보고서에서 작성하는 보고서에 대한 보고서 출력을 작업 BOM에 추가하십시오.

빠른 보고서는 Rational Build Forge의 별도 라이선스 기능입니다.

설명 보고서를 프로젝트에 추가하려면 .report 명령을 사용하십시오.

.report 명령은 보고서를 실행하고 해당 출력을 작업 BOM에 추가합니다. 작업이 실행될 때마다 BOM은 현재 데이터 기반 보고서 결과를 표시합니다.

## 구문

```
.report <report_name>
```

<report name>에 대한 제한사항은 다음과 같습니다.

- <report name>은 필수입니다.
  - 이름은 대소문자를 구분합니다.
  - 이름에 공백이 포함되는 경우 따옴표로 묶으십시오(예: "my\_report").
- 사용자가 지정하는 <report\_name>은 공용 보고서로 저장되어야 합니다.
- BOM 보고서 유형은 지원되지 않습니다.

## .retag

```
.retag <new_tag>
```

단계에서 .retag 명령을 사용하여 작업 중 작업의 태그를 변경하십시오. 변수 또는 명령을 새 태그 값으로 사용할 수 있습니다.

## .retry

```
.retry <count> <command>
```

실패 시 명령을 재시도하려면 .retry 명령을 사용하십시오. .retry 명령은 명령 재시도 횟수를 지정하는 단일 계수 인수를 갖습니다. 실행할 명령이 나머지 인수로 간주되므로 .retry 명령은 단계의 최종 점 명령이어야 합니다. 예를 들어, 다음 명령을 고려하십시오.

```
.retry 3 myscript.sh arg1 arg2 arg3
```

이는 단계가 실패하기 전에 『myscript.sh arg1 arg2 arg3』를 최대 3번 실행합니다. 성공 상태를 리턴하는 명령의 첫 번째 호출이 재시도 프로세스를 중지시킵니다.

## .rget

```
.rget server:[<path>] [<path>]
```

.rget 명령은 .get 명령과 같이 작동하지만 전체 디렉토리 트리를 재귀적으로 복사합니다. 매개변수로 디렉토리 이름을 제공해야 합니다. 예를 들어,

```
.rget winbuildserver1:config myconfig
```

명령은 서버 winbuildserver에 있는 config 디렉토리의 콘텐츠를 현재 서버의 myconfig 디렉토리로 복사합니다.

## 참고:



- 이 명령에 환경 변수를 사용하지 마십시오.
- 이 명령에 기호 링크를 사용하지 마십시오.
- 명령 사용을 시도하기 전에 지정된 디렉토리가 있어야 합니다. 지정된 디렉토리가 아직 없으면 명령에 실패합니다.

## **.rmdir**

```
.rmdir <relative_path>
```

.rmdir 명령은 <relative\_path>로 지정한 디렉토리를 제거합니다. 시스템은 경로 이름으로 지정된 기본 디렉토리와 해당 디렉토리의 모든 콘텐츠 및 하위 디렉토리를 제거합니다.

## **.rput**

```
.rput [<relative_path>] server:[<relative_path>]
```

.rput 명령은 .put 명령과 같이 작동하지만 전체 디렉토리 트리를 재귀적으로 복사합니다. 사용자가 제공하는 상대 경로는 파일이 아닌 디렉토리여야 합니다. 예를 들어,

```
.rput myconfig linuxserver5:feb2005
```

명령은 myconfig 디렉토리의 콘텐츠를 현재 서버에서 linuxserver5 서버의 feb2005 디렉토리로 복사합니다.

소스 경로는 단계의 작업 디렉토리와 연관되므로 단계의 절대 특성 값을 기반으로 프로젝트 및 태그 디렉토리를 포함하거나 포함하지 않습니다. 대상 경로는 오직 대상 서버의 경로 특성과 연관됩니다. 시스템의 경로 생성 방법에 대한 자세한 정보는 468 페이지의 『작업용 작업 디렉토리』를 참조하십시오.

### **참고:**

- 이 명령에 환경 변수를 사용하지 마십시오.
- 이 명령에 기호 링크를 사용하지 마십시오.
- 명령 사용을 시도하기 전에 지정된 디렉토리가 있어야 합니다. 지정된 디렉토리가 아직 없으면 명령에 실패합니다.

## **.run 및 .runwait**

```
.run [-c "<condition>"] "<ProjectName>" ["<ProjectSnapshotName>"]
```

```
.runwait [-c "<condition>"] "<ProjectName>" ["<ProjectSnapshotName>"]
```

.run 및 .runwait 명령을 사용하여 단계 명령에서 체인 프로젝트를 실행할 수 있습니다. 프로젝트의 스냅샷을 지정하려면 선택적 <ProjectSnapshotName> 매개변수를 사용하십시오.



프로젝트를 실행한 후의 동작 방법에서 두 명령이 다릅니다.

- `.run` 명령은 지정된 프로젝트를 체인으로 실행하며 체인 프로젝트에 대한 환경 변수 상속 규칙을 따릅니다.
- `.runwait` 명령은 지정된 프로젝트를 실행합니다. 실행 단계는 실행된 프로젝트가 완료될 때까지 대기합니다. 실행된 프로젝트가 완료되면 시스템은 실행된 프로젝트의 완료 상태에 따라 실행 단계의 결과 값을 패스 또는 실패로 설정합니다.

**중요사항:** `.runwait`를 사용하는 단계를 포함하는 프로젝트는 실행될 때 두 개의 작업 슬롯을 사용합니다. 사용 가능한 작업 슬롯이 충분하지 않은 경우, 오류로 인해 단계가 실패합니다.

**중요사항:** `.run` 또는 `.runwait` 명령을 통해 시작한 프로젝트는 빌드 결과 페이지에서 체인 링크 아이콘을 생성하지 않습니다.

## 조건부 실행

선택적 `-c` 매개변수를 사용하여 조건에 따라 실행할 수 있습니다. 조건에서 환경 변수를 사용할 수 있습니다. 조건은 다음 여러 양식 중 하나일 수 있습니다.

### 문자열 비교

동일(=) 또는 동일하지 않음(!=) 연산자를 사용하여 문자열을 평가할 수 있습니다. 비교가 참으로 평가되면 체인이 실행됩니다.

### 숫자 비교

`<`, `>`, `<=`, `>=` 또는 `=` 연산자를 사용하여 두 개의 숫자 값을 비교할 수 있습니다.

### 명령 성공

백틱(backtick)으로 묶은 명령을 `-c` 매개변수 값으로 사용할 수 있습니다. 시스템은 명령을 실행합니다. 명령이 성공하면 체인이 실행됩니다.

## 예제:

```
.run "BuildWindowsDriver"
```

시스템이 `BuildWindowsDriver` 프로젝트를 실행합니다. 실행 프로젝트는 다음 단계를 즉시 계속합니다.

```
.runwait "BuildWindowsDriver"
```

시스템이 `BuildWindowsDriver` 프로젝트를 실행합니다. 시스템은 `.runwait` 단계에서 실행 프로젝트를 일시정지합니다. `BuildWindowsDriver` 프로젝트가 완료되어 전달되면, `.runwait` 단계의 상태가 패스로 설정됩니다.

```
.run -c "$HOMEDRIVE=C:" "Simple Echo"
```

시스템은 HOMEDRIVE 변수 값이 C인 경우에만 Simple Echo 프로젝트를 실행합니다.

이 명령은 단계 로그의 EXEC 섹션에 다음과 같은 로그 출력을 생성합니다.

- HOMEDRIVE가 C:인 경우:

```
.run Condition: 'C:' = 'C:' satisfied.  
  
Queueing Project "Simple Echo" on server [WinBox].  
Queued Build 'BUILD_202' of project 'Simple Echo'.
```

- HOMEDRIVE가 C:가 아닌 경우:

```
.run -c "$HOMEDRIVE=C:" "Simple Echo"  
  
.run Condition: 'D:' = 'C:' unsatisfied, no project queued.
```

시스템은 문자열에 숫자가 포함된 경우 문자열을 수치적으로 비교합니다. 예를 들어, 다음 경우를 표시된 대로 처리합니다.

```
.runwait -c "a12b<c42d" "Simple Echo"  
  
.run Condition: '12' < '42' satisfied.  
Queueing Project "Simple Echo" on server [WinBox].  
.run 빌드(4411)가 완료되기를 기다리는 중입니다.  
.run 빌드가 지금 실행 중입니다.  
.run build has finished.  
Build 'BUILD_203' of project 'Simple Echo' completed.
```

```
.runwait -c "f43g<>h43i" "Simple Echo"  
  
.run Condition: '43' <> '43' unsatisfied, no project queued.
```

다음 예제는 명령을 조건으로 사용하는 방법을 보여줍니다. 따옴표와 백틱(backtick) 모두로 명령을 묶어야 한다는 점에 유의하십시오.

```
.run -c "`exit 1`" "Simple Echo"  
  
Env .run encountered an error during variable expansion,  
parameter [``exit1`] expanded to [].  
Expansion returned non-zero exit, project will not be queued.  
  
.run -c "`exit 0`" "Simple Echo"  
  
Expansion returned zero exit, project will be queued.  
Queueing Project "Simple Echo" on server [WinBox].  
Queued Build 'BUILD_204' of project 'Simple Echo'.
```

.runwait를 사용하고 빌드가 실패하는 경우, 로그는 다음과 같습니다.

```
.runwait "Fail Build"
```

Queueing Project "Fail Build" on server [WinBox].  
Waiting for .run build (4413) to complete.  
.run 빌드가 지금 실행 중입니다.  
.run build has finished.  
Build 'BUILD\_3' of project 'Fail Build' Failed, setting step status to fail.

## **.scan**

`.scan [-v][-i <ignorepattern>] baseline | checkpoint`

.scan 명령을 사용하여 작업 BOM에 저장된 데이터를 강화하십시오. 단계의 작업 디렉토리에서 각 파일에 대한 MD5 값과 함께 파일을 추적합니다.

### **.scan baseline**

단계의 작업 디렉토리에 모든 파일 목록을 저장합니다. 시스템은 목록을 작업 BOM에 범주로 표시합니다. 하나의 작업에 여러 개의 baseline 명령을 가질 수 있지만 명령이 실행될 때 각 명령은 목록을 단계의 작업 디렉토리 상태로 다시 설정합니다. 최종 BOM은 하나의 기준선 범주만 표시합니다.

### **.scan checkpoint**

작업에서 마지막 .scan baseline 또는 .scan checkpoint 명령 이후에 변경 및 삭제된 모든 새 파일 목록을 저장합니다. 시스템은 BOM에 목록을 표시합니다. 각 checkpoint 명령은 BOM에 새 범주를 작성합니다.

작업에서 첫 번째 .scan checkpoint 명령 전에 .scan baseline 명령을 사용해야 합니다. .scan baseline 명령에 우선하는 .scan checkpoint 명령은 무시됩니다.

명령 옵션:

- v** 작업 로그에 변경 정보 사본을 기록하십시오.
- i** 제공된 패턴과 일치하는 디렉토리를 무시하십시오. 패턴은 경로의 시작, 끝 또는 임의의 디렉토리 부분과 일치할 수 있습니다. 이 옵션을 사용하여 변경 목록에서 소스 제어 디렉토리를 제거할 수 있습니다.

CVS의 예제:

`.scan -i CVS checkpoint`

예제 명령은 보고서에서 CVS 디렉토리를 유지합니다.

서브버전의 예제:

`.scan -i .svn baseline`

-v가 -i와 함께 사용되는 경우, 시스템은 소스 제어 디렉토리에 대한 변경사항을 로깅하지만 변경사항이 BOM에는 포함되지 않습니다.

**참고:** 단일 단계에서 두 개 이상의 `.scan` 명령을 사용하지 마십시오. 단일 단계에서 두 개 이상의 명령을 사용하는 경우 시스템이 `.scan` 명령에 대한 정확한 출력을 제공할 수 없습니다.

이러한 명령 사용에 대한 자세한 정보는 459 페이지의 『`.scan` 명령을 사용하여 기준선 및 체크포인트 추가』를 참조하십시오.

## **.semget**

```
.semget <semaphore_name>
```

단계가 이 명령을 실행하면 시스템은 이 명령에 명시된 이름의 세마포어가 있는지 여부를 확인합니다.

- 해당 세마포어가 없는 경우, 시스템이 해당 세마포어를 작성한 다음 단계 작업에 지정합니다. 그러면, 다음 단계가 계속 실행됩니다.
- 일부 다른 작업이 이 세마포어 이름을 이미 청구한 경우, 다른 프로젝트가 세마포어를 해제할 때까지 `.semget` 단계에서 작업이 정지됩니다.

이 명령 사용에 대한 자세한 정보는 645 페이지의 『세마포어(semaphore)』를 참조하십시오.

## **.semput**

```
.semput <semaphore_name>
```

`.semput` 명령을 사용하여 `<semaphore_name>`이라는 이름의 세마포어를 해제하십시오. 이 명령 사용에 대한 자세한 정보는 645 페이지의 『세마포어(semaphore)』를 참조하십시오.

## **.set**

```
.set env <EnvGroupName>[(SnapshotName)]  
"<VariableName>=<DesiredValue>" [...]
```

`.set` 명령은 환경 변수에 값을 지정합니다. 추가 변수 및 값을 지정할 수 있습니다. 각 변수와 값을 따옴표 안에 넣으십시오. 스냅샷을 지정하지 않으면 이 명령은 환경 그룹의 기본 스냅샷에 대한 변수를 설정합니다. 스냅샷을 지정하는 경우에는 환경 그룹 이름과 스냅샷 이름 사이에 공백이 없도록 하여 괄호 안에 넣으십시오.

**참고:** 이 명령이 설정하는 변수가 이미 있어야 합니다.

`.set` 명령을 사용하여 환경의 *마스터 레코드*를 변경하십시오. 시스템은 프로젝트를 실행할 때 마스터 레코드에서 프로젝트 환경 사본을 작성하고 작업 레코드에 사본을 저장하며, 해당 사본을 프로젝트 기본값으로 사용합니다.

단계가 실행되면 마스터 레코드가 아니라 환경의 작업 사본을 사용합니다. 따라서 `.set` 를 사용하면 다음과 같은 결과를 초래합니다.

- `.set` 명령이 한 단계에서 실행되면, 기본 단계 환경을 사용하는 나중 단계에서는 변경사항이 확인되지 않습니다. 시스템은 단계에 대해 기본 환경의 작업 사본을 사용합니다.
- `.set` 명령이 지정된 환경에서 실행되면, 해당 환경을 지정하는 나중 단계에서 작성한 변경사항이 확인됩니다. 단계가 환경을 지정하면 시스템이 환경의 마스터 레코드를 읽습니다. 지정된 단계 환경이 프로젝트 기본값과 동일한 환경인 경우에도 이렇게 작동합니다.
- `.set` 명령으로 작성된 변경사항은 작업이 종료된 이후에도 지속됩니다. 향후 실행에서는 이전에 실행된 `.set` 명령으로 작성된 값을 사용합니다.

이 명령 사용에 대한 자세한 정보는 412 페이지의 『작업 데이터에 대한 작업』을 참조하십시오. 유사한 명령인 423 페이지의 『`.bset`』도 참조하십시오.

## **.sleep**

```
.sleep <seconds>
```

`.sleep` 명령을 사용하여 단계가 일시정지될 시간(초 단위)을 지정하십시오. 관리 콘솔이 이 명령을 처리하므로 원격 서버와 연결이 작성되지 않습니다. 또한 『`.sleep 0`』 명령을 플랫폼 독립 널(`null`) 명령으로 사용할 수 있습니다.

## **.snapshot**

`.snapshot` 명령을 사용하여 호출하는 프로젝트의 인스턴스를 새로 작성하고 데이터베이스에서 프로젝트 스냅샷으로 인스턴스를 저장하십시오. 프로젝트 스냅샷은 실행 가능 프로젝트입니다.

**설명** `.snapshot` 명령은 프로젝트의 스냅샷은 물론 스냅샷하거나 복사하기 위해 선택하는 연관 오브젝트도 작성합니다.

`.snapshot` 옵션을 사용하여 다음 테이블에서 설명하는 대로 스냅샷하거나 복사할 오브젝트를 지정하십시오. 옵션을 지정하지 않으면 프로젝트 정의, 단계 및 태그만 포함됩니다.

스냅샷 이름은 필수이고 프로젝트 스냅샷 세트에 고유해야 합니다. 또한 스냅샷 이름은 스냅샷하는 기타 오브젝트에도 지정됩니다.

`.snapshot`을 실행하는 프로젝트가 완료된 후 프로젝트 스냅샷이 호출하는 프로젝트의 하위로 UI에 표시됩니다. 또한 기타 스냅샷 오브젝트는 기본 스냅샷 또는 상위 오브젝트의 하위로도 UI에 표시됩니다.

## **구문**

```
.snapshot -v <"snapshot_name"> [-c <"comment">]  
[-e[f]] [-s[f]] [-pI] [-pC] [-a] [-t] [-r] [-g]
```

옵션	설명
-v <"snapshot_name">	프로젝트 스냅샷의 이름은 필수입니다. 스냅샷 이름은 프로젝트에 고유해야 합니다. 이름은 따옴표 안에 넣어야 합니다.
-c <"comment">	선택적 주석을 스냅샷의 파트로 저장합니다. 주석은 따옴표 안에 넣어야 합니다.
-e	프로젝트 스냅샷이 작성되면 프로젝트 및 단계 환경을 스냅샷합니다.
-ef	f 옵션을 추가하면 스냅샷 환경이 포함 환경 변수 유형과 함께 포함하는 모든 환경도 스냅샷합니다.
-s	프로젝트 스냅샷이 작성되면 프로젝트 및 단계 선택기를 스냅샷합니다.
-sf	f 옵션을 추가하면 스냅샷 선택기가 포함 선택기 특성 유형과 함께 포함하는 모든 선택기도 스냅샷합니다.
-pl	프로젝트 스냅샷이 작성되면 인라인 프로젝트 또는 라이브러리 및 해당 단계를 스냅샷합니다.  인라인 프로젝트 또는 라이브러리는 단계에서 트리거되고 단계 완료 후에 실행됩니다.
-pC	프로젝트 스냅샷이 작성되면 체인 프로젝트 또는 라이브러리 및 해당 단계를 스냅샷합니다.  프로젝트 패스/실패 조건 또는 단계 패스/실패 조건에 따라 프로젝트 또는 라이브러리를 트리거할 수 있습니다. 두 유형의 조건부 체인 프로젝트 또는 라이브러리에 대해 모두 스냅샷을 작성합니다.
-a	프로젝트 스냅샷이 작성되면 어댑터 링크를 복사합니다.  어댑터 링크는 프로젝트에 어댑터를 추가하고 어댑터 코드를 실행합니다.
-t	프로젝트 레벨 및 단계 레벨에서 설정되는 패스 및 실패 알림 이벤트에 대한 알림 템플릿을 복사합니다.
-r	프로젝트 스냅샷이 작성되면 프로젝트 레지스터를 복사합니다.
-g	프로젝트 태그 변수의 태그 값을 복사합니다.  태그 변수는 자동으로 복사되지만 해당 값은 복사되지 않습니다. 태그 값을 복사하지 않으면 1로 재설정됩니다.

## .source

**설명** 소스 코드 애플리케이션용 어댑터를 프로젝트 단계에 추가하려면 .source 명령을 사용하십시오. 소스 코드 어댑터는 Build Forge 오브젝트입니다. 소스 코드 애플리케이션용 어댑터 템플릿을 기반으로 합니다. 프로젝트가 실행될 때 단계에 대한 어댑터 코드가 실행됩니다.

### 구문

```
.source <adaptor_name> [entry_name]
```

<adaptor\_name>은 필수입니다. 이 이름은 관리 콘솔에서 어댑터에 지정된 이름입니다. <adaptor\_name>의 대소문자는 콘솔에서 사용되는 대소문자와 일치해야 합니다.

어댑터 템플릿에 여러 인터페이스 함수가 있는 경우, *entry\_name* 옵션을 사용하여 실행할 함수를 지정하십시오. *entry\_name* 옵션은 어댑터 템플릿의 `<interface>` 요소에 지정된 이름 속성과 일치해야 합니다. *entry\_name*에 지정된 `<interface>` 요소가 없거나 찾을 수 없는 경우 대신 기본 `<interface>` 요소가 실행됩니다. 다음 예제에서 항목 이름은 By Date입니다.

어댑터 링크를 사용하는 경우, 어댑터가 자동으로 호출되고 어댑터 템플릿의 첫 번째 인터페이스 함수가 실행됩니다. 다른 인터페이스를 실행하려면 어댑터 템플릿에서 실행하려는 인터페이스의 기본 속성을 `true(default="true")`로 설정하십시오.

#### 예제:

```
.source MyClearCaseAdaptor
.source MyClearCaseAdaptor "By Date"
```

**노트** 어댑터를 작성하거나 어댑터 목록을 보려면 **프로젝트 > 어댑터**를 선택하십시오. Build Forge 제품과 함께 제공되는 어댑터 템플릿은 다음 위치에 있습니다.

```
<bfinstall>/interface
```

## .stop

```
.stop <state of build>
```

.stop은 빌드 처리가 즉시 중지되도록 합니다. 이 명령을 사용하면 빌드가 종료됩니다. 빌드의 가능한 상태는 패스(P), 실패(F), 경고(W) 또는 중단(B)입니다.

## .strsub

```
.strsub <source> <replacement> file [file ...]
```

하나 이상의 텍스트 파일에서 기본 문자열 대체를 수행하려면 .strsub 명령을 사용하십시오. 시스템은 대상 파일에서 `<source>` 문자열을 스캔하며, 일치 발견되면 `<source>` 문자열을 `<replacement>`로 바꿉니다. .strsub 명령은 각 파일의 각각의 모든 행에서 문자열의 모든 인스턴스(소스)를 바꿉니다.

.strsub 명령은 서버에서 사용 가능한 특정 명령에 관계없이 운영 체제 전반에서 작동합니다.

about.c 파일에서 `_VERSION_` 문자열을 바꾸려면 다음 명령을 사용하십시오.

```
.strsub _VERSION_ 2.34 about.c
```

와일드카드를 사용하지 않고 하나 이상의 파일 이름을 정확히 지정해야 합니다. 예를 들어, 다음과 같은 명령은 실패합니다.

```
.strsub _VERSION_ 2.34 *.txt
```



그러나 명령에 변수를 사용할 수 있으므로 VERSION 및 FILENAME 변수가 환경에 정의된 경우에는 다음과 같은 명령이 작동합니다.

```
.strsub _VERSION_ ${VERSION} ${FILENAME}
```

참고: 공백을 사용하여 명령의 매개변수를 분리하십시오.

.strsub 명령은 .edit 명령과 유사하며, 차이점은 다음과 같습니다.

- .strsub 명령은 대형 텍스트 파일 또는 다수의 파일에서 대체를 수행할 때 .edit보다 속도가 빠릅니다.
- .edit 명령은 정규 표현식 검색 및 바꾸기를 수행할 수 있습니다.
- .edit 명령은 각 파일의 각 모든 행에서 문자열의 첫 번째 인스턴스만 (search\_expression) 바꿉니다.
- .strsub 명령은 각 파일의 각각의 모든 행에서 문자열의 모든 인스턴스(소스)를 바꿉니다.

.edit 명령은 기본적으로 POSIX 확장 정규 표현식 구문을 사용합니다. 에이전트가 Perl 호환 가능 정규 표현식 지원으로 컴파일된 경우, (PCRE 구문을 사용해야 함을 표시하는) "p" 문자 다음에 대체 표현식이 올 수 있습니다.

두 경우 모두 에이전트 처리에서 표현식이 두 번 해석됩니다. 따라서 단일 백슬래시가 일반적으로 사용되는 곳은 어디든지 4개의 백슬래시를 사용해야 합니다. 예를 들어, 다음과 같습니다.

4개의 백슬래시는 리터럴 마침표를 이스케이프합니다:  
\\\\.

또는 /x 플래그를 사용하여 백슬래시를 억제할 수 있습니다.

```
.strsub/x
```

## .test

**설명** 테스트 애플리케이션용 어댑터를 프로젝트 단계에 추가하려면 .test 명령을 사용하십시오. 테스트 어댑터는 Build Forge 오브젝트입니다. 테스트 애플리케이션용 어댑터 템플리트를 기반으로 합니다. 프로젝트가 실행될 때 단계에 대한 어댑터 코드가 실행됩니다.

### 구문

```
.test <adaptor_name> [entry_name]
```

<adaptor\_name>은 필수입니다. 이 이름은 관리 콘솔에서 어댑터에 지정된 이름입니다. <adaptor\_name>의 대소문자는 콘솔에서 사용되는 대소문자와 일치해야 합니다.

어댑터 템플릿에 여러 인터페이스 함수가 있는 경우, entry\_name 옵션을 사용하여 실행하려는 함수를 지정하십시오. entry\_name은 어댑터 템플릿의



<interface> 요소에 지정된 이름 속성과 일치해야 합니다. *entry\_name*에 지정된 <interface> 요소가 없거나 찾을 수 없으면 기본 인터페이스 요소가 사용됩니다. 다음 예제에서 항목 이름은 TestFunction입니다.

어댑터 링크를 사용하는 경우, 어댑터가 자동으로 호출되고 어댑터 템플릿의 첫 번째 인터페이스 함수가 실행됩니다. 어댑터 템플릿에서 다른 인터페이스를 실행하려면, 실행하려는 인터페이스의 기본 속성을 true(default="true")로 설정하십시오.

예제:

```
.test MyTestAdaptor
.test MyTestAdaptor TestFunction
```

노트 어댑터를 작성하거나 어댑터 목록을 보려면 프로젝트 > 어댑터를 선택하십시오. Build Forge 제품과 함께 제공되는 어댑터 템플릿은 다음 위치에 있습니다.

```
<bfinstall>/interface
```

## .tset

```
.tset env "<VariableName>=<DesiredValue>" [...]
```

.tset 명령은 단계 중에 프로젝트 설정을 일시적으로 변경합니다. .tset를 사용하여 아직 존재하지 않는 변수를 설정할 수 있습니다.

명령은 현재 단계에서 적용됩니다. 단계 내의 모든 명령 및 단계에 대해 지정된 모든 인라인에 적용됩니다. 그러나 스테딩은 이 동작에 영향을 미치지 않습니다. 예를 들어 다음과 같습니다.

1. 단계가 스테드되며 인라인을 지정합니다.
2. 인라인의 첫 번째 단계도 스테드됩니다.

예제에서 .tset 명령은 첫 번째 단계(1)에 적용되지만 인라인(2)의 첫 번째 단계에는 적용되지 않습니다. 두 단계 모두 동일한 스테드 블록의 일부입니다. 이들은 독립적으로 실행됩니다. 403 페이지의 『스테딩: 병렬로 단계 실행』의 내용을 참조하십시오.

.tset 명령으로 설정된 값은 작업 레코드에 기록됩니다. 이 값은 환경의 데이터베이스 레코드를 업데이트하지 않습니다. 이후 작업은 변경사항의 영향을 받지 않습니다.

## .unlock

```
.unlock
```

.unlock은 시스템에서 .lock 명령을 사용하여 잠긴 작업을 해제하도록 합니다. 이제 잠금 해제된 실행은 작업 > 완료됨을 통해 도달한 페이지에 나열되며 이는 잠금 페이지의 목록에서 제거됩니다.



## 제 21 장 작업에 대한 작업

이 주제는 관리 콘솔에서 작업을 보고 실행 및 관리하는 방법에 대해 설명합니다.

### 작업 정보

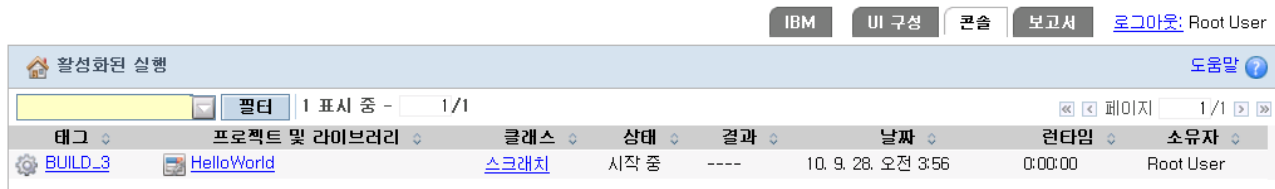
작업은 실행 중이거나 실행을 완료한 프로젝트입니다.

홈 패널 및 작업 패널을 사용하여 작업을 모니터링할 수 있습니다. 또한 작업 패널을 사용하여 작업을 시작하고 작업 결과를 보고 세마포어를 관리할 수 있습니다.

### 홈 패널 정보

홈 패널은 최근 작업 및 시스템 메시지에 대한 정보를 제공합니다.

홈 패널에 액세스하려면 왼쪽 메뉴에서 홈을 클릭하십시오.



활성화된 실행							
도움말 ?							
필터 1 표시 중 - 1/1							
태그	프로젝트 및 라이브러리	클래스	상태	결과	날짜	런타임	소유자
BUILD_3	HelloWorld	스크래치	시작 중	----	10. 9. 28. 오전 3:56	0:00:00	Root User

최근 또는 현재 작업을 보려면 홈 패널을 사용하십시오. 또한 이를 사용하여 알림 및 시스템 메시지를 봅니다. 다음을 보려면 메뉴 항목을 선택하십시오.

- 활성화된 실행 - 현재 실행 중인 작업
- 완료된 실행 - 완료된 작업
- 시스템 메시지 - 시스템 메시지 로그(메시지를 보려면 관리 > 메시지 사용)

### 작업 패널 정보

작업 패널을 사용하여 실행 중인 작업을 수행하고 작업 결과를 봅니다.

작업 패널에 액세스하려면 왼쪽 메뉴에서 작업을 클릭하십시오.

IBM UI 구성 콘솔 보고서 [로그아웃](#) Root User

작업

모두

완료됨

실행 중

아카이브됨

잠금

필터

1 표시 중 - 2/2

1/1

페이지

태그	프로젝트 및 라이브러리	클래스	상태	결과	날짜	런타임	소유자
BUILD_2	HelloWorld	<a href="#">스크래치</a>	완료됨	✓ 패스	10. 9. 27. 오전 4:05	0:00:02	Root User
BUILD_1	HelloWorld	<a href="#">스크래치</a>	완료됨	✓ 패스	10. 9. 27. 오전 4:04	0:00:03	Root User

## 작업 탭

작업 탭을 사용하여 상태(모두, 완료됨, 실행 중, 아카이브됨 또는 잠금)로 작업 목록을 볼 수 있습니다.

탭을 클릭하면 유형별로 작업을 볼 수 있습니다. 각 탭마다 목록의 작업을 관리하는 옵션이 다릅니다.

**모두** 실행 중인 모든 작업이 먼저 표시되며 시작 시간을 기준으로 정렬됩니다. 완료된 모든 작업이 다음에 표시되며 완료 시간을 기준으로 정렬됩니다. 선택란은 회색으로 표시됩니다. 원하는 링크를 클릭할 수 있습니다. 특수 조치는 없습니다.

**완료됨** 완료된 모든 작업이 표시되며 시작 시간을 기준으로 정렬됩니다. 실행 중이거나 아카이브되거나 잠긴 작업은 표시되지 않습니다. 완료된 작업은 제거하거나 잠글 수 있습니다. 원하는 작업의 선택란을 선택한 다음 제거 또는 잠금을 클릭하십시오. 완료된 작업이 제거되었지만 완전히 삭제되지 않은 경우 아카이브됨으로 이동됩니다.

### 실행 중

실행 중인 모든 작업이 표시되며 시작 시간을 기준으로 정렬됩니다. 하나 이상의 실행 중인 작업을 취소할 수 있습니다. 원하는 작업의 선택란을 선택한 다음 취소를 클릭하십시오.

### 아카이브됨

아카이브된 모든 작업이 표시되며 시작 시간을 기준으로 정렬됩니다. 아카이브된 작업은 일반적으로 작업 클래스의 클래스 정의에 지정된 자동 제거에서 일부 정보가 삭제된 작업입니다. 아카이브된 작업은 제거할 수 있습니다. 원하는 작업의 선택란을 선택한 다음 제거를 클릭하십시오.

**잠금** 잠긴 모든 작업이 표시되며 시작 시간을 기준으로 정렬됩니다. 하나 이상의 잠긴 작업을 잠금 해제할 수 있습니다. 원하는 작업의 선택란을 선택한 다음 잠금 해제를 클릭하십시오.

**참고:** 잠긴 작업을 직접 삭제할 수는 없습니다. 그러한 작업을 삭제하려면 먼저 잠금 해제한 다음 완료됨 탭으로 이동하여 제거를 사용하십시오.

452 Rational Build Forge: Information Center

## 작업 목록 필드

필드는 각 작업에 대한 정보를 보여줍니다.

탭의 작업 목록은 완료 시간을 기준으로 최신 작업이 먼저 표시됩니다. 각 작업마다 다음 정보가 옆에 표시됩니다.

**태그** 링크로 표시되는 빌드 태그입니다. 클릭하면 작업 단계에 대한 로그가 표시됩니다.

### 프로젝트 및 라이브러리

링크로 표시합니다. 클릭하면 작업에 사용된 프로젝트가 표시됩니다.

**클래스** 링크로 표시합니다. 클릭하면 작업에 사용된 클래스의 클래스 정의가 표시됩니다.

**상태** 대기 중, 실행 중, 완료됨, 아카이브됨 또는 잠김 중 하나입니다.

- 대기 중 상태는 실행 큐가 가득 찼거나 작업이 세마포어를 대기 중이어서 작업이 시작되지 않았음을 의미합니다. 실행 큐 크기는 시스템 설정 실행 큐 크기로 정의됩니다.
- 실행 큐에 있는 작업은 일반적으로 실행 중입니다. 그러나 실행 큐에 동일한 프로젝트의 여러 작업이 들어 있는 경우, 하나 이상의 작업이 현재 실행 중인지 여부는 프로젝트의 실행 한계 특성에 따라 다릅니다. 353 페이지의 『프로젝트 특성 변경』의 내용을 참조하십시오.

**결과** 패스, 실패, 실패했지만 계속됨, 경고, 중지됨 또는 취소됨 중 하나입니다.

**날짜** 작업 시작 전: 작업이 요청된 날짜 및 시간입니다.

작업 시작 후: 작업이 시작된 날짜 및 시간입니다.

**런타임** 실행 중인 작업의 경우 현재 경과 시간입니다. 완료된 작업의 경우 총 경과 시간.

**소유자** 작업을 시작한 사용자입니다.

## 작업 목록 필터링 및 정렬

탭에서 작업 목록을 필터링 및 정렬할 수 있습니다.

필터링은 작업 목록을 사용자가 지정하는 문자열과 일치하는 작업으로 제한합니다. 정렬을 수행하면 특정 열의 값으로 목록이 정렬됩니다.

- **필터**: 지정된 문자열이 포함된 작업만 표시하도록 작업 목록을 필터링할 수 있습니다. 문자열이 옆에 있으면 필터가 일치합니다.
  - 새 필터: 상자에 문자열을 입력한 다음 필터를 클릭하십시오.
  - 기존 필터: 상자 옆에 있는 화살표를 클릭한 다음 사용할 필터 문자열을 선택하십시오.

- 정렬: 단일 열의 값을 기준으로 작업 목록을 정렬할 수 있습니다. 열 이름 옆에 있는 화살표를 클릭하십시오. 화살표를 여러 번 클릭하면 다음 정렬 순서에 따라 정렬이 반복됩니다.
  1. 오름차순(열 헤더에서 위 화살표가 강조 표시됨)
  2. 내림차순(열 헤더에서 아래로 화살표가 강조 표시됨)
  3. 필터 없음

## 작업 실행 및 결과 보기

작업을 실행, 예약, 취소 및 다시 시작하고 작업 결과를 볼 수 있습니다.

### 작업 시작

작업을 시작하는 방법은 여러 가지가 있습니다.

### 시작하기 전에

작업으로 실행할 프로젝트에는 환경 및 선택기가 정의되어 있어야 합니다.

### 프로시저

- 작업 패널에서: 작업 > 시작을 선택하고 프로젝트 이름을 클릭하십시오. 프로젝트 시작을 클릭할 때처럼, 이 메소드는 프로젝트 시작 페이지를 표시합니다.

작업 세부사항

작업 단계

프로젝트 매개변수

스냅샷: 기본 스냅샷  
선택기: Selector1  
클래스: 스크래치  
태그 형식: BUILD\_.\$B  
태그 예제: BUILD\_1

프로젝트 태그

☐ 편집 가능 태그  
B

프로젝트 환경

환경 저장 ☐

작업이 기본 환경으로 시작됩니다.

- 프로젝트 패널에서: 빠른 시작 아이콘  을 클릭하십시오.


다음 조건이 검사됩니다. 검사를 패스하면 프로젝트가 즉시 시작됩니다.

- 프로젝트에 하나 이상의 단계가 있습니다.
- 프로젝트 환경에서 프로젝트에서 변수 특성에 변경 필수로 설정된 변수가 없습니다.

확인에 실패하면 빠른 시작 아이콘  을 사용하지 않습니다.

이와 같이 프로젝트를 실행하면 선택기, 클래스, 태그 및 환경 변수의 기본값이 사용 됩니다.

빠른 시작 사용 시스템 설정이 예로 설정되면 프로젝트 페이지는 모든 프로젝트를 검사하여 빠른 시작에 알맞은지 판별합니다. 많은 수의 프로젝트가 정의된 경우 프로젝트 페이지가 프로젝트 목록을 표시하는 데 오래 걸릴 수 있습니다. 다른 아이콘은

프로젝트가 시작될 수 있음을 표시합니다. . 이는 버전 7.1.1.1까지 기본 동작이었습니다.

- **단계 패널에서:** 프로젝트의 단계를 보면서 프로젝트 시작을 클릭하십시오. 이 메소드는 프로젝트의 프로젝트 시작 페이지를 표시하며, 이 페이지에서 프로젝트 매개변수, 환경 변수 값을 변경하고 실행에서 제외시킬 단계를 선택할 수 있습니다.
    - 프로젝트 매개변수의 새 값을 선택하십시오.
    - 프로젝트 태그 변수 값을 편집하십시오.
    - 프로젝트 환경 변수 값을 편집하십시오. 변경사항을 이러한 변수의 새 기본값으로 저장하려면 **환경 저장** 선택란을 클릭하십시오.
    - 작업 단계 탭을 선택하여 프로젝트 단계 목록을 표시하십시오. 이 실행에서만 제외시킬 개별 단계를 선택할 수 있습니다.
- 선택한 후에 실행을 클릭하여 프로젝트를 시작하십시오.

## 결과

프로젝트 실행 중에 작업 > 실행 중 페이지를 검토하여 프로젝트 상태를 확인하십시오.

작업 결과를 보려면 작업 > 완료를 선택하여 완료된 작업을 표시하십시오. 태그 이름을 클릭하여 작업 결과 보기 옵션에 액세스하십시오.

## 작업 결과 보기

### 이 태스크 정보

모든 작업 탭의 작업 결과를 볼 수 있습니다.

### 프로시저

1. 작업을 클릭하십시오. 완료 탭이 선택됩니다.
2. 원하는 탭을 클릭하십시오.
3. 보려는 작업의 태그를 클릭하십시오. 작업 결과는 처음에 다음과 같이 표시됩니다.
  - 기본 패널에 단계 목록이 표시됩니다. 각 단계마다 정보 열이 있습니다.

- 단계: 단계 번호
- 단계 이름: 단계 이름. 단계 이름 앞의 그래픽은 단계 유형(단계, 스레드, 브로드캐스트, 결합)을 나타냅니다. 단계 이름을 클릭하면 단계 로그가 표시됩니다.
- 결과: 단계 결과(패스, 실패, 취소)
- 서버(선택기): 단계가 실행된 서버 이름과 서버를 선택한 선택기.
- 런타임: 단계의 경과 시간(시간:분:초)
- 체인: 단계가 다른 작업을 시작한 경우 해당 작업에 대한 링크가 표시됩니다.
- 왼쪽 사이드 메뉴 맨 아래에 다음 항목을 보여주는 메뉴가 나타납니다.
  - 결과: 기본 패널에 단계 목록을 표시합니다.
  - 명세서(BOM): 기본 분할창에 명세서(BOM)를 보여줍니다. BOM에는 작업 단계, 단계 Manifest 및 체크포인트를 보여주는 링크가 포함됩니다(체크포인트를 사용한 경우).
  - 노트: 이 작업에 대해 입력된 노트를 보여줍니다.
  - 단계 로그(이 항목이 열려 있고 단계 이름이 표시됨): 단계 목록을 보여줍니다. 단계를 클릭하면 단계 로그가 표시됩니다.
  - 체인: 체인으로 실행된 작업의 작업 수를 표시합니다(.run 및/또는 .runwait로 시작된 모든 작업은 이 아이콘을 생성하지 않음).

4. 단계 로그를 보려면: 단계 목록에서 단계 이름을 클릭하십시오. 처음에는 모든 범주가 선택됩니다.

실행 중인 작업에서 단계를 보고 있고 단계가 완료되지 않은 경우, 부분 로그가 표시되고 모든 범주가 표시되지 않을 수 있습니다. 실행 중인 단계의 로그 보기를 업데이트하려면 빌드 태그(맨 위에 **작업 >> 태그**로 표시됨)를 클릭한 후 단계를 클릭하십시오. 단계 로그를 볼 때 모든 단계를 표시하려면 모두 표시를 클릭하십시오. 더 적은 단계를 보려면 **1 - m/n** 표시 제어에 *m*의 값을 입력하고 **Enter**를 누르십시오.

5. 단계 로그를 필터링하려면: 범주를 선택하거나 지운 다음 새로 고치기를 클릭하십시오.

```

1 7/8/10 4:43 PM STEP      Step using selector 'My selector'.
2 7/8/10 4:43 PM MANIFEST  BF_LAST_UPDATE=1278618783
3 7/8/10 4:43 PM MANIFEST  BF_NAME=rbf-14
4 7/8/10 4:43 PM MANIFEST  BF_LOADRATIO=0.3333333333333333
5 7/8/10 4:43 PM MANIFEST  BF_JOBS=1
6 7/8/10 4:43 PM MANIFEST  BF_AGENT_VERSION=버전
7 7/8/10 4:43 PM MANIFEST  BF_LAST_REFRESH=1278618783

```



## 작업 상태에 대한 RSS 데이터 피드

RSS 데이터 피드를 사용하여 개별 작업의 상태를 추적하고 필터할 수 있습니다. 작업에 대한 Build Forge RSS 데이터 공급은 Build Forge 관리 콘솔에서 서버 상태와 동일한 정보를 표시합니다.

작업 상태에 대한 RSS 데이터 공급에 등록하려면 다음을 수행하십시오.

1. Build Forge 관리 콘솔에서 작업을 선택하십시오.

웹 브라우저가 RSS 데이터 공급을 발견하고 브라우저 주소 표시줄에 RSS 아이콘을 표시합니다.

2. RSS 집계기 도구에서 Build Forge RSS 데이터 공급을 로드하십시오.

예를 들어, URL을 복사하여 RSS 데이터 공급 목록에 추가하거나 RSS 아이콘을 끌어서 놓아 RSS 데이터 공급 목록에 URL을 추가하십시오.

3. RSS 데이터 공급에 등록하여 URL을 저장하고 업데이트가 발생할 때 알림을 받으십시오.

참고:

- URL 로드 및 RSS 데이터 공급 등록에 대한 정보는 RSS 집계기 도구 문서를 참조하십시오.
- 영어가 아닌 다른 언어로 RSS 데이터 공급을 통해 Build Forge 작업 상태, 시스템 메시지 또는 서버 상태를 보려면 RSS 집계기 도구가 UTF-8 멀티바이트 문자 인코딩을 지원해야 합니다.
- RSS 피드는 작업 활동 보고를 위해 시간소인을 사용합니다. 시간소인은 Build Forge 사용자 계정에 대한 시간대 설정을 기반으로 합니다. 사용자 브라우저가 다른 시간대에 있는 경우 정확한 시간소인을 얻으려면 Build Forge 계정과 동일한 시간대로 이를 설정해야 합니다.

## 실패한 작업 다시 시작

작업이 실패한 경우 다시 시작할 수 있습니다. 다시 시작하면 동일한 태그 하에서 새 실행을 시작합니다. 실행은 작업이 실패한 지점부터 계속됩니다.

작업을 다시 시작하려면 작업 > 완료 탭의 빌드 목록에서 작업 태그를 클릭하십시오. 시스템은 빌드에 대한 정보를 표시하고 패널 맨 위 근처에 작업 다시 시작 단추를 포함합니다.

1. 작업 다시 시작을 클릭하십시오. 시스템이 다시 시작 페이지를 표시합니다.
2. 옵션을 선택하십시오. 시스템이 프로젝트 레코드의 해당 단계에서 명령에 대한 업데이트를 가져오려는 경우 명령 동기화 특성을 설정하십시오. 이를 설정하지 않은 경우 명령은 작업이 처음 시작된 경우와 똑같이 실행됩니다.
3. 다시 시작 페이지의 다시 시작 단추를 클릭하십시오.

다시 시작된 작업은 다음과 같은 점에서 새 작업과 다릅니다.

- 다시 시작된 작업은 실패한 실행과 동일한 태그 번호를 사용하며 완료 목록의 실패한 실행을 바꿉니다.
- 기본적으로 다시 시작된 작업은 실패한 단계에서 시작되며, 이전 실행에서 패스한 단계를 반복하지 않습니다. 그러나 작업을 다시 시작할 때 실제로 실행되는 단계를 선택할 수 있습니다.
- 기본적으로 시스템은 이전 실행에서 사용자가 제공한 것과 동일한 환경 변수 값을 제공합니다. 그러나 작업을 다시 시작하기 전에 사용자가 이 값을 변경할 수 있습니다.
- 시스템은 다시 시작된 작업 중에 작업이 실행하는 단계만을 기반으로 작업 성공을 평가합니다. 이전 실행에서의 실패는 다시 시작된 작업의 상태에 영향을 주지 않습니다.

어댑터 링크 단계를 포함하는 작업의 다시 시작에 대한 정보는 536 페이지의 『어댑터 링크에 대한 정보』를 참조하십시오.

### **브로드캐스트 다시 시작 단계**

실행 중에 브로드캐스트 단계로 표시된 단계가 이 단계가 실행될 서버마다 표시됩니다.

브로드캐스트 단계가 실패할 경우 모든 단계가 실패로 표시됩니다. 다시 시작할 때 다음을 수행하십시오.

1. 하나의 실패한 단계만 선택하십시오.
2. 동기화 명령 특성을 설정하십시오.
3. 다른 원하는 옵션을 설정하십시오.
4. 다시 시작 페이지에서 다시 시작을 클릭하십시오.

### **While 루프 다시 시작**

While 루프 유형 단계의 경우, BF\_ITERATION 시스템 변수 값에 따라 단계가 실패한 반복에서 단계가 다시 시작됩니다. 작업 플로우 예제는 다음과 같습니다.

1. 작업이 While 루프 유형 단계를 시작합니다.
2. 조건이 참으로 평가됩니다.
3. BF\_ITERATION이 1로 설정됩니다.
4. 명령과 인라인이 성공적으로 실행됩니다.
5. 단계 루프
6. 조건이 참으로 평가됩니다.
7. BF\_ITERATION이 2로 설정됩니다.
8. 명령 또는 인라인 실행 중에 작업이 중지됩니다.

위의 작업이 다시 시작되면 반복 2에서 다시 시작되고 단계에 대한 명령 및 인라인 실행을 시도합니다.

## 명세서 사용

시스템은 각 작업 후에 명세서(BOM)를 생성합니다. BOM은 작업의 단계에 대한 정보와 작업 결과로 발생한 파일 변경사항을 포함합니다. 새 작업의 콘텐츠를 이해하는데 도움이 되도록 작업 이용자(예: 품질 보증 부서)에게 BOM을 제공할 수 있습니다. BOM은 빌드 및 릴리스 프로세스에 대한 감사 솔루션의 역할을 할 수 있습니다. BOM을 사용하여 작업 콘텐츠의 전체 문서를 가져올 수 있습니다. BOM은 결과, 노트, 환경, 파일 목록 및 코드 변경사항을 포함할 수 있습니다. 이를 사용하여 엔터프라이즈 전반에서 빌드 상태를 비교하고 요약할 수 있습니다.

시스템은 각 작업에 대해 BOM을 자동으로 생성하지만, 단계에서 점 명령을 사용하여 시스템이 빌드 전/후에 파일 상태에 대한 추가 정보를 저장하게 할 수 있습니다.

### 명세서(BOM) 표시

완료된 빌드를 볼 때(작업 > 완료), 시스템은 기본적으로 단계 탭을 표시합니다. 명세서를 표시하려면 **BOM** 탭을 클릭하십시오.

범주 옆에 있는 +를 클릭하여 범주를 펼치십시오. 표시되는 실제 범주는 시스템 구성 방법과 프로젝트에 따라 다릅니다.

- 프로젝트 단계 범주는 모든 작업에 나타나며 이 작업에 대해 실행된 단계에 대한 정보를 제공합니다.
- 소스 변경사항 범주는 시스템에 소스 코드 어댑터가 포함되고 프로젝트에 어댑터로의 링크가 있는 경우에만 표시됩니다. 세부사항은 536 페이지의 『어댑터 및 작업 결과』의 내용을 참조하십시오. 어댑터를 구성할 때 소스 변경사항 범주의 형식과 이름이 변경될 수 있습니다.
- 기준선 및 체크포인트 섹션은 프로젝트에 .scan 명령이 포함된 경우에만 나타납니다.

### .scan 명령을 사용하여 기준선 및 체크포인트 추가

.scan 명령을 사용하여 BOM에 자세한 정보를 추가할 수 있습니다. .scan 명령이 실행될 때 시스템은 파일 상태에 대한 정보를 단계의 작업 디렉토리에 저장합니다. 이 절은 해당 정보의 사용 방법의 예제를 표시합니다. 443 페이지의 『.scan』에 대한 참조 정보도 참조하십시오.

이 명령 양식은 두 가지입니다.

#### .scan baseline

단계의 작업 디렉토리 트리에 각 파일에 대한 MD5 값과 함께 모든 파일 목록을 저장합니다. 시스템은 작업 BOM에 목록을 표시합니다. 일부 설정 단계를 수행하고 해당 파일 세트를 체크아웃한 후에 이 명령을 실행할 수 있습니다.

다. 하나의 프로젝트에 여러 개의 baseline 명령을 가질 수 있지만 .baseline 명령이 실행될 때 각 명령은 목록을 단계의 작업 디렉토리 상태로 다시 설정합니다.

### .scan checkpoint

프로젝트에서 마지막 .scan baseline 또는 .scan checkpoint 이후에 변경 및 삭제된 모든 새 파일 목록을 각 파일에 대한 MD5 값과 함께 저장합니다. .scan baseline 명령의 경우처럼 시스템은 BOM에 목록을 표시합니다. 프로젝트에서 첫 번째 .scan checkpoint 명령 전에 .scan baseline 명령을 실행해야 합니다. .scan baseline 명령에 우선하는 .scan checkpoint 명령은 무시됩니다.

다음 예제는 .scan baseline과 checkpoint 명령이 함께 작동하는 방법을 보여줍니다.

번호	단계	단계 후 파일	BOM 데이터
1	초기 파일 체크아웃	config.c execute.c	
2	.scan baseline	config.c execute.c	기준선: config.c execute.c
3	데이터 파일 추가	config.c execute.c data.txt	
4	.scan checkpoint	config.c execute.c data.txt	체크포인트 1: 추가된 data.txt
5	데이터 파일 계속 추가	config.c execute.c data.txt data2.txt data3.txt	
6	data.txt 삭제	config.c execute.c data2.txt data3.txt	
7	.scan checkpoint	config.c execute.c data2.txt data3.txt	체크포인트 2: 추가된 data2.txt, data3.txt 삭제된 data.txt

### BOM을 XML 파일로 내보내기

이 주제에서는 bf bomexport 명령 구문, 사용법 및 옵션 설명을 설명합니다.

Build Forge 명령은 Windows의 경우 Build Forge 설치 디렉토리 및 UNIX 및 Linux의 경우 `<bfinstall>/Platform` 디렉토리에 있습니다.

## 설명

작업 명세서(BOM)를 XML 파일로 내보내려면 `bfomexport` 명령을 사용하십시오. BOM 정보를 수집한 후에 `bfomexport`는 사용자가 지정하는 위치 및 파일 이름에 이를 저장합니다.

XML 파일에 저장할 BOM을 식별한 후에는 작업의 빌드 및 프로젝트를 식별해야 합니다.

`bfomexport` 명령은 관리 콘솔의 경우 Build Forge 설치 디렉토리에서, UNIX 또는 Linux에서는 `/Platform` 디렉토리에서 실행해야 합니다.

## 구문

```
bfomexport [-f filename] [-p projectID | -P projectName]
          [-b buildID | -t buildTag] [-L] [-H]
```

## 옵션

옵션	설명
-f filename	선택적 경로 및/또는 파일 이름입니다. 작업 BOM은 XML 형식으로 저장됩니다. 파일 이름이 제공되지 않은 경우, BOM은 표준 출력(stdout)에 기록됩니다. 경로 이름이 제공되지 않은 경우, 현재 작업 디렉토리가 사용됩니다.
-p projectID	작업의 프로젝트 ID(-l 옵션이 있는 <code>bfexport</code> 명령은 프로젝트 ID를 나열함)
-P projectName	프로젝트의 이름
-b buildID	빌드 ID
-t buildTag	빌드 태그 이름
-L	단계 로그를 포함
-H	도움말 메시지

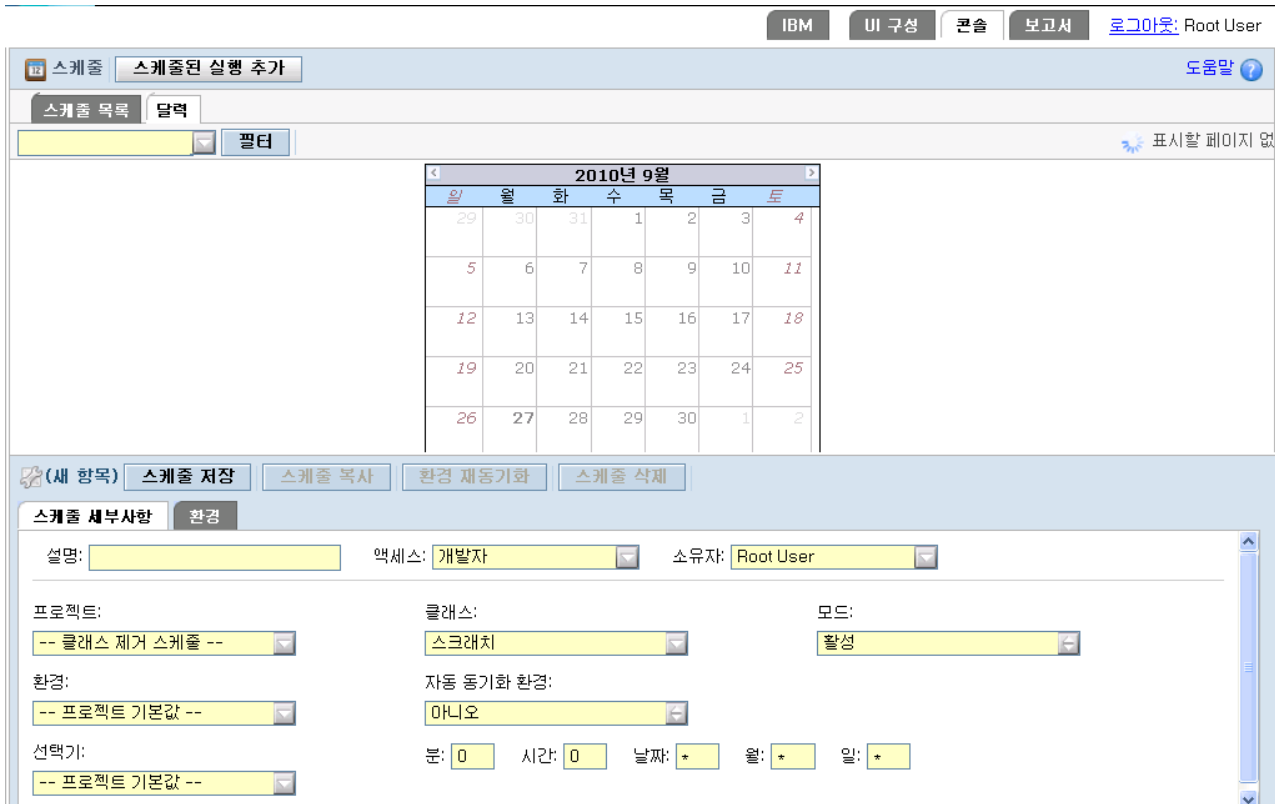
## 작업 스케줄링

스케줄 패널을 사용하여 프로젝트가 미래에 또는 정기적으로 반복되는 간격으로 실행되도록 스케줄하십시오. 예를 들어, 매시 또는 매일 프로젝트를 실행하도록 설정할 수 있습니다.

작업 스케줄을 보려면 스케줄을 클릭하십시오. 현재 월의 달력이 표시되며 그 아래에 스케줄 항목을 작성하고 수정하는 데 사용되는 패널이 표시됩니다.

달력의 각 날은 그 날에 스케줄된 프로젝트의 수를 표시합니다. 마우스를 날 위로 올리면 그 날에 스케줄된 작업의 이름 및 스케줄 매개변수가 표시됩니다.

두 개 이상의 프로젝트가 스케줄된 경우, 시스템은 프로젝트별로 달력을 필터링할 수 있도록 드롭다운 상자와 필터 단추를 표시합니다.



프로젝트를 스케줄러에 배치하려면 다음을 수행하십시오.

1. 왼쪽 메뉴에서 스케줄을 클릭하십시오.
2. 스케줄된 실행 추가를 클릭하십시오.
3. 설명에 스케줄 항목에 대한 설명을 입력하십시오.
4. 프로젝트 목록에서 프로젝트를 선택하십시오.
5. 프로젝트의 현재 스냅샷이 프로젝트 아래의 필드에 표시됩니다. 스케줄된 작업에 대해 다른 스냅샷이 실행되도록 하려면 목록에서 스냅샷을 선택하십시오.
6. 모드에 대해 활성(기본값), 비활성 또는 한 번을 선택하십시오. 한 번으로 설정된 작업은 다음 번에 시간 설정이 일치할 때 한 번만 실행됩니다. 활성으로 설정된 작업은 시간 설정이 일치할 때마다 실행됩니다.
7. 선택사항: 다음 프로젝트 특성을 대체할 수 있습니다. 여기에서 선택하는 설정은 스케줄된 작업의 이 인스턴스에만 적용됩니다.
  - 액세스: 지정되면 스케줄된 작업은 표시된 액세스 특성을 사용합니다.
  - 소유자: 지정되면 스케줄된 작업은 표시된 소유자가 수동으로 실행하는 것처럼 실행됩니다.

- **클래스:** 지정되면 스케줄된 작업은 표시된 클래스를 사용합니다.
- **환경:** 이 필드에는 명시적으로 설정되었는지 프로젝트 기본값으로 남아 있는지에 관계 없이 이 스케줄된 작업에 대한 환경의 사본이 작성됩니다. 환경이 지정된 경우에는 변수의 시작 값도 설정할 수 있습니다.

**중요사항:** 사본이 작성된 후 원본 환경의 변경사항 및 해당 변수는 스케줄된 작업에 대해 작성된 환경 사본에서 자동으로 업데이트되지 않습니다. 사본을 업데이트하여 원본 환경에 작성된 변경사항을 수동 또는 자동으로 반영할 수 있습니다. 수동 업데이트의 경우에는 **환경 다시 동기화**를 클릭하십시오. 자동 업데이트의 경우에는 **자동 동기화 환경**을 예로 설정하십시오. 자동 업데이트를 사용하면 환경은 스케줄된 작업이 실행될 때마다 업데이트됩니다. 그러나 환경 탭은 자동 동기화 환경이 아니므로 설정된 스케줄에 대해서만 사용 가능합니다.

- **선택기:** 지정되면 작업은 지정된 선택기를 사용합니다.
8. 시간 설정을 지정하십시오. 분, 시, 날짜, 월, 요일에 값을 입력하십시오. 사용할 값에 대해서는 464 페이지의 『스케줄 매개변수』의 내용을 참조하십시오.
  9. 스케줄 저장을 클릭하십시오.

다음 실행 열은 계산 중을 표시한 후 작업을 실행할 다음 스케줄된 시간을 표시합니다.

스케줄이 작성되면 스케줄 이름은 달력에 표시됩니다. 옆에 있는 아이콘은 모드를 표시합니다. 아이콘을 클릭하여 모드를 변경할 수 있습니다. 클릭할 때마다 변경됩니다.

- 초록색 원: 활성화
- 파란색 원: 한 번
- 빨간색 원: 비활성

스케줄된 작업이 실행하려고 시도할 때 시스템은 작업 큐, 하드 실행 한계 시스템 설정 및 잠재적으로 프로젝트의 실행 한계 특성을 검사합니다. 스케줄이 활성화될 때 하나 이상의 프로젝트 작업이 이미 실행 중인 경우 시스템 동작은 다음 표에서 설명하는 것처럼 하드 실행 한계 시스템 설정에 따라 다릅니다.

하드 실행 한계 값	시스템이 실행 한계 값을 검사합니까?	시스템이 작업을 실행합니까?
예	예	실행 중인 작업의 수가 실행 한계 값보다 적은 경우 예  실행 중인 작업의 수가 실행 한계 값과 같은 경우 아니오
아니오	아니오	예

참고: 이전 실행이 완료되지 않은 경우 실행을 건너뛰려면 실행 한계 값을 1로 설정하고 하드 실행 한계를 예로 설정하십시오.

## 스케줄 매개변수

이 절에서는 작업 실행 시기를 지정하는 데 사용할 수 있는 매개변수에 대해 설명합니다.

일련의 필드를 사용하여 작업이 실행되는 시기를 지정합니다.

필드	설명	범위
분	분 수	0-59
시간	하루 중 시간	0-23
날짜(월 중 일)	월의 날짜	1-31
월	연도의 월	1-12
요일	주의 요일(일요일 = 0).	0-6

필드에 입력하는 숫자 값은 다음과 같이 표시할 수 있습니다.

- 범위의 올바른 값을 모두 표시하려면 별표(\*)를 사용하십시오. 별표 뒤에 슬래시(/) 및 단계 값이 올 수 있습니다. 예를 들어, 시간 필드의 \*/2 값은 2시간마다 작업을 실행합니다.

참고: 별표는 날짜 및 일 수 필드에서 특별히 해석됩니다. UNIX/Linux 크론 기능에서 해당 의미와 일치합니다. 한 필드에 리터럴 값이 있고 다른 필드에 별표가 있는 경우, 리터럴 값에 대한 빈도만 사용됩니다. 예를 들어 월이 \*이고 날짜가 \*이고 일 수가 1이면, 월요일마다 실행이 일어납니다. 아래의 일/날짜 예제도 참조하십시오.

- 하이픈으로 분리된 숫자 범위를 사용하십시오. 예를 들어, 시간 필드의 8-11은 8, 9, 10 및 11시를 지정합니다. 범위 뒤에 포워드 슬래시(/) 및 단계 값이 올 수 있습니다. 예를 들어, 시간 필드의 0-23/2는 2시간마다 작업을 실행합니다.
- 쉼표로 구분된 숫자(또는 범위) 세트의 쉼표로 구분된 목록을 사용하십시오. 예: 1, 2, 3-5, 8.

스케줄은 필드에 지정된 값에서 생성됩니다. 예제:



값 원하는 스케줄	분	시간	날짜	월	일
매일 오후 5시에 작업 실행	0	17	*	*	*
매주 월요일마다 오후 4시 30분 에 작업 실행	30	16	*	*	1
매일 30분마다 작업 실행(주말 은 건너뛴)	*/30	*	*	*	1-5
격일로 오전 12시 30분에 작 업 실행	30	0	*/2	*	*

일 수 및 날짜 필드는 특별한 방법으로 별표를 사용합니다.

- 한 필드에 별표 값이 있고 다른 필드에 리터럴 값이 있는 경우, 리터럴 설정을 가지고 있는 필드에 따라 작업이 실행됩니다.
- 두 필드에 모두 별표 이외의 값이 있는 경우, 어느 한 조건이 발생하면 작업이 실행됩니다.
- 두 필드에 모두 별표 값이 있는 경우, 매일 작업이 실행됩니다.

일 수/날짜 예제:

값 원하는 스케줄	분	시간	날짜	월	일
매일 1일 오전 1시 01분에 작업 실행. 날짜는 리터럴 값을 사용 하고 일 수는 별표를 사용합니 다.	1	1	1	*	*
해당 월의 매주 월요일 오전 1시 01분에 작업 실행. 날짜는 별표 를 사용하고 일 수는 리터럴 값 을 사용합니다.	1	1	*	*	1
해당 월의 매주 월요일 및 매월 1일(요일에 관계 없음) 오전 1시 01분에 작업 실행. 날짜 및 일 수 모두 리터럴 값을 사용합니 다.	1	1	1	*	1

값					
원하는 스케줄	분	시간	날짜	월	일
매일 오전 1시 01분에 작업 실행. 날짜 및 일 수 모두 별표 값을 사용합니다.	1	1	*	*	*

## 작업 클래스 제거 스케줄링

작업 클래스 스케줄을 작성하여 시스템의 이전 작업 제거 수행 시기를 제어할 수 있습니다. 프로젝트 실행을 위한 스케줄을 작성할 때와 마찬가지로 스케줄을 작성하되, 프로젝트 대신 『클래스 제거 스케줄』 옵션을 선택하십시오. 이를 수행할 때 시스템은 스케줄에서 사용자가 선택하는 시간에 사용자가 선택하는 클래스를 제거할 작업을 확인합니다. 규정하는 각 작업에 대해 시스템은 제거 작업을 작성하고 대기 큐에 작업을 배치합니다.

기본적으로, 시스템은 제거 확인 시간 시스템 설정으로 설정된 간격(기본값은 매 15분)으로 자동 삭제 규칙을 정의하는 클래스 특성을 기반으로 제거되어야 하는 작업을 확인합니다. 이 동작으로 인해 제거는 일반 작업과 시스템 자원 경합을 벌일 수 있습니다.

작업의 클래스 스케줄을 작성하는 경우, 시스템은 스케줄이 활성화될 때 제거할 작업만 확인합니다. 특정 클래스에 대해 스케줄이 없는 경우, 시스템은 해당 클래스의 작업에 기본 동작을 사용합니다. 모든 제거를 특정 시간으로 제한하려면, 각 클래스에 대해 최소 하나의 스케줄을 작성해야 합니다.

제거 스케줄을 정의하려면 다음을 수행하십시오.

1. 일반적으로 스케줄을 작성하되, 프로젝트 필드에서 『클래스 제거 스케줄』을 선택하십시오.
2. 클래스 필드에서 클래스를 선택하십시오.

## 작업 관리

작업을 잠그고 아카이브하고 삭제할 수 있습니다.

### 작업 잠금

프로젝트가 자동으로 삭제되지 않도록 프로젝트를 잠글 수 있습니다.

작업을 잠그려면 작업 패널에서 완료됨 탭을 클릭하고 빌드를 선택하고 잠금을 클릭하십시오.

작업을 잠금 해제하려면 작업 패널에서 잠금 탭을 클릭하고 잠긴 빌드를 선택하고 잠금 해제를 클릭하십시오.

실행에 대한 클래스 제거 특성이 삭제를 요청할 때처럼 시스템이 잠긴 작업을 자동으로 제거하지 않습니다. 잠긴 작업은 수동으로 삭제할 수 있습니다.

잠금 탭을 클릭하여 잠긴 상태의 작업을 보십시오. 해당 프로젝트는 완료 페이지에 표시되지 않습니다.

## 작업 삭제

다음 주제는 작업을 삭제하는 몇 가지 방법에 대해 설명합니다.

### 완료 탭에서 작업 삭제

작업 패널의 완료 탭에서 작업을 수동으로 삭제할 수 있습니다.

이 목록에서 하나 이상의 작업을 수동으로 삭제할 수 있습니다. 이를 수행할 때 시스템은 각 실행 클래스를 기반으로 제거 작업을 실행합니다. 프로세스는 클래스 특성에 의해 자동 삭제가 트리거된 경우와 동일합니다. 시스템은 작업을 아카이브하고(완전히 삭제되지 않은 경우) 해당 클래스의 작업에 지정된 데이터를 작성합니다. 370 페이지의 『클래스』의 내용을 참조하십시오.

작업을 삭제하려면 다음을 수행하십시오.

1. 작업 > 완료 페이지를 선택하여 완료된 실행 목록을 표시하십시오.
2. 테이블 오른쪽 끝에서 하나 이상의 선택란을 클릭하여 삭제할 빌드를 선택하십시오.
3. 목록 맨 아래의 목록 상자에서 제거 옵션을 선택하십시오.
4. 이동 단추를 클릭하십시오.

출력 파일은 삭제하지만 콘솔 데이터를 보유하도록 클래스가 설정된 경우, 완료 목록에서 실행을 삭제하면 출력 파일을 삭제하고 완료 목록에서 아카이브 목록으로 작업 항목이 이동됩니다.

### 아카이브 목록에서 완전히 작업 삭제

작업 > 아카이브를 선택하여 아카이브 목록을 표시하십시오. 이 목록은 파일이 삭제된 작업에 대한 데이터를 표시합니다. 작업 > 완료 목록에서 수행하는 방식과 같이 여기서 작업을 삭제할 수 있습니다. 아카이브 목록에서 작업을 삭제하면 데이터베이스에서 모든 작업 추적을 제거하고 애플리케이션별로 보고된 통계에서 삭제합니다.

### 작업 자동 삭제

작업의 클래스 특성이 작업을 삭제해야 한다고 판별하면 시스템은 자동으로 작업을 삭제합니다. 이 기능을 사용하여 데이터가 누적되지 않도록 하고 작업 그룹을 삭제할 수 있습니다.

클래스 스케줄을 작성하는 경우, 시스템은 스케줄이 활성화될 때 제거할 작업만 확인합니다. 466 페이지의 『작업 클래스 제거 스케줄링』의 내용을 참조하십시오.

삭제하려고 할 때 작업이 삭제되도록 하려면 작업을 실행하거나 스케줄하기 전에 다음 설정을 확인하십시오.

1. 일정 일이 경과하거나 일정 수의 작업이 누적된 후 또는 두 경우 모두에 시스템이 작업을 삭제할 수 있도록 하나 이상의 클래스의 삭제 특성을 설정하십시오.
2. 작업 중인 프로젝트의 클래스 특성을 해당 클래스로 설정하십시오.

많은 작업을 생성한 후 이를 삭제하려는 경우, 관련 클래스의 삭제 특성을 일시적으로 변경할 수 있습니다. 또는 완료 탭에서 여러 개의 빌드를 선택하고 삭제하십시오. (각 작업 옆에 있는 선택란을 클릭하여 작업을 선택한 후 제거 단추를 클릭하십시오.)

예를 들어, 전날에 프로덕션 작업을 많이 생성한 경우 다음 프로세스를 사용하여 이를 삭제하십시오.

1. 프로덕션 클래스의 현재 설정을 적어 두십시오.
2. 프로덕션 클래스의 일 특성을 1로 변경한 후 클래스 저장 단추를 클릭하십시오. 15 분이 경과한 후에 시스템은 하루 전날의 작업을 삭제하기 시작합니다.
3. 작업이 삭제되면 프로덕션 클래스의 특성을 원래의 설정으로 다시 변경하십시오.

## 작업용 작업 디렉토리

시스템은 작업용 작업 디렉토리를 생성하여 각 실행이 작업할 분리된 영역을 갖도록 합니다. 시스템은 서버 경로, 프로젝트 이름 및 태그에 제공된 값을 사용하여 작업 디렉토리 이름을 작성합니다.

시스템은 명령을 실행할 때 단계에 대해 지정된 디렉토리에서 명령을 시작합니다. 기본적으로 해당 디렉토리는 작업용 작업 디렉토리지만 서버의 경로 특성에 관하여 일부 다른 디렉토리를 지정할 수도 있습니다. 이 절의 주제에서는 경로 및 디렉토리 작성 방법을 설명합니다.

**참고:** 시스템은 프로젝트를 실행할 때 프로젝트 디렉토리(아직 없는 경우)와 작업 디렉토리를 생성합니다. 시스템은 서버 디렉토리(서버의 경로 특성에 지정됨) 또는 단계 디렉토리(단계의 디렉토리 특성)를 생성하지 않습니다.

## 작업용 작업 디렉토리 이름

다음 예제는 단일 서버에서 발생하는 작업에서 시스템이 몇 개의 값을 사용하여 작업 디렉토리를 생성하는 방법을 보여줍니다.

시스템값	작성된 디렉토리
서버의 경로 필드: C:\BuildForge 프로젝트 이름: My Project 태그: Job 5	C:\BuildForge\My_Project\Job_5\ 시스템은 경로의 굵은체 부분만 작성합니다. 서버 디렉토리를 작성한 후에 프로젝트를 실행해야 하며, 그렇지 않으면 실패합니다.

**참고:** 시스템은 프로젝트 디렉토리를 작성할 때 **올바르지 않은 상대 디렉토리 문자** 시스템 설정에 지정된 문자를 밑줄 문자로 변경합니다. 기본적으로, 설정은 공백과 백틱(backtick) 문자를 포함하므로 My Project라는 프로젝트는 My\_Project라는 프로젝트 디렉토리를 수신합니다.

하나의 작업이 두 개 이상의 서버에서 발생하는 경우, 시스템은 각 서버에서 작업 디렉토리를 작성합니다. 프로젝트의 각 단계마다 서버를 다르게 지정할 수 있으므로 시스템은 잠재적으로 많은 디렉토리를 작성할 수 있습니다. 다음 예제는 두 개의 서버를 사용하는 프로젝트를 설명합니다.

시스템값	작성된 디렉토리
프로젝트 서버: 경로 값이 C:\BuildForge인 ServerA 프로젝트의 세 번째 단계는 경로 값이 C:\deployments인 ServerB를 지정합니다(기타 모든 단계는 기본(프로젝트) 서버인 ServerA를 사용함). 프로젝트 이름: My Project 태그: Job_6	ServerA에서: C:\BuildForge\My_Project\Job_6\ ServerB에서: C:\deployments\My_Project\Job_6\ 시스템은 경로의 굵은체 부분만 작성합니다.

위 예제에서, C:\deployments\My\_Project\Job\_6 디렉토리를 사용하는 세 번째 단계의 경우를 제외하고 단계의 출력 파일이 C:\BuildForge\My\_Project\Job\_6 디렉토리에 기본적으로 작성될 것으로 예상할 수 있습니다.

## 단계의 디렉토리 경로 생성

시스템은 단계를 실행할 때 작업에 대해 생성한 디렉토리에서 시작하거나 단계의 경로 옵션이 절대로 설정된 경우 프로젝트 및 태그 디렉토리를 무시할 수 있습니다.

- 단계의 경로 설정이 절대인 경우 시스템은 서버 경로 및 단계의 디렉토리 필드를 함께 추가하여 단계에 대한 경로를 구성합니다. 디렉토리 필드의 값은 서버 작업 디렉토리에 대해 상대적인 경로입니다.

단계 값	명령의 결과 경로
서버의 경로 필드: C:/BuildForge 단계의 디렉토리 필드: /bin 단계의 경로 설정: 절대	C:\BuildForge\bin  서버 디렉토리에 있는 디렉토리에 액세스하려면 이 양식을 사용하십시오.
서버의 경로 필드: C:/BuildForge 단계의 디렉토리 필드: /(기본값) 단계의 경로 설정: 절대	C:\BuildForge
서버의 경로 필드: C:/BuildForge 단계의 디렉토리 필드: C:/temp 단계의 경로 설정: 절대	C:\BuildForge\C:\temp  (이 예제는 오류가 발생하고, 단계는 실패합니다.)

**참고:** 백슬래시나 슬래시를 사용하여 경로 값을 입력할 수 있습니다. 시스템은 백슬래시를 사용하여 경로 값을 저장하고 Windows<sup>®</sup> 컴퓨터에서 필요에 따라 이를 포워드 슬래시로 변경합니다.

**중요사항:** 명령에 대한 경로를 루트 디렉토리로 설정하면 시스템 파일이 비의도적으로 삭제 또는 수정될 수 있습니다. 가능한 경우 다른 디렉토리에서 명령을 실행하십시오. 그러나 시스템 관리 태스크를 자동화하고 경로가 루트 디렉토리여야 하는 경우 서버의 경로를 루트 디렉토리로, 단계의 디렉토리를 ../로, 단계의 경로를 절대로 설정하십시오.

- 단계의 경로 옵션이 상대로 설정된 경우 시스템은 서버 경로, 프로젝트 이름, 태그 및 단계의 디렉토리 필드를 함께 추가하여 단계에 대한 경로를 구성합니다. 디렉토리 필드의 값은 작업의 작업 디렉토리에 상대적인 경로가 됩니다.

단계 값	명령의 결과 경로
서버의 경로 필드: C:/BuildForge  프로젝트 이름: My Project  태그: Job_5  단계의 디렉토리 필드: /bin  단계의 경로 설정: 상대	C:\BuildForge\My_Project\Job_5\bin  경로의 굵은체 부분은 시스템에 의해 생성됩니다(아직 없는 경우).  <b>참고:</b> 시스템은 프로젝트 디렉토리를 작성할 때 올바른지 않은 상대 디렉토리 문자 시스템 설정에 지정된 문자를 밑줄 문자로 변경합니다. 기본적으로, 설정은 공백과 백틱(backtick) 문자를 포함하므로 『My Project』라는 프로젝트는 『My_Project』라는 프로젝트 디렉토리를 수신합니다.

서버의 경로 필드 또는 단계의 디렉토리 필드에 지정된 디렉토리가 존재하지 않는 경우, 단계는 실패합니다. 시스템은 이러한 디렉토리를 작성하지 않습니다. 단계의 디렉토리 필드가 지정한 경로 부분은 이전 단계에 의해 프로젝트 중에 명시적으로 작성되어야 합니다.

일반적으로 프로젝트에서 초기 단계는 소스 코드 제어에서 디렉토리 트리를 체크아웃하고, 후속 단계는 해당 디렉토리에 작용합니다.

사용자가 새 단계를 추가할 때 시스템은 경로에 대해 사용자가 선택한 마지막 설정을 기억하며, 새 단계에서 해당 설정을 기본값으로 사용합니다.

## 경로의 슬래시

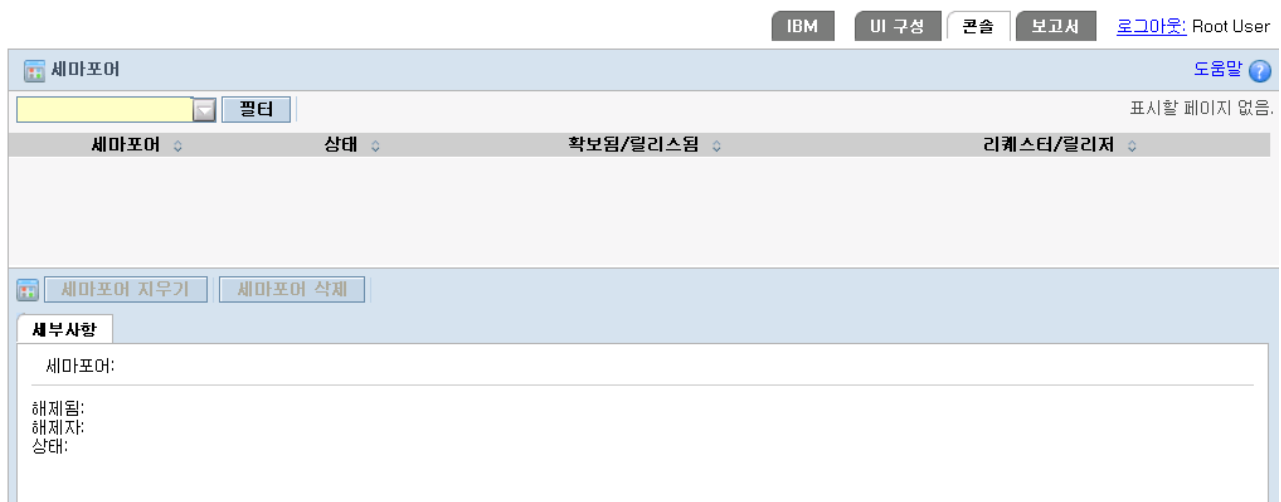
경로 또는 디렉토리 필드에 경로를 입력할 때, 시스템은 모든 백슬래시를 슬래시로 변환합니다. 시스템이 실제 경로를 작성할 때 사용 중인 서버가 Windows<sup>®</sup> 컴퓨터인 경우 시스템은 모든 포워드 슬래시를 백슬래시로 변환합니다. 따라서 경로에서 두 슬래시 유형 중 하나를 사용할 수 있습니다.

참고: 시스템은 단계의 명령 필드에 있는 슬래시를 수정하지 않습니다. 대상 서버에 필요에 따라 백슬래시 또는 슬래시를 사용하십시오.

## 세마포어

세마포어는 시스템에서 상호 독점(뮤텍스) 자원을 설정하는 글로벌 신호 플래그입니다. 일부 프로세스가 다른 프로세스의 완료를 기다리게 하려면 세마포어를 사용하십시오.

작업 > 세마포어를 사용하여 사용 중인 작업 세마포어를 볼 수 있습니다. 또한 세마포어를 지울 수도 있는데 이는 정지되거나 취소된 작업이 세마포어를 해제하는 데 실패하는 경우 필요합니다.



점 명령 쌍(.semget 및 .semput 명령)을 통해 세마포어를 구현하십시오. .semget 명령을 사용하여 레이블을 『점유』하십시오. 단계가 레이블을 가져온 후에는 동일한 레이블을 가져오려고 시도하는 프로젝트의 기타 단계는 원래의 요청자가 레이블을 해제할 때까지 기다려야 합니다.

**참고:** .semget 명령을 포함하는 단계는 세마포어가 해제될 때까지 대기합니다. 작업이 실패하여 세마포어가 활성 상태인 경우, 세마포어를 사용하는 모든 작업이 다시 실행되도록 하려면 세마포어를 수동으로 지워야 합니다.

예를 들어, 프린터 드라이버를 작성하는 프로그램이 있고 임의의 시점에서 하나의 프로세스만 프로그램을 사용하게 한다고 가정하십시오. 프로그램을 호출하는 각 프로젝트에서 다음 명령행을 사용하여 세 개의 단계를 설정하십시오.

단계	명령행
세마포어 가져오기	<code>.semget \$BF_PROJECTNAME_PHYS</code>
드라이버 작성 프로그램 실행	<code>printdrivermaker.exe windows</code>
세마포어 해제	<code>.semput \$BF_PROJECTNAME_PHYS</code>

조직의 주요 자원(예: 로드량이 많은 서버 또는 단일 사용자 라이선스가 있는 소프트웨어 프로그램)에 대해 세마포어를 설정할 수 있습니다. 보호할 자원을 사용하는 모든 단계를 .semget 및 .semput 명령으로 래핑해야 합니다.

세마포어는 다음 규칙을 준수합니다.

- 세마포어 점 명령은 사용될 때 단계의 유일한 점 명령이어야 합니다.
- 시스템은 레이블을 현상태대로 사용합니다. 레이블 구문 분석기와 혼동할 수 있으므로 레이블에서 특수 문자 또는 뒤 공백을 사용하지 마십시오.
- 세마포어는 글로벌로 적용되며, 스레드 블록 이외에 개별 프로젝트를 동기화하는 데 사용할 수 있습니다.
- 세마포어는 세마포어에 처음 액세스할 때 작성됩니다.
- 프로젝트가 확보한 세마포어는 해당 프로젝트가 종료되면 자동으로 해제됩니다.
- 두 단계가 동시에 하나의 세마포어를 요청하는 경우, 단계가 잠금을 가져오는 것과 관련하여 보장된 순서는 없습니다.

## 수동으로 세마포어 지우기

시스템은 프로젝트가 종료될 때 해당 프로젝트가 작성한 모든 세마포어를 자동으로 해제합니다. 그러나, 비정상 프로젝트 종료로 인해 세마포어가 릴리스되지 않는 경우 수동으로 지울 수 있습니다.

1. 작업 > 세마포어를 선택하여 세마포어 목록을 표시하십시오.



2. 지우려는 세마포어를 선택하십시오.
3. 지우기를 클릭하십시오.



---

## 제 22 장 보고서에 대한 작업

이 주제는 Rational Build Forge에 제공되는 기본 보고서에 대해 설명합니다. 또한 별도 라이선스 보고 도구인 빠른 보고서를 사용하여 보고서를 작성 및 실행하는 방법에 대해 설명합니다.

---

### 보고서 정보

보고서 범주는 성능 및 조회 패널에서 액세스하는 기본 보고서 범주와 빠른 보고서 도구를 사용하여 작성하는 보고서 범주로 분류됩니다.


시작하려면 오른쪽 상단 모서리의 보고서 탭을 클릭하여 다음과 같은 보고를 위한 선택 사항을 표시하십시오.

- 표준 보고서를 보려면 성능 패널을 선택하십시오.
- 사전 정의된 조회 보고서를 보고 실행하려면 조회 패널을 선택하십시오.
- 사용자 고유의 보고서를 작성하려면 빠른 보고서 패널을 선택하십시오. 빠른 보고서는 Rational Build Forge의 라이선스 옵션입니다.

### 성능 및 조회 패널

성능 패널은 Build Forge 데이터를 사용하여 여러 표준 보고서를 작성합니다. 성능 보고서를 열면 결과가 자동으로 업데이트되어 항상 현재 데이터를 볼 수 있습니다.

성능 패널에는 또한 사전 정의된 조회 보고서가 포함된 조회 패널이 있습니다. 이러한

보고서를 실행하려면 필수 데이터를 제공한 후 빠른 시작 아이콘()을 클릭하십시오. 조회 보고서는 현재 데이터에 대해 실행됩니다. 보고서를 다시 실행하면 최신 데이터를 기반으로 하는 결과가 표시됩니다.

성능 및 조회 패널의 보고서는 수정할 수 없습니다. 디자인 특성상 필드를 수정하거나 보고서 프리젠테이션을 변경할 수 없습니다.

### 빠른 보고서 도구

빠른 보고서는 Eclipse 기반 개방형 소스 보고 시스템인 BIRT를 기반으로 하는 보고 도구입니다. 빠른 보고서를 사용하면 사용자 정의 보고서 디자인을 작성할 수 있습니다. 제공된 보고서 유형을 시작점으로 사용한 다음 원하는 보고서 필드를 선택할 수 있습니다. 모든 보고서 유형은 Rational Build Forge 데이터베이스의 데이터를 해당 데이터 소스로 사용합니다.

빠른 보고서에서 디자인하는 모든 보고서의 경우 표 또는 차트 같은 여러 가지 보고서 형식에서 선택할 수 있습니다. 또한 그룹화 및 필터링 기능을 사용하여 보고서 프리젠테이션을 제어할 수 있습니다.

## 보고서 출력에 데이터를 표시하기 위한 전제조건

샘플 보고서를 포함하는 모든 보고서에는 Build Forge 데이터베이스의 데이터가 표시됩니다. 데이터베이스를 채우려면 먼저 프로젝트를 작성하고 작업을 실행해야 합니다.

**참고:** 데이터베이스에 데이터가 포함되기 전에 보고서를 작성하고 실행하면 보고서 결과가 비어 있습니다. 또한 보고서 오류가 발생할 수 있습니다.

## 보고서 데이터 표시를 위한 액세스 권한

보고서를 실행하는 사용자의 액세스 권한은 보고서 결과에 포함 및 표시되는 프로젝트 데이터를 결정합니다.

사용자의 프로젝트 액세스 권한은 다음과 같이 보고서 결과에 포함되는 프로젝트 데이터를 결정합니다.

- 성능 패널의 표준 보고서의 경우 사용된 권한은 보고서 페이지를 여는 사용자의 권한입니다.
- 조회 패널의 사전 정의된 보고서의 경우 사용된 권한은 보고서를 실행하는 사용자의 권한입니다.
- 빠른 보고서의 경우 사용된 권한은 보고서 디자이너의 권한이 아닌 보고서를 실행하는 사용자의 권한입니다.

다른 액세스 권한이 있는 사용자는 보고서 디자인을 수정하고 보고서를 실행할 수 있습니다. 483 페이지의 『빠른 보고서에 필요한 보고서 그룹 권한』의 내용을 참조하십시오.

## 기본 보고서에 대한 결과를 CSV 파일로 내보내기

여러 기본 보고서는 보고서 결과를 CSV 파일로 내보낼 수 있습니다. 다른 보고서 애플리케이션에 결과를 저장하거나 결과를 가져오려면 이 기능을 사용하십시오.

CSV에 저장 링크가 있는 기본 보고서의 경우 보고서 결과를 내보낼 수 있습니다.

**참고:** 이 기능은 빠른 보고서에는 제공되지 않습니다.

보고서 결과를 CSV 파일 형식으로 내보내려면 다음을 수행하십시오.

1. 보고서를 실행하여 결과를 생성하십시오.
  - 분석 보고서 링크를 선택하여 단계 및 서버 성능 보고서를 여십시오(보고서 > 성능).
  - 조회 패널에서 보고서를 실행하십시오(보고서 > 조회).

2. 보고서 결과 페이지에서 **CSV**에 저장을 클릭하십시오.
3. 파일 다운로드 대화 상자에서 열기 또는 저장을 클릭하십시오.
  - CSV 파일 형식을 지원하는 애플리케이션(예: Microsoft Excel 또는 선호하는 텍스트 기반 편집기)을 사용하여 CSV 파일을 여십시오.
  - 로컬 워크스테이션 또는 네트워크의 다른 위치에 파일을 저장하십시오.

---

## 성능에 대한 표준 보고서

표준 보고서에는 표준 보고서 페이지를 열 때 현재 데이터를 기반으로 한 결과가 표시됩니다.

성능 패널은 현재 프로젝트 및 작업 데이터를 사용하여 여러 표준 보고서를 작성합니다.

- 모든 프로젝트에 대한 작업 성능 보고서
- 프로젝트별 작업 지속 기간 보고서
- 프로젝트별 단계 및 서버 분석 보고서

### 프로젝트의 작업 성능 통계 보기

성능 패널에는 모든 프로젝트에 대한 상위 레벨 작업 통계를 요약하는 작업 성능 보고서가 포함됩니다.

- 마지막 작업의 경우 보고서는 완료 날짜-시간과 작업 지속 기간을 보여줍니다.
- 총 작업 수의 경우 통계에는 패스, 실패 또는 경고와 함께 패스된 작업의 수가 포함됩니다.

데이터 소스: Rational Build Forge 데이터베이스

보고서 형식: 표

표시 옵션: 목록에 표시되는 프로젝트를 제어하는 필터를 작성합니다.

필드:

- 신뢰 구간: 작업 평균 지속 기간 양쪽(+/-)의 값(초) 범위. 이 값 범위에 다음 작업 지속 기간이 포함될 수 있는 가능성은 95%입니다. 예를 들어, 신뢰 구간이 5.88초인 경우 작업 평균 지속 기간 양쪽에서 다음 작업 지속 기간을 포함할 수 있는 가능성이 95%인 값 범위는 5.88초입니다.

작업 성능 통계 보고서를 보려면 다음을 수행하십시오.

1. 보고서 탭을 여십시오.
2. 성능 패널을 클릭하십시오.

모든 프로젝트에 대한 작업 성능 보고서가 결과표로 표시됩니다.

## 프로젝트의 작업 지속 시간 보기

성능 패널에는 모든 프로젝트 작업의 지속 기간을 그래프로 보여주는 작업 지속 기간 보고서가 포함됩니다.

데이터 소스: Rational Build Forge 데이터베이스

보고서 형식: 그래프

프로젝트 작업 지속 기간 보고서를 보려면 다음을 수행하십시오.

1. 보고서 탭을 여십시오.
2. 성능 패널을 클릭하십시오.
3. 목록에서 프로젝트 이름을 클릭하십시오.

프로젝트 작업 지속 기간 보고서는 작업 시간 그래프로 표시됩니다.

## 프로젝트별로 단계 및 서버 성능 보기

성능 패널에는 프로젝트별 단계 및 서버 분석 보고서가 포함됩니다. 또한 프로젝트의 단계에 대한 중요 경로 분석 보고서가 포함됩니다.

단계 및 서버 분석 보고서에는 단계를 실행한 서버와 다른 실행(예를 들어, 가장 빠른 실행 또는 가장 느린 실행)에 대한 단계를 수행하는 데 필요한 시간이 각 프로젝트 단계마다 나열됩니다. 단계 시간은 패스, 실패 또는 경고로 범주화됩니다. 범주는 시간을 잘못 비교하지 않도록 예방합니다(예: 패스 시간과 실패 시간).

단계에 대한 중요 경로 분석 보고서에는 패스된 작업의 주요 메트릭이 나열됩니다.

데이터 소스: Rational Build Forge 데이터베이스

보고서 형식: 표

키: -/-/ 키는 다음 예와 같이 두 가지 단계 정보, 즉, 단계 결과와 단계 지속 기간(초)을 제공합니다.

14/-/ 패스/경고/실패	단계가 패스했으며, 총 실행 시간은 14초입니다.
-/1/ 패스/경고/실패	단계가 실패했으며, 총 실행 시간은 1초입니다.
-/1/ 패스/경고/실패	단계가 경고와 함께 패스했으며, 총 실행 시간은 1초입니다.

필드:

- 패스 단계: 모든 작업에 대한 단계 패스 횟수.

- 평균 실행: 패스된 작업의 평균 작업 지속 기간.
- 신뢰 구간: 작업 평균 지속 기간 양쪽(+/-)의 값(초) 범위. 이 값 범위에 다음 작업 지속 기간이 포함될 수 있는 가능성은 95%입니다. 예를 들어, 신뢰 구간이 5.88초인 경우 작업 평균 지속 기간 양쪽에서 다음 작업 지속 기간을 포함할 수 있는 가능성이 95%인 값 범위는 5.88초입니다.
- 가장 긴 지속 기간 가능성: 다음 작업이 패스된 작업의 가장 긴 지속 기간과 일치할 가능성.
- 가장 짧은 지속 기간 가능성: 다음 작업이 패스된 작업의 가장 짧은 지속 기간과 일치할 가능성.

분석 보고서를 보려면 다음을 수행하십시오.

1. 보고서 탭을 여십시오.
2. 성능 패널을 클릭하십시오.
3. 목록에서 프로젝트를 선택한 다음 분석을 클릭하십시오.

단계 및 서버에 대한 분석 보고서와 중요 경로 분석 보고서는 결과표로 표시됩니다.

---

## 조회에 대한 사전 정의 보고서

현재 프로젝트 및 작업 데이터를 사용하여 사전 정의된 보고서를 실행하는 데 필요한 기타 데이터 또는 날짜 범위를 제공합니다.

조회 패널은 현재 데이터를 사용하여 다음과 같은 여러 사전 정의된 보고서를 작성합니다.

- 프로젝트 및 단계의 선택기 활용도 히스토리
- 모든 서버의 현재 서버 Manifest
- 작업 결과 패스/실패/경고 히스토리
- 날짜 범위 내 작업의 서버 활용도 히스토리
- MD5 값 기반의 작업 파일 검색

### 선택기 활용도 히스토리 보기

모든 프로젝트의 선택기 활용도를 보여주는 보고서입니다.


조회 패널에는 모든 프로젝트의 선택기 활용도를 표시하는 보고서가 포함됩니다. 이 보고서를 사용하면 프로젝트와 단계에 지정된 선택기를 확인할 수 있습니다. 이 보고서는 특히 선택기를 사용하여 프로젝트 또는 단계를 실행할 서버를 동적으로 선택할 때 유용합니다.

데이터 소스: Rational Build Forge 데이터베이스

보고서 형식: 표

표시 옵션: 트리 계층 구조를 펼치고 목록의 모든 보고서 결과를 표시하려면 보고서 출력 표시를 선택하십시오.

단계 및 작업에 대한 서버 활용도 보고서를 보려면 다음을 수행하십시오.

1. 보고서 탭을 여십시오.
2. 조회 패널을 클릭하십시오.
3. 프로젝트 선택기 및 단계 서버 보고서를 보려면 빠른 시작 아이콘()을 클릭하십시오.

선택기 활용도 보고서는 결과표로 표시됩니다.

## 서버별로 현재 서버 Manifest 보기

모든 서버의 서버 Manifest 특성과 값을 나열하는 보고서입니다.

조회 패널에는 모든 서버의 서버 Manifest 특성과 값을 나열하는 보고서가 포함됩니다. 이 보고서를 사용하면 서버 특성을 비교하거나 동일한 선택기가 선택한 서버를 식별하거나 서버에 필요한 특성을 식별할 수 있습니다.


이 보고서에는 서버에 지정된 콜렉터가 서버에 지정한 모든 Manifest 특성이 표시됩니다. 모든 서버에 자동으로 지정되는 특수 Manifest 특성을 포함하려면 **BF\_** 특성 표시를 선택하십시오.

데이터 소스: Rational Build Forge 데이터베이스

보고서 형식: 표

표시 옵션: 트리 계층 구조를 펼치고 목록의 모든 보고서 결과를 표시하려면 보고서 출력 표시를 선택하십시오.

서버의 현재 서버 Manifest 보고서를 보려면 다음을 수행하십시오.

1. 보고서 탭을 여십시오.
2. 조회 패널을 클릭하십시오.
3. (선택적) 보고서 결과에 특수 Manifest 특성을 포함하려면 **BF\_** 특성 표시를 선택하십시오.
4. 현재 서버 Manifest 보고서를 보려면 빠른 시작 아이콘()을 클릭하십시오.

현재 서버 Manifest 보고서는 결과표로 표시됩니다.



## 작업 패스/실패/경고 결과 보기

지정된 날짜-시간 범위에서 완료된 작업의 패스/실패/경고 결과를 나열하는 보고서입니다.

조회 패널에는 지정된 날짜-시간 범위에서 완료된 작업의 패스/실패/경고 결과를 나열하는 보고서가 포함됩니다. 이 보고서는 모든 프로젝트의 모든 작업에 적용됩니다. 이 보고서를 사용하면 완료된 작업의 상태를 빠르게 확인할 수 있습니다.

데이터 소스: Rational Build Forge 데이터베이스

보고서 형식: 표


표시 옵션: 트리 계층 구조를 펼치고 목록의 모든 보고서 결과를 표시하려면 보고서 출력 표시를 선택하십시오.

프로젝트의 작업 결과 보고서를 보려면 다음을 수행하십시오.

1. 보고서 탭을 여십시오.
2. 조회 패널을 클릭하십시오.
3. 기본 시간 범위를 사용하거나 보고서 결과의 시간 범위를 지정하십시오. 시간을 지정하려면 UNIX 시간 형식을 사용하십시오. UNIX 시간 형식은 1970년 1월 1일 부터의 초 수입니다. [http://www.onlineconversion.com/unix\\_time.htm](http://www.onlineconversion.com/unix_time.htm)에 있는 것과 같은 도구를 사용하여 일반적인 날짜-시간 형식에서 UNIX 시간을 계산할 수 있습니다.

일은 자정(00:00)에서 시작하여 다음 날 자정에 끝납니다. 예를 들어, 2009년 7월 21일에 완료된 모든 작업을 리턴하려면 다음 UNIX 시간을 입력하십시오.

시작 : 07/21/09 00:00 AM	1248134400
종료 : 07/22/09 00:00 AM	1248220800

4. 작업 결과 보고서를 보려면 빠른 시작 아이콘()을 클릭하십시오.

작업 결과 보고서는 결과표로 표시됩니다.

## 날짜 범위의 작업에 대한 서버 활용도 보기

프로젝트의 모든 작업에 대한 서버를 나열하는 서버 활용도 보고서입니다.

조회 패널에는 프로젝트의 모든 작업에 대한 서버를 나열하는 서버 활용도 보고서가 포함됩니다. 이 보고서는 각 단계마다 단계를 실행한 서버를 식별하고 단계 지속 기간을 나열합니다. 이 보고서를 사용하면 다른 서버의 단계 지속 기간을 비교할 수 있습니다.

데이터 소스: Rational Build Forge 데이터베이스

보고서 형식: 표

표시 옵션: 트리 계층 구조를 펼치고 목록의 모든 보고서 결과를 표시하려면 보고서 출력 표시를 선택하십시오.

단계에 대한 서버 사용량 보고서를 보려면 다음을 수행하십시오.

1. 보고서 탭을 여십시오.
2. 조회 패널을 클릭하십시오.
3. 시작 및 끝 날짜를 선택하십시오.

일은 자정(00:00)에서 시작하여 다음 날 자정에 끝납니다.

4. 서버 활용도 보고서를 보려면 빠른 시작 아이콘()을 클릭하십시오.

서버 활용도 보고서는 결과표로 표시됩니다.

## MD5 값을 사용하여 작업 파일 검색

작업 BOM에서 MD5 값과 일치하는 파일을 검색하는 보고서입니다. MD5 해시 값은 파일의 디지털 지문입니다.

조회 패널에는 작업 BOM에서 MD5 값과 일치하는 파일을 검색하고 일치하는 파일을 찾으면 작업 태그와 파일 위치를 나열하는 보고서가 포함됩니다. 파일을 특정 버전을 검색하고 해당 MD5 값을 알아야 하는 경우 이 보고서를 사용하십시오.

전제조건으로, 프로젝트에는 MD5 값을 저장하기 위한 .scan 기준선 또는 .scan 체크포인트 명령이 포함되어야 합니다. 세부사항은 443 페이지의 『.scan』에 대한 참조 정보를 참고하십시오. .scan 명령은 다음 태스크를 완료합니다.


- 단계의 작업 디렉토리에 있는 파일 목록을 작성합니다.
- 작업 디렉토리에 있는 파일의 MD5 값을 생성하고 MD5 값을 작업 BOM에 저장합니다.

데이터 소스: Rational Build Forge 데이터베이스

보고서 형식: 표

표시 옵션: 트리 계층 구조를 펼치고 목록의 모든 보고서 결과를 표시하려면 보고서 출력 표시를 선택하십시오.

작업 BOM에서 MD5 값을 검색하려면 다음을 수행하십시오.

1. 보고서 탭을 여십시오.
2. 조회 패널을 클릭하십시오.
3. **MD5** 검색 값에서 찾으려는 MD5 값을 입력하십시오.
4. MD5 파일 보고서를 보려면 빠른 시작 아이콘()을 클릭하십시오.

MD5 파일 보고서는 결과표로 표시됩니다.

---

## 빠른 보고서를 사용하여 보고서 작성

빠른 보고서에서 제공하는 보고서 유형을 사용하여 사용자 정의 보고서 디자인을 작성합니다. 빠른 보고서는 Rational Build Forge의 라이선스 옵션입니다.

빠른 보고서에서 보고서 디자인을 작성, 저장, 편집 및 실행합니다. 보고서를 공용 디렉토리에 저장하면 다른 사용자와 보고서를 공유할 수 있습니다.

빠른 보고서의 보고서 유형은 Build Forge 데이터베이스의 데이터를 사용하여 보고서를 작성합니다. 보고서 유형마다 다른 보고서 필드가 포함되며 용량 및 빌드와 같은 해당 기능을 설명하는 이름을 갖습니다.

보고서를 작성하는 경우 표 또는 차트와 같은 여러 보고서 형식 중에서 선택하고 그룹화 및 정렬 기능을 사용하여 보고서 프리젠테이션을 제어합니다.

선택적으로 작업 BOM에서 보고서 결과를 실행하고 볼 수 있습니다. 438 페이지의 『report』에 대한 참조 정보를 참고하십시오.

시작하려면 보고서 탭을 클릭해서 보고를 위한 선택사항을 표시한 다음 빠른 보고서를 선택하십시오.

### 빠른 보고서에 필요한 보고서 그룹 권한

공용 또는 개인용 저장소에 보고서를 저장하여 빠른 보고서에서 작성된 보고서를 실행, 수정, 저장 및 삭제할 수 있는 그룹을 제어합니다.

보고서를 작성 및 저장하는 Rational Build Forge 사용자만 개인용 보고서에 액세스할 수 있습니다.

적절한 권한을 가진 액세스 그룹에 속한 모든 사용자는 공용 보고서에 액세스할 수 있습니다. 기본적으로, Build Engineer 그룹의 사용자는 공용 보고서로 저장된 보고서에 대한 읽기/쓰기/편집 액세스 권한이 있습니다.

액세스를 부여하려면, 다음 보고서 권한 중 하나 이상을 사용자의 액세스 그룹에 지정하십시오.

공용 보고서 읽기	빠른 보고서에서 공용 보고서 저장소에 저장된 보고서를 실행할 권한입니다.
보고서 저장, 편집 또는 삭제	빠른 보고서에서 공용 저장소에 저장된 보고서를 저장, 편집 또는 삭제할 권한입니다.

다음 테이블은 보고서 권한 그룹의 권한을 나열하고 액세스 그룹에 지정된 기본 권한을 식별합니다.

권한을 관리하려면 콘솔 UI에서 **관리 > 권한**을 선택하십시오.

	빌드 엔지니어	개발자	게스트	연산자	보안	시스템 관리자
공용 보고서 읽기	예	예			예	예
공용 보고서 저장	예					
공용 보고서 편집	예					
공용 보고서 삭제	예					

## 빠른 보고서에 대한 보고서 유형 참조

보고서 유형은 용량 및 빌드와 같은 해당 콘텐츠를 설명하는 기능 이름을 갖습니다. 이 주제는 각 보고서 유형에 대해 보고서 용도, 보고서 필드 및 보고서 예제를 설명합니다.

**분석**    **설명:** 단계 지속 기간 및 단계 실행 순서를 사용하여 단계 레벨에서 작업 성능을 보고하려면 분석 보고서 유형을 사용합니다.

**보고서 예제:**

- 작업에 포함된 각 단계의 단계 지속 기간과 단계 시퀀스를 보여주는 표 보고서를 작성합니다. 프로젝트 이름별로 작업을 그룹화합니다.
- 각 빌드 태그의 단계 지속 기간을 보여주는 선형 차트 보고서를 작성합니다. 단계 이름별로 빌드 태그를 그룹화합니다.

**필드 설명:**

필드 이름	설명
빌드 태그	작업 태그는 프로젝트 태그 형식을 기반으로 하는 고유 ID입니다.
프로젝트 이름	사용자 지정 프로젝트 이름
단계 지속 기간	단계 및 단계가 인라인한 모든 단계의 실행 시간을 초 단위로 지정합니다.
단계 이름	사용자 지정 단계 이름
단계 시퀀스	단계 실행 순서를 식별하는 숫자

**빌드** 설명: 프로젝트 레벨에서 작업 성능을 보고하려면 빌드 보고서 유형을 사용합니다.

**보고서 예제:**

- 각 빌드 태그의 빌드 결과, 시작 시간 및 지속 기간을 보여주는 테이블 보고서를 작성합니다. 프로젝트별로 빌드 태그를 그룹화합니다.
- 각 프로젝트의 빌드 계수를 보여주는 막대형 차트 보고서를 작성하십시오. 프로젝트별로 그룹화합니다.

**필드 설명:**

필드 이름	설명
빌드 계수(집계 필드)	총 작업 계수(완료된 작업과 실패 작업 포함).
빌드 지속 기간	총 작업 실행 시간(초).
빌드 결과	작업 결과(패스, 경고와 함께 패스 또는 실패).
빌드 시작 시간	작업 시작 시간.
빌드 상태	빌드 상태(실행 중, 완료, 아카이브됨 또는 잠김).
빌드 태그	작업 태그는 프로젝트 태그 형식을 기반으로 하는 고유 ID입니다.
프로젝트 이름	사용자 지정 프로젝트 이름
선택기 이름	사용자 지정 선택기 이름
사용자 로그인	작업을 시작한 사용자의 Build Forge 로그인 또는 사용자 이름
사용자 이름	작업을 시작한 사용자의 이름

**용량** 설명: 프로젝트별로 작업 성능을 보고하려면 용량 보고서 유형을 사용합니다.

**보고서 예제:**

- 각 빌드 태그의 빌드 시작, 빌드 지속 기간, 빌드 평균 지속 기간 및 빌드 결과를 보여주는 표 보고서를 작성합니다. 프로젝트별로 빌드 태그를 그룹화합니다.
- 각 프로젝트의 평균 빌드 시간을 보여주는 차트 보고서를 작성합니다. 선택기별로 프로젝트를 그룹화합니다.

**필드 설명:**

필드 이름	설명
빌드 평균 지속 기간(집계 필드)	총 작업 수를 기반으로 하는 평균 작업 실행 시간(완료된 작업과 실패 작업 포함).
빌드 지속 기간	총 작업 실행 시간(초).
빌드 결과	작업 결과(패스, 경고와 함께 패스 또는 실패).
빌드 시작 시간	작업 시작 시간.
빌드 태그	작업 태그는 프로젝트 태그 형식을 기반으로 하는 고유 ID입니다.
마지막 빌드 지속 기간(집계 필드)	마지막 작업의 실행 시간. 마지막 작업을 완료하는 총 시간(초).
프로젝트 이름	사용자 지정 프로젝트 이름

필드 이름	설명
선택기 이름	사용자 지정 선택기 이름

### 프로젝트

**설명:** 서버, 환경 및 단계 사용법을 프로젝트별로 보고하고 단계 성능을 프로젝트별로 보고하려면 프로젝트 보고서를 사용합니다.

#### 보고서 예제:

- 프로젝트, 클래스 이름 및 프로젝트 환경을 보여주는 표 보고서를 작성합니다. 서버별로 프로젝트를 그룹화합니다.
- 단계, 단계 결과, 단계 시퀀스, 단계 환경 및 서버를 보여주는 표 보고서를 작성합니다. 프로젝트별로 단계를 그룹화하고 단계 시퀀스를 기준으로 정렬합니다.
- 각 단계의 단계 계수를 보여주는 막대형 차트 보고서를 작성합니다. 단계 결과별로 단계를 그룹화합니다.

#### 필드 설명:

필드 이름	설명
클래스 이름	프로젝트의 사용자 지정 클래스(예: 프로덕션 또는 테스트).
실패 단계 계수(집계 필드)	선택된 그룹 필드(예: 프로젝트, 서버 또는 기타 필드 이름)에 대해 실패한 단계 수.
패스 단계 계수(집계 필드)	선택된 그룹 필드(예: 프로젝트, 서버 또는 기타 필드 이름)에 대해 패스한 단계 수.
프로젝트 환경 이름	프로젝트 환경 변수를 정의하는 데 사용되는 프로젝트 환경의 이름
프로젝트 레벨	프로젝트에 대한 사용자 지정 액세스 그룹입니다.
프로젝트 이름	사용자 지정 프로젝트 이름
서버 이름	사용자 지정 서버 이름
단계 계수(집계 필드)	선택된 그룹 필드(예: 프로젝트 또는 서버)의 총 단계 수
단계 환경 이름	단계의 환경 변수를 정의하는 데 사용되는 단계 환경의 이름
단계 레벨	단계에 대한 사용자 지정 액세스 그룹입니다.
단계 이름	사용자 지정 단계 이름
단계 결과	단계 결과(패스, 경고와 함께 패스 또는 실패)
단계 시퀀스	단계 실행 순서를 식별하는 숫자

### 단계 메트릭

**설명:** 단계 성공 및 실패 통계를 프로젝트별로 보고하려면 단계 메트릭 보고서를 사용합니다.

#### 보고서 예제:

- 패스 및 실패한 단계의 단계 이름, 단계 계수 및 백분율을 보여주는 표 보고서를 작성합니다. 프로젝트별로 단계를 그룹화합니다.

- 단계 지속 기간을 빌드 태그별로 보여주는 선형 차트 보고서를 작성합니다.  
단계 이름별로 빌드 태그를 그룹화합니다.

필드 설명:

필드 이름	설명
빌드 태그	작업 태그는 프로젝트 태그 형식을 기반으로 하는 고유 ID입니다.
실패 단계 백분율(집계 필드)	총 단계 계수에서 실패한 단계의 백분율. 총 단계 계수는 선택된 그룹 필드(예: 프로젝트)에 대한 계수입니다.
패스 단계 백분율(집계 필드)	총 단계 계수에서 패스한 단계의 백분율. 총 단계 계수는 선택된 그룹 필드(예: 프로젝트)에 대한 계수입니다.
프로젝트 이름	사용자 지정 프로젝트 이름
서버 이름	사용자 지정 서버 이름
단계 평균 지속 기간(집계 필드)	총 단계 수를 기반으로 하는 평균 단계 실행 시간(완료된 작업과 실패 작업 포함).
단계 계수(집계 필드)	총 단계 계수(완료된 단계와 실패 단계 포함). 총 단계 계수는 선택된 그룹 필드(예: 프로젝트)에 대한 계수입니다.
단계 지속 기간	단계 및 단계가 인라인한 모든 단계의 실행 시간을 초 단위로 지정합니다.
단계 이름	사용자 지정 단계 이름
단계 결과	단계 결과(패스, 경고와 함께 패스 또는 실패)
단계 시퀀스	단계 실행 순서를 식별하는 숫자
단계 시작 시간	단계 시작 시간

**품질 설명:** 작업 성공 및 실패 통계를 프로젝트별로 보고하려면 품질 보고서를 사용합니다.

**보고서 예제:**

- 빌드 결과를 빌드 태그별로 보여주는 테이블 보고서를 작성합니다. 프로젝트 별로 빌드 태그를 그룹화합니다.
- 패스 및 실패한 모든 빌드의 백분율을 프로젝트별로 보여주는 표 보고서를 작성합니다. 프로젝트별로 그룹화합니다.
- 빌드 계수를 프로젝트별로 보여주는 막대형 차트 보고서를 작성합니다. 빌드 결과별로 프로젝트를 그룹화합니다.

필드 설명:

필드 이름	설명
빌드 계수(집계 필드)	총 작업 계수(완료된 작업과 실패 작업 포함).
빌드 결과	작업 결과 또는 상태(패스, 경고와 함께 패스 또는 실패).
빌드 시작 시간	작업 시작 시간.
빌드 태그	작업 태그는 프로젝트 태그 형식을 기반으로 하는 고유 ID입니다.

필드 이름	설명
실패 빌드 백분율(집계 필드)	총 빌드 계수에서 실패한 빌드의 백분율. 총 빌드 계수는 선택된 그룹 필드(예: 프로젝트)에 대한 계수입니다.
패스 빌드 백분율(집계 필드)	총 빌드 계수에서 패스한 빌드의 백분율. 총 빌드 계수는 선택된 그룹 필드(예: 프로젝트)에 대한 계수입니다.
프로젝트 이름	사용자 지정 프로젝트 이름

**자원 설명:** 단계 및 작업 성능을 프로젝트와 서버별로 보고하려면 자원 보고서를 사용합니다.

**보고서 예제:**

- 단계 실행 시간을 서버별로 보여주는 표 보고서를 작성합니다. 단계 시퀀스, 단계 이름, 서버 이름, 단계 시작 시간 및 단계 지속 기간을 선택합니다. 단계 시퀀스와 시작 시간을 기준으로 정렬합니다.
- 작업 실행 시간을 서버별로 보여주는 표 보고서를 작성합니다. 빌드 태그, 서버 이름, 빌드 시작 시간, 빌드 지속 기간 및 빌드 결과를 선택합니다. 프로젝트별로 빌드 태그를 그룹화하고 빌드 시작 시간을 기준으로 정렬합니다.

**필드 설명:**

필드 이름	설명
빌드 지속 기간	총 작업 실행 시간(초).
빌드 결과	작업 결과(패스, 경고와 함께 패스 또는 실패).
빌드 시작 시간	작업 시작 시간.
빌드 태그	작업 태그는 프로젝트 태그 형식을 기반으로 하는 고유 ID입니다.
프로젝트 이름	사용자 지정 프로젝트 이름
선택기 이름	사용자 지정 선택기 이름
서버 이름	사용자 지정 서버 이름.
단계 지속 기간	단계 및 단계가 인라인한 모든 단계의 실행 시간을 초 단위로 지정합니다.
단계 시퀀스	단계 실행 순서를 식별하는 숫자
단계 시작 시간	단계 시작 시간

**BOM 설명:**

작업 BOM의 정보를 사용하여 보고서를 작성하려면 BOM 보고서를 사용합니다. BOM 보고서는 다음 데이터 세트에 대해 실행될 수 있습니다.

- 데이터베이스의 모든 프로젝트(기본값)
- 단일 프로젝트
- 단일 프로젝트에 있는 하나 이상의 빌드



선택할 수 있는 보고서 필드는 데이터 세트에 따라 다릅니다. 보고서 필드에는 단계 Manifest 특성, .scan 명령 출력, 어댑터에서 기록한 출력 및 .bom 명령을 사용하여 지정된 사용자 정의 열이 포함됩니다.

#### 보고서 예제:

- 여러 프로젝트에서 .scan 명령의 출력을 보여주는 표 보고서를 작성합니다. 빌드 태그, BOM 데이터, BOM 경로 및 BOM 유형 필드를 선택합니다. 프로젝트별로 빌드 태그를 그룹화합니다.
- 여러 프로젝트에서 필터 조치가 호출되는 횟수를 보여주는 표 보고서를 작성합니다. 단계 이름, 필터 이벤트 계수 유형 및 필터 계수를 선택합니다. 프로젝트별로 단계를 그룹화합니다.

#### 필드 설명:

필드 이름	설명
BOM 데이터	프로젝트에 <b>.scan checkpoint</b> 명령을 포함하는 경우 스캔한 파일에 대해 BOM 데이터 필드에 MD5 값이 표시됩니다.
BOM 경로	프로젝트에 <b>.scan checkpoint</b> 명령을 포함하는 경우, 스캔한 파일에 대한 BOM 경로 필드에 파일 경로가 표시됩니다.
BOM 유형	프로젝트에 <b>.scan checkpoint</b> 명령을 포함하는 경우, 스캔한 파일에 대해 BOM 유형 필드에 경로가 D(디렉토리), F(파일) 또는 S(기호 링크)인지 여부가 표시됩니다.
빌드 태그	작업 태그는 프로젝트 태그 형식을 기반으로 하는 고유 ID입니다.
필터 이벤트 계수	단계 출력에서 일치하는 패턴을 찾은 결과로서 필터가 필터 조치를 호출한 횟수.
필터 이벤트 계수 유형	단계 출력에서 일치하는 패턴을 찾을 때 필터가 호출한 필터 조치.
프로젝트 이름	사용자 지정 프로젝트 이름
결과 설명	결과 레코드의 설명 필드로, 일반적으로 단계 이름의 사본을 포함합니다.
단계 이름	사용자 지정 단계 이름
선택기 이름	사용자 지정 선택기 이름
서버 이름	사용자 지정 서버 이름
단계 지속 기간	단계 및 단계가 인라인한 모든 단계의 실행 시간을 초 단위로 지정합니다.
단계 결과	단계 결과(패스, 경고와 함께 패스 또는 실패)

## 빠른 보고서에 대한 보고서 형식 및 프리젠테이션 참조

빠른 보고서는 표, 막대형 차트, 선형 차트 및 원형 차트와 같은 여러 가지 공통 보고서 형식을 제공합니다.

**참고:** 빠른 보고서에 제공되는 샘플 보고서에는 데이터가 들어 있지 않습니다. 보고서 데이터를 보려면 먼저 프로젝트를 작성하고 관리 콘솔에서 작업을 실행해야 합니다.

## 표 보고서 형식

### 샘플

샘플 표 보고서를 보려면 빠른 보고서에서 **SampleAnalytic-StepDuration** 보고서를 선택하십시오.

프리젠테이션 옵션을 보려면 SampleAnalytic-StepDuration 보고서에 대한 편집

집 아이콘()을 선택하십시오.

### 요구사항


표의 경우 다음 요구사항을 준수하십시오.

- 하나 이상의 보고서 필드를 선택하십시오.
- 집약 보고서 필드를 선택하는 경우 그룹화 필드를 선택해야 합니다. 494 페이지의 『보고서 형식 및 집계 보고서 필드에 대한 선택 요구사항』의 내용을 참조하십시오.

### 보고서 필드

보고서 필드에서 보고서의 표 열에 될 하나 이상의 필드를 선택하십시오. 선택 목록에서는 순서가 중요합니다. 첫 번째 필드가 첫 번째 표 열이 되고 마지막 필드가 마지막 표 열이 됩니다. 첫 번째 표 열은 보고서 결과를 그룹화하는 데도 사용됩니다.


보고서 필드를 시험하려면 Sample Analytic-StepDuration 보고서에 대한 편집

아이콘()을 선택하십시오. 선택 목록에서 필드 순서를 변경하고 선택사항을 저장하고 보고서를 실행하면 테이블 열 순서대로 변경사항이 표시됩니다.

### 그룹 필드

그룹 필드에서 선택적으로 표 행을 그룹화할 필드를 선택합니다. 선택하는 그룹은 첫 번째 표 열에 트리 제어로 추가되며 보고서 결과를 그룹화하는 데 사용됩니다. 여러 그룹 필드를 선택할 수 있고 표 열로 표시되지 않는 그룹 필드를 선택할 수 있습니다.


그룹화를 시험하려면 SampleAnalytic-StepDuration 보고서에 대한 편집 아이

콘()을 선택하십시오. 프로젝트 이름을 그룹 필드로 선택하고 선택사항을 저장한 후 보고서를 실행하십시오. 빌드 태그 열에 프로젝트 이름의 트리 노드가 추가됩니다.

### 정렬 필드

정렬 필드에서 선택적으로 데이터를 정렬할 표 열을 선택하십시오. 정렬 방향을

지정하려면 화살표를 사용하십시오. 위로 화살표()는 데이터를 오름차순으


로(최하위 순위에서 최상위 순위로) 정렬하고, 아래로 화살표()는 데이터를 내림차순으로(최상위 순서에서 최하위 순서로) 정렬합니다.

정렬을 시험하려면 SampleAnalytic-StepDuration 보고서에 대한 편집 아이콘()을 선택하십시오. 프로젝트 이름을 정렬 필드로 선택하고 화살표를 사용하여 정렬 방향을 지정하고 선택사항을 저장한 후 보고서를 실행하십시오.

## 막대형 차트 보고서 형식

### 샘플

샘플 막대형 차트를 보려면 빠른 보고서에서 **SampleCapacity-RunTimeByProject** 보고서를 선택하십시오.

프리젠테이션 옵션을 보려면 SampleCapacity-RunTimeByProject 보고서에 대한 편집 아이콘()을 선택하십시오.

### 세로 차트 요소(y-축)

데이터 요소 세트(x-축)와 비교할 데이터 값(y-축)이 될 하나의 필드를 선택합니다. 필드 이름과 데이터 값 단위는 차트의 세로 또는 y-축에 표시됩니다. 세로 차트 요소로는 하나의 필드만 선택할 수 있습니다.

### 가로 차트 요소(x-축)

데이터 값(y-축)을 사용하여 비교할 데이터 요소 세트(x-축)가 될 하나의 필드를 선택합니다. 필드 이름은 x-축 레이블로 표시됩니다.

데이터 값은 막대 옆에 표시되며 값은 막대 옆 높이와 같습니다. 가로 차트 요소로는 하나의 필드만 선택할 수 있습니다.

### 그룹 필드

그룹 필드를 선택하여 차트의 가로 또는 x-축에서 데이터 요소를 선택적으로 그룹화할 수 있습니다.

### 요구사항

차트의 경우 다음 요구사항을 준수하십시오.

- x-축 및 y-축 보고서 필드를 모두 선택하십시오.
- x-축과 y-축에 다른 보고서 필드를 선택하십시오.
- x-축 데이터의 그룹화 필드를 선택하십시오.
- 집약 보고서 필드를 선택하는 경우 그룹화 필드를 선택해야 합니다. 494 페이지의 『보고서 형식 및 집계 보고서 필드에 대한 선택 요구사항』의 내용을 참조하십시오.

### 예제:

막대형 차트를 실험하려면 다음 보고서를 작성하십시오.

- 단계 보고서 유형의 경우 각 단계 이름(x-축)에 대해 실패 단계 백분율(y-축)을 비교하고 빌드 태그별로 결과를 그룹화합니다.
- 용량 보고서 유형의 경우 각 프로젝트(x-축)에 대해 마지막 빌드 지속 시간(y-축)을 비교하고 프로젝트 이름별로 결과를 그룹화합니다.
- 자원 보고서 유형의 경우 각 빌드 태그(x-축)에 대해 빌드 지속 시간(y-축)을 비교하고 프로젝트 이름별로 빌드 태그를 그룹화합니다.


## 선형 차트 보고서 형식

### 설명

선형 차트는 특정 시간 또는 일련의 이벤트에 대한 데이터 진행 상태를 보여줍니다. 기본 그룹화 필드는 x-축 보고서 필드입니다. 여러 그룹화 필드를 선택하면 보고서에 여러 선이 표시됩니다.

### 샘플

샘플 선형 차트를 보려면 빠른 보고서에서 **SampleResource-DurationOverTime** 보고서를 선택하십시오.

프리젠테이션 옵션을 보려면 SampleResource-DurationOverTime 보고서에 대한 편집 아이콘()을 선택하십시오.

### 세로 차트 요소(y-축)

데이터 요소 세트(x-축)와 비교할 데이터 값(y-축)이 될 하나의 필드를 선택합니다. 필드 이름과 데이터 값 단위는 차트의 y-축에 표시됩니다. 세로 차트 요소로는 하나의 필드만 선택할 수 있습니다.

### 가로 차트 요소(x-축)

데이터 값(y-축)을 사용하여 비교할 데이터 요소 세트(x-축)가 될 하나의 필드를 선택합니다. 필드 이름은 차트 x-축의 레이블로 표시됩니다.

x-축 데이터 요소의 데이터 값은 y-축의 해당 값과 같은 점 옆에 표시됩니다. 연속 선이 점을 연결하여 선형 차트를 나타냅니다. 가로 차트 요소로는 하나의 필드만 선택할 수 있습니다.

### 그룹 필드

그룹 필드를 선택하여 차트의 가로 또는 x-축에서 데이터 요소를 선택적으로 그룹화할 수 있습니다. 기본 그룹화 필드는 x-축 보고서 필드입니다. 여러 그룹화 필드를 선택하면 보고서에 여러 선이 표시됩니다.

### 요구사항

차트의 경우 다음 가이드라인을 준수하십시오.

- x-축 및 y-축 보고서 필드를 모두 선택하십시오.
- x-축과 y-축에 다른 보고서 필드를 선택하십시오.

- x-축 데이터의 그룹화 필드를 선택하십시오.
- 집약 보고서 필드를 선택하는 경우 그룹화 필드를 선택해야 합니다. 494 페이지의 『보고서 형식 및 집계 보고서 필드에 대한 선택 요구사항』의 내용을 참조하십시오.

#### 예제

선형 차트를 실험하려면 다음 보고서를 작성하십시오.

- 단계 보고서 유형의 경우 각 빌드 태그(x-축)에 대해 단계 지속 기간(y-축)을 보여주며 단계 이름별로 결과를 그룹화합니다.
- 빌드 보고서의 경우 각 빌드 시작 시간(x-축)에 대해 지속 기간(y-축)을 보여주며 프로젝트 이름별로 결과를 그룹화합니다.
- 용량 보고서 유형의 경우 각 빌드 태그(x-축)에 대해 빌드 지속 기간(y-축)을 보여주며 서버 이름별로 결과를 그룹화합니다.

#### 원형 차트 보고서 형식

##### Y-시리즈 요소

원형 차트의 데이터 값이 될 하나의 필드를 선택합니다. 썬치 크기는 데이터 값을 나타냅니다. 필드 이름과 데이터 값 단위는 원형 차트에 표시됩니다. y-시리즈 요소로는 하나의 필드만 선택할 수 있습니다.

##### X-시리즈 요소

데이터 값(y-시리즈)을 사용하여 평가할 데이터 요소 세트(x-축)가 될 하나의 필드를 선택합니다. 썬치 수는 데이터 요소 수를 나타냅니다. x-시리즈 요소로는 하나의 필드만 선택할 수 있습니다.

##### 그룹 필드

원형 차트에는 그룹이 필요합니다. x-시리즈 요소는 기본 그룹으로 사용됩니다. 다른 그룹을 선택하면 데이터 값(y-시리즈)의 다른 원형 차트가 생성됩니다.

##### 요구사항

차트의 경우 다음 가이드라인을 준수하십시오.

- x-시리즈 및 y-시리즈 보고서 필드를 모두 선택하십시오.
- x-시리즈와 y-시리즈에 다른 보고서 필드를 선택하십시오.
- x-축 데이터의 그룹화 필드를 선택하십시오.
- 집약 보고서 필드를 선택하는 경우 그룹화 필드를 선택해야 합니다. 494 페이지의 『보고서 형식 및 집계 보고서 필드에 대한 선택 요구사항』의 내용을 참조하십시오.

#### 예제

원형 차트를 실험하려면 다음 보고서를 작성하십시오.

- 프로젝트 보고서 유형의 경우 실패 단계 계수(y-시리즈)와 단계 이름(x-시리즈)을 선택하십시오.
- 용량 보고서 유형의 경우 빌드 평균 지속 기간(y-시리즈)과 프로젝트 이름(x-시리즈)을 선택하십시오.
- 분석 유형의 경우 단계 지속 기간(y-시리즈)과 단계 이름(x-시리즈)을 선택하고 단계 이름별로 그룹화하십시오.

## 보고서 형식 및 집계 보고서 필드에 대한 선택 요구사항

의미 있는 보고서를 생성하려면 표 및 차트와 집계 보고서 필드에 대한 몇 가지 기본 요구사항을 준수해야 합니다.

### 집계 보고서 필드

#### 요구사항

표 또는 차트에 집계 보고서 필드를 포함하는 경우 그룹화 필드를 포함해야 합니다.

#### 정의

집계 보고서 필드에는 Build Forge 데이터베이스의 하나 이상의 원본 데이터 필드에서 파생된 데이터가 들어 있습니다. 집계 보고서 필드의 데이터는 데이터베이스에 저장되지 않습니다.

다음 보고서 필드는 집계 필드입니다.

분석: 마지막 빌드 지속 기간	프로젝트: 실패 단계 계수	프로젝트: 패스 단계 계수
프로젝트: 단계 계수	단계 메트릭: 실패 단계 백분율	단계 메트릭: 패스 단계 백분율
단계 메트릭: 단계 평균 지속 기간	단계 메트릭: 단계 계수	품질: 빌드 계수
품질: 실패 빌드 백분율	품질: 패스 빌드 백분율	빌드: 빌드 계수

### 표 보고서 형식

#### 요구사항

표의 경우 다음 요구사항을 준수하십시오.

- 하나 이상의 보고서 필드를 선택하십시오.
- 집계 보고서 필드를 선택하는 경우 그룹 필드를 선택해야 합니다.

### 차트 보고서 형식

#### 요구사항

막대형 차트, 선형 차트 및 원형 차트의 경우 다음 요구사항을 준수하십시오.


- x-시리즈 및 y-시리즈 보고서 필드를 모두 선택하십시오.
- x-시리즈와 y-시리즈에 다른 보고서 필드를 선택하십시오.
- x-축 데이터의 그룹화 필드를 선택하십시오.
- 집계 보고서 필드를 선택하는 경우 그룹 필드를 선택해야 합니다.


## 샘플 보고서 참조

샘플 보고서는 제공된 보고서 유형으로 작성할 수 있는 보고서 예제입니다.

**참고:** 샘플 보고서는 Rational Build Forge 데이터베이스의 데이터를 사용하여 보고서 출력을 작성합니다. 이전에 프로젝트를 작성하고 작업을 실행한 경우에만 결과가 표시됩니다.

샘플 보고서 목록을 표시하려면 **빠른 보고서**를 클릭하십시오.

샘플 보고서의 필드 및 형식 옵션을 보려면 보고서의 **편집** 아이콘()을 선택하십시오.

샘플 보고서를 복사하려면 보고서의 **편집** 아이콘()을 선택하고 **보고서 복사**를 클릭하십시오.

샘플 보고서를 실행하려면 보고서 이름을 선택하십시오.

샘플 보고서 정의는 다음 표에 제공됩니다.

샘플 보고서 이름	설명
SampleAnalytic-StepDuration	<p>설명: 빌드의 단계 세부사항을 보고하는 표를 작성합니다.</p> <p>보고서 유형: 분석</p> <p>보고서 형식: 표</p> <p>보고서 필드: 빌드 태그, 프로젝트 이름, 단계 이름, 단계 지속 기간</p> <p>보고서 옵션: 그룹화 또는 정렬 옵션 없음</p>
SampleBuild-BuildsByState	<p>설명: 프로젝트별로 빌드 및 빌드 상태를 나열하는 표를 작성합니다.</p> <p>보고서 유형: 빌드</p> <p>보고서 형식: 표</p> <p>보고서 필드: 빌드 태그, 프로젝트 이름, 빌드 개수, 빌드 상태</p> <p>보고서 옵션: 프로젝트 이름별로 그룹화</p>
SampleCapacity- RuntimeByProject	<p>설명: 작업의 평균 빌드 지속 기간을 프로젝트별로 보여주는 막대형 차트를 작성합니다.</p> <p>보고서 유형: 용량</p> <p>보고서 형식: 막대형 차트</p> <p>보고서 필드: 빌드 평균 지속 기간(y-축) 및 프로젝트 이름(x-축)</p> <p>보고서 옵션: 프로젝트 이름별로 그룹화</p>

샘플 보고서 이름	설명
SampleProject-TotalsByProject	<p>설명: 프로젝트에서 패스 및 실패한 단계의 수를 보고하는 표를 작성합니다.</p> <p>보고서 유형: 프로젝트</p> <p>보고서 형식: 표</p> <p>보고서 필드: 프로젝트 이름, 패스된 단계 계수, 실패한 단계 계수, 단계 계수</p> <p>보고서 옵션: 프로젝트 이름별로 그룹화</p>
SampleQuality-PercentSuccess	<p>설명: 프로젝트에서 패스 및 실패한 작업의 백분율을 보고하는 표를 작성합니다.</p> <p>보고서 유형: 품질</p> <p>보고서 형식: 표</p> <p>보고서 필드: 프로젝트 이름, 빌드 계수, 실패 빌드 백분율, 패스 빌드 백분율</p> <p>보고서 옵션: 프로젝트 이름별로 그룹화</p>
SampleResource-DurationOverTime	<p>설명: 각 빌드의 빌드 시작 시간 및 빌드 지속 기간을 보여주는 차트를 작성합니다.</p> <p>보고서 유형: 자원</p> <p>보고서 형식: 표</p> <p>보고서 필드: 빌드 지속 기간(y-축) 및 빌드 시작 시간(x-축)</p> <p>보고서 옵션: 오름차순 정렬 시퀀스</p>

## 제공된 보고서 유형을 사용하여 보고서 작성

사용할 보고서 유형, 보고서 형식 및 보고서 필드를 선택하여 사용자 정의 보고서를 작성하십시오.

### 프로시저

1. 보고서 탭을 선택한 다음 **빠른** 보고서를 선택하십시오.
2. 보고서 이름에 고유 이름을 입력하십시오. 보고서 이름은 필수입니다. 보고서는 데이터베이스에 보고서를 저장하는 데 사용되며 고유해야 합니다.
3. 보고서 제목에 보고서의 설명 제목을 입력하십시오. 보고서 제목은 필수입니다. 제목은 보고서 맨 위에 표시됩니다.
4. 가시성에서 공용 또는 개인용을 선택하십시오. 개인용 보고서는 공유할 수 없습니다. 공용 보고서는 필수 액세스 권한이 있는 사용자가 공유할 수 있습니다. 세부사항은 483 페이지의 『빠른 보고서에 필요한 보고서 그룹 권한』의 내용을 참조하십시오.



5. 보고서 형식에서 표 또는 하나 이상의 차트 형식(막대형, 선형 또는 원형)을 선택하십시오. 보고서 형식에 대한 세부사항은 489 페이지의 『빠른 보고서에 대한 보고서 형식 및 프리젠테이션 참조』의 내용을 참조하십시오.
6. 보고서 유형에서 보고서 유형을 선택하십시오. 보고서 유형은 보고서에 포함되는 내용을 결정합니다. 세부사항 및 예제는 484 페이지의 『빠른 보고서에 대한 보고서 유형 참조』의 내용을 참조하십시오.
  - 표 또는 차트 보고서에 표시할 보고서 필드를 선택하십시오.
  - 보고서 프리젠테이션을 제어할 그룹 및 정렬 옵션을 선택하십시오.

**중요사항:** BOM 보고서 유형을 선택하면 프로젝트 상자가 표시됩니다. 특정 프로젝트에 대한 BOM 보고서를 작성하려면 먼저 프로젝트를 선택하여 보고서 필드의 전체 목록을 표시한 다음 보고서에 포함할 필드를 선택하십시오.

7. (BOM 보고서 유형에만 해당) 프로젝트에서 다음 옵션 중 하나를 선택하십시오.
  - 표준 BOM 보고서 필드만 표시하려면 모든 프로젝트를 선택하십시오(모든 프로젝트가 기본 설정임).

보고서 출력에는 데이터베이스의 모든 프로젝트에 대한 BOM 정보가 포함됩니다.

- 표준 BOM 보고서 필드와 프로젝트 특정 BOM 필드를 모두 표시하려면 하나의 프로젝트를 선택하십시오.

보고서 출력에는 단일 프로젝트에 대한 BOM 정보만 포함됩니다.

8. 보고서 저장을 클릭해서 선택사항을 저장하십시오. 보고서는 보고서 목록에 표시됩니다.

## 작업 BOM에 보고서 출력 추가

선택적으로 .report 명령을 사용하여 작업 BOM에 보고서 결과를 추가합니다.

### 프로시저

1. 빠른 보고서 도구를 사용하여 보고서를 작성하십시오. 496 페이지의 『제공된 보고서 유형을 사용하여 보고서 작성』의 내용을 참조하십시오.

**참고:** 이 기능은 BOM 보고서 유형 또는 개인용 보고서에는 지원되지 않습니다.

2. .report 명령을 사용하여 작업 BOM에 보고서를 추가하십시오. 438 페이지의 『.report』에 대한 참조 정보를 참고하십시오.

## 빠른 보고서에서 보고서 수정 및 관리

빠른 보고서에서 보고서를 실행하고 보고서 결과를 보고 보고서 디자인을 편집할 수 있습니다.

보고서 디자인을 변경하기 위해 보고서를 편집할 수 있습니다. 필터는 보고서 정의의 일 부분이므로 출력을 필터링하기 위해 보고서에 추가할 수 있습니다.

## 보고서 실행

빠른 보고서에서 작성하는 모든 보고서를 실행하려면 보고서 이름을 클릭하십시오.

## 이 태스크 정보

빠른 보고서 결과 보기에 보고서 결과가 표시됩니다.

## 결과

보고서 목록으로 되돌아가려면 웹 브라우저의 이전 화살표를 클릭하십시오.

## 보고서 편집

빠른 보고서에서 작성하는 모든 보고서를 편집하려면 보고서 이름 옆에 있는 편집 아이콘을 클릭하십시오.

## 이 태스크 정보

보고서 선택사항이 보고서 세부사항에 표시됩니다. 변경이 완료된 후 보고서 저장을 클릭해서 변경사항을 저장하십시오.

## 보고서 복사

샘플 보고서 또는 다른 보고서를 복사하면 해당 보고서 필드와 형식 옵션이 새 보고서 디자인에 복사되며 보고서에 고유 이름이 지정됩니다.

## 시작하기 전에

고유 보고서 이름은 보고서 이름에 Copy를 추가하여 작성됩니다(예: <Report\_Name> Copy).

보고서 사본을 여러 개 작성하는 경우 <Report\_Name> Copy <Copy\_Number> 구문을 사용하여 번호가 추가됩니다. 예: <Report\_Name> Copy 2.

## 프로시저

1. 복사될 보고서의 편집 아이콘을 클릭하십시오.
2. 보고서 복사를 클릭하십시오.

복사된 보고서는 보고서 목록에 표시됩니다.

## 보고서 출력에 대한 필터 작성

보고서 출력에 표시되는 정보를 제어하는 보고서 필터를 작성할 수 있습니다.

## 시작하기 전에

필터를 작성하기 전에 다음 요구사항과 제한사항을 검토하십시오.

- 먼저 보고서를 작성하십시오. 필터링은 필터 옵션을 제공하기 위해 보고서 정의를 사용합니다.
- 보고서 필터는 보고서 정의의 일부로 저장되며 단일 보고서에만 적용됩니다.
- 작성된 보고서 필터는 필터를 변경 또는 삭제할 때까지 해당 보고서를 실행할 때마다 적용됩니다.
- BOM 보고서 유형의 경우 프로젝트별로 필터링하고 빌드별로 필터링하고 특정 보고서 필드의 필터 기준을 지정할 수 있습니다.
- BOM 이외의 보고서 유형의 경우 특정 보고서 필드의 필터 기준을 지정해야 합니다.

## 프로시저

1. 보고서 목록에서 보고서를 선택하고 보고서에 대한 편집 아이콘을 선택하십시오.
2. (BOM 보고서 유형에만 필수) 프로젝트에서 다음 옵션 중 하나를 선택하십시오.
  - 표준 BOM 보고서 필드만 표시하려면 모든 프로젝트를 선택하십시오(모든 프로젝트가 기본 설정임).

보고서 출력에는 데이터베이스의 모든 프로젝트에 대한 BOM 정보가 포함됩니다.

- 표준 BOM 보고서 필드와 프로젝트 특정 BOM 필드를 모두 표시하려면 하나의 프로젝트를 선택하십시오.

보고서 출력에는 단일 프로젝트에 대한 BOM 정보만 포함됩니다.

3. 필터 탭을 선택하십시오.
4. (BOM 보고서 유형에만 선택적) 빌드 필터 표시를 클릭하고 보고서에 포함할 빌드 정보를 지정하기 위한 다음 옵션을 선택하십시오.
  - 모든 현재 및 미래 빌드에 대한 데이터를 포함하려면 선택하지 마십시오.
  - 모든 현재 빌드에 대한 데이터만 포함하려면 모든 현재 빌드만을 클릭하십시오.
  - 선택된 빌드의 모든 현재 및 미래 빌드에 대한 데이터만 포함하려면 목록에서 빌드 태그를 선택하십시오.
5. (다른 모든 보고서 유형에 필수) 필터 추가를 클릭해서 보고서 세부사항 탭에서 선택된 보고서 유형에서 필터로 사용할 보고서 필드를 선택하십시오.

필터 표현식을 작성하려면 다음을 수행하십시오.

- a. 필터 필드에서 보고서 데이터를 필터링하기 위해 사용할 보고서 필드를 선택하십시오.
- b. 필터 연산자에서 관계형 연산자를 선택하십시오.

- c. 필터 값에서 보고서 필드 값을 입력하십시오.
- 6. 보고서 저장을 클릭해서 보고서 필터 선택사항을 저장하십시오.

### **빠른 보고서에 대한 공통 문제점 해결**

Rational 소프트웨어 분석기를 사용할 때 문제가 발생하는 경우 이 주제의 정보를 검토하여 허용 가능한 해결 방법 또는 솔루션이 있는지 여부를 확인합니다.

#### **포트 충돌**

빠른 보고서는 설치 중에 지정한 애플리케이션 웹 서버를 사용하여 보고서를 표시합니다. 포트 충돌이 발생하는 경우 애플리케이션 웹 서버에 지정되지 않은 포트를 구성해야 합니다.

---

## 제 23 장 유틸리티에 대한 작업

이 주제는 Rational Build Forge가 제공하는 명령행 유틸리티를 설정 및 실행하는 방법에 대해 설명합니다.

---

### 명령행 유틸리티 액세스 및 실행

명령행 유틸리티는 Build Forge 설치 디렉토리에 위치하는데, 이 디렉토리는 `<bfinstall>` (Windows) 및 `<bfinstall>/Platform`(UNIX 및 Linux)입니다.

엔진 레벨 셸 명령이 작동하도록 하려면 환경(또는 시스템 레벨 환경)을 제대로 구성해야 합니다. 예를 들어 Oracle 및 UNIX에서는 ORACLE\_HOME, TNS\_ADMIN 및 LD\_LIBRARY\_PATH가 수동으로 설정되어야 하며, 그렇지 않으면 유틸리티가 실행되지 않습니다.

bfexport 또는 bfimport와 같은 명령행 유틸리티를 사용하는 경우 명령은 데이터베이스에 액세스하기 위해 buildforge.conf 파일을 찾을 수 있어야 하므로, 설치 디렉토리에서 명령을 실행하거나 buildforge.conf 파일의 전체 경로에 환경 변수 BF\_CONFIG\_FILE을 설정해야 합니다.

---

### 프로젝트 내보내기

bfexport 명령 또는 .export 점 명령을 사용하여 XML 파일로 프로젝트와 기타 Build Forge 오브젝트를 내보낼 수 있습니다.

내보낸 프로젝트는 XML 파일에 저장됩니다. 내보낸 프로젝트를 Build Forge에 다시 가져올 수 있습니다.

예제: bfexport를 실행하는 프로젝트 끝에 단계를 추가하여 프로젝트 구성 데이터를 저장하고 프로젝트 정의에 대한 백업으로 사용할 수 있습니다. 다른 Build Forge 설치로 프로젝트를 이동하는 데 사용할 수도 있습니다.

### bfexport 참조

bfexport 명령을 사용하여 이름 지정된 XML 파일로 프로젝트 데이터를 내보내거나 표시 터미널(stdout)로 보려는 프로젝트 데이터를 보냅니다. 내보내기 파일에는 단일 프로젝트 또는 프로젝트 스냅샷의 프로젝트 구성 데이터가 포함됩니다.

## 구문

bfexport

bfexport [-l]

bfexport [-l] <project\_name>

bfexport [-c "<comment>"] [-f <file\_name>] [-g] [-s] [-C] [-L ] [-n ]  
<project\_name> | <project\_name> <snapshot\_name> | <project\_id>

## 사용법

공동 프로젝트 내보내기 태스크를 완료하려면 다음 명령 옵션을 사용하십시오.

- 명령 구문을 표시하려면 옵션 없이 bfexport를 사용하십시오.
- Build Forge 데이터베이스에 저장된 프로젝트 이름과 프로젝트 ID를 나열하려면 bfexport -l을 사용하십시오.

스냅샷 이름이 명령 출력의 프로젝트 이름에 추가됩니다(<project\_ID>:  
<project\_name> - <snapshot\_name>).

- 프로젝트 데이터를 XML 파일로 전송하려면, bfexport -f <file\_name>을 사용하십시오. 프로젝트 데이터를 가져오는 데 사용할 수 있는 파일을 생성하려면 -f <file\_name> 옵션을 지정해야 합니다.
- z/Linux 플랫폼에서는 명령을 bfexport.pl로서 실행해야 합니다. 다른 모든 플랫폼의 경우 명령에 확장자가 필요하지 않습니다.

## 전제조건 및 제한사항

Build Forge 설치 디렉토리에서 bfexport 유틸리티를 찾으십시오.

서버의 서버 권한 비밀번호는 내보내기 파일에 포함되지 않으므로 가져오기 후 UI에서 서버 권한 비밀번호를 수동으로 입력해야 합니다.

bfexport 명령은 buildforge.conf 파일을 찾고 Build Forge 데이터베이스에 액세스할 수 있어야 합니다. buildforge.conf가 있는 디렉토리에서 bfexport를 실행하십시오. <bfinstall>(Windows) 및 <bfinstall>/Platform(UNIX 및 Linux)에 있습니다.

## 예제:

출력을 파일에 쓰려면 -f <file\_name> 옵션을 사용하십시오. 다음 예제에서는 helloworld가 출력 파일 이름이고 프로젝트 이름 대신 프로젝트 ID가 사용됩니다.

```
bfexport -c "Saving a copy of project before making changes"  
-f helloworld 675B57CC-8366-11DD-B2E0-043C04E44E1A
```

기본 프로젝트 스냅샷만 내보내려면 <project\_name>을 사용하십시오.

```
bfexport -f helloworld test_project
```

프로젝트의 스냅샷 하나를 내보내려면 `<project_name> <snapshot_name>`을 사용하십시오.

```
bfexport -f helloworld test_project snapshot_1
```

상위 프로젝트 스냅샷이 기본 프로젝트가 아닌 경우 `<project_name>` 다음에 상위 키워드를 지정하여 상위 프로젝트 스냅샷을 내보내야 합니다.

```
bfexport -f helloworld test_project parent
```

## 옵션 설명

옵션	설명
<code>&lt;project_name&gt;</code>	내보낼 프로젝트의 이름. 프로젝트 이름 또는 프로젝트 ID가 필요합니다. 프로젝트 이름에 공백이 포함되는 경우 이름을 따옴표로 묶어야 합니다.  프로젝트 이름은 명령 옵션 다음에 지정하십시오.
<code>&lt;snapshot_name&gt;</code>	내보낼 프로젝트 스냅샷의 이름. 다음 구문과 같이 프로젝트 이름이 필요합니다.  <code>&lt;project_name&gt; &lt;snapshot_name&gt;</code>  프로젝트 이름과 스냅샷 이름은 명령 옵션 다음에 지정하십시오. 프로젝트 또는 스냅샷 이름에 공백이 포함되는 경우 이름을 따옴표로 묶어야 합니다.
<code>&lt;project_id&gt;</code>	내보낼 프로젝트의 ID. 프로젝트 ID는 UUID입니다. 프로젝트 ID 또는 프로젝트 이름이 필요합니다.  프로젝트 ID는 명령 옵션 다음에 지정하십시오.
<code>-f &lt;file_name&gt;</code>	bfexport 출력의 XML 파일 이름. 경로 이름을 제공하지 않으면 현재 작업 디렉토리가 사용됩니다. 파일 이름에 공백이 포함되는 경우 이름을 따옴표로 묶어야 합니다.  파일 이름을 제공하지 않으면 bfexport 출력을 stdout로 보냅니다.  참고: stdout는 보기용으로만 사용하십시오. stdout를 파일로 재지정하지 마십시오. 결과 파일에는 로깅 메시지가 포함되며 bfimport 명령 또는 UI 가져오기 유틸리티에 대한 가져오기 파일로 사용할 수 없습니다.
상위	상위가 기본 프로젝트 스냅샷이 아닌 경우 상위 프로젝트 스냅샷을 내보내는 데 필요한 키워드.  상위 키워드는 다음과 같이 프로젝트 스냅샷 이름 다음에 지정하십시오. <code>bfexport -f helloworld test_project parent</code>
<code>-l</code>	데이터베이스의 프로젝트를 이름과 프로젝트 ID별로 나열합니다. -l 옵션은 다른 옵션과 함께 사용할 수 없습니다.
<code>-c "&lt;comment&gt;"</code>	주석을 포함합니다. 주석은 따옴표로 묶어야 합니다(예: "my project version 50"). 주석은 <buildforge> XML 요소에 추가됩니다.
<code>-g</code>	알림을 수신하도록 지정된 액세스 그룹의 구성원인 사용자를 XML 파일에 저장합니다. 사용자 및 해당 특성이 <user> XML 요소에 나열됩니다. -s가 필요합니다.
<code>-s</code>	관리 콘솔에서 정의된 서버를 XML 파일에 저장합니다. 서버 및 해당 특성이 연관된 <auth> 및 <collector> 정보와 함께 <server> XML 요소에 나열됩니다.

옵션	설명
-L	UI에서 정의된 LDAP 도메인 제어를 XML 파일에 저장합니다. LDAP 도메인 제어 및 해당 특성이 <ldap> XML 요소에 나열됩니다.
-n	프로젝트와 단계에 지정된 알림 템플릿을 XML 파일에 저장합니다. 알림 템플릿 및 해당 특성이 <mail-template> XML 요소에 나열됩니다.
-C	프로젝트의 서버에 지정된 콜렉터를 XML 파일에 저장합니다. 콜렉터 및 해당 특성이 <collector> XML 요소에 나열됩니다. -s가 필요합니다.

## 문제점 해결

보고서 파일 또는 라이브러리를 찾을 수 없다는 오류가 발생하는 경우 라이브러리 경로가 올바르게 설정되었는지 확인하십시오.

- Linux 시스템 및 UNIX 시스템의 *LD\_LIBRARY\_PATH*.
- AIX 시스템의 *LIBPATH*.

올바른 설정은 <bfinstall>/rc/buildforge입니다. 해당 파일에 라이브러리 경로 변수 및 기타 변수가 설정되었는지 확인하십시오. 필요에 따라 해당 변수를 재설정 후 내보내십시오.

## .export 사용

.export를 사용하여 프로젝트의 단계에서 프로젝트를 내보낼 수 있습니다.

.export 명령은 다른 어떤 오브젝트 데이터를 내보내는 옵션도 제공하지 않습니다. 기타 Build Forge 오브젝트를 내보내려면 bfexport 명령을 사용해야 합니다.

430 페이지의 『.export』의 내용을 참조하십시오.

## 프로젝트 가져오기

콘솔에서 가져오기 기능이나 bfimport 명령을 사용하여 이전에 내보낸 프로젝트 및 기타 Build Forge 오브젝트를 가져올 수 있습니다.

옵션 범위를 사용하여 설치할 오브젝트를 선택할 수 있습니다.

가져온 오브젝트에 액세스 그룹을 적용하는 방법에 대한 옵션이 있습니다. 이 옵션은 보안 액세스로 가져오기 시스템 설정을 통해 설정됩니다.

**참고:** Build Forge 사용자 인터페이스가 2M가 넘는 파일 가져오기를 거부할 수 있습니다. 이 경우에는 bfimport 명령행 도구를 직접 사용하거나 "upload\_max\_filesize" PHP 환경 변수를 조정하십시오.



## 가져오기 유틸리티를 사용하여 프로젝트 및 기타 오브젝트 가져오기

UI 가져오기 유틸리티를 사용하여 내보내기 파일에 저장된 기타 오브젝트 및 프로젝트에 대한 오브젝트 정의를 가져올 수 있습니다. 가져오기 유틸리티를 사용하여 내보내기 파일에서 가져올 오브젝트를 선택할 수 있습니다.

1. 관리 > 가져오기를 선택하십시오.

2. 프로젝트의 내보내기 XML 파일을 찾으려면 **찾아보기**를 클릭하십시오.

bfexport 명령 또는 .export 점 명령을 사용하여 내보내기 파일을 작성해야 합니다.

3. 내보내기 XML 파일에서 UI로 가져올 나머지 오브젝트와 프로젝트를 선택하십시오.

**참고:** 서버 오브젝트를 가져오는 경우 서버 권한 비밀번호를 UI에 수동으로 입력해야 합니다. bfexport 및 .export 명령은 내보내기 파일에 서버 권한 비밀번호를 저장하지 않습니다.

4. 가져오기 유틸리티가 이름이 같은 오브젝트를 바꾸거나 겹쳐쓰는지 또는 오브젝트 이름을 바꾸는지 여부를 지정하려면 엔티티 바꾸기 또는 엔티티 이름 바꾸기(기본 옵션)를 선택하십시오.

**중요사항:** 작업과 작업 이름을 바꾸는 방법을 이해하려면 510 페이지의 『가져오기에서 오브젝트 및 오브젝트 이름 바꾸기』의 내용을 참조하십시오.

## bfimport 참조

bfimport 유틸리티를 사용하여 프로젝트 및 기타 오브젝트에 대한 정의를 이전에 XML 파일로 내보낸 UI로 가져옵니다. 가져오기 유틸리티를 사용하여 XML 파일에서 선택된 오브젝트를 가져올 수도 있습니다.

이 주제는 bfimport 명령 구문과 자세한 사용법에 대해 설명합니다.

### 구문

bfimport

bfimport [-L] <file\_name>

bfimport [-p -I -s -S -e -c -C -u -T -f -d -r ] <file\_name.xml>

bfimport [-L | [-p -I -s -S -e -c -C -u -T -f -d -r ]] <file\_name.xml>

### 제한사항 및 고려사항

서버의 서버 권한 비밀번호는 내보내기 파일에 포함되지 않으므로 서버 권한 비밀번호는 수동으로 입력해야 합니다.

기본적으로 가져오기에서 가져온 오브젝트와 이름이 같은 오브젝트가 존재하는 경우 데이터베이스 오브젝트를 겹쳐쓰지 않도록 가져오는 오브젝트 이름이 바뀝니다. 또는 이름이 같은 오브젝트가 있는 경우 -r 옵션으로 오브젝트를 바꾸도록 선택할 수 있습니다. 스냅샷 오브젝트의 경우 오브젝트를 이름보다는 UUID로 비교합니다.

기본적으로 bfimport로 오브젝트 이름을 바꾸고 다음 이름 지정 규칙이 사용됩니다.

<object\_name>\_IMPORT\_<number>

예를 들어, 스냅샷 오브젝트가 상위 스냅샷의 하위인 경우라도 스냅샷 오브젝트는 스냅샷 이름을 잃고 새로운 기준 또는 상위 레벨 스냅샷으로서 가져옵니다.

이름 바꾸기에 대한 세부사항은 510 페이지의 『가져오기에서 오브젝트 및 오브젝트 이름 바꾸기』의 내용을 참조하십시오.

오브젝트를 바꾸려면 -r 옵션을 지정해야 합니다. 바꾸기 옵션은 기존 오브젝트를 겹쳐 씁니다. 7.0.2 이전의 내보내기 파일의 경우, 스냅샷 오브젝트가 바뀌지 않고, 대신 <object\_name>\_IMPORT\_<number> 규칙을 사용하여 이름이 바뀝니다. 7.1 내보내기 파일의 경우 해당 UUID가 일치할 경우 스냅샷 오브젝트가 바뀝니다.

## 전제조건

bfexport 명령 또는 .export 점 명령으로 작성된 내보내기 XML 파일.

Build Forge 설치 디렉토리에서 bfimport 유틸리티를 찾으십시오.

bfimport 명령은 buildforge.conf 파일을 찾고 Build Forge 데이터베이스에 액세스할 수 있어야 합니다. buildforge.conf가 있는 디렉토리에서 bfimport를 실행하십시오. <bfinstall>(Windows) 및 <bfinstall>/Platform(UNIX 및 Linux)에 있습니다.

## 사용법

공통 가져오기 태스크를 완료하려면 다음 명령 옵션을 사용하십시오.

- 명령 구문을 표시하려면 옵션 없이 bfimport를 사용하십시오.
- XML 파일의 Build Forge 오브젝트 및 해당 이름의 요약 목록을 표시하려면 bfimport -L <file\_name.xml>을 사용하십시오.
- zLinux 플랫폼에서는 명령을 bfexport.pl로서 실행해야 합니다. 다른 모든 플랫폼의 경우 명령에 확장자가 필요하지 않습니다.
- 옵션을 지정하지 않으면 오브젝트를 가져오지 않습니다. 개별 오브젝트를 가져오는 옵션을 지정해야 합니다.
- 존재하지 않는 오브젝트의 옵션을 지정하면 가져오기 유틸리티가 XML 파일에 없는 오브젝트를 건너뛰고 파일에 있는 오브젝트를 가져옵니다.

## 예제:

XML 파일의 Build Forge 오브젝트를 나열하려면 -L 옵션과 XML 파일 이름만 지정하십시오. 다음 예제에는 부분 명령 출력이 표시됩니다.

```
C:\Program Files\IBM\Build Forge>bfimport -L samples\projects\basic.xml
10/07/2008 5:31:55 PM: Import: 7624: CRRBF20081I: Importing export
file from a 7.0.10025 version console.
Project : [Basic Sample]
Tag Variable : [MAJ]
Tag Variable : [MIN]
Step : [Checkout Source]
Step : [Update Applet Version]
Step : [Create Baseline]
Environment : [Basic Environment]
Class : [Production]
Filter : []
Selector : [Web Server]
Selector : [Local Server]
```

XML 파일의 모든 오브젝트를 가져오려면 다음 예제와 같이 가져올 오브젝트에 대한 옵션을 지정하십시오. 가져오기에서 오브젝트의 이름이 바뀝니다. 가져오기에 성공하면 성공 명령문이 표시됩니다.

```
C:\Program Files\IBM\Build Forge>bfimport -p -I -s -S -e -c -C -u -T -d -f
"samples\projects\basic.xml"
```

## 옵션 설명

옵션	설명
<file_name.xml>	가져올 Build Forge 오브젝트가 포함된 내보내기 XML 파일의 이름. XML 파일은 bfexport 명령 또는 .export 점 명령을 사용하여 작성해야 합니다. XML 파일 이름은 필수이며 현재 디렉토리(bfexport 명령을 실행하는 디렉토리)에 XML 파일이 없는 경우 경로 이름을 제공해야 합니다.  파일 이름에 공백이 포함되는 경우 이름을 따옴표로 묶어야 합니다.
-L	내보내기 XML 파일의 오브젝트와 오브젝트 이름을 나열합니다. 이 옵션을 단독으로 사용하면 다른 bfimport 옵션과 함께 지정해서는 안 됩니다. -L 옵션 출력은 stdout로 보내거나 XML 또는 텍스트 파일로 재지정할 수 있습니다.
-p	XML 파일에서 프로젝트 구성 데이터를 가져옵니다. 프로젝트 구성 데이터에는 태그 변수를 포함한 단계 및 프로젝트 정의 데이터가 포함됩니다.  이름을 바꿀 때 프로젝트 이름을 UI에 <project_name>_IMPORT_<number>로 가져옵니다.
-I	프로젝트 또는 단계 레벨에서 참조되는 체인 프로젝트 또는 라이브러리를 가져옵니다.  이름을 바꿀 때 체인 형식으로 연결된 프로젝트나 라이브러리를 UI에 <project_or_library_name>_IMPORT_<number>로 가져옵니다.
-S	UI에서 정의되는 선택기 오브젝트를 가져옵니다.  이름을 바꿀 때 선택기를 <selector_name>_IMPORT_<number>로서 UI로 가져옵니다.
-s	bfexport에 -s 옵션이 지정된 경우 관리 콘솔에서 정의된 서버 오브젝트를 가져옵니다.  이름을 바꿀 때 서버를 UI에 <server_name>_IMPORT_<number>로서 가져옵니다.  서버의 서버 권한 비밀번호는 내보내기 XML 파일에 포함되지 않으므로 서버 권한 비밀번호는 수동으로 입력해야 합니다.
-e	프로젝트 또는 단계 레벨에서 참조되는 환경과 해당 변수를 가져옵니다.  이름을 바꿀 때 환경을 <environment_name>_IMPORT_<number>로서 가져옵니다.
-c	프로젝트에서 참조하는 클래스를 가져옵니다.  이름을 바꿀 때 클래스를 <class_name>_IMPORT_<number>로서 가져옵니다.
-C	bfexport에 -C 옵션이 지정된 경우 프로젝트의 서버에 지정되는 콜렉터를 가져옵니다.  이름을 바꿀 때 콜렉터를 <collector_name>_IMPORT_<number>로서 가져옵니다.

옵션	설명
-u	<p>bfexport에 -g 옵션이 지정된 경우 이메일 알림을 수신하도록 지정된 액세스 그룹의 구성원인 사용자를 가져옵니다.</p> <p>이름을 바꿀 때 사용자에 대한 정보를 &lt;users&gt;_IMPORT_&lt;number&gt;로서 가져옵니다.</p>
-T	<p>bfexport에 -n 옵션이 지정된 경우 프로젝트와 단계에 지정된 사용자 작성 알림 템플릿을 가져옵니다.</p> <p>이름을 바꿀 때 알림 템플릿을 &lt;template_name&gt;_IMPORT_&lt;number&gt;로서 가져옵니다.</p>
-f	<p>bfexport에 -n 옵션이 지정된 경우 프로젝트 단계에 지정된 로그 필터를 가져옵니다.</p> <p>이름을 바꿀 때 로그 필터를 &lt;filter_name&gt;_IMPORT_&lt;number&gt;로서 가져옵니다.</p>
-d	<p>bfexport에 -L 옵션이 지정된 경우 UI에 정의된 LDAP 도메인 콜렉터를 가져옵니다.</p> <p>이름을 바꿀 때 로그 필터를 &lt;LDAP_domain_controller&gt;_IMPORT_&lt;number&gt;로서 가져옵니다.</p>
-r	<p>가져온 오브젝트의 이름이 아닌 오브젝트 자체를 바꿉니다. 기본적으로 가져온 오브젝트의 이름이 바뀌고 다음 이름 지정 규칙이 사용됩니다.</p> <p>&lt;object_name&gt;_IMPORT_&lt;number&gt;</p> <p>7.1 오브젝트의 경우 바꾸기 옵션을 지정하면 bfimport 명령이 UI에서 7.1 오브젝트에 대한 오브젝트를 겹쳐줍니다.</p> <p>7.0.2 이전 오브젝트의 경우 스냅샷 오브젝트가 바뀌지 않습니다. 이러한 오브젝트는 다음 이름 지정 규칙을 사용하여 이름이 바뀝니다.</p> <p>&lt;object_name&gt;_IMPORT_&lt;number&gt;</p> <p>중요사항: 이름보다는 UUID로 스냅샷 오브젝트의 교체를 평가합니다.</p> <p>세부사항은 510 페이지의 『가져오기에서 오브젝트 및 오브젝트 이름 바꾸기』의 내용을 참조하십시오.</p>

## 가져온 오브젝트에 액세스 그룹을 지정하는 방법

보안 액세스로 가져오기 시스템 설정은 가져온 오브젝트에 액세스 그룹을 지정하는 방법을 제어합니다.

- 보안 액세스로 가져오기가 예(기본값)로 설정되면 기본 보안 액세스 그룹 가져오기 설정이 액세스 그룹을 지정합니다. 기본값은 빌드 엔지니어입니다.
- 보안 액세스로 가져오기가 아니므로 설정되면 비보안 기본 액세스 그룹 가져오기 설정이 액세스 그룹을 지정합니다. 기본값은 개발자입니다.

비보안 기본 액세스 그룹 가져오기의 액세스 그룹이 지정되지 않으면 가장 최근에 작성된 액세스 그룹이 사용됩니다. 작성된 액세스 그룹이 없는 경우에는 기본 액세스 그룹이 사용됩니다.

## 액세스 그룹 지정 및 보안

보안 측정과 마찬가지로 Build Forge에서는 액세스 그룹을 직접 내보내거나 가져올 수 없습니다. 허용되는 경우에는 콘솔 호스트에 대한 액세스 권한이 있는 사용자는 내보낸 오브젝트의 XML 파일을 직접 처리한 후 가져올 수 있게 됩니다. 가져오기에 따라 액세스 그룹을 지정하면 권한이 부여된 사용자만 액세스 그룹에 지정됩니다.

## 가져오기에서 오브젝트 및 오브젝트 이름 바꾸기

bfimport 명령과 가져오기 유틸리티는 데이터베이스에 이름이 같은 오브젝트가 이미 있는 경우 가져온 오브젝트의 이름을 바꿉니다. 가져오기에서 오브젝트 이름 바꾸기는 기존 동작입니다.

가져오기에서 이 동작을 변경하고 기존 오브젝트를 바꾸려면 bfimport의 -r 옵션을 지정하거나 UI에서 엔티티 바꾸기 옵션을 선택해야 합니다.

다음 주제는 가져온 오브젝트와 오브젝트 이름을 바꿀 때 bfimport 명령과 가져오기 유틸리티가 사용하는 이름 지정 규칙을 설명합니다.

스냅샷 오브젝트(프로젝트, 선택기 및 환경)는 스냅샷 이름이 같은 경우 해당 이름을 유지하거나 가져오기에서 기본 스냅샷 이름이 지정됩니다.

### 7.1 이전 내보내기 파일에서 오브젝트 이름 바꾸기

7.1 이전 내보내기 파일의 오브젝트의 경우 bfimport 명령과 가져오기 유틸리티는 다음 이름 지정 규칙을 사용하여 UI에서 오브젝트 이름을 바꿉니다.

오브젝트 상태	UI 오브젝트 이름	UI 스냅샷 이름(스냅샷 오브젝트에만 적용)
새 오브젝트, 데이터베이스에 없음	<new_object_name>	<snapshot_name>
데이터베이스에 있음	<existing_object_name>_IMPORT_<n>	<snapshot_name>   Base Snapshot

스냅샷 오브젝트가 상위 스냅샷의 하위인 경우에도 새 상위 레벨 스냅샷으로 가져옵니다. 프로젝트, 선택기 및 환경만 스냅샷 오브젝트가 될 수 있습니다. 스냅샷 이름이 있는 경우 해당 이름을 유지합니다. 그렇지 않으면 표와 같이 기본 기본 스냅샷 이름이 지정됩니다.

## 7.1 이전 내보내기 파일에서 오브젝트 바꾸기

7.1 이전 내보내기 파일의 오브젝트의 경우 `bfimport` 명령과 가져오기 유틸리티는 다음 이름 지정 규칙을 사용하여 UI에서 오브젝트를 바꿉니다. 이름보다는 UUID로 스냅샷 오브젝트의 교체를 평가합니다.

오브젝트 상태	UI 오브젝트 이름	UI 스냅샷 이름(스냅샷 오브젝트에만 적용)
새 오브젝트, 데이터베이스에 없음	<code>&lt;new_object_name&gt;</code>	기본 스냅샷
데이터베이스에 있음, 비스냅샷 오브젝트	<code>&lt;existing_object_name</code>	n/a
데이터베이스에 있음, 스냅샷 오브젝트	<code>&lt;existing_object_name&gt;_IMPORT_&lt;n&gt;</code>	기본 스냅샷

스냅샷 오브젝트의 경우 스냅샷 오브젝트가 상위 스냅샷의 하위인 경우라도 새 상위 레벨 스냅샷으로서 가져옵니다. 프로젝트, 선택기 및 환경만 스냅샷 오브젝트가 될 수 있습니다. 7.1 이전 내보내기 파일에는 스냅샷 이름을 사용하는 오브젝트가 포함될 수 없으므로 표와 같이 기본 기본 스냅샷 이름이 지정됩니다.

## 7.1 내보내기 파일의 오브젝트 바꾸기

7.1 내보내기 파일의 오브젝트의 경우 `bfimport` 명령과 가져오기 유틸리티는 다음 이름 지정 규칙을 사용하여 UI에서 오브젝트를 바꿉니다.

오브젝트 상태	UI 오브젝트 이름	UI 스냅샷 이름(스냅샷 오브젝트에만 적용)
새 오브젝트, 데이터베이스에 없음	<code>&lt;new_object_name&gt;</code>	<code>&lt;snapshot_name&gt;</code>   Base Snapshot
데이터베이스에 있음	<code>&lt;new_object_name&gt;</code>	<code>&lt;snapshot_name&gt;</code>   Base Snapshot

이름보다는 UUID로 스냅샷 오브젝트의 교체를 평가합니다. 스냅샷 오브젝트가 상위 스냅샷의 하위인 경우에도 새 상위 레벨 스냅샷으로 가져옵니다. 프로젝트, 선택기 및 환경만 스냅샷 오브젝트가 될 수 있습니다. 모든 7.1 오브젝트는 고유한 스냅샷 이름을 갖거나 기본 기본 스냅샷 이름을 사용합니다.

## 스냅샷 가져오기 및 교체 시나리오

이름보다는 UUID로 스냅샷 오브젝트의 교체를 평가합니다. 스냅샷 오브젝트를 여러 번 가져올 경우 주의하십시오.

다음 시나리오에서 동작을 시연합니다.

1. 스냅샷 오브젝트를 시스템 A에서 내보내고 빈 시스템 B에 가져옵니다. 시스템 A의 가져온 오브젝트는 시스템 B에 가져올 때와 동일한 UUID를 갖고 있습니다.
2. 시스템 A에서 이름을 포함한 스냅샷 오브젝트 속성이 수정됩니다.
3. 스냅샷 오브젝트를 다시 시스템 A에서 내보내고 교체 옵션을 사용하여 시스템 B에 가져옵니다. UUID가 일치하므로 스냅샷 오브젝트가 교체됩니다. 시스템 B의 교체된 오브젝트는 변경된 이름을 포함하여 시스템 A에서 오브젝트의 모든 변경된 속성을 갖게 됩니다.



## 제 24 장 UI 구성 탭에서 웹 자원 링크

Build Forge UI에 탭을 추가하려면 **UI 구성** 탭을 사용하십시오.

**UI 구성** 탭을 사용하여 Build Forge 사용자 인터페이스에 탭을 추가할 수 있습니다. 새 탭마다 URL을 포함합니다. 탭을 사용하여 애플리케이션, 운영 체제, 서버 또는 사용자에게 대한 정보와 같이 외부 자원에 링크할 수 있습니다.

**참고:** 내부 Build Forge URL에 링크하는 탭은 작성하지 마십시오.

탭을 추가하려면 다음을 수행하십시오.

1. **UI 구성** 탭을 선택하십시오.
2. 탭 추가를 클릭하십시오.
3. 이름에 탭 제목 또는 이름을 입력하십시오.
4. 링크에서 다음을 입력할 수 있습니다.

옵션	사용법	예제
프로토콜 있는 URL	<ul style="list-style-type: none"><li>• 프로토콜이 http가 아닌 경우 필수</li><li>• http인 경우 선택적</li></ul>	http://www.ibm.com
프로토콜 없는 URL	기본 http 프로토콜 사용	www.ibm.com

5. 대상에서 하나의 옵션을 선택하십시오.
  - 내부(기본값): 기존 브라우저 창에서 링크를 열려면 선택합니다.
  - 외부: 새 브라우저 창에서 링크를 열려면 선택합니다.
6. 사용에서 하나의 옵션을 선택하십시오.
  - 사용(기본값): 탭을 사용하려면 선택합니다. 사용이면 탭을 선택할 때 URL에 연결이 시도됩니다.
  - 사용 불가능: 탭을 사용 불가능하게 하려면 선택합니다. 사용 불가능한 경우 탭은 숨겨집니다.
7. 가시성에서 하나의 옵션을 선택하십시오.
  - 참(기본값): UI에서 탭을 표시하려면 선택합니다.
  - 거짓: UI에서 탭을 숨기려면 선택합니다.
8. 루트 전용에서 하나의 옵션을 선택하십시오.
  - 참(기본값): 루트 사용자 또는 루트 액세스가 있는 Build Forge 사용자만 탭을 선택하고 URL 링크에 연결할 수 있게 합니다.

- 거짓: 모든 Build Forge 사용자가 탭을 선택하고 URL 링크에 연결할 수 있게 합니다.

**참고:** 새 탭을 저장하면 탭 제목이 목록에 나타납니다. 탭 자체가 나타나지 않으면 페이지를 새로 고치십시오.

---

## 제 25 장 Build Catalyst

Build Catalyst는 make 기반 C 및 C++ 소프트웨어 빌드를 가속합니다. 이는 기존 make 파일을 해석하고 분석하며 가속된 빌드를 실행합니다. 가속된 빌드는 병렬 빌드 및 분산 빌드와 같은 기술을 사용하여 전체 빌드 시간을 줄일 수 있습니다.

Build Catalyst는 IBM® Rational® Build Forge®와 함께 제공되지만 추가 설치가 필요합니다. Build Catalyst를 Rational Build Forge와 함께 사용하려면 Rational Build Forge 에이전트가 설치된 호스트에 Build Catalyst를 설치하십시오. 설치되고 구성된 Build Catalyst 가속화 빌드는 Rational Build Forge 단계에서 호출할 수 있습니다. Rational Build Forge 프로젝트를 사용하면 Rational Build Forge가 제공하는 더 광범위한 빌드 자동화 도구에 액세스할 수 있습니다. 예를 들어, 다른 빌드에 관련된 더 큰 시나리오의 일부로 개선된 make 기반 빌드를 호출할 수 있습니다. Rational Build Forge 프로젝트는 또한 사전 빌드 또는 사후 빌드 프로시저를 실행할 수도 있습니다.

Build Catalyst는 GNU Make 3.80에 맞게 작성된 make 파일과 잘 호환됩니다. GNU Make 3.80에 맞게 작성된 make 파일이 있으면 대부분의 경우 make 파일을 변경하지 않고 Build Catalyst를 사용하여 소스 코드를 빌드할 수 있습니다. 약간 변경해야 하는 경우도 있습니다. Build Catalyst는 다음 빌드를 지원합니다.

- 병렬 빌드: Build Catalyst는 비종속 대상을 식별하여 병렬로 빌드할 수 있습니다. 병렬로 빌드하면 빌드 컴퓨터에서 자원(예: 멀티코어 프로세서)의 사용이 좋아집니다. Build Catalyst는 자원을 효율적으로 사용하여 빌드를 가속하고 빌드 시간을 줄입니다.
- 분산 빌드: 분산 빌드는 병렬 빌드와 비슷하지만 빌드 활동이 여러 컴퓨터에 분산됩니다.

Build Catalyst 문서는 소프트웨어 빌드 개념에 익숙한 Build Catalyst 신규 사용자 및 경험 있는 사용자를 위한 것입니다.

---

### 지원되는 운영 체제

여러 Linux®, Solaris 및 Microsoft® Windows® 운영 체제에서 Build Catalyst를 사용할 수 있습니다.

Build Catalyst를 지원하는 운영 체제는 다음과 같습니다.

- Red Hat Enterprise Linux 4, 5
- Solaris 9, 10(SPARC)
- Microsoft Windows XP Professional

- Microsoft Windows Server 2003

참고: Linux 및 Solaris 운영 체제만 분산 빌드를 지원합니다.

## 설치 개요

Linux®, Solaris 또는 Microsoft® Windows® 운영 체제에 아카이브 파일에서 Build Catalyst를 설치합니다.

자세한 정보는 515 페이지의 『지원되는 운영 체제』의 내용을 참조하십시오.

Build Catalyst에는 IBM® Rational® Build Forge® 설치 프로그램과 별도의 설치 프로그램이 있습니다. Build Catalyst를 Rational Build Forge와 함께 사용하려면 Rational Build Forge 에이전트가 설치된 컴퓨터에 Build Catalyst를 설치하십시오. 빌드 호스트로 사용할 계획인 컴퓨터에 지원되는 운영 체제 중 하나가 실행 중인 경우 Build Catalyst를 설치할 수 있습니다. 주어진 컴퓨터에서 Rational Build Forge 작업이 Build Catalyst를 사용할 계획이 아닌 경우 Rational Build Forge 에이전트가 설치되어 있을 필요가 없습니다.

IBM은 다음 항목을 포함하는 아카이브 파일로 Build Catalyst를 배포합니다.

- 설치 가능한 모든 파일(2진 파일, 다양한 내장 make 파일 등)이 들어 있는 "덤프 파일"
- 설치 스크립트

## Linux 및 Solaris 운영 체제에 Build Catalyst 설치

Build Catalyst 아카이브 파일을 사용하여 Linux® 및 Solaris 운영 체제에 Build Catalyst를 설치합니다.

### 이 태스크 정보

분산 빌드에 대해 Build Catalyst를 사용하려면 분산 빌드에 사용된 모든 컴퓨터에 이를 설치하십시오.

### 프로시저

1. <http://hostname/buildcatalyst>에서 Build Catalyst 아카이브 파일을 다운로드하십시오. 여기서 *hostname*은 관리 콘솔 호스트입니다.
2. 아카이브 파일을 디렉토리에 추출하십시오.
3. 슈퍼유저가 아닌 경우 su를 입력하여 슈퍼유저가 되십시오.
4. 설치 프로그램을 추출한 디렉토리로 이동한 후 `install.sh` 파일이 들어 있는 서브디렉토리로 이동하십시오.
5. 그 디렉토리에서 `./install.sh`를 실행하십시오.

- a. 설치 디렉토리를 지정하십시오. (기본 디렉토리는 /opt/rational/buildforge/buildcatalyst입니다.)

설치가 이미 있는 경우(사용자가 지정한 위치가 아닌 위치에 있는 경우에도), 설치 프로그램은 설치를 진행하면 기존 설치가 겹쳐써지거나 렌더링되어 사용할 수 없게 된다고 경고합니다. (현재 설치 프로세스는 기본 설치 디렉토리에서 새로 지정된 설치 디렉토리로 기호 링크를 작성하므로 기본 디렉토리에 있지 않은 기존 설치는 렌더링되어 사용할 수 없게 될 수 있습니다.) 그런 다음 설치 프로그램은 설치 디렉토리에 대해 다시 프롬프트합니다. 설치 위치를 제공하면 설치는 지정된 위치에 파일을 씁니다.

- b. 분산 빌드에 사용할 rsh 또는 ssh 2진 파일을 지정하십시오(기본 프로그램은 rsh이며 해당 경로에 있습니다).

Build Catalyst 2진 파일은 설치 디렉토리의 bin 디렉토리에 rafmake로 설치됩니다. 설치는 다음 항목을 작성합니다.

- /opt/rational/buildforge/buildcatalyst에서 설치 디렉토리로의 기호 링크
- 설치 디렉토리의 etc/rafbe\_starter에서 쉘 2진 파일로의 기호 링크. 쉘 2진 파일은 rsh 또는 ssh입니다.

## 결과

이제 IBM® Rational® Build Forge® 작업에서 rafmake를 시작하여 C 또는 C++ 소프트웨어를 빌드할 수 있습니다. 또한 Rational Build Forge 작업에서 독립적으로 rafmake를 시작할 수도 있습니다.

## Windows에 Build Catalyst 설치

Build Catalyst 아카이브 파일을 사용하여 Microsoft® Windows®에 Build Catalyst를 설치합니다.

### 시작하기 전에

Microsoft Visual C++ 2005 SP1 Redistributable Package 버전 8.0.50727.762가 컴퓨터에 설치되어 있는지 확인하십시오. 제어판의 프로그램 추가/제거 항목을 사용하여 설치된 패키지를 확인할 수 있습니다.

**중요사항:** Build Catalyst는 이 패키지의 다른 버전은 사용하지 않습니다. 버전 8.0.50727.762가 설치되어 있어야 합니다.

필요한 경우 [www.microsoft.com/downloads/](http://www.microsoft.com/downloads/)에서 올바른 버전을 다운로드할 수 있습니다.

Windows 지원에 대한 자세한 정보는 515 페이지의 『지원되는 운영 체제』의 내용을 참조하십시오.

## 프로시저

1. `http://hostname/buildcatalyst`에서 Build Catalyst 아카이브 파일을 다운로드하십시오. 여기서 `hostname`은 관리 콘솔 호스트입니다.
2. 아카이브 파일을 폴더에 추출하십시오.
3. 추출한 폴더를 연 다음 `install.bat` 파일이 있는 작성된 폴더를 여십시오.
4. `install.bat`를 실행하십시오. Build Catalyst 2진 파일이 `rafmake.exe` 파일로 설치됩니다. 파일이 저장된 `bin` 폴더를 다음 표에 나타내었습니다.

표 12. 설치 폴더

운영 체제 유형	위치
32비트 운영 체제	C:\Program Files\IBM\Build Forge\buildcatalyst
64비트 운영 체제	C:\Program Files (x86)\IBM\Build Forge\buildcatalyst

## 결과

이제 IBM® Rational® Build Forge® 작업에서 `rafmake`를 시작하여 C 또는 C++ 소프트웨어를 빌드할 수 있습니다. 또한 Rational Build Forge 작업에서 독립적으로 `rafmake`를 시작할 수도 있습니다.

## Build Catalyst 예제

다음 예제를 사용하여 단순 빌드, 병렬 빌드 및 분산 빌드에 대해 Build Catalyst를 사용하는 방법을 더 잘 이해할 수 있습니다.

### 단순 빌드

단순 빌드(병렬 또는 분산이 아닌 빌드)의 경우, 다음과 같이 **rafmake** 명령을 실행하십시오.

```
rafmake [ -f makefile ] [ options ] ... [ targets ] ...
```

여기서

- *makefile*은 사용할 make 파일입니다.
- *options*는 521 페이지의 『rafmake 유틸리티 참조』에서 설명하는 옵션을 포함합니다.
- *targets*는 빌드할 대상 파일입니다.

다음 목록은 운영 체제별 예제를 제공합니다.

- Linux 및 Solaris

- 특정 `make` 파일과 모든 종속 항목을 기본 대상에 무조건 빌드합니다.

```
% rafmake -u -f project.mk
```

- 상세 출력과 함께 특정 `make` 파일에 특정 대상을 빌드합니다.

```
% rafmake -v -f project.mk a.out
```

- Windows

- `INCL_DIR` 매크로가 설정하는 특정 값을 사용하여 기본 `make` 파일에 기본 대상을 빌드합니다.

```
Y:\> rafmake INCL_DIR=C:\src\include
```

## 병렬 빌드

**rafmake** 명령은 병렬 빌드를 지원하며, 전체 빌드 시간을 줄이기 위해 동시에 빌드할 수 있는 독립 컴포넌트로 빌드를 분리합니다.

병렬 빌드를 실행하는 구문은 다음과 같습니다.

```
rafmake -J num [ -f makefile ] [ options ] ... [ targets ] ...
```

여기서

- *num*은 동시에 독립 대상을 빌드하는 동시 빌드 작업의 수입니다.
- *makefile*은 사용할 `make` 파일입니다.
- *options*는 521 페이지의 『**rafmake** 유틸리티 참조』에서 설명하는 옵션을 포함합니다.
- *targets*는 빌드할 대상 파일입니다.

병렬 빌드를 실행하면 자원을 효율적으로 활용할 수 있습니다(특히 멀티코어 프로세서의 경우). 병렬 빌드에서 얻어지는 성능은 빌드의 유형에 따라 다릅니다. 예를 들어, CPU 중심 빌드의 경우 최적의 동시 빌드 작업 수는 프로세서(또는 코어)의 수와 같습니다. 그보다 많이 지정하는 경우 성능은 별로 향상되지 않습니다. 그러나 입출력 중심 빌드의 경우에는 병렬 작업 수를 늘리면 빌드 시간이 줄어듭니다. 자원에 맞는 최적의 동시 빌드 작업 수를 판별하기 위해 여러 가지 수로 몇 가지 빌드를 시도해 보십시오.

다음 목록은 운영 체제별 병렬 빌드의 예제를 제공합니다.

- Linux 및 Solaris

- 세 가지 동시 빌드 작업으로 특정 `make` 파일에 병렬로 기본 대상을 빌드하십시오.

```
% rafmake -J 3 -f project.mk
```

- Windows

- 다섯 개의 동시 작업으로 INCL\_DIR 매크로가 설정하는 값을 사용하여 기본 make 파일에 병렬로 특정 대상을 빌드하십시오.

```
Y:\> rafmake -J 5 INCL_DIR=C:\src\include software.exe
```

## 분산 빌드

병렬 빌드는 한 컴퓨터의 자원만 사용할 수 있습니다. 빌드를 분산하면 병렬 빌드에서 한 단계 더 나아가 여러 컴퓨터에 빌드 로드가 분산됩니다.

**참고:** 분산 빌드에 대해 Build Catalyst를 사용하려면 분산 빌드에 사용된 모든 컴퓨터에 이를 설치하십시오.

빌드를 분산하려면 사용자 환경이 다음 요구사항을 충족해야 합니다.

- 여러 컴퓨터에서 소스 코드가 공유됩니다.

빌드를 분산할 모든 컴퓨터에서 동일한 경로로 동일한 소스 코드를 액세스할 수 있어야 합니다. 일반적으로 NFS 마운트 공유에 소스 코드가 있는 경우(또한 모든 컴퓨터에서 동일한 경로에 마운트한 경우) 모든 빌드 컴퓨터의 동일한 경로에서 동일한 소스 디렉토리를 사용할 수 있습니다.

- 선택한 원격 셸(rsh 또는 ssh)을 통한 로그인은 비밀번호가 필요하지 않도록 구성됩니다.

사용자가 호스트 컴퓨터에서 분산 빌드를 시작하는 경우 **rafmake** 명령은 빌드를 분산할 모든 컴퓨터에서 내부적으로 빌드 엔진(rafbe)을 시작합니다. 빌드 엔진은 Build Catalyst를 설치할 때 구성된 **ssh** 또는 **rsh** 명령을 사용하여 시작됩니다. /opt/rational/buildforge/buildcatalyst/etc/rafbe\_starter 파일은 사용자가 설치 중에 선택한 원격 셸의 기호 링크입니다. 빌드를 시작하는 사용자는 선택한 원격 셸을 사용하는 모든 컴퓨터에 비밀번호 없이 로그인할 수 있어야 합니다. 비밀번호 없이 로그인하면 빌드 엔진이 여러 컴퓨터에서 자동으로 시작될 수 있습니다. **rsh** 또는 **ssh** 명령 매뉴얼을 참조하여 비밀번호가 필요하지 않은 로그인을 설정하십시오.

- 빌드 호스트 파일은 컴퓨터에 빌드를 분산하는 방법을 설명합니다.

빌드 호스트 파일에는 일반적으로 많은 컴퓨터의 호스트 이름 또는 IP 주소가 있는 여러 행이 있습니다. 이러한 컴퓨터는 빌드를 분산할 위치입니다. 파일에는 다른 옵션(예: -idle num)을 설명하는 행도 있습니다. 이 유형의 행은 아래 나열되는 컴퓨터에서 스케줄되는 빌드에 대해 컴퓨터가 얼마나 대기 상태여야 하는지 지정합니다. 명시적으로 언급되지 않는 경우 대기는 50으로 가정됩니다. 다음은 빌드 호스트 파일의 예제입니다.

```
### Contents of build hosts file
host1.mydomain.com
-idle 25
host2.mydomain.com
host2.mydomain.com
```



```
192.168.0.200
-idle 75
host3.mydomain.com
### End of build hosts file
```

이 예제에서 빌드는 host1, host2, host3 및 192.168.0.200의 네 컴퓨터에 분산됩니다. host1.mydomain.com 컴퓨터의 대기는 명시적으로 언급되지 않았으므로 50으로 가정합니다. host2 및 192.168.0.200 컴퓨터의 경우 대기는 25로 지정됩니다. 즉 해당 컴퓨터 중 하나가 75% 미만으로 로드되거나 사용 중(또는 25% 이상 대기)인 경우 빌드는 이 컴퓨터에 스케줄됩니다. 대기 중인 컴퓨터가 25% 미만인 경우에는 해당 컴퓨터에서 빌드가 스케줄되지 않습니다. 또한 예제는 host2 컴퓨터를 두 번 나열합니다. 따라서 두 개의 병렬 빌드 스레드가 host2 컴퓨터에서 시작될 수 있습니다. 컴퓨터 구성에 따라 빌드에 가장 잘 맞는 자원을 사용하도록 빌드 호스트 파일을 수정할 수 있습니다.

빌드 호스트 파일을 작성한 후 다음 명령 구문을 사용하여 분산 빌드를 시작할 수 있습니다.

```
rafmake -J num -B build_hosts_file [ -f makefile ] [ options ] ...
[ targets ] ...
```

-J 옵션의 *num* 값은 모든 컴퓨터에 대한 빌드 스레드의 총 수를 지정합니다.

-B 옵션을 사용하지 않으면 **rafmake** 명령은 분산 빌드 대신 병렬 빌드를 수행합니다. -B 옵션과 올바른 빌드 호스트 파일을 지정하면 빌드 엔진은 빌드 호스트 파일에 지정된 여러 컴퓨터에 분산 빌드를 스케줄합니다.

## rafmake 유틸리티 참조

Build Catalyst의 인터페이스는 rafmake라는 make 유틸리티입니다. 이 인터페이스에는 동작을 제어하는 다양한 옵션이 있습니다.

명령의 구문은 다음 표에 나타난 것처럼 운영 체제에 따라 다릅니다.

표 13. 운영 체제별 옵션

운영 체제	옵션
Linux 및 Solaris(대상 빌드)	rafmake [ -f <i>makefile</i> ] ... [ -ukinservwdpqUN ] [ -J <i>num</i> ] [ -B <i>bldhost-file</i> ] [ -c <i>compat-mode</i> ] [ -C <i>change-directory</i> ] [ -A <i>BOS-file</i> ] ... [ <i>macro=value ...</i> ] [ <i>target-name ...</i> ]
Windows(대상 빌드)	rafmake [ -f <i>makefile</i> ] ... [ -ukinservwdpqUN ] [ -J <i>num</i> ] [ -c <i>compat-mode</i> ] [ -C <i>change-directory</i> ] [ -A <i>BOS-file</i> ] ... [ <i>macro=value ...</i> ] [ <i>target-name ...</i> ]
모두(rafmake 유틸리티의 버전 정보 표시)	rafmake { -ver/sion   -VerAll }

## 옵션 설명

**rafmake** 명령은 GNU 버전 3.80 make 명령 공통 옵션을 대부분 지원합니다. **rafmake** 명령은 추가 옵션도 지원합니다.

팁: 인수를 받지 않는 옵션(예: **-rNi**)을 결합할 수 있습니다.

표 14. 옵션 설명

옵션	설명
<b>-f makefile</b>	make 파일을 입력 파일로 사용합니다. 이 옵션을 생각하면 <b>rafmake</b> 명령은 현재 작업 디렉토리에서 makefile 및 Makefile이라는 입력 파일을 차례로 검색합니다. 둘 이상의 <b>-f makefile</b> 인수 쌍을 지정할 수 있습니다. 여러 입력 파일은 효율적으로 연결됩니다.
<b>-u</b>	(무조건) 다시 빌드해야 하는지 여부에 상관없이 모든 지정된 대상 및 모든 해당 종속 항목을 다시 빌드합니다. ( <b>-U</b> 를 참조하십시오.)
<b>-k</b>	실패하는 경우 현재 항목에 대한 작업을 중단하지만 해당 항목에 종속되지 않는 다른 대상에서 계속합니다.
<b>-i</b>	명령이 리턴하는 오류 코드를 무시합니다.
<b>-n</b>	(비실행) 대상의 make 파일에서 @으로 시작하는 행을 포함하여 다시 빌드해야 하는 명령행을 나열하지만 실행하지는 않습니다. 예외: \$(MAKE) 문자열을 포함하는 명령은 항상 실행됩니다.
<b>-s</b>	(자동) 실행하기 전에 명령행을 나열하지 않습니다.
<b>-e</b>	환경 변수가 makefile 파일의 매크로 지정을 대체합니다. (그러나 명령행 또는 빌드 옵션 스펙의 매크로=값 지정은 환경 변수를 대체합니다.)
<b>-r</b>	(규칙 없음) builddcatalyst-home-dir/etc/builtin.mk 파일(Linux 및 Solaris) 또는 builddcatalyst-home-dir\etc\builtin.mk 파일(Windows)에서 내장 규칙을 사용하지 않습니다. <b>-c</b> 옵션과 함께 사용되는 경우 <b>-r</b> 옵션은 또한 플랫폼 특정 시작 파일을 읽을 수 없도록 설정합니다. 자세한 정보는 <b>-c</b> 옵션을 참조하십시오.
<b>-v</b>	(상세) 기본 출력 모드보다 약간 더 상세합니다.
<b>-w</b>	(작업 디렉토리) make 파일을 실행하기 전후에 작업 디렉토리에 포함된 메시지를 인쇄합니다.
<b>-d</b>	(디버그) 빌드 중에 <b>rafmake</b> 명령이 읽는 환경 변수의 목록을 아주 상세히 포함합니다. make 파일을 디버깅하는 경우에만 이 옵션을 사용하십시오.
<b>-p</b>	(인쇄) 대상 특정 매크로 정의 및 목시적 규칙을 포함하여 모든 대상 설명 및 모든 매크로 정의를 나열합니다. 아무것도 실행하지 않습니다.
<b>-q</b>	(조회) make 파일 대상을 평가하지만 빌드 스크립트를 실행하지는 않습니다. <b>rafmake</b> 명령은 다음 응답 중 하나를 리턴합니다. <ul style="list-style-type: none"> <li>• 0: 대상이 최신인 경우</li> <li>• 1: 대상을 다시 빌드해야 하는 경우</li> </ul>
<b>-U</b>	목표 대상만 무조건 빌드합니다. 하위 대상을 빌드하지 않습니다. 명령행에 대상을 지정하지 않으면 기본 대상이 목표가 됩니다. ( <b>-u</b> 옵션은 목표 대상 및 빌드 종속 항목을 무조건 빌드합니다.)
<b>-N</b>	하나 이상의 빌드 옵션 스펙(BOS) 파일을 읽는 기본 프로시저를 사용하지 않습니다.

표 14. 옵션 설명 (계속)

옵션	설명
<code>-J num</code>	병렬 빌드 기능을 사용합니다. 현재 대상 다시 빌드의 최대수는 정수 <i>num</i> 으로 설정됩니다. <i>num</i> =0인 경우에는 병렬 빌드가 사용되지 않습니다. (이는 <code>-J</code> 옵션을 지정하지 않는 것과 동일합니다.) 또는 <i>num</i> 을 <code>RAFMMAKE_CONC</code> 환경 변수의 값으로 지정할 수도 있습니다.
<code>-B bldhost-file</code>	<i>bldhost-file</i> 파일을 병렬 빌드의 빌드 호스트 파일로 사용합니다. <code>-B</code> 옵션을 지정하지 않으면 <b>rafmake</b> 명령은 홈 디렉토리의 <code>.bldhost.\$RAFMMAKE_HOST_TYPE</code> 파일을 사용합니다. <code>-B</code> 옵션을 사용하는 경우에는 <code>-J</code> 옵션도 사용해야 하며 또는 <code>RAFMMAKE_CONC</code> 환경 변수를 설정해야 합니다.
<code>-c compat-mode</code>	(호환성) 호환성 모드로 <b>rafmake</b> 명령을 호출합니다.  (또는 <code>RAFMMAKE_COMPAT</code> 환경 변수를 사용하여 호환성 모드를 지정할 수도 있습니다.)  <i>compat-mode</i> 변수에는 다음 값 중 하나를 설정할 수 있습니다.  <b>gnu</b>  Free Software Foundation의 Gnu <b>make</b> 프로그램을 에뮬레이트합니다. 내장 작성 규칙을 정의하기 위해 <b>rafmake</b> 명령은 <code>builtin.mk</code> 대신 <code>gnubuiltin.mk</code> 를 읽습니다.  이 모드는 기본 호환성 모드입니다.  <b>std</b> IBM Rational® ClearCase® <b>clearmake</b> 명령 호환성 모드를 사용 가능하게 하고 <b>rafmake</b> 명령을 실행합니다.  Linux 및 Solaris 시스템에서만 <i>compat-mode</i> 환경 변수는 다음 값 중 하나를 가질 수 있습니다. <code>-c</code> 옵션은 Linux 및 Solaris 플랫폼 독립적입니다. 그러나 일부 모드는 시스템 특정 파일을 읽으려고 시도합니다. 해당 파일이 없는 경우 명령이 실패합니다.  <b>sun</b> SunOS 시스템이 제공하는 표준 <b>make(1)</b> 명령을 에뮬레이트합니다.  <b>aix</b> IBM AIX® 시스템이 제공하는 표준 <b>make(1)</b> 명령을 에뮬레이트합니다.
<code>-C dir</code>	디렉토리를 변경합니다. <b>rafmake</b> 명령은 빌드를 시작하기 전에 <i>dir</i> 디렉토리로 변경합니다.
<code>-A BOS-file</code>	기본적으로 읽는 파일 바로 다음에 읽을 BOS 파일을 지정합니다. 여러 BOS 파일을 지정하려면 이 옵션을 여러 번 사용하십시오.  이 옵션을 <code>-N</code> 과 함께 사용하는 경우 <b>rafmake</b> 는 기본 BOS 파일 대신 지정된 BOS 파일을 읽습니다.  또는 콜론으로 구분된 BOS 파일 경로 이름 목록(Linux 및 Solaris)이나 세미콜론으로 구분된 경로 이름 목록을 환경 변수 <code>RAFMMAKE_OPTS_SPECS</code> 의 값으로 지정할 수도 있습니다.
<code>-ver/sion</code>	<b>rafmake</b> 명령에 대한 버전 정보를 인쇄합니다.

표 14. 옵션 설명 (계속)

옵션	설명
-VerAll	<b>rafmake</b> 명령 및 <b>rafmake</b> 명령이 사용하는 라이브러리(Linux 및 Solaris) 또는 DLL(Windows)에 대한 버전 정보를 인쇄합니다.

## Build Catalyst 환경 변수

Build Catalyst는 사용을 단순화하는 여러 환경 변수를 지원합니다.

변수는 다음 섹션에 운영 체제별로 그룹화되어 있습니다.

- 『Linux, Solaris 및 Microsoft Windows에 공통인 환경 변수』
- 526 페이지의 『Linux 및 Solaris 전용 환경 변수』
- 526 페이지의 『Windows 전용 환경 변수』

### Linux, Solaris 및 Microsoft Windows에 공통인 환경 변수

#### RAFMAKE\_CONC

**rafmake** 빌드에 동시성 레벨을 설정합니다. 이 변수의 값은 -J 옵션의 값과 같습니다. **rafmake** 명령을 사용하는 경우 -J 옵션을 지정하면 이 변수의 설정을 겹쳐씁니다.

#### RAFMAKE\_COMPAT

**rafmake** 명령 호환성 모드를 지정합니다. 이 변수의 값은 -c 옵션의 값과 같습니다. **rafmake** 명령을 사용하는 경우 -c 옵션을 지정하면 이 변수의 설정을 겹쳐씁니다.

기본값: 없음

#### RAFMAKE\_MAKEFLAGS

**rafmake** 명령 옵션 지정의 대체 또는 보충 메커니즘을 지정합니다. RAFMAKE\_MAKEFLAGS 환경 변수에는 명령행 옵션에 사용된 키 문자와 동일한 문자열이 포함될 수 있지만 인수를 받는 옵션은 올바르지 않습니다. 충돌이 발생하는 경우 **rafmake** 명령행에 지정하는 옵션이 이 환경 변수의 설정을 겹쳐씁니다.

기본값: 없음

#### RAFMAKE\_OPTS\_SPECS

콜론(Linux 및 Solaris) 또는 세미콜론(Windows)으로 구분되는 경로 이름 목록을 제공합니다. 각 경로 이름은 **rafmake** 명령이 읽는 빌드 옵션 스펙(BOS) 파일을 지정합니다. **rafmake** 명령행에 하나 이상의 -A 옵션을 사용하여 BOS 파일을 지정하는 대신 이 변수를 사용할 수 있습니다.

기본값: 정의되지 않음

## RAFMAKE\_SHELL\_FLAGS

빌드 스크립트 명령을 실행하는 서브셸 프로그램으로 전달할 **rafmake** 명령 옵션을 지정합니다.

기본값:

- Linux 및 Solaris: `-e`
- Windows: 없음

## RAFMAKE\_SHELL\_REQUIRED

SHELL 매크로로 지정하는 셸 프로그램에서 빌드 스크립트를 실행하는 **rafmake** 명령을 강제 실행합니다. 셸 프로그램에서 **rafmake** 명령이 빌드 스크립트를 실행하도록 하려면 이 변수를 TRUE로 설정하십시오. **rafmake** 명령이 빌드 스크립트를 직접 실행하도록 구성하려면 변수를 설정 해제하십시오.

기본값: **rafmake** 명령이 빌드 스크립트를 직접 실행합니다.

## RAFMAKE\_VERBOSITY

다음과 같이 **rafmake** 명령 메시지 로깅 레벨을 지정합니다.

- 1 명령행의 `-v(verbose)`와 동일
- 2 명령행의 `-d(debug)`와 동일
- 0 또는 정의되지 않음  
표준 메시지 로깅 레벨과 동일

명령행에 `-v` 또는 `-d` 옵션도 지정하는 경우, 더 높은 값이 사용됩니다.

기본값: 0

## MAKEFLAGS

**make** 명령에 전달할 하나 이상의 플래그를 나열합니다. **rafmake** 명령은 시작 시에 MAKEFLAGS 환경 변수의 콘텐츠를 읽습니다. 그런 다음 명령은 명령행에 전달된 Build Catalyst에 특정하지 않은 플래그가 포함되도록 변수를 수정합니다. Build Catalyst에 특정한 플래그는 RAFMAKE\_MAKEFLAGS 환경 변수를 통해 전달되며, **rafmake** 명령은 MAKEFLAGS 변수에서 이러한 플래그를 발견하면 이들을 RAFMAKE\_MAKEFLAGS 변수로 이동합니다.

MAKEFLAGS 변수를 통해 전달되는 플래그는 다음과 같습니다. `-I`, `-p`, `-N`, `-w`, `-e`, `-r`, `-i`, `-k`, `-n`, `-q`, `-s`

RAFMAKE\_MAKEFLAGS 변수를 통해 전달되는 플래그는 다음과 같습니다. `-A`, `-B`, `-N`, `-v`, `-c`, `-U`, `-M`, `-u`, `-d`

기본값: 없음

## Linux 및 Solaris 전용 환경 변수

### RAFMAKE\_BLD\_HOSTS

**rafmake** 명령이 대상을 빌드할 수 있는 하나 이상의 빌드 호스트를 지정합니다.

기본값: 정의되지 않음

### RAFMAKE\_HOST\_TYPE

병렬 빌드(-J 옵션) 중에 사용할 빌드 호스트 파일의 이름을 결정합니다. 홈 디렉토리의 `.bldhost.$RAFMAKE_HOST_TYPE`. (홈 디렉토리는 비밀번호 데이터베이스를 조사하여 결정됩니다.)

명령행에 -B 옵션을 지정하면 이 변수 설정을 겹쳐씹니다.

C 셸 사용자: `.login` 파일이 아니라 `.cshrc` 파일에 이 변수를 설정하십시오. 병렬 빌드 기능은 `.login` 파일을 읽지 않는 원격 셸을 호출합니다.

또한 `RAFMAKE_HOST_TYPE`을 **make** 매크로로 코딩할 수도 있습니다.

기본값: 없음

## Windows 전용 환경 변수

### RAFMAKE\_NO\_ESC\_PATT\_CHARS

**rafmake** 명령의 GNU 호환 모드에서 이스케이프 문자(\)를 대체합니다. 예를 들어, **rafmake**와 GNU **make** 명령 둘 다 \% 문자열이 리터럴 문자 %를 표시한다고 가정합니다. 명령은 규칙을 패턴 규칙으로 취급하지 않습니다. **rafmake** 명령이 이스케이프 문자를 사용하여 리터럴 문자를 표시하지 않도록 하려면 이 환경 변수를 `널`이 아닌 `값`으로 설정하십시오.

### RAFMAKE\_PNAME\_SEP

**rafmake** 명령이 생성하는 경로 이름의 경로 이름 분리문자를 설정합니다. 이 변수는 `make` 파일, `BOS` 파일, 명령행에 설정되거나 환경 변수로 설정될 수 있습니다.

기본값: 이 변수가 설정되지 않거나 슬래시(/) 또는 백슬래시(\)가 아닌 값으로 설정된 경우 **rafmake** 명령은 백슬래시(\)를 경로 이름 분리문자로 사용합니다.

---

## 제 26 장 Rational Build Forge 통합 방법

다음 방법을 사용하여 Build Forge를 기타 제품과 통합합니다.

- 명령행을 사용하여 통합. Rational Build Forge 단계는 에이전트 호스트에서 명령을 실행합니다.
- 소스 제어 관리 시스템과의 통합을 돕는 특수 기능이 있는 XML 기반 프로그램인 어댑터를 사용하여 통합.
- IDE 플러그인을 사용하여 통합. 이를 통해 IDE 사용자 인터페이스로 콘솔에 액세스할 수 있습니다.
- Websphere 제품과 통합. Apache Tomcat 대신 Websphere Application Server를 웹 애플리케이션 서버로 사용할 수 있습니다. Apache HTTP 서버 대신 IBM HTTP 서버를 사용할 수 있습니다.

---

### 명령행 인터페이스를 사용하여 통합

Build Forge는 명령행 인터페이스를 포함하는 애플리케이션을 쉽게 통합합니다.

이 종류의 통합을 설정하려면 다음을 수행하십시오.

1. 호스트에서 Build Forge 콘솔을 설치하십시오.
2. 애플리케이션 호스트 또는 애플리케이션에 액세스할 수 있는 호스트에 Build Forge 에이전트를 설치하십시오.
3. 콘솔에서 서버 자원 및 서버 인증을 작성하십시오. 설치한 Build Forge 에이전트에 액세스하도록 서버 자원을 구성하십시오.
4. 애플리케이션에서 실행할 명령에 필요한 대로, 에이전트 및 호스트 환경을 구성하십시오. 이는 실행 중인 에이전트에서 해당 PATH가 올바르게 설정됨을 보장하면서 Build Forge에서 사용할 사용자 계정을 작성하는 작업만큼 단순합니다. 또한 애플리케이션에서 명령을 실행하는 데 사용되는 클라이언트를 설치 및 구성해야 할 수도 있습니다. 예를 들어 Rational ClearCase 및 ClearQuest에서는 클라이언트 애플리케이션을 사용하여 명령을 실행해야 합니다.

설정이 완료되면 Build Forge에서 작성되는 프로젝트에 애플리케이션 명령을 실행하는 단계를 포함시킬 수 있습니다. 단계의 패스 또는 실패 상태를 기반으로 프로젝트 실행을 제어하거나 로그 필터를 설정하여 명령이 리턴하는 출력 패턴의 Build Forge 로그를 스캔할 수 있습니다. 애플리케이션은 프로젝트 논리에서 추가 설정이 필요할 수 있습니다. 예를 들어, ClearCase와 통합하려면 보기의 작성, 시작 및 채우기를 포함시켜야 합니다.



조건 단계 및 루프 단계를 사용하여 특정 명령의 성공 또는 실패에 대해 발생하는 일을 제어할 수 있습니다.

소스 제어 애플리케이션과의 일반 사용은 빌드 방지입니다. 명령은 빌드가 마지막으로 실행된 후 소스 코드 업데이트가 체크인되었는지 여부를 조회하기 위해 실행됩니다.

- 업데이트되지 않으면 빌드가 실행되지 않습니다.
- 업데이트된 경우 빌드가 실행됩니다.

모듈 종속성은 프로젝트 단계의 실행 논리로 표현될 수 있습니다. 특정 모듈의 빌드가 기타 모듈 또는 전체 소프트웨어 프로젝트의 빌드를 트리거하는지 여부를 제어할 수 있습니다.

알림 템플릿을 사용하여 프로젝트 멤버 그룹이 빌드 성공 또는 실패에 대해 알림을 받는 방법을 제어합니다. 연속 통합 환경에서 문제점 코드를 해결할 수 있도록 실패에 대해 알리는 것이 중요합니다. 명령행 통합을 사용하면 프로젝트 멤버의 전체 그룹에 알리는 것만 가능합니다. 프로젝트 멤버는 프로젝트의 단계 로그를 검사하여 실패의 원인이 되는 코드를 판별해야 합니다.

---

## 어댑터를 사용하여 통합

어댑터를 사용하여 Rational Build Forge를 외부 애플리케이션과 통합할 수 있습니다.

IBM Rational ClearCase 및 IBM Rational ClearQuest와의 통합을 위해 샘플 어댑터 템플릿이 제공됩니다. 사용자 정의 어댑터를 작성할 수 있는 어댑터 툴킷은 별도 라이선스 기능입니다.

어댑터의 동작은 포함된 DTD에 의해 요소가 지정되는 XML 파일에 의해 정의됩니다. Build Forge는 프로젝트 단계와 연합하여 어댑터를 실행합니다.

어댑터는 통합 기반 빌드를 위한 추가 도구를 제공합니다.

- 내부 조건 논리: 내부 변수의 설정을 기반으로 합니다.
- 명령 정의: 애플리케이션 명령은 명령문 및 변수를 기반으로 빌드할 수 있습니다.
- 응답 스캔: 각 명령에 대한 응답에서 스캔할 패턴을 정의할 수 있습니다.
- 동적 알림 그룹: 알림은 애플리케이션에서 생성된 데이터를 기반으로 할 수 있습니다. 예를 들어, 알릴 팀 구성원 그룹은 코드 변경사항을 체크인한 구성원만을 기반으로 빌드될 수 있습니다.
- 스크립트 실행: 어댑터는 Build Forge 호스트에서 스크립트를 실행할 수 있습니다. 이는 애플리케이션 호스트에서 실행되는 스크립트 및 명령과 독립적일 수 있습니다.



## 어댑터 요구사항

이 주제에서는 Build Forge 어댑터에 대한 설치 및 구성 요구사항과 소프트웨어 요구사항을 식별합니다.

환경이 Build Forge 제품과 함께 제공되는 어댑터 사용 요구사항을 충족하는지 확인하십시오.

- 외부 애플리케이션이 실행 중인 컴퓨터에 Build Forge 에이전트를 설치하십시오. 일부 제품에는 추가 설정 요구사항이 있습니다. Rational Clear Case 및 Rational ClearQuest를 사용하려면 추가 설정이 필요합니다. 이러한 제품에 대한 정보는 통합 섹션을 참조하십시오.
- ClearCase 또는 ClearQuest 이외의 다른 애플리케이션 템플리트를 사용할 경우 Build Forge 어댑터 툴킷용 라이선스 키를 설치하십시오.
- Build Forge 제품이 지원하는 외부 애플리케이션 버전을 실행하십시오.

표 15. 어댑터에 지원되는 애플리케이션 버전

애플리케이션	버전
Rational ClearCase	6.0 이상
Rational ClearQuest	7.0 이상
CVS	1.1, 1.2
Microsoft Visual SourceSafe	6.0
Perforce	2005.1
StarTeam	2005 릴리스 2
하위 버전	1.3.1 이상

## 어댑터

이 주제에서는 어댑터에 대한 몇 가지 일반 정보를 제공합니다. 또한 어댑터가 기타 Build Forge 오브젝트 및 기능과 상호작용하는 방법을 설명합니다.

이 절에 제공된 정보를 검토하여 어댑터를 시작하십시오.

### 어댑터에 대한 정보

어댑터는 외부 애플리케이션에 대한 인터페이스입니다. 어댑터를 사용하여 Build Forge 프로젝트는 외부 애플리케이션과 정보를 교환하여 목표를 달성할 수 있습니다.

예를 들어, 소스 코드 애플리케이션용 어댑터는 Build Forge 프로젝트 실행의 전제조건으로 저장소에서 소스 코드 변경사항을 확인합니다. 소스 코드 파일이 변경된 경우, 프로젝트가 실행됩니다. 변경사항이 없는 경우에는 프로젝트가 실행되지 않습니다.

**어댑터 패널에 대한 정보:** 어댑터를 작성하고 편집하려면 어댑터 패널을 사용하십시오. 패널을 보려면 프로젝트 > 어댑터를 선택하십시오.

IBM UI 구성 콘솔 보고서 로그아웃: Root User

어댑터
어댑터 추가
도움말

필터
1 표시 중 - 1/1
모두 표시

<<
<
페이지
1/1
>
>>

이름	소스	유형
MyAdaptor		

새 어댑터
어댑터 저장
어댑터 복사
어댑터 삭제

**세부사항**

이름:

유형:

소스

템플릿:

-- 없음 --

액세스:

개발자

인터페이스:

패널에는 다음과 같은 필드가 있습니다.

이름    공백이 없는 고유한 어댑터 이름

유형    어댑터 유형

템플릿

Rational Build Forge와 함께 포함된 어댑터 템플릿 중 하나 또는 전혀 없음

참고: Rational Build Forge 버전을 업데이트하면 이 설정이 지워집니다.

액세스 어댑터 보기 및 편집을 그룹 구성원으로 제한하기 위한 액세스 그룹

인터페이스

사용자가 제공하거나 템플릿에서 제공되는 XML 인터페이스

## 어댑터 템플릿

어댑터는 어댑터 템플릿의 인스턴스입니다. 어댑터를 작성할 때 어댑터에 고유 이름을 지정하고 템플릿과 연관시킵니다.

템플릿은 XML 파일입니다. XML은 정보, 정보 분석 지시사항 및 BOM 보고서에 결과를 표시하기 위한 형식화 세부사항을 수집하기 위한 애플리케이션 명령을 포함합니다.

Build Forge가 제공하는 템플릿은 수정없이 사용하도록 설계되었습니다. 그러나 템플릿을 수정하거나 템플릿을 새 어댑터 템플릿을 작성하기 위한 모델로 사용할 수 있습니다.

어댑터 템플리트는 다음 디렉토리에 설치됩니다.

Microsoft Windows 운영 체제를 실행 중인 컴퓨터의 경우 *bfinstall\interface*

UNIX 또는 Linux 운영 체제를 실행 중인 컴퓨터의 경우 *bfinstall/Platform/interface*

### 샘플 어댑터 템플리트

여러 애플리케이션 유형(소스 코드, 결함 추적, 테스트 및 패키징)에 샘플 어댑터 템플리트가 제공됩니다. 어댑터는 어댑터가 지원하는 애플리케이션 유형에 따라 분류됩니다.

Build Forge 제품은 다음 표의 애플리케이션에 샘플 어댑터 템플리트를 제공합니다. ClearCase 및 ClearQuest용 템플리트는 별도의 라이선스 키를 필요로 하지 않습니다. 기타 애플리케이션 템플리트에는 Build Forge 어댑터 툴킷을 통해 라이선스가 제공됩니다.

어댑터 템플리트 XML 파일이 .xml 파일 확장자로 다음 디렉토리에 나열됩니다.

Microsoft Windows 운영 체제를 실행 중인 컴퓨터의 경우 *bfinstall\interface*

UNIX 또는 Linux 운영 체제를 실행 중인 컴퓨터의 경우 *bfinstall/Platform/interface*

### 참고:

프로젝트가 어댑터 링크를 통해 접속된 어댑터로 실행되고 어댑터가 디버그 모드에 있지 않으며 어댑터 단계에 실패하는 경우 어댑터가 작업을 제거합니다.

표 16. 어댑터 템플리트 설명

어댑터 템플리트 이름	설명	유형
ClearCaseBaseline	ClearCase 보기에서 디렉토리를 스캔합니다.  ClearCase가 보고하는 분기 및 버전 정보를 BOM 보고서에 기록합니다.	소스
ClearCaseByBaselineActivities	ClearCase 보기의 콘텐츠에서 새 기준선을 작성합니다.  새 기준선과 이전 어댑터 실행의 기준선을 비교하여 변경 활동을 식별합니다.  각 변경 활동에 대해 BOM 보고서에 활동, 변경된 파일, 사용자, 날짜, 주석 및 버전 정보를 기록합니다.  변경된 각 파일에 대해 diff 명령 출력의 변경 세부사항을 BOM 보고서에 기록합니다.	소스

표 16. 어댑터 템플릿 설명 (계속)

어댑터 템플릿 이름	설명	유형
ClearCaseByBaselineVersions	<p>ClearCase 보기의 콘텐츠에서 새 기준선을 작성합니다.</p> <p>새 기준선과 이전 어댑터 실행의 기준선을 비교하여 변경된 파일을 식별합니다.</p> <p>변경된 각 파일에 대해 BOM 보고서에 파일 이름, 버전, 날짜, 사용자 및 주석 정보를 기록합니다.</p> <p>변경된 각 파일에 대해 diff 명령 출력의 변경 세부사항을 BOM 보고서에 기록합니다.</p>	소스
ClearCaseByDate	<p>ClearCase 보기에서 두 날짜 간의 변경사항을 조회합니다. 기본 날짜는 현재 시간소인 및 이전 어댑터 실행의 시간소인입니다.</p> <p>변경된 각 파일에 대해 BOM 보고서에 파일 이름, 버전, 날짜, 사용자 및 주석 정보를 기록합니다.</p> <p>변경된 각 파일에 대해 diff 명령 출력의 변경 세부사항을 BOM 보고서에 기록합니다.</p>	소스
ClearCaseByLabel	<p>새 레이블을 작성하여 ClearCase 보기의 콘텐츠에 적용합니다.</p> <p>새 레이블과 이전 어댑터 실행의 레이블을 비교하여 변경된 파일을 식별합니다.</p> <p>변경된 각 파일에 대해 BOM 보고서에 파일 이름, 버전, 날짜, 사용자 및 주석 정보를 기록합니다.</p> <p>변경된 각 파일에 대해 diff 명령 출력의 변경 세부사항을 BOM 보고서에 기록합니다.</p>	소스
ClearQuestBaseClearCaseByDate	<p>ClearCase 보기에서 두 날짜 간의 변경사항을 조회합니다. 기본 날짜는 현재 시간소인 및 이전 어댑터 실행의 시간소인입니다.</p> <p>변경된 각 파일에 대해 ClearQuest 변경 ID를 식별하는 CrmRequest 하이퍼링크 속성을 검색합니다. ClearQuest 상태가 결함 레코드 분석을 허용하는 경우 ClearQuest에서 결함 레코드를 분석하기 위해 작업 정보를 추가하여 변경 ID 분석을 시도합니다.</p> <p>변경된 각 파일에 대해 BOM 보고서에 파일 이름, 결함 ID, 결함 상태 및 ClearQuest 오류 정보를 기록합니다.</p>	결함
ClearQuestClearCaseByActivity	<p>ClearCase 활동 목록과 연관된 ClearQuest 결함 레코드를 찾습니다. 발견된 각 결함 레코드에 대해 ClearQuest 상태가 결함 레코드 분석을 허용하는 경우 ClearQuest에서 결함 레코드를 분석하기 위해 작업 정보를 추가합니다.</p> <p>BOM 보고서에 ClearCase 활동 ID와 연관된 파일 및 ClearQuest 결함 상태 정보를 기록합니다.</p>	결함
ClearQuestUCMClearCaseByDate	<p>ClearCase 보기에서 두 날짜 간의 변경사항을 조회합니다. 기본 날짜는 현재 시간소인 및 이전 어댑터 실행의 시간소인입니다.</p> <p>Rational 통합 변경 관리(UCM)를 사용하여 결과를 생성합니다.</p> <p>변경된 각 파일에 대해 BOM 보고서에 파일 이름, 결함 ID, 결함 상태 및 ClearQuest 오류 정보를 기록합니다.</p>	결함

표 16. 어댑터 템플릿 설명 (계속)

어댑터 템플릿 이름	설명	유형
CVSv1Baseline	Build Forge 에이전트에서 CVS 디렉토리를 스캔하여 변경된 파일을 찾습니다.  BOM 보고서에 변경된 파일 이름, 상태, 작업 버전, 저장소 버전 및 sticky 태그 정보를 기록합니다.	소스
CVSv1ByDate	CVS 보기에서 두 날짜 간의 변경사항을 조회합니다. 기본 날짜는 현재 시간소인 및 이전 어댑터 실행의 시간소인입니다.  BOM 보고서에 변경 유형, 날짜, 사용자 이름, 버전 및 파일 이름 정보를 기록합니다.  변경된 각 파일에 대해 diff 명령 출력의 변경 세부사항을 BOM 보고서에 기록합니다.	소스
CVSv1ByTag	CVS 모듈에 새 태그를 적용합니다.  새로 태그 지정된 모듈과 이전 어댑터 실행 중에 태그 지정된 모듈 간의 차이점을 비교합니다.  BOM 보고서에 파일 이름, 개정, 상태, 날짜, 시간, 변경 작성자 및 커밋 주석 정보를 기록합니다.  변경된 각 파일에 대해 diff 명령 출력의 변경 세부사항을 BOM 보고서에 기록합니다.	소스
CVSv2ByDate	CVS 보기에서 두 날짜 간의 변경사항을 조회합니다. 기본 날짜는 현재 시간소인 및 이전 어댑터 실행의 시간소인입니다.  BOM 보고서에 변경 유형, 날짜, 사용자 이름, 버전 및 파일 이름 정보를 기록합니다.  변경된 각 파일에 대해 diff 명령 출력의 변경 세부사항을 BOM 보고서에 기록합니다.	소스
JazzJBE	Rational Team Concert 3.x 이상 사용자를 위한 것입니다. 정보는 Rational Team Concert 문서를 참조하십시오.	소스
JazzSCM	Rational Team Concert에서 변경된 소스 파일을 조회합니다.  BOM 보고서에 소스 파일 변경, 변경된 파일 크기, 컴포넌트 변경 및 changeset 정보를 기록합니다.	소스
JazzSCMv2	Rational Team Concert 버전 2.x에서 변경된 소스 파일을 조회합니다.  BOM 보고서에 변경된 소스 파일, 변경된 스트림, 변경된 컴포넌트, changeset, 변경한 사용자, 체크인 주석 및 Jazz 작업 항목에 대한 정보를 기록합니다.	소스
JUnit	JUnit 테스트를 실행합니다.  BOM 보고서에 테스트 실행, 실패, 오류 및 경과 시간 정보를 씁니다.	테스트

표 16. 어댑터 템플릿 설명 (계속)

어댑터 템플릿 이름	설명	유형
PerforceByDate	Perforce 클라이언트에서 어댑터 실행 이후로 발생한 변경사항을 조회합니다.  BOM 보고서에 변경, 날짜, 시간, 사용자, Perforce 클라이언트 및 주석 정보를 기록합니다.  diff 명령 출력의 변경 세부사항을 BOM 보고서에 기록합니다.	소스
PerforceByRev	Perforce 클라이언트에서 마지막 저장소 개정 이후로 발생한 변경사항을 조회합니다.  BOM 보고서에 변경, 날짜, 시간, 사용자, Perforce 클라이언트 및 주석 정보를 기록합니다.  diff 명령 출력의 변경 세부사항을 BOM 보고서에 기록합니다.	소스
Quota	Windows 폴더를 조회하여 하위 폴더가 지정된 임계값 크기를 초과하는지 여부를 판별합니다.  각 하위 폴더에 대해 BOM 보고서에 폴더 크기, 소유자 및 마지막 수정 날짜 정보를 기록합니다.  BOM 보고서에 임계값 크기를 초과한 하위 폴더 목록을 기록합니다.  중요사항: 초과된 디렉토리를 소유하는 사용자에게 처음에는 Build Forge 시스템 내의 이름순으로, 그 다음에는 사용자 이름순으로 알림을 보냅니다. 알림을 받을 사용자가 없는 경우 <알림>에 실패합니다.	소스
StarTeamBaseline	StarTeam 보기용 폴더를 조회하여 파일에 대한 정보를 수집합니다.  BOM 보고서에 파일 이름, 상태, 개정 및 분기 정보를 기록합니다.	소스
StarTeamByDate	StarTeam API를 사용하여, StarTeam 보기를 조회하여 현재 날짜와 이전 어댑터 실행 간의 변경사항을 식별합니다.  BOM 보고서에 변경된 파일과 디렉토리, 사용자, 버전, 날짜 및 변경 주석 정보를 기록합니다.  diff 명령 출력의 변경 세부사항을 BOM 보고서에 기록합니다.	소스
SubversionByDate	Subversion에서 과거 날짜와 현재 날짜 간에 발생한 저장소 변경사항을 조회합니다.  BOM 보고서에 변경 유형, 개정, 사용자, 파일 또는 디렉토리와, 변경 날짜 정보를 기록합니다.  BOM 보고서에 파일 이름, 상태, 개정 및 분기 정보를 기록합니다.	소스
SubversionByRev	Subversion에서 현재 개정과 이전 개정 간에 발생한 저장소 변경사항을 조회합니다.  각 변경에 대해 BOM 보고서에 개정, 사용자, 변경 유형, 파일 또는 디렉토리 경로와, 변경 날짜 정보를 기록합니다.  diff 명령 출력의 변경 세부사항을 BOM 보고서에 기록합니다.	소스

표 16. 어댑터 템플릿 설명 (계속)

어댑터 템플릿 이름	설명	유형
SynergyByDate	<p>Rational Synergy에서 두 날짜 사이의 변경사항을 조회합니다. 기본 날짜는 현재 시간소인 및 이전 어댑터 실행의 시간소인입니다.</p> <p>변경된 각 파일에 대해 BOM 보고서에 파일 이름, 버전, 사용자, 프로젝트 및 날짜를 씁니다.</p> <p>변경된 각 파일에 대해 diff 명령 출력의 변경 세부사항을 BOM 보고서에 기록합니다.</p>	소스
VSSByDate	<p>Visual Source Safe 디렉토리에서 이전 날짜와 현재 날짜 간의 변경사항을 조회합니다.</p> <p>BOM 보고서에 프로젝트 및 파일에 대한 변경 정보(프로젝트 또는 파일, 버전, 사용자, 날짜, 시간, 프로젝트 활동, 파일 프로젝트 및 조치)를 기록합니다.</p> <p>diff 명령 출력의 변경 세부사항을 BOM 보고서에 기록합니다.</p>	소스

## 어댑터 및 프로젝트

어댑터 코드를 실행하고 외부 애플리케이션과 인터페이스하려면 Build Forge 프로젝트에 어댑터를 추가해야 합니다.

어댑터는 점 명령 또는 어댑터 링크를 사용하여 프로젝트에 추가됩니다.

애플리케이션 유형에 점 명령(.source, .defect, .test 또는 .pack)을 사용하여 프로젝트에 어댑터를 추가할 수 있습니다.

어댑터 링크를 사용할 경우 프로젝트에 소스 코드 어댑터만 추가할 수 있습니다. 어댑터 링크는 어댑터를 프로젝트에 연결하는 데 .source 명령 대신 사용됩니다.

## 어댑터 및 환경 변수

애플리케이션 명령을 실행하려면 어댑터에 환경 변수가 필요합니다. 어댑터 템플릿에서 환경 변수는 XML 파일의 <template> 섹션에 있는 <env> 요소에 나열됩니다.

예를 들어, ClearCaseBaseline 어댑터의 경우 ClearCaseBaseline.xml 파일에 다음 환경 변수가 나열됩니다.

```
<template>
<!-- Template section, these are parsed out of the final xml.
Use the list below to help identify the variables needed to run this interface
if you are integrating it during a regular BuildForge step.
-->
<env name="VIEW" value="my_adaptor_view" />
<env name="VOB_PATH" value="\AdaptorVob" />
<env name="CCSERVER" value="BFServerName" /></template>
```

Build Forge에서 환경 변수는 환경에 저장됩니다. 어댑터를 작성하기 전에 애플리케이션 환경 변수의 환경을 작성하십시오.

## 어댑터 및 알림

대부분의 어댑터 템플리트는 사용자에게 이메일 알림을 보냅니다. 예를 들어, ClearCaseByDate 어댑터가 실행될 때 소스 코드 파일을 변경한 사용자에게 이메일 알림을 보냅니다. 파일이 변경되지 않은 경우, 실패 이메일 알림을 보냅니다.

어댑터 알림을 선택적으로 수정할 수 있습니다.

- 어댑터 템플리트에서 <adduser> 요소를 복제하여 어댑터 알림 그룹에 사용자를 추가하십시오.
- 어댑터 템플리트에서 <notify> 요소를 사용하여 알림 메시지를 추가하거나 삭제하십시오.
- 어댑터 프로젝트의 경우, 프로젝트 레벨 알림을 설정하십시오.
- 어댑터 점 명령 단계의 경우, 단계 레벨 알림을 설정하십시오.

**중요사항:** <notify> 지시문이 실패하면(예를 들어 이메일 주소로 지정된 사용자가 존재하지 않는 경우), XML이 실패하고 모든 후속 알림이 실패합니다.

## 어댑터 및 작업 실행

어댑터 점(dot) 명령을 사용하는 어댑터 프로젝트를 스케줄된 작업으로 시작하거나 프로젝트에 수동 시작 옵션을 사용하여 시작할 수 있습니다.

어댑터 링크 프로젝트는 일반적으로 스케줄대로 실행됩니다. 그러나 몇 가지 추가 설정을 완료하면 어댑터 링크 프로젝트를 수동으로 시작할 수 있습니다. 547 페이지의 『어댑터 링크 프로젝트 수동 시작』의 내용을 참조하십시오.

## 어댑터 및 작업 결과

어댑터 점 명령 프로젝트의 경우, 다음과 같이 단계 로그 또는 BOM 보고서에서 작업 결과를 보십시오.

- 작업 > 완료를 선택하십시오. 작업 태그를 선택하여 단계 로그를 보십시오.
- 작업 > 완료를 선택하십시오. BOM 탭을 선택하여 BOM 보고서를 표시하고 범주별로 작업 결과를 보십시오.

어댑터 링크 프로젝트의 경우, 다음과 같이 BOM 보고서의 소스 변경사항 범주에서 작업 결과를 보십시오.

- 작업 > 완료를 선택하십시오. BOM 탭을 선택하십시오. BOM 보고서에서 소스 변경사항 범주를 찾으십시오.

## 어댑터 링크에 대한 정보

어댑터 링크는 어댑터를 프로젝트에 연결하고 환경을 어댑터와 연관시킵니다.

어댑터 링크는 오직 소스 코드 어댑터와 작동합니다. 점 명령을 사용하여 임의의 어댑터 유형을 프로젝트에 연결할 수 있습니다.



어댑터 링크의 기능은 다음과 같습니다.

- 단계 0으로 프로젝트에 어댑터 코드를 추가합니다(기타 프로젝트 단계를 실행하기 전에 소스 코드 변경사항을 테스트).
- 애플리케이션 환경 변수로 환경을 자동으로 채웁니다.
- 상태(활성, 비활성, 디버그)를 선택하여 어댑터 코드가 실행되는지 여부를 제어할 수 있습니다.
- 수동으로 시작된 작업으로 어댑터를 실행할 수 있게 합니다.

어댑터 링크에는 하나의 어댑터와 하나의 프로젝트에 대해서만 어댑터 링크가 정의된다는 제한사항이 있습니다. 다른 프로젝트에 동일한 어댑터를 사용하려면 어댑터의 다른 인스턴스를 작성해야 합니다.

작업을 다시 시작할 때 작업에 있는 어댑터 링크 단계의 다시 시작은 다음과 같이 작동합니다.

- 어댑터 링크의 상태가 비활성인 경우 어댑터 링크 단계가 실패하여 다시 시작 실행이 실패합니다.
- 어댑터 링크의 상태가 활성이고 링크에 실패하는 경우 어댑터 링크 단계가 실패하여 다시 시작 실행이 실패합니다.

두 경우 모두에서 빌드는 제거되지 않습니다. 그러나 빌드 상태는 실패로 설정됩니다.

**어댑터 링크 패널에 대한 정보:** 어댑터를 프로젝트에 연결하고 환경을 어댑터와 연관시키려면 어댑터 링크 패널을 사용하십시오. 패널을 보려면 프로젝트 > 어댑터 링크를 선택하십시오.



패널에는 다음과 같은 필드가 있습니다.

상태    활성, 디버그 또는 비활성 중 하나

**참고:** 이 상태는 하나의 어댑터 링크에만 적용됩니다. 링크 디버그 모드 시스템 설정을 사용하여 모든 어댑터 링크를 디버그 상태로 설정할 수 있습니다. 상태는 시스템 구성 설정보다 우선합니다.

어댑터 프로젝트에 링크하기 위한 어댑터

시작점 XML 인터페이스로의 시작점

프로젝트

어댑터를 사용할 프로젝트

환경 사용할 프로젝트의 환경

## 어댑터 태스크 개요

이 주제에서는 소스 코드 어댑터를 작성하고 어댑터 링크를 사용하여 프로젝트에 연결하며, 테스트 모드에서 어댑터 링크 프로젝트를 실행하는 데 필요한 모든 태스크를 설명합니다.

### 템플리트를 선택하여 어댑터 작성

템플리트를 선택하여 어댑터를 작성하려면 다음을 수행하십시오.

1. 프로젝트 → 어댑터를 선택하십시오.
2. 어댑터 추가를 클릭하십시오.
3. 이름에 어댑터의 고유 이름을 입력하십시오. 어댑터 이름은 전체 어댑터 세트에서 고유해야 하며 공백을 포함할 수 없습니다.
4. 유형에서 어댑터 유형을 선택하십시오.
5. 템플리트에서 템플리트를 선택하십시오. 목록은 Build Forge 제품과 함께 설치된 어댑터 템플리트를 포함합니다. ClearCase 및 ClearQuest 어댑터는 별도의 라이선스 키가 필요하지 않습니다. 기타 어댑터에는 어댑터 툴킷을 통해 별도로 라이선스가 제공됩니다.
6. 액세스에서 액세스 그룹을 선택하십시오. 어댑터를 보거나 편집할 수 있는 기능은 이 그룹 구성원으로 제한됩니다.
7. 어댑터 저장을 클릭하십시오.

### 비어 있는 환경 작성

비어 있는 환경을 작성하려면 다음을 수행하십시오.

1. 프로젝트 → 환경을 선택하십시오.
2. 환경 추가를 클릭하십시오.
3. 이름에 환경 이름을 입력하십시오. 환경의 목적을 설명하는 이름을 지정하십시오.
4. 액세스에서 액세스 그룹을 선택하십시오. 환경을 보거나 편집할 수 있는 기능은 이 그룹 구성원으로 제한됩니다.

5. 환경 저장을 클릭하십시오.


## 프로젝트에 어댑터 추가

프로젝트에 어댑터를 추가하려면 다음을 수행하십시오.

1. 프로젝트 → 어댑터 링크를 선택하십시오.
2. 어댑터 링크 추가를 클릭하십시오.
3. 어댑터에서 사용자가 작성한 어댑터(및 어댑터 템플릿)를 선택하십시오.
4. 프로젝트에서 프로젝트를 선택하십시오. 목록은 아직 어댑터에 링크되지 않은 프로젝트를 표시합니다.
5. 상태에서 활성을 선택하십시오.
6. 환경에서 어댑터 링크에 대해 작성한 빈 환경을 선택하십시오.
7. 환경 채우기에서 예를 선택하십시오. 어댑터 템플릿의 애플리케이션 환경 변수가 환경에 추가됩니다.
8. 저장을 클릭하여 어댑터를 프로젝트에 링크하십시오. 어댑터 및 프로젝트가 어댑터 링크 목록에 추가됩니다.

## 환경 변수 편집

환경 변수를 편집하려면 다음을 수행하십시오.

1. 환경을 선택하십시오.
2. 사용자가 작성한 환경의 경우, 편집 아이콘()을 클릭하십시오. 패널에는 환경에 자동으로 추가된 어댑터 환경 변수가 표시됩니다.
3. 어댑터 템플릿이 제공하는 환경 변수의 기본값을 검토하십시오.
4. 필요에 따라 소스 코드 애플리케이션의 기본값을 변경하여 어댑터 프로젝트를 실행하십시오.

## 조건 속성

condition 속성은 Perl 비교 연산자를 사용하여 일부 어댑터 특성에 적용될 조건을 허용합니다. 비교를 위해 문자열 리터럴, 숫자 또는 변수를 사용할 수 있습니다.

condition 속성 구문은 다음과 같습니다.

```
condition="true(<lvalue> <operator> <rvalue>)"
condition="false(<lvalue> <operator> <rvalue>)"
condition="hastext(variable)"
condition="isempty(variable)"
```

네 가지 유형 중 하나를 지정하십시오.

유형	설명
true	평가는 lvalue 및 rvalue가 같은 경우에 성공합니다.
false	평가는 lvalue 및 rvalue가 같지 않은 경우에 성공합니다.
hastext	평가는 값 길이가 0보다 큰 경우에 성공합니다.
isempty	평가는 값 길이가 0인 경우에 성공합니다.

lvalue와 rvalue는 문자열, 숫자 또는 문자열 또는 숫자를 포함하는 변수일 수 있습니다. 조건 연산자는 임의의 Perl 호환 가능 조건 연산자입니다. 문자열 및 숫자 조건 연산자가 있습니다. 적절한 연산자를 사용해야 하며, 그렇지 않으면 예측할 수 없는 결과를 수신합니다.

문자열 연산자	숫자 연산자
eq	==
ne	!=
gt	>
lt	<
ge	>=
le	<=

문자열과 함께 숫자 연산자를 사용하면 올바른 결과가 리턴되지 않으며, 숫자 값에 문자열 비교 연산자를 사용할 경우에도 마찬가지로 true가 유지됩니다.

예: `condition="true("PASS"=="FAIL")`

위 조건은 항상 true를 리턴하며, 이는 올바르지 않습니다.

조건의 예:

- `condition="true($BF_SERVER eq "TEST_BOX")` - 빌드 서버 변수가 TEST\_BOX를 포함하는 경우에만 항목을 실행합니다.
- `condition="false($BF_BID <=141)` - 빌드 태그가 141보다 크거나 141보다 작지 않은 경우에만 빌드를 실행합니다.

조건을 지원하는 어댑터 특성:

- adduser
- bom
- run
- setenv

조건 속성을 지원하는 특성에 대한 최신 정보는 Build Forge의 현재 설치에 대한 DTD를 재확인하십시오.

특성 지원 조건에 대해 다음과 같은 항목을 참조하십시오.

<!ATTLIST adduser condition CDATA #IMPLIED>

어댑터 DTD는 interface.dtd 파일과 같이 Windows의 경우 %BF\_HOME%\interface에 위치하고 UNIX의 경우 \$BF\_HOME/Platform/interface에 위치합니다.

## 테스트 모드에서 어댑터 실행

테스트 모드에서 어댑터를 실행하려면 다음을 수행하십시오.

1. 관리 → 시스템을 선택하십시오.
2. 시스템 구성 매개변수 목록에서 링크 디버그 모드를 선택하십시오.
3. 링크 디버그 모드에서 예를 선택하십시오.
4. 저장을 클릭하십시오.
5. 작업 → 시작을 선택하십시오.
6. 프로젝트 목록에서 프로젝트 시작 페이지에서 작성한 어댑터 링크 프로젝트를 선택하십시오.
7. 실행을 클릭하십시오.

## 작업 상태 및 로그 보기

어댑터 프로젝트의 작업 상태 및 로그 정보를 보려면 다음을 수행하십시오.

1. 작업을 여십시오.
2. 프로젝트 목록에서 어댑터 링크 프로젝트를 찾아 작업 패스/실패 상태를 보십시오.
3. 작업 로그를 보려면 다음을 수행하십시오.
  - 어댑터 프로젝트의 태그 이름을 선택하여 작업 로그 정보에 액세스하십시오.
  - 명세서(BOM)를 선택하여 BOM 보고서에 액세스하십시오.

## 핵심 어댑터 태스크

이 주제에서는 어댑터 작성 및 구성에 대한 정보를 제공합니다.

- 템플릿 선택: 어댑터 애플리케이션 템플릿을 각각 설명합니다.
- 환경 작성: 환경을 어댑터와 연관시키는 옵션을 설명합니다.
- 어댑터 작성: 어댑터를 작성하고 템플릿과 연관시키는 방법을 설명합니다.
- 프로젝트에 어댑터 추가: 프로젝트에 어댑터를 추가하는 옵션을 설명합니다.
- 어댑터 테스트: 어댑터 구성만 테스트하는 방법을 설명합니다.

또한 이 주제에서는 기타 어댑터 태스크에 대한 정보를 제공합니다.

- 어댑터 로그 레벨 설정: 어댑터에 로깅되는 정보의 양을 제어하는 방법을 설명합니다.
- 어댑터 링크 프로젝트 빠른 시작: 수동 시작에 필요한 설정을 설명합니다.

- 어댑터 템플릿 다시 설정: 템플릿 정보를 업데이트하기 위해 다시 설정해야 하는 시기를 설명합니다.

## 어댑터 환경 작성

애플리케이션 명령을 실행하려면 어댑터에 환경 변수가 필요합니다. 어댑터 템플릿에서 환경 변수는 XML 파일의 <template> 섹션에 있는 <env> 요소에 나열됩니다.

어댑터 템플릿 파일의 환경 변수를 편집하지 마십시오. Build Forge 제품에서는 환경을 사용하여 어댑터 환경 변수를 정의합니다.

기존 환경을 사용하거나 어댑터에 독점적으로 사용할 환경을 작성하십시오. 환경에 특정 사용 및 기술적 이름을 지정할 수 있으므로 환경 작성을 권장합니다. 또한 환경을 작성하면 문제점 해결이 단순화됩니다.

이 주제에서는 다음과 같이 환경을 어댑터와 연관시키는 방법을 설명합니다.

- 어댑터 링크를 통해
- 어댑터 점 명령과 연관된 단계에서

**어댑터 링크 환경 작성:** 어댑터 링크를 사용하여 프로젝트에 소스 코드 어댑터를 링크하는 경우 이 방법을 사용하십시오.

1. 프로젝트 > 환경을 선택하십시오.
2. 환경 추가를 클릭하십시오.
3. 이름에 환경 이름을 입력하십시오. 환경의 목적을 설명하는 이름을 지정하십시오.
4. 액세스에서 액세스 그룹을 선택하십시오. 환경을 보거나 편집할 수 있는 기능은 그룹 구성원으로만 제한됩니다.
5. 환경 저장을 클릭하십시오. 이 때 그룹에 환경 변수를 추가하지 마십시오. 환경 변수는 어댑터 링크를 작성할 때 어댑터 템플릿로부터 자동으로 채워집니다.

**어댑터 점 명령의 환경 작성:** 어댑터 점 명령을 사용하여 프로젝트에 어댑터를 추가하는 경우 이 방법을 사용하십시오.

이 태스크의 경우, 어댑터 환경 변수가 필요합니다.

<adaptor\_name>.xml 파일에서 외부 환경 변수는 <template>/<env> 요소에 나열됩니다.

다음 디렉토리에서 어댑터 템플릿을 찾으십시오.

<bfinstall>\interface

1. 프로젝트 > 환경을 선택하십시오.
2. 환경 추가를 클릭하십시오.
3. 이름에 환경 이름을 입력하십시오. 환경의 목적을 설명하는 이름을 지정하십시오.

4. 액세스에서 액세스 그룹을 선택하십시오. 환경을 보거나 편집할 수 있는 기능은 그룹 구성원으로만 제한됩니다.
5. 환경 저장을 클릭하십시오.
6. 환경 변수 추가를 클릭하십시오.
7. 이름에 XML <env> 요소에 표시된 대로 환경 변수 이름을 입력하십시오.
8. 값에서 XML <env> 요소의 대체 변수를 애플리케이션의 실제 값으로 변경하십시오. (올바른 값을 모르는 경우 나중에 값을 입력할 수 있습니다.)
9. 조치에서 설정을 선택하십시오.
10. 대상 프로젝트에서 정상을 선택하십시오.

### 템플릿에서 어댑터 작성

모든 어댑터는 어댑터 템플릿을 기반으로 합니다. 어댑터를 작성하기 전에 사용할 템플릿을 결정하십시오.

어댑터를 작성하려면 다음을 수행하십시오.

1. 프로젝트 > 어댑터를 선택하십시오.
2. 어댑터 추가를 클릭하십시오.
3. 이름에 어댑터의 고유 이름을 입력하십시오. 어댑터 이름은 전체 어댑터 세트에서 고유해야 하며 공백을 포함할 수 없습니다.
4. 유형에서 어댑터 유형을 선택하십시오.
5. 템플릿에서 템플릿을 선택하십시오. 목록은 Build Forge 제품과 함께 설치된 어댑터 템플릿을 포함합니다. ClearCase 및 ClearQuest 어댑터는 별도의 라이선스 키가 필요하지 않습니다. 기타 어댑터에는 어댑터 툴킷을 통해 별도로 라이선스가 제공됩니다.
6. 액세스에서 액세스 그룹을 선택하십시오. 어댑터를 보거나 편집할 수 있는 기능은 이 그룹 구성원으로 제한됩니다.
7. 어댑터 저장을 클릭하십시오.

### 프로젝트에 어댑터 추가

어댑터 코드를 실행하려면 프로젝트에 어댑터를 추가해야 합니다. 새 프로젝트를 작성하거나 기존 프로젝트에 어댑터를 추가하십시오.

이 절에서는 다음 방법을 사용하여 프로젝트에 어댑터를 추가하는 방법을 알려 줍니다.

- 어댑터 링크를 사용하여 프로젝트에 소스 코드 어댑터를 추가할 수 있습니다.
- 어댑터 점 명령을 사용하여 프로젝트에 어댑터(소스 코드 어댑터 포함)를 추가할 수 있습니다.

**어댑터 링크를 사용하여 소스 코드 어댑터 추가:** 어댑터 링크는 어댑터를 프로젝트에 연결하고 애플리케이션 환경 변수를 어댑터에 연결합니다.

이 태스크를 시작하기 전에 프로젝트 및 어댑터 환경을 작성하십시오.

이 태스크를 완료한 후에는 환경을 열고 애플리케이션 환경 변수에 실제 값을 제공하십시오(아직 제공하지 않은 경우).

1. 프로젝트 > 어댑터 링크를 선택하십시오.
2. 어댑터 링크 추가를 클릭하십시오.
3. 상태에서 상태를 선택하십시오.

상태	설명
활성	프로젝트가 실행될 때 어댑터 코드를 실행합니다.
비활성	프로젝트가 실행될 때 어댑터 코드를 건너뛵니다.
디버그	어댑터 코드만 실행합니다. 프로젝트가 실행될 때 기타 단계는 건너뛵니다.

4. 어댑터에서 어댑터 템플릿을 선택하십시오. 목록은 Build Forge 제품과 함께 설치된 어댑터 템플릿을 표시합니다.
5. 프로젝트에서 프로젝트를 선택하십시오. 목록은 아직 어댑터에 링크되지 않은 프로젝트를 표시합니다.
6. 저장을 클릭하여 어댑터를 프로젝트에 링크하십시오. 어댑터 이름이 목록에 추가됩니다.
7. 환경에서 어댑터 링크 환경을 선택하십시오.
8. 환경 채우기에서 예를 선택하십시오. 어댑터 템플릿의 애플리케이션 환경 변수가 환경에 추가됩니다.
9. 저장을 클릭하여 어댑터 링크를 저장하십시오.

**점 명령을 사용하여 어댑터 추가:** 어댑터 점 명령을 사용하여 프로젝트에 어댑터를 추가할 수 있습니다. 점 명령은 단계가 실행될 때 `<adaptor_name>.xml` 파일을 호출합니다.

이 태스크를 시작하기 전에 프로젝트 및 어댑터 환경을 작성하십시오.

이 태스크를 완료한 후에는 환경을 열고 애플리케이션 환경 변수에 실제 값을 제공하십시오(아직 제공하지 않은 경우).

프로젝트에 어댑터 점 명령을 단계로 추가하려면 다음을 수행하십시오.

1. 라이브러리를 선택하십시오.
2. 목록에서 프로젝트를 선택하십시오.
3. 단계 추가를 클릭하십시오.
4. 이름에 단계 이름을 입력하십시오.
5. 명령에 애플리케이션 유형의 어댑터 점 명령(.source, .defect, .test, .pack)을 입력하십시오.



6. 환경에서 어댑터에 대해 작성된 환경을 선택하십시오.
7. 단계 저장을 클릭하십시오.

## 어댑터 테스트

어댑터 프로젝트를 실행하여 어댑터 구성을 테스트하십시오. 어댑터가 외부 애플리케이션과 상호작용하고 예상하는 결과를 리턴하며, 기타 프로젝트 단계와 분리되어 어댑터 코드를 실행할 수 있는지 확인하십시오.

이 주제에서는 다음 어댑터를 테스트하는 방법을 설명합니다.

- 어댑터 링크를 통해 프로젝트에 추가된 어댑터
- 어댑터 점(DOT) 명령을 사용하여 프로젝트에 추가된 어댑터

**링크된 어댑터 테스트:** 어댑터 링크를 통해 프로젝트에 링크된 소스 코드 어댑터의 경우, 이 프로시저를 사용하여 어댑터 구성을 테스트하십시오.

일반적인 프로시저는 다음과 같습니다.

1. 소스 파일 변경
2. 링크된 어댑터를 사용하여 Build Forge 프로젝트 실행
3. BOM 보고서에서 변경된 소스 파일에 대한 정보 확인
4. 패스 또는 실패 이메일 알림 확인

링크된 어댑터를 테스트하려면 다음을 수행하십시오.

1. 프로젝트 > 어댑터 링크를 선택하십시오.
2. 목록에서 링크된 어댑터 및 프로젝트를 선택하십시오.
3. 상태에서 디버그를 선택하십시오.
4. 저장을 클릭하십시오.
5. 소스 코드 애플리케이션에서 하나 이상의 소스 파일을 변경하십시오. 변경사항을 제출하여 소스 코드 저장소를 업데이트하십시오.
6. 다음과 같이 어댑터 링크 프로젝트를 실행하십시오.
  - a. 관리 > 시스템을 선택하십시오.
  - b. 목록에서 링크 디버그 모드를 선택하십시오.
  - c. 링크 디버그 모드에서 예를 선택하십시오.
  - d. 저장을 클릭하십시오.
  - e. 작업 > 시작을 선택하십시오.
  - f. 프로젝트 목록에서 프로젝트 시작 페이지에서 작성한 어댑터 링크 프로젝트를 선택하십시오.
  - g. 실행을 클릭하십시오.
7. 작업에 대한 BOM 보고서를 검토하십시오.

- a. 작업을 여십시오.
- b. 완료 탭을 선택한 후 **BOM** 탭을 선택하십시오.

**어댑터 점 명령 테스트:** 점 명령을 사용하여 프로젝트에 추가된 어댑터의 경우, 이 선택적 프로시저를 사용하여 어댑터 구성을 테스트하십시오.

일반적인 프로시저는 다음과 같습니다.

1. 소스 파일 변경
2. 어댑터 점(dot) 명령을 사용하여 Build Forge 프로젝트 실행
3. BOM 보고서에서 변경된 소스 파일에 대한 정보 확인
4. 패스 또는 실패 이메일 알림 확인

어댑터 점 명령을 테스트하려면 다음을 수행하십시오.

1. 작업 > 시작을 선택하십시오.
2. 목록에서 프로젝트를 선택하십시오.
3. 작업 단계 탭을 여십시오.
4. 단계 이름 선택란을 사용하여 어댑터 점 명령을 제외한 모든 것에 대한 선택 표시를 지우십시오.
5. 실행을 클릭하여 프로젝트를 실행하십시오.
6. 작업에 대한 BOM 보고서를 검토하십시오.
  - a. 작업을 선택하십시오.
  - b. 완료 탭을 선택한 후 **BOM** 탭을 선택하십시오.

## 어댑터 로그 레벨 설정

어댑터의 단계 로그에 기록되는 정보의 양을 제어하려면 `_InterfaceLoggingLevel` 환경 변수를 사용하십시오.

1. 어댑터의 환경에 `_InterfaceLoggingLevel`을 추가하십시오.
    - 레벨 8은 대부분의 정보를 로깅하고 레벨 0은 최소한의 정보만 로깅합니다.
    - 로깅 레벨은 포괄적입니다. 예를 들어, 레벨 2는 레벨 1과 레벨 0의 정보를 포함합니다.
    - 레벨 7은 기본 로깅 레벨입니다.
  2. `_InterfaceLoggingLevel` 변수 값으로 로그 레벨을 지정하십시오.
- 0:** Exec 행 + 서버 연결 오류 또는 취소 알림, 그 밖엔 없음
- 1:** 구문 분석된 명령(서버로 보내질 명령)
- 2:** 구문 분석되지 않은 명령(로컬 변수가 설정되기 전의 명령)
- 3:** 빌드 및 환경 변수 SET 행

- 4: 임시 및 내부 변수 SET 행
- 5: 환경 평가, 이메일 그룹 추가, BOM 텍스트 로깅 행
- 6: 블록 및 하위 블록 시작/끝 행
- 7: (기본 로깅 레벨) 일치 패턴과 비교하여 확인될 에이전트 출력 + 패턴과 일치한 행
- 8: 모든 에이전트 출력

### 어댑터 링크 프로젝트 수동 시작

어댑터 링크 프로젝트를 스케줄대로 실행하거나 빠른 시작 옵션을 사용하여 실행하거나 링크 실행 선택란을 체크하는 경우 수동으로 시작할 수 있습니다. 링크 실행을 체크하지 않는 경우 프로젝트가 어댑터 단계 없이 실행됩니다.

1. 작업 > 시작을 선택하십시오.
2. 프로젝트 목록에서 어댑터 링크 프로젝트의 이름을 선택하십시오.

프로젝트 시작 페이지가 열립니다.

3. 링크 실행 선택란을 체크하십시오.
4. 프로젝트 시작 페이지에서 실행을 클릭하십시오.

### 어댑터 링크 프로젝트 빠른 시작

어댑터 링크 프로젝트를 스케줄 대로 실행하거나, 프로젝트 시작 페이지(작업 → 시작)에서 프로젝트 이름을 선택하여 수동으로 시작하거나, 빠른 시작 사용 시스템 구성 설정값을 설정한 경우 빠른 시작 아이콘을 사용하여 실행할 수 있습니다. 다음 단계에서는 빠른 시작 사용을 설정한 것으로 가정합니다.

1. 관리 > 시스템을 선택하십시오.
2. 목록에서 수동 작업 링크를 선택하십시오.
3. 수동 작업 링크에서 예를 선택하십시오.
4. 저장을 클릭하십시오.
5. 작업 > 시작을 선택하십시오.
6. 프로젝트 목록에서 어댑터 링크 프로젝트가 즉시 실행되도록 빠른 시작 아이콘(



)을 선택하십시오. 작업이 실행 중인지 확인하려면 작업 > 실행 탭을 선택하십시오.

### 어댑터 템플릿 다시 설정

#### 이 태스크 정보

어댑터 템플릿을 다시 설정하면 <bfinstall>\interface 디렉토리에서 Build Forge 데이터베이스로 최신 어댑터 템플릿이 복사됩니다.

다음은 수행할 때마다 어댑터 템플릿을 다시 설정하십시오.

- 유지보수 릴리스 또는 새 제품 버전 설치
- (관리 콘솔이 아닌) interface 디렉토리에서 버전을 편집하여 템플릿 수정
- 새 어댑터 템플릿 작성

### 프로시저

1. 관리 > 시스템을 선택하십시오.
2. 목록에서 어댑터 템플릿 다시 설정을 선택하십시오.
3. 어댑터 템플릿 다시 설정에서 예를 선택하십시오.
4. 저장을 클릭하십시오.

## ClearQuest 빌드 레코드 업데이트

시스템을 Rational ClearQuest와 통합하여 ClearQuest 데이터베이스에서 빌드 레코드를 자동으로 작성하고 업데이트할 수 있습니다.

시스템은 로그 데이터로의 링크를 사용하여 IBM Rational® ClearQuest® 데이터베이스에 빌드 레코드를 자동으로 작성할 수 있습니다. 뿐만 아니라, 작업 패스 시에 시스템은 ClearQuest 데이터베이스를 업데이트할 수 있으며 작업이 완료되면 종료 시간과 달성된 단계에 대한 요약を 기록합니다. 이 기능에는 Rational ClearQuest 버전 7.0 이상이 필요합니다.

**참고:** 빌드 레코드를 작성하고 해당 레코드를 빌드에 대한 정보로 채우려면 먼저 ClearQuest 유지보수 도구를 열고 데이터베이스에 대한 연결을 설정해야 합니다.

사용자가 ClearQuest 데이터베이스를 업데이트하도록 프로젝트를 구성할 때 시스템은 다음을 수행합니다.

### 작업 시작

시스템은 작업을 실행할 때 ClearQuest 빌드 레코드를 작성합니다. 빌드 레코드는 제출 상태에 있으며 작업 로그 URL, 시작 시간, 릴리스 이름 및 ID와, "빌드 XYZ가 시작되었음"을 표시하는 로그 항목을 포함합니다. 소스 제어 어댑터가 작업을 취소하는 경우(예를 들어, 소스 변경사항이 없어서) ClearQuest 빌드 레코드가 작성되지 않습니다.

**참고:** 프로젝트에 다른 프로젝트가 체인된 경우, 새 프로젝트는 자체의 고유 ClearQuest 빌드 ID를 가져옵니다.

### 작업 패스/실패

작업이 패스 또는 실패할 때 시스템은 ClearQuest에서 빌드 상태를 완료 또는 실패로 변경하고 빌드 종료 시간을 설정하며, ClearQuest 빌드 로그에 작업 단계 요약을 저장합니다. 요약에는 각 단계의 이름, 결과 상태 및 서버가 포함됩니다.

## 작업 다시 시작

작업이 다시 시작될 때 시스템은 ClearQuest에서 빌드 상태를 제출로 변경하고 "빌드 XYZ가 다시 시작되었음"을 표시하는 ClearQuest 빌드 로그 항목을 작성합니다.

특수 환경 변수를 통해 ClearQuest 통합을 구성합니다. ClearQuest 데이터베이스에 프로젝트를 링크하려면 다음 표의 변수가 프로젝트 환경에 포함되어 있는지 확인하십시오.

**참고:** 이러한 변수가 프로젝트 환경에 있어야 하며, 단계에 변수를 추가하는 것으로는 부족합니다. 그러나 다른 환경을 통해 이러한 변수를 포함하는 포함 유형으로 설정된 변수를 사용할 수 있습니다. 또한 CQ\_RELEASE\_NAME 값은 유일하게 프로젝트마다 다를 수 있으므로 다른 변수를 포함하는 환경을 작성하고 포함 유형의 변수를 사용하여 프로젝트 환경에 해당 환경을 포함시킬 수 있습니다(이 경우, CQ\_RELEASE\_NAME을 프로젝트 특정 환경 변수로 지정할 수도 있음).

표 17. Rational ClearQuest 통합에 필요한 환경 변수

변수	설명
CQ_DBNAME	업데이트할 ClearQuest 데이터베이스의 이름
CQ_INTERACTION	ClearQuest 빌드 레코드 작성을 사용하도록 프로젝트 환경에 올바른 환경 변수가 정의되어 있지만 빌드 레코드를 작성하지 않으려면 이 변수를 OFF로 설정하여 빌드 레코드 작성을 사용하지 않도록 하십시오.  빌드 레코드 작성을 사용하려면 이 변수를 ON으로 설정하십시오.  참고: ClearQuest 어댑터 중 하나를 사용 중인 경우, 이 환경 변수를 OFF로 설정하십시오.
CQ_USER	ClearQuest 데이터베이스에 로그인할 때 사용할 사용자 이름
CQ_RELEASE_NAME	ClearQuest 데이터베이스에서 업데이트할 릴리스의 이름
CQ_PASSWORD	ClearQuest 데이터베이스에 로그인할 때 사용할 비밀번호. 필수는 아니며, 기본값은 공백으로 설정됩니다.
CQ_DBSET	ClearQuest 데이터베이스 설정 값. 필수는 아니며, 기본값은 공백으로 설정됩니다.

## 고급 어댑터 태스크

Build Forge 어댑터 템플릿을 수정하거나 Build Forge 프로젝트와 인터페이스하게 할 외부 애플리케이션용 새 어댑터 템플릿을 작성할 수 있습니다.

템플릿 수정 및 작성은 다음에 대한 작업 지식을 필요로 하는 고급 태스크입니다.

- XML 언어
- 외부 애플리케이션에 대한 명령 언어
- 정규식

고급 태스크에 대한 작업을 하기 전에 다음 절의 정보를 읽으십시오.

- 541 페이지의 『핵심 어댑터 태스크』
- 558 페이지의 『어댑터 템플리트 구조』
- 556 페이지의 『어댑터 참조』

이 절은 다음 태스크에 대한 정보를 제공합니다.

- 어댑터 템플리트 수정: 모든 어댑터에 대해 템플리트를 수정하는 방법을 설명합니다.
- 단일 어댑터에 대한 템플리트 수정: 하나의 어댑터에 대해 템플리트를 수정하는 방법을 설명합니다.
- 새 어댑터 템플리트 작성: 새 템플리트를 작성하기 위한 일반 프로시저를 설명합니다.
- 예제: 어댑터 알림에 사용자 추가: 이메일 알림에 액세스 그룹을 추가하는 방법을 설명합니다.
- 예제: BOM 보고서에서 변경 세부사항 제거: BOM 보고서에서 변경 세부사항을 제거하는 방법을 설명합니다.

## 어댑터 템플리트 수정

어댑터 템플리트에서 작성된 모든 향후 어댑터가 템플리트 수정사항을 선택하도록 하려면 이 프로시저를 사용하십시오.

시작하기 전에 수정하려는 내용을 알아야 합니다. 예를 들어, 알림을 변경하거나 BOM 보고서 형식을 수정하려고 합니다.

1. XML 편집기를 사용하여 수정할 어댑터 템플리트를 여십시오. 어댑터 템플리트는 다음 디렉토리에 있습니다.

Microsoft Windows 운영 체제를 실행 중인 컴퓨터의 경우 *bfinstall\interface*

UNIX 또는 Linux 운영 체제를 실행 중인 컴퓨터의 경우 *bfinstall/Platform/interface*

2. 템플리트 수정사항을 입력하십시오.
3. 어댑터 템플리트를 저장하십시오. 템플리트 이름을 변경하지 마십시오. 어댑터와 어댑터 템플리트는 이름이 고유해야 합니다.

수정된 어댑터 템플리트를 사용하여 다음 번에 어댑터를 작성할 때 새 어댑터에 템플리트 수정사항이 포함됩니다.

## 단일 어댑터용 템플리트 수정

단일 어댑터 인스턴스용 템플리트만 수정하려면 이 프로시저를 사용하십시오.

시작하기 전에 수정하려는 내용을 알아야 합니다. 예를 들어, 알람을 변경하거나 BOM 보고서 형식을 수정하려고 합니다.

수정할 어댑터가 작성된 경우, 다음과 같이 관리 콘솔을 통해 템플리트를 수정하십시오.

1. 프로젝트 > 어댑터를 선택하십시오.
2. 텍스트 상자에 템플리트 수정사항을 입력하십시오.
3. 어댑터 저장을 클릭하십시오.

참고: 변경사항은 Build Forge 데이터베이스의 어댑터와 연관된 어댑터 템플리트 인스턴스에만 저장됩니다. interface 디렉토리의 어댑터 템플리트 파일에는 변경사항이 저장되지 않습니다.

수정할 어댑터가 작성되지 않은 경우, 다음과 같이 어댑터를 작성하기 전에 템플리트를 수정하십시오.

1. XML 편집기를 사용하여 어댑터 템플리트를 여십시오. 어댑터 템플리트는 다음 디렉토리에 있습니다.

Microsoft Windows 운영 체제를 실행 중인 컴퓨터의 경우 `bfinstall\interface`

UNIX 또는 Linux 운영 체제를 실행 중인 컴퓨터의 경우 `bfinstall/Platform/interface`

2. 템플리트 수정사항을 입력하십시오.
3. 어댑터 이름을 변경한 후 어댑터 템플리트를 저장하십시오.

참고: 어댑터와 어댑터 템플리트는 이름이 고유해야 합니다.

4. 어댑터 템플리트를 다시 설정하여 수정된 템플리트를 선택하고 관리 콘솔에서 사용 가능한 템플리트 목록에 이를 추가하십시오. 547 페이지의 『어댑터 템플리트 다시 설정』의 내용을 참조하십시오.

## 새 어댑터 템플리트 작성

Build Forge 프로젝트와 상호작용하게 할 외부 애플리케이션용 새 어댑터 인터페이스를 작성하려면 이 프로시저를 사용하십시오.

1. Build Forge 제품이 제공하는 어댑터 템플리트에서 XML 구조 및 요소를 검토하십시오. 어댑터 템플리트는 다음 디렉토리에 있습니다.

Microsoft Windows 운영 체제를 실행 중인 컴퓨터의 경우 `bfinstall\interface`

UNIX 또는 Linux 운영 체제를 실행 중인 컴퓨터의 경우 `bfinstall/Platform/interface`

2. 새 어댑터를 사용하여 수행할 내용을 계획하십시오.



- 새 어댑터가 실행할 명령 파악
  - 명령 구문 분석 방법 판별
  - 구문 분석 결과를 통해 수집된 데이터를 사용하여 수행할 작업 판별
  - 필요한 외부 환경 변수 파악
3. 모델로 사용할 어댑터 템플리트 중 하나를 인터페이스 디렉토리에서 선택하십시오. 가능하면 동일한 외부 애플리케이션을 기반으로 어댑터를 선택하십시오. 또는 작성 중인 어댑터와 유사한 기능을 가진 어댑터를 선택하십시오. 모델의 XML 계층 구조, 요소 및 요소 속성을 지침으로 사용하여 새 템플리트를 작성하십시오.
  4. XML 편집기를 사용하여 모델 어댑터 템플리트를 여십시오. 템플리트를 인터페이스 디렉토리에 새 이름으로 저장하십시오.
  5. 계획대로 새 어댑터의 XML 코드를 작성하십시오.
  6. 새 어댑터 템플리트를 저장하십시오.
  7. 인터페이스 디렉토리의 interface.dtd 파일을 사용하여 어댑터 템플리트의 유효성을 검증하십시오.
  8. 어댑터 템플리트를 다시 설정하여 새 템플리트를 선택하고 관리 콘솔에서 사용 가능한 템플리트 목록에 이를 추가하십시오. 547 페이지의 『어댑터 템플리트 다시 설정』의 내용을 참조하십시오.
  9. 어댑터 프로젝트를 작성하십시오.
  10. 어댑터 환경을 작성하십시오. 542 페이지의 『어댑터 환경 작성』을 참조하십시오.
  11. 새 어댑터 템플리트를 사용하여 어댑터를 작성하십시오. 543 페이지의 『템플리트에서 어댑터 작성』의 내용을 참조하십시오.
  12. 프로젝트에 어댑터를 추가하십시오. 543 페이지의 『프로젝트에 어댑터 추가』를 참조하십시오.
  13. 어댑터 프로젝트를 실행하여 어댑터를 테스트하십시오. 545 페이지의 『어댑터 테스트』를 참조하십시오.

## 여러 시작점 어댑터 작성

Build Forge가 제공하는 어댑터 템플리트는 단일 시작점 어댑터입니다.

단일 시작점 어댑터의 경우, 관리 콘솔(프로젝트 > 어댑터)에서 실행하려는 애플리케이션 및 함수의 템플리트 이름을 선택합니다. 예: ClearCaseBaseline 또는 ClearCaseByDate.

필요한 경우 ClearCase에 대한 다중 인터페이스 또는 함수를 포함하는 하나의 ClearCase 어댑터 템플리트를 작성할 수 있습니다. 다중 시작점 어댑터의 경우 시작점을 호출하는 이름으로 각 인터페이스를 식별합니다.

다중 시작점 어댑터를 작성하려면 다음을 수행하십시오.



1. 어댑터 템플리트를 작성하십시오. 템플리트를 작성하기 위해 제공된 템플리트 중 하나를 모델로 사용하고 필요한 경우 XML을 수정하는 옵션이 있습니다. 템플리트에 추가하는 각 인터페이스의 시작점을 식별하기 위해서는 템플리트에서 name 속성을 <interface> 요소에 추가해야 합니다. <interface> 요소 관련 구문이 다음 예제에 표시되어 있습니다.

```
<interface name="By Date" default="true">
</interface>
```

2. 공백이 없는 고유 이름을 사용하여 어댑터를 작성하고 이를 어댑터 템플리트와 연관시키십시오. 543 페이지의 『템플리트에서 어댑터 작성』의 내용을 참조하십시오.
3. 어댑터 링크 또는 어댑터 점(dot) 명령을 사용하여 프로젝트에 어댑터를 추가하십시오.

- 다음 예제는 .source 어댑터 명령을 사용하여, ClearCase 어댑터에서 By Date 인터페이스 함수를 호출하는 프로젝트 단계에 어댑터를 추가합니다.

```
.source ClearCase "By Date"
```

- 어댑터 링크를 사용하여 다중 시작점 어댑터를 호출하려면 어댑터 템플리트에서 다음 조치 중 하나를 수행하여 프로젝트가 실행될 때 실행할 인터페이스 함수를 지정하십시오.
  - 템플리트 파일에 첫 번째 <interface> 요소로 실행할 함수의 <interface> 요소 정의 놓기
  - 실행할 함수에 대해 <interface> 요소의 기본 속성을 참으로 설정 (default="true")

### 예제: 이메일 알림 사용

외부 애플리케이션의 변경을 초대하는 사용자에게 이메일 알림을 보내도록 어댑터 템플리트를 구성할 수 있습니다. 다음 예제는 두 가지 유형의 알림 설정 방법을 보여줍니다.

- 현재 빌드에 대해 파일을 체크인한 모든 사용자에게 알림
- Build Forge 액세스 그룹의 모든 구성원에게 알림

다음 프로시저는 ClearCaseByDate 템플리트의 요소를 참조합니다. 어댑터 템플리트는 이 요소를 사용하여 알림을 사용할 수 있습니다.

#### 현재 빌드에 대해 파일을 체크인한 모든 사용자에게 알림

<adduser> 명령을 사용하여 빌드에 대해 코드를 체크인한 사용자 그룹을 동적으로 빌드한 후 <notify> 명령을 사용하여 해당 그룹에 알림을 보낼 수 있습니다.

ClearCaseByDate 템플리트는 두 시간소인 간의 모든 변경사항을 보기 위해 ClearCase 를 조회합니다. 기본 시간소인은 현재 어댑터 실행 및 마지막 어댑터 실행에 대한 것입니다. 실제로는 마지막 빌드 이후로 현재 빌드에 대해 체크인된 모든 변경사항 목록으로 변환됩니다.

가정: 보기의 모든 사용자 이름이 알림에 해당 이름을 사용하는 SMTP 서버에 공개됩니다. 이는 ClearCase 사용자 이름이 이메일 사용자 이름과 일치해야 함을 의미합니다.

이 알림을 사용하려면 다음을 수행하십시오.

1. XML 편집기에서 ClearCaseByDate 어댑터 템플리트를 여십시오.
2. <adduser>를 찾아서 편집하여 다음과 같이 사용자 그룹을 작성하십시오.

```
<adduser group="MyChangers" user="$4">
```

위치 매개변수 \$4는 ClearCaseByDate 템플리트가 생성한 ClearCase 보기에 표시되는 사용자 이름 필드를 나타냅니다.

3. 이 그룹에 이메일을 보내도록 알림을 설정하십시오. 다음 설정은 프로젝트가 실패할 경우와 프로젝트가 성공할 경우 모두 이메일을 보냅니다. 일부 환경에서는 빌드가 실패하는 경우에만 알리도록 할 수 있습니다.

```
<!-- Set some notifications for when the build completes -->
<onproject result="fail">
  <notify group="MyChangers" subject="Build $BF_TAG ($CurDate) Failed."

  message="${Changing}${Changes}"/>
</onproject>
<onproject result="pass">
  <notify group="MyChangers" subject="Build $BF_TAG ($CurDate) Passed."

  message="${Changing}${Changes}"/>
</onproject>
```

4. 어댑터 템플리트를 저장하십시오.

어댑터가 실행될 때 MyChangers 그룹이 보기의 사용자 이름으로부터 빌드됩니다. 빌드 프로젝트가 완료되면 해당 그룹에 이메일 알림이 전송됩니다.

### Build Forge 액세스 그룹에 속한 모든 사용자에게 알림

이 예제에서는 Build Forge 액세스 그룹의 모든 구성원에게 알립니다. ClearCaseByDate 어댑터 템플리트가 예제에 사용됩니다. 가정: Build Forge의 모든 사용자 이름이 SMTP 서버의 이메일 사용자 이름에 해당합니다.

1. XML 편집기에서 ClearCaseByDate 어댑터 템플리트를 여십시오.
2. <adduser>를 찾아서 편집하여 다음과 같이 Build Forge 액세스 그룹에서 사용자 그룹을 작성하십시오.

```
<adduser group="Developer_Access_Group" user="Developer">
```

- 이 그룹에 이메일을 보내도록 알림을 설정하십시오. 다음 설정은 프로젝트가 실패할 경우와 프로젝트가 성공할 경우 모두 이메일을 보냅니다. 일부 환경에서는 빌드가 실패하는 경우에만 알리도록 할 수 있습니다.

```
<!-- Set some notifications for when the build completes -->
<onproject result="fail">
  <notify group="Developer_Access_Group" subject="Build $BF_TAG ($CurDate) Failed."

  message="${Changing}${Changes}"/>
</onproject>
<onproject result="pass">
  <notify group="Developer_Access_Group" subject="Build $BF_TAG ($CurDate) Passed."

  message="${Changing}${Changes}"/>
</onproject>
```

- 어댑터 템플리트를 저장하십시오.

어댑터가 실행될 때 Developer\_Access\_Group 그룹이 개발자 액세스 그룹에 속한 사용자 이름으로부터 빌드됩니다. 빌드 프로젝트가 완료되면 해당 그룹에 이메일 알림이 전송됩니다.

**중요사항:** <notify> 지시문이 실패하면(예를 들어 이메일 주소로 지정된 사용자가 존재하지 않는 경우), XML이 실패하고 모든 후속 알림이 실패합니다.

### 예제: BOM 보고서에서 변경 세부사항 제거

대부분의 어댑터 템플리트는 BOM 보고서에 변경 세부사항을 로깅합니다(변경 세부사항을 로깅하는 데 diff 명령이 사용됨).

다음 단계는 ClearCaseByDate 템플리트의 요소를 참조하지만 어댑터 템플리트 변경 세부사항을 제거하는 데 사용할 수 있습니다.

BOM 보고서에서 변경 세부사항을 제거하려면 다음을 수행하십시오.

- XML 편집기에서 어댑터 템플리트를 여십시오.
- diff 명령을 호출하는 <run> 요소를 찾으십시오. 다음 행을 제거하십시오.

```
<run command="cc_diff" params="$VIEW $1 $2" server="$CCSERVER" dir="/" timeout="360"/>
```

- diff 명령을 호출하는 <command> 요소를 찾으십시오. 다음 행을 제거하십시오.

```
<!-- The cc_diff command does a generic clearcase diff,
logging the full output
of the diff in the BuildForge BOM -->
<command name="cc_diff">
<execute>
pushd \\view\$1 && cleartool diff -pred -diff_format "$2$3"
</execute>
<resultsblock>
<match pattern=".+ ">
<bom category="Source" section="diff">
<field name="diff" text="$_" />
```

```

</bom>
</match>
<resultsblock/>
</command>

```

4. <bomformat> 섹션을 찾은 후 diff 명령 출력의 <section> 요소를 찾으십시오. 다음 행을 제거하십시오.

```

<section name="diff">
<field name="diff" title="Change Details"/>
</section>

```

5. 어댑터 템플리트를 저장하십시오.

## 어댑터 참조

어댑터가 Build Forge 프로젝트에 추가되고 수정없이 실행되도록 설계되었습니다. 어댑터를 수정하거나 새 어댑터를 작성하려면 Build Forge 어댑터 템플릿에 사용되는 XML 템플릿 구조와 요소를 이해해야 합니다.

**참고:** 이 절에서는 Build Forge 어댑터 템플릿에서 사용되는 외부 애플리케이션 명령을 설명하지 않습니다. 이러한 명령에 대한 정보는 외부 애플리케이션 문서를 참조하십시오.

Build Forge 제품과 함께 설치된 어댑터 템플릿은 다음 디렉토리에 있습니다.

Microsoft Windows 운영 체제를 실행 중인 컴퓨터의 경우 *bfinstall\interface*

UNIX 또는 Linux 운영 체제를 실행 중인 컴퓨터의 경우 *bfinstall/Platform/interface*

이 절에서는 다음 참조 정보를 제공합니다.

- **어댑터 요구사항:** 어댑터 사용에 대한 일반 요구사항 및 ClearQuest 어댑터 템플릿에 대한 특정 요구사항을 설명합니다.
- **어댑터 점(dot) 명령:** 어댑터 점 명령 구문을 설명합니다.
- **ClearCase 및 ClearQuest 환경 변수:** ClearCase 및 ClearQuest 어댑터에서 사용되는 환경 변수를 설명합니다.
- **Perforce 환경 변수:** Perforce에 필요한 몇 가지 추가 환경 변수를 설명합니다.
- **어댑터 템플릿 구조:** Build Forge 어댑터 템플릿의 일반적인 구조를 설명합니다.
- **어댑터 XML 참조:** Build Forge 어댑터 템플릿에서 사용되는 XML 요소를 설명합니다.

## 어댑터용 점 명령

일부 점 명령을 사용하여 외부 애플리케이션용 어댑터를 Build Forge 프로젝트에 프로젝트 단계로 추가할 수 있습니다.

- **.source:** 소스 코드 애플리케이션용 어댑터를 프로젝트 단계에 추가합니다.

- **.defect:** 결함 추적 애플리케이션용 어댑터를 프로젝트 단계에 추가합니다.
- **.test:** 테스트 애플리케이션용 어댑터를 프로젝트 단계에 추가합니다.
- **.pack:** 패키징 애플리케이션용 어댑터를 프로젝트 단계에 추가합니다.

420 페이지의 『점 명령 참조』의 내용을 참조하십시오.

## Rational ClearCase 및 Rational ClearQuest 환경 변수

Rational ClearCase 및 Rational ClearQuest 어댑터 템플리트는 다음 표에 있는 환경 변수를 사용하여 cleartool 명령을 실행합니다.

각 ClearCase 또는 ClearQuest 어댑터 템플리트에 다음 표에 있는 모든 환경 변수가 필요한 것은 아닙니다.

각 어댑터 템플리트에서 필수 환경 변수는 <template> 섹션의 <env> 요소에 나열됩니다.

ClearCase 또는 ClearQuest 어댑터 프로젝트를 실행하기 전에 필수 변수에 실제 값을 제공하거나 기본값을 승인하십시오. 어댑터에 지정된 환경에서 변수 값을 편집하십시오.

프로젝트 환경에 변수가 나타나야 합니다. 이 환경 변수는 단계 레벨에서 작동하지 않습니다.

어댑터 템플리트는 다음 디렉토리에 있습니다.

Microsoft Windows 운영 체제를 실행 중인 컴퓨터의 경우 *bfinstall\interface*

UNIX 또는 Linux 운영 체제를 실행 중인 컴퓨터의 경우 *bfinstall/Platform/interface*

표 18. ClearCase 및 ClearQuest의 환경 변수

환경 변수 이름	대체 변수	설명
VIEW	value=my_adaptor_view	이 변수를 어댑터와 함께 사용할 ClearCase 보기의 이름으로 설정하십시오.
VOB_PATH	value=c_vob	이 값을 컴포넌트 VOB의 이름으로 설정하고 선택적으로 서브디렉토리를 설정하십시오.
PROJECT_VOB	value=\ProjectVob	ByBaseline 어댑터를 사용할 때 이 변수를 프로젝트 VOB의 이름으로 설정하십시오(UCM ClearCase에서만 사용됨).
CCSERVER	value=BFServerName	이 변수를 ClearCase 클라이언트가 설치되어 실행 중인 Build Forge 서버의 이름으로 설정하십시오.
CurDate	value=.date %d-%b-%y.%H:%M:%S	어댑터에 현재 날짜를 제공합니다. .date 명령을 사용하여 ClearCase가 예상하는 형식으로 날짜를 생성합니다. 이 값을 변경하지 마십시오.

표 18. ClearCase 및 ClearQuest의 환경 변수 (계속)

환경 변수 이름	대체 변수	설명
LAST_RUN	value=1-Jan-05.00:00:00	ByDate 어댑터의 경우, 시스템은 이 값을 사용하여 변경사항이 발생했는지 여부를 판별합니다. 값은 마지막 성공적 실행 날짜입니다. 어댑터를 강제 실행하기 위해 어댑터를 테스트할 때 일부 변경사항에 선행하는 날짜를 선택하여 이 값을 조작할 수 있습니다. 실행을 계속하도록 허용하는 경우, 어댑터는 이 값을 현재 날짜로 자동으로 업데이트합니다. 기본 값은 1-Jan-05.00:00:00입니다.
LABEL	value=BUILD_1	ByLabel 어댑터의 경우, 사용자가 사용자의 어댑터를 사용하여 레이블 별로 ByLabel 어댑터와의 차이점을 생성할 때 시스템은 이 값을 레이블로 사용합니다.
BASELINE	value=BUILD_1	ByBaseline 어댑터의 경우, 사용자가 사용자의 어댑터를 사용하여 기준 선별로 차이점을 생성할 때 시스템은 이 값을 기준으로 사용합니다.
ACTIVITIES	value=SAMPL0001@\ProjectVob	ClearQuestClearCaseByActivity 어댑터의 경우, 공백으로 구분된 활동 ID 세트입니다.

## Perforce 환경 변수

다음 Perforce 환경 변수를 Perforce 어댑터에 지정된 Build Forge 환경 그룹에 추가하십시오.

- P4USER
- P4PASSWD

Perforce 서버에 액세스하려면 Build Forge에 올바른 사용자 이름과 비밀번호가 필요합니다. 단계 로그에서 Perforce 사용자 이름과 비밀번호는 일반 텍스트로 기록됩니다.

환경 변수의 지정 숨김 특성을 사용하여 Perforce 인증 정보를 암호화할 수는 없습니다.

## 어댑터 템플릿 구조

이 주제에서는 Build Forge 어댑터 템플릿의 일반 XML 구조 또는 요소 계층 구조를 설명합니다.

어댑터 템플릿은 <template>, <interface>, <command> 및 <bomformat> 섹션 요소로 구성됩니다. 이 절은 각각 하위 요소를 포함합니다.

요소 설명은 559 페이지의 『어댑터 XML 참조』의 내용을 참조하십시오.

```
<PROJECT_INTERFACE>
<template>
<env/>
</template>

<interface>
<setenv/>
<run/>
<ontempenv>
<step/>
```

```

</ontempenv>
<onproject>
<notify/>
</onproject>
</interface>

<command>
<execute> or <command>
command line
</execute> or </command>
<resultsblock>
<match>
<bom>
<field/>
</bom>
<adduser/>
<setenv/>
<run/>
</match>
</resultsblock>
</command>

<bomformat>
<section>
<field/>
</section>
</bomformat>
</PROJECT_INTERFACE>

```

## 어댑터 XML 참조

이 절에는 어댑터 XML 요소가 알파벳순으로 나열되며 어댑터 XML 언어에서 사용하는 요소에 대한 참조입니다. 몇 가지 예제와 의사 코드가 설명에 포함되어 있습니다.

### adduser

<interface> 요소 또는 <match> 요소 내에 <adduser> 요소를 사용하여 변경을 일으킨 사용자에게 어댑터가 알림을 보낼 수 있도록 변경 명령의 출력을 기반으로 임시 그룹에 사용자를 추가하십시오. 사용자가 이미 그룹의 구성원인 경우 여러 개의 알림을 방지하기 위해 시스템은 그룹에 사용자를 추가하지 않습니다. <adduser> 요소는 빈 요소입니다. 그룹 속성은 어댑터 논리 실행 중 작성된 임시 그룹을 지정합니다. 실제 알림이 송신되도록 <notify> 요소에서 같은 그룹을 참조해야 합니다.

```
<adduser group="MyChangers" user="$4"/>
```

<adduser> 요소가 임시 액세스 그룹에 변경을 일으킨 사용자를 추가하는지 여부를 제어하려면 condition="함수" 속성을 사용하십시오. 함수의 값은 참 또는 거짓으로 평가되는 표현식입니다. 표현식이 참으로 평가되면 사용자가 임시 그룹에 추가되고 그렇지 않으면 사용자가 추가되지 않습니다. 407 페이지의 『조건 함수』의 내용을 참조하십시오.



다음 이메일 주소 형식이 지원됩니다. *Username*은 이메일 사용자 이름입니다. *Name*은 임의의 문자열로, 대개 일부 양식에 지정된 이름입니다.

```
username@host.com
name username@host.com
name "<username@host.com>"
name <username@host.com>
name "username@host.com"
```

## bom

<bom> 요소는 작업 명세서(BOM)에 로깅될 정보를 정의합니다. 이 요소는 <match> 요소 안에 넣어야 합니다. <bom> 요소는 BOM에서 카테고리나 섹션을 지정해야 하며, <match> 요소가 수집한 번호 지정된 변수(\$1...\$n) 중 어떤 변수를 BOM 데이터 필드로 변환해야 하는지 정의합니다.

```
<bom category="Source" section="changesets" >
  <field name="Change" text="$1"/>
  <field name="Date" text="$2"/>
  <field name="User" text="$4"/>
</bom>
```

<bom> 요소가 BOM 보고서에 기록되는지 여부를 제어하려면 condition="함수" 속성을 사용하십시오. 함수의 값은 참 또는 거짓으로 평가되는 표현식입니다. 표현식이 참으로 평가되면 <bom> 요소의 정보가 BOM에 기록되고, 그렇지 않으면 기록되지 않습니다. 407 페이지의 『조건 함수』의 내용을 참조하십시오.

## bomformat

<bomformat> 요소를 사용하여 이전 <bom> 요소에서 수집된 데이터를 표시하는 방법을 정의하십시오. BOM 범주의 논리 이름을 지정하는 범주 속성은 물론 범주의 표시 이름을 지정하는 제목 속성을 갖습니다. <field> 요소를 포함하는 <section> 요소의 구조는 다음 예제에서와 같이 레이아웃을 정의합니다.

```
<bomformat category="Detail" title="Change Details">
  <section name="descriptions" parent="section name"

  expandable="yes">
    <field name="Description" title="Change Description"/>
  </section>
  <section name="diff">
    <field name="Diff" title="Differences"/>
  </section>
```

## 명령

어댑터 XML 파일은 여러 <command> 요소를 포함할 수 있습니다. 각 요소는 <interface> 요소에서 <run> 요소로 참조될 수 있는 이름 지정된 명령을 정의합니다. XML 파일의 여러 인터페이스가 같은 명령을 재사용할 수 있도록 <command> 요소를 <interface> 요소 외부에서 지정할 수 있습니다.



명령은 <resultsblock> 요소의 <match> 요소에 <run> 명령을 임베드하여 다른 명령을 호출할 수 있습니다.

아래 표시된 대로 <command> 요소가 <execute> 및 <resultsblock> 요소의 구조를 랩핑합니다.

```
<command name="p4_changes">
  <execute>
    command line
  </execute>
  <resultsblock>
    Has its own structure.
  </resultsblock>
</command>
```

또는 <execute> 요소는 블록에서 <integrate> 요소로 바뀔 수 있습니다.

<command> 요소의 모드를 식별하려면 모드 속성을 사용하십시오. 모드 속성 값은 다음과 같습니다.

- **conjoined**: 모든 명령 호출이 서버 처리를 위해 하나의 호출로 그룹화됩니다.
- **parallel**: 서버 슬롯을 사용할 수 있을 때 호출이 개별적으로 처리됩니다.
- **exec**: 명령이 시작되어 서버에 의해 즉시 처리됩니다.

## env

<env> 요소는 요소 내에서 프로젝트 링크와 함께 사용되는 환경에 복사할 수 있는 환경 변수(초기값과 함께)를 정의하는 데 사용됩니다. 각 <env> 요소에는 *name* 및 *value* 특성이 있어야 합니다. 값은 변수에 초기값을 제공합니다.

```
<env name="FILESPEC" value="//depot..." />
```

## execute

<command> 요소 내에서 <execute> 요소를 사용하여 명령을 지정하십시오. 요소 컨테츠는 어댑터가 사용하는 서버로 보내질 하나 이상의 텍스트 행입니다. <execute> 요소에서는 점 명령을 사용할 수 없습니다. <run> 요소가 <command> 요소를 호출할 때 시스템은 <execute> 요소 컨테츠의 위치 매개변수를 호출하는 <run> 요소에 지정된 매개변수로 바꿉니다. <execute> 요소 컨테츠의 \$1 매개변수는 첫 번째 매개변수로 바뀌고 \$2 매개변수는 두 번째 매개변수로 바뀝니다.

<execute> 요소에서 명령 실행을 제어하려면 condition="함수" 속성을 사용하십시오. 함수의 값은 참 또는 거짓으로 평가되는 표현식입니다. 표현식이 참으로 평가되면 시스템이 명령을 실행하고, 표현식이 거짓으로 평가되면 명령이 실행되지 않습니다. 407 페이지의 『조건 함수』의 내용을 참조하십시오.

```
<execute>
p4 changes -s submitted -t -i $2@$1,@now
</execute>
```

## field

<field> 요소를 <bom> 또는 <section> 요소에서 사용하여 필드를 지정하십시오.

<bom> 요소에서 사용할 때는 이름과 텍스트를 지정하십시오. 텍스트 속성은 데이터로 필드를 채우는 데 사용되는 변수를 정의합니다.

<bomformat> <section> 요소에서 사용할 때는 이름과 제목을 지정하십시오. 이름은 논리 이름을 지정하는 반면, 제목은 표시에 사용됩니다. <section>에 두 개 이상의 필드가 있는 경우, 순서 지정 속성을 포함시키십시오.

```
<section name="changesets">
  <field order="1" name="Change" title="Change ID"/>
  <field order="2" name="Date" title="Date"/>
  <field order="3" name="Time" title="Time"/>
  <field order="4" name="User" title="User ID"/>
  <field order="5" name="Client" title="Client"/>
  <field order="6" name="Comment" title="Comment"/>
</section>
```

<field> 요소가 BOM 보고서에 기록되는지 여부를 제어하려면 condition="함수" 속성을 사용하십시오. 함수의 값은 참 또는 거짓으로 평가되는 표현식입니다. 표현식이 참으로 평가되면 <field> 요소의 정보가 BOM에 기록됩니다. 표현식이 거짓으로 평가되면 해당 정보가 BOM에 기록되지 않습니다. 407 페이지의 『조건 함수』의 내용을 참조하십시오.

<field> 요소의 텍스트 형식을 정의하려면 템플릿(template) 속성을 사용하십시오. 예를 들어, 텍스트가 문자열인 경우 템플릿 값은 "Hello \$VALUE"일 수 있습니다. 필드가 BOM 보고서에 기록될 때 \$VALUE는 필드 텍스트로 바뀝니다.

## integrate

<integrate> 요소는 <execute> 요소와 유사합니다. <execute> 요소 대신에 <integrate> 요소를 사용할 수 있습니다. <execute> 요소와 마찬가지로 <integrate> 요소는 실행될 명령행을 지정합니다. 차이점은 다음과 같습니다.

- 명령행은 어댑터를 실행하는 서버가 아닌 관리 콘솔 시스템에서 실행됩니다.
- 명령행은 \integration 디렉토리(설치 디렉토리의 서브디렉토리)를 현재 디렉토리로 사용합니다.

<integrate> 요소는 관리 콘솔 컴퓨터에(특히, \integration 디렉토리에) 위치한 애플리케이션이나 스크립트를 실행하는 데 유용합니다.

<run> 요소가 <integrate> 요소를 포함하는 <command> 요소를 호출할 때 시스템은 <integrate> 요소의 위치 변수를 호출하는 <run> 요소에 지정된 매개변수로 바꿉니다. <integrate> 요소의 \$1은 첫 번째 매개변수로 바뀌고 \$2는 두 번째 매개변수로 바뀝니다.

<execute> 요소의 경우와 마찬가지로 <integrate> 요소에서는 점 명령을 사용할 수 없습니다.

IBM Rational ClearQuest 어댑터에서 발췌한 다음 예제는 CQperl 명령(Perl 코드 실행을 위한 ClearQuest 프로그램)을 실행하고 \integration 디렉토리에 있는 Perl 스크립트의 이름을 제공하여 ClearQuest로 데이터를 보냅니다. 이 예제에서는 ClearQuest가 관리 콘솔 시스템에 설치되어 있다고 가정합니다.

```
<integrate>
cqperl bfcqresolve.pl $2 Fixed "Fixed in build $BF_TAG"
</integrate>
```

## 인터페이스

<interface> 요소는 어댑터로의 시작점에 대한 컨테이너입니다. 포함된 요소가 어댑터의 프로그램 논리를 정의합니다. <setenv>, <run>, <ontempenv>, <onproject> 및 <adduser> 요소를 포함합니다.

어댑터 템플릿에는 여러 시작점이 있을 수 있습니다. 여러 시작점이 있는 어댑터 템플릿을 작성하는 경우, 다음을 수행하십시오.

- 각 시작점을 식별하도록 name 속성을 사용하십시오.
- 어댑터 템플릿이 이름 스펙 없이 호출되는 경우에 실행할 시작점을 식별하려면 default 속성을 사용하십시오.

시작점을 지정하려면 템플릿에 대한 .source 호출에서 시작점 이름을 매개변수로 사용하십시오.

예제: MyAdaptorTemplate라는 어댑터 템플릿을 작성합니다. 여기에 이름 지정된 시작점을 정의하도록 다음 코드를 넣고 기본값이 되게 합니다.

```
<interface name="By Date" default="true">
</interface>
```

이 인터페이스를 이름으로 호출하려면 프로젝트 단계에서 다음 명령을 사용하십시오.

```
.source MyAdaptorTemplate "By Date"
```

속성:

**name** 선택사항. 이 인터페이스의 이름입니다. 이름을 매개변수로 사용하여 어댑터가 호출되면, 이름이 해당 매개변수와 일치하는 인터페이스가 사용됩니다.

**default**

선택사항. 예 또는 아니오로 설정합니다. 예인 경우, 이름 매개변수 없이 어댑터가 호출될 때 해당 인터페이스가 사용됩니다.

## match

<match> 요소는 <resultsblock> 요소에서 사용하여 출력 행을 처리합니다. <match> 요소는 일치 행을 정의하는 패턴 속성을 갖습니다. 패턴은 Perl 정규식입니다.

일치 패턴은 \$1...\$n 변수에 저장되는 삽입 표현식을 포함할 수 있습니다.

```
<match pattern="^Change (\d+) on (.*) (.*) by (.*)@(.*) '(.*)'$">
```

<match> 요소는 <adduser>, <setenv>, <bom> 및 <field>를 하위 요소로 사용합니다. 보다 광범위한 예제를 보려면 566 페이지의 『resultsblock』에 대한 참조 정보를 참조하십시오.

## notify

<notify> 요소는 이메일 및 분배 목록을 지정합니다. 일반적으로 <onproject> 요소에서 사용되어 실행 결과에 따라 알람을 지정합니다.

속성:

**group** 이메일을 받는 사람입니다. 어댑터에 정의된 사용자 그룹입니다.

**message**

이메일 본문의 내용입니다. 텍스트를 지정합니다.

**subject**

이메일 제목입니다. 텍스트를 지정합니다.

## onproject

<onproject> 요소는 시스템이 프로젝트 단계를 실행한 후에 수행하는 알람 조치를 정의합니다. 요소는 패스 또는 실패 작업에 대해 조치가 수행되는지 여부를 지정하는 필수 결과 속성을 사용합니다. 일반적으로 어댑터 XML 파일에는 두 개의 <onproject> 요소(패스 경우와 실패 경우에 하나씩)가 포함됩니다. 다음 예제는 notify 요소를 사용하여 프로젝트 패스 또는 실패 여부에 따라 다른 메시지를 전송하는 <onproject> 요소 쌍을 보여줍니다.

```
<onproject result="fail">
  <notify group="MyChangers"
    subject="Run $BF_TAG ($CurDate) Failed."

    message="$Changing$Changes"/>
</onproject>

<onproject result="pass">
  <notify group="MyChangers"
    subject="Run $BF_TAG ($CurDate) Passed."

    message="$Changing$Changes"/>
</onproject>
```

## ontempenv

<ontempenv> 요소는 <interface> 요소에서 사용되며 if-then 문과 유사하게 작동합니다. 이 요소를 사용하여 패스 또는 실패 값을 프로젝트에 리턴하십시오. 패스는 시스템이 나머지 프로젝트를 계속 실행해야 함을 표시하는 반면, 실패는 중지해야 함을 표시합니다. 이 요소는 일반적으로 인터페이스가 프로젝트를 새로 실행할만한 관련 변경사항을 찾았는지 여부를 표시하는 데 사용됩니다.

시스템은 <run> 요소에 지정된 명령을 실행한 후에 <ontempenv> 요소를 처리합니다. 이 요소의 name 속성을 사용하여 임시 환경 변수를 지정하고, state 속성을 사용하여 값을 지정하십시오.

<ontempenv>는 <step> 요소를 래핑하며, <run> 요소 명령이 실행된 후에 임시 환경 변수 이름과 상태가 존재하는 경우에만 실행됩니다.

```
<ontempenv name="Changes" state="empty">
  <step result="FAIL"/>
</ontempenv>
```

## PROJECT\_INTERFACE

<PROJECT\_INTERFACE> 요소는 어댑터 템플릿의 기타 모든 태그를 래핑합니다. 어댑터 유형을 표시하는 하나의 속성(IFTYPE)을 가집니다. 올바른 유형은 소스, 테스트 및 결함입니다.

```
<PROJECT_INTERFACE IFTYPE="Source">
...all other elements...
</PROJECT_INTERFACE>
```

## relate

<relate> 요소는 아티팩트와 사용자 사이의 관계를 지정합니다. 프로젝트에 정의된 변경자에게 알림 유형의 로그 필터와 함께 사용되며, 어댑터를 호출하는 단계의 바로 앞에 실행하는 단계의 Result 속성에서 사용됩니다. 변경자에게 알림 필터는 일반적으로 성공 또는 실패를 표시하는 예상 로그 행을 지정합니다. 단계 실행 시 필터는 <relate> 요소에서 아티팩트로 지정된 텍스트와 필터가 비교됩니다. 일치하는 경우, 아티팩트와 연관된 사용자에게 이메일이 전송됩니다.

368 페이지의 『필터 조치』에서 변경자에게 알림 필터에 대한 설명을 참조하십시오.

속성:

### artifact

로그 필터로 검색되는 텍스트와 일치되는 텍스트입니다.

**user** 아티팩트와 연관된 사용자입니다. 일치가 있는 경우, 시스템에서 이 사용자에게 이메일을 전송합니다.

**text** 로그 필터와 이 관계 사이에 일치가 있을 때 BOM에 로그할 텍스트입니다.

## resultsblock

<resultsblock> 요소는 시스템이 관련 <execute> 요소에서 실행된 명령행 결과를 처리하는 방법을 정의합니다. <resultsblock> 요소는 <command> 요소에서만 사용됩니다. <resultsblock> 요소를 중첩하여 결과를 파티션할 수 있습니다.

<resultsblock> 요소는 Perl 정규 표현식을 사용하여 처리할 출력 행 범위를 정의하는 선택적 beginpattern 및 endpattern 속성을 가질 수 있습니다. 그러면 서로 다른 <resultsblock> 요소를 사용하여 다른 범위를 처리할 수 있습니다. 다음 의사 코드는 <resultsblock> 요소의 구조를 표시합니다.

```
<resultsblock startpattern="" endpattern="" >
  <match>
    <bom>
      <field/>
    </bom>
    <adduser/>
    <setenv/>
    <run/> (The <run> element can be used

to run commands within other commands)

  </match>
  <setenv/>
</resultsblock>
```

다음 예제는 <resultsblock>, <match> 및 <bom> 요소가 함께 작동하는 방법을 보여줍니다.

```
<resultsblock
beginpattern="^Change (\d+) by (.*)@(.*) on (.*) (.*)$"
endpattern="^Differences ...$"
  <match pattern="(?!^(:?!Differences ...).)*$.?">
    <bom category="Detail" section="descriptions">
      <field name="Description" text="$_" />
    </bom>
  </match>
</resultsblock>
```

## run

<interface> 요소에서 <run> 요소를 사용하여 실행한 이름 지정된 명령을 지정합니다. 명령은 나중에 같은 XML 파일에 정의됩니다. <run> 요소는 빈 요소입니다.

속성:

### condition

condition="기능" 형식의 선택사항입니다. 함수의 값은 참 또는 거짓으로 평가되는 표현식입니다. 표현식이 참으로 평가되면 명령이 실행됩니다. 거짓으로 평가되면 실행되지 않습니다. 407 페이지의 『조건 함수』의 내용을 참조하십시오.

### command

필수. 실행할 정의된 명령의 이름을 지정합니다. 명령은 이름이 지정되고 <command> 요소에서 정의됩니다.

**dir** 필수. 명령을 실행할 디렉토리를 지정합니다. dir은 명령이 실행되는 서버 자원에서 설정된 경로의 확장으로 해석됩니다.

**mode** 선택사항. 실행 명령의 실행 모드를 지정합니다. 값은 다음 중 하나일 수 있습니다.

- **conjoined**: 모든 명령 호출이 서버 처리를 위해 하나의 호출로 그룹화됩니다.
- **parallel**: 서버 슬롯을 사용할 수 있을 때 호출이 개별적으로 처리됩니다.
- **exec**: 명령이 시작되어 서버에 의해 즉시 처리됩니다.

### params

필수. 명령에 전달할 매개변수를 지정합니다. 공백을 사용하여 매개변수를 분리하십시오.

**server** 필수. 명령을 실행할 서버 자원을 지정합니다. 널(null)로 설정되면, 어댑터를 실행하는 단계에 사용된 서버에서 명령이 실행됩니다. 서버를 널(null)로 설정하려면 server=""를 사용하십시오.

### timeout

필수. 명령 제한시간이 초과되기 전의 초 수를 지정합니다.

필수 속성을 보여주는 예제:

```
<run command="UpdateEnv" params="" server="" dir="/" timeout="360"/>
```

예를 들어 다음과 같습니다.

```
<run command="p4_changes" params="$LAST_RUN $FILESPEC $LAST_VER"
server="$P4CLIENT" dir="/" timeout="360"/>
```

### section

<section> 요소를 사용하여 BOM 범주 부분 표시 방법을 정의하십시오. 이 요소는 이름 속성을 갖습니다. <bomformat> 요소 내에서만 <section> 요소를 사용할 수 있습니다.

### setenv

<setenv> 요소를 사용하여 <interface> 또는 <match> 요소에서 환경 변수 값을 초기화하십시오. <setenv> 요소는 기타 요소를 포함하지 않습니다.

다음 세 가지 다른 방법으로 이 요소를 사용할 수 있습니다.



- 그룹 이름을 지정할 때 `.set` 명령과 유사하게 작동합니다. 현재 단계에서 사용되는 복사본이 아니라 데이터베이스의 마스터 레코드에 변수값을 설정합니다. 현재 단계에서 실행 중인 어댑터는 변경을 알지 못합니다. 이 방식은 새 변수를 작성할 수 없습니다.
- 그룹 이름을 지정하지 않을 때 `.bset` 명령처럼 작동합니다. 실행 중인 작업의 환경에 변수값을 설정합니다. 작업의 모든 단계에 변경을 사용할 수 있습니다. 이 방식은 새 변수를 작성할 수 있습니다.
- 그룹 이름을 지정하지 않으면서 임시 변수(`type="temp"`)를 지정할 때 어댑터 논리 사용을 위해서만 임시 변수를 설정합니다. 어댑터 단계 실행 이후에 변수가 지속되지 않습니다. 이 방식은 새 변수를 작성할 수 있습니다.

속성:

#### condition

`condition="기능"` 형식의 선택사항입니다. 함수의 값은 참 또는 거짓으로 평가되는 표현식입니다. 표현식이 참으로 평가되면 명령이 실행됩니다. 거짓으로 평가되면 실행되지 않습니다. 407 페이지의 『조건 함수』의 내용을 참조하십시오.

**eval** 선택사항. 참 또는 거짓으로 설정합니다. 참이면 어댑터가 `value` 속성 표현식 평가를 시도하고 결과를 저장합니다.

**group** 선택사항. 변수가 정의된 Build Forge 환경을 지정합니다. 환경 이름을 지정할 때 지정된 환경에서 기존 변수를 참조해야 합니다.

[ADAPTOR]를 값으로 지정하면 실행 시 값이 설정됩니다. 단계의 환경 또는 어댑터를 호출하는 어댑터 링크로 설정됩니다.

Build Forge는 여러 환경의 같은 이름 변수를 허용합니다. 환경 상속 및 환경 포함의 우선순위는 실행 시 변수에 지정할 값을 결정하는 방법에 영향을 줄 수 있습니다.

**name** 필수. 설정할 변수의 이름을 지정합니다. 값은 변수일 수 있습니다. 이 경우, 실행 시 변수 이름이 설정됩니다.

**type** 선택사항. 변수를 설정하는 방법을 지정합니다. 다음 값 중 하나를 가집니다.

- `append 텍스트`: 지정된 값을 기존 값 다음에 넣습니다. 선택적 텍스트가 지정되면 값 사이에 텍스트가 놓입니다.
- `once`: 아직 설정되지 않은 경우에만 변수를 설정해야 합니다.
- `prepend 텍스트`: 해당 값을 기존 값 앞에 넣습니다.
- `temp`: 어댑터 컨텍스트에서만 변수를 설정해야 합니다. 선택적 텍스트가 지정되면 값 사이에 텍스트가 놓입니다. 아래 예제를 참조하십시오.

**value** 필수. 변수에 설정하는 값입니다. `eval` 속성도 지정된 경우에는 평가할 표현식일 수 있습니다. 평가 결과가 값으로 저장됩니다.

예제:



다음 예제는 value 속성의 표현식을 평가하고, LAST\_VER 변수에 결과를 저장합니다. \$LAST\_VER보다 크게 또는 \$1 변수의 값으로 설정됩니다.

```
<setenv group="Adaptor" name="LAST_VER"
  value="$LAST_VER>$1?$LAST_VER:$1" eval="true" />
```

다음 예제는 Changes 변수에 데이터를 추가하기 전에 줄 바꾸기 문자(\n)를 삽입합니다.

```
<setenv name="Changes" value="$4 - $1 - $6" type="temp append\n" />
```

다음 예제는 INFOPATH 변수 앞에 추가되는 값 뒤에 콜론을 삽입합니다.

```
<setenv name="INFOPATH" value="/usr/local" type="temp prepend:" />
```

## 단계

<step> 요소는 <ontempenv> 요소에서만 사용됩니다. 특수 어댑터 단계의 결과를 지정합니다. 이 요소는 비어 있습니다. 이 예제는 <step> 요소의 두 가지 양식을 보여줍니다.

```
<step result="FAIL"/>
```

```
<step result="pass"/>
```



## 제 27 장 Rational Build Forge 및 Rational ClearCase

BuildForge는 소스 코드 관리를 위해 Rational ClearCase와 함께 작업할 수 있습니다.

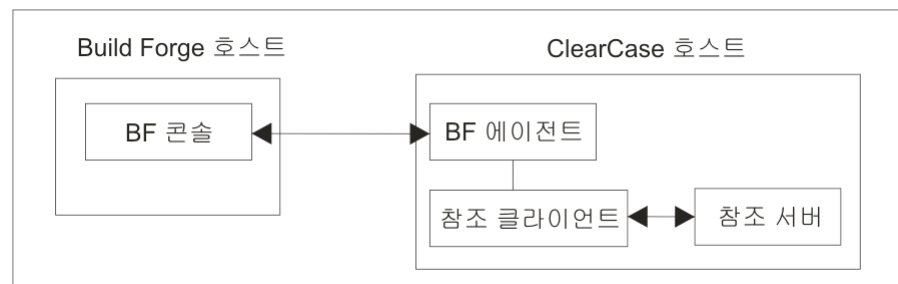
이 주제에서는 Rational ClearCase와 통합하기 위한 추가 설정 요구사항을 설명하고 제공된 샘플 어댑터 템플릿의 특성을 요약합니다.

ClearCase 어댑터 템플릿 샘플은 기준선에 대한 변경사항을 분석하는 방법을 제공합니다. 일반적으로 변경 분석은 빌드 방지를 위해 사용됩니다. 기준선 컴포넌트가 변경되지 않은 경우 다시 빌드되지 않습니다.

### ClearCase 어댑터에 대한 추가 설정 요구사항

일반 요구사항은 529 페이지의 『어댑터 요구사항』의 내용을 참조하십시오. 어댑터에 대한 일반 요구사항을 이행하는 것 외에도 다음을 수행하십시오.

1. ClearCase 서버에 연결할 수 있는 호스트에 Build Forge 에이전트를 설치하십시오.
2. 에이전트 호스트에 ClearCase 전체 클라이언트를 설치하십시오.
3. ClearCase 클라이언트를 통해 명령을 실행할 수 있도록 에이전트에 대한 환경을 설정하십시오.
4. 필요한 ClearCase 보기를 시작하는 방법과 시기 및 구현하는 방법을 판별하십시오. 제공된 템플릿은 동적 보기의 사용을 가정하고 cleartool 명령이 실행될 때 해당 보기를 시작하는 것을 포함합니다. 에이전트 호스트에 ClearCase 전체 클라이언트를 설치하십시오.



### ClearCase 보기

ClearCase는 데이터에 액세스하기 위한 수단으로 보기를 사용합니다. 두 유형의 보기를 사용할 수 있습니다.

- 동적 보기: 동적 보기로 로컬 시스템에 복사하지 않고 사용자가 지정하는 VOB에 있는 요소의 최신 버전을 볼 수 있습니다. 동적 보기를 시작해야 합니다. 보기에 마운트하도록 VOB를 지정합니다.
- 스냅샷 보기: 스냅샷 보기는 저장소의 사본입니다. 사용할 오브젝트의 버전을 지정할 수 있습니다. 체크인된 최신 항목을 가져오려면 스냅샷 보기를 수동으로 업데이트해야 합니다.

### 동적 보기

동적 보기는 보도록 선택하는 보기 컨텍스트에서 어떤 사항이든 변경될 때마다 계속 업데이트되는 작업공간을 표시합니다. 사용자가 체크아웃하는 요소만 작업공간에 복사되며 다른 사용자가 이를 사용할 수 있도록 하려면 이를 체크인해야 합니다. 동적 보기를 사용할 때 세 가지 스펙을 사용하여 원하는 데이터에 액세스할 수 있습니다.

- 보기: 구성 스펙을 통해 표시할 버전의 목록을 정의합니다. 명령을 사용하여 보기를 시작해야 합니다. 이는 로컬 디스크에 콘텐츠의 루트 디렉토리로 표시됩니다. 이는 보기 루트라고도 합니다. 일반적으로 사용되는 디렉토리는 다음과 같습니다.
  - Windows: 공유 드라이브인 M: 드라이브
  - UNIX 또는 Linux: /view, 마운트된 파일 시스템
- 보기 컨텍스트: 보기 태그를 통해 사용할 콘텐츠의 디렉토리를 정의합니다. 보기 태그는 ClearCase에서 정의됩니다. 디렉토리는 보기가 시작될 때 사용 가능해집니다.
- VOB(versioned object base): 보기 컨텍스트의 서브디렉토리를 정의합니다. VOB는 ClearCase에서 정의됩니다. 보기 컨텍스트에 VOB를 명시적으로 마운트합니다. 이 서브디렉토리는 버전화된 아티팩트(소스 파일 및 기타 아티팩트)를 포함합니다.

로컬 시스템의 VOB에 대한 경로는 다음 세 가지 항목을 통해 구성됩니다.

- Windows 시스템의 경로 예제(기본값 사용): M:\view\_tag\vob\_name
- UNIX 및 Linux 시스템의 경로 예제(기본값 사용): /view/view\_tag/vob\_name

### 스냅샷 보기

스냅샷 보기를 사용할 때 보기 컨텍스트를 지정하고 데이터도 로드해야 합니다. 그러면 사용자가 지정하는 모든 데이터의 로컬 사본을 가지게 됩니다. 데이터를 마지막으로 로드한 후 체크인한 변경사항을 가져오려면 스냅샷 보기를 다시 로드해야 합니다.

## 동적 보기를 지원하는 Rational Build Forge 기능

특수 환경 변수를 사용하여 Build Forge 작업에서 동적 보기를 정의하고 시작할 수 있습니다.

### **CLEARCASE\_VIEW**

지정된 ClearCase 보기를 시작합니다. 이 변수에 지정된 보기는 존재해야 하며 이 변수를 사용하는 단계는 "absolute"로 설정되어야 합니다. Microsoft Windows를 실행하는 시스템에서 이 변수는 `bfagent.conf`의 에이전트에 대한 `cc_suppress_server_root` 매개변수와 함께 사용해야 합니다.

### **\_CLEARCASE\_VIEWS**

명령 실행 전에 시작할 ClearCase 보기 목록을 지정합니다. 값을 쉼표로 분리된 보기 목록으로 설정하십시오(예제: "View1,View2,View3").

### **\_CLEARCASE\_VOBS**

명령 실행 전에 마운트할 ClearCase VOB 목록을 지정합니다. 값을 쉼표로 분리된 VOB 목록으로 설정하십시오(예제: "\Vob1,\Vob2,\Vob3").

또한 에이전트 매개변수를 사용하여 에이전트 호스트의 보기를 관리해야 합니다. ClearCase와 함께 사용하는 에이전트가 Windows 시스템에서 실행 중인 경우 에이전트 매개변수를 설정해야 합니다. 이는 또한 이 에이전트를 통해 사용하려는 보기 루트가 작업으로 설정되는 보기 루트와 다른 경우에도 유용합니다. 이는 둘 이상의 에이전트에 대해 작업을 사용했고 각 에이전트 호스트 시스템에서 스펙이 서로 달라야 하는 경우 발생할 수 있습니다.

### **ccviewroot root-path**

이 호스트의 기본 보기 루트를 지정합니다. 자세한 정보는 `init`에 대한 ClearCase 문서를 참조하십시오. 내부 기본값은 다음과 같습니다.

- Windows 시스템: `ccviewroot M:`
- UNIX 또는 Linux 시스템: `ccviewroot /view`

### **cc\_suppress\_server\_root**

이를 설정하면 보기 경로가 `ccviewroot`로 설정되는 경로가 됩니다. 설정하지 않을 경우에는 서버 정의에 설정된 경로가 `ccviewroot`로 설정되는 경로에 추가됩니다. 이 설정에는 값이 필요 없습니다. 이 설정이 `bfagent.conf`에 있는 경우에는 설정된 것입니다.

## ClearCase 어댑터 템플릿 샘플

다음 어댑터 템플릿 샘플이 제공됩니다.

### **ClearCaseBaseline**

1. ClearCase 보기에서 디렉토리를 스캔합니다.
2. ClearCase가 보고하는 분기 및 버전 정보를 BOM 보고서에 기록합니다.

사용되는 변수:

- INT\_STREAM
- VIEW
- PROJECT\_VOB
- CCSERVER
- UNIXCLIENT

#### **ClearCaseByBaselineActivities**

1. ClearCase 보기의 콘텐츠에서 새 기준선을 작성합니다.
2. 새 기준선과 이전 어댑터 실행의 기준선을 비교하여 변경 활동을 식별합니다.
3. 각 변경 활동에 대해 BOM 보고서에 활동, 변경된 파일, 사용자, 날짜, 주식 및 버전 정보를 기록합니다.
4. 변경된 각 파일에 대해 diff 명령 출력의 변경 세부사항을 BOM 보고서에 기록합니다.

사용되는 변수:

- CurDate
- LAST\_RUN
- BASELINE
- VIEW
- VOB\_PATH
- PROJECT\_VOB
- CCSERVER
- UNIXCLIENT

#### **ClearCaseByBaselineVersions**

1. ClearCase 보기의 콘텐츠에서 새 기준선을 작성합니다.
2. 새 기준선과 이전 어댑터 실행의 기준선을 비교하여 변경된 파일을 식별합니다.
3. 변경된 각 파일에 대해 BOM 보고서에 파일 이름, 버전, 날짜, 사용자 및 주식 정보를 기록합니다.
4. 변경된 각 파일에 대해 diff 명령 출력의 변경 세부사항을 BOM 보고서에 기록합니다.

사용되는 변수:

- CurDate
- LAST\_RUN

- LABEL
- BASELINE
- VIEW
- VOB\_PATH
- PROJECT\_VOB
- CCSERVER
- UNIXCLIENT
- \_CHAR\_NATIVE

### **ClearCaseByDate**

1. ClearCase 보기에서 두 날짜 간의 변경사항을 조회합니다. 기본 날짜는 현재 시간소인 및 이전 어댑터 실행의 시간소인입니다.
2. 변경된 각 파일에 대해 BOM 보고서에 파일 이름, 버전, 날짜, 사용자 및 주식 정보를 기록합니다.
3. 변경된 각 파일에 대해 diff 명령 출력의 변경 세부사항을 BOM 보고서에 기록합니다.

사용되는 변수:

- CurDate
- LAST\_RUN
- LABEL
- BASELINE
- VIEW
- VOB\_PATH
- PROJECT\_VOB
- CCSERVER
- UNIXCLIENT
- \_CHAR\_NATIVE

### **ClearCaseByLabel**

1. 새 레이블을 작성하여 ClearCase 보기의 콘텐츠에 적용합니다.
2. 새 레이블과 이전 어댑터 실행의 레이블을 비교하여 변경된 파일을 식별합니다.
3. 변경된 각 파일에 대해 BOM 보고서에 파일 이름, 버전, 날짜, 사용자 및 주식 정보를 기록합니다.
4. 변경된 각 파일에 대해 diff 명령 출력의 변경 세부사항을 BOM 보고서에 기록합니다.

사용되는 변수:

- CurDate
- LAST\_RUN
- LABEL
- BASELINE
- VIEW
- VOB\_PATH
- PROJECT\_VOB
- CCSERVER
- UNIXCLIENT
- \_CHAR\_NATIVE

## ClearCase 어댑터 변수

이 표는 어댑터 템플릿에 대한 변수 목록의 참조입니다.

환경 변수 이름	설명
BASLINE	ByBaseline 어댑터의 경우, 사용자가 사용자의 어댑터를 사용하여 기준선별로 차이점을 생성할 때 시스템은 이 값을 기준으로 사용합니다.
CCSERVER	이 변수를 ClearCase 클라이언트 및 Build Forge 에이전트가 설치된 호스트의 이름으로 설정하십시오.
CurDate	어댑터에 현재 날짜를 제공합니다. .date 명령을 사용하여 ClearCase가 예상하는 형식으로 날짜를 생성합니다. 이 값을 변경하지 마십시오.
LABEL	ByLabel 어댑터의 경우, 사용자가 사용자의 어댑터를 사용하여 레이블별로 ByLabel 어댑터와의 차이점을 생성할 때 시스템은 이 값을 레이블로 사용합니다.
LAST_RUN	ByDate 어댑터의 경우, 시스템은 이 값을 사용하여 변경사항이 발생했는지 여부를 판별합니다. 값은 마지막 성공적 실행 날짜입니다. 어댑터를 강제 실행하기 위해 어댑터를 테스트할 때 일부 변경사항에 선행하는 날짜를 선택하여 이 값을 조작할 수 있습니다. 실행을 계속하도록 허용하는 경우, 어댑터는 이 값을 현재 날짜로 자동으로 업데이트합니다. 기본값은 1-Jan-05.00:00:00입니다.
PROJECT_VOB	ByBaseline 어댑터를 사용할 때 이 변수를 프로젝트 VOB의 이름으로 설정하십시오(UCM ClearCase에서만 사용됨). 예를 들어 \ProjectVob를 설정할 수 있습니다.
UNIXCLIENT	플랫폼 특정 정보를 설정하기 위해 사용됩니다. 클라이언트가 Windows에서 실행 중인 경우 0으로 설정하십시오. 클라이언트가 UNIX 또는 Linux에서 실행 중인 경우 1로 설정하십시오.
VIEW	이 변수를 어댑터와 함께 사용할 ClearCase 보기의 이름으로 설정하십시오.
VOB_PATH	이 값을 컴포넌트 VOB의 이름으로 설정하고 선택적으로 서브디렉토리를 설정하십시오. 이름이 여러 개인 경우 쉼표로 구분되는 목록을 사용하십시오.
_CHAR_NATIVE	내부적으로 사용되며 항상 1로 설정됩니다.

다음 트리거 변수는 보기를 시작하고 vob를 마운트하기 위해 사용할 수도 있습니다. 그러나 어댑터 변수에 설정된 보기 및 vob와는 독립적입니다.

- CLEARCASE\_VIEW
- \_CLEARCASE\_VIEWS



- \_CLEARCASE\_VOBS

337 페이지의 『트리거 변수 참조』의 내용을 참조하십시오.



---

## 제 28 장 Rational Build Forge 및 Rational ClearQuest

BuildForge는 빌드 레코드 업데이트를 위해 Rational ClearQuest와 함께 작동할 수 있습니다.

Build Forge에는 두 가지의 ClearQuest와의 통합이 있습니다.

- 작업 상태를 기반으로 하는 자동 빌드 레코드 작성 또는 업데이트. 이 기능은 프로젝트 환경에서 필요한 환경 변수가 설정된 경우에 자동으로 활성화됩니다.
- ClearQuest 어댑터

이 기능은 서로 완전히 독립적입니다. 특히, 어댑터는 어댑터에 대해 작성된 환경과 연관됩니다. 해당 환경의 변수는 자동 빌드 레코드를 활성화하기 위해 설정된 변수와 독립적입니다.

### 빌드 레코드의 자동 생성 설정

시스템은 로그 데이터로의 링크를 사용하여 IBM Rational® ClearQuest® 데이터베이스에 빌드 레코드를 자동으로 작성할 수 있습니다. 뿐만 아니라, 작업 패스 시에 시스템은 ClearQuest 데이터베이스를 업데이트할 수 있으며 작업이 완료되면 종료 시간과 달성된 단계에 대한 요약을 기록합니다. 이 기능에는 Rational ClearQuest 버전 7.0 이상이 필요합니다.

ClearQuest 데이터베이스를 업데이트하도록 프로젝트를 구성하는 경우 시스템은 다음과 같이 작성 또는 업데이트 빌드 레코드를 수행합니다.

#### 작업 시작

시스템은 작업을 실행할 때 ClearQuest 빌드 레코드를 작성합니다. 빌드 레코드는 제출 상태에 있으며 작업 로그 URL, 시작 시간, 릴리스 이름 및 ID와, "빌드 XYZ가 시작되었음"을 표시하는 로그 항목을 포함합니다. 소스 제어 어댑터가 작업을 취소하는 경우(예를 들어, 소스 변경사항이 없어서) ClearQuest 빌드 레코드가 작성되지 않습니다.

**참고:** 프로젝트에 다른 프로젝트가 체인된 경우, 새 프로젝트는 자체의 고유 ClearQuest 빌드 ID를 가져옵니다.

#### 작업 패스/실패

작업이 패스 또는 실패할 때 시스템은 ClearQuest에서 빌드 상태를 완료 또는 실패로 변경하고 빌드 종료 시간을 설정하며, ClearQuest 빌드 로그에 작업 단계 요약을 저장합니다. 요약에는 각 단계의 이름, 결과 상태 및 서버가 포함됩니다.

## 작업 다시 시작

작업이 다시 시작될 때 시스템은 ClearQuest에서 빌드 상태를 제출로 변경하고 "빌드 XYZ가 다시 시작되었음"을 표시하는 ClearQuest 빌드 로그 항목을 작성합니다.

특수 환경 변수를 통해 자동 빌드 레코드 업데이트를 구성합니다. ClearQuest 데이터베이스에 프로젝트를 링크하려면 다음 표의 변수가 프로젝트 환경에 포함되어 있는지 확인하십시오.

**참고:** 이러한 변수가 프로젝트 환경에 있어야 하며, 단계에 변수를 추가하는 것으로는 부족합니다. 그러나 다른 환경을 통해 이러한 변수를 포함하는 포함 유형으로 설정된 변수를 사용할 수 있습니다. 또한 CQ\_RELEASE\_NAME 값은 유일하게 프로젝트마다 다를 수 있으므로 다른 변수를 포함하는 환경을 작성하고 포함 유형의 변수를 사용하여 프로젝트 환경에 해당 환경을 포함시킬 수 있습니다(이 경우, CQ\_RELEASE\_NAME을 프로젝트 특정 환경 변수로 지정할 수도 있음).

Build Forge 작업에서 빌드 레코드의 자동 업데이트를 활성화하려면 프로젝트에 대해 다음 환경 변수를 설정해야 합니다. 이 환경 변수는 단계 레벨에서 작동하지 않습니다.

변수	설명
CQ_DBNAME	필수. 업데이트할 ClearQuest 데이터베이스의 이름
CQ_DBSET	ClearQuest 데이터베이스 설정 값. 필수가 아니며 기본값은 공백입니다.
CQ_INTERACTION	ClearQuest 빌드 레코드 작성을 사용하도록 프로젝트 환경에 올바른 환경 변수가 정의되어 있지만 빌드 레코드를 작성하지 않으려면 이 변수를 OFF로 설정하여 빌드 레코드 작성을 사용하지 않도록 하십시오.  빌드 레코드 작성을 사용하려면 이 변수를 ON으로 설정하십시오.  참고: ClearQuest 어댑터 중 하나를 사용 중인 경우, 이 환경 변수를 OFF로 설정하십시오. 어댑터는 빌드 레코드와 직접 상호작용합니다.
CQ_PASSWORD	필수. ClearQuest 데이터베이스에 로그인할 때 사용할 비밀번호. 필수는 아니며, 기본값은 공백으로 설정됩니다.
CQ_RELEASE_NAME	필수. ClearQuest 데이터베이스에서 업데이트할 릴리스의 이름
CQ_USER	필수. ClearQuest 데이터베이스에 로그인할 때 사용할 사용자 이름

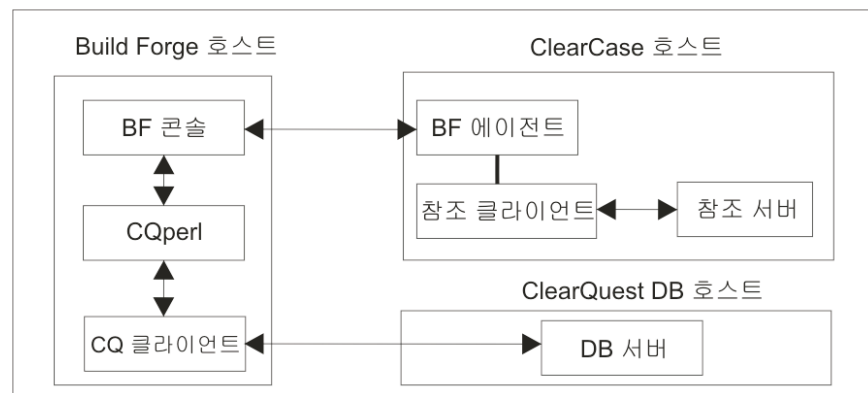
## ClearQuest 어댑터에 대한 추가 설정 요구사항

ClearQuest 어댑터 템플릿 샘플은 ClearQuest에서 빌드 레코드를 업데이트하고 ClearCase를 스캔하는 방법을 제공합니다. 이는 일반적으로 Build Forge에서 실행되는 빌드의 성공 또는 실패로 링크됩니다. 일반 요구사항은 529 페이지의 『어댑터 요구사항』의 내용을 참조하십시오. 또한 다음 구성을 완료해야 합니다.

- 소스를 스캔하려면 ClearCase에 액세스하십시오. 작업을 실행하는 동안 어댑터는 에이전트 및 ClearCase 클라이언트를 통해 cleartool 명령을 실행합니다. 어댑터는 cqperl(ClearQuest Perl API)을 사용하여 명령을 실행합니다.
- 빌드 레코드를 업데이트하려면 ClearQuest에 액세스하십시오. 작업을 실행하는 동안 어댑터는 콘솔 호스트의 cqperl 스크립트를 직접 실행합니다. 이는 Cqperl 유틸리티로 해석되고 ClearQuest 클라이언트를 통해 실행되며 둘 모두 콘솔 호스트에 설치됩니다.

다음은 수행하십시오.

1. ClearCase 서버에 연결할 수 있는 호스트에 Build Forge 에이전트를 설치하십시오.
2. 에이전트 호스트에 ClearCase 전체 클라이언트를 설치하십시오.
3. ClearCase 클라이언트를 통해 명령을 실행할 수 있도록 에이전트에 대한 환경을 설정하십시오.
4. Build Forge 콘솔 호스트에 ClearQuest 전체 클라이언트를 설치하십시오.
5. cqperl(ClearQuest Perl API)를 시스템 경로에 직접 추가하십시오.
6. Build Forge 호스트의 ClearQuest 클라이언트가 ClearQuest 데이터베이스에 액세스하는 데 사용할 수 있는 연결을 정의하십시오. ClearQuest 클라이언트 호스트에서 해당 조치를 수행하십시오.
  - a. cqreg 명령을 사용하여 데이터베이스 세트(cqreg add\_dbset)를 추가하십시오.
  - b. CQ 유지보수 도구를 사용하여 ClearQuest 데이터베이스에 대한 연결을 설정하십시오.
7. 필요한 ClearCase 보기를 시작하는 방법과 시기 및 구현하는 방법을 판별하십시오.



에이전트를 설치할 필요는 없습니다. ClearQuest 어댑터는 ClearQuest Perl API를 사용하여 클라이언트를 통해 ClearQuest와 직접 통신합니다.

**중요사항:** ClearQuest 어댑터는 단계에서 점(dot) 명령으로만 호출할 수 있습니다. 소스 어댑터가 아니므로 어댑터 링크를 사용할 수 없습니다.

## ClearQuest 어댑터 템플릿 샘플

다음 어댑터 템플릿 샘플이 제공됩니다.

### ClearQuestBaseClearCaseByDate

1. ClearCase 보기에서 두 날짜 간의 변경사항을 조회합니다. 기본 날짜는 현재 시간소인 및 이전 어댑터 실행의 시간소인입니다.
2. 변경된 각 파일에 대해 ClearQuest 변경 ID를 식별하는 CrmRequest 하이퍼링크 속성을 검색합니다. ClearQuest 상태가 결함 레코드 분석을 허용하는 경우 ClearQuest에서 결함 레코드를 분석하기 위해 작업 정보를 추가하여 변경 ID 분석을 시도합니다.
3. 변경된 각 파일에 대해 BOM 보고서에 파일 이름, 결함 ID, 결함 상태 및 ClearQuest 오류 정보를 기록합니다.

어댑터 템플릿에 정의된 변수:

- CurDate
- LAST\_RUN
- VIEW
- VOB\_PATH
- CQ\_USER
- CQ\_PASSWORD
- BFSERVER
- UNIXCLIENT
- \_CHAR\_NATIVE

### ClearQuestClearCaseByActivity

1. ClearCase 활동 목록과 연관된 ClearQuest 결함 레코드를 찾습니다.
2. 발견된 각 결함 레코드에 대해 ClearQuest 상태가 결함 레코드 분석을 허용하는 경우 ClearQuest에서 결함 레코드를 분석하기 위해 작업 정보를 추가합니다.
3. BOM 보고서에 ClearCase 활동 ID와 연관된 파일 및 ClearQuest 결함 상태 정보를 기록합니다.

어댑터 템플릿에 정의된 변수:

- CurDate
- VIEW
- VOB\_PATH
- ACTIVITIES
- CQ\_USER

- CQ\_PASSWORD
- PROJECT\_VOB
- BFSERVER
- UNIXCLIENT
- \_CHAR\_NATIVE

### ClearQuestUCMClearCaseByDate

1. ClearCase 보기에서 두 날짜 간의 변경사항을 조회합니다. 기본 날짜는 현재 시간소인 및 이전 어댑터 실행의 시간소인입니다. Rational 통합 변경 관리(UCM)를 사용하여 결과를 채웁니다.
2. 변경된 각 파일에 대해 BOM 보고서에 파일 이름, 결함 ID, 결함 상태 및 ClearQuest 오류 정보를 기록합니다.

어댑터 템플릿에 정의된 변수:

- CurDate
- LAST\_RUN
- VIEW
- VOB\_PATH
- CQ\_USER
- CQ\_PASSWORD
- BFSERVER
- UNIXCLIENT
- \_CHAR\_NATIVE

### ClearQuest 어댑터 변수

이 표는 어댑터 템플릿에 대한 변수 목록의 참조입니다.

표 19. Rational ClearQuest 통합에 필요한 환경 변수

변수	설명
ACTIVITIES	ClearQuestClearCaseByActivity 어댑터의 경우, 공백으로 구분된 활동 ID 세트입니다. (예: SAMPL0001@\ProjectVob)
BFSERVER	이 변수를 Build Forge 콘솔에 대한 호스트의 이름으로 설정하십시오.
CQ_PASSWORD	필수. ClearQuest 데이터베이스에 로그인할 때 사용할 비밀번호. 필수는 아니며, 기본값은 공백으로 설정됩니다.
CQ_USER	필수. ClearQuest 데이터베이스에 로그인할 때 사용할 사용자 이름
CurDate	어댑터에 현재 날짜를 제공합니다. .date 명령을 사용하여 날짜를 생성합니다. 이 값을 변경하지 마십시오.

표 19. Rational ClearQuest 통합에 필요한 환경 변수 (계속)

변수	설명
LAST_RUN	ByDate 어댑터의 경우, 시스템은 이 값을 사용하여 변경사항이 발생했는지 여부를 판별합니다. 값은 마지막 성공적 실행 날짜입니다. 어댑터를 강제 실행하기 위해 어댑터를 테스트할 때 일부 변경사항에 선행하는 날짜를 선택하여 이 값을 조작할 수 있습니다. 실행을 계속하도록 허용하는 경우, 어댑터는 이 값을 현재 날짜로 자동으로 업데이트합니다. 기본값은 1-Jan-05.00:00:00입니다.
UNIXCLIENT	플랫폼 특정 정보를 설정하기 위해 사용됩니다. 클라이언트가 Windows에서 실행 중인 경우 0으로 설정하십시오. 클라이언트가 UNIX 또는 Linux에서 실행 중인 경우 1로 설정하십시오.
VIEW	이 변수를 어댑터와 함께 사용할 ClearCase 보기의 이름으로 설정하십시오.
VOB_PATH	이 값을 컴포넌트 VOB의 이름으로 설정하고 선택적으로 서브디렉토리를 설정하십시오. 이름이 여러 개인 경우 쉼표로 구분되는 목록을 사용하십시오.
_CHAR_NATIVE	내부적으로 사용되며 항상 1로 설정됩니다.

## ClearQuest 통합 작업 다시 시작

ClearQuest 통합이 완료되면 일반적으로 Rational Build Forge에서 다시 시작할 수 없습니다. 단순 임시 해결책으로는 새 작업으로 작업을 시작할 수 있습니다.

다시 시작을 사용하려면 ClearQuest 디자이너 도구를 사용하여 ClearQuest 스키마를 편집해야 합니다. 완료에서 제출로 상태 전이를 허용하려면 빌드 레코드의 워크플로우를 수정해야 합니다.



---

## 제 29 장 Rational Build Forge 및 CVS

BuildForge는 소스 코드 관리를 위해 CVS와 함께 작동할 수 있습니다.

이 주제에서는 CVS와 통합하기 위한 추가 설정 요구사항을 설명하고 제공된 샘플 어댑터 템플릿의 특성을 요약합니다.

CVS 어댑터 템플릿 샘플은 기준선에 대한 변경사항을 분석하는 방법을 제공합니다. 일반적으로 변경 분석은 빌드 방지를 위해 사용됩니다. 기준선 컴포넌트가 변경되지 않은 경우 다시 빌드되지 않습니다.

### CVS 어댑터에 대한 추가 설정 요구사항

일반 요구사항은 529 페이지의 『어댑터 요구사항』의 내용을 참조하십시오. CVS에 대한 추가 요구사항은 없습니다.

### CVS 어댑터 템플릿 샘플

다음 어댑터 템플릿 샘플이 제공됩니다.

#### CVSv1Baseline

1. Build Forge 에이전트에서 CVS 디렉토리를 스캔하여 변경된 파일을 찾습니다.
2. BOM 보고서에 변경된 파일 이름, 상태, 작업 버전, 저장소 버전 및 sticky 태그 정보를 기록합니다. ClearCase 보기에서 디렉토리를 스캔합니다.

사용되는 변수:

- CVSROOT
- MODULE
- CVSCLIENT

#### CVSv1ByDate

1. CVS 보기에서 두 날짜 간의 변경사항을 조회합니다. 기본 날짜는 현재 시간소인 및 이전 어댑터 실행의 시간소인입니다.
2. BOM 보고서에 변경 유형, 날짜, 사용자 이름, 버전 및 파일 이름 정보를 기록합니다.
3. 변경된 각 파일에 대해 diff 명령 출력의 변경 세부사항을 BOM 보고서에 기록합니다.

사용되는 변수:

- LAST\_RUN

- CurDate
- CVSROOT
- MODULE
- BRANCH
- CVSCLIENT

#### **CVSv1ByTag**

1. CVS 모듈에 새 태그를 적용합니다.
2. 새로 태그 지정된 모듈과 이전 어댑터 실행 중에 태그 지정된 모듈 간의 차이점을 비교합니다.
3. BOM 보고서에 파일 이름, 개정, 상태, 날짜, 시간, 변경 작성자 및 커밋 주석 정보를 기록합니다.
4. 변경된 각 파일에 대해 diff 명령 출력의 변경 세부사항을 BOM 보고서에 기록합니다.

사용되는 변수:

- LAST\_RUN
- CurDate
- CVSROOT
- MODULE
- BRANCH
- LAST\_TAG
- CVSCLIENT

#### **CVSv2ByDate**

1. CVS 보기에서 두 날짜 간의 변경사항을 조회합니다. 기본 날짜는 현재 시간소인 및 이전 어댑터 실행의 시간소인입니다.
2. BOM 보고서에 변경 유형, 날짜, 사용자 이름, 버전 및 파일 이름 정보를 기록합니다.
3. 변경된 각 파일에 대해 diff 명령 출력의 변경 세부사항을 BOM 보고서에 기록합니다.

사용되는 변수:

- LAST\_RUN
- CurDate
- CVSROOT
- MODULE
- CVSCLIENT

---

## 제 30 장 IDE 통합

플러그인을 사용하면 IDE(Integrated Development Environment)와 Rational Build Forge를 통합할 수 있습니다.

이 절에서는 IDE(Integrated Development Environment)에서 관리 콘솔 기능에 액세스할 수 있는 플러그인을 설치 및 사용하는 방법에 대해 설명합니다.

---

### IDE 통합 정보

개발자는 IDE(Integrated Development Environment) 플러그인을 사용하여 관리 콘솔에 연결할 수 있습니다.

제공된 플러그인을 사용하여 개발자는 해당 IDE에서 다음 태스크를 직접 완료할 수 있습니다.

- 프로젝트 보기
- 작업 실행
- 작업 결과 조사

기타 기능은 플러그인마다 다릅니다.

각 개발자는 Build Forge가 해당 사용자 목록에서 인식하는 사용자 이름으로 콘솔에 연결합니다. 프로젝트에 대한 액세스는 액세스 그룹의 사용자 이름 멤버십으로 제어됩니다. 프로젝트 내 단계에 대한 액세스 또한 액세스 그룹에서 제어합니다. 단계는 액세스 그룹을 명시적으로 지정할 수 있습니다. 명시적으로 지정하지 않으면 프로젝트의 액세스 그룹을 상속합니다.

플러그인은 프로젝트 및 단계를 편집 또는 삭제하는 기능을 제공하지 않습니다.

IDE를 통해 Build Forge에 액세스하는 각 사용자는 사용자가 브라우저 클라이언트 세션을 통해 라이선스를 이용하는 것처럼 라이선스를 이용합니다.

플러그인이 제공되는 IDE는 다음과 같습니다.

- Eclipse<sup>™</sup>
- Rational<sup>®</sup> Application Developer(Eclipse<sup>™</sup> 기반 IBM IDE)
- Rational<sup>®</sup> Team Concert(IBM이 배포한 Jazz.net)

---

## 테스트 프로젝트의 특수 변수

플러그인을 사용하여 프로젝트의 테스트 빌드를 실행할 때 몇몇 특수 환경 변수를 사용하여 시스템에서 서버로 파일을 복사하기 전후에 실행할 명령을 지정할 수 있습니다.

모든 명령은 프로젝트 디렉토리에서 실행됩니다.

- **PRECMD** 변수를 사용하여 개발자의 컴퓨터에서 빌드를 실행 중인 서버로 복사되는 디렉토리와 파일에서 명령을 실행하십시오. 명령은 프로젝트 단계 전에 실행됩니다. 예제: 이 명령을 사용하면 파일이 복사되기 전에 소스 제어 시스템에서 파일을 체크아웃할 수 있습니다.
- **POSTCMD** 변수를 사용하여 프로젝트 단계가 실행된 후에 디렉토리와 파일에서 명령을 실행하십시오. 예제: 이 명령을 사용하여 Rational ClearCase 같은 개념을 사용하는 소스 제어 시스템에서 체크아웃된 가상 디렉토리를 비울 수 있습니다.

명령은 리플렉터 플러그인에 Build Forge 프로젝트 아티팩트로 표시된 디렉토리와 파일에서 실행됩니다. 리플렉터 플러그인의 디렉토리 트리를 순회할 때 명령이 적용됩니다.

**참고:** 디렉토리 트리 순회는 **PRECMD** 명령의 경우에는 너비 우선 하향식이고 **POSTCMD** 명령의 경우에는 반대입니다. 디렉토리 명령과 파일 명령은 순회 중에 적절히 실행됩니다.

### **\_PRISM\_DIR\_PRECMD**

트리 순회 중에 디렉토리가 발견되면 디렉토리에서 실행될 명령을 지정합니다. 명령은 최소 하나의 파일을 포함하는 모든 디렉토리에 대해 한 번 실행됩니다. 시스템은 명령에서 첫 번째 \$1을 디렉토리 이름으로 바꿉니다.

### **\_PRISM\_FILE\_PRECMD**

트리 순회 중에 파일이 발견되면 파일에서 실행될 명령을 지정합니다. 명령은 모든 파일에 대해 한 번 실행됩니다. 시스템은 명령에서 첫 번째 \$1을 파일 이름으로 바꿉니다.

### **\_PRISM\_DIR\_POSTCMD**

트리 순회 중에 디렉토리가 발견되면 디렉토리에서 실행될 명령을 지정합니다. 명령은 최소 하나의 파일을 포함하는 모든 디렉토리에 대해 한 번 실행됩니다. 시스템은 명령에서 첫 번째 \$1을 디렉토리 이름으로 바꿉니다.

### **\_PRISM\_FILE\_POSTCMD**

트리 순회 중에 파일이 발견되면 파일에서 실행될 명령을 지정합니다. 명령은 모든 파일에 대해 한 번 실행됩니다. 시스템은 명령에서 첫 번째 \$1을 파일 이름으로 바꿉니다.

---

## Eclipse 및 Rational Application Developer의 플러그인

플러그인은 Eclipse™ 및 Rational® Application Developer IDE에서 관리 콘솔 기능에 대한 액세스를 제공합니다.

Eclipse 및 Rational Application Developer 환경에 사용 가능한 플러그인은 다음과 같습니다.

### 빈도

빈도 플러그인을 사용하여 개발자는 다음 태스크를 완료할 수 있습니다.

- 하나 이상의 관리 콘솔에 액세스하여 프로젝트 보기
- 작업 실행
- 작업 상태 보기
- 실행 중이거나 완료된 작업의 빌드 로그 보기

### 리플렉터

리플렉터 플러그인은 로컬 환경에서 파일을 사용하여 작업을 실행합니다. 기타 개발자나 프로덕션 빌드에서 사용할 수 있도록 소스 제어를 확인하기 전에 새 코드를 테스트하기 위해 일반적으로 이 작업을 실행합니다.

Eclipse 플러그인 사용자에게는 프로젝트 환경 변수 값을 대체할 수 있는 옵션이 있습니다. Build Forge 프로젝트를 시작하면 작업 설정 팝업이 표시됩니다. 환경 변수 변경 사항은 작업에만 적용됩니다. 프로젝트의 기본 변수 값은 변경되지 않습니다.

### Eclipse 또는 Rational Application Developer에서 플러그인 사용

플러그인을 설치한 후에는 다음과 같은 방법으로 플러그인을 활성화할 수 있습니다.

- 관리 콘솔에 액세스하여 작업을 실행하고 프로젝트 로그를 보려면 빈도 플러그인을 사용하십시오. 사용자의 IDE에서, 창 > 퍼스펙티브 열기 > 기타를 선택하십시오. 사용자의 IDE가 퍼스펙티브 유형 목록이 있는 대화 상자를 표시합니다. Build Forge 퍼스펙티브를 선택하십시오. 시스템이 콘솔 탐색기, 빌드 정보 및 빌드 로그 창을 표시합니다. 콘솔 탐색기를 마우스 오른쪽 단추로 클릭하고 새 콘솔을 선택하여 관리 콘솔과의 연결을 구성하십시오. 빈도 플러그인 사용에 관한 자세한 정보는 플러그인과 함께 제공되는 온라인 도움말을 참조하십시오.

**참고:** LDAP/Active Directory 도메인으로의 액세스를 구성해야 하는 경우, LDAP 도메인 오브젝트에 도메인의 실제 이름이 아닌 Build Forge 시스템 이름을 사용하십시오.

- 테스트 빌드를 실행하려면 리플렉터 플러그인을 사용하십시오. 사용자의 IDE에서, 프로젝트를 선택하고 마우스 오른쪽 단추를 클릭하여 리플렉터를 구성하십시오. 팝업 메뉴에서 특성을 선택하십시오. 특성 대화 상자의 특성 옵션 목록에서 **Build Forge 프로젝트 아티팩트**를 선택하십시오. 프로젝트와 작동할 Build Forge 프로젝트로 대화

상자를 구성하고 시스템에 업로드될 파일을 선택하십시오. 자세한 정보는 플러그인과 함께 제공되는 온라인 도움말을 참조하십시오.

참고: 리플렉터 플러그인에 빈도(Frequency) 플러그인이 필요합니다.

## Eclipse 또는 Rational Application Developer 플러그인 설치

Build Forge 서버에서 IDE 환경에 대한 플러그인을 설치하십시오.

### 시작하기 전에

전제조건:

- Eclipse 버전 3.0.2 이상 또는 Rational Application Developer 버전 7.0 이상
- Java 2 SE 버전 5.0
- Build Forge 시스템이 실행 중이어야 함

### 이 태스크 정보

플러그인을 설치하려면 IDE에서 이 프로시저를 완료하십시오.

### 프로시저

1. 도움말 > 소프트웨어 업데이트 > 찾기 및 설치를 선택하십시오.
2. 설치할 새 기능 검색 단일 선택 단추를 클릭한 다음 다음을 클릭하십시오. 방문할 업데이트 사이트 대화 상자가 표시됩니다.
3. 새 원격 사이트 단추를 클릭하십시오. 새 원격 사이트 대화 상자가 표시됩니다.
  - a. 이름 필드에 『Build Forge 업데이트 사이트』를 입력하십시오.
  - b. URL 필드에 관리 콘솔 컴퓨터의 호스트 이름을 사용하여 다음 업데이트 사이트 URL을 입력하십시오. `http://<console_host_name>/prism/eclipse/updateSite/site.xml`.
    - Build Forge를 실행 중인 시스템과 동일한 시스템에서 Eclipse를 실행 중인 경우 localhost를 호스트 이름으로 사용할 수 있습니다.
    - 콘솔이 80이 아닌 포트에서 실행되는 경우 포트 번호를 포함합니다예: `http://myhostname:11812/prism/eclipse/updateSite/site.xml`
  - c. 확인을 클릭하십시오.
4. 방문할 업데이트 사이트 대화 상자에서 **Build Forge** 업데이트 사이트 선택란을 선택한 다음 완료를 클릭하십시오.
5. 검색 결과 대화 상자에 사용 가능한 플러그인의 목록이 시스템에 표시됩니다. 제공된 플러그인을 모두 선택한 후 다음을 클릭하십시오.

참고: 리플렉터 플러그인에 빈도(Frequency) 플러그인이 필요합니다. 단독으로 설치된 경우 실행되지 않습니다.

6. 라이선스 계약을 읽고 라이선스 계약 내용에 동의합니다를 선택한 후 다음을 클릭하십시오.
7. 기능을 설치할 위치를 선택하십시오. 새 위치를 추가하려면, 새 위치를 클릭한 다음 원하는 위치를 찾아보십시오.
8. 완료를 클릭하십시오.
9. 기능 검증 대화 상자가 표시되면 설치를 클릭하십시오. 플러그인은 서명되지 않은 기능이므로 대화 상자가 나타납니다. 모두 설치를 선택하지 않으면, 각 기능이 설치될 때마다 대화 상자가 표시됩니다.
10. 변경사항이 적용되도록 하려면 Eclipse를 다시 시작해야 합니다. 예를 클릭하십시오.

## SSL 사용 시 대체 설치

Build Forge 시스템에서 SSL을 사용하는 경우 대체 설치 방법을 사용합니다.

### 시작하기 전에

현재 Eclipse 및 Rational Application Developer 버전은 SSL에 사용할 수 없습니다. 따라서 Build Forge 시스템에서 SSL을 사용하는 경우 Eclipse 및 Rational Application Developer 플러그인을 Build Forge 시스템에서 설치할 수 없습니다. 해결 방법은 비보안 웹 서버에서 플러그인 설치 파일을 사용 가능하게 하거나 수동으로 사용자에게 분배하는 것입니다.

파일을 패키지로 만들려면 다음을 수행하십시오.

1. prism이라는 디렉토리를 작성하십시오.
2. prism에서 eclipse라는 디렉토리를 작성하십시오.
3. eclipse에서 updateSite라는 디렉토리를 작성하십시오.

<bfinstall>/webroot/public/prism/eclipse/updateSite에서 다음 파일을 사용자가 작성한 updateSite 디렉토리에 복사하십시오.

- features 디렉토리
- plugins 디렉토리
- site.xml 파일

### 이 태스크 정보

prism 디렉토리를 사용 가능하게 한 경우 사용자는 해당 IDE에서 다음 단계를 수행합니다.

### 프로시저

1. 도움말 > 소프트웨어 업데이트 > 찾기 및 설치를 선택하십시오.

2. 설치할 새 기능 검색 단일 선택 단추를 클릭한 다음 다음을 클릭하십시오. 방문할 업데이트 사이트 대화 상자가 표시됩니다.
3. 새 사이트를 작성하십시오. 다음 프로시저 중 하나를 선택하십시오.
  - 원격 서버에서 파일 가져오기:
    - a. 새 원격 사이트 단추를 클릭하십시오. 새 원격 사이트 대화 상자가 표시됩니다.
    - b. 이름 필드에 『Build Forge 업데이트 사이트』를 입력하십시오.
    - c. 파일의 위치를 입력하십시오.  
`http://host/path/prism/eclipse/updateSite/site.xml`  
*host*는 웹 서버의 IP 주소 또는 호스트 이름입니다.  
*path*는 서버 루트에서 파일 저장 위치까지의 경로입니다.
    - d. 확인을 클릭하십시오.
  - 로컬 호스트에서 파일 가져오기:
    - a. 새 로컬 사이트 단추를 클릭하십시오. 시스템에 새 로컬 사이트 대화 상자가 표시됩니다.
    - b. 이름 필드에 『Build Forge 업데이트 사이트』를 입력하십시오.
    - c. 파일의 위치를 입력하십시오.  
`file://path/prism/eclipse/updateSite/site.xml`  
*path*는 파일의 위치를 지정합니다.
    - d. 확인을 클릭하십시오.
4. 방문할 업데이트 사이트 대화 상자에서 **Build Forge** 업데이트 사이트 선택란을 선택한 다음 완료를 클릭하십시오.
5. 검색 결과 대화 상자에 사용 가능한 플러그인의 목록이 시스템에 표시됩니다. 제공된 플러그인을 모두 선택한 후 다음을 클릭하십시오.  
 참고: 리플렉터 플러그인에 빈도(Frequency) 플러그인이 필요합니다. 단독으로 설치된 경우 실행되지 않습니다.
6. 라이선스 계약을 읽고 라이선스 계약 내용에 동의합니다를 선택한 후 다음을 클릭하십시오.
7. 기능을 설치할 위치를 선택하십시오. 새 위치를 추가하려면, 새 위치를 클릭한 다음 원하는 위치를 찾아보십시오.
8. 완료를 클릭하십시오.
9. 기능 검증 대화 상자가 표시되면 설치를 클릭하십시오. 플러그인은 서명되지 않은 기능이므로 대화 상자가 나타납니다. 모두 설치를 선택하지 않으면, 각 기능이 설치될 때마다 대화 상자가 표시됩니다.



10. 변경사항이 적용되도록 하려면 Eclipse를 다시 시작해야 합니다. 예를 클릭하십시오.

## Eclipse 및 Rational Application Developer의 플러그인 사용

실행 작업에 액세스하여 프로젝트 로그(빈도 플러그인)를 보려면 다음을 수행하십시오.

1. 사용자의 IDE에서, 창 > 퍼스펙티브 열기 > 기타를 선택하십시오.
2. **Build Forge** 퍼스펙티브를 선택하십시오. 퍼스펙티브에는 다음 창이 포함됩니다.
  - 콘솔 탐색기
  - 빌드 정보
  - 빌드 로그
3. 콘솔 탐색기를 마우스 오른쪽 단추로 클릭하고 새 콘솔을 선택하여 관리 콘솔과의 연결을 구성하십시오.
4. **Build Forge** 서비스 계층 호스트 이름에 호스트 이름 또는 IP 주소를 입력하고 다른 필드를 검증 또는 편집한 후 확인을 클릭하십시오.

연결이 설정되면 빌드 정보 창이 사용자가 실행할 수 있는 작업으로 채워집니다. 로컬 파일을 사용하여 작업을 실행하려면 사용할 작업과 사용할 파일을 구성하십시오.

1. 사용자 IDE에서 Build Forge에 연결하십시오.
2. 콘솔 탐색기 창에서 작업을 마우스 오른쪽 단추로 클릭한 다음 특성을 선택하십시오.
3. 특성에서 **Build Forge** 프로젝트 아티팩트를 선택하십시오.
4. **Build Forge** 프로젝트 아티팩트에서 작업을 수행할 프로젝트를 선택하고 사용할 로컬 파일을 선택하십시오.

자세한 정보는 플러그인과 함께 제공되는 온라인 도움말을 참조하십시오.

---

## Rational Team Concert 플러그인

Rational Team Concert 클라이언트의 플러그인은 Rational Team Concert와 Build Forge의 통합 컴포넌트 중 하나입니다. 서버 확장과 어댑터 템플릿 또한 필요합니다. Rational Team Concert와의 통합이 설정되면 Rational Team Concert의 사용자가 다음 태스크를 완료할 수 있습니다.

- Build Forge를 RTC 빌드 서버로 설정
- Build Forge 프로젝트를 RTC 빌드 정의로 설정
- 프로젝트 보기, 작업 실행 및 RTC 클라이언트에서 작업 결과 보기

통합 설정에 대한 지시사항은 601 페이지의 제 33 장 『Rational Build Forge 및 Rational Team Concert』을 참조하십시오.

## Rational Team Concert 플러그인 사용

이 지시사항은 Rational Team Concert 통합이 설정되고 Rational Team Concert 클라이언트에 플러그인이 설치된 것으로 가정합니다.

통합 설정에 대한 지시사항은 601 페이지의 제 33 장 『Rational Build Forge 및 Rational Team Concert』을 참조하십시오.

빌드 정의를 설정하고 빌드를 실행하려면 다음을 수행하십시오.

1. 빌드 정의를 설정하십시오.
  - a. 팀 아티팩트 보기에서 프로젝트 폴더를 펼치십시오.
  - b. 빌드를 마우스 오른쪽 단추로 클릭한 다음 새 빌드 정의를 클릭하십시오.
  - c. 새 빌드 정의에서 새 빌드 작성을 선택한 후 다음을 클릭하십시오.
  - d. 일반 정보에 빌드 ID와 설명을 입력하십시오. 사용 가능한 템플릿 목록에서 **Rational Build Forge**를 선택하십시오. 다음을 클릭하십시오.
  - e. 추가 구성에서 일반과 특성을 모두 선택한 다음 완료를 클릭하십시오. 입력한 빌드 ID 레이블이 지정된 탭이 작성됩니다.
  - f. **Build Forge** 탭을 클릭하십시오.
2. 빌드 정의에 대한 프로젝트를 선택하십시오.
  - a. Build Forge 탭을 클릭하십시오. Build Forge에 연결하는 데 필요한 다음 정보를 입력하십시오.
    - 호스트 이름 - Build Forge가 실행되는 호스트의 호스트 이름. 이는 설정된 경우 콘솔 URL 시스템 설정과 일치해야 합니다. 관리 > 시스템에서 해당 설정에 액세스할 수 없는 경우 관리자에게 문의하십시오.
    - 포트 - Build Forge와 통신하는 데 사용되는 포트. 기본값은 포트 3966입니다. **Build Forge**에 대한 보안 연결이 선택되면 기본적으로 포트 49150이 표시됩니다. 설치 포트 번호가 다른 경우 해당 포트 번호를 입력하십시오. 이는 설정된 경우 콘솔 URL 시스템 설정에서 지정된 포트와 일치해야 합니다. 관리 > 시스템에서 해당 설정에 액세스할 수 없는 경우 관리자에게 문의하십시오.
    - 사용자 이름 - Build Forge에 연결하기 위한 사용자 이름. 사용자는 Build Forge에 이미 있어야 합니다.
    - 비밀번호 - 사용자 이름의 비밀번호
    - 비밀번호 확인 - 사용자 이름의 비밀번호
  - b. 프로젝트 가져오기를 클릭하십시오.
  - c. **Build Forge** 프로젝트 목록이 나타나면 빌드 정의에 대한 프로젝트를 선택하십시오.
3. 빌드를 요청하십시오.

- a. 빌드 정의를 마우스 오른쪽 단추로 클릭한 다음 빌드 요청을 선택하십시오.
  - b. 원하는 빌드 옵션과 빌드 특성을 지정한 다음 제출을 클릭하십시오.
4. 빌드 결과를 확인하십시오.
- a. 빌드가 완료된 후 빌드 탭의 목록에서 선택하십시오. 창이 표시됩니다.
  - b. 외부 링크 아래에서 **Build Forge** 결과 링크를 클릭하십시오.
  - c. Build Forge 로그인 패널이 표시됩니다. 로그인하십시오.
  - d. 홈 > 완료된 실행으로 이동한 다음 목록에서 빌드를 선택하십시오.
  - e. 빌드 단계와 결과가 표시됩니다. 단계 링크를 클릭하면 단계 로그가 표시됩니다.
  - f. 완료되면 로그아웃을 클릭하고 창을 닫으십시오.

## Rational Team Concert 플러그인 문제점 해결

이 절에서는 알려진 문제점과 해결 방법을 설명합니다.

### 기한이 지남 상태의 작업

작업이 몇 분 이상 기한이 지남 상태에 있으면 정지될 수도 있습니다.

#### 해결 방법

Build Forge 엔진을 중지하고 다시 시작하십시오. 286 페이지의 『엔진 시작 및 중지』의 내용을 참조하십시오.



---

## 제 31 장 Rational Build Forge 및 Perforce

BuildForge는 소스 코드 관리를 위해 Perforce와 함께 작동할 수 있습니다.

이 주제에서는 Perforce와 통합하기 위한 추가 설정 요구사항을 설명하고 제공된 샘플 어댑터 템플릿의 특성을 요약합니다.

Perforce 어댑터 템플릿 샘플은 기준선에 대한 변경사항을 분석하는 방법을 제공합니다. 일반적으로 변경 분석은 빌드 방지를 위해 사용됩니다. 기준선 컴포넌트가 변경되지 않은 경우 다시 빌드되지 않습니다.

### Perforce 어댑터에 대한 추가 설정 요구사항

일반 요구사항은 529 페이지의 『어댑터 요구사항』의 내용을 참조하십시오.

1. 에이전트와 동일한 호스트에서 P4 명령행 클라이언트를 설치하십시오.
2. 사용할 Build Forge에 대한 Perforce 사용자 계정을 설정하십시오.
3. 클라이언트에 액세스할 수 있도록 에이전트 환경을 설정하십시오.

### Perforce 어댑터 템플릿 샘플

다음 어댑터 템플릿 샘플이 제공됩니다.

#### PerforceByDate

1. Perforce 클라이언트에서 어댑터 실행 이후로 발생한 변경사항을 조회합니다.
2. BOM 보고서에 변경, 날짜, 시간, 사용자, Perforce 클라이언트 및 주식 정보를 기록합니다.
3. diff 명령 출력의 변경 세부사항을 BOM 보고서에 기록합니다.

어댑터에 설정된 변수:

- LAST\_RUN
- CurDate
- P4PORT
- BFCLIENT
- P4CLIENT
- FILESPEC

환경에 대한 추가 필수 변수:

- P4USER

- P4PASSWD

이 변수에는 숨겨짐 지정 특성을 사용할 수 없습니다. 사용자 및 비밀번호는 단계 로그에 일반 텍스트로 기록됩니다.

### **PerforceByRev**

1. Perforce 클라이언트에서 마지막 저장소 개정 이후로 발생한 변경사항을 조회합니다.
2. BOM 보고서에 변경, 날짜, 시간, 사용자, Perforce 클라이언트 및 주식 정보를 기록합니다.
3. diff 명령 출력의 변경 세부사항을 BOM 보고서에 기록합니다.

어댑터에 설정된 변수:

- LAST\_RUN
- CurDate
- P4PORT
- BFCLIENT
- P4CLIENT
- FILESPEC

환경에 대한 추가 필수 변수:

- P4USER
- P4PASSWD

이 변수에는 숨겨짐 지정 특성을 사용할 수 없습니다. 사용자 및 비밀번호는 단계 로그에 일반 텍스트로 기록됩니다.

---

## 제 32 장 Rational Build Forge 및 Rational Quality Manager

Rational Build Forge는 Rational Quality Manager와 함께 작동할 수 있습니다.

Rational® Build Forge®를 Rational Quality Manager와 통합하여 Rational Quality Manager에서 소프트웨어를 빌드 및 패키징하는 프로젝트를 정의하고 실행할 수 있습니다.

통합은 Rational Quality Manager에서 완료됩니다. Rational Quality Manager의 버전은 Information Center를 참조하십시오. 링크는 Rational Quality Manager 정보 페이지를 참조하십시오.

- Rational Quality Manager 버전 2는 통합 > **Rational Quality Manager** 및 소프트웨어 인벤토리 및 자동화 도구 > **Rational Build Forge**와 통합을 참조하십시오.
- Rational Quality Manager 버전 1은 통합 > 소프트웨어 인벤토리 및 프로비저닝 도구와 통합 > **Rational Build Forge**와 통합을 참조하십시오.





---

## 제 33 장 Rational Build Forge 및 Rational Team Concert

Build Forge를 Rational Team Concert와 통합할 수 있습니다.

통합이 설정되면 Rational Team Concert의 사용자가 다음을 수행할 수 있습니다.

- Build Forge를 RTC 빌드 서버로 설정
- 프로젝트 보기, 작업 실행 및 RTC 클라이언트에서 작업 결과 보기
- Build Forge 프로젝트를 RTC 빌드 정의로 설정

---

### Rational Team Concert 버전 3.x

통합 컴포넌트는 Rational Team Concert 설치와 함께 제공됩니다.

Rational Team Concert 3.x와 Rational Build Forge 버전 7.1.1.3 이상을 통합하는 것은 이전 버전의 Rational Team Concert를 통합할 때에 비하면 분명한 장점이 있습니다. Rational Team Concert 3.x는 Rational Build Forge와의 통합에 우수한 플랫폼입니다.

Rational Build Forge 및 Rational Team Concert를 설치한 후에 Rational Team Concert에서 통합을 구성하십시오. 일반적으로 두 제품은 다른 호스트에 설치됩니다.

통합을 구성하고 문제점을 해결하기 위한 지시사항은 [jazz.net](http://jazz.net)의 Rational Build Forge 위키를 참조하십시오.

---

### Rational Team Concert 버전 1.x 및 2.x

Rational Team Concert 버전 1.x 및 2.x와 통합하려면 Rational Build Forge와 함께 제공된 컴포넌트를 설치해야 합니다.

통합하려면 다음 작업을 수행해야 합니다.

- Rational Build Forge에서 컴포넌트 설치
- Rational Build Forge와 통신하도록 Rational Team Concert 구성

컴포넌트 설치

다음 컴포넌트를 설치합니다.

- Rational Team Concert 서버의 서버 확장: 이 컴포넌트는 Rational Team Concert와 Build Forge 간의 통신을 수행할 수 있도록 합니다. 서버 확장은 RTC 서버에 설치

되어 있습니다. 설치 지시사항은 『Rational Team Concert 서버 확장 설치』에 있습니다. Build Forge 사용자를 Rational Team Concert 서버에 할당하십시오.

- 클라이언트 플러그인: 이 컴포넌트는 Rational Team Concert 클라이언트의 플러그인입니다. 이 컴포넌트를 사용하면 Build Forge에 액세스하여 프로젝트를 보고 작업을 실행하며 작업 결과를 볼 수 있습니다. Build Forge에 액세스할 각 RTC 클라이언트에 클라이언트 플러그인이 설치되어 있어야 합니다. 설치 지시사항은 603 페이지의 『Rational Team Concert의 클라이언트 플러그인 설치』에 있습니다.
- 어댑터: 이 컴포넌트는 프로젝트에서 Rational Team Concert 저장소에 있는 파일에 액세스할 수 있도록 하는 소스 유형 어댑터입니다. RTC 버전 1.x와 RTC 버전 2.x에 서로 다른 어댑터가 사용됩니다. 607 페이지의 『Rational Team Concert 어댑터 구성』의 내용을 참조하십시오.

### Rational Team Concert 구성

통합을 구성하고 문제점을 해결하기 위한 지시사항은 jazz.net의 Rational Build Forge 위키를 참조하십시오.

## Rational Team Concert 서버 확장 설치

### 시작하기 전에

전제조건:

- Rational Team Concert가 버전 1.x 또는 2.x여야 합니다. 프로시저의 버전별로 지시사항 및 파일이 다른 경우에 유의하십시오.
- Build Forge 시스템이 실행 중이어야 합니다.

### 이 태스크 정보

Rational Team Concert 서버 호스트에서 다음 단계를 수행하십시오.

### 프로시저

1. Rational Team Concert 서버 호스트에 로그인하십시오.
2. Build Forge 서버를 탐색하고 Rational Team Concert 버전에 적절한 URL을 사용하여 서버 확장기능을 임시 위치에 다운로드하십시오.
  - Rational Team Concert 버전 1.x:  
`http://<bf_console_hostname>/rtc-server/BuildForgeConnectorServer.zip`
  - Rational Team Concert 버전 2.x:  
`http://<bf_console_hostname>/rtc2-server/BuildForgeConnectorServer.zip`
  - Build Forge가 실행 중인 동일한 시스템에서 Rational Team Concert를 실행하는 경우 localhost를 호스트 이름으로 사용할 수 있습니다.

- 콘솔이 80이 아닌 포트에서 실행되는 경우 포트 번호를 포함합니다(예: `http://myhostname:11812/clients`).

3. 파일 콘텐츠를 `RTC_install/jazz/server`에 추출하십시오. Rational Team Concert 버전 1.x의 확장을 다운로드한 경우 다음 파일이 추가됩니다.

```
RTC_install/jazz/server/buildforgeconnector-update-site(디렉토리)
RTC_install/jazz/server/provision_profiles/buildforgeconnector-profile.ini
```

Rational Team Concert 버전 2.x의 확장을 다운로드한 경우 다음 파일이 추가됩니다.

```
RTC_install/jazz/server/buildforgeconnector-update-site(디렉토리)
RTC_install/jazz/server/conf/provision_profiles/buildforgeconnector-profile.ini
```

WebSphere Application Server에서 RTC 서버를 실행하려면 `buildforgeconnector-update-site` 디렉토리에 대한 경로를 완전히 나타내도록 `buildforgeconnector-profile.ini` 파일을 편집해야 합니다.

4. Rational Team Concert 서버를 다시 시작하십시오.

## 결과

Rational Team Concert 서버는 Build Forge 사용자 라이선스를 이용하여 콘솔과 통신합니다.

## Rational Team Concert의 클라이언트 플러그인 설치

Build Forge 서버에서 Rational Team Concert의 클라이언트 플러그인을 설치합니다.

### 시작하기 전에

전제조건:

- Rational Team Concert 버전 1.x 또는 2.x - 각 버전에 서로 다른 플러그인이 필요합니다.
- Build Forge Connector 서버 확장이 Rational Team Concert 서버에 설치되어 있어야 합니다.
- Build Forge 시스템이 실행 중이어야 합니다.
- 각 사용자에게 콘솔에서 사용할 두 개의 로그인 ID가 있어야 합니다. Build Forge에서는 사용자를 하나의 로그인 세션으로 제한합니다. 관리 콘솔 및 RTC 클라이언트를 통해 동일한 사용자 이름을 동시에 로그인할 수 없습니다. 새 세션이 시작되면 기존 세션이 종료됩니다. RTC 클라이언트 사용자에게 다음과 같은 두 가지 ID가 있어야 합니다.
  - UserRTC - 빌드를 실행하기 위해 빌드 정의를 구성하는 경우 사용됩니다. 이 사용자 이름은 빌드를 실행하기 위해 Build Forge 콘솔에 액세스하는 데 사용됩니다.

- UserBF - 빌드 결과를 확인하는 경우 사용됩니다. RTC 클라이언트의 창에서 Build Forge 관리 콘솔에 직접 로그인해야 합니다.

## 이 태스크 정보

Build Forge에 액세스할 모든 Rational Team Concert 클라이언트에 클라이언트 플러그인이 설치되어 있어야 합니다.

플러그인을 설치하려면 Rational Team Concert 클라이언트에서 다음 단계를 수행하십시오.

## 프로시저

1. 도움말 > 소프트웨어 업데이트 > 찾기 및 설치를 선택하십시오.
2. 설치할 새 기능 검색을 선택한 후 다음을 클릭하십시오. 방문할 업데이트 사이트 대화 상자가 표시됩니다.
3. 새 원격 사이트를 클릭하십시오. 새 원격 사이트 대화 상자가 표시됩니다.
  - a. 이름 필드에 Build Forge 커넥터 업데이트 사이트를 입력하십시오.
  - b. **URL** 필드에 업데이트 사이트 URL을 입력하십시오. URL은 사용 중인 Rational Team Concert의 버전에 따라 다릅니다.  
  
 Rational Team Concert 버전 1.x:  
`http://<console_host_name>/rtc/site.xml`  
  
 Rational Team Concert 버전 2.x:  
`http://<console_host_name>/rtc2/site.xml`
    - Build Forge가 실행 중인 동일한 시스템에서 Rational Team Concert를 실행하는 경우 localhost를 호스트 이름으로 사용할 수 있습니다.
    - 콘솔이 80이 아닌 포트에서 실행되는 경우 포트 번호를 포함합니다. 다음 예제는 Rational Team Concert 버전 1.x를 사용 중이고 Build Forge가 11812 포트에서 실행 중인 경우에 해당됩니다. `http://myhostname:11812/rtc/site.xml`
  - c. 확인을 클릭하십시오.
4. 방문할 업데이트 사이트 대화 상자에서 **Build Forge** 커넥터 업데이트 사이트 섹션을 선택한 다음 완료를 클릭하십시오.
5. 검색 결과 대화 상자에 업데이트 사이트에서 사용 가능한 기능의 목록이 표시됩니다. 제공된 기능을 모두 선택한 후 다음을 클릭하십시오.
6. 라이선스 계약을 읽고 라이선스 계약 내용에 동의합니다를 선택한 후 다음을 클릭하십시오.
7. 기능을 설치할 위치를 선택하십시오. 새 위치를 추가하려면 위치 변경을 클릭한 다음 원하는 위치를 찾으십시오.
8. 완료를 클릭하십시오.

9. 기능 검증 대화 상자가 표시되면 설치를 클릭하십시오. 플러그인은 서명되지 않은 기능이므로 대화 상자가 나타납니다. 모두 설치를 선택하지 않으면, 각 기능이 설치될 때마다 대화 상자가 표시됩니다.
10. 변경사항을 적용하기 위해 Rational Team Concert를 다시 시작할 것인지 묻는 메시지가 표시됩니다. 예를 클릭하십시오.
11. Build Forge를 다시 시작하십시오.

## 결과

통합이 완료되면 플러그인을 사용하여 작업을 실행하고 작업 결과를 점검할 수 있습니다. 빌드에 사용할 빌드 엔진으로 RationalBuildForgeConnector를 지정해야 합니다.

## SSL 사용 시 대체 설치

Build Forge 시스템에서 SSL을 사용하는 경우 대체 설치 방법을 사용합니다.

## 시작하기 전에

현재 버전의 Rational Team Concert는 SSL에 사용할 수 없습니다. 따라서 Build Forge 시스템에서 SSL이 사용 가능한 경우 Build Forge 시스템에서 플러그인을 설치할 수 없습니다. 해결 방법은 비보안 웹 서버에서 플러그인 설치 파일을 사용 가능하게 하거나 수동으로 사용자에게 분배하는 것입니다.

<bfinstall>/webroot/public/rtc(RTC 버전 1.x의 경우) 또는 <bfinstall>/webroot/public/rtc2(RTC 버전 2.x의 경우)에서 다음 파일이 필요합니다.

- features 디렉토리
- plugins 디렉토리
- site.xml 파일

## 이 태스크 정보

rtc 또는 rtc2 디렉토리를 사용할 수 있게 되면 사용자가 Rational Team Concert 클라이언트에서 다음 단계를 수행합니다.

## 프로시저

1. 도움말 > 소프트웨어 업데이트 > 찾기 및 설치를 선택하십시오.
2. 설치할 새 기능 검색을 클릭한 후 다음을 클릭하십시오. 방문할 업데이트 사이트 대화 상자가 표시됩니다.
3. 새 사이트를 작성하십시오. 다음 프로시저 중 하나를 선택하십시오.
  - 원격 서버에서 파일 가져오기:
    - a. 새 원격 사이트를 클릭하십시오. 새 원격 사이트 대화 상자가 표시됩니다.
    - b. 이름 필드에 『Build Forge 커넥터 업데이트 사이트』를 입력하십시오.

- c. 파일의 위치를 입력하십시오.

`http://host/prism/eclipse/updateSite/site.xml`

*host*는 웹 서버의 IP 주소 또는 호스트 이름입니다.

*path*는 서버 루트에서 파일 저장 위치까지의 경로입니다.

- d. 확인을 클릭하십시오.

- 로컬 호스트에서 파일 가져오기:

- a. 새 로컬 사이트를 클릭하십시오. 시스템에 새 로컬 사이트 대화 상자가 표시됩니다.

- b. 이름 필드에 『Build Forge 커넥터 업데이트 사이트』를 입력하십시오.

- c. 파일의 위치를 입력하십시오.

`file://path/prism/eclipse/updateSite/site.xml`

*path*는 파일의 위치를 지정합니다.

- d. 확인을 클릭하십시오.

4. 방문할 업데이트 사이트 대화 상자에서 **Build Forge** 커넥터 업데이트 사이트 선택란을 선택한 다음 완료를 클릭하십시오.

5. 검색 결과 대화 상자에 사용 가능한 기능의 목록이 표시됩니다. 제공된 기능을 모두 선택한 후 다음을 클릭하십시오.

참고: 리플렉터 플러그인에 빈도(Frequency) 플러그인이 필요합니다. 단독으로 설치된 경우 실행되지 않습니다.

6. 라이선스 계약을 읽고 라이선스 계약 내용에 동의합니다를 선택한 후 다음을 클릭하십시오.

7. 기능을 설치할 위치를 선택하십시오. 새 위치를 추가하려면 위치 변경을 클릭한 다음 원하는 위치를 찾으십시오.

8. 완료를 클릭하십시오.

9. 기능 검증 대화 상자가 표시되면 설치를 클릭하십시오. 플러그인은 서명되지 않은 기능이므로 대화 상자가 나타납니다. 모두 설치를 선택하지 않으면, 각 기능이 설치될 때마다 대화 상자가 표시됩니다.

10. 변경사항을 적용하기 위해 Rational Team Concert를 다시 시작할 것인지 묻는 메시지가 표시됩니다. 예를 클릭하십시오.

11. Build Forge를 다시 시작하십시오.

## Rational Team Concert 어댑터 구성

### 이 태스크 정보

Rational Build Forge를 Rational Team Concert 소스 저장소에 연결하는 어댑터는 Build Forge를 설치할 때 자동으로 설치됩니다. 어댑터는 Rational Build Forge 콘솔 호스트에서 실행되며 클라이언트로 Rational Team Concert와 직접 통신합니다. Rational Build Forge 콘솔 호스트에 Rational Team Concert 클라이언트를 설치할 필요가 없습니다. 다음과 같이 개별 Build Forge 프로젝트를 구성하여 Rational Team Concert 소스에 액세스할 수 있습니다.

### 프로시저

#### 1. 서버 자원을 정의하십시오.

- 이 서버 자원의 이름을 입력하십시오.
- 호스트를 입력하십시오. Rational Team Concert 서버가 실행되는 호스트의 도메인 이름을 완전하게 입력해야 합니다.

참고: 보통 이 특성은 Build Forge 에이전트를 나타냅니다. Rational Team Server 호스트에는 Build Forge 에이전트를 설치할 필요가 없습니다. 이 특성은 특히 Rational Team Concert와 직접 통신하는 데 사용됩니다.

- 기타 특성에 대한 값을 적절하게 입력하십시오.

참고: RTC 도구 scm이 RTC 서버에 대한 시작 프로파일 또는 사용자 프로파일의 경로에 있어야 합니다. 어댑터에서 scm을 사용하여 소스 파일에 액세스합니다.

#### 2. Rational Team Concert 저장소를 사용하는 각 프로젝트에 대한 어댑터 링크를 설정하십시오. 다음 지시사항은 프로젝트에 대한 어댑터 링크를 설정하는 방법에 대해 설명합니다.

- a. 콘솔에서 프로젝트 > 어댑터 링크로 이동하십시오.
- b. 새 어댑터 링크를 클릭하십시오. 세부사항 탭에서 해당 특성을 설정하고 저장을 클릭하십시오.
  - 상태 - 활성을 선택합니다.
  - 이름 - 어댑터 링크의 이름을 입력합니다.
  - 어댑터
    - Rational Team Concert 버전 1.x의 경우 JazzSCM 선택
    - Rational Team Concert 버전 2.x의 경우 JazSCMv2 선택
  - 프로젝트 - 어댑터 링크를 적용할 프로젝트를 선택합니다.
  - 환경 - 이 어댑터 링크에 사용할 환경을 선택합니다. 이는 기존 환경이어야 합니다.



- c. 작성한 어댑터 링크를 클릭하십시오.
  - d. 환경 채우기를 클릭하십시오. 이 단계에서 지정된 환경을 어댑터에 정의된 변수로 채웁니다(JazzSCM.xml).
  - e. 저장을 클릭하십시오.
3. 어댑터에서 제공된 변수를 업데이트하십시오. 설정하고 채운 환경에서 어댑터에서 제공된 변수를 편집합니다.

어댑터에 다음과 같은 네 가지 환경 변수가 설정되어 있습니다.

- **Current\_Date** - 현재 날짜를 설정합니다. 이는 시간소인을 적용하는 데 사용됩니다. 이 정의를 변경하지 마십시오.
- **Last\_Run** - 시스템에서 자동으로 업데이트됩니다. 이 정의를 변경하지 마십시오.
- **Directory\_Path** - 저장소에서 검색된 소스 파일의 위치를 설정합니다. 기본적으로 C:\temp로 설정되어 있습니다. 이 디렉토리를 사용할 임시 디렉토리로 변경하십시오.

작업 실행 후 기본적으로 디렉토리가 정리되지 않습니다. 이미 실행된 작업에서 이전 디렉토리를 삭제하십시오.

- **Jazz\_Server** - Rational Team Concert 서버의 위치를 설정합니다. 기본적으로 \$BFServer로 설정되어 있습니다. 이 설정을 Rational Team Concert 서버를 지정하는 서버 자원의 이름 특성으로 변경해야 합니다.
- a. 콘솔에서 환경으로 이동하십시오.
  - b. 어댑터 링크에서 사용되는 환경을 선택하십시오.
  - c. 변경할 변수를 선택하고 편집하십시오.
  - d. 저장을 클릭하십시오.

## 결과

이제 프로젝트를 실행할 수 있습니다. 프로젝트를 실행할 때마다 Rational Team Concert 저장소에 연결됩니다. 이를 통해 마지막으로 프로젝트를 실행한 이후에 RTC 서버에서 변경된 소스 파일이 업데이트됩니다.

추가 자원:

- Rational Team Concert 버전 1.x: JazzSCM 어댑터는 `<bfinstall>/interface/JazzSCM.xml`에 있습니다. 문서 편집기 또는 XML 리더를 사용하여 이를 열 수 있습니다.
- Rational Team Concert 버전 2.x: JazzSCMv2 어댑터는 `<bfinstall>/interface/JazzSCMv2.xml`에 있습니다. 문서 편집기 또는 XML 리더를 사용하여 이를 열 수 있습니다.



- 어댑터에서 사용되는 SCM 명령은 Rational Team Concert 문서에 설명되어 있습니다. 서버에 대한 연결을 테스트하고 어댑터에서 사용할 명령을 테스트하는 경우 이 명령 및 기타 명령을 사용하십시오.



---

## 제 34 장 Rational Build Forge 및 StarTeam

BuildForge는 소스 코드 관리를 위해 StarTeam과 함께 작동할 수 있습니다.

이 주제에서는 StarTeam과 통합하기 위한 추가 설정 요구사항을 설명하고 제공된 샘플 어댑터 템플릿의 특성을 요약합니다.

StarTeam 어댑터 템플릿 샘플은 기준선에 대한 변경사항을 분석하는 방법을 제공합니다. 일반적으로 변경 분석은 빌드 방지를 위해 사용됩니다. 기준선 컴포넌트가 변경되지 않은 경우 다시 빌드되지 않습니다.

### StarTeam 어댑터에 대한 추가 설정 요구사항

일반 요구사항은 529 페이지의 『어댑터 요구사항』의 내용을 참조하십시오.

1. 에이전트와 동일한 호스트에서 StarTeam 명령행 클라이언트를 설치하십시오.
2. 사용할 Build Forge에 대한 StarTeam 사용자 계정을 설정하십시오.
3. 클라이언트에 액세스할 수 있도록 에이전트 환경을 설정하십시오.

### StarTeam 어댑터 템플릿 샘플

다음 어댑터 템플릿 샘플이 제공됩니다.

#### StarTeamBaseline

1. StarTeam 보기용 폴더를 조회하여 파일에 대한 정보를 수집합니다.
2. BOM 보고서에 파일 이름, 상태, 개정 및 분기 정보를 기록합니다.

어댑터에 설정된 변수:

- USER
- PASS
- HOST
- PORT
- PROJECT
- VIEW
- DIR
- BFCLIENT

환경에 대한 추가 필수 변수:

- P4USER
- P4PASSWD

이 변수에는 숨겨짐 지정 특성을 사용할 수 없습니다. 사용자 및 비밀번호는 단계 로그에 일반 텍스트로 기록됩니다.

### **StarTeamByDate**

1. StarTeam API를 사용하여, StarTeam 보기를 조회하여 현재 날짜와 이전 어댑터 실행 간의 변경사항을 식별합니다.
2. BOM 보고서에 변경된 파일과 디렉토리, 사용자, 버전, 날짜 및 변경 주석 정보를 기록합니다.
3. diff 명령 출력의 변경 세부사항을 BOM 보고서에 기록합니다.

어댑터에 설정된 변수:

- STA\_USER
- STA\_PASS
- STA\_HOST
- STA\_PORT
- STA\_PROJECT
- STA\_VIEW
- STA\_DIR
- BFCLIENT
- LASTRUN
- CURDATE
- EMAILCHANGES
- STARTEAM80JAR

환경에 대한 추가 필수 변수:

- P4USER
- P4PASSWD

이 변수에는 숨겨짐 지정 특성을 사용할 수 없습니다. 사용자 및 비밀번호는 단계 로그에 일반 텍스트로 기록됩니다.

---

## 제 35 장 Rational Build Forge 및 Subversion

BuildForge는 소스 코드 관리를 위해 SVN(Subversion)과 함께 작동할 수 있습니다.

이 주제에서는 SVN과 통합하기 위한 추가 설정 요구사항을 설명하고 제공된 샘플 어댑터 템플릿의 특성을 요약합니다.

SVN 어댑터 템플릿 샘플은 기준선에 대한 변경사항을 분석하는 방법을 제공합니다. 일반적으로 변경 분석은 빌드 방식을 위해 사용됩니다. 기준선 컴포넌트가 변경되지 않은 경우 다시 빌드되지 않습니다.

### SVN 어댑터에 대한 추가 설정 요구사항

일반 요구사항은 529 페이지의 『어댑터 요구사항』의 내용을 참조하십시오. SVN에 대한 추가 요구사항은 없습니다.

### SVN 어댑터 템플릿 샘플

다음 어댑터 템플릿 샘플이 제공됩니다.

#### SubversionByDate

1. Subversion에서 과거 날짜와 현재 날짜 간에 발생한 저장소 변경사항을 조회합니다.
2. BOM 보고서에 변경 유형, 개정, 사용자, 파일 또는 디렉토리와, 변경 날짜 정보를 기록합니다.
3. BOM 보고서에 파일 이름, 상태, 개정 및 분기 정보를 기록합니다.

사용되는 변수:

- SVN\_CLIENT
- SVN\_OPTS
- SVN\_REPOSITORY
- SVN\_LAST\_REV
- SVN\_LAST\_DATE

#### SubversionByRev

이 어댑터를 설정하는 방법을 설명하는 문서는 [jazz.net](https://jazz.net/wiki/bin/view/Main/RationalBuildForge/SVNAdaptorExample)에 있습니다. <https://jazz.net/wiki/bin/view/Main/RationalBuildForge/SVNAdaptorExample>.

1. Subversion에서 현재 개정과 이전 개정 간에 발생한 저장소 변경사항을 조회합니다.

2. 각 변경에 대해 BOM 보고서에 개정, 사용자, 변경 유형, 파일 또는 디렉토리 경로와, 변경 날짜 정보를 기록합니다.
3. diff 명령 출력의 변경 세부사항을 BOM 보고서에 기록합니다.

사용되는 변수:

- SVN\_CLIENT
- SVN\_OPTS
- SVN\_REPOSITORY
- SVN\_LAST\_REV
- SVN\_LAST\_DATE

---

## 제 36 장 Rational Build Forge 및 WebSphere 제품

이 절에서는 Build Forge®를 WebSphere 제품과 통합하는 방법에 대해 설명합니다.

- Apache Tomcat이 아닌 WebSphere Application Server를 사용하여 Build Forge 서비스 및 빠른 보고서 실행
- Apache가 아닌 IHS(IBM HTTP Server)를 웹 애플리케이션 서버로 사용

---

### Apache Tomcat 대신 WebSphere Application Server 사용

제공된 Apache Tomcat 애플리케이션 서버 대신 WebSphere Application Server를 애플리케이션 서버로 사용할 수 있습니다. Build Forge 서비스 및 Build Forge 도움말 애플리케이션을 실행하도록 이를 구성합니다.

WebSphere Application Server에는 세 가지 구성 영역이 있습니다.

- Build Forge 서비스 및 도움말 애플리케이션을 실행하도록 이를 구성하십시오. 필수 항목입니다.
- Build Forge 서비스 애플리케이션에 대해 Java 2 보안을 구성하십시오. WebSphere Application Server에서 Java 2 보안을 실행 중인 경우 이는 필수입니다.
- Build Forge에서 SSL 및 비밀번호 암호화에 대한 지원을 구성하십시오. Build Forge에서 이러한 기능을 구성한 경우 이는 필수입니다.

#### 전제조건

- 특정 제품에 대한 세부 시스템 요구사항 사이트를 확인하십시오. 지원되는 WebSphere Application Server의 버전을 알아보려면 Rational Build Forge 에디션 및 버전에 대한 조회를 실행하십시오.
- 라이선스 서버의 제한사항 때문에 Build Forge 콘솔 및 WAS는 동일한 운영 체제 및 하드웨어 플랫폼에서 실행되어야 합니다.
- Build Forge 서비스 계층을 배치하려면 먼저 WebSphere Application Server에 최신 수정팩이 적용되어야 합니다.

**중요사항:** 이 구성을 사용하는 경우 Build Forge를 시작하기 전에 WebSphere Application Server에서 Build Forge 서비스 및 Build Forge 도움말 애플리케이션을 시작해야 합니다. 이들은 자동으로 시작되지 않습니다.

## Build Forge 애플리케이션을 실행하도록 Websphere Application Server 구성

서비스 애플리케이션 및 도움말 애플리케이션을 실행하도록 WebSphere Application Server를 구성하십시오.

### 이 태스크 정보

WebSphere 관리 콘솔에서 다음 프로시저의 단계를 수행하십시오.

WebSphere 버전 8.0부터는 WebSphere 관리 콘솔의 사용자 인터페이스가 약간 변경되었습니다. 아래 지시사항은 버전 7 이하에 대한 지시사항입니다. 버전 8.0에서의 다음 차이점은 이 프로시저에 적용됩니다.

- 애플리케이션에 액세스하려면 애플리케이션 > 애플리케이션 유형 > 엔터프라이즈 애플리케이션을 사용하십시오.
- 사용자가 애플리케이션을 선택한 후 컨텍스트 루트를 설정하는 위치가 이동되었습니다.
- 자원에 대한 MBeans 작성에 대한 선택란이 표시되지 않습니다.

### 프로시저

1. WebSphere 관리 콘솔을 여십시오. 콘솔의 URL은 다음과 같습니다.
  - `http://<was_host>:<was_port>/ibm/console`. 9060은 기본 포트입니다.
  - `https://<was_host>:<was_port>/ibm/console`. 기본 포트는 9043입니다. WebSphere 관리 보안이 사용 가능한 경우 이 URL을 사용하십시오.
2. JDBC 드라이버에 대한 지원을 구성하십시오.
  - a. 새 변수 RBF\_JDBC\_DRIVER\_PATH를 작성하십시오. 환경 > WebSphere 변수에서 변수를 작성하십시오. 해당 범위는 WAS 노드 및 서버여야 합니다. 데이터베이스 드라이버 JAR 파일이 포함되어 있는 디렉토리에 값을 설정하십시오.
  - b. 마스터 구성에 대한 변경사항을 저장하십시오.
  - c. 새 변수를 사용 가능하게 하려면 WebSphere 서버를 중지하고 다시 시작하십시오.
  - d. 새 공유 라이브러리 RBF\_JDBC\_LIBRARY를 작성하십시오.

환경 > 공유 라이브러리에서 라이브러리를 작성하십시오. 작성한 RBF\_JDBC\_DRIVER\_PATH를 사용하여 JDBC 디바이스 드라이버의 JAR 파일 이름을 추가하십시오. 목록에 둘 이상의 jar 파일이 있는 경우 고유한 행에 각 jar 파일을 배치하십시오(DB2와 동일함).

다음은 MySQL 데이터베이스 드라이버에 대한 예제입니다.

```
${RBF_JDBC_DRIVER_PATH}\mysql-connector-java-5.0.5-bin.jar
```



다음 예제는 DB2용입니다.

```
${RBF_JDBC_DRIVER_PATH}\db2jcc.jar  
${RBF_JDBC_DRIVER_PATH}\db2jcc_license_cu.jar
```

Unix 또는 Linux의 경우 이 예제에서 백슬래시(\) 대신 포워드 슬래시(/)를 사용하십시오.

- e. 마스터 구성에 대한 변경사항을 저장하십시오.
3. Build Forge 애플리케이션 WAR 파일을 설치하십시오.
  - a. 애플리케이션 > 엔터프라이즈 애플리케이션을 여십시오.
  - b. 설치를 클릭하십시오.
  - c. rbf-services.war 파일을 찾아보십시오. rbf-services를 컨텍스트 루트로 사용하십시오.
    - Windows 시스템:  
`<bfinstall>\PrepForExternal\rbf-services.war`
    - UNIX 및 Linux 시스템:  
`<bfinstall>/Platform/PrepForExternal/rbf-services.war`
  - d. 다음을 클릭하고, 선택되어 있지 않은 경우 다음 선택란을 지우십시오.
    - 자원에 대한 MBeans 작성
  - e. 완료 단추가 표시될 때까지 다음을 클릭한 후 완료를 클릭하십시오.
  - f. 설치 텍스트의 맨 아래에 있는 저장을 클릭하십시오.
4. Build Forge 도움말 WAR 파일을 설치하십시오.
  - a. 애플리케이션 > 엔터프라이즈 애플리케이션을 여십시오.
  - b. 설치를 클릭하십시오.
  - c. Build Forge 설치 디렉토리에서 BuildForgeHelp.war 파일을 찾아보십시오. 컨텍스트 루트로 /BuildForgeHelp를 사용하십시오.
  - d. 다음을 클릭하고, 선택되어 있지 않은 경우 다음 선택란을 지우십시오.
    - 자원에 대한 MBeans 작성
  - e. 완료 단추가 표시될 때까지 다음을 클릭한 후 완료를 클릭하십시오.
  - f. 설치 텍스트의 맨 아래에 있는 저장을 클릭하십시오.
5. RBF\_JDBC\_LIBRARY를 공유 라이브러리 참조로 설정하십시오.
  - a. 애플리케이션 > 엔터프라이즈 애플리케이션을 여십시오.
  - b. rbf-services.war 링크를 클릭하십시오.
  - c. 공유 라이브러리 참조를 클릭하십시오.
  - d. 서비스 계층 로그인 서블릿 상자를 선택하십시오.
  - e. 참조 공유 라이브러리를 클릭하십시오.

- f. 목록에 RBF\_JDBC\_LIBRARY를 추가하십시오.
6. 클래스 로더 순서를 설정하십시오. 모듈 관리에서 서비스 계층 로그인 서블릿을 선택하고 드롭 다운 상자에서 클래스 로더 순서를 찾으십시오. 해당 값을 먼저 로컬 클래스 로더와 함께 로드된 클래스(마지막 상위)로 변경하십시오.
7. 마스터 구성에 대한 변경사항을 저장하십시오.
8. Build Forge 구성 파일을 확인하십시오. 파일의 `services_url` 항목이 올바른 URL을 가리키는지 확인하십시오. WAS를 사용하는 경우 `buildforge.conf`에서 이 특성 값은 다음과 비슷합니다. `https://<hostname>:9443/rbf-services` 또는 `https://<hostname>:9080/rbf-services`. `buildforge.conf` 파일은 다음 위치에 있습니다.
  - Windows 시스템  
`<bfinstall>\`
  - UNIX 및 Linux 시스템  
`<bfinstall>/Platform/`
9. `rbf-services`의 경우 `bin` 디렉토리의 모든 파일에 실행 권한을 추가하십시오 (UNIX 및 Linux 시스템). 이 예제에서는 프로파일 `AppSrv01`이 사용됩니다. `AppSrv01`이 아닌 경우 프로파일 이름을 사용하십시오.  
`chmod -R 755 <WAS-install>/AppServer/profiles/AppSrv01/installedApps/<node>/rbf-services_war.ear/rbf-services.war/bin`
10. 토큰 라이브러리 디렉토리를 포함하도록 `PATH` 변수를 업데이트하십시오.
  - Windows 시스템  
`<bfinstall>\token_libs`
  - UNIX 및 Linux 시스템  
`<bfinstall>/Platform/token_libs`
11. 64비트 Linux 또는 WAS를 사용 중인 경우 제공된 64비트 라이브러리를 복사해야 합니다.
  - a. 32비트 라이브러리 파일의 이름을 바꾸십시오.
    - Windows 시스템: `<bfinstall>\token_libs\rcl_rational.dll`을 `<bfinstall>\token_libs\rcl_rational.dll.old`로 이름을 바꾸십시오.
    - UNIX 및 Linux 시스템: `<bfinstall>/Platform/token_libs/librcl_rational.so`를 `<bfinstall>/Platform/token_libs/librcl_rational.so.old`로 이름을 바꾸십시오.
  - b. 64비트 라이브러리 파일을 `token_libs_64`에서 `token_libs`로 복사하십시오. 명령에서는 행 바꾸기를 사용하지 마십시오. 여기서는 읽기 쉽게 하기 위해 사용되었습니다.
    - Windows 시스템

```
copy <bfinstall>\token_libs_64\rcl_rational.dll  
<bfinstall>\token_libs\rcl_rational.dll
```

- UNIX 및 Linux 시스템

```
cp <bfinstall>/Platform/token_libs_64/librcl_rational.so  
<bfinstall>/Platform/token_libs/librcl_rational.so
```

## WebSphere Application Server에서 Build Forge에 대해 Java 2 보안 구성

Java 2 보안이 사용 가능하도록 설정된 WebSphere Application Server를 실행하는 경우 Build Forge 서비스 계층을 구성하여 사용해야 합니다.

### 이 태스크 정보

WebSphere 관리 콘솔에서의 구성을 위해 다음 단계를 수행하십시오.

#### 프로시저

1. 애플리케이션 > 엔터프라이즈 애플리케이션을 여십시오.
2. **rbf-services\_war**을 선택하고 업데이트를 클릭하십시오.
3. 단일 파일 바꾸기 또는 추가를 선택하십시오.
4. 바꾸거나 추가할 파일에 설치된 애플리케이션 아카이브 파일로 시작하는 경로 지정에 META-INF/was.policy를 입력하십시오.
5. 로컬 파일 시스템을 선택하고 운영 체제의 was.policy 파일을 찾아본 후 다음을 클릭하십시오.
  - Windows 시스템:  
`<bfinstall>/samples/projects/was.policy`
  - UNIX 및 Linux 시스템:  
`<bfinstall>Platform/samples/projects/was.policy`
6. 확인을 클릭하십시오.
7. 마스터 구성에 대한 변경사항을 저장한 다음 Build Forge 서비스 애플리케이션을 중지한 후 시작하십시오.

## WebSphere Application Server에서 Build Forge에 대해 SSL 및 비밀번호 암호화 지원 구성

WebSphere Application Server에서 Build Forge 애플리케이션에 대해 SSL 및 비밀번호 암호화 또는 둘 다 사용하도록 설정

### 이 태스크 정보

SSL 및 비밀번호 암호화에 대한 지원을 사용하도록 설정하는 추가 구성이 필요합니다. 하나 또는 둘 다 구성할 수 있습니다. WebSphere Application Server에서 SSL에 대

한 구성은 Build Forge에서 SSL을 사용하도록 설정하기 전에 수행되어야 합니다.

## 프로시저

1. `<bfinstall>\keystore` 디렉토리를 `<WAS_INSTALL_DIR>\AppServer\profiles\AppSrv01`에 복사하십시오. AppSrv01이 아닌 프로파일을 사용할 경우, 키 저장소 디렉토리를 해당 프로파일 디렉토리에 복사하십시오.

경로는 UNIX 및 Linux 시스템에서 동일합니다. 이 디렉토리에는 애플리케이션 서버 프로파일이 포함되어 있습니다. 디렉토리는 WebSphere Application Server의 버전에 따라 달라질 수 있습니다.

2. WebSphere 콘솔에 로그인하십시오. WebSphere 관리용 콘솔을 사용하십시오.
3. 사용자 정의 특성 페이지로 이동하십시오. 서버 > 서버 유형 > **WebSphere Application Server** > **server1** > **Java** 및 프로세스 관리 > 프로세스 정의 > **JVM(Java Virtual Machine)** > 사용자 정의 특성을 여십시오.

기본 서버 이름은 **server1**입니다. 서버에 다른 이름이 있는 경우 그 이름을 사용하십시오.

4. SSL을 사용하도록 설정하십시오. 사용자 정의 특성이 `bfclient.conf` 파일을 가리키도록 설정하십시오.

- Windows

`com.buildforge.client.config=<bfinstall>\bfclient.conf`

- UNIX 또는 Linux

`com.buildforge.client.config=<bfinstall>/Platform/bfclient.conf`

5. 비밀번호 암호화를 사용하도록 설정하십시오. 사용자 정의 특성이 `bfpassword.conf` 파일을 가리키도록 설정하십시오.

- Windows

`com.buildforge.password.encryption.file=<bfinstall>\bfpassword.conf`

- UNIX 또는 Linux

`com.buildforge.password.encryption.file=<bfinstall>/Platform/bfpassword.conf`

6. WebSphere Application Server를 다시 시작하십시오. 다시 시작하면 변경사항이 적용됩니다.

## Build Forge 애플리케이션을 실행하기 위해 Websphere Application Server를 사용하기 위한 시작 순서

Build Forge를 시작하기 전에 WebSphere Application Server에서 Build Forge 애플리케이션을 시작합니다.

## 이 태스크 정보

Build Forge를 시작하기 전에 WebSphere Application Server에서 서비스 및 도움말 애플리케이션을 시작하십시오.

### 프로시저

1. WebSphere 관리 콘솔을 여십시오.
2. 애플리케이션 > 엔터프라이즈 애플리케이션을 선택하십시오.
3. **rbf-services.war** 및 **BuildForgeHelp.war**을 선택하십시오.
4. 시작을 클릭하십시오.
5. Build Forge를 시작하십시오.

---

## Apache HTTP Server 대신 IBM HTTP Server 사용

기본적으로 설치되어 있는 Apache HTTP 서버 대신 관리 콘솔에서 사용할 IHS(IBM HTTP Server)를 구성할 수 있습니다.

### 시작하기 전에

- 라이선스 서버의 제한사항 때문에 Build Forge 콘솔 및 IBM HTTP Server는 동일한 운영 체제 및 하드웨어 플랫폼에서 실행되어야 합니다.
- 제공되는 사전 컴파일된 **PHP** 모듈을 사용해야 합니다. 사용자가 직접 컴파일한 PHP 사용을 시도하지 마십시오. 운영 체제용 모듈이 실제로 포함되어 있는지 확인하려면 릴리스 정보를 검사하십시오. 그렇지 않을 경우 지원을 사용하여 모듈을 컴파일하도록 해야 합니다.
- 이 설정 사용 시 어려움이 있는 경우 IBM 지원 센터로 문의하십시오.
- 예제에서 다음 디렉토리는 IBM HTTP Server 설치 디렉토리로 사용됩니다. 설치 디렉토리가 서로 다른 경우 해당 디렉토리를 대체하십시오.
  - Windows: C:\Program Files\IBM\HTTPServer
  - Unix 및 Linux: /opt/IBM/HTTPServer

## 이 태스크 정보

이 주제는 다음과 같은 태스크를 수행하는 방법에 대해 설명합니다.

1. 사전 컴파일된 PHP 파일을 설정하십시오.
2. Build Forge 웹 애플리케이션을 지정하도록 IBM HTTP Server 구성 파일을 수정하십시오.
3. 프록시 서버를 사용하여 데이터베이스에 액세스하는 경우 프록시 서버를 사용하도록 PHP를 수정하십시오.

4. SSL을 사용하는 경우 SSL을 통해 Build Forge에서 작업하도록 IHS를 구성하십시오.
5. Windows 시스템: IBM HTTP Server 설치 디렉토리의 libeay32.dll을 대체하십시오.
6. AIX 시스템: 추가 설정

## 사전 컴파일된 PHP 파일 설정

### 이 태스크 정보

이 태스크는 제공된 PHP 파일을 찾아서 지정된 위치에 복사하는 방법에 대해 설명합니다. 파일 디렉토리는 다음과 같습니다.

- Windows 시스템
 

```
<binstall>\ihs_modules
```
- UNIX 및 Linux 시스템
 

```
<binstall>/Platform/ihs_modules
```

### 프로시저

1. PHP 모듈을 복사하십시오. 필요한 경우 modules 디렉토리를 작성하십시오.
  - Windows 시스템
 

```
cd <binstall>\ihs_modules\modules\
copy *.dll C:\Program Files\IBM\HTTPServer\modules
```
  - UNIX 및 Linux 시스템
 

```
cd <binstall>/Platform/ihs_modules/modules
cp libphp5.so /opt/IBM/HTTPServer/modules
```
2. PHP 확장기능을 복사하십시오. 필요한 경우 extensions 디렉토리를 작성하십시오.
  - Windows 시스템
 

```
cd <binstall>\ihs_modules\lib\php\extensions\
copy *.dll C:\Program Files\IBM\HTTPServer\extensions
```
  - UNIX 및 Linux 시스템
 

```
cd <binstall>/Platform/ihs_modules/lib/php/extensions
/no-debug-zts-20090626
cp *.so /opt/IBM/HTTPServer/extensions
```
3. 제공된 php.ini 파일을 복사하십시오. 필요한 경우 conf 디렉토리를 작성하십시오.
  - Windows 시스템
 

```
cd <binstall>\ihs_modules
copy php.ini C:\Program Files\IBM\HTTPServer\conf
```
  - UNIX 및 Linux 시스템
 

```
cd <binstall>/Platform/ihs_modules
cp php.ini /opt/IBM/HTTPServer/conf
```

4. conf 디렉토리에 복사한 php.ini 파일을 편집하여 extensions 디렉토리에 포인터를 포함시키십시오. 다음 행을 추가하거나 대체하십시오.
  - Windows 시스템
 

```
extension_dir = C:\Program Files\IBM\HTTPServer\extensions
```
  - UNIX 및 Linux 시스템
 

```
extension_dir = /opt/IBM/HTTPServer/extensions
```
5. php.ini 파일을 시스템에 대한 정보로 편집하십시오. Apache HTTP Server를 위해 제공된 php.ini 파일을 안내서로 사용할 수 있습니다. 이 파일은 다음 위치에 제공되어 있습니다.
  - Windows 시스템
 

```
<bfinstall>\Apache\php\php.ini
```
  - UNIX 및 Linux 시스템
 

```
<bfinstall>/server/apache/conf/php.ini
```
6. php.ini에서 사용되는 모든 디렉토리에 대한 쓰기 액세스를 추가하십시오. UNIX 및 Linux 시스템에서는 디렉토리에 대해 다음 명령을 실행하십시오.
 

```
chmod -R 777
```

## IBM HTTP Server 구성 파일 편집

### 프로시저

1. 서버 설치의 conf 디렉토리에서 IHS(IBM HTTP Server)에 대한 httpd.conf 파일을 찾으십시오.
2. 예제에 표시된 대로 Build Forge 웹 애플리케이션을 지정하도록 DocumentRoot 설정을 수정하십시오. 다음 예제에서 Build Forge 설치 디렉토리는 /opt/buildforge입니다.

```
<VirtualHost *:80>
    ServerAdmin build@yourdomain.com
    DocumentRoot /opt/buildforge/webroot/public
    ServerName ausbuild01.yourdomain.com
    ServerAlias build.yourdomain.com mc.yourdomain.com

    #optional server aliases
    ErrorLog logs/ausbuild.error_log
    CustomLog logs/ausbuild.access_log common
</VirtualHost>
```

3. PHP 관련 항목을 추가하십시오.

- Windows 시스템
 

```
LoadModule php5_module "C:\Program Files\IBM\HTTPServer\modules\php5apache2_2.dll"

<IfModule dir_module>
    DirectoryIndex index.html index.php
</IfModule>
```

```
AddType application/x-httpd-php .php

# PHPIniDir is the location of the php.ini file
PHPIniDir "C:\Program Files\IBM\HTTPServer\conf"
```

- UNIX 및 Linux 시스템

```
LoadModule php5_module "/opt/IBM/HTTPServer/modules/libphp5.so"
```

```
<IfModule dir_module>
    DirectoryIndex index.html index.php
</IfModule>
```

```
AddType application/x-httpd-php .php

# PHPIniDir is the location of the php.ini file
PHPIniDir "/opt/IBM/HTTPServer/conf"
```

4. <bfinstall>/webroot/templates\_c에 대한 기본 사용자인 daemon을 사용하도록 User 및 Group 명령문을 변경하십시오.

```
User daemon
Group daemon
```

5. 필요한 경우 IHS 포트 번호를 변경하십시오. 기본 포트 번호는 80입니다. httpd.conf에 대해 필요한 기타 모든 사항을 변경하십시오.

## PHP에서 프록시 서버 지정

### 이 태스크 정보

선택적: 이 단계는 관리 콘솔이 데이터베이스에 액세스할 때 프록시 서버를 사용해야 하는 경우에만 필요합니다.

### 프로시저

PHP 구성 파일 php.ini를 편집하십시오. <php-install>/lib,에 위치합니다(예: /usr/local/php-5.3.6).

다음 항목을 추가하십시오.

```
bf_proxyHost=<your_proxy_server_hostname>
bf_proxyPath=<your_proxy_path>
bf_symlinkPath=<symlink_to_proxy_path>
```

## IHS에 필요한 SSL 구성

### 이 태스크 정보

IHS의 일반 SSL 설정 이외에 IHS가 Build Forge에서 작동하는 데 필요한 추가 요구 사항이 있습니다.



## 프로시저

1. 경로에 Build Forge 도구 디렉토리를 포함하십시오.
  - Windows 시스템
    - *bfinstall\ibmjdk\bin*
    - *bfinstall\openssl*
  - UNIX 또는 Linux 시스템
    - *bfinstall/server/ibmjdk/bin*
2. 라이브러리 경로에 Build Forge 도구 디렉토리를 포함시키십시오.
  - AIX 시스템(LIBPATH)
    - *bfinstall/openssl*
  - UNIX 또는 Linux 시스템(LD\_LIBRARY\_PATH)
    - *bfinstall/openssl*
3. JAVA\_HOME에 Build Forge 도구 디렉토리를 포함하십시오.
  - Windows
    - *bfinstall\ibmjdk*
  - UNIX 또는 Linux
    - *bfinstall/server/ibmjdk*
4. Build Forge 키를 PKCS12에서 CMS로 변환하십시오. GSKIT 도구의 최신 버전을 사용하십시오. gsk7\bin(Windows) 또는 bin(UNIX 또는 Linux)에서 다음 명령을 실행하십시오(명확히 하기 위해 행 바꾸기가 추가됨).

```
gsk7cmd -keydb
        -convert
        -db bfinstall\keystore\buildForgeKeyStore.p12
        -pw buildForgeKeyStore_password
        -old_format pkcs12
        -new_format cms
```
5. Stash 파일에 비밀번호를 저장하십시오. IHS에서는 이 파일을 사용하여 시작 중에 비밀번호를 가져옵니다. 이 파일이 없으면 IHS가 비밀번호를 묻는 프롬프트를 표시합니다. GSKIT 도구의 최신 버전을 사용하십시오. gsk7\bin(Windows) 또는 bin(UNIX 또는 Linux)에서 다음 명령을 실행하십시오(명확히 하기 위해 행 바꾸기가 추가됨).

```
gsk7cmd -keydb
        -stashpw
        -db bfinstall\keystore\buildForgeKeyStore.kdb
        -pw buildForgeKeyStore_password
```
6. httpd.conf를 수정하십시오. Windows의 경우 다음 항목을 포함하십시오. 사용자가 https를 통해서만 액세스하도록 할 경우 Listen 80을 주석 처리합니다.

```
LoadModule ibm_ssl_module modules/mod_ibm_ssl.so
# Listen 80
Listen 0.0.0.0:443
<VirtualHost *:443>
SSLEnable
SSLClientAuth None
SSLProtocolDisable SSLv2
SSLServerCert buildforge
KeyFile bfinstall\keystore\buildForgeKeyStore.kdb
SSLStashFile bfinstall\keystore\buildForgeKeyStore.sth
ErrorLog bfinstall\Apache\logs\ssl_error.log
TransferLog bfinstall\Apache\logs\transfer.log
</VirtualHost>
```

UNIX 및 Linux의 경우 다음 항목을 포함하십시오. 사용자가 https를 통해서만 액세스하도록 할 경우 Listen 80을 주석 처리합니다.

```
LoadModule ibm_ssl_module modules/mod_ibm_ssl.so
# Listen 80
Listen 0.0.0.0:443
<VirtualHost *:443>
SSLEnable
SSLClientAuth None
SSLProtocolDisable SSLv2
SSLServerCert buildforge
KeyFile bfinstall/Platform/keystore/buildForgeKeyStore.kdb
SSLStashFile bfinstall/Platform/keystore/buildForgeKeyStore.sth
ErrorLog bfinstall/server/apache/logs/ssl_error.log
TransferLog bfinstall/server/apache/logs/transfer.log
</VirtualHost>
```

*bfinstall*은 설치 디렉토리의 루트입니다. 위 단계에 대해 UNIX 및 Linux에서는 올바른 디렉토리에 도달하려면 종종 /Platform을 *bfinstall*에 추가해야 합니다. 또한 디렉토리를 구분하도록 백슬래시(\) 대신 슬래시(/)를 사용해야 합니다.

## 결과

자세한 정보는 SSL 설정에 대한 IHS 문서를 참조하십시오.

## Windows: DLL 대체

### 이 태스크 정보

IBM HTTP Server 8.0 이전 버전을 사용할 경우 DLL을 Rational Build Forge와 함께 제공된 DLL로 바꾸어야 합니다.

### 프로시저

IBM HTTP Server 설치 디렉토리의 여러 위치에서 기존 libeay32.dll 파일을 찾아서 바꾸십시오.

- <bfinstall>/openssl에 제공된 libeay32.dll을 사용하십시오.
- DLL의 모든 인스턴스를 검색하여 바꾸십시오. Windows 2003의 예제 위치:

```
<ihs_install>\Plugins\gsk7\gsk7_32\icc\osslib  
<ihs_install>\gsk7\icc\osslib
```

## 결과

DLL을 대체한 후 HTTP 서버를 정상적으로 시작할 수 있습니다. 대체하지 않은 경우 HTTP 서버가 시작되지 않습니다.

## AIX 시스템: 추가 설정

AIX 시스템에 필요한 추가 설정이 있습니다.

### 프로시저

- AIX 6 시스템.
  - <bfinstall>/Platform/his\_modules/libxml2.a를 /opt/IBM/HTTPServer/modules에 복사하십시오.
  - /opt/IBM/HTTPServer/modules를 LIBPATH에 추가하십시오.
- AIX 7 시스템.
  - <bfinstall>/Platform/his\_modules/libxml2.a 및 <bfinstall>/Platform/his\_modules/libxml2.so.2를 /opt/IBM/HTTPServer/modules에 복사하십시오.
  - /opt/IBM/HTTPServer/modules를 LIBPATH에 추가하십시오.
  - 다음 행을 httpd.conf에 추가하십시오.  
LoadFile "/opt/IBM/HTTPServer/modules/libxml2.so.2"



---

## 제 37 장 API에 대한 작업

Build Forge는 Java 클라이언트 API 및 Perl 클라이언트 API를 제공합니다.

클라이언트 파일은 `<binstall>/webroot/public/clients/`에 저장됩니다.

실행 중인 관리 콘솔의 파일에 액세스할 수 있습니다. 클라이언트 다운로드 디렉토리는 다음 URL에 있습니다. `http://<hostname>:<portnumber>/clients/`

### Build Forge에 대한 API 액세스

API를 사용하는 프로그램은 직접 서비스 계층과 통신합니다. 서비스 계층은 Apache Tomcat상의 애플리케이션입니다. 설치 중에 Apache Tomcat 서버는 특정 포트에서 청취하도록 구성됩니다. API가 Build Forge와 통신하기 위해서는 해당 포트가 열려 있어야 합니다. 기본적으로 포트는 다음과 같이 설정됩니다.

- 3966(비보안)
- 49150(보안 - SSL 사용)

API 클라이언트에 대한 SSL 구성은 `bfclient.conf`에 있으며 클라이언트에 포함되어 있어야 합니다. SSL이 사용되는 경우, 클라이언트는 Build Forge와 통신하기 위해 키 저장소 및 인증서를 가지고 있어야 합니다. 자세한 정보는 130 페이지의 『API 클라이언트에 적합한 SSL 사용(Perl 또는 Java)』의 내용을 참조하십시오.

---

## API 프로그램용 Build Forge 사용자 작성

인증에 사용할 프로그램의 사용자를 관리 콘솔에서 작성하십시오.

### 이 태스크 정보

관리 콘솔에 로그인하는 데 사용할 API 프로그램의 사용자를 작성하십시오. 테스트 로그인을 통해 이 사용자가 작동하는지 확인하십시오.

프로그램이 콘솔에 액세스할 때마다 사용자 신임 정보로 콘솔에 자체 인증을 해야 합니다.

인증 후, 사용자 세션에 대해 세션 ID가 생성되어 데이터베이스에 저장됩니다. 프로그램이 기존 사용자와 동일한 로그인을 사용하는 경우 사용자 세션이 닫힙니다.

단일 스레드 또는 프로세스만 사용자 신임 정보를 사용할 수 있습니다. 다른 스레드 또는 프로세스가 동일한 사용자 신임 정보를 사용하여 세션을 설정하려고 시도하면 원래 세션이 종료됩니다.

참고: LDAP/Active Directory 인증이 제공하는 사용자를 사용하지 마십시오. 관리 콘솔에서 사용자를 작성하십시오.

---

## Java 클라이언트 API

Java 클라이언트 API를 사용하여 관리 콘솔에 액세스하는 Java 프로그램을 기록하십시오.

Java 클라이언트 API를 사용하여 작성된 프로그램은 클라이언트 호스트에서 실행되고 관리 콘솔의 데이터에 액세스합니다. Java 클라이언트 API는 해당 오브젝트에 대한 오퍼레이션을 제공하는 관리 콘솔 오브젝트 메소드를 정의하는 클래스가 들어 있는 .jar 파일로 구성됩니다.

Java 클라이언트 API와 함께 사용하려면 Java SDK 1.5 또는 1.6이 필요합니다.

문서는 JavaDoc로 제공됩니다.

참고: 인증하는 데 사용할 프로그램의 경우 관리 콘솔에 Build Forge 사용자를 정의해야 합니다.

## Java 클라이언트 API 패키지 가져오기

관리 콘솔 호스트에서 Java 클라이언트 소프트웨어 패키지를 다운로드할 수 있습니다.

### 프로시저

Java API를 다운로드하려면 다음을 수행하십시오.

1. 클라이언트 다운로드 디렉토리에 액세스하십시오. 웹 브라우저에서 다음 URL에 액세스하십시오.  
`http://<hostname>:<portnumber>/clients/`
2. JAR 파일을 저장하십시오. Java 클라이언트에서 **JAR** 파일 링크를 마우스 오른쪽 단추로 클릭하고 **다른 이름으로 링크 저장**을 선택하십시오. JAR 파일을 저장할 위치를 지정하십시오.
3. JavaDoc을 저장하십시오. Java 클라이언트에서 **JavaDoc** 참조 **ZIP** 링크를 마우스 오른쪽 단추로 클릭하고 **다른 이름으로 링크 저장**을 선택하십시오. JAR 파일을 저장할 위치를 지정하십시오.

### 결과

관리 콘솔을 통해 문서에 액세스할 수 있습니다. Java 클라이언트의 클라이언트 다운로드 디렉토리 페이지에서 **JavaDoc** 참조를 클릭하십시오.

## Java 클라이언트 API 설정

클라이언트 호스트에 Java API를 배치하고 이를 사용하도록 Java SDK를 설정합니다.

## 이 태스크 정보

호스트는 관리 콘솔 호스트에 대한 클라이언트로서 동작합니다.

### 프로시저

1. 원하는 곳에 .jar 파일을 배치하십시오.
2. CLASSPATH를 업데이트하십시오. rbf-services-client-java.jar을 배치한 디렉토리를 포함하도록 CLASSPATH를 설정하십시오.

---

## Perl 클라이언트 API

Perl 클라이언트 API를 사용하여 관리 콘솔에 액세스하는 Perl 프로그램을 기록하십시오.

Perl 클라이언트는 관리 콘솔 데이터 오브젝트 및 메소드의 추상에 대한 액세스를 제공하는 Perl 모듈 세트입니다.

Perl 클라이언트 모듈의 문서는 클라이언트 API 패키지에 두 가지 양식으로 포함되어 있습니다.

- 파일: apidoc.txt
- POD(Plain Old Documentation) 형식의 Perl 문서입니다. 자세한 정보는 <http://www.perl.org> 웹 사이트의 온라인 문서를 참조하십시오.

Perl 클라이언트를 사용하려면 다음을 수행해야 합니다.

- 관리 콘솔 컴퓨터에서 Perl 클라이언트 패키지를 가져오십시오.
- 패키지를 설치하십시오(Perl을 아직 설치하지 않은 경우 함께 설치).

참고: 인증하는 데 사용할 프로그램의 경우 관리 콘솔에 Build Forge 사용자를 정의해야 합니다.

## Perl 클라이언트 API 패키지 가져오기

관리 콘솔 호스트에서 Perl 클라이언트 API를 다운로드할 수 있습니다.

### 프로시저

Perl 클라이언트 API를 다운로드하려면 다음을 수행하십시오.

1. 클라이언트 다운로드 디렉토리에 액세스하십시오. 웹 브라우저에서 다음 URL에 액세스하십시오.  
`http://<hostname>:<portnumber>/clients/`
2. ZIP 파일을 저장하십시오. Perl 클라이언트에서 **ZIP** 파일 링크를 마우스 오른쪽 단추로 클릭하고 다른 이름으로 링크 저장을 선택하십시오. ZIP 파일을 저장할 위치를 지정하십시오.

3. 문서를 저장하십시오. Perl 클라이언트에서 **PerlDoc** 참조 **tar.gz** 링크를 마우스 오른쪽 단추로 클릭하고 **다른 이름으로 링크 저장**을 선택하십시오. ZIP 파일을 저장할 위치를 지정하십시오. 각 모듈의 문서에 액세스할 수 있도록 파일의 압축을 해제하십시오.

## 결과

관리 콘솔을 통해 문서에 액세스할 수 있습니다. Perl 클라이언트의 클라이언트 다운로드 디렉토리 페이지에서 **PerlDoc** 참조를 클릭하십시오.

## Perl 클라이언트 API 설정

Perl 클라이언트 API를 사용하려면 애플리케이션을 실행하려는 호스트에 소프트웨어를 설치해야 합니다.

## 이 태스크 정보

호스트는 관리 콘솔 호스트에 대한 클라이언트로서 동작합니다.

## 프로시저

1. 클라이언트 호스트에 Perl 해석기(예: ActiveState의 ActivePerl 버전 5.8.4 이상)를 설치하십시오. 다음 Perl 전제조건 모듈이 필요합니다(ActivePerl 버전 5.8.8이 이를 포함).

- Exporter
- LWP::UserAgent
- HTTP::Request

Perl 모듈 설치에 대한 정보는 Perl 문서를 참조하십시오.

2. 다운로드한 Perl 클라이언트 API 패키지를 임시 디렉토리에 압축 해제하십시오.
3. apidoc.txt 파일에 설명된 대로 Perl 클라이언트 API를 표준 Perl 분배로 설치하십시오.

Windows에서는 nmake 1.5(Visual Studio에 포함되어 있거나 Microsoft 웹 사이트에서 다운로드 가능)가 필요합니다. PATH 환경 변수로 찾을 수 있는 위치(예: C:\Windows)에 nmake 1.5를 설치해야 합니다. Perl 클라이언트 패키지의 압축을 해제한 임시 디렉토리에서 다음 명령을 실행하십시오.

```
perl Makefile.PL
nmake
nmake install
```

UNIX나 Linux 시스템(또는 Windows의 Cygwin 환경)에서는 다음 명령을 실행하십시오.



```
perl Makefile.PL  
make  
make install
```

설치 후 최상위 Perl 클라이언트 모듈은 BuildForge::Services::DB0입니다. 각 모듈에 대한 자세한 정보는 PerlDoc를 참조하십시오.



---

## 제 38 장 관리 콘솔 버전 번호 판별

작업 중인 관리 콘솔의 버전을 알려면 페이지의 왼쪽 상단 모서리에 있는 Rational Build Forge 로고 위에 마우스 커서를 위치하십시오. 시스템은 팝업 도구 팁에 버전 번호를 표시합니다.



---

## 제 39 장 성능 및 확장성

시스템은 로드의 범위가 광범위할 경우에도 정상적으로 작동하도록 설계되어 있습니다. 일부 구성과 자원 할당 작업을 수행할 때 높은 로드로 확장할 수 있기 때문에 시스템에 반드시 필요합니다.

다음 문서는 추가 자원으로 사용될 수 있습니다. 31 페이지의 제 5 장 『설치 계획』의 내용을 참조하십시오.

- IBM Rational Build Forge 관리 콘솔 배치: 결과 및 성능 가져오기: 이 문서에서는 계획 및 배치 단계에 대해 설명하며, 그에 따른 자원 성능 및 튜닝과 관련한 권장사항을 제공합니다.
- Rational Build Forge 성능 테스트 결과: 7.1.1.4와 비교하여 7.1.2에서 향상된 성능 평가: 이 문서에서는 로드가 증가하는 상황에서도 다양한 시스템 기능을 수행하는 방법에 대한 일반적인 관점을 제공하며, 운영 체제 및 데이터베이스의 다양한 조합에 대한 정보를 제공합니다. 또한 버전 7.1.1.4와 비교하여 버전 7.1.2가 제공하는 성능 개선사항에 대해서도 설명합니다.



## 제 40 장 제품과 함께 설치된 실행 가능 명령

아래 표는 Rational® Build Forge®가 제공하고 사용하는 실행 가능 명령에 대해 설명합니다.

Windows의 경우, 명령 파일은 Build Forge 설치 디렉토리에 있으며, 이 디렉토리는 기본적으로 C:\Program Files\IBM\Build Forge입니다.

UNIX/Linux의 경우, 명령 파일은 <bfinstall>/Platform 디렉토리에 있으며 여기에서 <bfinstall>은 기본적으로 /opt/buildforge입니다.

참고: z/Linux에서 관리 콘솔을 실행하는 경우 .pl extension을 지정하여 명령을 실행해야 합니다.

모든 실행 가능 명령에 대한 버전 번호를 표시하려면 -v 옵션을 사용하십시오. 실행 가능 명령이 설치되어 있는 디렉토리에서 명령을 실행해야 합니다.

bfproject -v

다음 예제에 표시된 대로 명령에 대한 -v 옵션은 명령 이름과 해당 버전 번호를 표시합니다.

bfproject.exe 7.0.351

실행 파일	서비스	설명
bfdbmigrate	N	스키마가 버전 7.0으로 변환된 후 기본 데이터베이스를 버전 3.8(MySQL)에서 DB2®로 변환하려면 이 마이그레이션 스크립트를 사용하십시오. 이 스크립트는 bfmigrate.exe/bfmigrate.pl을 사용한 후에 사용됩니다.
bfproject	N	buildforge.exe는 이 명령을 실행하여 작업을 시작합니다.
bfengine	Y	이 명령은 buildforge.exe 및 웹 서버를 시작합니다. Windows 전용
bfexport	N	이 유틸리티를 사용하여 데이터베이스에서 데이터를 내보냅니다.
bf bomexport	N	데이터베이스에서 BOM을 내보내려면 이 유틸리티를 사용하십시오.
bfimport	N	이 유틸리티를 사용하여 프로젝트 데이터를 데이터베이스로 가져옵니다.
bfmigrate	N	이 마이그레이션 스크립트를 사용하여 버전 3.8 데이터베이스를 버전 7.0으로 업그레이드합니다.
bfpurge	N	buildforge.exe는 이 명령을 실행하여 빌드를 제거합니다.

실행 파일	서비스	설명
bfrefresh	N	빌드 시스템에서 이 명령을 사용하여 서버의 Manifest를 업데이트합니다.
bfsched.exe 또는 bfsched.pl	N	buildforge.exe는 이 명령을 사용하여 스케줄된 작업에 대한 데이터베이스를 확인하고 해당되는 경우 작업을 실행합니다.
bfstepcmd	N	장기 실행 단계의 경우 bfproject는 이 명령을 사용하여 해당 단계의 개별 프로젝트를 작성합니다.
buildforge	N	이 명령은 빌드, 제거 및 스케줄 프로세스를 관리합니다.
console_uninst	N	관리 콘솔을 설치 제거하려면 이 명령을 사용하십시오. Windows 전용
bfdispatch	Y	이 명령은 에이전트 서비스를 시작합니다. Windows 전용
bfagent	N	에이전트 실행 파일
bfpwncrypt	N	비밀번호를 암호화하는 유틸리티입니다.



---

## 제 41 장 용어집

이 주제에서는 시스템 전체에서 사용되는 개념과 용어의 정의를 제공합니다.

---

### 액세스 그룹(access group)

권한, 알림 및 LDAP 그룹 특성을 공유하는 사용자 콜렉션입니다. 액세스 그룹을 LDAP 그룹에 맵핑할 수 있습니다. 그룹을 중첩할 수 있습니다. 사용자는 현재 구성원으로 속해 있는 그룹의 권한을 상속합니다.

---

### 어댑터(adaptor)

어댑터는 Build Forge 시스템이 외부 시스템(예: 소스 제어 시스템, 디버깅 데이터베이스 또는 테스트 시스템)과 상호작용할 수 있게 하는 추가 기능입니다. 예를 들어, 소스 코드 어댑터를 사용하여 시스템이 소스 제어 시스템(예: IBM® Rational® ClearCase®, Perforce, Visual SourceSafe 및 CVS)의 변경사항을 모니터 및 추적하고 해당 변경사항을 기반으로 조치를 수행할 수 있습니다. 명세서(BOM)에 스토리지 정보를 수집하거나 다른 정보 시스템에 정보를 다시 푸시하도록 어댑터를 구성할 수 있습니다.

---

### 에이전트(agent)

Build Forge® 시스템의 컴포넌트입니다. 에이전트는 시스템에서 서버 자원으로 정의할 컴퓨터에 설치해야 합니다. 각 에이전트는 관리 콘솔과 통신하여 단계에 정의된 명령을 실행하며 또한 단계 실행의 결과 출력을 모아 단계 로그에 리턴합니다.

---

### 아카이브(archive)

출력 파일은 삭제되었지만 데이터베이스에 계속 데이터가 있는 작업의 목록입니다. 이 목록은 작업 패널에서 볼 수 있습니다.

---

### BOM

완료된 작업에 대한 데이터의 목록입니다. BOM은 Bill of Materials의 약자입니다. 작업을 볼 때 BOM 탭에 표시되는 반면 개별 단계 실행에 대한 데이터는 단계 탭에 표시됩니다. BOM에는 작업의 단계에 대한 정보와 작업 결과로 발생한 파일 변경사항이 포함됩니다. 일반적인 사용법의 예는 소프트웨어 빌드의 소스 코드 어댑터에 대해 사용하는 경우이며 이 경우 소스 파일 변경사항을 감사해야 합니다. .scan 명령을 사용하면 소스 변경사항의 기준선을 설정한 다음 마지막 .scan 명령 이후 변경사항을 요약하는 체크포인트를 설정할 수 있습니다.

---

## 클래스(class)

글로벌 특성을 갖는 프로젝트 그룹입니다. 특성은 일반적으로 완료된 작업을 정기적으로 삭제하거나 특정 정리 태스크를 수행하는 다른 작업을 시작하여 완료된 작업을 관리하는 데 사용됩니다.

---

## 정리(clobber)

프로젝트 및 연관된 모든 작업을 데이터베이스에서 삭제합니다.

---

## 컬렉터(collector)

서버 자원에서 수집되거나 서버 자원에 지정되는 정보를 결정하는 오브젝트입니다. 해당 정보는 컬렉터에서 특성을 통해 지정됩니다. 서버에 지정된 컬렉터는 서버 Manifest의 스펙 기능을 수행합니다. 서버 > 컬렉터 패널에서 컬렉터를 정의합니다.

---

## 데이터베이스(database)

데이터베이스는 관리 콘솔에 입력된 모든 정보를 저장합니다. 또한 데이터베이스는 프로젝트를 실행하거나 사용자 조치를 로그하는 경우 시스템에서 작성하는 데이터를 저장합니다.

---

## 동적(dynamic)

실행 시 또는 처리하는 동안 발생하는 이벤트와 관련됩니다.

---

## 엔진(engine)

시스템의 컴포넌트입니다. 엔진은 관리 콘솔에서 입력되고 데이터베이스에 저장된 정보를 사용하여 프로젝트 실행을 제어하고 알림 이메일을 보내며 서버에서 실행 중인 에이전트와 통신합니다.

---

## 환경(environments)

환경은 변수 목록의 컨테이너입니다. 환경은 서버, 프로젝트 및 단계에 명시적으로 지정될 수 있습니다. 단계 환경은 서버 환경, 프로젝트 환경 및 단계 환경을 적용하여 이 순서대로 구성됩니다. 해당 환경 중 여러 환경에서 변수가 나타나는 경우 지정된 마지막 값을 갖습니다.

---

## 데이터 교환(handshake)

클라이언트가 공용 키 기술을 사용하여 서버를 인증(또는 선택적으로 서버가 클라이언트를 인증)하도록 허용한 후 클라이언트와 서버가 변경의 암호화, 암호 해독 및 발견을 위한 대칭 키 작성에 협력하도록 허용하는 SSL(Secure Sockets Layer) 세션의 시작 시 메시지 교환입니다.

---

## 인터셉터(interceptor)

수신 메시지를 인증하기 위해 웹 서비스가 사용하는 핸들러입니다. Build Forge에서 인터셉터는 단일 사인온을 구현하기 위해 제공됩니다.

---

## 인터페이스(interface)

인터페이스는 어댑터 템플릿의 인스턴스입니다. 어댑터를 사용하도록 인터페이스를 작성(및 편집)해야 합니다. 원래 어댑터 템플릿은 변경되지 않습니다. 또한 인터페이스는 두 개 이상의 <interface> 요소(각각은 별도로 실행 가능한 조치임)를 포함할 수 있음에 유의하십시오.

---

## 작업(job)

실행 중인 프로젝트의 인스턴스입니다. 시스템은 단계 로그 및 BOM 데이터 등 완료된 각 작업에 대한 데이터를 저장합니다.

---

## 라이브러리(library)

라이브러리는 작업에 대한 실행 가능 정의로서 단계로 구성됩니다. 해당 동작은 특성을 통해 제어되며 실행되는 서버를 판별하려는 경우 선택기가 없다는 점에서 프로젝트와 다릅니다. 라이브러리는 프로젝트 내의 단계에서 호출됩니다.

---

## LDAP(Lightweight Directory Access Protocol)

TCP/IP를 사용하여 X.500 모델을 지원하며 더 복잡한 X.500 DAP(Directory Access Protocol)의 자원 요구사항을 초래하지 않는 디렉토리에 대한 액세스를 제공하는 공개 프로토콜입니다. 예를 들어, LDAP은 인터넷 또는 인트라넷 디렉토리에서 사용자, 조직 및 기타 자원을 찾는 데 사용될 수 있습니다.

---

## Manifest

컬렉터가 수집한 서버에 대한 데이터의 목록입니다. Manifest 데이터는 선택기가 서버를 선택하기 위해 사용됩니다. 서버에 대한 Manifest는 자동으로 업데이트됩니다. Manifest는 수동으로 업데이트할 수도 있습니다. 서버 > *servername*에서 서버를 볼 때 큐 Manifest 새로 고치기 단추를 사용하십시오.

---

## 관리 콘솔(Management Console)

시스템을 조정하기 위해 단일 컴퓨터에 설치된 시스템의 컴포넌트입니다. 관리 콘솔에 로그인하여 프로젝트를 정의 및 실행하고 결과와 보고서를 봅니다. 관리 콘솔은 작업을 완료하기 위해 에이전트에 지시사항을 발행합니다.

---

## 알림 템플릿(notification templates)

알림 템플릿은 특정 이벤트가 발생할 때 액세스 그룹에 보내는 이메일의 형식과 콘텐츠를 정의합니다. 시스템에는 많은 기본 템플릿이 제공됩니다. 템플릿을 편집하거나 프로젝트 특정 템플릿을 새로 작성할 수 있습니다.

---

## 플러그인(plugin)

기존 프로그램, 애플리케이션 또는 인터페이스에 기능을 추가하는 독립적으로 설치 가능한 소프트웨어 모듈입니다.

---

## 프로젝트(project)

프로젝트는 작업에 대한 실행 가능 정의로서 단계로 구성됩니다. 해당 동작은 프로젝트 특성을 통해 제어됩니다. 프로젝트에는 프로젝트가 실행될 수 있는 하나 이상의 서버를 결정하는 연관된 선택기가 있습니다. 프로젝트에는 고유한 환경이 지정될 수 있습니다. 실행 프로젝트가 작업이며 선택기가 지정되지 않은 프로젝트는 라이브러리입니다.

---

## 선택기(selector)

프로젝트 또는 단계가 실행되는 서버를 선택하는 프로젝트 또는 단계와 연관되는 오브젝트입니다. 선택기의 특성은 서버 선택 방법을 결정합니다. 선택기는 정적 정보를 사용할 수 있습니다. 예를 들어, 선택기는 서버 이름을 지정할 수 있습니다. 또한 서버는 동적 정보를 사용할 수 있습니다. 예를 들어, 선택기는 특성(예: CPU 유형 또는 디스크 크기 또는 현재 로드)을 지정한 서버를 지정할 수 있습니다. 런타임에 시스템은 선택기를 사용하여 일치하는 서버 목록을 컴파일하고 해당 서버 중 하나에 프로젝트 또는 단계를 지정합니다. 관리 > 선택기 패널에서 선택기를 정의합니다. 프로젝트를 정의하려면 하나 이상의 선택기를 정의해야 합니다.

---

## 세마포어(semaphore)

시스템에서 활동이 동시에 발생하지 않도록 방지하는 글로벌 플래그입니다. 각 세마포어는 시스템이 관리하는 레이블입니다. 일반적으로 자원의 독점 사용이 요구되는 프로젝트 또는 단계는 해당 독점 사용을 보장하도록 세마포어를 확보합니다.

**.semget** 명령을 사용하여 단계에서 세마포어를 설정합니다. 세마포어는 **.semput** 명령에 의해 별도 단계에서 해제됩니다. 세마포어를 얻으면 다른 단계에서 가져올 수 없습니다. 세마포어를 얻으려는 단계는 해제될 때까지 대기합니다.

프로젝트가 완료되면 프로젝트가 사용한 세마포어가 시스템에서 자동으로 해제됩니다. 때때로 예를 들어, 시스템 오류로 인해 작업이 종료되는 경우 세마포어가 해제되지 않습니다. 이러한 경우 수동으로 해제할 수 있습니다.

---

## 서버(server)

Build Forge에서 서버는 호스트와 연관된 오브젝트입니다. 서버는 서버 자원이라고도 합니다. 호스트에서는 프로젝트 또는 단계가 실행됩니다. 사용할 서버는 프로젝트 또는 단계와 연관된 선택기로 정의됩니다.

Build Forge에서 서버로 사용 가능할 컴퓨터를 설정하려면 다음을 수행해야 합니다.

- 컴퓨터에 에이전트를 설치합니다(자세한 정보는 161 페이지의 제 10 장 『에이전트 설치』 참조).
- 관리 콘솔을 사용하여 서버 자원 작성.

서버 자원은 서버 패널에서 정의합니다.

---

## 서비스(services)

시스템 컴포넌트이며 클라이언트와 데이터베이스 간의 추상 계층 기능을 수행하므로 서비스 계층이라고도 합니다. 클라이언트에는 시스템 자체와 함께 제공된 Java API 또는 Perl API로 구성되는 클라이언트도 포함됩니다.

---

## 스냅샷(snapshot)

특정 시간의 백업 데이터 레코드입니다.

---

## 정적(static)

사전 정의된 시간 또는 고정된 시간에 발생하는 오퍼레이션과 관련됩니다.

---

## 단계(step)

단계는 프로젝트 또는 라이브러리의 컴포넌트이며 실행할 하나 이상의 명령행을 포함합니다. 단계와 연관된 선택기는 사용할 서버를 결정합니다. 선택기가 지정되지 않으면 프로젝트의 선택기가 사용됩니다. 단계 특성은 단계 실행 방법과 출력 처리 방법을 결정합니다. 프로젝트 또는 라이브러리를 작성하거나 편집할 때 단계를 정의합니다.

---

## 단계 로그(step log)

완료된 작업 내에서 완료된 단계에 대한 데이터의 목록입니다. 작업을 볼 때 단계 탭에 단계 로그가 표시됩니다. 각 단계에 대한 정보는 열에 표시됩니다. 작업 > 작업 이름을 선택하면 단계 목록이 표시됩니다. 단계를 클릭하여 단계 로그를 표시하십시오.

---

## 스레딩(threading)

다양한 관련 트랜잭션을 동시에 실행하는 방법입니다.

---

## 사용자(user)

시스템의 로그인입니다. 시스템은 자체 사용자 및 권한 설정 세트를 유지보수합니다. 프로덕션 설치에서 LDAP는 사용자 관리에 사용되고 LDAP 항목 및 그룹은 시스템에 맵핑됩니다. 사용자는 시스템 자원에 대한 특정 액세스 권한을 부여하는 액세스 그룹과 연관되어 있습니다.

---

## 부록. 주의사항

이 정보는 미국에서 제공되는 제품 및 서비스용으로 작성된 것입니다. IBM은 다른 국가에서 이 책에 기술된 제품, 서비스 또는 기능을 제공하지 않을 수도 있습니다. 현재 사용할 수 있는 제품 및 서비스에 대한 정보는 한국 IBM 담당자에게 문의하십시오. 이 책에서 IBM 제품, 프로그램 또는 서비스를 언급했다고 해서 해당 IBM 제품, 프로그램 또는 서비스만을 사용할 수 있다는 것을 의미하지는 않습니다. IBM의 지적 재산권을 침해하지 않는 한, 기능상으로 동등한 제품, 프로그램 또는 서비스를 대신 사용할 수도 있습니다. 그러나 비IBM 제품, 프로그램 또는 서비스의 운영에 대한 평가 및 검증은 사용자의 책임입니다.

IBM은 이 책에서 다루고 있는 특정 내용에 대해 특허를 보유하고 있거나 현재 특허 출원 중일 수 있습니다. 이 책을 제공한다고 해서 특허에 대한 라이선스까지 부여하는 것은 아닙니다. 라이선스에 대한 의문사항은 다음으로 문의하십시오.

135-700

서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩

한국 아이.비.엠 주식회사

고객만족센터

전화번호: 080-023-8080

2바이트(DBCS) 정보에 관한 라이선스 문의는 한국 IBM 고객만족센터에 문의하거나 다음 주소로 서면 문의하시기 바랍니다.

Intellectual Property Licensing

Legal and Intellectual Property Law

IBM Japan, Ltd.

1623-14, Shimotsuruma, Yamoto-shi

Kanagawa, 242-8502 Japan

다음 단락은 현지법과 상충하는 영국이나 기타 국가에서는 적용되지 않습니다. IBM은 타인의 권리 비침해, 상품성 및 특정 목적에의 적합성에 대한 묵시적 보증을 포함하여 (단, 이에 한하지 않음) 묵시적이든 명시적이든 어떠한 종류의 보증 없이 이 책을 "현 상태대로" 제공합니다. 일부 국가에서는 특정 거래에서 명시적 또는 묵시적 보증의 면책사항을 허용하지 않으므로, 이 사항이 적용되지 않을 수도 있습니다.

이 정보에는 기술적으로 부정확한 내용이나 인쇄상의 오류가 있을 수 있습니다. 이 정보는 주기적으로 변경되며, 변경된 사항은 최신판에 통합됩니다. IBM은 이 책에서 설명한 제품 및/또는 프로그램을 사전 통지 없이 언제든지 개선 및/또는 변경할 수 있습니다.

이 정보에서 언급되는 비IBM의 웹 사이트는 단지 편의상 제공된 것으로, 어떤 방식으로든 이들 웹 사이트를 옹호하고자 하는 것은 아닙니다. 해당 웹 사이트의 자료는 본 IBM 제품 자료의 일부가 아니므로 해당 웹 사이트 사용으로 인한 위험은 사용자 본인이 감수해야 합니다.

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

(i) 독립적으로 작성된 프로그램과 기타 프로그램(본 프로그램 포함)간의 정보 교환 및  
(ii) 교환된 정보의 상호 이용을 목적으로 본 프로그램에 관한 정보를 얻고자 하는 라이선스 사용자는 다음 주소로 문의하십시오.

135-700

서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩

한국 아이.비.엠 주식회사

고객만족센터

이러한 정보는 해당 조건(예를 들면, 사용료 지불 등)하에서 사용될 수 있습니다.

본 문서에 기술된 라이선스가 있는 프로그램 및 사용 가능한 모든 라이선스가 있는 자료는 IBM이 IBM 기본 계약, IBM 프로그램 라이선스 계약(IPLA) 또는 이와 동등한 계약에 따라 제공한 것입니다.

본 문서에 포함된 모든 성능 데이터는 제한된 환경에서 산출된 것입니다. 따라서 다른 운영 환경에서 얻어진 결과는 상당히 다를 수 있습니다. 일부 성능은 개발 단계의 시스템에서 측정되었을 수 있으므로 이러한 측정치가 일반적으로 사용되고 있는 시스템에서도 동일하게 나타날 것이라고는 보증할 수 없습니다. 또한 일부 성능은 추정을 통해 추측되었을 수도 있으므로 실제 결과는 다를 수 있습니다. 이 책의 사용자는 해당 데이터를 본인의 특정 환경에서 검증해야 합니다.

비IBM 제품에 관한 정보는 해당 제품의 공급업체, 공개 자료 또는 기타 범용 소스에서부터 얻은 것입니다. IBM에서는 이러한 제품들을 테스트하지 않았으므로, 비IBM 제품과 관련된 성능의 정확성, 호환성 또는 기타 청구에 대해서는 확신할 수 없습니다. 비IBM 제품의 성능에 대한 의문사항은 해당 제품의 공급업체에 문의하십시오.

IBM이 제시하는 방향 또는 의도에 관한 모든 언급은 특별한 통지 없이 변경될 수 있습니다.

이 정보는 계획 수립 목적으로만 사용됩니다. 이 정보는 기술된 제품이 GA(General Availability)되기 전에 변경될 수 있습니다.

이 정보에는 일상의 비즈니스 운영에서 사용되는 자료 및 보고서에 대한 예제가 들어 있습니다. 이들 예제에는 개념을 가능한 완벽하게 설명하기 위하여 개인, 회사, 상표 및



제품의 이름이 사용될 수 있습니다. 이들 이름은 모두 가공의 것이며 실제 기업의 이름 및 주소와 유사하더라도 이는 전적으로 우연입니다.

저작권 라이선스:

이 정보에는 여러 운영 플랫폼에서의 프로그래밍 기법을 보여주는 원어로 된 샘플 응용프로그램이 들어 있습니다. 귀하는 이러한 샘플 프로그램의 작성 기준이 된 운영 플랫폼의 응용프로그램 프로그래밍 인터페이스(API)에 부합하는 응용프로그램을 개발, 사용, 판매 또는 배포할 목적으로 추가 비용 없이 이들 샘플 프로그램을 어떠한 형태로든 복사, 수정 및 배포할 수 있습니다. 이러한 샘플 프로그램은 모든 조건하에서 완전히 테스트된 것은 아닙니다. 따라서 IBM은 이들 샘플 프로그램의 신뢰성, 서비스 가능성 또는 기능을 보증하거나 진술하지 않습니다.

이러한 샘플 프로그램 또는 파생 제품의 각 사본이나 그 일부에는 반드시 다음과 같은 저작권 표시가 포함되어야 합니다.

© Copyright IBM Corp. 2003, 2013.

이 정보를 소프트웨어로 확인하는 경우에는 사진과 컬러 삽화가 제대로 나타나지 않을 수도 있습니다.

---

## 상표

IBM, IBM 로고 및 [ibm.com](http://ibm.com)은 전세계 많은 관할 구역에서 등록된 IBM Corp.의 상표 또는 등록상표입니다. 기타 제품 및 서비스 이름은 IBM 또는 타사의 상표입니다. 현재 IBM 상표 목록은 웹([www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml))에 있습니다.

Adobe, Adobe 로고, PostScript 및 PostScript 로고는 미국 또는 기타 국가에서 사용되는 Adobe Systems Incorporated의 등록상표 또는 상표입니다.

Intel은 미국 및 기타 국가에서 사용되는 Intel Corporation 또는 자회사의 상표 또는 등록상표입니다.

Windows는 미국 또는 기타 국가에서 사용되는 Microsoft Corporation의 등록상표입니다.

Java 및 모든 Java 기반 상표는 미국 또는 기타 국가에서 사용되는 Sun Microsystems, Inc.의 상표입니다.

Linux는 미국 또는 기타 국가에서 사용되는 Linus Torvalds의 등록상표입니다.

UNIX는 미국 및 기타 국가에서 사용되는 The Open Group의 등록상표입니다.

기타 회사, 제품 또는 서비스 이름은 타사의 상표 또는 서비스표입니다.