

# Reference 안내서

JEUS 9.1

**TMAXSOFT**

## 저작권 공지

Copyright 2026. TmaxSoft Co., Ltd. All Rights Reserved.

## 회사 정보

(주)티맥스소프트

주소 : 경기도 성남시 분당구 정자일로 45, 티맥스소프트타워

기술 서비스 센터: 1544-8629

홈페이지: <https://www.tmaxsoft.com>

## 제한된 권리

이 소프트웨어(JEUS®) 사용설명서와 프로그램은 저작권법과 국제 조약에 의해 보호됩니다. 사용설명서와 프로그램은 TmaxSoft Co., Ltd.와의 사용권 계약 하에서만 사용할 수 있으며, 사용설명서는 사용권 계약의 범위 내에서만 배포 또는 복제할 수 있습니다. 이 사용설명서의 전부 또는 일부를 TmaxSoft의 사전 서면 동의 없이 전자, 기계, 녹음 등의 수단으로 전송, 복제, 배포하거나 2차적 저작물을 작성할 수 없습니다.

이 소프트웨어 사용설명서와 프로그램의 사용권 계약은 어떠한 경우에도 사용설명서 및 프로그램과 관련된 지적 재산권(등록 여부를 불문)을 양도하는 것으로 해석되지 않으며, 브랜드나 로고, 상표 등을 사용할 권한을 부여하지 않습니다. 사용설명서는 오로지 정보 제공만을 목적으로 하며, 이로 인한 계약상의 직접적 또는 간접적 책임을 지지 않습니다. 또한 사용설명서 상의 내용이 법적 또는 상업적인 특정 조건을 만족시킬 것을 보장하지 않습니다. 사용설명서는 제품의 업그레이드나 수정에 따라 예고 없이 변경될 수 있으며, 내용상의 오류가 없음을 보장하지 않습니다.

## 상표 공지

JEUS®는 TmaxSoft Co., Ltd.의 등록 상표입니다.

Java, Solaris는 Oracle Corporation 및 그 자회사, 관계회사의 등록 상표입니다.

Microsoft, Windows, Windows NT는 Microsoft Corporation의 등록 상표 또는 상표입니다.

HP-UX는 Hewlett Packard Enterprise Company의 등록 상표입니다.

AIX는 International Business Machines Corporation의 등록 상표입니다.

UNIX는 X/Open Company, Ltd.의 등록 상표입니다.

Linux는 Linus Torvalds의 등록 상표입니다.

Noto는 Google Inc.의 상표입니다. Noto 글꼴은 오픈 소스입니다. 모든 Noto 글꼴은 SIL Open Font License, 버전 1.1에 따라 게시됩니다. (<https://www.google.com/get/noto/>)

기타 모든 제품들과 회사 이름은 각각 해당 소유주의 상호, 상표, 또는 등록 상표입니다.

본 사용설명서에 기재된 회사, 시스템, 제품 이름 등에 반드시 상표 표시(™, ®)를 하지는 않습니다.

## 오픈 소스 소프트웨어 공지

본 제품의 일부 파일 또는 모듈은 다음의 라이선스를 준수합니다.: APACHE2.0, CDDL1.0, EDL1.0, OPEN SYMPHONY SOFTWARE1.1, TRILEAD-SSH2, Bouncy Castle, BSD, MIT, SIL OPEN FONT1.1

관련 상세 정보는 제품의 다음 디렉터리에 기재된 사항을 참고하시기 바랍니다. :

`${INSTALL_PATH}/license/oss_licenses`

## 유지 보수

구분	지원항목	서비스 내용
제품지원	패치 & 업그레이드	무상 패치 서비스 제공  메이저 버전 업그레이드 시 할인 혜택  웹 지원을 통한 패치 내역 제공
기술 지원 - 기본 서비스	장애 지원	장애 발생 시 원인 분석 및 조치  Service Desk팀 → 기술팀 → R&D의 3단계 장애 분석 및 조치
	일상 지원(온라인 지원)	E-mail, 전화, 원격, 웹 사이트 등 온라인 자원을 통한 질의 응답 서비스
	고객 맞춤 지원(방문 지원)	고객의 요청으로 수행하는 방문 지원 서비스
기술 지원 - 옵션 서비스	예방 지원	정기 점검을 통한 시스템 운영현황 보고 및 장애 예방  ◦ 관리자 또는 운영자의 요구사항 수렴 ◦ 운영 현황(시스템, 엔진 운영) 보고서 제공 ◦ 필요 시 시스템 개선 권장 사항 보고
유지 보수 비용 및 기간	계약 시 별도 협의	계약 시 EOL/EOS 문서 제공

## 안내서 이력

제품 버전	안내서 버전	발행일	비고
JEUS 9.1.1	3.4.1	2026-06-08	Fix 릴리스
JEUS 9.1	3.3.1	2025-09-30	GA 버전
JEUS 9 Fix#1	3.2.2	2025-07-10	JDK 21 지원 내용 추가
JEUS 9 Fix#1	3.2.1	2025-06-30	RC 버전
JEUS 9	3.1.2	2025-01-03	-

제품 버전	안내서 버전	발행일	비고
JEUS 9	3.1.1	2024-09-27	-

# 목차

Part I. 시스템 프로퍼티	1
1. 시스템 프로퍼티	2
1.1. 개요	2
1.2. 서버 시스템 프로퍼티	2
1.3. 네트워크 시스템 프로퍼티	11
1.4. JNDI 시스템 프로퍼티	14
1.5. JDBC 시스템 프로퍼티	17
1.6. 트랜잭션 시스템 프로퍼티	19
1.6.1. TM 클라이언트를 위한 시스템 프로퍼티	22
1.6.2. 하위 호환성을 위한 시스템 프로퍼티	24
1.7. 웹 엔진 프로퍼티	24
1.7.1. 웹 엔진 레벨 프로퍼티	24
1.7.2. JVM 시스템 프로퍼티	27
1.7.3. 가상 호스트 레벨 프로퍼티	28
1.7.4. 웹 애플리케이션 레벨 프로퍼티	28
1.7.4.1. 공통 또는 Servlet 엔진 관련 프로퍼티	28
1.7.4.2. JSP 엔진 관련 프로퍼티	31
1.7.4.3. 호환성 제공 프로퍼티	35
1.8. 세션 서버 시스템 프로퍼티	39
1.9. EJB 시스템 프로퍼티	40
1.9.1. EJB 시스템 프로퍼티	40
1.9.2. JPA 시스템 프로퍼티	46
1.9.3. RMI 시스템 프로퍼티	46
1.10. 보안 시스템 프로퍼티	49
1.10.1. Standard Java SE & Jakarta™ EE 보안 시스템 프로퍼티	49
1.10.2. JEUS 보안 시스템 프로퍼티	50
1.11. JEUS MQ 시스템 프로퍼티	52
1.11.1. JEUS MQ 서버	52
1.11.2. JEUS MQ 클라이언트	53
1.12. Deployment 시스템 프로퍼티	55
1.13. Console 시스템 프로퍼티	57
Part II. 콘솔 명령어와 툴	58
2. 명령어 설명 규칙	59
3. JEUS 서버 기동 및 종료	60
3.1. 개요	60
3.2. 사용법	60
3.3. 마스터 패스워드 입력	64
3.4. 디버그 모드의 JEUS 서버 실행	64
4. 콘솔 툴	66

4.1. 개요	66
4.2. jeusadmin	66
4.2.1. 기동 및 종료 명령어	68
4.2.1.1. jeusadmin	68
4.2.1.2. exit	71
4.2.2. Local 명령어	72
4.2.2.1. connect	72
4.2.2.2. disconnect	73
4.2.2.3. help	74
4.2.2.4. history	78
4.2.2.5. p	79
4.2.2.6. remove-login-cache	79
4.2.2.7. set-server-option	80
4.2.2.8. set-servers-option	81
4.2.2.9. start-record	82
4.2.2.10. stop-record	83
4.2.2.11. unset-server-option	84
4.2.2.12. unset-servers-option	84
4.2.2.13. verbose	85
4.2.3. Server Management 관련 명령어	86
4.2.3.1. add-custom-resource-to-servers	88
4.2.3.2. add-data-sources-to-server	89
4.2.3.3. add-external-resource-to-servers	90
4.2.3.4. add-invocation	91
4.2.3.5. add-invocation-library	92
4.2.3.6. add-jvm-option	93
4.2.3.7. add-lifecycle-invocation	94
4.2.3.8. add-listener	96
4.2.3.9. add-log-handler	97
4.2.3.10. add-logger	98
4.2.3.11. config-listener-ssl	99
4.2.3.12. disable-engine-init-on-boot	101
4.2.3.13. disable-engines	102
4.2.3.14. dump	103
4.2.3.15. enable-engine-init-on-boot	104
4.2.3.16. enable-engines	105
4.2.3.17. jndi-info	106
4.2.3.18. lifecycle-invocation-info	108
4.2.3.19. list-blocked-clients	109
4.2.3.20. list-jvm-options	109
4.2.3.21. list-lifecycle-invocations	110

4.2.3.22. list-log-handlers	111
4.2.3.23. list-loggers	112
4.2.3.24. list-server-listeners	112
4.2.3.25. local-shutdown	113
4.2.3.26. local-start-server	114
4.2.3.27. log-level	115
4.2.3.28. mbean-info	116
4.2.3.29. memory-info	117
4.2.3.30. modify-invocation	118
4.2.3.31. modify-invocation-library	120
4.2.3.32. modify-jvm-option	121
4.2.3.33. modify-lifecycle-invocation	121
4.2.3.34. modify-listener	122
4.2.3.35. modify-log-handler	124
4.2.3.36. modify-logger	125
4.2.3.37. remove-custom-resource-from-servers	126
4.2.3.38. remove-data-sources-from-server	127
4.2.3.39. remove-external-resource-from-servers	127
4.2.3.40. remove-invocation	128
4.2.3.41. remove-invocation-library	129
4.2.3.42. remove-jvm-option	130
4.2.3.43. remove-lifecycle-invocation	130
4.2.3.44. remove-listener	131
4.2.3.45. remove-log-handler	132
4.2.3.46. remove-logger	133
4.2.3.47. remove-replicate-group	134
4.2.3.48. resume-server	134
4.2.3.49. run-garbage-collection	135
4.2.3.50. server-info	136
4.2.3.51. server-log	137
4.2.3.52. set-replicate-group	138
4.2.3.53. show-replicate-group	139
4.2.3.54. show-current-scf-view	140
4.2.3.55. start-cluster	142
4.2.3.56. start-domain	142
4.2.3.57. start-node	143
4.2.3.58. start-server	144
4.2.3.59. stop-cluster	144
4.2.3.60. stop-domain	145
4.2.3.61. stop-node	146
4.2.3.62. stop-server	147

4.2.3.63. suspend-server	147
4.2.3.64. system-info	148
4.2.3.65. unblock-client	149
4.2.4. Domain Configuration 관련 명령어	149
4.2.4.1. add-cluster	151
4.2.4.2. add-custom-resource	152
4.2.4.3. add-custom-resource-to-clusters	153
4.2.4.4. add-data-sources-to-cluster	154
4.2.4.5. add-external-resource	155
4.2.4.6. add-external-resource-to-clusters	156
4.2.4.7. add-server	156
4.2.4.8. add-servers-to-cluster	159
4.2.4.9. add-server-template	159
4.2.4.10. add-dynamic-servers	161
4.2.4.11. apply-configuration-plan	162
4.2.4.12. config-system-clustering-framework	163
4.2.4.13. create-domain	164
4.2.4.14. delete-domain	166
4.2.4.15. disable-json-command	167
4.2.4.16. disable-to-resynchronize-applications	168
4.2.4.17. enable-json-command	168
4.2.4.18. enable-to-resynchronize-applications	169
4.2.4.19. list-clusters	170
4.2.4.20. list-custom-resources	170
4.2.4.21. list-external-resources	171
4.2.4.22. list-domains	172
4.2.4.23. list-servers	172
4.2.4.24. modify-cluster	173
4.2.4.25. modify-server	174
4.2.4.26. modify-dynamic-servers	175
4.2.4.27. pack-domain	176
4.2.4.28. remove-cluster	177
4.2.4.29. remove-custom-resource	178
4.2.4.30. remove-custom-resource-from-clusters	179
4.2.4.31. remove-data-sources-from-cluster	179
4.2.4.32. remove-domain-log-config	180
4.2.4.33. remove-external-resource	181
4.2.4.34. remove-external-resource-from-clusters	181
4.2.4.35. remove-server	182
4.2.4.36. remove-servers-from-cluster	183
4.2.4.37. set-domain-backup	184



4.2.4.38. set-domain-log-config .....	185
4.2.4.39. show-domain-log-config .....	185
4.2.4.40. show-dynamic-servers .....	186
4.2.4.41. show-server-template .....	187
4.2.4.42. show-system-clustering-framework .....	187
4.2.4.43. unpack-domain .....	188
4.2.5. Thread Management 관련 명령어 .....	189
4.2.5.1. interrupt-thread .....	189
4.2.5.2. modify-service-thread-pool .....	190
4.2.5.3. modify-system-thread-pool .....	192
4.2.5.4. print-stack-trace .....	194
4.2.5.5. show-service-thread-pool .....	195
4.2.5.6. show-system-thread-pool .....	195
4.2.5.7. thread-info .....	196
4.2.6. Application 관련 명령어 .....	200
4.2.6.1. add-application-repository .....	201
4.2.6.2. add-application-target .....	202
4.2.6.3. application-info .....	202
4.2.6.4. deploy-application .....	205
4.2.6.5. deploy-library .....	209
4.2.6.6. deployment-plan-info .....	209
4.2.6.7. distribute-application .....	210
4.2.6.8. install-application .....	213
4.2.6.9. install-deployment-plan .....	214
4.2.6.10. install-library .....	214
4.2.6.11. library-info .....	215
4.2.6.12. list-application-repositories .....	216
4.2.6.13. redeploy-application .....	216
4.2.6.14. remove-application-repository .....	217
4.2.6.15. remove-application-target .....	218
4.2.6.16. start-application .....	219
4.2.6.17. stop-application .....	219
4.2.6.18. undeploy-application .....	220
4.2.6.19. undeploy-library .....	221
4.2.6.20. uninstall-application .....	221
4.2.6.21. uninstall-deployment-plan .....	222
4.2.6.22. uninstall-library .....	222
4.2.7. EJB 엔진 관련 명령어 .....	223
4.2.7.1. cancel-ejb-timer .....	223
4.2.7.2. ejb-timer-info .....	224
4.2.7.3. modify-active-management .....	225

4.2.7.4. modify-check-resolution .....	225
4.2.7.5. show-active-management .....	226
4.2.7.6. show-check-resolution .....	227
4.2.8. 웹 엔진 관련 명령어 .....	227
4.2.8.1. 공통적용 사항 .....	229
4.2.8.2. add-ajp-listener .....	230
4.2.8.3. add-backup-webtob .....	231
4.2.8.4. add-http-listener .....	231
4.2.8.5. add-response-header .....	232
4.2.8.6. add-tcp-listener .....	233
4.2.8.7. add-tmax-connector .....	234
4.2.8.8. add-virtual-host .....	235
4.2.8.9. add-valve .....	236
4.2.8.10. add-web-cookie-policy .....	237
4.2.8.11. add-web-encoding .....	238
4.2.8.12. add-web-properties .....	239
4.2.8.13. add-webtob-connector .....	240
4.2.8.14. clear-thread-local .....	242
4.2.8.15. clear-web-statistics .....	242
4.2.8.16. modify-jsp-engine .....	243
4.2.8.17. modify-response-header .....	245
4.2.8.18. modify-tmax-connector .....	246
4.2.8.19. modify-virtual-host .....	247
4.2.8.20. modify-web-cookie-policy .....	248
4.2.8.21. modify-web-encoding .....	249
4.2.8.22. modify-web-properties .....	250
4.2.8.23. modify-web-engine-configuration .....	251
4.2.8.24. modify-web-listener .....	253
4.2.8.25. modify-webtob-connector .....	254
4.2.8.26. notify-auto-scale .....	256
4.2.8.27. precompile-jsp .....	257
4.2.8.28. reload-web-context .....	258
4.2.8.29. remove-backup-webtob .....	259
4.2.8.30. remove-response-header .....	259
4.2.8.31. remove-tmax-connector .....	260
4.2.8.32. remove-virtual-host .....	261
4.2.8.33. remove-valve .....	262
4.2.8.34. remove-valve-property .....	262
4.2.8.35. remove-web-cookie-policy .....	263
4.2.8.36. resume-web-component .....	264
4.2.8.37. remove-web-encoding .....	265

4.2.8.38. remove-web-listener	266
4.2.8.39. remove-web-properties	266
4.2.8.40. remove-webtob-connector	267
4.2.8.41. set-valve-property	268
4.2.8.42. show-request-processing-flow	269
4.2.8.43. show-web-engine-configuration	269
4.2.8.44. show-web-statistics	271
4.2.8.45. show-webtob-connector	272
4.2.8.46. suspend-web-component	272
4.2.9. 세션 관련 명령어	273
4.2.9.1. 공통 적용 사항	275
4.2.9.2. list-session	275
4.2.9.3. remove-session	276
4.2.9.4. show-session-server-backup-table	277
4.2.9.5. show-session-configuration	278
4.2.9.6. modify-session-configuration	278
4.2.9.7. list-sessionstorages	280
4.2.9.8. add-sessionstorage	280
4.2.9.9. remove-sessionstorage	281
4.2.9.10. rename-sessionstorage	282
4.2.9.11. set-sessionstorage-property	283
4.2.9.12. remove-sessionstorage-property	284
4.2.9.13. add-sessionstorage-scope	284
4.2.9.14. set-sessionstorage-scope	285
4.2.9.15. remove-sessionstorage-scope	286
4.2.9.16. set-sessionstorage-scope-session-config	287
4.2.9.17. list-sessionserver-property	289
4.2.9.18. set-sessionserver-property	289
4.2.9.19. remove-sessionserver-property	290
4.2.9.20. add-central-session-server	291
4.2.9.21. modify-central-session-server	291
4.2.9.22. remove-central-session-server	292
4.2.9.23. add-central-session-server-definition	292
4.2.9.24. remove-central-session-server-definition	293
4.2.9.25. list-central-session-server-session	294
4.2.9.26. show-central-session-server-statistics	295
4.2.10. JMS 엔진 관련 명령어	295
4.2.10.1. 공통적용 사항	297
4.2.10.2. add-jms-connection-factory	297
4.2.10.3. add-jms-destination	298
4.2.10.4. add-jms-durable-subscription	300

4.2.10.5. add-jms-message-sort	301
4.2.10.6. add-jms-quota	302
4.2.10.7. add-jms-service-config	303
4.2.10.8. ban-jms-client	304
4.2.10.9. commit-jms-pending-transaction	304
4.2.10.10. control-jms-destination	305
4.2.10.11. control-jms-durable-subscription	306
4.2.10.12. delete-jms-messages	306
4.2.10.13. export-jms-messages	307
4.2.10.14. import-jms-messages	308
4.2.10.15. list-jms-clients	309
4.2.10.16. list-jms-connection-factories	310
4.2.10.17. list-jms-destinations	311
4.2.10.18. list-jms-durable-subscriptions	312
4.2.10.19. list-jms-message-sorts	313
4.2.10.20. list-jms-messages	314
4.2.10.21. list-jms-pending-transactions	315
4.2.10.22. list-jms-quotas	315
4.2.10.23. list-jms-service-configs	316
4.2.10.24. modify-jms-connection-factory	317
4.2.10.25. modify-jms-destination	319
4.2.10.26. modify-jms-durable-subscription	321
4.2.10.27. modify-jms-message-sort	322
4.2.10.28. modify-jms-quota	323
4.2.10.29. modify-jms-service-config	324
4.2.10.30. move-jms-messages	325
4.2.10.31. remove-jms-connection-factory	326
4.2.10.32. remove-jms-destination	327
4.2.10.33. remove-jms-durable-subscription	329
4.2.10.34. remove-jms-message-sort	330
4.2.10.35. remove-jms-quota	330
4.2.10.36. remove-jms-service-config	331
4.2.10.37. set-jms-engine-quota	332
4.2.10.38. set-jms-failover-check	333
4.2.10.39. set-jms-persistence-store	334
4.2.10.40. view-jms-message	335
4.2.11. 데이터소스 관련 명령어	336
4.2.11.1. add-cluster-data-source	337
4.2.11.2. add-data-source	339
4.2.11.3. list-cluster-data-sources	347
4.2.11.4. list-data-sources	348

4.2.11.5. modify-cluster-data-source .....	349
4.2.11.6. modify-data-source .....	352
4.2.11.7. remove-cluster-data-source .....	360
4.2.11.8. remove-data-source .....	361
4.2.11.9. test-data-source-config .....	361
4.2.12. Connection Pool 제어 및 모니터링 명령어 .....	362
4.2.12.1. connection-pool-info .....	363
4.2.12.2. control-connection-pool .....	364
4.2.12.3. control-cluster-data-source .....	365
4.2.12.4. create-connection-pool .....	366
4.2.12.5. destroy-connection .....	367
4.2.12.6. disable-connection-pool .....	367
4.2.12.7. enable-connection-pool .....	368
4.2.12.8. refresh-connection-pool .....	369
4.2.12.9. return-connection .....	369
4.2.12.10. shrink-connection-pool .....	370
4.2.13. 트랜잭션 관련 명령어 .....	371
4.2.13.1. modify-transaction-manager .....	371
4.2.13.2. recover-transactions .....	373
4.2.13.3. show-transaction-manager .....	374
4.2.13.4. transaction-info .....	375
4.2.14. 보안 관련 명령어 .....	376
4.2.14.1. 공통 적용 사항 .....	377
4.2.14.2. add-custom-password-validator .....	378
4.2.14.3. add-group .....	378
4.2.14.4. add-resource .....	379
4.2.14.5. add-role .....	380
4.2.14.6. add-user .....	381
4.2.14.7. add-user-to-group .....	382
4.2.14.8. assign-resource-to-role .....	383
4.2.14.9. assign-role-to-principal .....	384
4.2.14.10. check-group-member .....	385
4.2.14.11. expire-user .....	385
4.2.14.12. list-group-names .....	386
4.2.14.13. list-user-names .....	387
4.2.14.14. lock-user .....	387
4.2.14.15. modify-default-password-validator .....	388
4.2.14.16. modify-resource .....	389
4.2.14.17. modify-role .....	390
4.2.14.18. remove-custom-password-validator .....	391
4.2.14.19. remove-group .....	392

4.2.14.20. remove-resource	392
4.2.14.21. remove-role	393
4.2.14.22. remove-user	394
4.2.14.23. remove-user-from-group	394
4.2.14.24. set-password	395
4.2.14.25. show-custom-password-validator	396
4.2.14.26. show-default-password-validator	397
4.2.14.27. show-subject-validation	397
4.2.14.28. show-group	398
4.2.14.29. show-policy	398
4.2.14.30. show-user	400
4.2.14.31. unassign-role-from-principal	400
4.2.14.32. unassign-resource-from-role	401
4.2.14.33. unexpire-user	402
4.2.14.34. unlock-user	402
4.2.15. 노드 관리 관련 명령어	403
4.2.15.1. add-java-node	403
4.2.15.2. add-ssh-node	404
4.2.15.3. apply-patch	405
4.2.15.4. check-jeus-installed	406
4.2.15.5. check-ssh-node	407
4.2.15.6. install-jeus	407
4.2.15.7. list-nodes	408
4.2.15.8. modify-java-node	408
4.2.15.9. modify-node	409
4.2.15.10. modify-ssh-node	410
4.2.15.11. patch-info	411
4.2.15.12. remove-node	412
4.2.15.13. remove-patch	412
4.2.15.14. show-node	413
4.2.15.15. uninstall-jeus	414
4.2.16. 노드 매니저 관련 명령어	414
4.2.16.1. connect-nodemanager	415
4.2.16.2. disconnect-nodemanager	416
4.2.16.3. stop-nodemanager	417
4.2.16.4. nm-start-server	418
4.2.16.5. nm-stop-server	419
4.2.16.6. nm-state-server	420
4.2.17. 설정 편집 명령어	421
4.2.17.1. activate-configuration	423
4.2.17.2. cancel	424

4.2.17.3. change-current-path .....	425
4.2.17.4. create-empty-element .....	426
4.2.17.5. delete-element .....	428
4.2.17.6. start-configuration-editing .....	429
4.2.17.7. set-element-value .....	430
4.2.17.8. show-element .....	432
4.2.18. OSGi 관련 명령어 .....	433
4.2.18.1. add-osgi-framework .....	434
4.2.18.2. install-bundle .....	435
4.2.18.3. list-osgi-frameworks .....	436
4.2.18.4. modify-osgi-framework .....	437
4.2.18.5. remove-osgi-framework .....	438
4.2.18.6. show-osgi-framework-info .....	439
4.2.18.7. start-bundle .....	440
4.2.18.8. stop-bundle .....	441
4.2.18.9. uninstall-bundle .....	441
4.2.18.10. update-bundle .....	442
4.2.19. JSON을 사용한 명령어 실행 .....	443
4.2.19.1. 일반 명령어 실행 .....	443
4.2.19.2. Application deploy .....	445
4.2.20. Script 모드 사용법 및 Script 작성법 .....	446
4.2.20.1. Script 모드 사용법 .....	446
4.2.20.2. Script 작성법 .....	447
4.3. appcompiler .....	449
4.4. ejbdddinit .....	452
4.5. encryption .....	458
4.6. java2wsdl .....	460
4.7. jeusddupgrade .....	461
4.8. schemagen .....	462
4.9. tcpmon .....	463
4.9.1. 실행 .....	463
4.9.2. Listener 모드의 사용 .....	464
4.9.3. Proxy 모드의 사용 .....	467
4.9.4. 기타 기능 .....	467
4.10. webddgen .....	468
4.11. wsgen .....	469
4.12. wsimport .....	471
4.13. wsdl2java .....	473
4.14. wsdl2uddi .....	476
4.15. xjc .....	477
Part III. Ant Tasks .....	480

5. Ant Task .....	481
5.1. 개요 .....	481
5.2. Setup Ant Task .....	481
5.2.1. create-domain .....	482
5.3. EJB Ant Task .....	483
5.3.1. appcompiler .....	483
5.3.2. ejbddinit .....	485
5.4. 웹 서비스 Ant Task .....	487
5.4.1. java2wsdl .....	487
5.4.2. wsdl2java .....	489
5.4.3. wsgen .....	492
5.4.4. wsimport .....	493
5.4.5. xjc .....	495
5.4.6. schemagen .....	496
5.5. 웹 서비스 Maven Plugin .....	497
5.5.1. wsgen .....	497
5.5.2. wsimport .....	500
5.6. jeusadmin 명령어 실행 .....	503
5.7. Maven Plugin .....	505
5.7.1. 설치 .....	505
5.7.2. Maven Plugin Goals .....	506
5.7.2.1. start-master-xml .....	506
5.7.2.2. start-ms-xml .....	508
5.7.2.3. stop-jeus .....	510
5.7.2.4. deploy .....	512
5.7.2.5. undeploy .....	513
5.7.2.6. start-app .....	515
5.7.2.7. stop-app .....	517
5.7.2.8. install-app .....	518
5.7.2.9. uninstall-app .....	520
5.7.2.10. jeusadmin .....	522



# Part I. 시스템 프로퍼티

# 1. 시스템 프로퍼티

## 1.1. 개요

시스템 프로퍼티는 JVM에 -D 옵션으로 설정하는 환경변수이다. JEUS는 기본적으로 Launch JVM을 이용해서 서버 JVM을 구동하기 때문에 domain.xml의 각 서버별 설정에 시스템 프로퍼티를 설정해야 한다.

시스템 프로퍼티를 설정하는 방법은 아래와 같다.

```
<domain>
  <servers>
    <server>
      <name>server1</name>
      <jvm-config>
        <jvm-option>-Xmx512m -XX:MaxPermSize=250m</jvm-option>
      </jvm-config>
      ...
    
```



1. 도메인 Admin 서버 또는 서버 스크립트에 설정하는 것은 Launch JVM에만 적용되므로 주의한다.
2. 시스템 프로퍼티는 JEUS 전반적인 동작과 모든 애플리케이션에 영향을 미치므로 설정에 주의한다. 시스템 프로퍼티보다는 XML 설정을 사용하길 권장한다.

## 1.2. 서버 시스템 프로퍼티

본 절에서는 서버 시스템 프로퍼티에 대해 설명한다.

- jeus.home

설명	XML이나 코드 등을 생략한다.
기본값	JEUS가 설치된 디렉터리

- jeus.launcher.log.home

설명	JEUS Launcher Log가 생성되는 위치를 지정한다. 지정하지 않을 경우 JEUS_SERVER_HOME/logs에 남게 된다.
기본값	null

- jeus.boot.printclasspath

설명	시스템 클래스 로더의 클래스 패스에 대한 출력 여부를 결정한다.
기본값	false

- jeus.jdk.home

<b>설명</b>	JDK Home 디렉터리 경로로 javac, tools.jar 경로를 설정할 때 사용한다.
<b>기본값</b>	지정되어 있지 않으면 java.home을 사용

- jeus.properties.replicate

<b>설명</b>	Luncher에서 서버를 띄울 때 전달할 시스템 프로퍼티의 prefix이다.  domain.xml의 server/jvm-config/jvm-option이 우선한다.
<b>기본값</b>	jeus

- jeus.server.check-interrupted-status

<b>설명</b>	JEUS에서 EJB, JDBC operation을 수행하는 Thread가 Interrupt 신호를 받으면 Exception의 발생 여부를 설정한다.  JEUS에서는 특정 작업하는 경우 Thread가 Interrupt가 걸려있으면 Exception이 발생하므로 해당 Exception에 대해서는 사용자 애플리케이션에서 처리해 주어야 한다.
<b>기본값</b>	true

- jeus.threadpool.queuesize

<b>설명</b>	JEUS 시스템의 Threrad Pool에서 사용하는 Thread Pool의 내부 Queue size를 설정한다. Thread Pool에서는 Task 수행을 위해 내부 Queue를 사용하는데 이 Queue의 크기를 설정한다. 만약 시스템의 부하가 많으나 설정 Thread의 개수가 작을 경우 Queueing되는 Task들이 증가할 것이고, 이때 최대 설정한 Queue 크기만큼만 Queueing을 허용한다. 이 시스템 프로퍼티보다 domain.xml의 server/system-thread-pool/queue-size 값이 우선한다.
<b>기본값</b>	4096

- jeus.threadpool.keepalive

<b>설명</b>	JEUS 시스템의 Threrad Pool에서 사용하는 Thread Pool에 적용될 Keep alive time을 설정한다. 만약 min 이하의 Thread에 대해서 설정된 시간 동안 사용되지 않는다면, 해당 Thread는 Thread Pool에서 제거된다.  이 시스템 프로퍼티보다 domain.xml의 server/system-thread-pool/keep-alive-time 값이 우선한다.
<b>기본값</b>	600000 (단위: ms)

- jeus.rmi.defaultPort

<b>설명</b>	RMI의 default export port이다. export-port가 지정되지 않은 모든 EJB에도 적용된다.
<b>기본값</b>	- base port + 7

- jeus.license.file.path

<b>설명</b>	라이선스 파일 경로를 지정한다. 파일을 지정하면 해당 파일을 읽고, 디렉터리를 지정하면 경로 내의 license.tmax 파일을 읽는다.
<b>기본값</b>	null

- jeus.vm.bits

<b>설명</b>	JVM의 bit 수이다.  서버가 기동할 때 64bit JVM인 경우 64를 설정하면 -d64가 추가된다(IBM의 경우에는 -d64가 필요없다).
-----------	---

- jeus.interop.ssl.keystore

<b>설명</b>	CORBA CSIV2에서 사용할 SSL Keystore를 지정한다.
<b>기본값</b>	jeus.net.ssl.keystore의 값

- jeus.interop.ssl.keypass

<b>설명</b>	CORBA CSIV2에서 사용할 SSL Keystore의 패스워드를 지정한다.
<b>기본값</b>	jeus.net.ssl.keypass의 값

- jeus.interop.ssl.truststore

<b>설명</b>	CORBA CSIV2에서 사용할 SSL Truststore를 지정한다.
<b>기본값</b>	jeus.net.ssl.truststore의 값

- jeus.interop.ssl.trustpass

<b>설명</b>	CORBA CSIV2에서 사용할 SSL Truststore의 패스워드를 지정한다.
<b>기본값</b>	jeus.net.ssl.trustpass의 값

- jeus.jvm.version

<b>설명</b>	JEUS가 사용하는 JVM의 버전에 대한 정보이다.
<b>기본값</b>	hotspot (-server를 사용함)

- jeus.server.cpumonitor.enabled

<b>설명</b>	HOST CPU 사용률을 측정하기 위한 모니터링 기능을 활성화한다.
<b>기본값</b>	false

- jeus.server.cpumonitor.interval

<b>설명</b>	HOST CPU 사용률을 모니터링하는 기능을 사용하는 경우 CPU 사용률을 측정할 주기이다.
<b>기본값</b>	60000 (단위: ms)

- jeus.server.process.cpumonitor.enabled

<b>설명</b>	Process CPU 사용률을 측정하기 위한 모니터링 기능을 활성화한다.
<b>기본값</b>	false

- jeus.server.process.cpumonitor.interval

<b>설명</b>	Process CPU 사용률을 모니터링하는 기능을 사용하는 경우 CPU 사용률을 측정할 주기이다.
<b>기본값</b>	2000 (단위: ms)

- jeus.server.cpumonitor.ratio

<b>설명</b>	Process CPU 사용률을 모니터링하는 기능을 사용하는 경우 CPU 오버플로우의 기준치를 설정한다. 소수점으로 설정하며 CPU 사용량을 최대 100으로 볼 때 최대 몇 퍼센트의 CPU를 사용하고 있는 경우 오버플로우로 가정할지를 설정한다.
<b>기본값</b>	0.8

- jeus.server.enable.restart.in.memory.shortage

<b>설명</b>	메모리가 오버플로우될 때 서버의 재기동 여부를 설정한다.
<b>기본값</b>	true

- jeus.server.memorymonitor.enabled

<b>설명</b>	서버의 메모리 사용률을 측정하기 위한 모니터링 기능을 활성화한다. 모니터링할 때 Thread Dump가 로그에 남게 된다.
<b>기본값</b>	false

- jeus.server.memorymonitor.interval

<b>설명</b>	메모리 사용률을 모니터링하는 기능을 사용하는 경우 메모리 사용률을 측정하고자 하는 주기이다.
<b>기본값</b>	2000 (단위: ms)

- jeus.server.memorymonitor.duration

<b>설명</b>	메모리 사용률을 모니터링하는 기능을 사용하는 경우 메모리 오버플로우 현상의 지속 시간을 설정한다. 설정한 시간동안 메모리 오버플로우 시간이 지속되면 서버가 종료된다.
-----------	--

<b>기본값</b>	60000 (단위: ms)
------------	----------------

- jeus.server.memorymonitor.ratio

<b>설명</b>	메모리 사용률을 모니터링하는 기능을 사용하는 경우 메모리 오버플로우의 기준치를 설정한다. 소수점으로 설정하며 메모리 사용량을 최대 100으로 볼 때 최대 몇 퍼센트의 메모리를 사용하고 있는 경우 오버플로우로 가정할지를 설정한다.
<b>기본값</b>	0.8

- jeus.server.maxdowntime

<b>설명</b>	메모리 사용률을 모니터링하는 기능을 사용하는 경우 메모리 오버플로우가 발생하여 서버를 종료할 때 서버가 종료될 때까지 대기하는 시간을 설정한다.  설정한 시간이 지나도 서버가 종료되지 않으면 서버는 강제종료된다.
<b>기본값</b>	0 (단위: ms)

- jeus.server.configfile.backup.size

<b>설명</b>	동적 변경할 때 저장되는 백업 domain.xml의 개수를 지정한다.
<b>기본값</b>	50

- jeus.server.standby.retrycount

<b>설명</b>	서버를 STANDBY 상태로 기동한 경우 기동을 재시도할지 여부 및 횟수를 지정한다.  STANDBY 상태로 기동하는 원인을 정확하게 파악하고 있고, 확실하게 복구 가능하다고 판단되는 경우에만 설정해야 한다.
<b>기본값</b>	0 (재시도하지 않음), 5 (클러스터에 속한 서버인 경우)

- jeus.server.standby.retryinterval

<b>설명</b>	jeus.server.standby.retrycount를 통해 횟수를 지정하고, 본 시스템 프로퍼티를 통해 재시도 사이에 간격을 지정한다.
<b>기본값</b>	60000 (단위: ms)

- jeus.print-system-properties

<b>설명</b>	설정된 시스템 프로퍼티를 출력할지를 설정한다.
<b>기본값</b>	false

- jeus.management.snmp.dynamic

<b>설명</b>	<p>SNMP를 서버에 사용하는 경우 Dynamic OID 생성 여부를 설정한다.</p> <p>SNMP를 서버에 사용하는 경우 OID를 Static으로 할지, Dynamic으로 할지 설정한다.</p> <ul style="list-style-type: none"> <li>• true : Static일 경우로 MIB 파일 기반의 OID를 생성한다.</li> <li>• false : Dynamic일 경우로 객체에 따라서 동적으로 생성된 OID가 Static OID의 접미사로 추가된다.</li> </ul>
<b>기본값</b>	true

- jeus.management.snmp.dynamic.shortoid

<b>설명</b>	<p>Dynamic OID의 길이를 선택한다.</p> <p>MIB 기반의 OID에 추가적으로 더해지는 OID의 구성요소의 자리 수를 결정한다.</p> <ul style="list-style-type: none"> <li>• true : Short OID일 경우로 8자리의 ID가 추가된다.</li> <li>• false : Short OID가 아닐 경우로 32자리의 ID가 추가된다.</li> </ul>
<b>기본값</b>	false

- jeus.domainadminserver.deploy.socket

<b>설명</b>	<p>MASTER(Master Server)가 유지하는 deploy 용도 소켓 커넥션 개수의 최댓값이다.</p> <p>MS(Managed Server)에서는 부팅 시간을 단축시키기 위해 MASTER로부터 애플리케이션을 받아올 때 커넥션을 여러 개 맺고 병렬로 받아올 수 있다. 이때 MASTER는 한순간에 여러 MS로부터 많은 커넥션을 맺게 되므로 이 최댓값을 설정할 수 있도록 한다.</p> <p>MASTER의 deploy 커넥션이 최댓값이 된 상태에서 다른 MS로부터 애플리케이션 전송 요청이 올 경우에는 해당 MS는 1개의 커넥션만을 사용하여 순차적으로 애플리케이션을 받아오도록 한다.</p>
<b>기본값</b>	100

- jeus.managedserver.deploy.socket

<b>설명</b>	<p>MASTER로부터 애플리케이션을 받아올 때 MS가 사용하는 boot time deploy connection의 최댓값이다.</p> <p>MS가 boot하면서 MASTER로부터 애플리케이션을 받아오게 되는데 이때 병렬적으로 받아오기 위해 사용하는 MS boot time deploy connection 개수의 최댓값을 설정한다. MS가 너무 많은 커넥션을 설정해서 MASTER에서의 커넥션 개수가 full이 되버릴 경우 다른 MS가 영향을 받을 수 있으므로 주의하여 사용한다.</p> <p>애플리케이션 전송이 끝나면 해당 커넥션은 바로 해제되어 MASTER의 가용한 커넥션의 개수를 다시 늘려준다. 또한 해당 설정의 커넥션 개수를 크게 설정하더라도 받아올 애플리케이션의 개수가 작다면 내부적으로 유지하는 커넥션의 개수는 애플리케이션의 개수로 자동으로 설정된다.</p>
<b>기본값</b>	5

- jeus.logging.useAsync

<b>설명</b>	프로퍼티에 대한 자세한 내용은 JEUS Server 안내서의 "개요"를 참고한다.
<b>기본값</b>	true

- jeus.access.logging.skip.when.busy

<b>설명</b>	jeus.logging.useAsync가 true일 때에 한해 적용되는 옵션이다. 프로퍼티에 대한 자세한 내용은 JEUS Server 안내서의 "개요"를 참고한다.
<b>기본값</b>	false

- jeus.logging.gclog.timestamp.on

<b>설명</b>	JVM에서는 기본적으로 GC Log가 jvm.log에 다른 로그들과 함께 기록된다. 이때, "-Xloggc:[파일]"을 jvm option으로 제공하여 원하는 [파일]에 GC Log를 redirect하여 기록할 수 있다. JEUS에서는 설정한 [파일]의 이름에 자동으로 timestamp를 추가하도록 되어있다. 이러한 동작은 JEUS가 기동될 때마다 동일한 [파일]에 GC Log를 덮어쓰지 않도록하기 위함이다. 그러나 사용자에게 따라 이러한 동작을 원치 않을 수 있다. 이 경우 이 옵션을 해제하면 된다. 이 옵션을 해제할 경우에는 동일한 [파일]에 GC Log를 덮어쓰기 때문에 이전 로그를 잃으므로 주의하여 사용한다.
<b>기본값</b>	true

- jeus.jmx.ping.period

<b>설명</b>	JMX 커넥션에서 연결 상태를 체크하기 위해 사용하는 Ping의 주기를 설정한다.
<b>기본값</b>	15000 (단위: ms)

- jeus.jmx.ping.timeout

<b>설명</b>	JMX 커넥션에서 연결 상태를 체크하기 위해 사용하는 Ping의 타임아웃을 설정한다.  Ping을 보낸 후 응답이 타임아웃 시간만큼 오지 않으면 연결에 문제가 있다고 판단하여 IOException을 발생시킨다.
<b>기본값</b>	15000 (단위: ms)



jeus.server.memorymonitor.enabled, jeus.server.memorymonitor.ratio, jeus.server.memorymonitor.interval, jeus.server.memorymonitor.duration, jeus.server.enable.restart.in.memory.shortage 프로퍼티에 대한 자세한 내용은 JEUS Server 안내서의 "메모리 사용량에 따른 제어"를 참고한다.

- jeus.config.lock.timeout



<b>설명</b>	동적 설정 변경을 위한 Lock을 획득하지 못했을 때 대기할 시간을 지정한다. 0 이하의 값을 지정한 경우에는 대기하지 않는다.
<b>기본값</b>	60000 (단위: ms)

- jeus.jmx.worker.max

<b>설명</b>	JMX 요청 메시지를 처리할 스레드를 제공하는 스레드 풀이 가질 수 있는 스레드의 최대치를 지정한다.
<b>기본값</b>	20

- jeus.server.command.threadpool.maxsize

<b>설명</b>	MASTER에서 명령을 병렬로 수행할 때 사용할 스레드 풀이 가질 수 있는 스레드의 최대치를 지정한다. 이를 설정할 때 너무 큰 값을 설정할 경우 MASTER의 CPU 점유율이 오를 수 있으므로 주의해야 한다.
<b>기본값</b>	10

- jeus.server.invocation.keepBootOnError

<b>설명</b>	서버 기동 중 life cycle invocation 설정 문제로 에러가 발생한 경우 서버를 Standby 상태가 아닌 Running 상태로 기동시킬지 여부를 결정한다.
<b>기본값</b>	false

- jeus.boot.ignorepatchoverlap

<b>설명</b>	패치 파일에 중복 클래스가 존재하는 경우, 이를 무시하고 기동을 진행할지 여부를 지정한다.
<b>기본값</b>	false

- jeus.server.enable.auto-deploy

<b>설명</b>	기동 시 지정한 위치에 있는 애플리케이션 파일을 자동으로 배포하는 boot-time auto deployment 기능을 사용할지 여부를 설정한다.
<b>기본값</b>	true

- jeus.server.auto-deploy.dir

<b>설명</b>	Boot-time auto deployment 수행 시 애플리케이션 파일을 탐색할 경로를 지정한다. 지정하지 않을 경우 내부 기본값으로 DOMAIN_HOME/auto-deploy를 탐색한다.
<b>기본값</b>	null

- jeus.logging.defaultfilehandler.rotationcount

<b>설명</b>	JEUS Launcher Logger 등에서 사용하는 기본 파일 핸들러에 대해 rotation count를 지정한다. 서버 기동 스크립트에 설정해야 한다.
<b>기본값</b>	0

- jeus.launcher.jvmlog.rotationcount

<b>설명</b>	jvm.log 파일에 대한 rotation count를 지정한다. 서버 기동 스크립트에 설정해야 한다.
<b>기본값</b>	0

- jeus.server.thread.add-jvm-id-and-thread-id

<b>설명</b>	JEUS ThreadPool에서 생성하는 Thread의 이름 뒤에 [JVM ID-Thread ID]를 붙일지 여부를 지정한다.
<b>기본값</b>	false

- jeus.server.disable.lock.file

<b>설명</b>	기동 시 SERVER_HOME/.workspace/tmp/SERVER_NAME.lck 파일에 대한 락을 사용할지 여부를 지정한다.
<b>기본값</b>	false

- jeus.master.protocol

<b>설명</b>	Master 서버 주소의 프로토콜을 지정한다.
<b>기본값</b>	http

- jeus.master.host

<b>설명</b>	Master 서버 주소의 host를 지정한다.
<b>기본값</b>	localhost

- jeus.master.port

<b>설명</b>	Master 서버 주소의 port를 지정한다.
<b>기본값</b>	9736

- jeus.master.aggregator.context.path

<b>설명</b>	jeus 측에서 MasterAggregator spring app을 기동시킬 context path를 지정하여 부팅시킬 수 있다.
<b>기본값</b>	/jeus/

- jeus.master.admin.ui.context.path

설명	jeus 측에서 MasterAdminUI spring app을 기동시킬 context path를 지정하여 부팅시킬 수 있다.
기본값	/admin/

## 1.3. 네트워크 시스템 프로퍼티

본 절에서는 네트워크 시스템 프로퍼티에 대해 설명한다.

- jeus.server.checktmout

설명	JEUS를 관리하는 RMI 커넥션이나 JMX 커넥터에 대해 적용할 타임아웃을 설정한다.
기본값	60 * 1000 (단위: ms)

- jeus.net.crosswait

설명	두 JEUS 서버가 서로 커넥션을 맺으려고 시도하는 Cross Connection 상황일 때 커넥션이 거부당한 프로세스에서 상대방이 접속 요청을 하여 커넥션이 맺어지길 기다리는 시간을 지정한다.
기본값	10000 (단위: ms)

- jeus.net.timewait

설명	CLOSE 메시지가 전달되기를 기다리는 시간이다. 이 시간이 짧으면 정상적인 종료임에도 IOException이 발생했다는 trace가 출력될 수 있다.
기본값	10000 (단위: ms)

- jeus.net.connect.interval

설명	JEUS 프로세스 간 커넥션 시도 시 실패로 인해 재시도할 경우 재시도 사이의 시간 간격을 설정한다.
기본값	3000 (단위: ms)

- jeus.net.tcpbuffer

설명	JEUS가 운영되는 환경의 TCP Buffer Size를 설정한다.  이 값은 JEUS 서버 내부의 Read Buffer의 크기로 사용되기 때문에 이 값이 정확해야 최대의 I/O 성능을 낼수 있다.
기본값	8*1024 (단위: byte)

- jeus.net.recvbuffer

<b>설명</b>	JEUS 서버 환경의 TCP Receive Buffer의 크기를 설정한다.
-----------	---

- jeus.net.sendbuffer

<b>설명</b>	JEUS 클라이언트 환경의 TCP Send Buffer의 크기를 설정한다.
-----------	---

- jeus.net.busywrite

<b>설명</b>	non-blocking으로 동작하는 경우 write를 할 때 Selector에게 맡기지 않고 계속 시도하는 횟수를 설정한다. 받는 쪽에서 read를 하지 않는 경우를 제외하고는 이 횟수 내에서는 select()를 다시 호출하는 일이 없으므로 빠르다.
<b>기본값</b>	2

- jeus.net.msg.max

<b>설명</b>	JEUS 프로토콜을 통해서 전달될 수 있는 한 메시지의 최대 크기를 설정한다.  이 크기를 초과하는 메시지를 보낼 경우 수신측에서 Exception이 발생하고 연결이 끊어질 수 있다.
<b>기본값</b>	5 * 1000 * 1000 (단위: byte)

- jeus.net.ping.enable

<b>설명</b>	양방향 Ping 기능의 사용 여부를 설정한다.
<b>기본값</b>	false

- jeus.net.ping.timeout

<b>설명</b>	Ping 메시지의 응답을 기다리는 시간을 설정한다.
<b>기본값</b>	60*1000 (단위: ms)

- jeus.net.ping.period

<b>설명</b>	Ping 메시지를 보내는 주기를 설정한다 .  값이 0이면 Ping을 보내어 상대 노드가 살아 있는지 체크하지 않는다.
<b>기본값</b>	10*60*1000 (단위: ms)

- jeus.net.nio.select.limit

<b>설명</b>	non-blocking으로 동작할 때 select()의 호출이 이 횟수 이상 실패하면 새로운 Selector를 만들어서 대체시킨다.
<b>기본값</b>	5

- jeus.net.nio.write.limit

<b>설명</b>	non-blocking으로 동작하는 경우 Write Thread가 블록되기까지 Write Queue에 쌓이는 메시지 크기를 설정한다. 현재 write를 할 수 없는 상황이면 Write Queue에 넣게 된다.  지정한 byte 수를 초과하면 Write Thread를 블록시킨다.
<b>기본값</b>	10* 1024 (단위: byte)

- jeus.net.nio.write.restart

<b>설명</b>	non-blocking으로 동작하는 경우 Write Thread가 다시 시작할 때 Write Queue에 쌓인 메시지 크기를 설정한다. Write Event로 Write Queue의 byte 수가 지정한 byte보다 적어지면 블록되어 있는 Write Thread들이 모두 깨어난다.
<b>기본값</b>	3*1024 (단위: byte)

- jeus.net.nio.write.repeat.count

<b>설명</b>	한 메시지를 끝까지 쓰기 위해 write를 시도하는 횟수를 설정한다.  Selector Thread가 한 소켓에 큰 메시지가 있어서 그것을 write하느라 다른 소켓의 메시지를 write 못하는 경우에 사용한다. 여러 소켓에 사용되는 메시지의 크기가 모두 큰 경우에는 이 옵션을 적용하면 response가 오히려 느려질 수 있으니 주의한다.
<b>기본값</b>	-1(무한대)

- jeus.net.reuseAddress

<b>설명</b>	서버 소켓의 SO_REUSEADDR이다.
<b>기본값</b>	false

- jeus.ssl.enabledProtocols

<b>설명</b>	SSL 인증에 사용할 SSL 프로토콜 버전들을 설정한다. 각 프로토콜 버전들은 콤마(,)로 구분하여 나열한다.
<b>기본값</b>	null

- jeus.net.client.use-ssl

<b>설명</b>	JEUS에 접속하는 클라이언트가 SSL을 사용할 지의 여부를 설정한다. 다른 설정들은 별도의 시스템 프로퍼티로 설정해야 한다.
<b>기본값</b>	false

- jeus.net.client.ssl-properties-file

<b>설명</b>	JEUS에 접속하는 클라이언트가 사용할 설정이 담긴 프로퍼티 파일(Properties File)을 설정한다.
<b>기본값</b>	null

- jeus.ssl.keystore 혹은 javax.net.ssl.keyStore

<b>설명</b>	SSL 인증에 사용할 Keystore의 경로를 설정한다.
-----------	---------------------------------

- jeus.ssl.keypass 혹은 javax.net.ssl.keyStorePassword

<b>설명</b>	SSL 인증에 사용할 Keystore의 Password를 설정한다.
-----------	---------------------------------------

- jeus.ssl.keystorepass

<b>설명</b>	SSL 인증에 사용할 Keystore의 Key Password를 설정한다.
-----------	---

- jeus.ssl.truststore 혹은 javax.net.ssl.TrustStore

<b>설명</b>	SSL 인증에 사용할 Truststore의 경로를 설정한다.
-----------	-----------------------------------

- jeus.ssl.trustpass 혹은 javax.net.ssl.TrustStorePassword

<b>설명</b>	SSL 인증에 사용할 Truststore의 Password를 설정한다.
-----------	---

- jeus.net.ban-client-address-threshold

<b>설명</b>	클라이언트의 주소를 자동으로 차단할 때 유예 횟수를 지정한다.
<b>기본값</b>	5

## 1.4. JNDI 시스템 프로퍼티

본 절에서는 JNDI 시스템 프로퍼티에 대해 설명한다.

- java.naming.factory.initial (필수)

<b>설명</b>	JNDI 서비스 프로바이더에 대한 InitialContext Factory 클래스 이름을 설정한다.  이 Factory 클래스를 통해 InitialContext를 생성한다. 이 프로퍼티에 jeus.jndi.JNSContextFactory를 넣어서 JEUS의 InitialContext를 생성한다.
<b>기본값</b>	jeus.jndi.JNSContextFactory

- java.naming.factory.url.pkgs

<b>설명</b>	JEUS의 InitialContext에서 URL scheme을 사용해서 lookup하기 위해 설정한다.
<b>기본값</b>	jeus.jndi.jns.url

- java.naming.provider.url

<b>설명</b>	JNDI 서비스를 제공하는 MS의 DNS 호스트 이름과 포트를 정의한다. 즉, JEUS의 JNSServer의 IP 주소를 설정한다. 클러스터링으로 묶인 MS에 JNDI 요청을 할 때에는 클러스터링된 모든 서버의 주소를 콤마(,)로 구분해야 한다.  (예: yohan:9736, yohan1:9836)
<b>기본값</b>	127.0.0.1:9736

- java.naming.security.principal

<b>설명</b>	JEUS Naming Server에서 인증받기 위한 사용자명을 입력한다. 사용자명이 설정되지 않으면 “anonymous”로 처리된다.  사용자명은 JEUS Security 도메인에 정의되어 있어야 한다.
<b>기본값</b>	anonymous

- java.naming.security.credentials

<b>설명</b>	JEUS Naming Server에서 인증받기 위한 패스워드를 입력한다.
<b>기본값</b>	N/A

- jeus.jndi.jns.resolution

<b>설명</b>	JNSClient가 리소스를 관리하는 체크 시간의 간격을 지정한다.
<b>기본값</b>	30 * 1000 (단위: ms)

- jeus.jndi.jns.connecttimeout

<b>설명</b>	JNSClient가 JNSServer로 접속하는 데 사용하는 타임아웃을 지정한다.  이 시간 동안 접속을 시도하다가 시간이 초과되면 jeus.net.ConnectException이 발생한다.
<b>기본값</b>	0 (무한대를 의미, OS에 따라 다를 수 있음) (단위: ms)

- jeus.jndi.jns.connectionduration

<b>설명</b>	JNSClient가 JNSServer로 접속한 커넥션을 얼마나 유지할지 설정한다. 지정한 시간 동안 커넥션을 사용하지 않으면 커넥션이 자동으로 종료된다.  기본값은 0으로 커넥션이 계속 유지된다.
-----------	---

<b>기본값</b>	0 (단위: ms)
------------	------------

- jeus.jndi.local.timeout

<b>설명</b>	JNDI 서비스에서 클라이언트 역할을 하는 JNSClient에서 요청을 보낸 후 응답이 올 때까지 기다리는 시간을 지정한다. 이 시간이 지나도 JNSServer로부터 응답이 오지 않으면 jeus.util.WaitTimeoutException이 발생한다.
<b>기본값</b>	20 * 1000 (단위: ms)

- jeus.jndi.cluster.recheckto

<b>설명</b>	JEUSClusterContext를 사용할 때 fail된 서버가 다시 살아났는지 확인할 주기를 지정한다.
<b>기본값</b>	300 * 1000 (단위: ms)

- jeus.cluster.retry

<b>설명</b>	컨테이너나 클라이언트에서 JEUS Manager로 클러스터 커넥션을 맺을 때 예외가 발생할 경우 재시도하는 횟수를 지정한다.
<b>기본값</b>	1

- jeus.jndi.clusterlink.selection-policy

<b>설명</b>	<p>클러스터링된 서버에 바인드된 어떤 객체를 lookup해서 사용하는 경우 어떤 서버에서 객체를 가져올 것인지에 대한 정책을 설정한다. EJB 2.x를 사용하는 경우에는 사용할 EJB 엔진을 선택하는 데에도 사용된다. 직접 Hashtable을 이용하여 환경값을 준 경우에는 system property보다는 주어진 환경 값을 우선한다.</p> <p>다음 중 하나를 선택할 수 있다.</p> <ul style="list-style-type: none"> <li>• locallinkPreference : 로컬 MS에 있는 객체를 사용하도록 한다. (기본값)</li> <li>• roundrobin : 처음 요청에서는 랜덤하게 선택한 MS의 객체를 사용하고, 그 후 요청부터는 하나씩 증가하면서 서버를 선택한다.</li> <li>• random : 클러스터링된 MS들 중에서 랜덤하게 하나를 선택하여 사용한다.</li> <li>• xid-hashing : xid의 hashing값에 의해 서버를 결정한다. 그 결과 동일한 xid를 가진 요청은 동일한 서버만을 선택한다.</li> </ul>
<b>기본값</b>	locallinkPreference

- jeus.jndi.enable.cache

<b>설명</b>	<p>JNDI 클라이언트(JNSClient)에서 캐시(Cache)를 사용할 것인지를 여부를 설정한다.</p> <ul style="list-style-type: none"> <li>• true : 캐시를 사용한다. (기본값)</li> <li>• false : 캐시를 사용하지 않는다.</li> </ul>
-----------	--



기본값	true
-----	------

- jeus.jndi.jns.replicatebindings

설명	바인드하는 객체를 클러스터링으로 묶인 MS 전체에서 공유할 것인지 여부를 설정한다.  <ul style="list-style-type: none"> <li>• true : 객체를 공유한다. (기본값)</li> <li>• false : 객체를 공유하지 않는다.</li> </ul>
기본값	true

- jeus.jndi.jns.cachebindings

설명	객체를 바인드할 때 이 옵션을 설정하면 JNSServer에서 바인드를 완료한 후 JNSClient에 그 객체를 caching한다.  <ul style="list-style-type: none"> <li>• true : 객체를 caching한다. (기본값)</li> <li>• false : 객체를 caching하지 않는다.</li> </ul>
기본값	true

- jeus.jndi.jns.localbindings

설명	객체를 자신의 JVM에 있는 JNSClient에만 바인드한다.  <ul style="list-style-type: none"> <li>• true : 객체를 JEUS Naming Client에만 바인드한다. 이렇게 바인드한 객체는 JNDI 요청을 하는 클라이언트에서만 사용할 수 있다.</li> <li>• false : 객체를 JEUS Naming Server에 바인드한다. (기본값)</li> </ul>
기본값	false

- jeus.jndi.compat

설명	JNDI compatibility를 위한 설정이다. JEUS 7 Fix#2 이후 버전과 이전 버전 사이에 JNDI 서비스를 제공하려면 해당 property를 true로 설정해야 한다.  <ul style="list-style-type: none"> <li>• true : JNDI 서버가 JEUS 7 Fix#2 이전 버전이고, JNDI 클라이언트가 JEUS 7 Fix#2를 포함한 이후 버전일 때 JNDI compatibility를 위해 해당 property를 true로 설정해야 한다.</li> <li>• false : JEUS 7 Fix#2 이후 버전과 이전 버전 사이에 JNDI compatibility를 유지하지 않는다. (기본값)</li> </ul>
기본값	false

## 1.5. JDBC 시스템 프로퍼티

본 절에서는 JDBC 시스템 프로퍼티에 대해 설명한다.

- jeus.jdbc.connection-pool.initialization-on-boot

<b>설명</b>	<p>서버 내의 JDBC Connection pool들을 서버 기동 단계에서 초기화 할 것인지 여부를 설정한다.</p> <ul style="list-style-type: none"> <li>• true : JDBC Connection pool 초기화 작업을 서버 기동 단계에서 수행한다.</li> <li>• false : JDBC Connection pool 초기화 작업을 서버 기동 단계가 아닌 최초 서비스 요청 단계에서 수행한다. (기본값)</li> </ul>
<b>기본값</b>	false

- jeus.jdbc.connection-pool.keep-initialization-on-connection-error

<b>설명</b>	<p>데이터소스 Connection pool 초기화 중 JDBC Connection 생성 단계에서 에러가 발생해도 Connection pool 초기화 작업을 완료 할 것인지 여부를 설정한다.</p> <ul style="list-style-type: none"> <li>• true : Connection pool 초기화 중 JDBC Connection 생성 에러가 발생하더라도 Connection pool만 empty 상태로 Connection pool 초기화 작업을 완료하며, Connection pool의 상태를 별도로 failed 상태로 관리한다. (기본값)</li> <li>• false : Connection pool 초기화 중 JDBC Connection 생성 에러가 발생하는 경우 초기화 전체 작업을 실패로 처리한다.</li> </ul>
<b>기본값</b>	true

- jeus.jdbc.monitoring.active-connection-number-recording-buffer

<b>설명</b>	<p>JDBC 모니터링 값 중에 "Active Average"라는 값이 있다. 해당 값의 산정은 정해진 시간 기간 동안의 최근 active count 수치를 정해진 시간 간격으로 획득하여 평균을 내는 방식으로 이루어진다.</p> <p>본 옵션은 최근 몇 초의 시간 동안 축적된 데이터를 바탕으로 active connection average count를 보여줄 지를 결정하는 값이다. 예를 들어 기본값인 3600이 적용될 경우 최근 1시간 동안의 active count들에 대한 평균을 "Active Average" 값으로 보여주게 된다.</p>
<b>기본값</b>	3600 (단위: 초)

- jeus.jdbc.monitoring.active-connection-number-recording-period

<b>설명</b>	<p>JDBC 모니터링 값 중에 "Active Average"라는 값이 있다. 해당 값의 산정은 정해진 시간 기간 동안의 최근 active count 수치를 정해진 시간 간격으로 획득하여 평균을 내는 방식으로 이루어진다.</p> <p>본 옵션은 백그라운드에서 몇 초 간격으로 active connection count를 체크하여 데이터를 축적할 것인지를 결정하는 값이다. 예를 들어 기본값인 5가 적용될 경우 5초 간격으로 active count를 저장하여 그 평균 수치를 "Active Average" 값으로 보여주게 된다.</p>
<b>기본값</b>	5 (단위: 초)

- jeus.jdbc.config.data-source-account-provider-class-name

<b>설명</b>	사용자가 데이터소스의 DB 접속 계정 관리를 Customize하고 싶을 때 그것을 위해 구현한 클래스의 전체 이름을 설정한다. 이때 그 클래스는 반드시 jeus.jdbc.helper.DataSourceAccountProvider 인터페이스를 구현해야 한다. 자세한 내용은 JEUS Server 안내서의 "DataSourceAccountProvider 인터페이스"를 참고한다.  만약 본 옵션이 설정되지 않으면 JEUS는 JEUS Security와 연동하는 기본 구현체를 사용한다.
<b>기본값</b>	null

- jeus.jdbc.xa-check

<b>설명</b>	validation 과정에서 check query를 수행하기전에 XAResource#start가 가능한지 체크해본다.
<b>기본값</b>	false

- jeus.jdbc.config.database-name-as-service-name

<b>설명</b>	서버에 등록된 datasource를 SID가 아닌 Service Name으로 Database Name을 설정하여 사용하고 싶은 경우 true로 설정한다.
<b>기본값</b>	false

- jeus.jdbc.config.exclude.database-name-as-service-name

<b>설명</b>	jeus.jdbc.config.database-name-as-service-name 프로퍼티가 true일 때 Database Name을 Service Name으로 치환하지 않을 datasourceID를 공백 하나로 구분하여 나열한다.  <div style="border: 1px solid #ccc; padding: 5px; margin: 5px 0;">-Djeus.jdbc.config.exclude.database-name-as-service-name="ds1 ds2"</div>
<b>기본값</b>	빈 문자열

## 1.6. 트랜잭션 시스템 프로퍼티

본 절에서는 트랜잭션 시스템 프로퍼티에 대해 설명한다.

- jeus.tm.forcedReg

<b>설명</b>	서브 코디네이터가 루트 코디네이터에 등록하는 방식을 결정한다.  <ul style="list-style-type: none"> <li>• true : 트랜잭션이 연동되면 바로 등록한다.</li> <li>• false : RM이 있어야만 등록한다.</li> </ul>
<b>기본값</b>	true

- jeus.tm.checkReg

<b>설명</b>	서브 코디네이터가 루트 코디네이터에 제대로 등록되었는지를 확인하고 진행할지 확인하지 않고 진행할지 여부를 결정한다.
<b>기본값</b>	true

• jeus.tm.disableJoin

<b>설명</b>	동일한 RM에서 나온 리소스들을 Join할 것인지 여부를 결정한다.
<b>기본값</b>	false

• jeus.tm.noLogging

<b>설명</b>	트랜잭션 매니저가 복구를 위해 트랜잭션 정보를 로그로 남기게 되는데 이를 남길지 여부를 설정한다.  <ul style="list-style-type: none"> <li>• true : 로그를 남기지 않는다.</li> <li>• false : 로그를 남긴다.</li> </ul>
<b>기본값</b>	false

• jeus.tm.ignore.broken.log.file

<b>설명</b>	TM 로그 파일이 깨져서 복구가 불가능한 상황일 때 해당 옵션이 true일 경우 JEUS에서 미리 파일이 깨진 것을 detect하고 전부 지워버린다. 파일이 깨진 채로 복구를 시도하게 될 경우 boot가 실패하기 때문에 깨진 파일로 인해 복구가 필요없다. boot는 성공해야 할 때 쓰도록 한다.
<b>기본값</b>	false

• jeus.tm.removeIncompleteTx

<b>설명</b>	복구 대상이 되는 XA 리소스에 대해 복구가 끝나면 복구 리스트에서 삭제할지 여부를 결정한다. 수동으로 복구해야 할 상황을 대비할 경우에는 false로 지정한다.
<b>기본값</b>	false

• jeus.tm.recoveryTrial

<b>설명</b>	복구 실패한 리소스에 대한 복구 재시도 횟수를 설정한다.  복구 재시도는 백그라운드로 구동된다.
<b>기본값</b>	30

• jeus.tm.recoveryInterval

<b>설명</b>	복구 실패한 리소스에 대해 복구를 재시도하는 시간 간격을 설정한다.
<b>기본값</b>	120000 (단위: ms)

- jeus.tm.destroy-timeout

<b>설명</b>	트랜잭션이 timeout(active timeout)이 되었으나 명시적인 commit, rollback을 받지 못하면 트랜잭션 매니저는 타임아웃된 해당 리소스를 계속 유지하게 된다.  이에 대해 cleanup 작업을 해주는 타임아웃을 지정한다. 즉, 타임아웃된 트랜잭션에 대해 이 설정된 값 이상의 시간이 흐르면 내부적으로 destroy 작업을 수행한다.
<b>기본값</b>	1800000 (단위: ms)

- jeus.tm.destroy-timeout-check-interval

<b>설명</b>	destroy timeout을 검사하기 위한 주기를 설정한다.  설정된 주기를 통해 jeus.tm.destroy-timeout의 동작 여부를 확인한다.
<b>기본값</b>	1800000 (단위: ms)

- jeus.tm.ots.recoveryInterval

<b>설명</b>	OTS를 복구할 때 코디네이터가 준비되어 있지 않을 경우 재시도를 하는 시간 간격을 설정한다.
<b>기본값</b>	10000 (단위: ms)

- jeus.tm.ots.decisionTO

<b>설명</b>	전파되고 prepare가 호출된 후에 이 값에 지정된 시간 내에 commit/rollback이 오지 않으면 recovery 코디네이터의 replay_completion을 사용해서 commit/rollback을 한다.
<b>기본값</b>	60000 (단위: ms)

- jeus.tm.net.address-mapping-properties

<b>설명</b>	JEUS TM 간에 통신할 때 사용할 IP 주소를 매핑시키는 프로퍼티 파일을 지정한다. 이에 대한 자세한 설명은 JEUS Server 안내서의 "IP 대역이 다른 서버 간에 트랜잭션 통신 문제"를 참고한다.
<b>기본값</b>	프로퍼티 파일의 위치를 지정하지 않으면 이 기능을 사용하지 않는다.

- jeus.tm.profile.classes

<b>설명</b>	TM Profile을 위해 사용되는 ProfileListener 클래스들을 지정한다.  프로파일 기능에 대한 자세한 설명은 JEUS Server 안내서의 "트랜잭션 프로파일 기능"을 참고한다.
<b>기본값</b>	없음

- jeus.tm.ignore.xaEmulation-resume-suspend-exception

<b>설명</b>	<p>XAEmulation을 쓰는 경우 XAResource가 아니므로 suspend, resume 기능을 쓸 수 없다.</p> <p>하지만 간혹 프레임워크 단에서 suspend,resume을 호출하는 경우가 있는데 이때 LocalXAResourceWrapper 인 경우 delist,enlist시도시 Exception을 던지게된다.</p> <p>로직에는 영향을 주지 않으니 LocalXAResourceWrapper인경우 bypass 할 수 있도록 한다.</p>
<b>기본값</b>	false

### 1.6.1. TM 클라이언트를 위한 시스템 프로퍼티

클라이언트에서 TM을 사용할 때에는 XML 설정을 사용할 수 없으므로 시스템 프로퍼티로 설정해야 한다.

- jeus.tm.not\_use

<b>설명</b>	<p>트랜잭션 매니저의 사용 여부를 결정한다.</p> <ul style="list-style-type: none"> <li>• true : 클라이언트의 경우 JNDI를 사용하긴 하나 트랜잭션은 사용하지 않을 경우를 뜻한다.</li> <li>• false : 클라이언트의 경우 JNDI와 트랜잭션을 사용하는 경우를 뜻한다.</li> </ul>
<b>기본값</b>	false

- jeus.tm.version

<b>설명</b>	<p>사용할 TM 런타임 버전을 설정한다.</p> <ul style="list-style-type: none"> <li>• client : 클라이언트 TM 사용</li> <li>• server : 서버 TM 사용</li> </ul>
<b>기본값</b>	client

- jeus.tm.port

<b>설명</b>	JEUS 트랜잭션 매니저가 사용할 TCP/IP 포트 번호를 설정한다.
-----------	--

- jeus.tm.usenio

<b>설명</b>	<p>트랜잭션 매니저 간의 I/O 방식을 결정한다.</p> <ul style="list-style-type: none"> <li>• true : NIO</li> <li>• false : 블로킹 I/O</li> </ul>
<b>기본값</b>	true

- jeus.tm.tmMin

<b>설명</b>	트랜잭션에서 사용하는 Thread Pool의 최소 크기를 설정한다.
-----------	---------------------------------------

<b>기본값</b>	2
------------	---

- jeus.tm.tmMax

<b>설명</b>	트랜잭션에서 사용하는 Thread Pool의 최대 크기를 설정한다.
<b>기본값</b>	30

- jeus.tm.activeto

<b>설명</b>	트랜잭션이 시작되고, commit이나 rollback을 하는 데까지의 제한시간을 설정한다.  설정된 시간 내에 결정이 내려지지 않을 경우 강제로 rollback시킨다.
<b>기본값</b>	600000 (단위: ms)

- jeus.tm.prepareto

<b>설명</b>	루트 코디네이터가 서브 코디네이터로부터 prepare 완료 신호를 받는 데까지의 제한시간을 설정한다.  이 시간 안에 어떤식으로든 결정이 내려지지 않을 경우 강제로 rollback시킨다.
<b>기본값</b>	120000 (단위: ms)

- jeus.tm.preparedto

<b>설명</b>	서브 코디네이터가 루트 코디네이터에게 prepare 완료 신호를 보낸 후 결정을 기다리는 제한시간을 설정한다.  시간이 지날 경우 반복하여 결정 대기 신호를 루트 코디네이터로 보낸다.
<b>기본값</b>	60000 (단위: ms)

- jeus.tm.committo

<b>설명</b>	루트 코디네이터가 서브 코디네이터에게 commit 신호를 보낸 후 결과를 받아야 하는 제한시간을 설정한다. 시간이 경과하면 부결 트랜잭션으로 기록한다.
<b>기본값</b>	240000 (단위: ms)

- jeus.tm.recoveryto

<b>설명</b>	루트 코디네이터가 서브 코디네이터로부터 복구 정보를 받아야 하는 제한시간을 설정한다. 시간이 지나면 더 이상 진행하지 않는다.
<b>기본값</b>	120000 (단위: ms)

- jeus.tm.incomplete.to

설명	부결된 트랜잭션이 보관되는 시간을 설정한다.
기본값	86400000 (단위: ms)

## 1.6.2. 하위 호환성을 위한 시스템 프로퍼티

- jeus.tm.propagation.recalculated-timeout

설명	트랜잭션이 다른 컨테이너로 전파될 때 전파되기 전에 소모한 시간을 Active Timeout에서 감소시키는 설정이다.  JEUS 6 Fix#9부터 제공되는 설정으로 하위 호환성을 위해서는 false로 지정한다.
기본값	false

## 1.7. 웹 엔진 프로퍼티

웹 엔진에서 제공하는 모든 프로퍼티는 JVM 시스템 프로퍼티뿐만 아니라 가상 호스트별, 웹 애플리케이션별로 적용할 수 있다. 이를 적용하는 방법은 JEUS Web Engine 안내서의 "가상 호스트 설정", "jeus-web-dd.xml 설정"을 참고한다.



JVM 시스템 프로퍼티나 웹 엔진 레벨에서 설정하기 보다는 되도록 가상 호스트 또는 웹 애플리케이션 레벨에서 설정하는 것을 권장한다.

### 1.7.1. 웹 엔진 레벨 프로퍼티

domain.xml의 웹 엔진 레벨로 설정할 수 있는 프로퍼티이다. 가상 호스트, jeus-web-dd.xml에 설정해도 무시한다.

- jeus.servlet.collecStatistics

설명	스레드의 통계 정보를 업데이트 할지 설정한다.
기본값	true

- jeus.servlet.prevent-forcibly-threadlocal-memoryleak

설명	웹 애플리케이션에서 ThreadLocal 변수를 사용한 후 이를 정확하게 해제하지 않을 경우 발생하는 메모리 누수를 웹 엔진에서 해당 웹 애플리케이션이 리로드되거나 redeploy, undeploy될 때 강제적으로 ThreadLocal 변수를 초기화한다. 그러나 이 옵션을 사용할 경우 다른 웹 애플리케이션에서 설정한 ThreadLocal 변수도 초기화하므로 주의한다.
기본값	false

- jeus.servlet.request.enableDns



<b>설명</b>	ServletReqeust.getRemoteHost()를 호출할 때 DNS를 통해서 호스트명을 얻어올지 여부를 설정한다. 웹 엔진이 있는 서버의 DNS 접근 여부를 설정하는 것이므로 웹 엔진 레벨 프로퍼티로 제공한다.
<b>기본값</b>	true

- jeus.servlet.request.url.relativePathInURL

<b>설명</b>	URL에 상대 경로('.') 혹은 '..')가 포함됐을 때 URL 계산 방식을 정의한다 <ul style="list-style-type: none"> <li>• not_allowed : URL에 상대 경로가 포함되는 것을 허용하지 않으며 403 Forbidden 에러로 응답</li> <li>• replace : 상대 경로를 절대 경로로 계산 후 처리</li> <li>• keep : 상대 경로를 계산하지 않고 그대로 처리</li> </ul>
<b>기본값</b>	replace

- jeus.servlet.useCaseInsensitiveFileSystem

<b>설명</b>	Linux OS + Windows file server처럼 대소문자 구분 OS에 대소문자 비구분 파일 서버를 사용하는 조합의 경우 요청 주소의 대소문자 구분을 하지 않게되어 원하는 동작을 하지 않을 수 있다. 이 프로퍼티를 사용하면 이런 조합에서도 대소문자를 구분하여 처리하며, 추가적으로 Windows에서만 동작하는 다음 프로퍼티도 사용할 수 있게 된다. <ul style="list-style-type: none"> <li>• jeus.servlet.resource.ignore-filename-case</li> <li>• jeus.servlet.jsp.ignore-filename-case</li> </ul>
<b>기본값</b>	false

- jeus.servlet.allowUrlEncodeJsessionId

<b>설명</b>	cookie header(JSESSIONID)에 대해 URL Encoding을 강제로 진행한다. 이 프로퍼티와 동시에 apply-url-encoding-rule가 true일 경우 유효하다.
<b>기본값</b>	false

- jeus.servlet.validate-2\_5-web-xml-changes

<b>설명</b>	web.xml이 2.5 버전을 기준으로 작성되었는지 검사한다.
<b>기본값</b>	true

- jeus.servlet.jsp.noPackagePrefix

<b>설명</b>	JSP에서 생성되는 Servlet 클래스에 패키지 추가 여부를 설정한다.
<b>기본값</b>	false

- jeus.servlet.response.header.encoding

<b>설명</b>	응답 헤더에 사용할 캐릭터 인코딩을 설정한다.
<b>기본값</b>	null

- jeus.servlet.request.allowContentLengthAndTransferEncodingSimultaneously

<b>설명</b>	JEUS는 transfer-encoding 헤더와 content-length 헤더를 동시에 수신하거나 content-length 헤더를 여러 개 수신한 경우, HTTP Smuggling 공격으로 인식하여 해당 요청을 일괄 차단한다. 단, RFC 7230 상으로는 이와 상충되는 내용이 있으므로, 동시에 수신하더라도 공격으로 인식하지 않도록 설정하는 옵션을 제공한다.
<b>기본값</b>	false

- jeus.servlet.makeThreadDumpBoundarySize

<b>설명</b>	현재 서버에서 동작하는 스레드 개수가 설정 값을 초과하는 경우 덤프를 생성하도록 설정한다.
<b>기본값</b>	0

- jeus.servlet.makeThreadDumpRestartCount

<b>설명</b>	스레드 모니터에 의해 증가하는 restart count가 설정 값을 초과하는 경우 restart를 시도할 수 있게 설정한다.
<b>기본값</b>	0

- jeus.servlet.makeThreadDumpRestartWithoutNodemanager

<b>설명</b>	노드 매니저가 없어도 시스템을 종료할 수 있게 설정한다.
<b>기본값</b>	false

- jeus.servlet.makeThreadDumpMonitoringInterval

<b>설명</b>	스레드 모니터가 jeus.servlet.makeThreadDumpBoundarySize 값을 체크하는 주기를 설정한다.
<b>기본값</b>	0

- jeus.servlet.makeThreadDumpIntervalSecond

<b>설명</b>	스레드 모니터뿐만 아니라 새로운 스레드가 시작 전에 jeus.servlet.makeThreadDumpBoundarySize 값을 체크하여 스레드 덤프를 생성하도록 설정한다.
<b>기본값</b>	300

- jeus.servlet.ignoreUserInfoInRequestURI

<b>설명</b>	RFC 7230에 따르면 request-uri에 user info가 포함된 경우 에러로 처리하라고 명시되어 있으나, 이전 JEUS 버전에서는 해당 처리 없이 request uri로 인식하였다. 이에 따라 request-uri에 user info가 포함된 경우 에러를 발생시키도록 수정하였으며, 하위 호환성을 위해 user info가 포함되어 있더라도 에러를 발생시키지 않는 옵션을 제공한다.
<b>기본값</b>	false

## 1.7.2. JVM 시스템 프로퍼티

다음은 JVM 시스템 프로퍼티만으로 설정해야 하는 옵션이다.

- jeus.websocket.client.requestBufferSize

<b>설명</b>	WebSocketContainer.connectToServer() 호출 시 웹 소켓 헤더들을 포함한 요청을 저장할 버퍼 크기를 설정한다.
<b>기본값</b>	8*1024 (단위: byte)

- org.glassfish.web.rfc2109\_cookie\_names\_enforced

<b>설명</b>	Cookie 이름에 RFC2109에서 허용하지 않는 문자가 들어간 경우 IllegalArgumentException이 발생한다. jakarta.servlet.http.Cookie에 정의된 내용인데 JEUS 6까지는 API 구현에서 제대로 처리하지 않았다.
<b>기본값</b>	true

- jeus.servlet.wjpReconnectTime

<b>설명</b>	WJP 재연결 시 WebtoB는 서버로부터 마지막 ACK 또는 RST를 수신해야 연결이 끊겼다고 인지한다.  타이밍에 따라 해당 응답을 받기 전에 재연결이 시도되는 문제가 발생할 수 있으며, 이를 방지하기 위해 대기 시간을 설정한다.
<b>기본값</b>	150ms

- jeus.websocket.maxMessagePartQueueSize

<b>설명</b>	WebSocket Session의 RemoteEndpoint.Async를 사용하여 메시지를 쓰는 경우, 메시지를 저장한 Queue의 최대 크기를 제한한다. Async 방식에서는 메시지가 Queue에 순차적으로 적재되며, 하나씩 출력된다. 이 때 최대로 메시지를 저장할 수 있는 Queue의 크기이다.  타이밍에 따라 해당 응답을 받기 전에 재연결이 시도되는 문제가 발생할 수 있으므로, 이를 방지하기 위해 재연결 전 대기 시간을 설정한다.
<b>기본값</b>	Integer.MAX_VALUE

### 1.7.3. 가상 호스트 레벨 프로퍼티

domain.xml 웹 엔진 설정에서 가상 호스트에 설정할 수 있는 프로퍼티이다. jeus-web-dd.xml에 설정해도 무시한다.



현재 이 레벨의 프로퍼티는 존재하지 않는다.

### 1.7.4. 웹 애플리케이션 레벨 프로퍼티

jeus-web-dd.xml에 설정해서 각 웹 애플리케이션별로 적용할 수 있는 프로퍼티이다. 이 프로퍼티는 웹 엔진 또는 가상 호스트 레벨 프로퍼티로 사용 가능하다.

#### 1.7.4.1. 공통 또는 Servlet 엔진 관련 프로퍼티

Servlet 또는 JSP 엔진에 적용되는 프로퍼티들이다.

- jeus.servlet.response.header.serverInfo

설명	헤더에 JEUS 버전 정보 포함 여부를 설정한다.
기본값	false

- jeus.servlet.keep-original-query-string-on-forward

설명	Forward될 때 original query string을 Request 객체에 그대로 둘지 아니면 forward할 때 넘겨준 query string으로 overwrite할지 여부를 설정한다.
기본값	false

- jeus.servlet.keep-original-query-string-on-include

설명	include될 때 original query string을 Request 객체에 그대로 둘지 아니면 include할 때 넘겨준 query string으로 overwrite할지 여부를 설정한다.
기본값	false

- jeus.servlet.out.ensureContentOrder

설명	response.getOutputStream(), response.getWriter(), JspWriter(jsp out)을 섞어서 사용할 때의 출력 순서를 유지할지 여부를 설정한다. 기본적으로는 세 가지 방법을 섞어서 사용하면 안 되고 하나로 통일해서 써야 한다.
기본값	false

- jeus.servlet.classloader.allowServletAPI

<b>설명</b>	애플리케이션의 클래스 중에서 서블릿 API 관련 클래스를 로딩할지 여부를 설정한다.
<b>기본값</b>	false

- jeus.servlet.context.attribute.serialize

<b>설명</b>	Context Auto reload할 때 ServletContext attribute를 serialize하여 유지할지 여부를 설정한다.
<b>기본값</b>	false

- jeus.servlet.resource.ignore-filename-case

<b>설명</b>	ResourceServlet에서 리소스 파일 이름의 대소문자 구별 여부를 설정한다.  Windows OS 혹은 jeus.servlet.useCaseInsensitiveFileSystem이 true인 경우 유효하다.
<b>기본값</b>	false

- jeus.servlet.scan-servlet-container-initializers

<b>설명</b>	Servlet 3.0부터 추가된 ServletContainerInitializer 메커니즘 사용 여부를 결정한다.  Servlet 표준을 준수하기 위하여 기본적으로 scan 작업을 수행하지만 이로 인해서 디플로이 시간이 늘어날 수 있다.
<b>기본값</b>	true

- jeus.servlet.sortWebinfLibraries

<b>설명</b>	WEB-INF/lib에 포함된 jar 파일들을 로딩할 때 정렬을 하는 옵션이다.  <ul style="list-style-type: none"> <li>• name_asc : 파일 이름으로 오름차순 정렬</li> <li>• name_dsc : 파일 이름으로 내림차순 정렬</li> <li>• time_asc : 파일의 LMF로 오름차순 정렬</li> <li>• time_dsc : 파일의 LMF로 내림차순 정렬</li> </ul>
<b>기본값</b>	null

- jeus.servlet.useMetadataCompleteDeploy

<b>설명</b>	Java EE 5부터는 웹 애플리케이션 디플로이 시점에 반드시 Java EE 표준에서 정의한 어노테이션들이 정의된 클래스를 찾도록 되어 있다. WEB-INF/classes의 클래스 파일들 뿐만 아니라 WEB-INF/lib에 포함된 것들도 찾아야 하기 때문에 디플로이 시간이 오래 걸릴 수 있다.  찾는 동작을 하지 않기 위해서는 web.xml의 metadata-complete 설정을 true로 지정해야 한다. 애플리케이션에 포함된 web.xml을 수정하지 않고 jeus-web-dd.xml을 추가해서 metadata-complete 설정을 true로 할 수 있다.
-----------	--

<b>기본값</b>	false
------------	-------

- jeus.servlet.jsp.allowOnlyStandardizedMethod

<b>설명</b>	JSP 또는 리소스 요청에 대해서 GET, HEAD, POST, OPTIONS, PUT, DELETE, TRACE, CONNECT의 8개의 HTTP method만 허용한다.
<b>기본값</b>	false

- jeus.servlet.postdataParsingForSPEC

<b>설명</b>	Servlet 표준에 따라 POST 메소드인 경우에만 application/x-www-form-urlencoded을 파싱한다.
<b>기본값</b>	true

- jeus.servlet.response.forceChunkedResponse

<b>설명</b>	모든 http 응답 header를 transfer-encoding :chunked로 나가게 한다.
<b>기본값</b>	false

- jeus.servlet.tempDir

<b>설명</b>	app에서 사용할 temp dir을 지정할 수 있다.
<b>기본값</b>	null

- jeus.servlet.response.header.include-content-type-charset

<b>설명</b>	Response 헤더 내 Content-Type에 charset을 포함시킨다.
<b>기본값</b>	true

- jeus.servlet.request.showDefaultPortInRequestURL

<b>설명</b>	JEUS 7 이전 버전에서는 프로토콜이 기본 포트(http: 80, https: 443)로 요청을 보냈을 경우 포트를 삭제하고 응답을 만들었으나 JEUS 7부터 이를 옵션으로 설정이 가능하다.
<b>기본값</b>	false

- jeus.servlet.response.usePercentEncodedLocationHeaderWhenRedirect

<b>설명</b>	IE에서 요청이 수신된 경우 redirect 시 서버의 인코딩으로 location 헤더를 생성한다. 문제는 IE 요청의 인코딩 방식이 서버의 기본 인코딩 방식과 다를 경우 location 헤더의 query string을 제대로 인식하지 못할 수 있다. 이를 해결하기 위해 query string을 퍼센트 인코딩으로 변환하여 재작성하는 옵션을 제공한다.
<b>기본값</b>	false

- jeus.servlet.response.sendRedirect.useRelativeRedirects

<b>설명</b>	RFC 7231에 따르면 location 헤더에 절대 경로 외에 상대 경로도 허용하도록 변경되었으나, 서블릿 스펙에서는 절대 경로를 사용하도록 명시되어 있다. 이에 따라 상대 경로도 허용하는 옵션을 제공한다.
<b>기본값</b>	false

- jeus.servlet.response.sendError.forceFlushBufferWhenInternalServerError

<b>설명</b>	OutputStream 또는 Writer를 사용하여 쓰기(write) 도중 500 응답이 발생한 경우, 버퍼에 남아 있는 내용을 강제로 flush 할 수 있는 옵션을 제공한다.
<b>기본값</b>	false

- jeus.servlet.loader.ignoreAutoReloadAfterHotSwapFailed

<b>설명</b>	이전 JEUS 버전에서는 JEUS Hot Swap 옵션이 적용되어 있을 때 클래스가 하나라도 실패하면 auto-reload(context 전체 reload)를 진행했으나 이를 옵션으로 설정할 수 있게 제공한다.
<b>기본값</b>	false

- jeus.servlet.security.sendRedirectForFailedLoginForm

<b>설명</b>	폼(form)으로 인증하는 경우, 로그인 또는 에러 페이지를 출력할 때 forward 대신 sendRedirect를 사용할지 여부를 설정한다.
<b>기본값</b>	false

#### 1.7.4.2. JSP 엔진 관련 프로퍼티

JSP 엔진에서만 사용하는 프로퍼티들이다.

- jeus.servlet.jsp.reload

<b>설명</b>	JSP 리로딩 지원 여부를 설정한다.
<b>기본값</b>	true

- jeus.servlet.jsp.precompile

<b>설명</b>	JSP precompile 지원 여부를 설정한다. (precompile 요청 파라미터에 따라 JSP를 호출할 때 compile만 하는 기능)
<b>기본값</b>	true

- jeus.servlet.jsp.print.null.as.emptystring

<b>설명</b>	JSP에서 null로 출력할 때 공백 문자의 사용 여부를 설정한다.
<b>기본값</b>	false

- jeus.servlet.jsp.jspwriter.null

<b>설명</b>	JSP에서 null로 출력할 때 대체할 문자열을 설정한다.
<b>기본값</b>	"null"

- jeus.servlet.jsp.compile.retrycount

<b>설명</b>	JSP에서 생성된 서블릿을 로딩할 수 없을 때 재컴파일을 시도하는 횟수를 설정한다.
<b>기본값</b>	2

- jeus.servlet.jsp.compile.delay

<b>설명</b>	FTP 업로드할 때 JSP 지연 컴파일 시간을 설정한다.  동일한 기능의 jeus.jsp.compile.delay는 현재 버전에서는 삭제되어 사용할 수 없다.
<b>기본값</b>	0L (단위: ms)

- jeus.servlet.jsp.helpReloadNativeLibrariesByCallingSystemGC

<b>설명</b>	JSP를 reload할 때 JSP class loader가 사용한 Native Library가 제대로 cleanup되지 않는 JVM 문제를 해결하기 위한 방법이다.  JSP class loader가 GC(Garbage Collection)될 때 Native Library가 정리되기 때문에 JEUS에서는 System.gc()를 호출해 줄 수 있을 뿐이며, 100% GC된다는 보장이 없다. 따라서 해당 JSP 내에서 System.loadLibrary() 등을 사용하는 경우 reload가 실패할 수 있다.
<b>기본값</b>	false



System.gc()와 System.runFinalization()을 호출하므로 시스템 성능을 저하시킬 수 있다. 따라서 Native Library를 사용하는 JSP를 실제 서비스 운영 중에 수정하지 않기를 권장한다. 가장 좋은 방법은 Native Library를 사용하는 웹 애플리케이션을 별도로 작성해서 그쪽으로 dispatch를 하는 방법이 있다. 그 웹 애플리케이션은 Context Auto Reload나 redeploy하지 않아야 한다.

- jeus.servlet.jsp.ignore-filename-case

<b>설명</b>	JSP 파일 이름의 대소문자 구별 여부를 설정한다. Windows OS 혹은 jeus.servlet.useCaseInsensitiveFileSystem이 true인 경우 유효하다.
<b>기본값</b>	false



- jeus.servlet.jsp.assure-utf8-file-encoding-detection

<b>설명</b>	Jasper에서 JSP 파일 Encoding이 UTF-8이라고 확신할 수 없더라도 이를 UTF-8으로 취급한다.
<b>기본값</b>	false



included JSP 파일에도 pageEncoding tag를 기술해야 한다.

- jeus.servlet.jsp.compile-java-source-concurrently

<b>설명</b>	다수의 JSP 파일에 대한 컴파일을 동시에 진행할 것인지에 대한 여부를 설정한다.
<b>기본값</b>	false



사용하는 JSP 컴파일러에 동시성 이슈가 없음이 보장될 때만 사용해야 한다.

- jeus.servlet.jsp.compileConcurrencyLevel

<b>설명</b>	다수의 JSP 파일을 동시에 컴파일할 때 사용할 스레드 개수를 설정한다. jeus.servlet.jsp.compile-java-source-concurrently가 true인 경우 유효하다.
<b>기본값</b>	1

- jeus.servlet.jsp.development-mode

<b>설명</b>	JEUS 내 JSP의 development mode를 사용하도록 설정한다. 이 mode를 true로 설정하면, jsp와 tag 파일이 모두 수정될 경우 tag 파일을 재컴파일한다.
<b>기본값</b>	false

- jeus.servlet.jsp.preventByteOrderMarkJspFileByForcedResponseEncoding

<b>설명</b>	<response-encoding><forced> 설정 값이 UTF-8이 아닐 경우 BOM이 적용된 JSP 파일을 허용하지 않도록 한다. 예러 메시지로 BOMJspFileNotAllowedException이라는 별도의 exception을 던진다.
<b>기본값</b>	false

- jeus.servlet.jsp.ignore-self-closing

<b>설명</b>	XML 형식의 파일 파싱 시 element의 self-closing 여부(<script .../>)를 설정한다. true이면 self-closing하지 않는다.
<b>기본값</b>	false

- jeus.servlet.jsp.enforceJspRecompile

<b>설명</b>	JSP 변경 여부에 관계없이 항상 재컴파일한다.
<b>기본값</b>	false

- jeus.servlet.jsp.jspParam.urlEncode

<b>설명</b>	URL의 파라미터에 대한 인코딩 여부를 설정한다.
<b>기본값</b>	true

- jeus.servlet.jsp.include.enableFlush

<b>설명</b>	JSP에서 PageContext.include한 경우 무조건 flush하도록 하는 설정이다.
<b>기본값</b>	false

- jeus.servlet.jsp.ignore-taglib-setter-name-case

<b>설명</b>	taglib의 setter 메서드의 이름을 대문자로 변경하지 않는 설정이다.
<b>기본값</b>	false

- jeus.servlet.jsp.removeJspClassPathExceptContextPath

<b>설명</b>	ejb shared 상태에서 JSP 컴파일이 오래 걸리는 버그가 있었고, 주요 원인은 classpath를 받아오는 동작에서 시간이 오래 걸렸던 것으로 추정된다. 따라서 JSP classpath를 직접 관리하여 이 시간을 줄일 수 있고 이를 위해 classpath를 삭제할 수 있는 옵션을 제공한다. 삭제 시 classpath의 일부만 넣어도 된다.
<b>기본값</b>	null

- jeus.servlet.jsp.addJspClassPath

<b>설명</b>	ejb shared 상태에서 JSP 컴파일이 오래 걸리는 버그가 있었고, 주요 원인은 classpath를 받아오는 동작에서 시간이 오래 걸렸던 것으로 추정된다. 따라서 JSP classpath를 직접 관리하여 이 시간을 줄일 수 있고 이를 위해 classpath를 추가할 수 있는 옵션을 제공한다. 추가 시 classpath 전체를 넣어야 한다.
<b>기본값</b>	null

- jeus.servlet.jsp.taglib.scanTldExceptWEBINF

<b>설명</b>	WEB-INF 상에 과도한 양의 파일을 포함시키면 Tld 스캔 시 너무 오래 걸리거나 StackOverflowError가 발생하기도 한다. 따라서 WEB-INF 폴더 스캔을 생략할 수 있도록 옵션을 제공한다.
<b>기본값</b>	false

### 1.7.4.3. 호환성 제공 프로퍼티

JEUS는 기본적으로 Servlet 및 JSP 표준을 준수해야 한다. 하지만 표준 해석 오류, 요구 사항에 대한 구현 누락 등과 같은 개발자의 실수 또는 불분명한 설명, 표준 기술 문서와 API 문서 간의 설명 불일치 등으로 인한 표준 문서의 오류로 인해 API 동작이 달라질 수 있다.

- jeus.servlet.jsp.modern

설명	false로 설정하는 경우 JEUS 4 호환 JSP 엔진을 사용한다. 이 경우 JSP 1.2 버전만 지원된다. 해당 JSP 엔진은 deprecated 되었으므로 더이상 사용하지 않기를 권장하며 본 호환성 프로퍼티도 차기 버전 또는 차기 Fix에서 삭제될 예정이다.
기본값	true

- jeus.servlet.jsp.compileEncoding

설명	JSP 컴파일 시 사용할 인코딩 값을 지정한다. jeus.servlet.jsp.modern이 false일 때만 사용 가능하다.
기본값	null

- jeus.servlet.request.url.allowNonStandardPercentCharacterUsage

설명	URL에 단독으로 '%' 문자 사용을 허용한다. 하지만 이는 RFC 2396, 2.4.3. Excluded US-ASCII Characters에 따르면 '%' 문자는 URL escape를 위한 예약어이며, 결코 단독 문자로 사용할 수 없고 정해진 형식으로만 사용 가능하다. 따라서 클라이언트에서 이를 '%25'로 escape해서 보내야 한다.
기본값	false

- jeus.servlet.request.6CompatibleSetCharacterEncoding

설명	ServletRequest.setCharacterEncoding()으로 세팅한 인코딩은 반드시 Request Body에만 적용해야 한다. 그러나 JEUS 6에서 이를 준수하지 않은 버그가 발견되어, Query String, Cookie와 같이 Request Header를 읽을 때도 사용한다. JEUS 6의 잘못된 동작에 기반하여 작성한 애플리케이션을 수정없이 지원해야 하는 경우 이 옵션을 사용해야 한다.
기본값	false

- jeus.servlet.request.useResetCharacterEncoding

설명	이미 한번 인코딩된 값을 다른 인코딩 방식으로 재인코딩 하기 위해 사용한다. ServletRequest.setCharacterEncoding()은 getParameter() 또는 getReader() 호출 후에는 비활성화 시키도록 되어있어 재인코딩이 불가능하지만 WebLogic 등의 다른 WAS에서 이를 허용하는 경우가 있어 원백 시 JEUS도 지원할 수 있도록 한다.
기본값	false

- jeus.servlet.response.defaultContentType

<b>설명</b>	Content-Type 응답 헤더에 설정할 기본 값이다. 이 값이 null이 아닐 경우 아래와 같은 상황에서 응답 헤더를 설정한다.  ServletResponse.setCharacterEncoding()이나 ServletResponse.setLocale() 호출할 때, Servlet이 명시적으로 ServletResponse.setContentType()을 호출하지 않았다면 이 프로퍼티 값을 Content-Type 헤더로 설정한다. 하지만 이러한 동작은 Content-Type 응답 헤더에 관한 Servlet 표준에 맞지 않다.
<b>기본값</b>	null



서블릿이 명시적으로 ServletResponse.setContentType()을 호출하지 않았다면 응답 헤더에 Content-Type Header를 넣으면 안 된다고 서블릿 표준에서 규정하고 있다. 또한 ServletResponse.setContentType()을 호출할 때에는 null을 리턴하도록 하고 있다. 이는 Servlet 5.0에서도 명시된 내용으로 JEUS에서 오래 전부터 지키지 않고 있던 사항이다.

따라서 JEUS 7 Fix#1부터는 이를 준수하기 위해 기본값을 null로 변경하였다. 만약 여기에 의존해서 애플리케이션이 동작하고 있었다면 이 프로퍼티 값을 적절한 값으로 설정하기 바란다.

- jeus.servlet.response.applyContentLanguageProperlyWithSetLocaleAPI

<b>설명</b>	ServletResponse.getWriter() 호출 이후에 ServletResponse.setLocale()을 호출했을 때, Content-Language 응답 헤더는 반드시 설정되어야 한다. 물론 Character Encoding은 바뀌지 않는다. 만약 원하지 않게 Content-Language 응답 헤더가 설정되는 경우에는 이값을 false로 지정한다.
<b>기본값</b>	true

- jeus.servlet.response.6CompatibleForcedEncoding

<b>설명</b>	xml 설정으로 적용하는 <response-encoding><forced>의 적용 우선 순위를 원래의 의미에 맞지 않게 구현된 JEUS 6 방식으로 동작하게 한다. 원래 forced 옵션은 우선 순위가 가장 높다. 애플리케이션 API를 통해서 세팅하거나 JSP 파일의 page 태그 contentType에 설정된 HTTP 응답 인코딩에 우선한다.
<b>기본값</b>	false

- jeus.servlet.response.6CompatibleSetCharacterEncoding

<b>설명</b>	ServletResponse.setCharacterEncoding()으로 세팅한 인코딩은 forced 보다 우선 순위가 낮아야 한다. 그러나 JEUS 6 이전까지는 이를 준수하지 않은 버그가 발견되어, 항상 forced 설정을 무시하는 문제가 있다. 만약 이러한 동작에 의존해서 구현된 애플리케이션이 있고 이 동작을 유지하고자 한다면 이 옵션을 true로 설정한다.  Request 쪽과 이름이 유사하므로 서로 헷갈리지 않도록 한다.
<b>기본값</b>	false

- `jeus.servlet.response.allowBothWriterAndOutputStream`

<b>설명</b>	<code>HttpServletResponse.getWriter()</code> 와 <code>HttpServletResponse.getOutputStream</code> 을 동시에 사용할 수 있도록 하는 옵션이다. 스펙에는 <code>reset()</code> 호출시에만 서로 다른 메소드를 사용할 수 있도록 허용하고 있지만, 해당 옵션으로 동시에 사용할 수 있도록 한다.
<b>기본값</b>	false

- `jeus.servlet.session.invalidateBySetMaxInactiveIntervalZero`

<b>설명</b>	<p><code>HttpSession.setMaxInactiveInterval(int timeout)</code>의 경우 Spec에 의해 0 이하의 값은 제거되지 않아야 하는 세션을 의미한다.</p> <p>그러나 이전 스펙에서 모호하게 타임아웃만큼 유지되는 시간으로 설명이 되어 "0"을 <code>invalidate</code>로 사용하는 유저들이 있었다.</p> <p>이러한 의미로 사용했던 유저들의 호환성을 위해 설정을 제공한다.</p>
<b>기본값</b>	false

- `jeus.servlet.engine.disableRequestFinishCheck`

<b>설명</b>	<p>Request 객체는 서블릿의 <code>service</code> 메소드 내부 혹은 필터의 <code>doFilter</code> 메소드 안에서만 유효하며 이미 처리가 끝난 Request 객체를 사용하려고 하면 <code>IllegalStateException</code>을 발생시키도록 스펙에 정의 되어 있다. 그러나 JEUS 7 이전까지는 이를 확인하지 않고 있어서 처리가 끝난 Request 객체를 사용하는 애플리케이션이 정상적으로 동작하였다.</p> <p>이러한 애플리케이션을 사용하는 유저들의 호환성을 위해 설정을 제공한다.</p>
<b>기본값</b>	false

- `jeus.servlet.ignoreWebComponentFailureWhenDeploying`

<b>설명</b>	디플로이 과정에서 <code>load-on-startup</code> 으로 등록된 서블릿의 <code>init()</code> , <code>Filter.init()</code> , <code>ServletContextListener</code> 에서 <code>Exception</code> 이 발생하더라도 디플로이가 실패하지 않는다.
<b>기본값</b>	false

- `jeus.servlet.classloader.spec23`

<b>설명</b>	SRV.9.7.2의 권고대로 WEB-INF 아래의 클래스를 컨테이너 전체에 적용되는 클래스보다 먼저 로딩한다.
<b>기본값</b>	false

- `jeus.servlet.ensure.contentlength`

<b>설명</b>	HTTP 1.0 일 때 <code>content-length</code> 를 항상 보내도록 설정한다.
<b>기본값</b>	false

- jeus.servlet.query.encoding.Enabled

<b>설명</b>	HttpServletRequest#getQueryString() return 값에 encoding을 적용하도록 설정한다.
<b>기본값</b>	false

- jeus.servlet.jsp.superclass

<b>설명</b>	WebLogic의 경우 weblogic.xml에 JSP 파라미터로 상속 구조를 지정할 수 있으나 JEUS는 org.apache.jasper.runtime.HttpJspBase를 사용한다. 이 값은 시스템 프로퍼티로는 변경이 가능하지만 애플리케이션 레벨에서는 변경이 불가능하기 때문에 애플리케이션 단위로 설정할 수 있는 옵션을 제공한다.
<b>기본값</b>	org.apache.jasper.runtime.HttpJspBase

- jeus.servlet.request.reuseInputStreamAfterGetParameter

<b>설명</b>	서블릿 스펙에서는 inputStream을 두 번 읽어들이지 못하게 하고 있으나 원백 시 호환성 문제가 생기는 경우가 있어 옵션을 제공한다.
<b>기본값</b>	false

- jeus.servlet.request.parameter.UnescapeJavascriptEscapedParameter

<b>설명</b>	resin과 인코딩 호환성 옵션을 제공한다. 참고로 스펙 위반이다.
<b>기본값</b>	false

- jeus.servlet.response.allow-write-after-forward

<b>설명</b>	Forward에서 돌아온 후 response body를 다시 write할 수 있도록 하는 설정이다.
<b>기본값</b>	false

- jeus.servlet.request.returnDecodedRequestURI

<b>설명</b>	HttpServletRequest.getRequestURI()의 값을 디코딩한 후에 리턴하도록 하는 설정이다.
<b>기본값</b>	false

- jeus.servlet.response.sendRedirect.statusCode301

<b>설명</b>	sendRedirect시 status code를 301로 응답하는 설정이다.
<b>기본값</b>	false

- jeus.servlet.deployment.descriptor.checkTransportGuarantee

설명	web.xml에 <transport-guarantee>의 값이 허용되지 않는 값인 경우 에러를 발생시키는 설정이다.
기본값	false

- jeus.servlet.deployment.descriptor.webXmlIgnoreJspFileNull

설명	web.xml이 3.1버전보다 낮은 경우, <servlet> 내에 <servlet-class>와 <jsp-file>이 모두 없을 때 발생하는 에러를 무시할지 설정한다.
기본값	true

## 1.8. 세션 서버 시스템 프로퍼티

본 절에서는 세션 서버 시스템 프로퍼티에 대해 설명한다.

아래의 프로퍼티들은 <session-cluster> 절 하부 <properties>에 설정하는 것을 기본으로 한다.

- jeus.sessionmanager.receive.backup.despite.no.deploy

설명	DOMAIN_WIDE 모드일 경우 애플리케이션을 디플로이하지 않더라도 리모트로부터 백업받을 수 있도록 더미 애플리케이션을 디플로이하도록 한다.
기본값	false

- jeus.sessionmanager.cluster.stable.waiting.sec

설명	클러스터의 이벤트 처리를 보장하기 위해 일정 시간(초) 애플리케이션의 디플로이를 지연시킨다.
기본값	0 (초)

- jeus.sessionmanager.dumpbackup.max

설명	dumpbackup을 수행하는 count 단위를 설정한다.  자신이 백업할 서버가 재기동할 경우 dumpbackup을 수행한다. 이때 가지고 있는 세션이 너무 많을 경우에 일정 단위로 나눠서 백업을 수행하도록 한다.  Session Cluster의 Properties에 설정할 때는 모든 Cluster에 적용되며, Session Cluster Config의 properties에 적용할 해당 클러스터에만 적용된다.
기본값	100 (개)

- jeus.distributed.session.failback.unit.size

설명	failback을 보낼 때 한 번에 보낼 session 개수를 설정한다.
기본값	100 (개)

- jeus.distributed.session.clear-backup-delay

<b>설명</b>	클러스터링된 서버에서 join, fail, stop 이벤트가 발생하고, 몇 초 후에 backup session을 정리할지 설정한다.  만약 backup session이 정리되기 전 join, fail, stop 이벤트가 발생하면 마지막 이벤트가 발생한 시점을 기준으로 한다.
<b>기본값</b>	30초

- jeus.distributed.session.clear.upgrade.table

<b>설명</b>	fail, stop 이벤트가 발생하여 session upgrade가 실행되면 몇 초 후에 upgrade history를 지울지 설정한다.  만약 upgarde history를 지우기 전 fail, stop 이벤트가 발생하면 마지막 이벤트가 발생한 시점을 기준으로 한다.
<b>기본값</b>	session timeout 설정 값

## 1.9. EJB 시스템 프로퍼티

본 절에서는 EJB 시스템 프로퍼티에 대해 설명한다.

### 1.9.1. EJB 시스템 프로퍼티

다음은 EJB 시스템 프로퍼티에 대한 설명이다.

- jeus.ejb.transaction.attribute.default



<b>설명</b>	<p>Container-managed Transaction Bean의 메소드에 Transaction 속성이 설정되어 있지 않는 경우 사용할 속성을 설정한다.</p> <p>다음 중에 하나를 설정한다.</p> <ul style="list-style-type: none"> <li>• NotSupported</li> <li>• Supports</li> <li>• Required</li> <li>• RequiresNew</li> <li>• Mandatory</li> <li>• Nerver</li> </ul> <p>속성의 우선순위는 다음과 같다.</p> <ol style="list-style-type: none"> <li>1. ejb-jar/trans-attribute</li> <li>2. jeus-ejb-dd/unspecified-container-transaction</li> <li>3. 시스템 프로퍼티</li> </ol>
<b>기본값</b>	<ul style="list-style-type: none"> <li>• EJB 3.0 Bean 이상의 경우 : Required(스펙 기준)</li> <li>• EJB 3.0 미만의 Bean의 경우 : Supports(하위 호환성을 위함)</li> <li>• Timer Service 메소드의 경우 : NotSupported</li> </ul>

• jeus.ejb.sharedcolumn.modify

<b>설명</b>	CMP Entity Bean에서 column sharing되어 있는 cmp, cmr 필드를 하나의 트랜잭션에서 모두 수정하는 것을 허용할지 여부를 결정한다.
<b>기본값</b>	false

• jeus.ejb.checkTable

<b>설명</b>	CMP Entity Bean을 deploy할 때 데이터베이스의 테이블이 존재하는지 컬럼이 제대로 존재하는지를 확인할지 여부를 설정한다.
<b>기본값</b>	false

• jeus.ejb.enable.configDeleteOption

<b>설명</b>	<p>CMP Entity Bean을 undeploy할 때 데이터베이스 테이블을 삭제하는 것을 허용할지 여부를 설정한다.</p> <ul style="list-style-type: none"> <li>• true : &lt;deleting-table&gt;의 설정이 적용된다.</li> <li>• false : &lt;deleting-table&gt;의 설정이 무시된다.</li> </ul>
<b>기본값</b>	false

- jeus.ejb.csi.trusthosts

<b>설명</b>	CORBA CSIV2를 사용할 때의 Trust 호스트이다.
-----------	----------------------------------

- jeus.client.interop

<b>설명</b>	클라이언트 JVM에서 CORBA interoperability를 사용할지 여부를 설정한다.
<b>기본값</b>	false

- jeus.corba.WorkerThreadInactivityTO

<b>설명</b>	corba WorkerThread의 inactive timeout을 지정한다.
<b>기본값</b>	120000 ms

- jeus.corba.minWorkerThreads

<b>설명</b>	corba WorkerThreadPool에서의 최소 유지 스레드 개수를 설정한다.
<b>기본값</b>	10

- jeus.corba.maxWorkerThreads

<b>설명</b>	corba WorkerThreadPool에서의 최대 생성 가능 스레드 개수를 설정한다.
<b>기본값</b>	1000

- jeus.corba.iiop.tcpConnectTO

<b>설명</b>	corba iiop 통신시 생성된 커넥션의 connect timeout을 설정한다. Cumulative Waiting 방식으로 동작하고 콜론(:)으로 구분된 4개의 int(단위 ms) 값으로 설정하며 각각은 (연결 실패 이후 재시도까지)최초 대기 시간, 최대 누적 대기 시간, 대기 시간 증가율, 최대 단일 대기 시간을 의미한다.
<b>기본값</b>	1000:5000:20:2500

- jeus.corba.iiop.tcpReadTO

<b>설명</b>	corba iiop 통신시 생성된 커넥션의 read timeout을 설정한다. Cumulative Waiting 방식으로 동작하고 콜론(:)으로 구분된 4개의 int(단위 ms) 값으로 설정하며 각각은 (연결 실패 이후 재시도까지)최초 대기 시간,최대 누적 대기 시간, 대기 시간 증가율, 최대 단일 대기 시간을 의미한다.
<b>기본값</b>	1000:5000:20:2500

- jeus.corba.useMetaOperation

<b>설명</b>	CORBA/IIOP 스펙에 존재하는 meta operation 수행 여부를 설정한다. meta operation은 클라이언트와 서버간 EJB 메소드 파라미터 클래스의 필드가 서로 다른 경우 이를 해소하기 위해 수행된다.
<b>기본값</b>	true

- jeus.corba.clientORBPort

<b>설명</b>	스탠드어론 클라이언트에서 사용할 ORB Port를 설정한다. 클라이언트와 서버간 EJB 메소드 파라미터 클래스의 필드 차이로 인한 meta operation이 수행될때, 서버에서 해당 클라이언트 포트로 요청한다. 기본값 0으로 설정된 경우 random port를 사용한다.
<b>기본값</b>	0

- jeus.ejb.csi.defaultUser

<b>설명</b>	CORBA CSIv2를 사용할 때 기본 사용자를 설정한다.
-----------	----------------------------------

- jeus.ejb.notreload

<b>설명</b>	shared 클래스 로더를 사용하는 EJB가 deploy, undeploy, redeploy될 때 reload에서 제외될 웹 모듈 리스트이다.  다음 형식으로 설정한다.  <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">context_name1;context_name2</div>
-----------	--

- jeus.corba.props

<b>설명</b>	EJB에서 사용하는 ORB의 프로퍼티이다.  <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">key=value[,key=value]*</div>
-----------	--

- jeus.ejb.cluster.useip

<b>설명</b>	EJB에서 Active Stub을 사용하는 경우 서버로 접속하기 위한 주소를 IP 주소로 할지 호스트명으로 할지 여부를 설정한다. 여기에서 사용하는 IP는 jeus.net.localip로 지정된 IP를 사용한다.  이 옵션이 true가 되면 classftp의 주소도 IP 주소를 사용하게 된다.
<b>기본값</b>	false

- jeus.ejb.all.disconnect

<b>설명</b>	<p>모든 Bean 타입에 적용될 disconnect timeout을 설정한다.</p> <p>-1로 설정하면 disconnect 기능을 사용하지 않는다.</p> <p>[Deprecated] EJB 3.1 스펙에서는 @StatefulTimeout 또는 ejb-jar.xml의 &lt;stateful-timetout&gt;으로 설정 가능하다.</p>
<b>기본값</b>	3600000 (단위: ms)

- jeus.ejb.entity.disconnect

<b>설명</b>	<p>Entity Bean에 적용될 disconnect timeout을 설정한다. (단위 : ms)</p> <p>jeus.ejb.all.disconnect 설정에 우선한다.</p> <p>-1로 설정하면 disconnect 기능을 사용하지 않는다.</p>
-----------	---

- jeus.ejb.stateful.disconnect

<b>설명</b>	<p>Stateful Session Bean에 적용될 disconnect timeout을 설정한다. (단위: ms)</p> <p>jeus.ejb.all.disconnect 설정에 우선한다.</p> <p>-1로 설정하면 disconnect 기능을 사용하지 않는다.</p> <p>[Deprecated] EJB 3.1 스펙에서는 @StatefulTimeout 또는 ejb-jar.xml의 &lt;stateful-timetout&gt;으로 설정 가능하다.</p>
-----------	--

- jeus.ejb.all.passivate

<b>설명</b>	<p>모든 Bean type에 적용될 passivation timeout을 설정한다.</p> <p>-1로 설정하면 disconnect 기능을 사용하지 않는다.</p>
<b>기본값</b>	300000 (단위: ms)

- jeus.ejb.entity.passivate

<b>설명</b>	<p>Entity Bean에 적용될 passivation timeout을 설정한다. (단위 : ms)</p> <p>jeus.ejb.all.passivate 설정에 우선한다.</p> <p>-1로 설정하면 disconnect 기능을 사용하지 않는다.</p>
-----------	---

- jeus.ejb.stateful.passivate

<b>설명</b>	Stateful Session Bean에 적용될 passivation timeout을 설정한다. (단위: ms)  jeus.ejb.all.passivate 설정에 우선한다.  -1로 설정하면 disconnect 기능을 사용하지 않는다.
-----------	---

- jeus.ejb.exportPort

<b>설명</b>	EJB에서 사용할 default export port를 설정한다.  jeus-ejb-dd.xml의 export port가 0인 EJB는 이 값을 사용한다.  0이면 RMI가 제공하는 기본 포트를 사용한다.
<b>기본값</b>	Managed Server base port + 7

- jeus.ejb.entity.lock-wait-timeout

<b>설명</b>	같은 entity를 사용할 때 lock을 기다리는 timeout 값이다.  SINGLE 또는 EXCLUSIVE일 경우에만 적용된다. (단위: ms)
<b>기본값</b>	0 (무한 대기)

- jeus.ejb.timer.generate-table-ddl

<b>설명</b>	EJB Timer Handle을 저장할 테이블을 생성하기 위한 DDL을 파일로 생성한다.  내부적인 용도로만 사용해야 한다. JEUS를 실행한 디렉터리에 createDDL.jdbc라는 이름으로 생성된다.
<b>기본값</b>	false

- jeus.ejb.generate-no-interface-view-files

<b>설명</b>	no-interface-view인 EJB의 Local business interface의 클래스 파일을 작성한다.  '_generated_' 폴더 하위에 class 파일이 생성된다.
<b>기본값</b>	false

- jeus.ejb.cluster.selection-policy

<b>설명</b>	<p>클러스터링된 EJB 엔진에 속한 EJB를 호출할 경우 어떤 EJB 엔진의 Bean을 선택할 것인지에 대한 정책을 설정한다(EJB 3 Stateless Bean 한정. 자세한 내용은 JEUS EJB 안내서의 "EJB 클러스터링" 참고).</p> <p>다음 중 하나를 선택할 수 있다.</p> <ul style="list-style-type: none"> <li>• locallinkPreference : 로컬 MS에 있는 객체를 사용하도록 한다.</li> <li>• roundrobin : 처음 요청에서는 랜덤하게 선택한 MS의 객체를 사용하고, 그 후 요청부터는 하나씩 증가하면서 서버를 선택한다.</li> <li>• random : 클러스터링된 MS들 중에서 랜덤하게 하나를 선택하여 사용한다.</li> </ul>
<b>기본값</b>	locallinkPreference

- jeus.ejb.iio.usm

<b>설명</b>	<p>IIOP EJB 사용 시 JEUS가 제공하는 Servant Manager를 적용할지 여부를 지정한다.</p> <p>Servant Manager를 적용할 경우 IIOP 요청 처리 전 후에 개입하여 TCCL을 상황에 맞게 변경하는 작업을 수행한다.</p> <p>애플리케이션에 클래스가 제대로 패키징이 되어 있는 상황에서 IIOP 요청 시 매개변수를 deserialize하는 과정에서 클래스를 못 찾는 현상이 발생하는 경우 이 옵션을 통한 해결을 시도해 볼 수 있다.</p> <p>옵션 적용 시 JEUS가 생성하는 IIOP Stub 코드가 달라지므로, 이 옵션을 적용한 서버에 배포할 EJB를 precompile하는 경우 appcompiler에도 이 옵션을 적용해야 한다.</p>
<b>기본값</b>	false

## 1.9.2. JPA 시스템 프로퍼티

다음은 JPA 시스템 프로퍼티에 대한 설명이다.

- jeus.persistence.defaultProvider

<b>설명</b>	기본 Java Persistence API Provider 클래스이름이다.
-----------	---

## 1.9.3. RMI 시스템 프로퍼티

다음은 RMI 시스템 프로퍼티에 대한 설명이다.



프로퍼티의 설명에서 [client]는 RMI 호출하고, [server]는 EJB가 deploy되어 있는 쪽에 설정해야 한다는 의미이다. RMI에 관련된 설정 및 좀 더 다양한 옵션에 대한 내용은 다음을 참고한다.

- [java.rmi 속성 문서](#)

• java.rmi.server.codebase

설명	[server] RMI를 사용할 때 EJB Bean 클래스 이외에 다른 클래스를 전달해야 할 필요가 있을 경우 class annotation에 넣어주는 codebase이다.
----	--

• java.rmi.server.useLocalHostName

설명	[server] 클라이언트가 서버로 접속할 때 서버의 IP를 사용하지 않고 호스트명을 사용할 때 사용한다.
기본값	false

• java.rmi.server.hostname

설명	[server] 클라이언트가 서버로 접속할 때 사용하는 호스트명이다.  java.rmi.server.useLocalHostName보다 우선순위가 낮다.
기본값	ip

• jeus.ejb.rmi.socketfactory

설명	[client/server] JEUS가 사용하는 RMISocketFactory 클래스의 fully qualified name을 지정한다. 지정된 클래스는 EJB뿐만 아니라 모든 RMI 통신에 사용된다.  클라이언트도 같은 RMISocketFactory를 사용해야 통신이 가능하다. 이 클래스는 no argument의 생성자(constructor)를 반드시 제공해야 하며 JEUS_HOME/lib/application에 위치할 수 있다.
기본값	jeus.ejb.util.EJBRMISocketFactory

• jeus.ejb.operationTimeout

설명	[client] 리모트 호출을 하고 나서 결과를 기다리는 최대 시간이다(SoTimeout).  sun.rmi.transport.tcp.responseTimeout 값을 JEUS에서 overwirte한다.
기본값	5 * 60 * 1000

• sun.rmi.transport.connectionTimeout

설명	[client] 클라이언트 쪽 RMI Connection이 Idle 상태로 남아 있는 최대 시간을 설정한다(KeepAliveTimeout).
기본값	15 * 1000

- sun.rmi.transport.tcp.readTimeout

<b>설명</b>	[server] 서버에서 클라이언트의 요청을 읽어들이는 데 걸릴 수 있는 최대 시간을 설정한다(SoTimeout).  클라이언트의 SoTimeout인 jeus.ejb.operationTimeout(sun.rmi.transport.tcp.responseTimeout)보다 큰 값을 지정해야 한다.
<b>기본값</b>	2 * 60 * 60 * 1000

- sun.rmi.transport.tcp.maxConnectionThreads

<b>설명</b>	[server] 서버에서 요청을 처리하는 Thread 개수의 최댓값을 설정한다.  관련 있는 JEUS 설정으로는 jeus-ejb-dd.xml에 각 Bean별로 설정하는 <thread-max> 값이 있다. <thread-max>는 각 Bean에 최대로 할당될 수 있는 Thread의 최댓값으로 maxConnectionThreads의 값을 아무리 크게 잡아도 <thread-max>가 작으면 Thread 수가 증가하지 않는다.  예를 들어 EJB A와 EJB B 두 Bean이 존재하고 <thread-max>를 따로 설정하지 않고 기본값인 100을 사용하면 maxConnectionThreads가 Integer.MAX_VALUE이더라도 200까지 밖에 증가하지 않게 된다.
<b>기본값</b>	Integer.MAX_VALUE

- sun.rmi.transport.tcp.threadKeepAliveTime

<b>설명</b>	[server] Thread가 요청이 끝난 후(Idle 상태로) 다른 요청을 기다리는 최대 시간을 설정한다(KeepAliveTimeout).
<b>기본값</b>	1*60*1000

- jeus.ejb.compiler.rmic.fork

<b>설명</b>	[server] IIOP 클래스들을 생성할 때 RMIC 컴파일러 프로세스를 fork할지 여부를 설정한다.
<b>기본값</b>	false

- jeus.ejb.compiler.rmic.verbose

<b>설명</b>	[server] IIOP 클래스들을 생성할 때 verbose 로그를 남길지 여부를 설정한다.
<b>기본값</b>	false

- jeus.ejb.compiler.rmic.vcompat

<b>설명</b>	IIOP을 컴파일할 때 RMI 1.1, 1.2 compatible protocol로 generate할지 여부를 결정한다.
<b>기본값</b>	false



- jeus.ejb.compiler.rmic.enable

설명	IIOP Stub 파일을 만들지 말지를 결정한다. 일반적인 경우에는 IIOP Stub 파일이 필요 없다.
기본값	false

- jeus.ejb.compiler.class.version

설명	[server] EJB 생성된 클래스를 컴파일할 때 source와 target version을 설정한다.
기본값	1.4

- jeus.ejb.compiler.use-java-compiler-api

설명	JDK 6부터 제공하는 Java Compiler API를 사용하도록 하는 옵션이다.  기본적으로 사용하도록 되어 있지만 기존처럼 JDK의 lib/tools.jar 내에 포함된 SUN 내부 클래스를 직접 이용해서 컴파일하는 방식을 원할 때는 이 옵션을 false로 설정한다.  컴파일된 클래스 차이에 의해 호환성 문제 등이 발생했을 때 사용할 수 있다.
기본값	true

## 1.10. 보안 시스템 프로퍼티

본 절에서는 보안 시스템 프로퍼티에 대해 설명한다.

### 1.10.1. Standard Java SE & Jakarta™ EE 보안 시스템 프로퍼티

다음은 Standard Java SE & Jakarta™ EE 보안 시스템 프로퍼티에 대한 설명이다.

- java.security.manager

설명	코드 보호를 통해 JEUS 서버의 견고성을 향상하기 위해서 Java SE SecurityManager를 사용한다. 이는 퍼포먼스를 떨어뜨릴 수 있다.  기본으로 SecurityManager는 JEUS에서 사용되지 않는다.
----	--

- jeus.security.enable.default.manager

설명	System.exit 메소드가 호출되어도 JEUS JVM 프로세스가 shutdown되지 않는다.  -Djeus.security.enable.default.manager=true로 설정하면 JDK의 Java Security Manager가 아닌 Java Security Manager를 확장한 JEUS Security Manager가 적용되어서 애플리케이션 프로그램 코드에서 System.exit 메소드가 호출되어도 JEUS JVM 프로세스가 shutdown되지 않는다.
----	--

기본값	false
-----	-------

- java.security.policy

설명	SecurityManager에서 사용하는 Java SE 정책 파일의 경로를 설정한다.
기본값	JAVA_HOME/lib/security/java.security

- jakarta.security.jacc.policy.provider

설명	JACC Policy 구현 클래스 이름을 설정한다.
기본값	jeus.security.impl.jacc.JACCPolicyWrapper

- jakarta.security.jacc.PolicyConfigurationFactory.provider

설명	JACC PolicyConfigurationFactory 구현 클래스 이름을 설정한다.
기본값	jeus.security.impl.jacc.JACCPolicyConfigurationFactoryImpl

## 1.10.2. JEUS 보안 시스템 프로퍼티

다음은 JEUS 보안 시스템 프로퍼티에 대한 설명이다.

- jeus.security.enable

설명	JEUS를 기동할 때 보안 서버를 설치할지 여부를 설정한다.
기본값	true

- jeus.security.default.configDir

설명	보안 시스템이 사용하는 설정 루트 디렉터리를 설정한다.
기본값	JEUS_HOME/domains/<domain-name>/config/security

- jeus.security.default.domainName

설명	디폴트 도메인명을 설정한다.
기본값	SYSTEM_DOMAIN

- jeus.security.installer.classname

설명	SecurityInstaller의 구현 클래스 이름을 설정한다.
기본값	jeus.security.impl.installer.JeusSecurityDomainInstaller

- jeus.security.jacc.principalRoleMapper

<b>설명</b>	jeus.security.impl.aznrep.JACCPrincipalRoleMapper 인터페이스를 구현하는 클래스 이름을 설정한다. 이 클래스는 JACC Provider에서 principal-role 매핑을 담당한다.
<b>기본값</b>	jeus.security.impl.jacc.JACCDefaultPrincipalRoleMapper

- jeus.security.keypath

<b>설명</b>	security.key 파일 위치를 지정한다.
<b>기본값</b>	JEUS 내부에서 기본적으로 설정된다.

- jeus.security.master

<b>설명</b>	Master Password를 지정한다.
<b>기본값</b>	보안상의 안전을 위해 프롬프트로 입력받는 것을 권장한다.

- jeus.net.logListenPort

<b>설명</b>	서버 소켓을 만들 때 Port의 logging 여부를 설정한다.  jeus.security.enable.default.manager=true를 설정해야 작동한다.
<b>기본값</b>	true

- jeus.net.logConnectInfo

<b>설명</b>	소켓을 만들 때 접속하는 호스트, Port의 logging 여부를 설정한다.  jeus.security.enable.default.manager=true를 설정해야 작동한다.
<b>기본값</b>	false

- jeus.net.logListenStack

<b>설명</b>	서버 소켓을 생성할 때의 Thread Stack Trace의 logging 여부를 설정한다.  jeus.security.enable.default.manager=true를 설정해야 작동한다.
<b>기본값</b>	false

- jeus.net.logConnectStack

<b>설명</b>	소켓을 만들고 접속할 때의 Thread Stack Trace의 logging 여부를 설정한다.  jeus.security.enable.default.manager=true를 설정해야 작동한다.
<b>기본값</b>	false

- jeus.security.keylength

설명	암호화 알고리즘(AES/DES/DESE/SEED/Blowfish 등)을 사용할 때 key 크기를 지정한다. jeus.security.keylength=256을 설정하면 시스템 전체에 256bit로 통일된 key 크기가 적용된다.
기본값	256

- jeus.security.timeout

설명	Security에서 request를 보낸 후 응답이 올때까지 기다리는 시간을 설정한다. 단위는 ms이다.
기본값	20 * 1000

- jeus.security.jaspic.auth-config-factory

설명	JEUS에서 사용할 JASPIC(JSR-196) AuthConfigFactory 구현체의 class 이름을 지정한다.
기본값	jeus.servlet.security.jaspic.JEUSAuthConfigFactory

- jeus.security.authentication.secureMode

설명	secure authentication mode로 지정한다.  false 일 경우 seed 값은 empty string 으로 동작한다.
기본값	true

- jeus.security.auth-server-url

설명	JEUS에서 인증/인가를 외부 서버 API를 통해 수행할 때, 인증/인가 서버의 URL을 설정한다.
----	---

## 1.11. JEUS MQ 시스템 프로퍼티

본 절에서는 JEUS MQ 시스템 프로퍼티에 대해 설명한다.

### 1.11.1. JEUS MQ 서버

다음은 JEUS MQ 서버 시스템 프로퍼티에 대한 설명이다.

- jeus.jms.file.blocksize

설명	FileMessage를 사용하여 클라이언트로 파일을 블록 단위로 전송할 때의 블록 크기를 설정한다.
기본값	4096 (단위: byte)

- jeus.jms.server.deleteFileMessage

설명	FileMessage 전달을 완료한 뒤 해당 파일을 서버에서 삭제할지 여부를 설정한다.
기본값	true

- jeus.jms.server.discard-stored-on-start

설명	JEUS MQ 서버를 시작할 때 설정된 스토리지에 존재하는 모든 데이터를 삭제할지 여부를 설정한다.
기본값	false

- jeus.jms.server.rowFetchSize

설명	데이터베이스 스토리지부터 메시지를 쿼리할 때 한 번에 가져올 레코드 개수를 설정한다.
기본값	1000

- jeus.jms.server.create-facility-mbeans

설명	Connection, Session, Message Producer, Consumer와 같은 Client Facility들에 대한 MBean을 생성할지 여부를 설정한다.
기본값	false

## 1.11.2. JEUS MQ 클라이언트

다음은 JEUS MQ 클라이언트 시스템 프로퍼티에 대한 설명이다.

- jeus.jms.client.default-redelivery-limit

설명	JMS_JEUS_RedeliveryLimit 메시지 프로퍼티의 기본값을 설정한다.  메시지 송신자에서 별도로 지정하지 않은 경우 송신한 메시지는 최대 이 횟수만큼만 복구되어 메시지 수신자에게 전달된다.
기본값	3

- jeus.jms.file.blocksize

설명	FileMessage를 사용하여 서버로 파일을 블록 단위로 전송할 때의 블록 크기를 설정한다.
기본값	4096 (단위: byte)

- jeus.jms.client.connect.timeout

설명	ConnectionFactory.createConnection() 메소드가 서버로부터 응답을 기다리는 제한시간을 설정한다.
기본값	5000 (단위: ms)

- jeus.jms.client.workdir

<b>설명</b>	FileMessage를 사용하여 서버로부터 수신한 파일이 저장되는 경로를 설정한다.  JavaEE 클라이언트의 경우 이 설정은 무시되고 JEUS_HOME/logs/jms/BROKER_NAME/client/ 경로가 사용된다.
<b>기본값</b>	USER_HOME/.jeusmq_client_work/

- jeus.jms.client.maxThread

<b>설명</b>	JEUS MQ 클라이언트 라이브러리에서 사용하는 Thread Pool의 크기를 설정한다.  클라이언트 라이브러리가 메시지 송신과 수신 등 여러 가지 작업을 동시에 수행하기 위해 필요하다. 서버와 맺는 커넥션마다 하나씩의 Thread Pool이 생성된다.
<b>기본값</b>	100

- jeus.jms.client.optimizedProduce

<b>설명</b>	메시지 송신 성능을 향상시키기 위해 서버로부터 Acknowledge를 받지 않을지 여부를 설정한다.  true로 설정하면 세션의 승인 모드를 JeusSession.NONE_ACKNOWLEDGE로 설정하지 않아도 메시지를 송신할 때 항상 서버로부터 Acknowledge를 받지 않는다. 단, FileMessage를 송신할 때에는 이 설정에 관계없이 Acknowledge를 받는다.
<b>기본값</b>	false

- jeus.jms.client.use-single-server-entry

<b>설명</b>	커넥션 간에 물리적 연결을 공유하여 사용할지 여부를 설정한다.  한 ConnectionFactory당 하나의 물리적 연결을 가지게 된다.
<b>기본값</b>	true

- jeus.jms.client.single-server-entry.shutdown-delay

<b>설명</b>	커넥션 간에 공유하던 물리적 연결이 더 이상 사용되지 않을 경우 재사용을 위해 Shutdown이 지연되는 시간을 설정한다.
<b>기본값</b>	600000 (단위: ms)

- jeus.jms.client.use-pooled-connection-factory

<b>설명</b>	한 번 사용했던 Client Facility들을 재사용하기 위해서 Pooling을 할지 여부를 설정한다.
<b>기본값</b>	true

- jeus.jms.client.pooled-connection.check-period

<b>설명</b>	Pooling된 Client Facility들 중 사용되지 않는 것들을 제거하기 위해서 검사하는 주기를 설정한다.
<b>기본값</b>	60000 (단위: ms)

- jeus.jms.client.pooled-connection.unused-timeout

<b>설명</b>	Pooling된 Client Facility들 중 이 시간 동안 사용되지 않으면 Pool에서 제거한다.
<b>기본값</b>	120000 (단위: ms)

- jeus.jms.client.send-by-lpq-only

<b>설명</b>	메시지를 송신하는 경우 모든 메시지를 신뢰도를 높이기 위한 LPQ 서비스를 이용해 보낸다.
<b>기본값</b>	false

- jeus.jms.client.lpq-configuration-path

<b>설명</b>	<p>LPQ 서비스를 이용할 때 사용할 설정 파일의 세부 경로를 설정한다.</p> <ul style="list-style-type: none"> <li>• 절대 경로로 주어진 경우 : 그 경로의 파일을 읽는다.</li> <li>• 상대 경로로 주어진 경우 : 다음의 순서대로 발견되는 파일을 읽어온다. <ul style="list-style-type: none"> <li>1. DEPLOYED_HOME/myApp/WEB-INF/</li> <li>2. DEPLOYED_HOME/myApp/META-INF/</li> <li>3. DEPLOYED_HOME/myApp/</li> </ul> </li> </ul>
<b>기본값</b>	120000 (단위 : ms)

## 1.12. Deployment 시스템 프로퍼티

본 절에서는 Deployment 시스템 프로퍼티에 대해 설명한다.

- jeus.classloading

<b>설명</b>	AppCompiler를 사용하는 경우 애플리케이션의 클래스 로더들이 서로의 클래스 로더를 참조할지 여부를 설정한다.
<b>기본값</b>	ISOLATED

- jeus.app.graceful-timeout

<b>설명</b>	Graceful undeployment를 수행할 때 요청 처리가 종료될 때까지 얼마나 기다릴지를 설정한다.
<b>기본값</b>	0

- jeus.app.allow-deprecated-dd

<b>설명</b>	JEUS 6 이전 버전에서 작성한 deployment descriptor를 배포할 때 JEUS 8에 맞게 변환할지 여부를 설정한다.
<b>기본값</b>	false

- jeus.application.cleanup-deployed-image

<b>설명</b>	애플리케이션을 undeploy할 때 생성된 deployment image(application root directory와 generation directory)의 삭제 여부를 지정한다.
<b>기본값</b>	true

- jeus.app.deploy.persistent

<b>설명</b>	애플리케이션을 deploy/undeploy할 때 해당 애플리케이션에 대한 정보를 domain.xml에 기록할지 여부를 지정한다.
<b>기본값</b>	true

- jeus.deploy.concurrent

<b>설명</b>	boot time에 둘 이상의 애플리케이션 deploy 요청에 대해 concurrent deploy를 지원할지 여부이다.
<b>기본값</b>	true

- jeus.deploy.ignore-not-found-classes

<b>설명</b>	애플리케이션을 deploy할 때 loading할 class를 찾을 수 없어 loading이 불가능하면 deploy를 실패시킬지 아니면 무시하고 deploy를 완료할지 여부를 결정한다.
<b>기본값</b>	false

- jeus.deploy.concurrent.runtime

<b>설명</b>	run time에 둘 이상의 애플리케이션 deploy 요청에 대해 concurrent deploy를 지원할지 여부이다.
<b>기본값</b>	false

- jeus.cdi.enable-implicit-bean-archive

<b>설명</b>	배포할 때 CDI implicit bean archive에 대한 처리를 진행할지 여부를 지정한다.
<b>기본값</b>	true

- jeus.cdi.enabled

<b>설명</b>	배포할 때 JEUS가 제공하는 CDI 기능을 사용할지 여부를 지정한다.
-----------	---



기본값	true
-----	------

- jeus.deploy.ignore-persistence-descriptor

설명	배포할 때 애플리케이션이 포함하고 있는 persistence.xml을 무시할지 여부를 지정한다.
기본값	false

## 1.13. Console 시스템 프로퍼티

본 절에서는 Console 시스템 프로퍼티에 대해 설명한다.

- jeus.tool.console.patchinfo

설명	콘솔 툴 실행 시 패치 정보를 출력할지 여부를 지정한다. 각종 툴에 적용하는 옵션이기에, 서버 설정이 아니라 JEUS_HOME/bin/jeus.properties에 있는 TOOL_OPTION이나 툴 실행 스크립트 등을 수정하는 방식으로 지정하여야 한다.
기본값	false

- jeus.console.table.width

설명	콘솔 툴에서 출력되는 표의 너비를 지정한다.
기본값	80

## Part II. 콘솔 명령어와 툴

## 2. 명령어 설명 규칙

설명할 명령어들에 대한 간단한 규칙들에 대해 설명한다. 모든 명령어들에 대해 각 절 제목으로 명령어의 이름과 간략한 내용을 기술하고, 본문에서 설명, alias, 사용법, 파라미터, 사용 예로 나누어 설명한다.

- 설명 : 명령어의 기능에 대해 설명한다.
- 관련 스키마 : 해당 명령어가 관련되어 있는 스키마 항목을 표기한다.
- alias : 명령어에 대한 별칭(alias name)을 표기한다.
- 사용법 : 명령어의 문법에 대해 설명한다. jeusadmin에서 help를 실행했을 때 usage 라인에 출력되는 문법과 동일하다.
  - 명령어 : 명령어는 **굵은 글씨체**를 사용한다.
  - 파라미터 : 명령어의 옵션을 비롯한 파라미터들에 대해 각각이 어떤 의미를 가지고 있으며, 어떤 경우에 어떻게 사용하는지를 설명한다.

파라미터는 일반 글씨체와 이탤릭체를 사용한다.

구분	설명
일반 글씨체	옵션을 표현한다. 사용자는 안내서에 기술된 그대로를 사용한다.
이탤릭체	사용자가 입력할 파라미터를 나타낸다. 해당하는 값을 사용자가 직접 입력한다.
[ ]	생략가능한 항목을 나타낸다. []로 묶여있지 않은 항목은 필수 항목이다.
또는 /	'or'를 나타내는 것으로 이것으로 구분된 파라미터들은 그 중에 하나만을 사용할 수 있다.

- 파라미터 : 명령어에 사용할 수 있는 옵션, 인자 등 파라미터에 대해 설명한다.
- 사용 예 : jeusadmin에서의 실제 사용 예와 출력되는 결과에 대해 간략하게 설명한다.

## 3. JEUS 서버 기동 및 종료

본 장에서는 JEUS 서버를 기동 및 종료하는 명령어에 대해 설명한다.

### 3.1. 개요

JEUS 서버를 실행하는 명령어는 다음의 2가지로 나뉜다.

- JEUS Master Server를 실행하는 **startMasterServer** 명령어
- Managed Server를 실행하는 **startManagedServer** 명령어

서버 실행 명령어들은 해당 서버의 Launcher를 실행한다. Launcher는 서버를 실행시키기 위한 준비작업을 완료한 뒤 실제 서버 JVM을 기동한다. JEUS Master Server는 실제 서비스가 되는 Managed Server를 관리하는 서버로 관리자의 명령을 Managed Server로 전달하는 Agent 역할을 한다.

JEUS 서버를 종료하는 명령어는 서버 종류에 관계 없이 하나이다.

- **stopServer** 명령어

서버 기동 및 종료 명령어는 다음의 디렉터리에 위치한다.

```
JEUS_HOME/bin/
```

### 3.2. 사용법

본 절에서는 JEUS Master Server와 Managed Server를 기동 및 종료하는 방법에 대해서 설명한다.

JEUS에서 제공하는 관리 도구를 통해 Managed Server를 관리하기 위해서는 먼저 JEUS Master Server를 기동해야 한다.

JEUS Master Server는 각 Managed Server와 통신하고 각 Managed Server는 각 HOST 머신에서 동작한다. 도메인에 속한 모든 서버는 아래에 위치한 하나의 설정 파일을 공유한다. 해당 설정 파일에서 자신의 동작에 필요한 설정들을 읽어 동작한다.

```
JEUS_HOME/domains/DOMAIN_NAME/config/domain.xml
```

#### JEUS Master Server 기동

다음은 JEUS Master Server를 기동하기 위해 startMasterServer 명령을 실행하는 과정이다.

1. JEUS Master Server를 기동하려면 각 OS별로 다음의 명령을 실행한다.
  - UNIX나 Linux

2. 명령 프롬프트에 **startMasterServer**를 입력한 뒤 아래의 설정들을 추가한다.

domain.xml을 수정하면 서버를 디버그 모드로 운영할 수 있다. 이 설정 방법은 [디버그 모드의 JEUS 서버 실행](#)을 참고한다.

```
startMasterServer -u <user-name> -p <password>
                  [-domain <domain-name> -server <server-name>]
                  [-cachellogin]
                  [-f <file-name>]
                  [-force]
                  [-standby]
                  [-verbose]
                  [-fg]
                  [-h]
```

다음은 명령어 옵션에 대한 설명이다.

옵션	설명
-u <user-name>	서버를 기동하는 주체를 의미한다.
-p <password>	
[-domain <domain-name>]	기동할 JEUS Master Server가 속한 도메인의 이름을 설정한다.
[-server <server-name>]	기동할 JEUS Master Server의 이름을 설정한다.
[-cachellogin]	-u, -p 옵션으로 입력한 로그인 정보를 캐시한다.  캐시한 로그인 정보가 존재할 경우 -p 옵션을 대체할 수 있다. 권장하지 않는 기능으로 자세한 내용은 "JEUS Security 안내서"를 참고한다.
[-f <file-name>]	캐시 로그인 정보를 가진 파일의 경로를 지정한다.
[-force]	등록된 애플리케이션이 모두 RUNNING 상태가 아니더라도, 서버의 상태를 강제로 RUNNING으로 만든다.
[-standby]	서버를 STANDBY 상태로 기동시킨다.
[-verbose]	Launcher를 종료하지 않고 서버에서 발생하는 로그를 Launcher 화면에 출력할지 여부를 지정한다.
[-fg]	서버를 Background 모드가 아닌 Foreground 모드로 기동시킨다. 이 경우 Launcher는 서버가 기동되는 시점에 종료된다.
[-h]	명령의 도움말을 출력한다.

Managed Server를 기동할 때 옵션은 JEUS Master Server를 기동할 때의 옵션과 대부분이 동일하다. Managed Server에서는 특별히 masterurl 옵션이 추가로 존재하며, 이는 Managed Server를 기동할 때 JEUS Master Server로부터 설정 파일을 받아오는 데 사용된다.

## Managed Server 기동

다음은 Managed Server를 기동하기 위해 startManagedServer 명령을 실행하는 과정이다.

1. Managed Server를 기동하려면 다음의 명령을 실행한다.

- UNIX나 Linux

```
startManagedServer
```

2. 명령 프롬프트에 **startManagedServer**를 입력한 뒤 아래의 설정들을 추가한다.

```
startManagedServer -u <user-name> -p <password>
                    -server <server-name>
                    [-domain <domain-name>]
                    [-cachellogin]
                    [-f <file-name>]
                    [-force]
                    [-standby]
                    [-verbose]
                    [-masterurl <master-url>]
                    [-fg]
                    [-h]
```

다음은 명령어 옵션에 대한 설명이다.

옵션	설명
-u <user-name>	서버를 기동하는 주체를 의미한다.
-p <password>	-u 및 -p는 관리자 ID와 패스워드를 의미하며, 서버 실행을 위한 필수 입력사항이다.
-server <server-name>	기동할 Managed Server의 이름을 설정한다. 필수 입력사항이다.
[-domain <domain-name>]	기동할 Managed Server가 속한 도메인의 이름을 설정한다.
[-cachellogin]	[-u], [-p] 옵션으로 입력한 로그인 정보를 캐시한다.  캐시한 로그인 정보가 존재할 경우 [-p] 옵션을 대체할 수 있다. 권장하지 않는 기능으로 자세한 내용은 "JEUS Security 안내서"를 참고한다.
[-f <file-name>]	캐시 로그인 정보를 가진 파일의 경로를 지정한다.
[-force]	등록된 애플리케이션이 모두 RUNNING 상태가 아니더라도, 서버의 상태를 강제로 RUNNING으로 만든다.
[-standby]	서버를 STANDBY 상태로 기동시킨다.
[-verbose]	Launcher를 종료하지 않고 서버에서 발생하는 로그를 Launcher 화면에 출력할지 여부를 지정한다.

옵션	설명
[-masterurl <master-url>]	자신이 속한 도메인을 관리하는 JEUS Master Server의 URL 주소를 지정한다. JEUS Master Server로부터 새로운 설정 파일을 받아올 수 있도록 URL 주소를 설정할 것을 권장한다.
[-fg]	서버를 Background 모드가 아닌 Foreground 모드로 기동시킨다. 이 경우 Launcher는 서버가 기동되는 시점에 종료된다.
[-h]	명령어의 도움말을 출력한다.

## 서버 종료

다음은 서버를 종료하기 위해 stopServer 명령을 실행하는 과정이다.

1. 서버를 종료하려면 다음의 명령을 실행한다.

- UNIX나 Linux

```
stopServer
```

2. 명령 프롬프트에 **stopServer**를 입력한 뒤 아래의 설정들을 추가한다.

```
stopServer -host <host:port>
           [-server <server-name> -masterurl <address:port>]
           -u <user-name> -p <password>
           [-cachelgin -domain <domain-name> -f <cachelgin-file>]
           [-g -to <timeout>]
```

다음은 명령어 옵션에 대한 설명이다.

옵션	설명
-host <host:port>	종료시키려는 서버의 호스트 정보를 host:port 형태로 지정한다.
[-server <server-name> -masterurl <address:port>]	종료시키려는 Managed Server의 이름을 [-server] 옵션에 지정하고, Managed Server가 위치한 JEUS Master Server의 주소 정보를 [-masterurl] 옵션에 address:port 형태로 지정한다.  -host 옵션과 함께 사용할 수 없다.
-u <user-name> -p <password>	서버를 종료하는 주체를 의미한다.  [-u] 및 [-p]는 관리자 ID와 패스워드를 의미한다.

옵션	설명
<code>[-cachelogin -domain &lt;domain-name&gt; -f &lt;cachelogin-file&gt;]</code>	<p><code>[-u]</code>, <code>[-p]</code> 옵션으로 입력한 로그인 정보를 캐시한다.</p> <p>캐시 로그인 정보를 지정하는 key로 domain이름을 사용한다. 따라서 <code>[-domain]</code> 옵션을 통해 domain 이름을 지정해야 한다.</p> <p><code>[-f]</code> 옵션은 캐시 로그인 정보를 가진 파일의 경로를 지정하기 위한 옵션으로, 미지정할 경우 기본값인 <code>~/jeusadmin/.jeuspasswd</code>로 지정된다.</p> <p>캐시한 로그인 정보가 존재할 경우 <code>[-p]</code> 옵션을 대체할 수 있다. 권장하지 않는 기능으로 자세한 내용은 "JEUS Security 안내서"를 참고한다.</p>
<code>[-g -to &lt;timeout&gt;]</code>	<p>Graceful하게 종료할지 여부를 설정한다.</p> <p><code>[-to]</code> 옵션을 통해 Graceful하게 종료할 때 사용할 타임아웃을 설정할 수 있다. 처리 중인 요청을 입력한 시간만큼 완료되길 기다린다. (단위: second)</p> <p><code>[-to]</code> 옵션을 미지정할 경우 처리 중인 요청이 끝나길 무한히 기다린다.</p>

### 3.3. 마스터 패스워드 입력

암호화 툴을 이용하여 security.key 파일을 마스터 패스워드로 암호화한 경우 서버를 기동할 때 마스터 패스워드를 입력해야 한다. security.key 파일에는 대칭키 암호화 알고리즘의 키가 저장된다. 암호화 툴에 관한 자세한 사항은 [encryption](#)을 참조한다.

서버를 기동 및 종료할 때 security.key 파일이 암호화되어 있는 경우 다음과 같은 콘솔 입력 화면이 나타나면서 마스터 패스워드의 입력을 기다린다. 마스터 패스워드가 맞지 않는다면 서버 기동 및 종료가 불가능하다.

```
The encryption key file is encrypted. Enter the master password.
Password>
```

### 3.4. 디버그 모드의 JEUS 서버 실행

디버그 모드로 JEUS 서버를 실행할 경우, JPDA 지원 디버거를 사용하여 서블릿과 EJB 애플리케이션을 디버그할 수 있다. JEUS 서버를 디버그 모드로 시작하기 위해서는 각 서버의 jvm-option 설정에 다음과 같이 디버그 옵션을 추가한다.

```
-agentlib:jdwp=transport=dt_socket,server=y,suspend=n,address=8888
```

“8888”의 디버그 Port(디버그 Port는 JPDA 디버거가 JVM에 접근할 때 사용된다)를 사용하여 JEUS 서버를 디버그 모드에서 동작하도록 한다.

수정된 설정으로 JEUS 서버를 기동한다. 그런 다음 JPDA를 지원하는 IDE(Integrated Development Environment)를 사용하여 EJB와 서블릿 애플리케이션을 디버그한다.



각 서버의 jvm-option 설정을 통해 시스템 환경변수 등도 설정이 가능하다. JEUS에서 사용하는 시스템 환경변수에 대해서는 "Part I. 시스템 프로퍼티"를 참조한다. [개요](#)에서는 JVM에서 사용하는 "-D" 옵션에 대하여 설명하고 있다.



jvm-option 설정은 해당 서버에만 적용된다는 것을 유의한다.

## 4. 콘솔 툴

본 장에서는 JEUS에서 사용하는 콘솔 툴의 종류와 사용법에 대해 설명한다.

### 4.1. 개요

다음은 JEUS에서 사용하는 콘솔 툴의 목록이다.

- Admin 툴

콘솔 툴	설명
<a href="#">jeusadmin</a>	JEUS를 직접 관리하기 위해 사용하는 콘솔 툴이다.

- 기타 툴

콘솔 툴	설명
<a href="#">appcompiler</a>	애플리케이션의 EJB 인터페이스 Impl, Skeleton, Stub 클래스, JSP를 컴파일하여 서블릿 클래스, 웹 서비스의 Endpoint 클래스 등을 생성한다.
<a href="#">ejbddinit</a>	EJB 모듈의 DD 파일(jeus-ejb-dd.xml)을 ejb-jar.xml 정보와 사전에 생성된 프로퍼티 파일을 이용하여 생성할 수 있다.
<a href="#">encryption</a>	암호화할 문자열을 암호화하거나 그 반대로 복호화하는 기능을 제공한다.
<a href="#">java2wsdl</a>	Java 클래스로부터 WSDL 파일과 JAX-RPC 매핑 파일을 생성한다.
<a href="#">jeusddupgrade</a>	이전 버전의 애플리케이션 안의 JEUS DD 파일(jeus-ejb-dd.xml, jeus-web-dd.xml, jeus-application-dd.xml)을 현재 버전에 맞게 업그레이드할 수 있다.
<a href="#">schemagen</a>	Java 클래스들에 나타나는 각각의 Namespace에 대해 하나의 스키마 파일을 생성한다.
<a href="#">tcpmon</a>	송수신하는 TCP 패킷을 조회한다. tcpmon은 HTTP의 SOAP 메시지를 조회할 수 있다.
<a href="#">webddgen</a>	WEB 모듈의 DD 파일(jeus-web-dd.xml)을 생성하거나 업데이트할 수 있다.
<a href="#">wsген</a>	JAX-WS 웹 서비스를 위한 툴로 Java 클래스로부터 WSDL 파일과 JAX-RPC 매핑 파일을 생성한다. 또한 웹 서비스 정책 설정 파일을 생성할 수 있다.
<a href="#">wsimport</a>	WSDL 파일로부터 클라이언트 측 Java Stub 소스 파일들과 서버 측 웹 서비스 인터페이스 Java 소스 파일을 생성한다. 또한 웹 서비스 정책 설정 파일을 생성할 수 있다.
<a href="#">wsdl2java</a>	WSDL 파일로부터 클라이언트 측 Java Stub 소스 파일과 서버 측 웹 서비스 인터페이스 Java 소스 파일을 생성한다.
<a href="#">wsdl2uddi</a>	웹 서비스의 WSDL 파일로부터 UDDI에 Publish할 수 있다.
<a href="#">xjc</a>	XML 스키마 파일을 Java 프로그래밍 언어로 된 JAXB Content 클래스들로 변환한다.

### 4.2. jeusadmin

본 절에서는 JEUS의 관리 툴인 jeusadmin의 명령어와 그 사용법에 대해서 설명한다.

jeusadmin은 JEUS를 직접 관리하기 위해 사용하는 콘솔 툴이다. 이 툴을 사용하면 JEUS 서버의 기동과 종료와 같은 기본적인 관리 작업을 할 수 있으며 애플리케이션의 deploy/undeploy와 logger 레벨 변경 작업, JMX MBean의 목록 조회 등 여러 가지 주요 작업을 할 수 있다.

jeusadmin에서 사용하는 명령어에 대한 기본 문법은 다음과 같다.

```
COMMAND [COMMAND_OPTIONS]
        [GLOBAL_OPTIONS]
(>|>>) FILENAME
```

- COMMAND

명령어 이름 또는 alias를 설정한다.

- COMMAND\_OPTIONS

해당 명령어에 대한 옵션을 설정한다.

옵션으로 명령어를 주거나 옵션이 공백을 포함할 경우 큰따옴표(" ")로 COMMAND\_OPTIONS)를 묶는 것을 원칙으로 한다.

다만, 큰따옴표(" ") 묶음을 두 번 이상 사용할 경우에는 사용하는 운영체제별로 이스케이프 문자를 사용하도록 한다.

운영체제	큰따옴표에 대한 이스케이프 문자
Linux OS	역슬래시(\)

다음은 이에 대한 명령어 예시이다.

```
[ Linux OS ] jeusadmin -u jeus -p 11111111 -host localhost:9116 "help -g \"Data Source and Connection Pool\""
```

- GLOBAL\_OPTIONS

모든 명령어에 대해 적용할 수 있는 공용 옵션을 설정한다.

옵션	설명
-verbose	오류가 발생하는 경우 Stack Trace를 출력한다.
-repeat <count>	명령어를 repeat count 횟수만큼 반복 실행한다.
-interval <seconds>	반복해서 실행하는 경우 실행 간격을 초 단위로 지정한다. -repeat 옵션을 주지 않고 사용할 경우 오류가 발생한다.

- (>|>>) FILENAME

명령어 실행 결과를 파일에 저장할 때 사용한다. 저장할 파일 이름은 FILENAME으로 지정한다.

다음과 같이 사용할 수 있다.

```
offline>help > result.txt
또는
offline>help >> result.txt
```

## 4.2.1. 기동 및 종료 명령어

jeusadmin을 실행하고 종료하는 명령어는 다음과 같다.

명령어	설명
<a href="#">jeusadmin</a>	jeusadmin을 실행한다.
<a href="#">exit</a>	관리 작업을 완료하고 jeusadmin을 종료할 때 사용하는 명령어이다.

### 4.2.1.1. jeusadmin

jeusadmin을 실행한다. 이 파일은 JEUS\_HOME/bin에 위치한다.



정해진 횟수 이상 로그인에 실패하면 jeusadmin을 수행하는 클라이언트 주소가 서버에 의해 접속 차단될 수 있으므로 주의한다.

#### • 사용법

```
jeusadmin [-u,--username <user-name>]
           [-p,--password <password>]
           [-f <file-name>]
           [-cachelogin]
           [-d,--domain <domain-name>]
           [-host <server-address>]
           [-port <server-port>]
           [-verbose]
           [-help]
           [<target-command>]
           [-version]
           [-fullversion]
           [-buildversion]
           [-licensedue]
           [-licenseinfo]
           [-debug]
           [-script (<script-file>|"<script-file> <args> ...")]
           [-i,--ignore]
```

#### • 파라미터

파라미터	설명
[-u, --username <user-name>]	접속할 때 사용할 사용자 이름을 지정한다.  옵션을 준 경우 실행하면서 서버에 접속을 시도한다.
[-p, ---password <password>]	접속할 때 사용할 비밀번호를 지정한다.  옵션을 준 경우 실행하면서 서버에 접속을 시도한다.
[-f <file-name>]	접속할 때 사용할 사용자 이름과 비밀번호를 담은 파일을 지정한다.
[-cachellogin]	특정 도메인(또는 기본 도메인)에 접속할 때 사용했던 사용자 이름과 비밀번호를 기록한다.
[-d, --domain <domain-name>]	접속하고자하는 도메인의 이름 지정한다.
[-host <server-address>]	접속할 때 사용할 주소를 지정한다.  [-u] 옵션과 [-p] 옵션을 준 상태에서 이 옵션을 별도로 주지 않으면 localhost로 접속을 시도한다.
[-port <server-port>]	접속할 때 사용할 포트 번호를 지정한다.  [-u] 옵션과 [-p] 옵션을 준 상태에서 이 옵션을 별도로 주지 않으면 9736 포트로 접속을 시도한다.
[-verbose]	실행 중 오류가 발생할 경우 세부 사항까지 출력한다.
[-help]	jeusadmin에 대한 도움말을 출력한다.
[<target-command>]	<target-command>를 한 번 실행한다. 스크립트에서 사용할 때 유용하다. 만약 서버에 접속이 필요한 명령어라면 [-u], [-p], [-host], [-port] 등 필요한 옵션을 같이 주어야 한다.
[-version]	JEUS의 메이저 버전을 출력한다.
[-fullversion]	JEUS의 메이저 버전과 마이너 버전을 출력한다.
[-buildversion]	JEUS의 메이저 버전과 빌드 버전을 출력한다.
[-licensedue]	라이선스의 남은 기간을 출력한다. 남은 기간이 0이 되면 JEUS는 더 이상 동작하지 않으므로, TmaxSoft로부터 새로운 라이선스를 발급받아야 한다.
[-licenseinfo]	현재 설치된 라이선스에 대한 정보를 출력한다.
[-debug]	jeusadmin에서 Jeus Logger를 통해 출력되는 로그 메시지들을 표시하도록 한다.
[-script (<script-file>   "<script-file> <args> ...")]	Script 모드로 jeusadmin을 실행한다. 이때 파일의 확장자는 .py(Python), .rb(Ruby)이어야 한다.
[-i,--ignore]	Script 모드에서 JEUS 커맨드 예외 발생 시 스크립트를 중단하지 않고, 계속 스크립트를 실행할 수 있는 옵션이다. 이 옵션은 Script 모드에서만 사용가능하다.

#### • 예제

다음은 JEUS 사용자 이름은 'administrator', 비밀번호는 'jeus'로 가정한 jeusadmin 예제이다.

- jeusadmin 실행

다음은 jeusadmin을 실행하는 예제이다.

```
JEUS_HOME/bin$ jeusadmin
JEUS 9 Administration Tool
To view help, use the 'help' command.
offline>
```

- 실행하면서 바로 서버에 접속 #1

다음은 localhost:9736에 있는 서버에 접속하는 경우에 대한 예제이다.

```
JEUS_HOME/bin$ jeusadmin -u administrator -p jeus
Attempting to connect to 127.0.0.1:9736.
The connection has been established to JEUS Master Server [adminServer] in the domain
[domain1].
JEUS 9 Administration Tool
To view help, use the 'help' command.
[MASTER]domain1.adminServer>
```

- 실행하면서 바로 서버에 접속 #2

다음은 IP 주소와 포트 번호를 지정해서 접속하는 경우에 대한 예제이다.

```
JEUS_HOME/bin$ jeusadmin -u administrator -p jeus -host 192.168.0.1 -port 9746
Attempting to connect to 192.168.0.1:9746.
The connection has been established to JEUS Master Server [adminServer] in the domain
[domain1].
JEUS 9 Administration Tool
To view help, use the 'help' command.
[MASTER]domain1.adminServer>
```

- 라이선스 정보 보기

다음은 JEUS 라이선스를 조회하는 예제이다.

```
JEUS_HOME/bin$ jeusadmin -licensedue
59 days left.

JEUS_HOME/bin$ jeusadmin -licenseinfo
===== LICENSE INFORMATION =====
=== License Type : TEST
=== License Edition : ENTERPRISE
=== Product Name : JEUS
=== Major Version : 9
=== Issued date : 20260423
=== Expiration date : 20260622
=== Hostname : device.hostname
=== Cpu limits : unlimited
=== Max user limits : 5
```

```
=== License Seq : JEU-xxxx-xxx-xxxx
```

```
=====
```

- SSL을 사용하여 jeusadmin에 접속

SSL을 사용하여 jeusadmin에 접속할 경우 jeusadmin 실행 스크립트 파일을 열어 아래 시스템 프로퍼티를 설정한다(시스템 프로퍼티의 값은 서버의 SSL 설정에 따라 다를 수 있다).

```
. . .
# execute jeusadmin
"${JAVA_HOME}/bin/java" -Xmx128m -classpath "${BOOTSTRAP_CLASSPATH}" ${TOOL_OPTION}
    -Djmx.remote.x.request.timeout=600000
    -Djeus.home="${JEUS_HOME}"
    -Djeus.tool.console.useJLine="false"
    -Djava.naming.factory.initial=jeus.jndi.JEUSContextFactory
    -Djava.naming.factory.url.pkgs=jeus.jndi.jns.url
    -Djava.util.logging.config.file="${JEUS_HOME}/bin/logging.properties"
    -Djeus.net.client.use-ssl=true
    -Djavax.net.ssl.trustStore=${JEUS_HOME}/domains/domain1/config/truststore
    -Djavax.net.ssl.trustStorePassword=changeit
    -Djavax.net.ssl.trustStoreType=JKS
    -Dssl.TrustManagerFactory.algorithm=SunX509
    ${JAVA_ARGS}
    jeus.tool.console.ConsoleBootstrapper ${BOOT_PARAMETER}
```

#### 4.2.1.2. exit

관리 작업을 완료하고 jeusadmin을 종료할 때 사용하는 명령어이다.

- alias

quit

- 사용법

```
exit
```

- 예제

```
[MASTER]domain1.server1>exit
JEUS_HOME/bin$
```

- 참고

본 명령어는 서버에 접속하지 않은 상태에서도 사용할 수 있다.

## 4.2.2. Local 명령어

다음은 서버 접속, 도움말, 히스토리 기능 등을 제공하는 기본 명령어들이다.

명령어	설명
<code>connect</code>	JEUS Master Server나 Managed Server에 접속을 시도한다.
<code>disconnect</code>	접속을 끊고 Off-line 상태로 돌아간다.
<code>help</code>	명령어에 대한 도움말을 출력한다.
<code>history</code>	입력했던 명령어들에 대한 기록을 출력한다.
<code>p</code>	입력했던 명령어들 중 하나를 골라 실행한다. 옵션을 주지 않고 실행하는 경우 직전에 실행했던 명령어를 다시 실행한다.
<code>remove-login-cache</code>	cachelogin 기능을 통해 저장된 특정 사용자 로그인 정보를 캐시 파일에서 제거한다.
<code>set-server-option</code>	server 옵션에 대한 기본값을 설정한다.
<code>set-servers-option</code>	servers 옵션에 대한 기본값을 설정한다.
<code>start-record</code>	입력하는 명령어들을 스크립트 형태로 저장하는 레코딩 기능을 시작한다.
<code>stop-record</code>	레코딩 기능을 중지한다.
<code>unset-server-option</code>	set-server-option으로 설정한 기본값을 삭제한다.
<code>unset-servers-option</code>	set-servers-option으로 설정한 기본값을 삭제한다.
<code>verbose</code>	명령어를 수행할 때 상세 출력 모드를 켜거나 끈다.

### 4.2.2.1. connect

서버에 접속을 시도한다. 사용자 이름과 비밀번호를 입력해 주어야한다. host를 입력받지 않으면 localhost로 접속을 시도한다.

- alias
- login
- 사용법

```
connect -u,--username <user-name>
        -p, --password <password>
        [-f <file-name>]
        [-cachelogin]
        [-h, --host <server-address>]
        [-port <server-port>]
        [-d, --domain <domain-name>]
```



- 파라미터

파라미터	설명
-u, --username <user-name>	접속할 때 사용할 사용자 이름을 지정한다.
-p, --password <password>	접속할 때 사용할 비밀번호를 지정한다.
[-f <file-name>]	접속할 때 사용할 사용자 이름과 비밀번호를 저장한 파일을 지정한다.
[-cachelgin]	접속할 때 사용한 사용자 이름과 비밀번호를 파일에 기록한다.
[-h, --host <server-address>]	접속할 서버의 주소를 지정한다. 아무 것도 입력하지 않으면 localhost로 접속을 시도한다.
[-port <server-port>]	접속할 서버의 포트를 지정한다. 별도로 지정하지 않는 경우 물어보지 않고 기본값을 사용한다. (기본값: 9736)
[-d, --domain <domain-name>]	도메인 이름을 지정한다. security.key 파일을 찾기 위해 사용된다.

- 예제

- -u 옵션과 -p 옵션을 주고, [-h] 옵션으로 접속할 호스트 정보를 입력하여 명령을 실행한다.

```
offline>connect -u administrator -p jeus -h 127.0.0.1
Attempting to connect to 127.0.0.1:9736.
The connection has been established to JEUS Master Server [adminServer] in the domain
[domain1].
[MASTER]domain1.adminServer>
```

- [-h] 옵션을 넣지 않은 경우 localhost로 접속하게 된다.

```
offline>connect -u administrator -p jeus
Attempting to connect to 127.0.0.1:9736.
The connection has been established to JEUS Master Server [adminServer] in the domain
[domain1].
[MASTER]domain1.adminServer>
```

- 참고

본 명령어는 서버에 접속하지 않은 상태에서도 사용할 수 있다.

#### 4.2.2.2. disconnect

접속을 끊고 Off-line 상태로 돌아가는 명령어이다.

- alias

logout

- 사용법

```
disconnect
```

- 예제

```
offline>connect
Enter the server address:
User name: administrator
Password:
Attempting to connect to 127.0.0.1:9736.
The connection has been established to JEUS Master Server [adminServer] in the domain [domain1].

[MASTER]domain1.adminServer>disconnect
The connection has been closed.
offline>
```

- 참고

본 명령어는 서버에 접속하지 않은 상태에서도 사용할 수 있다.

#### 4.2.2.3. help

명령어들에 대한 도움말을 출력한다. 별도의 옵션을 주지 않고 실행할 경우 도움말과 함께 현재 사용할 수 있는 명령어 목록, 명령어 기본 문법에 대한 설명을 조회한다.

- 사용법

```
help [-g,--group <group-name> | <command-name> | -l,--list]
```

- 파라미터

파라미터	설명
[-g,--group <group-name>]	주어진 그룹(group name)에 속한 명령어에 대한 목록과 간단한 설명을 출력한다.
[<command-name>]	특정 명령어(command name)에 대한 도움말을 출력한다.  명령어 이름, 별칭(alias), 사용법, 옵션에 대한 정보를 조회한다.  옵션을 생략하고 사용할 수도 있다.
[-l,--list]	모든 명령어에 대한 간단한 설명을 출력한다.

- 예제

- help 명령어만 사용할 경우

```

offline>help
LIST OF AVAILABLE COMMANDS
[Local]-----
  connect                disconnect
  exit                   help
  history                p
  remove-login-cache     set-server-option
  set-servers-option     start-record
  stop-record            unset-server-option
  unset-servers-option   verbose

[Server Management]-----
  local-shutdown

[Domain Configuration]-----
  apply-configuration-plan  create-domain
  delete-domain            list-domains
  pack-domain              unpack-domain

[Application]-----
  install-application      install-deployment-plan
  redeploy-application

[Library]-----
  install-library

[Web]-----
  precompile-jsp

[NodeManager]-----
  connect-nodemanager      disconnect-nodemanager
  nm-start-server          nm-state-server
  nm-stop-server           stop-nodemanager

COMMAND GRAMMAR
  Command Usage
    COMMAND [COMMAND_OPTIONS] [GLOBAL_OPTIONS] [(>|>>) FILENAME]
  Details
    COMMAND: target command to execute.
    [COMMAND_OPTIONS]: Options for target command.
    [GLOBAL_OPTIONS]: Common options for all commands.
      -verbose
        Prints the stack trace for errors.
      -repeat <count>
        Repeats the target command X times.
      -interval <seconds>
        Specifies the interval at which to repeat execution, in
        seconds.
    [(>|>>) FILENAME]
      Prints the result of the target command to the specified file.
  Examples
    help -l -repeat 3 -interval 10 > result.txt

To show detailed information for a command, use 'help [COMMAND_NAME]'.
ex) help connect
offline>

```

- help 명령어에 [-l] 옵션과 같이 사용하는 경우

```
offline>help -l
[Local]-----
  connect          Attempting to authenticate with the target
                   server.
  disconnect        Disconnects from the currently connected server.

  .....

[Server Management]-----
  local-shutdown    Shuts down the currently connected server.
  local-start-server On off-line mode, start managed server via SSH.
  You should specify -node, -domain, -server, -u,
  -p, -masterurl.
  On on-line mode, start standby server.

  .....

[NodeManager]-----
  connect-nodemanager Connects to the node manager.
  disconnect-nodemanager Closes the connection to the node manager.
  nm-start-server      Starts a server using the node manager.
  nm-state-server       Shows the state of a server using node manager.
  nm-stop-server        Stops a server using the node manager.
  stop-nodemanager      Stops the node manager.

To show detailed information for a command, use 'help [COMMAND_NAME]'.
ex) help connect
offline>
```

- help 명령어에 [-g] 옵션과 같이 사용하는 경우

다음은 'Server Management' 그룹 명령어 목록을 출력하는 예제이다.

```
offline>help -g "Server Management"
[Server Management]-----
  local-shutdown    Shuts down the currently connected server.
  local-start-server On off-line mode, start managed server via SSH.
  You should specify -node, -domain, -server, -u,
  -p, -masterurl.
  On on-line mode, start standby server.

To show detailed information for a command, use 'help [COMMAND_NAME]'.
ex) help connect
offline>
```

- [<command-name>] 옵션을 사용해 특정 명령어에 대한 도움말을 조회하는 경우

다음은 add-server 명령어 설명을 출력하는 예제이다.

```
[MASTER]domain1.adminServer>help add-server
NAMES
  add-server
```

Adds a new server to the domain configuration. If a server name is not provided, the current server list in this domain will be shown.

#### ALIAS

addserver

#### USAGE

```
add-server [-node,--nodeName <node-name>]
           [<server-name>]
           [-jvm,--jvmOptions <jvm-options>]
           [-logdir,--logHomeDirectory <server-log-home-directory>]
           [-a,--actionOnResourceLeak <action-on-resource-leak>]
           [-l,--logStdoutToRawFormat <print-stdout-to-raw-format>]
           [-m,--mejb <enable-MEJB>]
           [-c,--classFtp <enable-class-ftp-service>]
           [-target,--duptargetserver <target-server>]
           [-port,--baseport <base-port>]
           [-addr,--baseaddr <base-addr>]
           [-http,--httpport <http-port>]
           [-rg,--replicate-group <name>]
           [-f,--forceLock]
           [-detail]
```

#### OPTIONS

```
[-node,--nodeName <node-name>]
    name of the node this server is located

[<server-name>]
    name of the server you want to add

[-jvm,--jvmOptions <jvm-options>]
    jvm configurations applied to this server jvm

[-logdir,--logHomeDirectory <server-log-home-directory>]
    the log directory which has all the log files created by this server

[-a,--actionOnResourceLeak <action-on-resource-leak>]
    strategy when resource leak is detected. it must be one of NoAction,
    Warning, AutoClose

[-l,--logStdoutToRawFormat <print-stdout-to-raw-format>]
    [Dynamic] whether stdout is printed in row format or in JEUS Logger
    format.

[-m,--mejb <enable-MEJB>]
    [Dynamic] whether using MEJB or not .true, false

[-c,--classFtp <enable-class-ftp-service>]
    [Dynamic] whether using class ftp service or not .true, false

[-target,--duptargetserver <target-server>]
    duplicate the configuration of the target server

[-port,--baseport <base-port>]
    base service listen port of the server

[-addr,--baseaddr <base-addr>]
    base service listen address of the server

[-http,--httpport <http-port>]
    http listen port of the server
```

```
[-rg,--replicate-group <name>]
    Replicate Group Name

[-f,--forceLock]
    Acquires the configuration lock forcibly.

[-detail]
    Show detail results
```

- 참고

본 명령어는 서버에 접속하지 않은 상태에서도 사용할 수 있다.

#### 4.2.2.4. history

이전에 실행했던 명령어 목록을 조회한다.

명령어 목록은 `${user.home}/.jeusadmin/commandHistory`에 저장된다. 기록한 명령어를 다시 실행할 경우에는 `p` 명령어를 사용한다. 실행했던 명령어 기록은 최대 100개까지 가능하다.

- 사용법

```
history [-clear | <size>]
```

- 파라미터

파라미터	설명
[-clear]	실행했던 명령어 목록을 지우고 초기화한다.
[<size>]	출력할 명령어 개수를 설정한다. 최근에 실행한 순으로 <i>size</i> 만큼 출력한다.

- 예제

```
offline>history
1 : help history
2 : connect -u administrator -p jeus
3 : disconnect

To execute a command in the history, please use '!' command (e.g. ! 7)
offline>history 3
3 : disconnect

To execute a command in the history, please use '!' command (e.g. ! 7)
offline>history -clear
The command history has been cleared.
offline>
```

- 참고

본 명령어는 서버에 접속하지 않은 상태에서도 사용할 수 있다.

#### 4.2.2.5. p

실행했던 명령어를 다시 실행하려고 할 때 사용한다. 실행했던 명령어 목록은 [history](#) 명령어를 통해 확인할 수 있다.

- alias

!

- 사용법

```
p [<history-number>]
```

- 파라미터

파라미터	설명
[<history-number>]	명령어 기록에서 실행할 명령어의 번호를 지정한다.

- 예제

```
offline>history
1 : connect -u jeus -p jeus
2 : disconnect

To execute a command in the history, please use '!' command (e.g. ! 7)
offline>p 1
connect -u jeus -p jeus
Attempting to connect to 127.0.0.1:9736.
The connection has been established to JEUS Master Server [adminServer] in the domain [domain1].
[MASTER]domain1.adminServer>! 2
disconnect
The connection has been closed.
offline>
```

- 참고

본 명령어는 서버에 접속하지 않은 상태에서도 사용할 수 있다.

#### 4.2.2.6. remove-login-cache

cachelogin 기능을 통해 저장된 특정 사용자 로그인 정보를 캐시 파일에서 제거한다.

cachelogin 기능에 의해 저장된 로그인 정보는 <domain-name>과 <user-name>에 의해 구분된다.

- 사용법

```
remove-login-cache -domain <domain-name>
                  -u <user-name>
                  [-f <file-name>]
```

- 파라미터

파라미터	설명
-domain <domain-name>	로그인 정보를 저장할 때 사용했던 도메인 이름을 지정한다.
-u <user-name>	로그인 정보를 저장할 때 사용했던 사용자 이름을 지정한다.
[-f <file-name>]	사용자 이름과 비밀번호가 저장된 비밀번호 파일 이름을 지정한다.

- 예제

```
offline>remove-login-cache -domain domain1 -u user1
The cached login information [domain1:user1] has been successfully removed from
/home/OSUser/.jeusadmin/.jeuspasswd.
```

- 참고

본 명령어는 서버에 접속하지 않은 상태에서도 사용할 수 있다.

#### 4.2.2.7. set-server-option

-server 옵션에 대한 기본값을 설정할 수 있는 기능을 제공한다. 옵션을 주지 않고 사용할 경우 현재 설정한 기본값을 출력한다.

- alias

setserver

- 사용법

```
set-server-option <server-name>
```

- 파라미터

파라미터	설명
<server-name>	server 옵션에 대해 기본값으로 사용할 서버 이름을 설정한다.

- 예제

adminServer를 기본값으로 설정하여 **jndi-info** 명령어를 실행한다. set-server-option 명령어를 사용해



<server-name>를 지정해주기 전에는 jndi-info 명령어를 실행하면 -server 옵션이 주어지지 않았다는 오류가 발생하지만, 지정해 준 후에는 별도로 -server 옵션을 주지 않아도 adminServer에 대해 jndi-info 명령어가 동작함을 확인할 수 있다.

```
[MASTER]domain1.adminServer>jndi-info
The following options are missing: server

[MASTER]domain1.adminServer>set-server-option
The default server name is not set.

[MASTER]domain1.adminServer>set-server-option adminServer
The default server name has been set: adminServer.

[MASTER]domain1.adminServer>set-server-option
The current default server name: adminServer.

[MASTER]domain1.adminServer>jndi-info
The JNDI list on the adminServer
List of the context /
=====
+-----+-----+-----+
| Name | Value | Local Binding |
+-----+-----+-----+
| ConnectionFactory | jeus.jms.client.facility.factory.JeusC | false |
| | onnectionFactory | |
| ..... |
[MASTER]domain1.adminServer>
```

#### • 참고

본 명령어는 서버에 접속하지 않은 상태에서도 사용할 수 있다.

### 4.2.2.8. set-servers-option

-servers 옵션에 대한 기본값을 설정한다. 옵션을 주지 않고 실행한 경우 현재 설정되어 있는 기본값을 출력한다.

#### • alias

setservers

#### • 사용법

```
set-servers-option [<server-list>]
```

#### • 파라미터

파라미터	설명
[<server-list>]	servers 옵션에 대해 기본값으로 사용할 서버 이름들을 지정한다. 각 서버 이름은 콤마(,)로 구분한다.

- 예제

set-servers-option 명령어를 사용하여 deploy할 서버 목록을 지정한 후 deploy 명령어를 통해 지정한 서버 애플리케이션을 배포하는 예제이다.

```
[MASTER]domain1.adminServer>set-servers-option
The default server list is not set.

[MASTER]domain1.adminServer>set-servers-option adminServer,server1
The default server list has been set: adminServer,server1.

[MASTER]domain1.adminServer>set-servers-option
The current default server list: adminServer,server1.

[MASTER]domain1.adminServer>deploy sample
deploy the application for the application [sample] succeeded.

[MASTER]domain1.adminServer>appinfo
Application information for the domain [domain1].
=====
+-----+-----+-----+-----+-----+-----+
| Applicati|Application| State | Target | Target | Application Path |
| on ID    | Type      |       | Servers| Clusters|                   |
+-----+-----+-----+-----+-----+-----+
| sample   | WAR       | RUNNING | server1,adminServer |      | ${INSTALL_HOME}/sample/sample.war |
+-----+-----+-----+-----+-----+-----+
=====
```

- 참고

본 명령어는 서버에 접속하지 않은 상태에서도 사용할 수 있다.

#### 4.2.2.9. start-record

입력한 명령어들을 스크립트 파일에 기록하는 레코딩 기능을 시작한다.

지정한 레코딩 파일의 확장자에 따라 해당 스크립트로 명령어들이 저장된다. 레코딩이 시작되면 프롬프트에 "-Rec\*"이 추가된다.

- alias

startrecord

- 사용법

```
start-record [<recfile>]
```

- 파라미터

파라미터	설명
[<recfile>]	명령어 기록할 파일을 지정한다. 스크립트 파일로 저장이 되므로 현재 지원하는 스크립트의 확장자인 .py, .rb만 확장자로 사용이 가능하다.

- 예제

- 옵션을 주지 않고 실행한 경우는 record\_script\_[timestamp].py 파일에 저장한다.

```
offline>start-record
Starting recording to file : record_script_20150101123456.py
offline-Rec*>
```

- [<recfile>]을 지정한 경우 해당 파일에 저장한다.

```
offline>start-record record.py
Starting recording to file : record.py
offline-Rec*>
```

- 참고

본 명령어는 서버에 접속하지 않은 상태에서도 사용할 수 있다.

#### 4.2.2.10. stop-record

레코딩 기능을 중지한다.

- alias

stoprecord

- 사용법

```
stop-record
```

- 예제

```
offline-Rec*>stop-record
Stopping recording to file : recrod.py
offline>
```

- 참고

본 명령어는 서버에 접속하지 않은 상태에서도 사용할 수 있다.

#### 4.2.2.11. unset-server-option

set-server-option 명령어로 설정한 기본값을 삭제한다.

- alias

unsetserver

- 사용법

```
unset-server-option
```

- 예제

unset-server-option을 사용하여 set-server-option으로 설정한 기본값을 해제하였다. 해제한 후 **jndi-info** 명령어를 실행하면 server 옵션이 필요하다는 오류가 발생하는 것을 확인할 수 있다.

```
[MASTER]domain1.adminServer>set-server-option
The current default server name: adminServer.

[MASTER]domain1.adminServer>unset-server-option
The default server name has been unset.

[MASTER]domain1.adminServer>set-server-option
The default server name is not set.

[MASTER]domain1.adminServer>jndi-info
The following options are missing: server
```

- 참고

본 명령어는 서버에 접속하지 않은 상태에서도 사용할 수 있다.

#### 4.2.2.12. unset-servers-option

set-servers-option 명령어로 지정한 -servers 옵션에 대한 기본값 설정을 삭제한다.

- alias

unsetservers

- 사용법

```
unset-servers-option
```

- 예제

```
[MASTER]domain1.adminServer>set-servers-option
```

```
The current default server list: adminServer,server1.
```

```
[MASTER]domain1.adminServer>unset-servers-option  
The default server list has been unset.
```

```
[MASTER]domain1.adminServer>set-servers-option  
The default server list is not set.
```

```
[MASTER]domain1.adminServer>
```

- 참고

본 명령어는 서버에 접속하지 않은 상태에서도 사용할 수 있다.

### 4.2.2.13. verbose

명령어 수행 과정을 상세히 출력할지 여부를 설정한다. 옵션을 주지 않은 경우 현재 상세 출력이 켜져 있는지 꺼져 있는지 알려준다. 상세 출력이 켜져 있는 경우 명령어 수행 중 예외가 발생하면 Stack Trace를 출력하게 된다.

- 사용법

```
verbose [-off | -on]
```

- 파라미터

파라미터	설명
[-on]	상세 출력을 켜다.
[-off]	상세 출력을 끄다.

- 예제

- 현재 상세 출력이 켜져 있는지 여부를 확인하는 예제

```
offline>verbose  
Verbose output disabled.  
offline>
```

- 실제 사용 예제

```
offline>verbose  
Verbose output disabled.  
  
offline>help notexistcommand  
Command notexistcommand not found  
  
offline>verbose -on  
Verbose output is enabled.
```

```

offline>help notexistcommand
Command notexistcommand not found
jeus.tool.console.executor.CommandException: Command notexistcommand not found

.....

offline>verbose -off
Verbose output disabled.

offline>help notexistcommand
Command notexistcommand not found
offline>

```

- 참고

본 명령어는 서버에 접속하지 않은 상태에서도 사용할 수 있다.

### 4.2.3. Server Management 관련 명령어

다음은 Server Management 관련 명령어의 목록이다.

명령어	설명
<a href="#">add-custom-resource-to-servers</a>	동적으로 서버에 custom resource를 추가한다.
<a href="#">add-data-sources-to-server</a>	동적으로 서버에 데이터소스들을 등록한다.
<a href="#">add-external-resource-to-servers</a>	동적으로 서버에 external resource를 추가한다.
<a href="#">add-invocation</a>	특정 Lifecycle Invocation에 Invocation을 추가한다.
<a href="#">add-invocation-library</a>	특정 Lifecycle Invocation에 Library Ref를 추가한다.
<a href="#">add-jvm-option</a>	특정 서버에 JVM 설정을 추가한다.
<a href="#">add-lifecycle-invocation</a>	특정 서버 혹은 클러스터에 Lifecycle Invocation을 추가한다.
<a href="#">add-listener</a>	특정 서버에 리스너를 추가한다.
<a href="#">add-log-handler</a>	동적으로 특정 서버의 특정 로거에 핸들러를 추가한다.
<a href="#">add-logger</a>	동적으로 특정 서버에 로거를 추가한다.
<a href="#">config-listener-ssl</a>	Listener의 SSL을 설정한다.
<a href="#">disable-engine-init-on-boot</a>	서버의 내부 엔진(Servlet, EJB, JMS)의 초기화 시점을 애플리케이션 deploy할 때로 설정한다.
<a href="#">disable-engines</a>	서버의 내부 엔진(Servlet, EJB, JMS)을 선택적으로 사용하지 않도록 설정한다.
<a href="#">dump</a>	특정 서버 또는 특정 클러스터에 속한 서버들의 Thread Dump를 찍는다.
<a href="#">enable-engine-init-on-boot</a>	서버의 내부 엔진(Servlet, EJB, JMS)의 초기화 시점을 서버 부팅시로 설정한다.
<a href="#">enable-engines</a>	서버의 내부 엔진(Servlet, EJB, JMS)을 선택적으로 사용하도록 설정한다.

명령어	설명
jndi-info	특정 서버의 JNDI 컨텍스트에 바인딩되어 있는 객체의 바인딩(이름) 목록을 확인한다.
lifecycle-invocation-info	특정 Lifecycle Invocation의 정보를 조회한다.
list-blocked-clients	접속이 차단된 클라이언트 주소들을 조회한다.
list-jvm-options	특정 서버의 JVM 설정을 조회한다.
list-lifecycle-invocations	특정 서버 혹은 클러스터의 Lifecycle Invocation을 조회한다.
list-log-handlers	특정 서버의 특정 로거에 등록된 핸들러들의 정보를 조회한다.
list-loggers	특정 서버에 등록된 로거들의 정보를 조회한다.
list-server-listeners	특정 서버에 있는 리스너들의 설정을 조회한다.
local-shutdown	현재 접속해 있는 서버를 종료한다.
local-start-server	On-line 모드에서는 현재 접속해 있는 서버가 STANDBY 상태일 경우 서버를 시작한다.
log-level	특정 서버의 특정 로거와 해당 로거에 등록된 핸들러들의 레벨을 확인한다.
mbean-info	특정 서버에 등록된 MBean들의 리스트를 조회한다.
memory-info	특정 서버 또는 특정 클러스터에 속한 서버들의 메모리 정보를 확인한다.
modify-invocation	특정 Lifecycle Invocation의 Invocation을 수정한다.
modify-invocation-library	특정 Lifecycle Invocation의 Library Ref를 수정한다.
modify-jvm-option	특정 서버의 JVM 설정을 수정한다.
modify-lifecycle-invocation	특정 서버 혹은 클러스터의 Lifecycle Invocation을 수정한다.
modify-listener	특정 서버에 있는 리스너의 설정을 수정한다.
modify-log-handler	특정 서버의 특정 로거에 등록된 특정 핸들러를 수정한다.
modify-logger	동적으로 특정 서버에 등록된 특정 로거를 수정한다.
remove-custom-resource-from-servers	서버에 등록된 custom resource를 동적으로 삭제한다.
remove-data-sources-from-server	서버에 등록된 데이터소스들을 동적으로 제거한다.
remove-external-resource-from-servers	서버에 등록된 external resource를 동적으로 삭제한다.
remove-invocation	특정 Lifecycle Invocation의 Invocation을 제거한다.
remove-invocation-library	특정 Lifecycle Invocation의 Library Ref를 제거한다.
remove-jvm-option	특정 서버의 JVM 설정을 제거한다.
remove-lifecycle-invocation	특정 서버 혹은 클러스터의 Lifecycle Invocation을 제거한다.
remove-listener	특정 서버에 있는 리스너를 삭제한다.
remove-log-handler	동적으로 특정 서버의 특정 로거에 등록된 특정 핸들러를 삭제한다.
remove-logger	동적으로 특정 서버에 등록된 특정 로거를 삭제한다.

명령어	설명
<code>remove-replicate-group</code>	동적으로 특정 서버 또는 서버 템플릿에 지정한 Replicate Group을 삭제한다.
<code>resume-server</code>	일시 정지된 특정 서버를 다시 서비스되도록 한다.
<code>run-garbage-collection</code>	특정 서버 또는 특정 클러스터에 속한 서버들에 full gc를 수행한다.
<code>server-info</code>	서버들의 현재 상태를 확인한다.
<code>server-log</code>	특정 서버의 로그를 command에서 확인한다.
<code>set-replicate-group</code>	동적으로 특정 서버 또는 서버 템플릿에 Replicate Group을 지정한다.
<code>show-replicate-group</code>	서버 또는 서버 템플릿의 Replicate Group을 확인한다.
<code>show-current-scf-view</code>	현재 SCF View를 모니터링한다.
<code>start-cluster</code>	특정 클러스터에 속한 MS들을 시작한다. JEUS Master Server-Managed Server 구조에서만 수행 가능한 명령어이다.
<code>start-domain</code>	현재 도메인에 속한 모든 MS들을 시작한다. JEUS Master Server-Managed Server 구조에서만 수행 가능한 명령어이다.
<code>start-node</code>	특정 노드에 속한 MS들을 시작한다. JEUS Master Server-Managed Server 구조에서만 수행 가능한 명령어이다.
<code>start-server</code>	특정 MS를 시작한다.
<code>stop-cluster</code>	특정 클러스터에 속한 Managed Server들을 종료한다. 클러스터 내에 JEUS Master Server가 포함된 경우 JEUS Master Server는 대상에서 제외한다. JEUS Master Server-Managed Server 구조에서만 수행 가능한 명령어이다.
<code>stop-domain</code>	현재 도메인에 속한 Managed Server들을 종료한다. JEUS Master Server-Managed Server 구조에서만 수행 가능한 명령어이다.
<code>stop-node</code>	특정 노드에 속한 Managed Server들을 종료한다. 노드 내에 JEUS Master Server가 포함된 경우 JEUS Master Server는 대상에서 제외한다. JEUS Master Server-Managed Server 구조에서만 수행 가능한 명령어이다.
<code>stop-server</code>	특정 Managed Server를 종료한다.
<code>suspend-server</code>	서비스 중인 특정 서버를 일시 정지시킨다.
<code>system-info</code>	특정 서버의 시스템 정보를 확인한다.
<code>unblock-client</code>	특정 클라이언트 주소에 대한 접속 차단을 해제한다.

#### 4.2.3.1. add-custom-resource-to-servers

동적으로 서버에 custom resource를 추가한다.

- alias  
add-cr-to-servers



- 사용법

```
add-custom-resource-to-servers <export-name>
                                -servers <server-list>
                                [-f,--forceLock]
                                [-detail]
```

- 파라미터

파라미터	설명
<export-name>	서버에 추가할 custom resource의 JNDI 바인드 이름이다.
-servers <server-list>	custom resource를 주어진 server list에 등록한다.
[-f,--forceLock]	설정 변경을 강제로 진행하여 반영한다.
[-detail]	동적 변경 결과에 대한 상세한 정보가 출력된다.

- 예제

```
[MASTER]domain1.adminServer>add-custom-resource-to-servers custom/dog -servers server1
Successfully performed the ADD operation for A custom resource.
Check the results using "list-custom-resources"
```

#### 4.2.3.2. add-data-sources-to-server

동적으로 서버에 데이터소스들을 등록한다.

- alias

adddstosvr

- 사용법

```
add-data-sources-to-server -server <server-name>
                            [-ids <data-source-id-list> | -all]
                            [-f,--forceLock]
                            [-detail]
```

- 파라미터

파라미터	설명
-server <server-name>	데이터소스를 등록할 서버의 이름이다.
[-ids <data-source-id-list>]	서버에 등록할 데이터소스들의 ID 목록이다.  한 개 이상인 경우 콤마(,)로 구분한다.

파라미터	설명
[-all]	도메인에 정의된 모든 데이터소스를 서버에 등록한다.
[-f,--forceLock]	설정 변경을 강제로 진행하여 반영한다.
[-detail]	동적 변경 결과에 대한 상세한 정보가 출력된다.

- 예제

```
[MASTER]domain1.adminServer>add-data-sources-to-server -server server1 -all
Successfully performed the ADD operation for data sources to the server [server1].
Check the results using "add-data-sources-to-server -server server1"
```

#### 4.2.3.3. add-external-resource-to-servers

동적으로 서버에 external resource를 추가한다.

- alias

add-er-to-servers

- 사용법

```
add-external-resource-to-servers <export-name>
                                -servers <server-list>
                                [-f,--forceLock]
                                [-detail]
```

- 파라미터

파라미터	설명
<export-name>	서버에 추가할 external resource의 이름이다.
-servers <server-list>	external resource를 주어진 server list에 등록한다.
[-f,--forceLock]	설정 변경을 강제로 진행하여 반영한다.
[-detail]	동적 변경 결과에 대한 상세한 정보가 출력된다.

- 예제

```
[MASTER]domain1.adminServer>add-external-resource-to-servers test/ext -servers server1
Successfully performed the ADD operation for A external resource.
Check the results using "list-external-resources"
```

#### 4.2.3.4. add-invocation

특정 Lifecycle Invocation에 Invocation을 추가한다.

- alias

addinvo, addinvocation

- 사용법

```
add-invocation <method>
    -type <invocation-type>
    [-params <method-params>]
    [-args <invocation-argument>]
    -class <class>
    -s,--server <server> | -c,--cluster <cluster>
    [-f,--forceLock]
    [-detail]
```

- 파라미터

파라미터	설명
<method>	추가할 Invocation의 메서드 이름이다.
-type <invocation-type>	Invocation의 타입(메서드 호출 시점)을 지정한다.  다음 중 하나를 지정할 수 있다. <ul style="list-style-type: none"><li>• BOOT</li><li>• BEFORE_DEPLOY</li><li>• AFTER_DEPLOY</li><li>• READY</li><li>• BEFORE_UNDEPLOY</li><li>• AFTER_UNDEPLOY</li></ul> 타입에 대한 자세한 내용은 JEUS Server 안내서의 "Lifecycle Invocation 설정"의 Invocation Type을 참고한다.
[-params <method-params>]	메서드 파라미터의 Fully Qualified Class Name을 지정한다. 매개 변수가 여러 있을 경우 공백으로 구분한다.  (설정값 예시: -params java.lang.String boolean)
[-args <invocation-argument>]	메서드를 호출할 때 사용하는 Argument를 지정한다. 인자가 여러 개 있을 경우 공백으로 구분한다. 만약 공백이 포함된 값을 인자로 해야할 경우 큰따옴표로 묶어 사용하면 된다.  (설정값 예시: -args "tmaxsoft jeus" "hello world")

파라미터	설명
-class <class>	Invocation이 추가될 Lifecycle Invocation의 클래스 이름을 지정한다.
-s,--server <server>   -c,--cluster <cluster>	Lifecycle Invocation가 위치한 서버 혹은 클러스터의 이름을 지정한다.
[-f,--forceLock]	설정 변경을 강제로 진행하여 반영한다.
[-detail]	설정 변경 결과에 대한 상세한 정보가 출력된다.

- 예제

```
[MASTER]domain1.adminServer>add-invocation methodName -type BOOT -class invo.invoTestClass -s
server1 -params java.lang.String boolean -args "invocation test" true
Successfully performed the ADD operation for Invocation [methodName](Invocation ID = 0),
but all changes were non-dynamic. They will be applied after restarting.
Check the results using "list-lifecycle-invocations or lifecycle-invocation-info".
```

#### 4.2.3.5. add-invocation-library

특정 Lifecycle Invocation에 Library Ref를 추가한다.

- alias

addinvolib, addinvocationlibrary

- 사용법

```
add-invocation-library <library>
    -specVer <version>
    [-specMatch <true | false>]
    -implVer <version>
    [-implMatch <true | false>]
    [-fail,--failOnError <true | false>]
    -class <class>
    -s,--server <server> | -c,--cluster <cluster>
    [-f,--forceLock]
    [-detail]
```

- 파라미터

파라미터	설명
<library>	추가할 Library Ref의 라이브러리 이름이다.
-specVer <version>	추가할 라이브러리의 Specification Version을 지정한다.
[-specMatch <true   false>]	Specification Version이 정확히 일치해야 하는지 여부를 지정한다. (기본값: false)

파라미터	설명
-implVer <version>	추가할 라이브러리의 Implementation Version을 지정한다.
[-implMatch <true   false>]	Implementation Version이 정확히 일치해야 하는지 여부를 지정한다. (기본값: false)
[-fail,--failOnError <true   false>]	해당 Shared Library를 찾지 못할 경우 Deploy를 실패할지 여부를 설정한다. (기본값: false)
-class <class>	Library Ref가 추가될 Lifecycle Invocation의 클래스 이름을 지정한다.
-s,--server <server>   -c,--cluster <cluster>	Lifecycle Invocation가 위치한 서버 혹은 클러스터의 이름을 지정한다.
[-f,--forceLock]	설정 변경을 강제로 진행하여 반영한다.
[-detail]	설정 변경 결과에 대한 상세한 정보가 출력된다.

- 예제

```
[MASTER]domain1.adminServer>add-invocation-library libName -specVer 1 -implVer 1 -class
invo.invoTestClass -s server1
Successfully performed the ADD operation for Invocation Library [libName], but all changes were
non-dynamic.
They will be applied after restarting.
Check the results using "list-lifecycle-invocations or lifecycle-invocation-info".
```

#### 4.2.3.6. add-jvm-option

특정 서버에 JVM 설정을 추가한다.

- alias

add-jvmopt, addjvmopt, add-jvm-config, add-jvmcfg, addjvmcfg

- 사용법

```
add-jvm-option -server <server-name>
                [-opt,--jvmOption <jvm-option>]
                [-f,--forceLock]
                [-detail]
```

- 파라미터

파라미터	설명
-server <server-name>	서버의 이름이다.
[-opt,--jvmOption <jvm-option>]	추가할 JVM 설정이다.
[-f,--forceLock]	설정 변경을 강제로 진행하여 반영한다.

파라미터	설명
[-detail]	설정 변경 결과에 대한 상세한 정보가 출력된다.

- 예제

```
[MASTER]domain1.adminServer>add-jvm-option -server server1 -opt "-Xmx256m -XX:MaxPermSize=128m"
Successfully performed the ADD operation for JVM configuration for the server
(sever1)., but all changes were non-dynamic. They will be applied after restarting.
Check the results using "list-jvm-options or add-jvm-option"
```

#### 4.2.3.7. add-lifecycle-invocation

특정 서버 혹은 클러스터에 Lifecycle Invocation을 추가한다.

- alias

addlifeinvo, addlifecycleinvocation

- 사용법

```
add-lifecycle-invocation <class>
    -s,--server <server> | -c,--cluster <cluster>
    [-m,--method <invocation-method>]
    [-type <invocation-type>]
    [-params <method-params>]
    [-args <invocation-argument>]
    [-f,--forceLock]
    [-detail]
```

- 파라미터

파라미터	설명
<class>	추가할 Lifecycle Invocation의 클래스 이름이다.
-s,--server <server>   -c,--cluster <cluster>	Lifecycle Invocation을 추가할 서버 혹은 클러스터의 이름을 지정한다.
[-m,--method <invocation-method>]	<b>[Invocation 옵션]</b> Lifecycle Invocation을 추가하면서 Invocation을 함께 추가하려면 Invocation의 메서드 이름을 지정한다. Invocation을 추가하려면 Invocation Type이 필수이므로, <b>[Invocation 옵션]</b> 중 -type 옵션을 함께 필수로 지정해야 한다.

파라미터	설명
<code>[-type &lt;invocation-type&gt;]</code>	<p><b>[Invocation 옵션]</b> Invocation의 타입(메서드 호출 시점)을 지정한다. Lifecycle Invocation을 추가하면서 Invocation을 함께 추가할 경우 <b>[Invocation 옵션]</b> 중 <code>-m,--method</code> 옵션과 함께 필수로 지정해야 한다.</p> <p>다음 중 하나를 지정할 수 있다.</p> <ul style="list-style-type: none"> <li>• BOOT</li> <li>• BEFORE_DEPLOY</li> <li>• AFTER_DEPLOY</li> <li>• READY</li> <li>• BEFORE_UNDEPLOY</li> <li>• AFTER_UNDEPLOY</li> </ul> <p>타입에 대한 자세한 내용은 JEUS Server 안내서의 "Lifecycle Invocation 설정"의 Invocation Type을 참고한다.</p>
<code>[-params &lt;method-params&gt;]</code>	<p><b>[Invocation 옵션]</b> 메서드 파라미터의 Fully Qualified Class Name을 지정한다. 매개 변수가 여러 개인 경우 공백으로 구분한다.</p> <p>(설정값 예시: <code>-params java.lang.String boolean</code>)</p>
<code>[-args &lt;invocation-argument&gt;]</code>	<p><b>[Invocation 옵션]</b> 메서드를 호출할 때 사용하는 Argument를 지정한다. 인자가 여러 개인 경우 공백으로 구분한다. 만약 공백이 포함된 값을 인자로 해야 할 경우 큰따옴표로 묶어 사용하면 된다.</p> <p>(설정값 예시: <code>-args "tmaxsoft jeus" "hello world"</code>)</p>
<code>[-f,--forceLock]</code>	설정 변경을 강제로 진행하여 반영한다.
<code>[-detail]</code>	설정 변경 결과에 대한 상세한 정보가 출력된다.

#### • 예제

```
[MASTER]domain1.adminServer>add-lifecycle-invocation sampleClass1 -s server1
Successfully performed the ADD operation for Lifecycle Invocation Class [sampleClass1],
but all changes were non-dynamic. They will be applied after restarting.
Check the results using "list-lifecycle-invocations".

[MASTER]domain1.adminServer>add-lifecycle-invocation sampleClass2 -s server1 -m methodName -type
BOOT
Successfully performed the ADD operation for Lifecycle Invocation Class [sampleClass2] and
Invocation
[methodName](Invocation ID = 0), but all changes were non-dynamic. They will be applied after
restarting.
Check the results using "list-lifecycle-invocations".
```

#### 4.2.3.8. add-listener

동적으로 특정 서버에 리스너를 추가한다.

- alias

addlistener, createlister

- 사용법

```
add-listener -server <server-name>
              -name <listener-name>
              [-addr <address>]
              -port <port>
              [-selectors <selectors>]
              [-dual]
              [-backlog <backlog>]
              [-timeout <read-timeout>]
              [-keepaliveTimeout <keepalive-timeout>]
              [-rt,--reservedthreads <reserved-threads>]
              [-f,--forceLock]
```

- 파라미터

파라미터	설명
-server <server-name>	서버의 이름이다.
-name <listener-name>	추가할 리스너의 이름이다.
[-addr <address>]	추가할 리스너의 주소이다.
-port <port>	추가할 리스너의 포트 번호이다. 1 ~ 65535 사이의 값을 설정해야 하며, 다른 리스너의 포트 번호와 겹치지 않아야 한다.
[-selectors <selectors>]	사용할 Selector의 개수이다.
[-dual]	read/write 작업에 각각 별개의 Selector를 사용한다.
[-backlog <backlog>]	추가할 리스너의 backlog이다.
[-timeout <read-timeout>]	<b>[동적 변경]</b> 추가할 리스너의 read 타임아웃이다.
[-keepaliveTimeout <keepalive-timeout>]	사용하지 않는 소켓을 닫을 때까지의 시간을 설정한다.
[-rt,--reservedthreads <reserved-threads>]	추가할 리스너가 사용할 Reserved Thread의 개수이다.
[-f,--forceLock]	추가를 위한 lock을 강제로 취한다.

- 예제

```
[MASTER]domain1.adminServer>add-listener -server server1 -name Test -port 10000
Executed Successfully
```



```

Check the result using 'list-server-listeners -server server1 -name Test.
[MASTER]domain1.adminServer>list-server-listeners -server server1 -name Test
=====
Test

+-----+-----+
| address          | 0.0.0.0 |
| port             | 10000   |
| selectors        | 1       |
| dual-selector    | false   |
| backlog          | 128     |
| read-timeout     | 30000   |
| keepalive-timeout | not-set |
| reserved-threads | 0       |
+-----+-----+
=====

```

#### 4.2.3.9. add-log-handler

동적으로 특정 서버의 특정 로거에 핸들러를 추가한다.

- alias

addloghandler, add-handler, addhandler

- 사용법

```

add-log-handler -server <server-name>
                -logger <logger-name>
                [<handler-name>]
                [-level <handler-level>]
                [-filter <filter-class>]
                [-encoding <encoding>]
                [-filename <file-name>]
                [-enable <true | false>]
                [-count <rotation-count>]
                [-dir <rotation-dir>]
                [-buffer <buffer-size>]
                [-append <true | false>]
                [-hour <hour> | -day <day> | -size <size>]
                [-f,--forceLock]
                [-detail]
                [-permission <permission>]
                [-chown <chown>]

```

- 파라미터

파라미터	설명
-server <server-name>	서버의 이름이다.

파라미터	설명
-logger <logger-name>	핸들러를 추가할 로거의 이름이다.
[<handler-name>]	추가할 핸들러의 이름이다.
[-level <handler-level>]	<b>[동적 변경]</b> 핸들러의 로그 레벨이다. (기본값: FINEST)
[-filter <filter-class>]	로그 메시지의 필터 클래스 이름이다.
[-encoding <encoding>]	로그 메시지의 인코딩 값이다.
[-filename <file-name>]	로그를 기록할 파일 이름이다.  (기본값: loggerName.log. jeus logger의 경우는 JeusServer.log)
[-enable <true   false>]	파일 로테이션을 가능하게 할지 여부이다. (기본값: true)
[-count <rotation-count>]	백업하는 로그 파일의 최대 개수이다.
[-dir <rotation-dir>]	백업하는 로그 파일이 위치하는 디렉터리 경로이다.
[-buffer <buffer-size>]	로그 메시지를 파일에 기록할 때의 버퍼 사이즈이다.
[-append <true   false>]	재부팅할 때 이전 파일에 계속 로그를 남길지 여부이다.  (true   false, 기본값: true)
[-hour <hour>]	입력된 시간마다 새로운 로그 파일을 작성한다.
[-day <day>]	입력된 날마다 새로운 로그 파일을 작성한다.
[-size <size>]	입력된 사이즈에 따라 새로운 로그 파일을 작성한다.
[-f,--forceLock]	설정 변경을 강제로 진행하여 반영한다.
[-detail]	동적 변경 결과에 대한 상세한 정보가 출력된다.
[-permission <permission>]	입력된 퍼미션으로 로그파일을 설정한다. (설정값 예시: rwxrwxrwx)
[-chown <chown>]	입력된 소유자로 로그파일을 설정한다. owner와 group은 콤마(,)로 구분한다. (설정값 예시: owner-id,group-id)

- 예제

```
[MASTER]domain1.adminServer>add-log-handler -server server1 -logger jeus.jndi fileHandler -day 10
Successfully performed the ADD operation for The handler for the logger(jeus.jndi)
on the server(server1)., but all changes were non-dynamic. They will be applied
after restarting.
Check the results using "list-log-handlers or add-log-handler"
```

#### 4.2.3.10. add-logger

동적으로 특정 서버에 로거를 추가한다.

- alias

addlogger

- 사용법

```
add-logger -server <server-name>
            [<logger-name>]
            [-level <log-level>]
            [-useParentHandlers <true | false>]
            [-filter <filter-class>]
            [-formatter <formatter-class>]
            [-f,--forceLock]
            [-detail]
```

- 파라미터

파라미터	설명
-server <server-name>	서버의 이름이다.
[<logger-name>]	서버에 추가할 로거의 이름이다.
[-level <log-level>]	<b>[동적 변경]</b> 로거의 로그 레벨이다.
[-useParentHandlers <true   false>]	<b>[동적 변경]</b> 상위 로거의 핸들러를 사용할지 여부이다. (true   false)
[-filter <filter-class>]	출력되는 로그 메시지의 필터 클래스 이름이다.
[-formatter <formatter-class>]	로거에 적용할 Formatter의 클래스 이름이다.
[-f,--forceLock]	설정 변경을 강제로 진행하여 반영한다.
[-detail]	동적 변경 결과에 대한 상세한 정보가 출력된다.

- 예제

```
[MASTER]domain1.adminServer>add-logger -server server1 jeus.security -level FINE
Successfully performed the ADD operation for The logger for the server(server1).
, but all changes were non-dynamic. They will be applied after restarting.
Check the results using "list-loggers or add-logger"
```

#### 4.2.3.11. config-listener-ssl

Listener의 SSL을 설정한다.

- alias

configssl

- 사용법

```
config-listener-ssl -server <server-name>
                    -name <listener-name>
                    -set | -del | -show
```

```

[-need | -want | -noauth,--unnecessary]
[-p,--protocol <protocol>]
[-cs,--cipherSuite <cipher-suites>]
[-ks,--keystore <keystore-file>]
[-kp,--keystorePass <keystore-pass>]
[-kkp,--keystoreKeypass <keystore-keypass>]
[-kt,--keystoreType <keystore-type>]
[-kma,--keyManagementAlgorithm <key-management-algorithm>]
[-alias,--keyAlias <key-alias>]
[-ts,--truststore <truststore-file>]
[-tp,--truststorePass <truststore-pass>]
[-tt,--truststoreType <truststore-type>]
[-tma,--trustManagementAlgorithm <trust-management-algorithm>]
[-crl <crl-file>]
[-f,--forceLock]

```

#### • 파라미터

파라미터	설명
-server <server-name>	서버의 이름이다.
-name <listener-name>	리스너의 이름이다.
-set   -del   -show	지정한 리스너의 SSL 관련 내용을 설정할 것인지 해제할 것인지 설정 상태를 보여줄 것인지 선택한다.
[-need   -want   -noauth,--unnecessary]	클라이언트 인증 방식을 선택한다.
[-p,--protocol <protocol>]	지정한 리스너에서 사용할 SSL 프로토콜의 종류를 선택한다.
[-cs,--cipherSuite <cipher-suites>]	지정한 리스너에서 사용할 암호화 방식을 선택한다.
[-ks,--keystore <keystore-file>]	지정한 리스너에서 사용할 서버 비밀키와 그에 매칭되는 인증서를 저장한 파일을 지정한다.
[-kp,--keystorePass <keystore-pass>]	지정한 keystore 파일을 열기 위한 암호값이다.
[-kkp,--keystoreKeypass <keystore-keypass>]	지정한 keystore 파일에 저장된 서버 비밀키에 대한 암호값이다.
[-kt,--keystoreType <keystore-type>]	지정한 keystore의 타입이다.
[-kma,--keyManagementAlgorithm <key-management-algorithm>]	keystore에 저장된 key의 관리 알고리즘이다.
[-alias,--keyAlias <key-alias>]	인증서의 Key Alias이다.
[-ts,--truststore <truststore-file>]	서버 인증서를 저장하고 있는 파일을 지정한다.
[-tp,--truststorePass <truststore-pass>]	지정한 truststore 파일을 열기 위한 암호값이다.

파라미터	설명
<code>[-tt,--truststoreType &lt;truststore-type&gt;]</code>	지정한 truststore의 타입이다.
<code>[-tma,--trustManagementAlgorithm &lt;trust-management-algorithm&gt;]</code>	truststore에 저장된 key의 관리 알고리즘이다.
<code>[-crl &lt;crl-file&gt;]</code>	Certificate Revocation Lists를 저장하고 있는 파일을 지정한다.
<code>[-f,--forceLock]</code>	설정 변경을 강제로 진행하여 반영한다.

- 예제

```
[MASTER]domain1.adminServer>config-listener-ssl -server server1 -name Test -set
Executed successfully, but some configurations were not applied dynamically. It might be
necessary to restart the server.
```

#### 4.2.3.12. disable-engine-init-on-boot

서버의 내부 엔진(Servlet, EJB, JMS, ProObject)의 초기화 시점을 애플리케이션 deploy할 때로 설정한다. 서버를 다시 시작해야 적용된다.

- 관련 스키마

jeus-domain.xsd

- alias

lazyengine

- 사용법

```
disable-engine-init-on-boot <server-list>
                        [-f,--forceLock]
```

- 파라미터

파라미터	설명
<code>&lt;server-list&gt;</code>	서버의 이름이다. 서버가 한 개 이상인 경우는 콤마(,)로 구분한다.
<code>[-f,--forceLock]</code>	설정 변경을 강제로 진행하여 반영한다.

- 예제

```
[MASTER]domain1.adminServer>disable-engine-init-on-boot adminServer,server1
EngineInitOnBoot was successfully disabled.
Applying configuration ...
```

```

=====
+-----+
|                                     |
|                               Result |
|-----+
| Successfully changed only the JEUS Domain Configuration. |
| Restart the server to apply the changes.                 |
|-----+
=====

```

#### 4.2.3.13. disable-engines

서버의 내부 엔진(Servlet, EJB, JMS, ProObject)을 사용하지 않도록 설정한다. 서버를 다시 시작해야 적용된다.

- 관련 스키마

jeus-domain.xsd

- alias

engineoff

- 사용법

```

disable-engines <server-list>
                [-all]
                [-web]
                [-ejb]
                [-jms]
                [-f,--forceLock]

```

- 파라미터

파라미터	설명
<server-list>	서버의 이름이다. 서버가 한 개 이상인 경우는 콤마(,)로 구분한다.
[-all]	Web, EJB, JMS 엔진에 대하여 모두 설정한다.
[-web]	Web 엔진에 대하여 설정한다.
[-ejb]	EJB 엔진에 대하여 설정한다.
[-jms]	JMS 엔진에 대하여 설정한다.
[-f,--forceLock]	설정 변경을 강제로 진행하여 반영한다.

- 예제

```

[MASTER]domain1.adminServer>disable-engines adminServer -all
[adminServer]
Change Engine to Disabled: Web EJB JMS

Applying configuration ...

```

```

=====
+-----+
|                      Result                      |
+-----+
| Successfully changed only the JEUS Domain Configuration. |
| Restart the server to apply the changes.                |
+-----+
=====
...

[MASTER]domain1.adminServer>disable-engines server1 -all
[server1]
Change Engine to Disabled: Web EJB JMS

Applying configuration ...
=====
+-----+
|                      Result                      |
+-----+
| Successfully changed only the JEUS Domain Configuration. |
| Restart the server to apply the changes.                |
+-----+
=====
...

[MASTER]domain1.adminServer>disable-engines server2 -web -ejb
[server2]
Change Engine to Disabled: Web EJB

Applying configuration ...
=====
+-----+
|                      Result                      |
+-----+
| Successfully changed only the JEUS Domain Configuration. |
| Restart the server to apply the changes.                |
+-----+
=====
...

```

#### 4.2.3.14. dump

특정 서버 또는 특정 클러스터에 속한 서버들의 Thread Dump를 찍는다. 해당 서버의 서버 로그에 기록된다.

- 사용법

```
dump [-clusters <cluster-list> | -servers <server-list>]
```

- 파라미터

파라미터	설명
[-clusters <cluster-list>]	클러스터의 이름이다. 클러스터가 한 개 이상인 경우는 콤마(,)로 구분한다.

파라미터	설명
[-servers <server-list>]	서버의 이름이다. 서버가 한 개 이상인 경우는 콤마(,)로 구분한다.

- 예제

```
[MASTER]domain1.adminServer>dump -servers server1
Dumping to the servers.
The result of executing a dump of server processes.
=====
+-----+-----+
| Server |          Result          |
+-----+-----+
| server1| Successfully dumped.     |
+-----+-----+
=====
```

#### 4.2.3.15. enable-engine-init-on-boot

서버의 내부 엔진(Servlet, EJB, JMS, ProObject)의 초기화 시점을 서버가 시작하는 시점으로 설정한다. 서버를 다시 시작해야 적용된다.

- 관련 스키마

jeus-domain.xsd

- alias

eagerengine

- 사용법

```
enable-engine-init-on-boot <server-list>
                        [-f,--forceLock]
```

- 파라미터

파라미터	설명
<server-list>	서버의 이름이다. 서버가 한 개 이상인 경우는 콤마(,)로 구분한다.
[-f,--forceLock]	설정 변경을 강제로 진행하여 반영한다.

- 예제

```
[MASTER]domain1.adminServer>enable-engine-init-on-boot adminServer
EngineInitOnBoot was successfully enabled.
Applying configuration ...
=====
+-----+-----+
```



```

|                                     Result                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Successfully changed only the JEUS Domain Configuration.                     |
| Restart the server to apply the changes.                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
=====

```

#### 4.2.3.16. enable-engines

서버의 내부 엔진(Servlet, EJB, JMS, ProObject)을 사용하도록 설정한다. 서버를 다시 시작해야 적용된다.

- 관련 스키마

jeus-domain.xsd

- alias

engineon

- 사용법

```

enable-engines <server-list>
                [-all]
                [-web]
                [-ejb]
                [-jms]
                [-f,--forceLock]

```

- 파라미터

파라미터	설명
<server-list>	서버의 이름이다. 서버가 한 개 이상인 경우는 콤마(,)로 구분한다.
[-all]	Web, EJB, JMS 엔진에 대하여 모두 설정한다.
[-web]	Web 엔진에 대하여 설정한다.
[-ejb]	EJB 엔진에 대하여 설정한다.
[-jms]	JMS 엔진에 대하여 설정한다.
[-f,--forceLock]	설정 변경을 강제로 진행하여 반영한다.

- 예제

```

[MASTER]domain1.adminServer>enable-engines adminServer -all
[adminServer]
Change Engine to Enabled: Web EJB JMS

Applying configuration ...
=====
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

```

|                                     Result                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Successfully changed only the JEUS Domain Configuration.                     |
| Restart the server to apply the changes.                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
=====
...

[MASTER]domain1.adminServer>enable-engines server1 -all
[server1]
Change Engine to Enabled: Web EJB JMS

Applying configuration ...
=====
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Result                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Successfully changed only the JEUS Domain Configuration.                     |
| Restart the server to apply the changes.                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
=====
...

[MASTER]domain1.adminServer>enable-engines server2 -web -ejb
[server2]
Change Engine to Enabled: Web EJB

Applying configuration ...
=====
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Result                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Successfully changed only the JEUS Domain Configuration.                     |
| Restart the server to apply the changes.                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
=====
...

```

#### 4.2.3.17. jndi-info

특정 서버의 JNDI 컨텍스트에 바인드되어 있는 객체의 바인딩(이름) 목록을 확인한다.

- alias

jndiinfo, jndilist

- 사용법

```
jndi-info -server <server-name>
          [<context-name>]
```

- 파라미터

파라미터	설명
-server <server-name>	서버의 이름이다.
[<context-name>]	JNDI 컨텍스트의 이름이다.  <context-name>이 컨텍스트이면 해당 컨텍스트에 바인딩되어 있는 모든 객체를 출력하고, 컨텍스트가 아니라 leaf node라면 해당 객체의 상세 정보를 출력한다.

- 예제

- context

```
[MASTER]domain1.adminServer>jndi-info -server server1
The JNDI list on the server1
List of the context /
=====
+-----+-----+-----+
| Name | Value | Local Binding |
+-----+-----+-----+
| JEUSMQ_DLQ | jeus.jms.common.destination.JeusQueue | false |
| mgmt | jeus.jndi.JNSContext | false |
+-----+-----+-----+
=====

[MASTER]domain1.adminServer>jndi-info -server server1 mgmt
The JNDI list on the server1
List of the context /mgmt
=====
+-----+-----+-----+
| Name | Value | Local Binding |
+-----+-----+-----+
| rmbs | jeus.jndi.JNSContext | false |
+-----+-----+-----+
=====
```

- leaf node

```
[MASTER]domain1.adminServer>jndi-info -server server1 mgmt/rmbs/adminServer
The JNDI list on the server1
List of the context /mgmt/rmbs/server1
=====
+-----+-----+-----+
| Name | Value | Local Binding |
+-----+-----+-----+
| mgmt/rmbs/server1 | javax.management.remote.JMXConnector | false |
+-----+-----+-----+

Reference class name : javax.management.remote.JMXConnector
Reference Factory class name : jeus.management.remote.jeusmp.JeusmpConnectorFactory
[0] StringRefAddr : jmxmp, RefAddr type : protocol
[1] StringRefAddr : Administrator, RefAddr type : host
[2] StringRefAddr : 19736, RefAddr type : port
[3] StringRefAddr : /JEUSMP_server1, RefAddr type : urlPath
```

```
[4] StringRefAddr : 60000, RefAddr type : checktmout
[5] StringRefAddr : 61.77.153.160, RefAddr type : ipaddr
end of RefAddr
```

```
=====
```

#### 4.2.3.18. lifecycle-invocation-info

특정 Lifecycle Invocation의 정보를 조회한다.

- alias

invoinfo, lifecycleinvocationinfo, lifeinvoinfo

- 사용법

```
lifecycle-invocation-info -s,--server <server> | -c,--cluster <cluster>
                        [-class <lifecycle-invocation-class>]
                        [-lib,--library]
                        [-invo,--invocation]
```

- 파라미터

파라미터	설명
-s,--server <server>   -c,--cluster <cluster>	조회하고자 하는 Lifecycle Invocation이 위치한 서버 혹은 클러스터를 지정한다.
[-class <lifecycle-invocation-class>]	특정 Lifecycle Invocation의 Class만 조회할 경우 클래스 이름을 지정한다.
[-lib,--library]	Lifecycle Invocation의 Library Ref를 조회한다.
[-invo,--invocation]	Lifecycle Invocation의 Invocation을 조회한다.

- 예제

```
[MASTER]domain1.adminServer>lifecycle-invocation-info -s server1
Target Name           : server1
Target Type           : Server
Lifecycle Invocation Class Name: invocation.ClassName1
Library Ref Information:
=====
+-----+-----+-----+-----+
| Library Name | Specification Version | Implementation Version | Failon Error |
|              | (Exact match)         | (Exact match)          |               |
+-----+-----+-----+-----+
(No data available)
=====

Invocation Information:
=====
```

Invocation ID	Invocation Type	Invocation Argument	Method Name	Method Params
0	BOOT		bootMethod	

#### 4.2.3.19. list-blocked-clients

접속이 차단된 클라이언트 주소들을 조회한다.

- alias

listblocked, listblockedclients

- 사용법

```
list-blocked-clients -server <server-name>
```

- 파라미터

파라미터	설명
-server	접속이 차단된 클라이언트 주소를 조회할 서버의 이름이다.

- 예제

```
[MASTER]domain1.adminServer>list-blocked-clients -server adminServer
=====
+-----+
| Blocked addresses |
+-----+
| 192.168.14.220    |
| 192.168.14.222    |
+-----+
=====
```

#### 4.2.3.20. list-jvm-options

특정 서버의 JVM 설정을 조회한다.

- alias

list-jvmopt, listjvmopt, list-jvm-config, list-jvmcfg, listjvmcfg

- 사용법

```
list-jvm-options <server-name>
```

- 파라미터

파라미터	설명
<server-name>	서버의 이름이다.

- 예제

```
[MASTER]domain1.adminServer>list-jvm-config adminServer
List of JVM Configurations
=====
+-----+-----+
|                               JVM Configs                               |
+-----+-----+
| -Xmx256m -XX:MaxPermSize=128m                                         |
+-----+-----+
=====
```

#### 4.2.3.21. list-lifecycle-invocations

특정 서버 혹은 클러스터의 Lifecycle Invocation을 조회한다.

- alias

listinvo, listlifecycleinvocations, listlifeinvo

- 사용법

```
list-lifecycle-invocations [-s,--server <server> | -c,--cluster <cluster>]
```

- 파라미터

파라미터	설명
[-s,--server <server>   -c,--cluster <cluster>]	서버 혹은 클러스터를 지정할 경우 해당 서버 혹은 클러스터의 Lifecycle Invocation만 조회한다.

- 예제

```
[MASTER]domain1.adminServer>list-lifecycle-invocations
List of Lifecycle invocations
=====
+-----+-----+-----+-----+
| Target | Lifecycle Invocation | Invocation Library | Invocation |
|         | Class               | Ref               |           |
+-----+-----+-----+-----+
```

[Server]adminServer	lifecycle.LifecycleTester		[0]boot
[Server]server1	invocation.ClassName1		[0]bootMethod
[Server]server1	invocation.ClassName2	libName	[0]method1, [1]method2

=====

Use the "lifecycle-invocation-info" command for more information about Lifecycle invocation.

#### 4.2.3.22. list-log-handlers

특정 서버의 특정 로거에 등록된 핸들러들의 정보를 조회한다.

- alias

listloghandlers, listhandlers, list-handlers

- 사용법

```
list-log-handlers -server <server-name>
                  <logger-name>
```

- 파라미터

파라미터	설명
-server <server-name>	서버의 이름이다.
<logger-name>	로거의 이름이다.

- 예제

```
[MASTER]domain1.adminServer>list-log-handlers -server adminServer jeus.deploy
List of Loggers
=====
```

Name	Value
Handler Name	fileHandler
Handler Type	FileHandlerType
Handler Level	FINEST
Filename	jeus.deploy.log
Enable Rotation	true
Rotation Directory	\${SERVER_HOME}/logs
Valid Day	1
Buffer Size	1024
Logging to Append	true

=====

#### 4.2.3.23. list-loggers

특정 서버에 등록된 로거들의 정보를 조회한다.

- alias

listloggers

- 사용법

```
list-loggers <server-name>
          [-logger <logger-name>]
```

- 파라미터

파라미터	설명
<server-name>	서버의 이름이다.
[-logger <logger-name>]	로거의 이름이다.

- 예제

```
[MASTER]domain1.adminServer>list-loggers server1
List of Loggers
=====
+-----+-----+-----+-----+-----+
|  Logger Name  | Level | Use Parent Handlers | Filter | Formatter |
+-----+-----+-----+-----+-----+
| jeus          | INFO  | false               |        |            |
| jeus.security | FINE  | true                |        |            |
+-----+-----+-----+-----+-----+
=====
```

#### 4.2.3.24. list-server-listeners

동적으로 특정 서버에 있는 리스너들의 설정을 조회한다.

- alias

listener, serverlistener

- 사용법

```
list-server-listeners -server <server-name>
```



`[-name <listener-name>]`

- 파라미터

파라미터	설명
<code>-server &lt;server-name&gt;</code>	서버의 이름이다.
<code>[-name &lt;listener-name&gt;]</code>	조회할 리스너의 이름이다.

- 예제

```
[MASTER]domain1.adminServer>list-server-listeners -server adminServer
=====
+-----+-----+-----+
| listener-name | address | port |
+-----+-----+-----+
| base          | 0.0.0.0 | 9736 |
| jms           | 0.0.0.0 | 9741 |
| http-lisetner | not-set | 8088 |
+-----+-----+-----+
=====

[MASTER]domain1.adminServer>list-server-listeners -server adminServer -name base
=====
base
+-----+-----+
| address | 0.0.0.0 |
| port    | 9736    |
| selectors |        |
| dual-selector | false  |
| backlog  | 128     |
| read-timeout | 30000  |
| reserved-threads | 0      |
+-----+-----+
=====
```

#### 4.2.3.25. local-shutdown

현재 접속해 있는 서버를 종료한다.

- alias

localdown, localjeusexit

- 사용법

```
local-shutdown [-to,--shutdowntimeout <timeout> | -f,--force | -g,--graceful]
               [-i]
```

- 파라미터

파라미터	설명
[-f, --force]	Graceful shutdown을 시도하지 않고 서버를 강제로 종료한다.
[-g, --graceful]	Graceful하게 종료할지 여부를 설정한다. 처리 중인 요청이 끝나길 무한히 기다린다.
[-to, --shutdowntimeout <timeout>]	Graceful하게 종료할 때 사용할 타임아웃을 설정한다. 처리 중인 요청을 입력한 시간만큼 완료되길 기다린다. <timeout>의 값으로는 1 이상의 숫자를 입력해야 한다. (단위: second)
[-i]	사용자의 동의를 구하지 않고 바로 서버를 종료한다.

- 예제

local-shutdown 명령어를 사용해 현재 접속한 서버를 종료한다.

```
[MASTER]domain1.adminServer>local-shutdown
Executing this command affects the service. Do you want to continue? (y/n)n
[local-shutdown] has been canceled.
[MASTER]domain1.adminServer>local-shutdown -i
The server [adminServer] has been shut down successfully.
offline>
```

#### 4.2.3.26. local-start-server

On-line 모드에서는 현재 접속해 있는 서버가 STANDBY 상태일 경우 서버를 시작한다. Off-line 모드에서는 SSH를 통하여 특정 MS를 시작한다. 이때 -node, -domain, -server, -u, -p, -masterurl은 필수 항목이다.

- alias

localstartserver, local-boot

- 사용법

아래의 사용법은 On-line 모드일 때 기준이다.

```
local-start-server [-node <node-name>]
                  [-domain <domain-name>]
                  [-server <server-name>]
                  [-u <user-name>]
                  [-p <password>]
                  [-masterurl <masterurl>]
                  [-f, --force]
                  [-s, --standby]
```

- 파라미터

파라미터	설명
[-node <node-name>]	노드의 이름이다.
[-domain <domain-name>]	도메인의 이름이다.
[-server <server-name>]	서버의 이름이다.
[-u <user-name>]	사용자의 이름이다.
[-p <password>]	사용자의 암호이다.
[-masterurl <masterurl>]	Master Server의 주소이다.
[-f, --force]	등록된 애플리케이션이 RUNNING 상태가 아니더라도 강제로 시작한다.
[-s, --standby]	서버를 준비 상태인 STANDBY 상태로 기동한다.

- 예제

```
offline>local-start-server -node node1 -domain domain1 -server server1
-u administrator -p password -masterurl 192.168.34.55:9736
The Server [server1] was successfully started.
```

- 참고

본 명령어는 서버에 접속하지 않은 상태에서도 사용할 수 있다.

#### 4.2.3.27. log-level

특정 서버의 특정 로거와 해당 로거에 등록된 핸들러들의 레벨을 확인한다.

- alias

loglevel

- 사용법

```
log-level -server <server-name>
          <logger-name>
```

- 파라미터

파라미터	설명
-server <server-name>	서버의 이름이다.
<logger-name>	확인할 로거의 이름이다.

- 예제

```
[MASTER]domain1.adminServer>log-level -server server1 jeus
The logger[jeus] information for the server [server1]
Information about the logger[jeus].
=====
Logger Name : jeus
Level : INFO
Use Parent Handlers : false

+-----+-----+-----+
| Handler Name| Handler Type|      Handler Level      |
+-----+-----+-----+
| FileHandler | FileHandler | FINEST                   |
+-----+-----+-----+
=====
```

#### 4.2.3.28. mbean-info

특정 서버에 등록된 MBean들의 리스트를 조회한다.

- alias

mbeaninfo

- 사용법

```
mbean-info -server <server-name>
            [<object-name>]
```

- 파라미터

파라미터	설명
-server <server-name>	서버의 이름이다.
[<object-name>]	Object 이름에 대한 특정 쿼리이다.  특정 Object 이름을 갖는 MBean의 목록을 조회하는 경우 사용한다. 쿼리하는 Object 이름에 대해서는 "JEUS JMX 안내서"를 참고한다.

- 예제

```
[MASTER]domain1.adminServer>mbean-info -server server1
The object names of MBeans on the server [server1].
=====
+-----+-----+-----+
| JEUS:j2eeType=JeusService,jeusType=ServerNodeManagerService,JMXManager=server |
| 1,J2EEServer=server1,name=server1                                           |
| JEUS:j2eeType=J2EEServer,JMXManager=server1,isTargetable=true,J2EEDomain=doma |
| in1,name=server1                                                             |
+-----+-----+-----+
```

```
| JEUS:j2eeType=JMSResource,JMXManager=server1,J2EEServer=server1,name=server1_
| jms
| JEUS:j2eeType=JeusService,jeusType=SecurityDomain,JMXManager=server1,J2EEDoma
| in=domain1,SecurityService=SecurityService,name=SYSTEM_DOMAIN
| JEUS:j2eeType=JeusService,jeusType=JEUSMPConnector,JMXManager=server1,J2EESer
| ver=server1,name=server1
| JEUS:j2eeType=JeusService,jeusType=ThreadPool,JMXManager=server1,J2EEServer=s
| erver1,name=threadpool.System
| JEUS:j2eeType=JeusService,jeusType=WebListener,JMXManager=server1,WebEngine=s
| erver1_servlet,J2EEServer=server1,name=ADMIN-HTTP
| JEUS:j2eeType=JeusService,jeusType=ThreadPool,JMXManager=server1,J2EEServer=s
| erver1,name=EJBTimerService
| JEUS:j2eeType=JDBCResource,JMXManager=server1,J2EEServer=server1,name=server1
| JEUS:j2eeType=JeusService,jeusType=ThreadPool_WEBC,JMXManager=server1,WebEngi
| ne=server1_servlet,J2EEServer=server1,WebListener=ADMIN-HTTP,name=ADMIN-HTTP
| JEUS:j2eeType=JeusService,jeusType=EJBEngine,JMXManager=server1,J2EEServer=se
| rver1,name=server1_ejb
| JEUS:j2eeType=JeusService,jeusType=JNDIResourceService,JMXManager=server1,J2E
| EServer=server1,name=server1
| JEUS:j2eeType=JeusService,jeusType=SecurityService,JMXManager=server1,J2EEDom
| ain=domain1,J2EEServer=server1,name=SecurityService
| JEUS:j2eeType=JeusService,jeusType=WebEngine,JMXManager=server1,J2EEServer=se
| rver1,name=server1_servlet
| JEUS:j2eeType=JeusService,jeusType=SecuritySubject,JMXManager=server1,J2EEDom
| ain=domain1,SecurityDomain=SYSTEM_DOMAIN,name=Subject
| JEUS:j2eeType=JeusService,jeusType=ThreadPool,JMXManager=server1,J2EEServer=s
| erver1,name=server1
| JEUS:j2eeType=JeusService,jeusType=JMSServiceChannel,JMXManager=server1,J2EES
| erver=server1,JMSResource=server1_jms,name=JMSServiceChannel-default
| JEUS:j2eeType=Console,JMXManager=server1,J2EEServer=server1,name=server1
| JEUS:j2eeType=JeusService,jeusType=JDBCResourceService,JMXManager=server1,J2E
| EServer=server1,name=server1
| JEUS:j2eeType=JVM,JMXManager=server1,J2EEServer=server1,name=server1
| JEUS:j2eeType=JTAResource,JMXManager=server1,J2EEServer=server1,name=server1
| JEUS:j2eeType=JeusService,jeusType=JMSEngine,JMXManager=server1,J2EEServer=se
| rver1,name=server1_jms
| JEUS:j2eeType=JeusService,jeusType=JMSDestinationResource,JMXManager=server1,
| J2EEServer=server1,JMSResource=server1_jms,name=JEUSMQ_DLQ
| JEUS:j2eeType=JeusService,jeusType=SecurityPolicy,JMXManager=server1,J2EEDoma
| in=domain1,SecurityDomain=SYSTEM_DOMAIN,name=Policy
+-----+
=====
```

#### 4.2.3.29. memory-info

특정 서버 또는 특정 클러스터에 속한 서버들의 메모리 정보를 확인한다.

- alias

memorystatus, memstatus, memoryinfo, meminfo

- 사용법

```
memory-info [-clusters <cluster-list> | -servers <server-list>]
```

- 파라미터

파라미터	설명
[-clusters <cluster-list>]	클러스터의 이름이다. 클러스터가 한 개 이상인 경우는 콤마(,)로 구분한다.
[-servers <server-list>]	서버의 이름이다. 서버가 한 개 이상인 경우는 콤마(,)로 구분한다.

- 예제

```
[MASTER]domain1.adminServer>memory-info -servers server1
Domain [domain1] Memory Information
Memory Information
=====
+-----+-----+-----+
| Server | Total Amount of Memory | The Current Amount of Memory |
+-----+-----+-----+
| server1|          110362624 |          55153456 |
+-----+-----+-----+
=====
```

#### 4.2.3.30. modify-invocation

특정 Lifecycle Invocation의 Invocation을 수정한다.

- alias

modinvo, modifyinvocation, modifyinvo

- 사용법

```
modify-invocation -id <invocation-id> | -m,--method <invocation-method>
                    [-mv,--rename <invocation-method>]
                    [-type <invocation-type>]
                    [-params <method-params>]
                    [-args <invocation-argument>]
                    -class <class>
                    -s,--server <server> | -c,--cluster <cluster>
                    [-f,--forceLock]
                    [-detail]
```

- 파라미터

파라미터	설명
-id <invocation-id>   -m,--method <invocation-method>	수정할 Invocation을 ID 혹은 메서드 이름으로 지정한다. 만약 동일한 이름의 메서드가 여럿 있을 경우에는 ID로 지정해야 한다.

파라미터	설명
<code>[-mv,--rename &lt;invocation-method&gt;]</code>	Invocation의 메서드 이름을 변경한다.
<code>[-type &lt;invocation-type&gt;]</code>	<p>Invocation의 타입(메서드 호출 시점)을 변경한다.</p> <p>다음 중에 하나를 지정할 수 있다.</p> <ul style="list-style-type: none"> <li>• BOOT</li> <li>• BEFORE_DEPLOY</li> <li>• AFTER_DEPLOY</li> <li>• READY</li> <li>• BEFORE_UNDEPLOY</li> <li>• AFTER_UNDEPLOY</li> </ul> <p>타입에 대한 자세한 내용은 JEUS Server 안내서의 "Lifecycle Invocation 설정"의 Invocation Type을 참고한다.</p>
<code>[-params &lt;method-params&gt;]</code>	<p>메서드 파라미터의 Fully Qualified Class Name을 변경한다. 매개 변수가 여러 개인 경우 공백으로 구분한다.</p> <p>(설정값 예시: <code>-params java.lang.String boolean</code>)</p>
<code>[-args &lt;invocation-argument&gt;]</code>	<p>메서드를 호출할 때 사용하는 Argument를 변경한다. 인자가 여러 개인 경우 공백으로 구분한다. 만약 공백이 포함된 값을 인자로 해야 할 경우 큰따옴표로 묶어 사용하면 된다.</p> <p>(설정값 예시: <code>-args "tmaxsoft jeus" "hello world"</code>)</p>
<code>-class &lt;class&gt;</code>	Invocation이 위치한 Lifecycle Invocation의 클래스 이름을 지정한다.
<code>-s,--server &lt;server&gt;   -c,--cluster &lt;cluster&gt;</code>	Lifecycle Invocation가 위치한 서버 혹은 클러스터의 이름을 지정한다.
<code>[-f,--forceLock]</code>	설정 변경을 강제로 진행하여 반영한다.
<code>[-detail]</code>	설정 변경 결과에 대한 상세한 정보가 출력된다.

#### • 예제

```
[MASTER]domain1.adminServer>modify-invocation -m bootMethod -mv newMethod
-class invocation.ClassName1 -s server1
Successfully performed the MODIFY operation for Invocation [newMethod](Invocation ID = 0), but
all changes were non-dynamic. They will be applied after restarting.
Check the results using "list-lifecycle-invocations or lifecycle-invocation-info".
```

### 4.2.3.31. modify-invocation-library

특정 Lifecycle Invocation에 Library Ref를 수정한다.

- alias

modinvolib, modifyinvocationlibrary, modifyinvolib

- 사용법

```
modify-invocation-library <library>
    [-mv,--rename <invocation-library>]
    [-specVer <version>]
    [-specMatch <true | false>]
    [-implVer <version>]
    [-implMatch <true | false>]
    [-fail,--failOnError <true | false>]
    -class <class>
    -s,--server <server> | -c,--cluster <cluster>
    [-f,--forceLock]
    [-detail]
```

- 파라미터

파라미터	설명
<library>	수정할 Library Ref의 라이브러리 이름이다.
[-mv,--rename <invocation-library>]	Library Ref의 라이브러리 이름을 변경한다.
[-specVer <version>]	해당 라이브러리의 Specification Version을 수정한다.
[-specMatch <true   false>]	Specification Version이 정확히 일치해야하는지 여부 설정을 수정한다.
[-implVer <version>]	해당 라이브러리의 Implementation Version을 수정한다.
[-implMatch <true   false>]	Implementation Version이 정확히 일치해야하는지 여부 설정을 수정한다.
[-fail,--failOnError <true   false>]	해당 Shared Library를 찾지 못할 경우 Deploy를 실패할지 여부 설정을 수정한다.
-class <class>	Library Ref가 위치한 Lifecycle Invocation의 클래스 이름을 지정한다.
-s,--server <server>   -c,--cluster <cluster>	Lifecycle Invocation가 위치한 서버 혹은 클러스터의 이름을 지정한다.
[-f,--forceLock]	설정 변경을 강제로 진행하여 반영한다.
[-detail]	설정 변경 결과에 대한 상세한 정보가 출력된다.

- 예제



```
[MASTER]domain1.adminServer>modify-invocation-library libName -mv newLibName
-class invo.invoTestClass2 -s server1
Successfully performed the MODIFY operation for Invocation Library [newLibName], but all changes
were non-dynamic.
They will be applied after restarting.
Check the results using "list-lifecycle-invocations or lifecycle-invocation-info".
```

#### 4.2.3.32. modify-jvm-option

특정 서버의 JVM 설정을 수정한다.

- alias

modify-jvmopt, modifyjvmopt, modify-jvm-config, modify-jvmcfg

- 사용법

```
modify-jvm-option -server <server-name>
                  [-old,--oldOption <jvm-option>]
                  [-new,--newOption <jvm-option>]
                  [-f,--forceLock]
                  [-detail]
```

- 파라미터

파라미터	설명
-server <server-name>	서버의 이름이다.
[-old,--oldOption <jvm-option>]	수정하기 전 JVM 설정이다. 현재 JVM 설정과 같다.
[-new,--newOption <jvm-option>]	수정 후 바뀔 JVM 설정이다.
[-f,--forceLock]	설정 변경을 강제로 진행하여 반영한다.
[-detail]	설정 변경 결과에 대한 상세한 정보가 출력된다.

- 예제

```
[MASTER]domain1.adminServer>modify-jvm-option -server server1
-old "-Xmx256m -XX:MaxPermSize=128m" -new "-Xmx512m -XX:MaxPermSize=128m"
Successfully performed the MODIFY operation for JVM configuration for the server
(server1)., but all changes were non-dynamic. They will be applied after restarting.
Check the results using "list-jvm-options or modify-jvm-option"
```

#### 4.2.3.33. modify-lifecycle-invocation

특정 서버 혹은 클러스터의 Lifecycle Invocation을 수정한다.

- alias

modlifeinvo, modifylifecyleinvocation, modifylifeinvo

- 사용법

```
modify-lifecycle-invocation <class>
    [-mv,--rename <lifecycle-invocation-class>]
    -s,--server <server> | -c,--cluster <cluster>
    [-f,--forceLock]
    [-detail]
```

- 파라미터

파라미터	설명
<class>	수정할 Lifecycle Invocation의 클래스 이름이다.
[-mv,--rename <lifecycle-invocation-class>]	Lifecycle Invocation의 클래스 이름을 변경한다.
-s,--server <server>   -c,--cluster <cluster>	Lifecycle Invocation이 위치한 서버 혹은 클러스터의 이름을 지정한다.
[-f,--forceLock]	설정 변경을 강제로 진행하여 반영한다.
[-detail]	설정 변경 결과에 대한 상세한 정보가 출력된다.

- 예제

```
[MASTER]domain1.adminServer>modify-lifecycle-invocation sampleClass1 -mv className3 -s server1
Successfully performed the ADD operation for Lifecycle Invocation Class [className3], but all
changes were non-dynamic.
They will be applied after restarting.
Check the results using "list-lifecycle-invocations".
```

#### 4.2.3.34. modify-listener

특정 서버에 있는 리스너의 설정을 수정한다.

- alias

modifylistener, editlistener

- 사용법

```
modify-listener -server <server-name>
    -name <listener-name>
    [-addr <address>]
    [-port <port>]
```

```

[-selectors <selectors>]
[-dual]
[-backlog <backlog>]
[-timeout <read-timeout>]
[-keepaliveTimeout <keepalive-timeout>]
[-rt,--reservedthreads <reserved-threads>]
[-f,--forceLock]

```

#### • 파라미터

파라미터	설명
-server <server-name>	서버의 이름이다.
-name <listener-name>	수정할 리스너의 이름이다.
[-addr <address>]	수정할 리스너의 바뀔 주소이다.
[-port <port>]	수정할 리스너의 바뀔 포트 번호이다. 1 ~ 65535 사이의 값을 설정해야 하며, 다른 리스너의 포트 번호와 겹치지 않아야 한다.
[-selectors <selectors>]	사용할 Selector의 개수이다.
[-dual]	read/write 작업에 각각 별개의 Selector를 사용한다.
[-backlog <backlog>]	수정할 리스너의 바뀔 backlog이다.
[-timeout <read-timeout>]	<b>[동적 변경]</b> 수정할 리스너의 바뀔 read 타임아웃이다.
[-keepaliveTimeout <keepalive-timeout>]	사용하지 않는 소켓을 닫을 때까지의 시간을 설정한다.
[-rt,--reservedthreads <reserved-threads>]	수정할 리스너가 사용할 Reserved Thread의 바뀔 개수이다.
[-f,--forceLock]	수정을 위한 lock을 강제로 취한다.

#### • 예제

```

[MASTER]domain1.adminServer>modify-listener -server adminServer -name base -dual -b acklog 250
Executed successfully, but some configurations were not applied dynamically. It
might be necessary to restart the server.
Check the result using 'list-server-listeners -server adminServer -name base.

```

```

[MASTER]domain1.adminServer>list-server-listeners -server adminServer -name base
=====

```

```

base

```

```

+-----+-----+
| address                | 0.0.0.0 |
| port                   | 9736    |
| selectors               | 1       |
| dual-selector           | true    |
| backlog                | 250     |
| read-timeout            | 30000   |
| keepalive-timeout       | not-set |
| reserved-threads        | 0       |

```

+-----+-----+  
=====

#### 4.2.3.35. modify-log-handler

특정 서버의 특정 로거에 등록된 특정 핸들러를 수정한다.

- alias

modifyloghandler, modify-handler, modifyhandler

- 사용법

```
modify-log-handler -server <server-name>
                   -logger <logger-name>
                   <handler-name>
                   [-level <handler-level>]
                   [-filter <filter-class>]
                   [-encoding <encoding>]
                   [-filename <file-name>]
                   [-enable <true | false>]
                   [-count <rotation-count>]
                   [-dir <rotation-dir>]
                   [-buffer <buffer-size>]
                   [-append <true | false>]
                   [-hour <hour> | -day <day> | -size <size>]
                   [-f,--forceLock]
                   [-detail]
                   [-permission <permission>]
                   [-chown <chown>]
```

- 파라미터

파라미터	설명
-server <server-name>	서버의 이름이다.
-logger <logger-name>	로거의 이름이다.
<handler-name>	변경할 핸들러의 이름이다.
[-level <handler-level>]	<b>[동적 변경]</b> 핸들러의 로그 레벨이다. (기본값: FINEST)
[-filter <filter-class>]	로그 메시지의 필터 클래스 이름이다.
[-encoding <encoding>]	로그 메시지의 인코딩 값이다.
[-filename <file-name>]	로그 메시지를 기록할 파일 이름이다. (기본값: loggerName.log. jeus logger의 경우는 JeusServer.log)
[-enable <true   false>]	파일 로테이션을 가능하게 할지 여부를 설정한다. (기본값: true)

파라미터	설명
<code>[-count &lt;rotation-count&gt;]</code>	백업하는 로그 파일의 최대 개수이다.
<code>[-dir &lt;rotation-dir&gt;]</code>	백업하는 로그 파일이 위치하는 디렉터리 경로이다.
<code>[-buffer &lt;buffer-size&gt;]</code>	로그 메시지를 파일에 기록할 때의 버퍼 사이즈이다.
<code>[-append &lt;true   false&gt;]</code>	재부팅할 때 이전 파일에 계속 로그를 남길지 여부를 설정한다. (true   false, 기본값: true)
<code>[-hour &lt;hour&gt;]</code>	입력된 시간마다 새로운 로그 파일을 작성한다.
<code>[-day &lt;day&gt;]</code>	입력된 날마다 새로운 로그 파일을 작성한다.
<code>[-size &lt;size&gt;]</code>	입력된 사이즈에 따라 새로운 로그 파일을 작성한다.
<code>[-f,--forceLock]</code>	설정 변경을 강제로 진행하여 반영한다.
<code>[-detail]</code>	동적 변경 결과에 대한 상세한 정보가 출력된다.
<code>[-permission &lt;permission&gt;]</code>	입력된 퍼미션으로 로그파일을 설정한다. (설정값 예시: rwxrwxrwx)
<code>[-chown &lt;chown&gt;]</code>	입력된 소유자로 로그파일을 설정한다. owner와 group은 콤마(,)로 구분한다 (설정값 예시: owner-id,group-id)

- 예제

```
[MASTER]domain1.adminServer>modify-log-handler -server server1 -logger jeus newHandler -level
CONFIG
Successfully performed the MODIFY operation for The handler(FileHandler) for the
  logger(jeus) in server (server1).
Check the results using "modify-log-handler"
```

#### 4.2.3.36. modify-logger

특정 서버에 등록된 특정 로거를 수정한다.

- alias

modifylogger

- 사용법

```
modify-logger -server <server-name> <logger-name>
  [-level <log-level>]
  [-useParentHandlers <true | false>]
  [-filter <filter-class>]
  [-formatter <formatter-class>]
  [-f,--forceLock]
  [-detail]
```

- 파라미터

파라미터	설명
-server <server-name>	서버의 이름이다.
<logger-name>	로거의 이름이다.
[-level <log-level>]	<b>[동적 변경]</b> 로거의 로그 레벨이다.
[-useParentHandlers <true   false>]	<b>[동적 변경]</b> 상위 로거의 핸들러를 사용할지 여부이다. (true   false)
[-filter <filter-class>]	로거에 적용할 필터의 클래스 이름이다.
[-formatter <formatter-class>]	로거에 적용할 Formatter의 클래스 이름이다.
[-f,--forceLock]	설정 변경을 강제로 진행하여 반영한다.
[-detail]	동적 변경 결과에 대한 상세한 정보가 출력된다.

- 예제

```
[MASTER]domain1.adminServer>modify-logger -server server1 jeus.security -level FINEST
-useParentHandlers false
Successfully performed the MODIFY operation for The logger[jeus.security] inform
ation for the server [server1].
Check the results using "modify-logger"
```

#### 4.2.3.37. remove-custom-resource-from-servers

동적으로 서버에 등록된 custom resource를 삭제한다.

- alias

remove-cr-from-servers

- 사용법

```
remove-custom-resource-from-servers <export-name>
                                -servers <server-list>
                                [-f,--forceLock]
                                [-detail]
```

- 파라미터

파라미터	설명
<export-name>	서버에서 제거할 custom resource의 JNDI 바인드 이름이다.
-servers <server-list>	custom resource를 제거할 server list를 설정한다.
[-f,--forceLock]	설정 변경을 강제로 진행하여 반영한다.
[-detail]	동적 변경 결과에 대한 상세한 정보가 출력된다.

- 예제

```
[MASTER]domain1.adminServer>remove-custom-resource-from-servers custom/dog -servers server1
Successfully performed the REMOVE operation for A custom resource.
Check the results using "list-custom-resources"
```

#### 4.2.3.38. remove-data-sources-from-server

동적으로 서버에 등록된 데이터소스들을 제거한다.

- alias

rmdsfromsvr

- 사용법

```
remove-data-sources-from-server -server <server-name>
                                [-ids <data-source-id-list> | -all]
                                [-f,--forceLock]
                                [-detail]
```

- 파라미터

파라미터	설명
-server <server-name>	데이터소스들 제거할 서버의 이름이다.
[-ids <data-source-id-list>]	서버로부터 제거할 데이터소스들의 ID 목록이다. 한 개 이상인 경우 콤마(,)로 구분한다.
[-all]	서버에 등록된 모든 데이터소스를 제거한다.
[-f,--forceLock]	설정 변경을 강제로 진행하여 반영한다.
[-detail]	동적 변경 결과에 대한 상세한 정보가 출력된다.

- 예제

```
[MASTER]domain1.adminServer>remove-data-sources-from-server -server server1 -ids data-source1
Successfully performed the REMOVE operation for data sources from the server [server1].
Check the results using "remove-data-sources-from-server -server server1"
```

#### 4.2.3.39. remove-external-resource-from-servers

동적으로 서버에 등록된 external resource를 삭제한다.

- alias

## remove-er-from-servers

### • 사용법

```
remove-external-resource-from-servers <export-name>
                                     -servers <server-list>
                                     [-f,--forceLock]
                                     [-detail]
```

### • 파라미터

파라미터	설명
<export-name>	서버에서 제거할 external resource의 이름이다.
-servers <server-list>	external resource를 제거할 server list를 설정한다.
[-f,--forceLock]	설정 변경을 강제로 진행하여 반영한다.
[-detail]	동적 변경 결과에 대한 상세한 정보가 출력된다.

### • 예제

```
[MASTER]domain1.adminServer>remove-external-resource-from-servers test/ext -servers server1
Successfully performed the REMOVE operation for A external resource.
Check the results using "list-external-resources"
```

## 4.2.3.40. remove-invocation

특정 Lifecycle Invocation의 Invocation을 제거한다.

### • alias

rminvo, removeinvocation, removeinvo

### • 사용법

```
remove-invocation -id <invocation-id> | -m,--method <invocation-method>
                  -class <class>
                  -s,--server <server> | -c,--cluster <cluster>
                  [-f,--forceLock]
                  [-detail]
```

### • 파라미터

파라미터	설명
-id <invocation-id>   -m,--method <invocation-method>	제거할 Invocation을 ID 혹은 메서드 이름으로 지정한다. 만약 동일한 이름의 메서드가 여러 개인 경우 ID로 지정해야 한다.



파라미터	설명
-class <class>	Invocation이 위치한 Lifecycle Invocation의 클래스 이름을 지정한다.
-s,--server <server>   -c,--cluster <cluster>	Lifecycle Invocation가 위치한 서버 혹은 클러스터의 이름을 지정한다.
[-f,--forceLock]	설정 변경을 강제로 진행하여 반영한다.
[-detail]	설정 변경 결과에 대한 상세한 정보가 출력된다.

- 예제

```
[MASTER]domain1.adminServer>remove-invocation -m bootMethod -class invocation.ClassName1 -s
server1
Successfully performed the REMOVE operation for Invocation [bootMethod](Invocation ID = 0), but
all changes were non-dynamic.
They will be applied after restarting.
Check the results using "list-lifecycle-invocations or lifecycle-invocation-info".
```

#### 4.2.3.41. remove-invocation-library

특정 Lifecycle Invocation에 Library Ref를 제거한다.

- alias

rminvolib, removeinvocationlibrary, removeinvolib

- 사용법

```
remove-invocation-library <library>
    -class <class>
    -s,--server <server> | -c,--cluster <cluster>
    [-f,--forceLock]
    [-detail]
```

- 파라미터

파라미터	설명
<library>	제거할 Library Ref의 라이브러리 이름이다.
-class <class>	Library Ref가 위치한 Lifecycle Invocation의 클래스 이름을 지정한다.
-s,--server <server>   -c,--cluster <cluster>	Lifecycle Invocation가 위치한 서버 혹은 클러스터의 이름을 지정한다.
[-f,--forceLock]	설정 변경을 강제로 진행하여 반영한다.
[-detail]	설정 변경 결과에 대한 상세한 정보가 출력된다.

- 예제

```
[MASTER]domain1.adminServer>remove-invocation-library newLibName -class invo.invoTestClass2 -s
server1
Successfully performed the REMOVE operation for Invocation Library [newLibName], but all changes
were non-dynamic.
They will be applied after restarting.
Check the results using "list-lifecycle-invocations or lifecycle-invocation-info".
```

#### 4.2.3.42. remove-jvm-option

특정 서버의 JVM 설정을 제거한다.

- alias

remove-jvmopt, rm-jvmopt, rmjvmopt, remove-jvm-config, remove-jvmcfg, rm-jvmcfg, rmjvmcfg

- 사용법

```
remove-jvm-option -server <server-name>
                    [-opt,--jvmOption <jvm-option>]
                    [-f,--forceLock]
                    [-detail]
```

- 파라미터

파라미터	설명
-server <server-name>	서버의 이름이다.
[-opt,--jvmOption <jvm-option>]	서버로부터 제거할 JVM 설정이다.
[-f,--forceLock]	설정 변경을 강제로 진행하여 반영한다.
[-detail]	설정 변경 결과에 대한 상세한 정보가 출력된다.

- 예제

```
[MASTER]domain1.adminServer>remove-jvm-option -server server1 -opt "-Xmx256m -XX:Ma
xPermSize=128m"
Successfully performed the REMOVE operation for JVM configuration for the server
(server1)., but all changes were non-dynamic. They will be applied after restarting.
Check the results using "list-jvm-options or remove-jvm-option"
```

#### 4.2.3.43. remove-lifecycle-invocation

특정 서버 혹은 클러스터의 Lifecycle Invocation을 제거한다.

- alias

rmlifeinvo, removelifecycleinvocation, removelifeinvo

- 사용법

```
remove-lifecycle-invocation <class>
                        -s,--server <server> | -c,--cluster <cluster>
                        [-f,--forceLock]
                        [-detail]
```

- 파라미터

파라미터	설명
<class>	제거할 Lifecycle Invocation의 클래스 이름이다.
-s,--server <server>   -c,--cluster <cluster>	Lifecycle Invocation이 위치한 서버 혹은 클러스터의 이름을 지정한다.
[-f,--forceLock]	설정 변경을 강제로 진행하여 반영한다.
[-detail]	설정 변경 결과에 대한 상세한 정보가 출력된다.

- 예제

```
[MASTER]domain1.adminServer>remove-lifecycle-invocation className3 -s server1
Successfully performed the REMOVE operation for Lifecycle Invocation Class [className3], but all
changes were non-dynamic.
They will be applied after restarting.
Check the results using "list-lifecycle-invocations".
```

#### 4.2.3.44. remove-listener

특정 서버에 있는 리스너를 삭제한다. base 리스너나 다른 설정에서 참조하고 있는 리스너는 삭제할 수 없다.

- alias

removelistener, deletelistener

- 사용법

```
remove-listener -server <server-name>
                -name <listener-name>
                [-f,--forceLock]
```

- 파라미터

파라미터	설명
-server <server-name>	서버의 이름이다.

파라미터	설명
-name <listener-name>	삭제할 리스너의 이름이다.
[-f,--forceLock]	수정을 위한 lock을 강제로 취한다.

- 예제

```
[MASTER]domain1.adminServer>list-server-listeners -server adminServer
=====
+-----+-----+-----+
| listener-name | address | port |
+-----+-----+-----+
| base          | 0.0.0.0 | 9736 |
| jms           | 0.0.0.0 | 9741 |
| http-lisetner | not-set | 8088 |
| Test          | not-set | 10000 |
+-----+-----+-----+
=====

[MASTER]domain1.adminServer>remove-listener -server adminServer -name Test
Executed Successfully, but some configurations were not applied dynamically. It
might be necessary to restart the server.
Check the result using 'list-server-listeners -server adminServer.
[MASTER]domain1.adminServer>list-server-listeners -server adminServer
=====
+-----+-----+-----+
| listener-name | address | port |
+-----+-----+-----+
| base          | 0.0.0.0 | 9736 |
| jms           | 0.0.0.0 | 9741 |
| http-lisetner | not-set | 8088 |
+-----+-----+-----+
=====
```

#### 4.2.3.45. remove-log-handler

동적으로 특정 서버의 특정 로그에 등록된 특정 핸들러를 삭제한다.

- alias

removeloghandler, removehandler, remove-handler

- 사용법

```
remove-log-handler -server <server-name>
                  -logger <logger-name>
                  [<handler-name>]
                  [-f,--forceLock]
                  [-detail]
```

- 파라미터

파라미터	설명
-server <server-name>	서버의 이름이다.
-logger <logger-name>	핸들러를 제거할 로거의 이름이다.
[<handler-name>]	제거할 핸들러의 이름이다.
[-f,--forceLock]	설정 변경을 강제로 진행하여 반영한다.
[-detail]	동적 변경 결과에 대한 상세한 정보가 출력된다.

- 예제

```
[MASTER]domain1.adminServer>remove-log-handler -server server1 -logger jeus newHandler
Successfully performed the REMOVE operation for The handler for the logger(jeus)
on the server(server1)..
Check the results using "list-log-handlers or remove-log-handler"
```

#### 4.2.3.46. remove-logger

동적으로 특정 서버에 등록된 특정 로거를 삭제한다.

- alias

removelogger

- 사용법

```
remove-logger -server <server-name>
                [<logger-name>]
                [-f,--forceLock]
                [-detail]
```

- 파라미터

파라미터	설명
-server <server-name>	서버의 이름이다.
[<logger-name>]	제거할 로거의 이름이다.
[-f,--forceLock]	설정 변경을 강제로 진행하여 반영한다.
[-detail]	동적 변경 결과에 대한 상세한 정보가 출력된다.

- 예제

```
[MASTER]domain1.adminServer>remove-logger -server server1 jeus.security
Successfully performed the REMOVE operation for The logger for the
server(server1)..
```

Check the results using "list-loggers or remove-logger"

#### 4.2.3.47. remove-replicate-group

동적으로 특정 서버 또는 서버 템플릿에 지정한 Replicate Group을 삭제한다.

- alias

rmrg

- 사용법

```
remove-replicate-group [-template <name>]
                        [-server <name>]
                        [-f,--forceLock]
                        [-detail]
```

- 파라미터

파라미터	설명
[-template <name>]	Replicate Group을 삭제할 서버 템플릿의 이름이다. 서버 옵션과 함께 사용할 수 없다.
[-server <name>]	Replicate Group을 삭제할 서버의 이름이다. 서버 템플릿 옵션과 함께 사용할 수 없다.
[-f,--forceLock]	설정 변경을 강제로 진행하여 반영한다.
[-detail]	동적 변경 결과에 대한 상세한 정보가 출력된다.

- 예제

```
[MASTER]domain1.adminServer>remove-replicate-group -server server1
Successfully performed the REMOVE operation for Replicate Group Name.
Check the results using "show-replicate-group".
```

#### 4.2.3.48. resume-server

일시 정지된 특정 서버 또는 특정 클러스터를 다시 서비스되도록 한다.

- alias

resume

- 사용법

```
resume-server [-clusters <cluster-list> | -servers <server-list>]
```

- 파라미터

파라미터	설명
[-clusters <cluster-list>]	클러스터의 이름이다. 클러스터가 한 개 이상인 경우는 콤마(,)로 구분한다.
[-servers <server-list>]	서버의 이름이다. 서버가 한 개 이상인 경우는 콤마(,)로 구분한다.

- 예제

```
[MASTER]domain1.adminServer>resume-server -servers server1  
Successfully resumed the servers.
```

#### 4.2.3.49. run-garbage-collection

특정 서버 또는 특정 클러스터에 속한 서버들에 Full Garbage Collection를 수행한다.

- alias

rungc, gc

- 사용법

```
run-garbage-collection [-clusters <cluster-list> | -servers <server-list>]
```

- 파라미터

파라미터	설명
[-clusters <cluster-list>]	클러스터의 이름이다. 클러스터가 한 개 이상인 경우는 콤마(,)로 구분한다.
[-servers <server-list>]	서버의 이름이다. 서버가 한 개 이상인 경우는 콤마(,)로 구분한다.

- 예제

```
[MASTER]domain1.adminServer>run-garbage-collection -servers server1  
=====
```

+-----+	+-----+	+-----+
Server	Result	
+-----+	+-----+	+-----+
server1	Succeeded to GC.	
+-----+	+-----+	+-----+

```
=====
```

#### 4.2.3.50. server-info

서버의 현재 상태를 확인한다.

- alias

svrinfo, serverinfo, si

- 사용법

```
server-info [-server <server-name>]
            [-state | -pid]
            [-g, --group <group-name> | -c, --cluster <cluster-name> |
            -n, --node <node-name>]
```

- 파라미터

파라미터	설명
[-server <server-name>]	특정 서버에 대한 정보를 출력한다.
[-state]	특정 서버를 함께 지정하였을 경우 해당 서버의 state만 출력한다.
[-pid]	특정 서버를 함께 지정하였을 경우 해당 서버의 pid만 출력한다.
[-g, --group <group-name>]	해당 그룹으로 분류된 서버에 대한 정보를 출력한다.
[-c, --cluster <cluster-name>]	해당 클러스터로 분류된 서버에 대한 정보를 출력한다.
[-n, --node <node-name>]	해당 노드로 분류된 서버에 대한 정보를 출력한다.

- 예제

```
[MASTER]domain1.adminServer>server-info
Information about Domain (domain1)
=====
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Server | Status | Node | PID | Clus | Latest | Need to | Listen Ports | Running |
|         |         | Name |     | ter  | StartTime/Shut | Restart |              | Engines |
|         |         |      |     |     | downTime      |        |              |          |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| adminSe | RUNNING | N/A | 12154 | N/A | 2016-08-26 | false | base-0.0.0.0 | jms, |
| rver (*) | (00:21:46 |    |    |    | (금) 오후 |    | :9736 | ejb, web |
|         | )        |    |    |    | 02:29:03 KST |    | http-server- |      |
|         |         |    |    |    |             |    | 0.0.0.0:8088 |      |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| server1 | RUNNING | node1 | 12171 | clus | 2016-08-26 | false | base-192.167 | jms, |
|         | (00:04:03 |    |    | ter1 | (금) 오후 |    | .0.124:9836 | ejb, web |
|         | )        |    |    |     | 02:36:22 KST |    |             |      |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| server2 | RUNNING | node1 | 12172 | clus | 2016-08-26 | false | base-192.167 | jms |
|         | (00:04:03 |    |    | ter1 | (금) 오후 |    | .0.124:9936 | ejb, web |
|         | )        |    |    |     | 02:36:22 KST |    |             |      |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| ds1 (d) | SHUTDOWN | node1 | N/A | temp | N/A | N/A | N/A | N/A |
```



				late1					
ds2 (d)	SHUTDOWN	node1	N/A	temp	N/A		N/A	N/A	N/A
				late1					



1. JEUS Master Server의 경우 서버 이름 옆에 애스터리스크(\*) 표기가 붙는다.
2. dynamic 서버의 경우 이름 옆에 (d) 표기가 붙는다.

#### 4.2.3.51. server-log

특정 서버의 log 정보를 확인한다.

- alias

serverlog

- 사용법

```
server-log -server <server-name>
          [-basetime <base-time>]
          [-endtime <end-time>]
          [-line <number-of-line>]
```

- 파라미터

파라미터	설명
-server <server-name>	log를 가져올 서버를 지정한다.
[-basetime <base-time>]	log의 시작 위치에 해당하는 지점을 시간으로 지정한다. (형식: "yyyy.MM.dd HH:mm:ss" 또는 "HH:mm:ss")
[-endtime <end-time>]	log의 끝 위치에 해당하는 지점을 시간으로 지정한다. (형식: "yyyy.MM.dd HH:mm:ss" 또는 "HH:mm:ss")
[-line <number-of-line>]	log의 시작 위치에 해당하는 지점을 line으로 지정한다.

- 예제

```
[MASTER]domain1.adminServer>server-log -server adminServer -line 10
[2016.08.26 20:00:39][2] [adminServer-33] [NodeManager-0062] Exceed to count to
connecting to the node manager. Verify that the node manager is functioning normally.
[2016.08.26 20:00:59][2] [adminServer-33] [Console-0010] Attempting to execute
```

```
a command: server-log from heejin@192.168.15.59.
[2016.08.26 20:00:59][2] [adminServer-33] [Console-0012] An exception occurred
while processing a command: server-log from heejin@192.168.15.59.
[2016.08.26 20:01:45][2] [adminServer-76] [Console-0010] Attempting to execute
a command: server-log from heejin@192.168.15.59.
[2016.08.26 20:01:45][2] [adminServer-76] [Console-0012] An exception occurred
while processing a command: server-log from heejin@192.168.15.59.
[2016.08.26 20:02:26][2] [adminServer-36] [Console-0010] Attempting to execute
a command: server-log from heejin@192.168.15.59.
[2016.08.26 20:02:26][2] [adminServer-36] [Console-0014] The command: server-log
from heejin@192.168.15.59 is executed successfully.
[2016.08.26 20:02:40][2] [adminServer-73] [Console-0010] Attempting to execute
a command: server-log from heejin@192.168.15.59.
[2016.08.26 20:02:40][2] [adminServer-73] [Console-0014] The command: server-log
from heejin@192.168.15.59 is executed successfully.
[2016.08.26 20:08:16][2] [adminServer-76] [Console-0010] Attempting to execute
a command: server-log from heejin@192.168.15.59.
Successfully got the log messages from the server(adminServer).
[MASTER]domain1.adminServer>
```

#### 4.2.3.52. set-replicate-group

동적으로 특정 서버 또는 서버 템플릿에 Replicate Group을 지정한다.

- alias

setrg

- 사용법

```
set-replicate-group <ReplicateGroupName>
    [-template <name>]
    [-server <name>]
    [-f,--forceLock]
    [-detail]
```

- 파라미터

파라미터	설명
<ReplicateGroupName>	Replicate Group의 이름이다.
[-template <name>]	Replicate Group을 지정할 서버 템플릿의 이름이다. 서버 옵션과 함께 사용할 수 없다.
[-server <name>]	Replicate Group을 지정할 서버의 이름이다. 서버 템플릿 옵션과 함께 사용할 수 없다.
[-f,--forceLock]	설정 변경을 강제로 진행하여 반영한다.
[-detail]	동적 변경 결과에 대한 상세한 정보가 출력된다.

- 예제

```
[MASTER]domain1.adminServer>set-replicate-group RepGrp -server server1
Successfully performed the ADD operation for Replicate Group Name, but all changes were non-
dynamic. They will be applied after restarting.
Check the results using "show-replicate-group".
```

#### 4.2.3.53. show-replicate-group

서버 또는 서버 템플릿의 Replicate Group을 확인한다.

- alias

showrg

- 사용법

```
show-replicate-group [-template <name>]
                    [-server <name>]
```

- 파라미터

파라미터	설명
[-template <name>]	Replicate Group을 조회할 서버 템플릿의 이름이다. 서버 옵션과 함께 사용할 수 없다.
[-server <name>]	Replicate Group을 조회할 서버의 이름이다. 서버 템플릿 옵션과 함께 사용할 수 없다.

- 예제

```
[MASTER]domain1.adminServer>show-replicate-group
Replicate Group Name in Server / Server Template
=====
+-----+-----+-----+
| Type | Name | Replicate Group Name |
+-----+-----+-----+
| Server | adminServer | |
| Server | server1 | RepGrp |
| Server | server2 | |
| Template | tm1 | Test |
+-----+-----+-----+
=====
[MASTER]domain1.adminServer>show-replicate-group -server ms1
Replicate Group Name in Server / Server Template
=====
+-----+-----+-----+
| Type | Name | Replicate Group Name |
+-----+-----+-----+
| Server | server1 | RepGrp |
+-----+-----+-----+
=====
```

```
[MASTER]domain1.adminServer>show-replicate-group -template tm1
Replicate Group Name in Server / Server Template
=====
+-----+-----+-----+
|  Type  | Name |      Replicate Group Name      |
+-----+-----+-----+
| Template | tm1  | Test                            |
+-----+-----+-----+
=====
```

#### 4.2.3.54. show-current-scf-view

현재 SCF View를 모니터링한다.

- alias

scfview, showscf

- 사용법

```
show-current-scf-view [-dump]
```

- 파라미터

파라미터	설명
[-dump]	옵션을 지정할 경우 Server log를 dump한다.

- 예제

```
[MASTER]domain1.adminServer>show-current-scf-view
=====
Current SCF Views

+-----+-----+-----+
|      Group Leader      |
+-----+-----+-----+
| adminServer            |
+-----+-----+-----+
=====

=====
Synchronized View

+-----+-----+-----+-----+-----+-----+
| Server | Address | PID | Started Time | State | Groups |
| Name   |         |     |              |      |        |
+-----+-----+-----+-----+-----+-----+
| adminSer | 192.168.1.1:9 | 29660 | Tue Aug 30 02:43:15 | ALIVE | [domain1] |
| ver      | 736         |      | KST 2022       |      |          |
+-----+-----+-----+-----+-----+-----+
| server1  | 192.168.1.5:1 | 20500 | Mon Aug 29 10:21:27 | ALIVE | [domain1] |
+-----+-----+-----+-----+-----+-----+
```

```

|          |9736          |          |KST 2022          |          |          |
+-----+-----+-----+-----+-----+-----+
=====

Individual View

+-----+-----+-----+-----+-----+-----+
| Server | Address | PID | Started Time | State | Groups |
| Name   |         |     |              |       |         |
+-----+-----+-----+-----+-----+-----+
| adminSer | 192.168.1.1:9 | 29660 | Tue Aug 30 02:43:15 | ALIVE | [domain1] |
| ver      | 736         |      | KST 2022         |       |         |
+-----+-----+-----+-----+-----+-----+
| server1 | 192.168.1.5:1 | 20500 | Mon Aug 29 10:21:27 | ALIVE | [domain1] |
|          | 9736         |      | KST 2022         |       |         |
+-----+-----+-----+-----+-----+-----+
=====

[MASTER]domain1.adminServer>show-current-scf-view -dump
=====

Current SCF Views

+-----+-----+-----+-----+-----+-----+
|                                     |
| Group Leader                       |
+-----+-----+-----+-----+-----+-----+
| adminServer                       |
+-----+-----+-----+-----+-----+-----+
=====

Synchronized View

+-----+-----+-----+-----+-----+-----+
| Server | Address | PID | Started Time | State | Groups |
| Name   |         |     |              |       |         |
+-----+-----+-----+-----+-----+-----+
| adminSer | 192.168.1.1:9 | 29660 | Tue Aug 30 02:43:15 | ALIVE | [domain1] |
| ver      | 736         |      | KST 2022         |       |         |
+-----+-----+-----+-----+-----+-----+
| server1 | 192.168.1.5:1 | 20500 | Mon Aug 29 10:21:27 | ALIVE | [domain1] |
|          | 9736         |      | KST 2022         |       |         |
+-----+-----+-----+-----+-----+-----+
=====

Individual View

+-----+-----+-----+-----+-----+-----+
| Server | Address | PID | Started Time | State | Groups |
| Name   |         |     |              |       |         |
+-----+-----+-----+-----+-----+-----+
| adminSer | 192.168.1.1:9 | 29660 | Tue Aug 30 02:43:15 | ALIVE | [domain1] |
| ver      | 736         |      | KST 2022         |       |         |
+-----+-----+-----+-----+-----+-----+
| server1 | 192.168.1.5:1 | 20500 | Mon Aug 29 10:21:27 | ALIVE | [domain1] |
|          | 9736         |      | KST 2022         |       |         |
+-----+-----+-----+-----+-----+-----+
=====

Successfully dumped to server log.

```

```
[MASTER]domain1.adminServer>
```

#### 4.2.3.55. start-cluster

특정 클러스터에 속한 MS들을 시작한다. SHUTDOWN 상태 또는 STANDBY 상태일 경우 사용 가능하다. 서버가 SHUTDOWN 상태일 경우에는 Node 설정이 제대로 되어 있어야 동작한다.

- alias

startcluster

- 사용법

```
start-cluster <cluster-name>
               [-f, --force]
               [-s, --standby]
               [-seq, --sequential]
```

- 파라미터

파라미터	설명
<cluster-name>	클러스터의 이름이다.
[-f, --force]	강제로 서버를 시작한다.
[-s, --standby]	클러스터를 준비 상태인 STANDBY 상태로 기동한다.
[-seq, --sequential]	클러스터에 속한 서버를 순차적으로 기동한다.

- 예제

```
[MASTER]domain1.adminServer>start-cluster cluster1
The cluster [cluster1] has been successfully started.
```

#### 4.2.3.56. start-domain

현재 도메인에 속한 모든 MS들을 시작한다. SHUTDOWN 상태 또는 STANDBY 상태일 경우 사용 가능하다. 서버가 SHUTDOWN 상태일 경우에는 Node 설정이 제대로 되어 있어야 동작한다.

- alias

startdomain, bootall

- 사용법

```
start-domain [-f, --force]
              [-s, --standby]
```

`[-seq,--sequential]`

- 파라미터

파라미터	설명
<code>[-f, --force]</code>	강제로 서버를 시작한다.
<code>[-s, --standby]</code>	서버를 준비 상태인 STANDBY 상태로 기동한다.
<code>[-seq, --sequential]</code>	도메인에 속한 서버를 순차적으로 기동한다.

- 예제

```
[MASTER]domain1.adminServer>start-domain
All managed servers in the domain have been successfully started.
```

#### 4.2.3.57. start-node

특정 노드에 속한 모든 MS들을 시작한다. SHUTDOWN 상태 또는 STANDBY 상태일 경우 사용 가능하다. 서버가 SHUTDOWN 상태일 경우에는 Node 설정이 제대로 되어 있어야 동작한다.

노드 내에 JEUS Master Server가 포함된 경우 JEUS Master Server는 대상에서 제외한다.

- alias

startnode, bootnode

- 사용법

```
start-node <node-list>
          [-f, --force]
          [-s, --standby]
          [-seq,--sequential]
```

- 파라미터

파라미터	설명
<code>&lt;node-list&gt;</code>	노드의 이름이다.
<code>[-f, --force]</code>	강제로 서버를 시작한다.
<code>[-s, --standby]</code>	서버를 준비 상태인 STANDBY 상태로 기동한다.
<code>[-seq, --sequential]</code>	도메인에 속한 서버를 순차적으로 기동한다.

- 예제

```
[MASTER]domain1.adminServer>start-node node1
```

All managed servers in the node [node1] have been successfully started.

#### 4.2.3.58. start-server

특정 MS를 시작한다. 서버가 STANDBY 상태일 경우 사용 가능하다.

- alias

startserver, boot

- 사용법

```
start-server <server-name>
            [-f, --force]
            [-s, --standby]
            [-r, --restart-policy]
            [-seq, --sequential]
```

- 파라미터

파라미터	설명
<server-name>	서버의 이름이다.
[-f, --force]	강제로 서버를 시작한다.
[-s, --standby]	서버를 준비(STANDBY) 상태로 기동한다.
[-r, --restart-policy]	호스트 인스턴스의 RestartPolicy를 설정한다. (기본값 Never) <ul style="list-style-type: none"><li>◦ Never: 프로세스가 종료되면 재시작하지 않음.</li><li>◦ Onfailure: 사용자의 명시적인 종료 및 sigkill로 프로세스가 종료되지 않으면 재시작.</li><li>◦ Always: 사용자의 명시적인 종료로 프로세스가 종료되지 않으면 재시작.</li></ul>
[-seq, --sequential]	주어진 서버를 순차적으로 기동한다.

- 예제

```
[MASTER]domain1.adminServer>start-server server1
Successfully started the server[server1]. The server state is now RUNNING.
```

#### 4.2.3.59. stop-cluster

특정 클러스터에 속한 Managed Server들을 종료한다. 클러스터 내에 JEUS Master Server가 포함된 경우 JEUS Master Server는 대상에서 제외한다.



- alias

stopcluster, shutdowncluster, downcluster

- 사용법

```
stop-cluster <cluster-name>
        [-to,--shutdowntimeout <timeout> | -f, --force | -g,--graceful]
```

- 파라미터

파라미터	설명
<cluster-name>	클러스터의 이름이다.
[-to,--shutdowntimeout <timeout>]	Graceful하게 종료할 때 사용할 타임아웃을 설정한다. 처리 중인 요청을 입력한 시간만큼 완료되길 기다린다. (단위: ms)
[-f, --force]	Graceful 타임아웃을 적용하지 않고 강제로 종료할지 여부를 설정한다.
[-g, --graceful]	Graceful하게 종료할지 여부를 설정한다. 처리 중인 요청이 끝나길 무한히 기다린다.

- 예제

```
[MASTER]domain1.adminServer>stop-cluster cluster1
The cluster [cluster1] was successfully stopped.
```

#### 4.2.3.60. stop-domain

현재 도메인에 속한 Managed Server들을 종료한다. JEUS Master Server의 종료는 [local-shutdown](#) 명령어를 통해서 가능하다.

- alias

stopdomain, shutdownall, downall

- 사용법

```
stop-domain [-to,--shutdowntimeout <timeout> | -f, --force | -g,--graceful]
        [-i]
```

- 파라미터

파라미터	설명
[-f, --force]	Graceful 타임아웃을 적용하지 않고 강제로 종료할지 여부를 설정한다.

파라미터	설명
[-g, --graceful]	Graceful하게 종료할지 여부를 설정한다. 처리 중인 요청이 끝나길 무한히 기다린다.
[-to, --shutdowntimeout <timeout>]	Graceful하게 종료할 때 사용할 타임아웃을 설정한다. 처리 중인 요청을 입력한 시간만큼 완료되길 기다린다. (단위: second)
[-i]	사용자의 동의를 구하지 않고 바로 종료한다.

- 예제

```
[MASTER]domain1.adminServer>stop-domain
Executing this command affects the service. Do you want to continue? (y/n)n
[stop-domain] has been canceled.
[MASTER]domain1.adminServer>stop-domain -i
Stopping servers [adminServer, ms1].
Stop server message to all managed servers in the domain were successfully sent.
Unable to shut down JEUS Master Server. To shutdown JEUS Master Server, use the jeusadmin's
"local-shutdown" command instead.
```

#### 4.2.3.61. stop-node

특정 노드에 속한 Managed Server들을 종료한다. 노드 내에 JEUS Master Server가 포함된 경우 JEUS Master Server는 대상에서 제외한다.

- alias

stopnode, downnode

- 사용법

```
stop-node <node-list>
        [-to, --shutdowntimeout <timeout> | -f, --force | -g, --graceful]
```

- 파라미터

파라미터	설명
<node-list>	노드의 이름이다.
[-to, --shutdowntimeout <timeout>]	Graceful하게 종료할 때 사용할 타임아웃을 설정한다. 처리 중인 요청을 입력한 시간만큼 완료되길 기다린다. (단위: ms)
[-f, --force]	Graceful 타임아웃을 적용하지 않고 강제로 종료할지 여부를 설정한다.
[-g, --graceful]	Graceful하게 종료할지 여부를 설정한다. 처리 중인 요청이 끝나길 무한히 기다린다.

- 예제

```
[MASTER]domain1.adminServer>stop-node node1
Stop server message to all managed servers in the node were successfully sent.
```

#### 4.2.3.62. stop-server

입력한 단일 혹은 다수의 Managed Server를 종료시킨다. timeout을 주거나 graceful 옵션을 주어 안전하게 종료시킬 수 있다.

- alias

stopserver, down, shutdown, jeusexit

- 사용법

```
stop-server <server-list>
        [-to,--shutdowntimeout <timeout> | -f, --force | -g,--graceful]
```

- 파라미터

파라미터	설명
<server-list>	서버의 이름이다. 서버가 한 개 이상인 경우는 콤마(,)로 구분한다.
[-to,--shutdowntimeout <timeout>]	Graceful하게 종료할 때 사용할 타임아웃을 설정한다. 처리 중인 요청을 입력한 시간만큼 완료되길 기다린다. (단위: second)
[-f, --force]	Graceful shutdown을 시도하지 않고 바로 서버를 종료한다. (기본값)
[-g, --graceful]	Graceful하게 종료할지 여부를 설정한다. 처리 중인 요청이 끝나길 무한히 기다린다.

- 예제

```
[MASTER]domain1.adminServer>stop-server server1
Stop server message to server [server1] was successfully sent.
```

#### 4.2.3.63. suspend-server

서비스 중인 특정 서버를 일시 정지시킨다.

- alias

suspend

- 사용법

```
suspend-server [-clusters <cluster-list>| -servers <server-list>]
```

`[-i]`

- 파라미터

파라미터	설명
<code>[-clusters &lt;cluster-list&gt;]</code>	클러스터의 이름이다. 서버가 한 개 이상인 경우는 콤마(,)로 구분한다.
<code>[-servers &lt;server-list&gt;]</code>	서버의 이름이다. 서버가 한 개 이상인 경우는 콤마(,)로 구분한다.
<code>[-i]</code>	명령어 실행 시 사용자 동의를 구하지 않도록 하는 옵션이다.

- 예제

```
[MASTER]domain1.adminServer>suspend-server -servers server1
Successfully suspended server(s).
```

#### 4.2.3.64. system-info

특정 서버의 시스템 정보를 확인한다.

- alias

systeminfo, sysinfo, sys-info, system-information, system-ststus, systemstatus, systemst

- 사용법

```
system-info [-process,--process | -jvm,--jvm | -CPU,--cpu |
             -patch,--patch | -memory,--memory]
             <server-name>
```

- 파라미터

파라미터	설명
<code>[-process,--process]</code>	프로세스 정보를 출력한다.
<code>[-jvm,--jvm]</code>	JVM 정보를 출력한다.
<code>[-CPU,--cpu]</code>	CPU 정보를 출력한다.
<code>[-patch,--patch]</code>	서버에 적용된 패치 정보를 출력한다.
<code>[-memory,--memory]</code>	메모리 정보를 출력한다.
<code>&lt;server-name&gt;</code>	시스템 정보를 출력하고자 하는 서버의 이름이다.

- 예제

```
[MASTER]domain1.adminServer>system-info -process adminServer
System Information of Server [adminServer]
```

#### Process Information [adminServer]

=====	
Key	Value
Process Id	719151
Thread Count	106
Process CPU Load	0.6 %
User Name	User

### 4.2.3.65. unblock-client

특정 클라이언트 주소에 대한 접속 차단을 해제한다.

- alias

unblockclient, unbanclient

- 사용법

```
unblock-client -server <server-name> -address <client-ip>
```

- 파라미터

파라미터	설명
-server	접속 차단을 해제할 서버의 이름이다.
-address	접속 차단을 해제할 클라이언트의 주소이다.

- 예제

```
[MASTER]domain1.adminServer>unblock-client -server adminServer -address 192.168.2.15
Successfully unblocked.
```

## 4.2.4. Domain Configuration 관련 명령어

Domain Configuration 관련 명령어의 목록은 다음과 같다.

명령어	설명
<a href="#">add-cluster</a>	동적으로 클러스터를 추가한다. JEUS Master Server-Managed Server 구조에서만 수행 가능한 명령어이다.
<a href="#">add-custom-resource</a>	동적으로 custom resource를 추가한다.
<a href="#">add-custom-resource-to-clusters</a>	클러스터에 동적으로 custom resource를 추가한다. JEUS Master Server-Managed Server 구조에서만 수행 가능한 명령어이다.

명령어	설명
<a href="#">add-data-sources-to-cluster</a>	클러스터에 동적으로 데이터소스들을 등록한다. JEUS Master Server-Managed Server 구조에서만 수행 가능한 명령어이다.
<a href="#">add-external-resource</a>	동적으로 external resource를 추가한다.
<a href="#">add-external-resource-to-clusters</a>	클러스터에 동적으로 external resource를 추가한다. JEUS Master Server-Managed Server 구조에서만 수행 가능한 명령어이다.
<a href="#">add-server</a>	동적으로 서버를 추가한다. JEUS Master Server-Managed Server 구조에서만 수행 가능한 명령어이다.
<a href="#">add-servers-to-cluster</a>	동적으로 특정 서버를 특정 클러스터에 추가한다. JEUS Master Server-Managed Server 구조에서만 수행 가능한 명령어이다.
<a href="#">add-server-template</a>	동적으로 여러 서버에 공통적으로 적용할 수 있는 설정 템플릿을 추가한다. JEUS Master Server-Managed Server 구조에서만 수행 가능한 명령어이다.
<a href="#">add-dynamic-servers</a>	미리 만들어진 설정 템플릿을 이용하여 자동 생성한 서버들로 이루어진 클러스터를 추가한다. JEUS Master Server-Managed Server 구조에서만 수행 가능한 명령어이다.
<a href="#">apply-configuration-plan</a>	기존 domain.xml에 configuration plan을 적용한 새로운 domain.xml을 생성하는 Off-line 명령어이다.
<a href="#">config-system-clustering-framework</a>	System Clustering Framework에 관련된 설정을 변경한다.
<a href="#">create-domain</a>	새로운 도메인을 생성하는 Off-line 명령어이다. 기본으로 생성되는 사용자에게 대한 암호는 별도로 입력해야 한다.
<a href="#">delete-domain</a>	생성된 도메인을 제거하는 Off-line 명령어이다.
<a href="#">disable-json-command</a>	JsonCommand를 비활성화한다.
<a href="#">disable-to-resynchronize-applications</a>	Managed Server가 INDEPENDENT 상태에서 DEPENDENT 상태로 변경되었을 때 애플리케이션을 동기화하지 않는다.
<a href="#">enable-json-command</a>	JsonCommand를 활성화한다.
<a href="#">enable-to-resynchronize-applications</a>	Managed Server가 INDEPENDENT 상태에서 DEPENDENT 상태로 변경되었을 때 애플리케이션을 동기화한다.
<a href="#">list-clusters</a>	현재 도메인에 존재하는 클러스터의 정보를 확인한다. JEUS Master Server-Managed Server 구조에서만 수행 가능한 명령어이다.
<a href="#">list-custom-resources</a>	현재 도메인에 설정된 custom resource를 확인한다.
<a href="#">list-external-resources</a>	현재 도메인에 설정된 external resource를 확인한다.
<a href="#">list-domains</a>	현재 구성된 도메인의 목록을 조회하는 Off-line 명령어이다.
<a href="#">list-servers</a>	현재 도메인에 존재하는 서버의 정보를 확인한다.
<a href="#">modify-cluster</a>	동적으로 특정 클러스터의 설정을 변경한다. JEUS Master Server-Managed Server 구조에서만 수행 가능한 명령어이다.
<a href="#">modify-server</a>	동적으로 특정 서버의 설정을 변경한다.

명령어	설명
<code>modify-dynamic-servers</code>	Dynamic Servers 설정으로 만들어진 클러스터의 설정 정보를 변경한다. JEUS Master Server-Managed Server 구조에서만 수행 가능한 명령어이다.
<code>pack-domain</code>	도메인 백업을 이용한 장애 극복을 위해 특정 도메인 폴더를 압축하는 명령어이다.
<code>remove-cluster</code>	동적으로 클러스터를 삭제한다. JEUS Master Server-Managed Server 구조에서만 수행 가능한 명령어이다.
<code>remove-custom-resource</code>	동적으로 custom resource를 삭제한다.
<code>remove-custom-resource-from-clusters</code>	클러스터에 등록된 custom resource를 동적으로 삭제한다. JEUS Master Server-Managed Server 구조에서만 수행 가능한 명령어이다.
<code>remove-data-sources-from-cluster</code>	클러스터에 등록된 데이터소스들을 동적으로 제거한다. JEUS Master Server-Managed Server 구조에서만 수행 가능한 명령어이다.
<code>remove-domain-log-config</code>	도메인 내 log 파일들의 소유자 및 권한 설정을 삭제한다.
<code>remove-external-resource</code>	동적으로 external resource를 삭제한다.
<code>remove-external-resource-from-clusters</code>	클러스터에 등록된 external resource를 동적으로 삭제한다. JEUS Master Server-Managed Server 구조에서만 수행 가능한 명령어이다.
<code>remove-server</code>	동적으로 특정 서버를 삭제한다.
<code>remove-servers-from-cluster</code>	동적으로 특정 클러스터에서 특정 서버를 삭제한다. JEUS Master Server-Managed Server 구조에서만 수행 가능한 명령어이다.
<code>set-domain-backup</code>	도메인 백업 기능에 대한 정책을 설정한다.
<code>set-domain-log-config</code>	도메인 내 log 파일들의 소유자 및 권한을 설정한다. 도메인에 속해 있는 서버들의 로거 설정 중 로그 파일 소유자 및 권한 설정이 없을 경우 이 설정을 따르게 된다.
<code>show-domain-log-config</code>	도메인 내 log 파일들의 소유자 및 권한 설정을 보여준다.
<code>show-dynamic-servers</code>	동적 서버 생성으로 만들어진 클러스터의 설정 정보를 보여준다. JEUS Master Server-Managed Server 구조에서만 수행 가능한 명령어이다.
<code>show-server-template</code>	설정된 서버 템플릿의 리스트를 조회한다. JEUS Master Server-Managed Server 구조에서만 수행 가능한 명령어이다.
<code>show-system-clustering-framework</code>	System Clustering Framework에 관련된 설정을 조회한다.
<code>unpack-domain</code>	서버에 장애가 생겼을 때 pack-domain에 의해 압축된 파일을 해제하는 명령어이다.

#### 4.2.4.1. add-cluster

동적으로 클러스터를 추가한다.

- 사용법

```
add-cluster [<cluster-name>]
            [-servers <server-list>]
            [-f,--forceLock]
            [-detail]
```

- 파라미터

파라미터	설명
[<cluster-name>]	추가할 클러스터의 이름이다.
[-servers <server-list>]	이 클러스터에 포함할 서버의 이름이다. 서버가 한 개 이상인 경우는 콤마(,)로 구분한다.
[-f,--forceLock]	설정 변경을 강제로 진행하여 반영한다.
[-detail]	동적 변경 결과에 대한 상세한 정보가 출력된다.

- 예제

- 현재 존재하는 클러스터 확인

```
[MASTER]domain1.adminServer>list-clusters
List of Clusters
=====
+-----+-----+-----+-----+-----+
| Cluster | Server List | MEJB | Class FTP | Running |
+-----+-----+-----+-----+-----+
| cluster1 | server1, server2 | false | false | true |
+-----+-----+-----+-----+-----+
=====
```

- 현재 존재하지 않는 클러스터 이름으로 클러스터를 생성하고 다른 클러스터에 포함되지 않는 서버들로 클러스터 구성한 경우

```
[MASTER]domain1.adminServer>add-cluster cluster2 -servers server3,server4
Successfully performed the ADD operation for cluster (cluster2).
Check the results using "list-clusters or add-cluster"
```

#### 4.2.4.2. add-custom-resource

동적으로 custom resource를 추가한다.

- alias

addcustomresource, addcr

- 사용법



```
add-custom-resource [<export-name>]
                    [-resource,--resourceClass <resource-class>]
                    [-factory,--factoryClass <factory-class>]
                    [-p,--properties <properties>]
                    [-servers <server-list>]
                    [-clusters <cluster-list>]
                    [-f,--forceLock]
                    [-detail]
```

- 파라미터

파라미터	설명
[<export-name>]	추가할 custom resource의 JNDI 바인드 이름이다.
[-resource,--resourceClass <resource-class >]	custom resource의 클래스 이름을 지정한다.
[-factory,--factoryClass <factory-class >]	custom resource를 생성하는 factoryClass 이름을 지정한다.
[-p,--properties <properties>]	custom resource의 property를 지정한다.
[-servers <server-list>]	custom resource를 도메인에 추가함과 동시에 주어진 server list에 등록한다.
[-clusters <cluster-list>]	custom resource를 도메인에 추가함과 동시에 주어진 cluster list에 등록한다.
[-f,--forceLock]	설정 변경을 강제로 진행하여 반영한다.
[-detail]	동적 변경 결과에 대한 상세한 정보가 출력된다.

- 예제

```
[MASTER]domain1.adminServer>add-custom-resource custom/dog -resource dog.Dog -factory
dog.DogFactory
Successfully performed the ADD operation for A custom resource.
Check the results using "list-custom-resources or add-custom-resource"
```

#### 4.2.4.3. add-custom-resource-to-clusters

동적으로 클러스터에 custom resource를 추가한다.

- alias

add-cr-to-clusters

- 사용법

```
add-custom-resource-to-clusters <export-name>
```

```
-clusters <cluster-list>
[-f,--forceLock]
[-detail]
```

- 파라미터

파라미터	설명
<export-name>	클러스터에 추가할 custom resource의 JNDI 바인드 이름이다.
-clusters <cluster-list>	custom resource를 주어진 cluster list에 등록한다.
[-f,--forceLock]	설정 변경을 강제로 진행하여 반영한다.
[-detail]	동적 변경 결과에 대한 상세한 정보가 출력된다.

- 예제

```
[MASTER]domain1.adminServer>add-custom-resource-to-clusters custom/dog -clusters cluster1
Successfully performed the ADD operation for A custom resource.
Check the results using "list-custom-resources"
```

#### 4.2.4.4. add-data-sources-to-cluster

동적으로 클러스터에 데이터소스들을 등록한다.

- alias

adddstocluster

- 사용법

```
add-data-sources-to-cluster -cluster <cluster-name>
[-ids <data-source-id-list> | -all]
[-f,--forceLock]
[-detail]
```

- 파라미터

파라미터	설명
-cluster <cluster-name>	데이터소스를 등록할 클러스터의 이름이다.
[-ids <data-source-id-list>]	클러스터에 등록할 데이터소스들의 ID 목록이다. 한 개 이상인 경우 콤마(,)로 구분한다.
[-all]	도메인에 정의된 모든 데이터소스를 클러스터에 등록한다.
[-f,--forceLock]	설정 변경을 강제로 진행하여 반영한다.
[-detail]	동적 변경 결과에 대한 상세한 정보가 출력된다.

- 예제

```
[MASTER]domain1.adminServer>add-data-sources-to-cluster -cluster cluster1 -ids datasource1
Successfully performed the ADD operation for data sources to the cluster [cluster1].
Check the results using "add-data-sources-to-cluster -cluster cluster1"
```

#### 4.2.4.5. add-external-resource

동적으로 external resource를 추가한다.

- alias

addexternalresource, adder

- 사용법

```
add-external-resource [<export-name>]
                        [-class,--resourceClass <resource-class>]
                        [-p,--properties <properties>]
                        [-servers <server-list>]
                        [-clusters <cluster-list>]
                        [-f,--forceLock]
                        [-detail]
```

- 파라미터

파라미터	설명
[<export-name>]	추가할 external resource의 이름이다.
[-class,--resourceClass <resource-class>]	external resource의 클래스 이름을 지정한다.
[-p,--properties <properties>]	external resource의 property를 지정한다.
[-servers <server-list>]	external resource를 도메인에 추가함과 동시에 주어진 server list에 등록한다.
[-clusters <cluster-list>]	external resource를 도메인에 추가함과 동시에 주어진 cluster list에 등록한다.
[-f,--forceLock]	설정 변경을 강제로 진행하여 반영한다.
[-detail]	동적 변경 결과에 대한 상세한 정보가 출력된다.

- 예제

```
[MASTER]domain1.adminServer>add-external-resource test/ext -resource
test.ext.TestResourceBootstrapper
Successfully performed the ADD operation for A external resource.
Check the results using "list-external-resources or add-external-resource"
```

#### 4.2.4.6. add-external-resource-to-clusters

동적으로 클러스터에 external resource를 추가한다.

- alias

add-er-to-clusters

- 사용법

```
add-external-resource-to-clusters <export-name>
                                -clusters <cluster-list>
                                [-f,--forceLock]
                                [-detail]
```

- 파라미터

파라미터	설명
<export-name>	클러스터에 추가할 external resource의 이름이다.
-clusters <cluster-list>	external resource를 주어진 cluster list에 등록한다.
[-f,--forceLock]	설정 변경을 강제로 진행하여 반영한다.
[-detail]	동적 변경 결과에 대한 상세한 정보가 출력된다.

- 예제

```
[MASTER]domain1.adminServer>add-external-resource-to-clusters test/ext -clusters cluster1
Successfully performed the ADD operation for A external resource.
Check the results using "list-external-resources"
```

#### 4.2.4.7. add-server

동적으로 서버를 추가한다.

- alias

addserver

- 사용법

```
add-server [<server-name>]
          [-target,--duptargetserver <target-server>]
          [-port,--baseport <base-port>]
          [-addr,--baseaddr <base-addr>]
          [-http,--httpport <http-port>]
          [-rg,--replicate-group <name>]
          [-node,--nodeName <node-name>]
```

```

[-jvm,--jvmOptions <jvm-options>]
[-logdir,--logHomeDirectory <server-log-home-directory>]
[-a,--actionOnResourceLeak <action-on-resource-leak>]
[-l,--logStdoutToRawFormat <true | false>]
[-m,--mejb <true | false>]
[-c,--classFtp <true | false>]
[-f,--forceLock]
[-detail]

```

#### • 파라미터

파라미터	설명
[<server-name>]	서버의 이름이다.
[-target,--duptargetserver <target-server>]	서버 설정을 복사할 타겟 서버의 이름이다.
[-port,--baseport <base-port>]	서버의 기본 동작 포트 번호이다.  기본값으로 9736이 지정되나 같은 머신에 여러 개의 서버를 생성할 경우 구분이 필요한 설정이다.
[-addr,--baseaddr <base-addr>]	서버의 기본 동작 주소이다.
[-http,--httpport <http-port>]	서버의 기본 HTTP Listener의 포트 번호이다.  기본값으로 8088이 지정되나 같은 머신에 여러 개의 서버를 생성할 경우 구분이 필요한 설정이다.
[-rg,--replicate-group <name>]	서버의 JNDI Naming Server Replicate Group을 지정한다.  클러스터와는 별개로 독립적인 JNDI Naming Server만의 그룹 스코프로 같은 값을 갖는 서버들간에는 바인딩 오브젝트의 복제, 그룹 관리 등 JNDI에 한하여 클러스터와 동일하게 동작한다. 서버 클러스터링은 하지 않지만 JNDI와 EJB에 한해서만 클러스터링 효과를 가져가고 싶은 경우 사용 가능하다.
[-node,--nodeName <node-name>]	서버가 실제로 동작하게 될 머신을 의미하는 노드의 이름이다.
[-jvm,--jvmOptions <jvm-options>]	서버에 적용하고 하는 JVM 옵션이다.  공백이 존재하는 경우 큰따옴표(" ")로 묶어서 입력한다.
[-logdir,--logHomeDirectory <server-log-home-directory>]	서버에서 생성되는 로그들이 공통적으로 들어갈 로그 디렉터리를 의미한다.
[-a,--actionOnResourceLeak <action-on-resource-leak>]	리소스에 leak이 발생할 경우 취할 액션을 설정한다.  다음 중 하나를 입력한다. <ul style="list-style-type: none"> <li>• AutoClose</li> <li>• NoAction</li> <li>• Warning</li> </ul>

파라미터	설명
<code>[-l,--logStdoutToRawFormat &lt;true   false&gt;]</code>	<b>[동적 변경]</b> stdout을 그대로 출력할 것인지 JEUS 로거 포맷으로 출력할 것인지 설정한다. (true   false)
<code>[-m,--mejb &lt;true   false&gt;]</code>	<b>[동적 변경]</b> MEJB를 사용할지 여부를 설정한다. (true   false)
<code>[-c,--classFtp &lt;true   false&gt;]</code>	<b>[동적 변경]</b> class FTP 서비스를 사용할지 여부를 설정한다. (true   false)
<code>[-f,--forceLock]</code>	설정 변경을 강제로 진행하여 반영한다.
<code>[-detail]</code>	동적 변경 결과에 대한 상세한 정보가 출력된다.

- 예제

- 현재 존재하는 서버 리스트 확인

```
[MASTER]domain1.adminServer>list-servers
List of Editable Servers
=====
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Serv | Base | Node | JVM | Action | Stdout | MEJB | Cla | Server | Type |
| er | Listen | | Configs | On | to Raw | | ss | Log Home | |
| | Address | | | Resource | Format | | FTP | Directo | |
| | /Port | | | Leak | | | | ry | |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| admi | 0.0.0.0 | node1 | -Xmx10 | Warning | true | false | true | none | server |
| nServ | / 9736 | | 24m | | | | | | |
| er | | | -XX:Max | | | | | | |
| | | | Metaspa | | | | | | |
| | | | ceSize= | | | | | | |
| | | | 512m | | | | | | |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| serve | 0.0.0.0 | node1 | -Xmx10 | Warning | true | false | false | none | server |
| r1 | / 19736 | | 24m | | | | | | |
| | | | -XX:Max | | | | | | |
| | | | Metaspa | | | | | | |
| | | | ceSize= | | | | | | |
| | | | 512m | | | | | | |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
=====
```

- 현재 존재하지 않는 이름으로 서버 추가

```
[MASTER]domain1.adminServer>add-server server2 -port 10000 -node node1 -m false
Successfully performed the ADD operation for server (server2).
NOTICE : base-addr [0.0.0.0] base-port [10000] http-port [8088]
Check the results using "list-servers or add-server"
[MASTER]domain1.adminServer>add-server server3 -port 11000 -http 18088 -node node1
Successfully performed the ADD operation for server (server3).
NOTICE : base-addr [0.0.0.0] base-port [11000] http-port [18088]
Check the results using "list-servers or add-server"
```

#### 4.2.4.8. add-servers-to-cluster

동적으로 특정 서버를 특정 클러스터에 추가한다.

- 사용법

```
add-servers-to-cluster <cluster-name>
                        [-servers <server-list>]
                        [-f,--forceLock]
                        [-detail]
```

- 파라미터

파라미터	설명
<cluster-name>	클러스터의 이름이다.
[-servers <server-list>]	이 클러스터에 포함하고 시키려는 서버의 이름이다. 서버가 한 개 이상인 경우는 콤마(,)로 구분한다.
[-f,--forceLock]	설정 변경을 강제로 진행하여 반영한다.
[-detail]	동적 변경 결과에 대한 상세한 정보가 출력된다.

- 예제

```
[MASTER]domain1.adminServer>add-servers-to-cluster cluster1 -servers server1
Successfully performed the ADD operation for The server list for cluster(cluster1)..
Check the results using "list-clusters cluster1 or add-servers-to-cluster cluster1"
```

#### 4.2.4.9. add-server-template

동적으로 여러 서버에 공통적으로 적용할 수 있는 설정 템플릿을 추가한다.

- 사용법

```
add-server-template [<server-template-name>]
                    [-port,--baseport <base-port>]
                    [-addr,--baseaddr <base-addr>]
                    [-http,--httpport <http-port>]
                    [-node,--nodeName <node-name>]
                    [-rg,--replicate-group <name>]
                    [-jvm,--jvmOptions <jvm-options>]
                    [-logdir,--logHomeDirectory <server-log-home-directory>]
                    [-a,--actionOnResourceLeak <action-on-resource-leak>]
                    [-l,--logStdoutToRawFormat <true | false>]
                    [-m,--mejb <true | false>]
                    [-c,--classFtp <true | false>]
                    [-f,--forceLock]
```

- 파라미터

파라미터	설명
<server-template-name>	추가할 서버 템플릿의 이름이다.
[-port,--baseport <base-port>]	서버의 기본 동작 포트 번호이다.  기본값으로 9736이 지정된다.
[-addr,--baseaddr <base-addr>]	서버의 기본 동작 주소이다.
[-http,--httpport <http-port>]	서버의 기본 HTTP Listener의 포트 번호이다.  기본값으로 8088이 지정된다.
[-node,--nodeName <node-name>]	서버가 실제로 동작하게 될 머신을 의미하는 노드의 이름이다.
[-rg,--replicate-group <name>]	서버 템플릿의 JNDI Naming Server Replicate Group을 지정한다.  클러스터와는 별개로 독립적인 JNDI Naming Server만의 그룹 스코프로 같은 값을 갖는 서버들간에는 바인딩 오브젝트의 복제, 그룹 관리 등 JNDI에 한하여 클러스터와 동일하게 동작한다. 서버 클러스터링은 하지 않지만 JNDI와 EJB에 한해서만 클러스터링 효과를 가져가고 싶은 경우 사용 가능하다.
[-jvm,--jvmOptions <jvm-options>]	템플릿에 추가할 JVM 옵션이다.  공백이 존재하는 경우 큰따옴표(" ")로 묶어서 입력한다.
[-logdir,--logHomeDirectory <server-log-home-directory>]	템플릿으로 설정된 서버에서 생성되는 로그들이 공통적으로 들어갈 로그 디렉터리를 의미한다.
[-a,--actionOnResourceLeak <action-on-resource-leak>]	리소스에 leak이 발생할 경우 취할 액션을 설정한다.  다음 중에 하나를 설정한다. <ul style="list-style-type: none"> <li>• AutoClose</li> <li>• NoAction</li> <li>• Warning</li> </ul>
[-l,--logStdoutToRawFormat <true   false>]	<b>[동적 변경]</b> stdout을 그대로 출력할 것인지 JEUS 로거 포맷으로 출력할 것인지 설정한다. (true   false)
[-m,--mejb <true   false>]	<b>[동적 변경]</b> MEJB를 사용할지 여부를 설정한다. (true   false)
[-c,--classFtp <true   false>]	<b>[동적 변경]</b> class FTP 서비스를 사용할지 여부를 설정한다. (true   false)
[-f,--forceLock]	설정 변경을 강제로 진행하여 반영한다.
[-detail]	동적 변경 결과에 대한 상세한 정보가 출력된다.



- 예제

```
[MASTER]domain1.adminServer>add-server-template temp1
Successfully performed the ADD operation for server template (temp1), but all changes were non-
dynamic. They will be applied after restarting.
NOTICE : base-addr [0.0.0.0] base-port [9736] http-port [8088]
Check the results using "list-servers or show-server-template or add-server-template".
```

#### 4.2.4.10. add-dynamic-servers

미리 설정된 템플릿을 이용하여 자동 생성한 서버들로 이루어진 클러스터를 생성한다. 미리 템플릿을 통하여 설정 정보를 저장한 다음에 사용하여야 한다.

- alias

addservers

- 사용법

```
add-dynamic-servers <cluster-name>
    -count <the-number-of-dynamic-servers>
    [-nodes,--nodeNames <node-names>]
    [-prefix,--serverNamePrefix <server-name-prefix>]
    -template,--serverTemplateName <server-template-name>
    [-step,--listenPortStep <listen-port-step>]
    [-f,--forceLock]
    [-detail]
```

- 파라미터

파라미터	설명
<cluster-name>	동적 설정 서버들을 이용하여 생성할 클러스터 이름을 지정한다.
-count <the-number-of-dynamic-servers>	클러스터에 생성할 서버의 수를 정한다.
[-nodes,--nodeNames <node-names>]	추가될 서버가 위치하게 될 노드들을 설정한다. 콤마(,)를 구분자로 하여 여러 개를 설정할 수 있다.
[-prefix,--serverNamePrefix <server-name-prefix>]	생성될 서버의 이름 앞에 붙을 prefix를 설정한다.  Prefix가 설정되지 않은 경우에는 클러스터 이름 뒤에 '_server_'를 붙인다.
-template, --serverTemplateName <server-template-name>	동적(Dynamic) 서버들에게 공통적으로 적용될 설정 템플릿을 지정한다.
[-step,--listenPortStep <listen-port-step>]	같은 노드에 생성된 서버들의 포트 번호를 몇 씩 증가시킬지 지정한다.

파라미터	설명
[-f,--forceLock]	설정 변경을 강제로 진행하여 반영한다.
[-detail]	동적 변경 결과에 대한 상세한 정보가 출력된다.

- 예제

```
[MASTER]domain1.adminServer>add-dynamic-servers cluster1 -count 2 -nodes node1 -template temp1
Successfully performed the ADD operation for Dynamic servers configuration in the cluster
(cluster1).
Check the results using "list-clusters or show-dynamic-servers".
```

#### 4.2.4.11. apply-configuration-plan

기존 domain.xml에 configuration plan을 적용한 새로운 domain.xml을 생성하는 명령어이다.

- alias

applycp

- 사용법

```
apply-configuration-plan -plan,--configPlanPath <configuration-plan-path>
                        -src,--configSourcePath <configuration-source-path>
                        -target,--configTargetPath <configuration-target-path>
```

- 파라미터

파라미터	설명
-plan,--configPlanPath <configuration-plan-path>	적용할 configuration plan의 경로를 지정한다.
-src,--configSourcePath <configuration-source-path>	configuration plan을 적용할 기존 domain.xml의 경로를 지정한다.
-target,--configTargetPath <configuration-target-path>	configuration plan이 적용된 결과 파일의 경로를 지정한다.

- 예제

```
offline>apply-configuration-plan -src /home/jeus/domains/domain1/config/domain.xml
-plan /home/plans/jeus-configuration-plan.xml -target
/home/jeus/domains/domain1/config/domain.xml.new
Successfully applied configuration plan.
```

- 참고

본 명령어는 서버에 접속하지 않은 상태에서도 사용할 수 있다.

#### 4.2.4.12. config-system-clustering-framework

System Clustering Framework 기능과 관련된 설정을 변경한다.

- alias

config-scf, configscf, scf

- 사용법

```
config-system-clustering-framework
    [-addr <transport-address>]
    [-port <transport-port>]
    [-transport <transport>]
    [-discovery <leader-discovery-timeout>]
    [-detect <failure-detection-timeout>]
    [-max <max-thread-pool-size>]
    [-min <min-thread-pool-size>]
    [-f, --forceLock]
```

- 파라미터

파라미터	설명
[-addr <transport-address>]	Transport의 동작에 필요한 주소값을 설정한다.  설정된 Transport의 종류에 따라서 의미가 달라진다. Hybrid Transport의 경우에는 Multicast-address를, Tree Type Virtual Multicast의 경우에는 Root Node의 IP Address를 설정해야 한다. Mesh Type Virtual Multicast의 경우에는 각 서버의 Listen Address를 설정값으로 사용하므로 이곳에 명시할 필요는 없다. (기본값: 224.0.0.1)
[-port <transport-port>]	Transport의 동작에 필요한 포트 번호를 설정한다.  이 부분 역시 설정한 Transport의 종류에 따라서 의미가 달라진다. (기본값: 12488)

파라미터	설명
[-transport <transport>]	<p>Transport의 종류를 명시한다.</p> <p>DUMMY, HYBRID, MESH, TREE 네 가지 모드 중에서 선택할 수 있다. (기본값: HYBRID)</p> <ul style="list-style-type: none"> <li>◦ DUMMY: 타 서버와 일체의 통신을 하지 않고 독립적으로 동작하는 모드</li> <li>◦ HYBRID: 각 서버간의 Health Check 및 간단한 메시징은 Multicast로 수행하고, 대용량의 메시징은 TCP/IP로 수행하는 모드</li> <li>◦ MESH: TCP/IP를 이용하여 모든 서버가 Full Mesh 방식으로 연결하는 모드</li> <li>◦ TREE: TCP/IP를 이용하여 연결하되 각 서버들은 Root로만 연결하여 Tree 형태가 되도록 하는 모드</li> </ul>
[-discovery <leader-discovery-timeout>]	<p>최초에 기동하여 기존에 형성되어있는 Group의 Leader를 찾아내기까지 허용되는 최대 시간을 설정한다.</p> <p>설정된 시간이 지나도 Leader를 찾지 못한다면 자신이 Leader가 되었다고 선언한다. 이 시간은 새 Leader 선출에도 이용된다. (기본값: 3000 (ms))</p>
[-detect <failure-detection-timeout>]	<p>상대방 Peer들의 장애 감지를 수행하는 주기 및 timeout 등으로 사용되는 값을 설정한다. (기본값: 3000 (ms))</p>
[-max <max-thread-pool-size>]	<p>System Clustering Framework에서 사용하는 Thread pool 크기의 최대치를 설정한다. (기본값: 10)</p>
[-min <min-thread-pool-size>]	<p>System Clustering Framework에서 사용하는 Thread pool 크기의 최소치를 설정한다. (기본값: 0)</p>
[-f,--forceLock]	<p>설정 변경을 강제로 진행하여 반영한다.</p>

#### • 예제

```
[MASTER]domain1.adminServer>config-system-clustering-framework -port 15000
Executed successfully, but some configurations were not applied dynamically. It might be
necessary to restart the server.
Check the result using 'show-system-clustering-framework'.
```

### 4.2.4.13. create-domain

새로운 도메인을 생성하는 명령어이다. 기본으로 생성되는 사용자에게 암호는 별도로 입력해야 한다.

사용자 암호는 AES로 암호화되어 저장된다. option에 대한 default 값은 JEUS\_HOME\setup\domain-config-template.properties의 값을 따른다.

#### • alias

createdomain

#### • 사용법

```

create-domain -domain <domain-name>
               [-mastername <master-name>]
               [-baseport <master-base-port>]
               [-baseaddress <master-base-address>]
               [-jmsport <jms-port>]
               [-user <user-name>]
               [-httpport <http-port>]
               [-jvmconfig <jvm-config>]
               [-node <node-name>]
               [-pw,--password]
               [-ttype,--transport.type <transport-type>]
               [-productionmode,--productionmode]
               [-taddress,--transport.address <transport-address>]
               [-tport,--transport.port <transport-port>]

```

#### • 파라미터

파라미터	설명
-domain <domain-name>	도메인의 이름이다.
[-mastername <master-name>]	도메인을 관리하는 JEUS Master Server의 이름이다.
[-baseport <master-base-port>]	JEUS Master Server의 기본 동작 포트 번호이다.
[-baseaddress <master-base-address>]	JEUS Master Server의 기본 동작 주소이다.
[-user <user-name>]	도메인의 사용자 이름이다.
[-httpport <http-port>]	JEUS Master Server의 HTTP 포트 번호이다.
[-jvmconfig <jvm-config>]	JEUS Master Server에 적용할 JVM 설정이다.
[-node <node-name>]	JEUS Master Server에 적용할 노드 이름이다.
[-pw,--password]	템플릿 프로퍼티 파일과 상관없이 무조건 password를 콘솔에서 입력받게 된다.
[-ttype,--transport.type <transport-type>]	SCF의 전송 형식을 지정한다.
[-productionmode,--productionmode]	생성하는 JEUS의 도메인을 production mode에서 사용할지 여부이다.
[-taddress,--transport.address <transport-address>]	SCF의 전송 주소를 지정한다.
[-tport,--transport.port <transport-port>]	SCF의 전송 포트를 지정한다.

#### • 예제

```

offline>create-domain -domain domain1 -mastername adminServer -baseport 9763 -jvmconfig "-Xmx1024m -XX:MaxPermSize=256m"

```

```

Enter the password for [administrator]: *****
Confirm the password: *****
Do you want to encrypt the password? (y/n): y
=====
The domain [domain1] was created successfully.

+-----+-----+
| Property | Value |
+-----+-----+
| Domain name | domain1 |
+-----+-----+
| JEUS Master Server Name | adminServer |
+-----+-----+
| JEUS Master Server Base Listen | 9763 |
| Port | |
+-----+-----+
| JEUS Master Server Base Listen | 0.0.0.0 |
| Address | |
+-----+-----+
| HTTP Port | 8088 |
+-----+-----+
| User Name | administrator |
+-----+-----+
| Node Name | node1 |
+-----+-----+
| Password | {AES:256}lB1jYlI1BrujFDC7v3NSoHZQprgnncVmdB |
| | bkt4oxDRg= |
+-----+-----+
| JVM Option | -Xmx1024m -XX:MaxMetaspaceSize=256m |
+-----+-----+
| Production Mode | true |
+-----+-----+
| Transport Type | HYBRID |
+-----+-----+
=====

```

- 참고

본 명령어는 서버에 접속하지 않은 상태에서도 사용할 수 있다.

#### 4.2.4.14. delete-domain

생성된 도메인을 삭제하는 명령어이다. 현재 접속 중인 서버가 속한 도메인은 삭제하지 못한다.

- alias

deletedomain

- 사용법

```

delete-domain -domain <domain-name>
               [-f, --force]

```

- 파라미터

파라미터	설명
-domain <domain-name>	도메인의 이름이다.
[-f, --force]	도메인 삭제 여부를 묻지 않고 삭제한다.

- 예제

```
offline>delete-domain -domain newDomain
Delete the following domain: [newDomain]
Do you want to execute the delete domain command? (y/n)y
Deleting the domain directory was successful.

offline>delete-domain -domain newDomain -f
Deleting the domain directory was successful.
```

- 참고

본 명령어는 서버에 접속하지 않은 상태에서도 사용할 수 있다.

#### 4.2.4.15. disable-json-command

JsonCommand를 비활성화한다.

- alias

suspend-json-command, jsonoff

- 사용법

```
disable-json-command [-f,--forceLock]
```

- 파라미터

파라미터	설명
[-f,--forceLock]	설정 변경을 강제로 진행하여 반영한다.

- 예제

```
[MASTER]domain1.adminServer>disable-json-command
JsonCommand was successfully disabled.
Applying configuration ...
=====
+-----+
|                                     Result                                     |
+-----+
```

```
+-----+
| Successfully applied the configuration changes. |
| Restart the server to apply the changes.       |
+-----+
=====
```

#### 4.2.4.16. disable-to-resynchronize-applications

INDEPENDENT 상태인 Managed Server가 DEPENDENT 상태가 되었을 때 애플리케이션을 동기화하지 않는다.

- alias

disalbe-resync-apps, disable-resyncapps

- 사용법

```
disable-to-resynchronize-applications [-f,--forceLock]
```

- 파라미터

파라미터	설명
[-f,--forceLock]	설정 변경을 강제로 진행하여 반영한다.

- 예제

```
[MASTER]domain1.adminServer>disable-to-resynchronize-applications
ResyncAppsCommand was successfully disabled.
Applying configuration ...
=====
+-----+
|                               Result                               |
+-----+
| Successfully applied the configuration changes.                   |
+-----+
=====
```

#### 4.2.4.17. enable-json-command

JsonCommand를 활성화한다.

- alias

resume-json-command, jsonon

- 사용법

```
enable-json-command [-f,--forceLock]
```



- 파라미터

파라미터	설명
[-f,--forceLock]	설정 변경을 강제로 진행하여 반영한다.

- 예제

```
[MASTER]domain1.adminServer>enable-json-command
JsonCommand was successfully enabled.
Applying configuration ...
=====
+-----+
|                                     Result                                     |
+-----+
| Successfully applied the configuration changes.                             |
| Restart the server to apply the changes.                                   |
+-----+
=====
```

#### 4.2.4.18. enable-to-resynchronize-applications

INDEPENDENT 상태인 Managed Server가 DEPENDENT 상태가 되었을 때 애플리케이션을 동기화한다.

- alias

enable-resynch-apps, enable-resyncapps, enableresyncapps

- 사용법

```
enable-to-resynchronize-applications [-f,--forceLock]
```

- 파라미터

파라미터	설명
[-f,--forceLock]	설정 변경을 강제로 진행하여 반영한다.

- 예제

```
[MASTER]domain1.adminServer>enable-to-resynchronize-applications
ResyncAppsCommand was successfully enabled.
Applying configuration ...
=====
+-----+
|                                     Result                                     |
+-----+
| Successfully applied the configuration changes.                             |
+-----+
=====
```

#### 4.2.4.19. list-clusters

현재 도메인에 존재하는 클러스터의 정보를 확인한다.

- 사용법

```
list-clusters [<cluster-name>]
```

- 파라미터

파라미터	설명
[<cluster-name>]	클러스터의 이름이다.

- 예제

```
[MASTER]domain1.adminServer>list-clusters
List of Clusters
=====
+-----+-----+-----+-----+
| Cluster | Server List | MEJB | Class FTP | Running |
+-----+-----+-----+-----+
| cluster1 | newServer, server1 | | | true |
+-----+-----+-----+-----+
=====
```

#### 4.2.4.20. list-custom-resources

현재 도메인에 설정된 custom resource를 확인한다.

- alias

listcustomresources, list-cr, listcr

- 사용법

```
list-custom-resources [-name <export-name>] |
                      [-cluster <cluster-name>] |
                      [-server <server-name>]
```

- 파라미터

파라미터	설명
[-name <export-name>]	custom resource의 이름이다.
[-cluster <cluster-name>]	클러스터에 등록된 custom resource를 확인하기 위한 클러스터 이름이다.

파라미터	설명
<code>[-server &lt;server-name&gt;]</code>	서버에 등록된 custom resource를 확인하기 위한 서버 이름이다.

- 예제

```
[MASTER]domain1.adminServer>list-custom-resources
List of Custom Resources
=====
+-----+-----+-----+-----+
| Export Name | Resource Class | Factory Class | Properties |
+-----+-----+-----+-----+
| custom/dog | dog.Dog        | dog.DogFactory | [test=1,   |
|              |                 |                 | test1=2]   |
+-----+-----+-----+-----+
=====
```

#### 4.2.4.21. list-external-resources

현재 도메인에 설정된 external resource를 확인한다.

- alias

listexternalresources, list-er, lister

- 사용법

```
list-external-resources [-name <export-name>] |
                        [-cluster <cluster-name>] |
                        [-server <server-name>]
```

- 파라미터

파라미터	설명
<code>[-name &lt;export-name&gt;]</code>	external resource의 이름이다.
<code>[-cluster &lt;cluster-name&gt;]</code>	클러스터에 등록된 external resource를 확인하기 위한 클러스터 이름이다.
<code>[-server &lt;server-name&gt;]</code>	서버에 등록된 external resource를 확인하기 위한 서버 이름이다.

- 예제

```
[MASTER]domain1.adminServer>list-external-resources
List of External Resources
=====
+-----+-----+-----+
| Export Name | Resource Class | Properties |
+-----+-----+-----+
```

```
| test/ext | test.ext.TestResourceBootstrapper | [] |
+-----+-----+-----+
=====
```

#### 4.2.4.22. list-domains

현재 구성된 도메인의 목록을 조회하는 명령어이다.

- alias

listdomains

- 사용법

```
list-domains
```

- 예제

```
offline>list-domains
List Domains
=====
+-----+-----+
|                               Domains                               |
+-----+-----+
| domain1                               |
+-----+-----+
=====
```

- 참고

본 명령어는 서버에 접속하지 않은 상태에서도 사용할 수 있다.

#### 4.2.4.23. list-servers

현재 도메인에 존재하는 서버의 설정 정보를 확인한다. 상태 정보 확인을 위해서는 [server-info](#) 명령어를 사용한다.

- 사용법

```
list-servers [<server-name>]
```

- 파라미터

파라미터	설명
[<server-name>]	서버의 이름이다.

- 예제

```
[MASTER]domain1.adminServer>list-servers
```

```
List of Editable Servers
```

Serv er	Base Listen Address /Port	Node	JVM Configs	Action On Resource Leak	Stdout to Raw Format	MEJB	Clas s FTP	Server Log Home Directory	Type
adminServer	0.0.0.0 / 9736	node1	-Xmx1024m -XX:MaxMetaspaceSize=512m	Warning	true	false	true	none	server
server1	192.168.0.124 / 9836	node1	-Xmx1024m -XX:MaxMetaspaceSize=512m	Warning	true	false	false	none	server
server2	192.168.0.124 / 9936	node1	-Xmx1024m -XX:MaxMetaspaceSize=512m	Warning	true	false	false	none	server
dynamic_server_1	0.0.0.0 / 9737	node1	-Xmx1024m	Warning	true	false	false	none	dynamic(t1)
dynamic_server_1	0.0.0.0 / 9837	node1	-Xmx1024m	Warning	true	false	false	none	dynamic(t1)

#### 4.2.4.24. modify-cluster

동적으로 특정 클러스터의 설정을 변경한다.

- alias

```
modifycluster, clusterconf, clusterconfig
```

- 사용법

```
modify-cluster <cluster-name>  
[-m,--mejb <true | false>]
```

```
[-c,--classFtp <true | false>]
[-f,--forceLock]
[-detail]
```

- 파라미터

파라미터	설명
<cluster-name>	클러스터의 이름이다.
[-m,--mejb <true   false>]	<b>[동적 변경]</b> MEJB를 사용할지 여부를 설정한다. (true   false)
[-c,--classFtp <true   false>]	<b>[동적 변경]</b> class FTP 서비스를 사용할지 여부를 설정한다. (true   false)
[-f,--forceLock]	설정 변경을 강제로 진행하여 반영한다.
[-detail]	동적 변경 결과에 대한 상세한 정보가 출력된다.

- 예제

```
[MASTER]domain1.adminServer>modify-cluster cluster1 -m true
Successfully performed the MODIFY operation for cluster (cluster1).
Check the results using "list-clusters cluster1 or modify-cluster cluster1"
```

#### 4.2.4.25. modify-server

동적으로 특정 서버의 설정을 변경한다.

- alias

serverconf, serverconfig, modify-server-template

- 사용법

```
modify-server <server-name>
    [-removeLogdir,--removeLogHomeDirectory]
    [-node,--nodeName <node-name>]
    [-jvm,--jvmOptions <jvm-options>]
    [-logdir,--logHomeDirectory <server-log-home-directory>]
    [-a,--actionOnResourceLeak <action-on-resource-leak>]
    [-l,--logStdoutToRawFormat <true | false>]
    [-m,--mejb <true | false>]
    [-c,--classFtp <true | false>]
    [-f,--forceLock]
    [-detail]
```

- 파라미터

파라미터	설명
<server-name>	서버의 이름이다.
[-removeLogdir,--removeLogHomeDirectory]	서버에서 생성되는 로그들이 공통적으로 들어갈 로그 디렉터리를 삭제할지의 여부이다.
[-node,--nodeName <node-name>]	서버가 실제로 동작하게 될 머신을 의미하는 노드의 이름이다.
[-jvm,--jvmOptions <jvm-options>]	서버에 적용하고 하는 JVM 옵션이다. 공백이 존재하는 경우 큰따옴표(" ")로 묶어서 입력한다.
[-logdir,--logHomeDirectory <server-log-home-directory>]	서버에서 생성되는 로그들이 공통적으로 들어갈 로그 디렉터리를 의미한다.
[-a,--actionOnResourceLeak <action-on-resource-leak>]	리소스에 leak이 발생할 경우 취할 액션을 설정한다. 다음 중에 하나를 설정한다. <ul style="list-style-type: none"> <li>• AutoClose</li> <li>• NoAction</li> <li>• Warning</li> </ul>
[-l,--logStdoutToRawFormat <true   false>]	<b>[동적 변경]</b> stdout을 그대로 출력할 것인지 JEUS 로거 포맷으로 출력할 것인지 설정한다. (true   false)
[-m,--mejb <true   false>]	<b>[동적 변경]</b> MEJB를 사용할지 여부를 설정한다. (true   false)
[-c,--classFtp <true   false>]	<b>[동적 변경]</b> class FTP 서비스를 사용할지 여부를 설정한다. (true   false)
[-f,--forceLock]	설정 변경을 강제로 진행하여 반영한다.
[-detail]	동적 변경 결과에 대한 상세한 정보가 출력된다.

#### • 예제

```
[MASTER]domain1.adminServer>modify-server server1 -a AutoClose -l false
Successfully performed the MODIFY operation for server (server1).
Check the results using "list-servers server1 or modify-server server1"
```

#### 4.2.4.26. modify-dynamic-servers

Dynamic servers 설정으로 생성한 클러스터의 설정을 변경한다.

#### • alias

modifydservers

#### • 사용법

```

modify-dynamic-servers <cluster-name>
    [-count <the-number-of-dynamic-servers>]
    [-nodes,--nodeNames <node-names>]
    [-prefix,--serverNamePrefix <server-name-prefix>]
    [-template,--serverTemplateName <server-template-name>]
    [-step,--listenPortStep <listen-port-step>]
    [-f,--forceLock]
    [-detail]

```

#### • 파라미터

파라미터	설명
<cluster-name>	동적 설정 서버들의 설정을 변경할 클러스터 이름을 지정한다.
[-count <the-number-of-dynamic-servers>]	클러스터에 추가할 서버의 수를 변경한다.
[-nodes,--nodeNames <node-names>]	서버가 위치할 노드를 변경한다.  coma(,)를 구분자로 하여 여러 개를 설정할 수 있다. 모든 노드를 대상으로 하는 경우 -nodes 옵션 뒤에 아무 인자도 넘겨주지 않으면 된다.
[-prefix,--serverNamePrefix <server-name-prefix>]	서버의 이름 앞에 붙을 prefix를 변경한다.  Prefix가 설정되지 않은 경우에는 클러스터 이름 뒤에 'server'를 붙인다.
[-template,--serverTemplateName <server-template-name>]	동적(Dynamic) 서버들에게 공통적으로 적용될 설정 템플릿을 지정한다.
[-step,--listenPortStep <listen-port-step>]	같은 노드에 생성된 서버들의 포트 번호를 몇 씩 증가시킬지 변경한다.
[-f,--forceLock]	설정 변경을 강제로 진행하여 반영한다.
[-detail]	동적 변경 결과에 대한 상세한 정보가 출력된다.

#### • 예제

```

[MASTER]domain1.adminServer>modify-dynamic-servers cluster1 -count 3
Successfully performed the MODIFY operation for Dynamic servers configuration in the cluster (cluster1).
Check the results using "modify-dynamic-servers or show-dynamic-servers".

```

### 4.2.4.27. pack-domain

도메인 백업을 이용한 장애 극복을 위해 특정 도메인 폴더를 압축하는 명령어이다.



- alias

packdomain

- 사용법

```
pack-domain <domain-name>
          [-p <dir-path>]
          [-node <node-name>]
```

- 파라미터

파라미터	설명
<domain-name>	도메인의 이름이다.
[-p <dir-path>]	압축 파일의 경로이다.  (기본값: JEUS_HOME/backups/<domain-name>_packed.zip)
[-node <node-name>]	원격 노드에 압축 파일을 저장하기 위한 노드 이름이다.  이 옵션을 사용하기 위해서는 jeusadmin과 MASTER가 연결되어 있어야 하고, 원격 노드에 노드 매니저가 켜져 있어야 한다.

- 예제

```
offline>pack-domain domain1
Packing the domain [domain1] configuration completed successfully at the path
[JEUS_HOME/backups/domain1_packed.zip].
```

#### 4.2.4.28. remove-cluster

동적으로 클러스터를 삭제한다.

- 사용법

```
remove-cluster [<cluster-name>]
              [-f,--forceLock]
              [-detail]
```

- 파라미터

파라미터	설명
[<cluster-name>]	클러스터의 이름이다.
[-f,--forceLock]	설정 변경을 강제로 진행하여 반영한다.

파라미터	설명
[-detail]	동적 변경 결과에 대한 상세한 정보가 출력된다.

- 예제

- 현재 존재하는 클러스터 확인

```
[MASTER]domain1.adminServer>list-clusters
List of Clusters
=====
+-----+-----+-----+-----+-----+
| Cluster | Server List | MEJB | Class FTP | Running |
+-----+-----+-----+-----+-----+
| cluster1 | server1, server2 | false | false | true |
+-----+-----+-----+-----+-----+
| cluster2 | server3, server4 | true | true | true |
+-----+-----+-----+-----+-----+
=====
```

- 현재 존재하는 클러스터 리스트에서 삭제할 클러스터 이름 확인 후 클러스터 삭제

```
[MASTER]domain1.adminServer>remove-cluster cluster2
Successfully performed the REMOVE operation for cluster (cluster1).
Check the results using "list-clusters or remove-cluster"
```

#### 4.2.4.29. remove-custom-resource

동적으로 custom resource를 삭제한다.

- alias

removecr, rmcr

- 사용법

```
remove-custom-resource [<export-name>]
                        [-f,--forceLock]
                        [-detail]
```

- 파라미터

파라미터	설명
[<export-name>]	제거할 custom resource의 JNDI 바인드 이름이다.
[-f,--forceLock]	설정 변경을 강제로 진행하여 반영한다.
[-detail]	동적 변경 결과에 대한 상세한 정보가 출력된다.

- 예제

```
[MASTER]domain1.adminServer>remove-custom-resource custom/dog
Successfully performed the REMOVE operation for A custom resource.
Check the results using "list-custom-resources or remove-custom-resource"
```

#### 4.2.4.30. remove-custom-resource-from-clusters

클러스터에 등록된 custom resource를 동적으로 삭제한다.

- alias

remove-cr-from-clusters

- 사용법

```
remove-custom-resource-from-clusters <export-name>
                                -clusters <cluster-list>
                                [-f,--forceLock]
                                [-detail]
```

- 파라미터

파라미터	설명
<export-name>	클러스터에서 제거할 custom resource의 JNDI 바인드 이름이다.
-cluster <cluster-list>	custom resource를 제거할 cluster list를 설정한다.
[-f,--forceLock]	설정 변경을 강제로 진행하여 반영한다.
[-detail]	동적 변경 결과에 대한 상세한 정보가 출력된다.

- 예제

```
[MASTER]domain1.adminServer>remove-custom-resource-from-clusters custom/dog -clusters cluster1
Successfully performed the REMOVE operation for A custom resource.
Check the results using "list-custom-resources"
```

#### 4.2.4.31. remove-data-sources-from-cluster

클러스터에 등록된 데이터소스들을 동적으로 제거한다.

- alias

rmdatafromcluster

- 사용법

```
remove-data-sources-from-cluster -cluster <cluster-name>
                                [-ids <data-source-id-list> | -all]
                                [-f,--forceLock]
                                [-detail]
```

- 파라미터

파라미터	설명
-cluster <cluster-name>	데이터소스를 제거할 클러스터의 이름이다.
[-ids <data-source-id-list> ]	클러스터로부터 제거할 데이터소스들의 ID 목록이다.  한 개 이상인 경우 콤마(,)로 구분한다.
[ -all]	클러스터에 등록된 모든 데이터소스를 제거한다.
[-f,--forceLock]	설정 변경을 강제로 진행하여 반영한다.
[-detail]	동적 변경 결과에 대한 상세한 정보가 출력된다.

- 예제

```
[MASTER]domain1.adminServer>remove-data-sources-from-cluster -cluster cluster1 -all
Successfully performed the REMOVE operation for data sources from the cluster [cluster1].
Check the results using "remove-data-sources-from-cluster -cluster cluster1"
```

#### 4.2.4.32. remove-domain-log-config

도메인 내 log 파일들의 소유자 및 권한 설정을 삭제한다.

해당 설정을 삭제하더라도 기존 log 파일들의 소유자 및 권한 설정이 변경되지는 않는다.

- alias

rmdomainlog, rmdomainlogcon, rmdomainlogconfig

- 사용법

```
remove-domain-log-config [-perm,--permission]
                        [-own,--owner]
```

- 파라미터

파라미터	설명
[-perm,--permission]	도메인 내 log 파일들의 권한 설정을 삭제한다.
[-own,--owner]	도메인 내 log 파일들의 소유자 설정을 삭제한다.

- 예제

```
[MASTER]domain1.adminServer>remove-domain-log-config -own
Successfully performed the REMOVE operation for Domain Log file configuration for domain1.
Check the results using "show-domain-log-config".
```

#### 4.2.4.33. remove-external-resource

동적으로 external resource를 삭제한다.

- alias

removecr, rmcr

- 사용법

```
remove-external-resource [<export-name>]
                        [-f,--forceLock]
                        [-detail]
```

- 파라미터

파라미터	설명
[<export-name>]	제거할 external resource의 이름이다.
[-f,--forceLock]	설정 변경을 강제로 진행하여 반영한다.
[-detail]	동적 변경 결과에 대한 상세한 정보가 출력된다.

- 예제

```
[MASTER]domain1.adminServer>remove-external-resource test/ext
Successfully performed the REMOVE operation for A external resource.
Check the results using "list-external-resources or remove-external-resource"
```

#### 4.2.4.34. remove-external-resource-from-clusters

클러스터에 등록된 external resource를 동적으로 삭제한다.

- alias

remove-er-from-clusters

- 사용법

```
remove-external-resource-from-clusters <export-name>
```

```
-clusters <cluster-list>
[-f,--forceLock]
[-detail]
```

- 파라미터

파라미터	설명
<export-name>	클러스터에서 제거할 external resource의 이름이다.
-clusters <cluster-list>	external resource를 제거할 cluster list를 설정한다.
[-f,--forceLock]	설정 변경을 강제로 진행하여 반영한다.
[-detail]	동적 변경 결과에 대한 상세한 정보가 출력된다.

- 예제

```
[MASTER]domain1.adminServer>remove-external-resource-from-clusters test/ext -clusters cluster1
Successfully performed the REMOVE operation for A external resource.
Check the results using "list-external-resources"
```

#### 4.2.4.35. remove-server

동적으로 특정 서버를 삭제한다.

- 사용법

```
remove-server [<server-name>]
              [-f,--forceLock]
              [-detail]
```

- 파라미터

파라미터	설명
[<server-name>]	서버의 이름이다.
[-f,--forceLock]	설정 변경을 강제로 진행하여 반영한다.
[-detail]	동적 변경 결과에 대한 상세한 정보가 출력된다.

- 예제

- 현재 존재하는 서버 리스트 확인

```
[MASTER]domain1.adminServer>server-info
Information about Domain (domain1)
=====
+-----+-----+-----+-----+-----+-----+-----+-----+
| Server | Status | Node | PID | Clu | Latest | Need | Listen | Running |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

		Name		ster	Start Time / Shutdown Time	to Restart	Ports	Engines
adminS erver (*)	RUNNIN G(00:46 :06)	nod e1	237 60	N/A	2021-03- 25 (목) 오후 01:13:48 KST	false	base-0.0. 0.0:9736 http-serv er-0.0.0.0 :8088	jms, web, ejb
server 1	RUNNIN G(00:02 :45)	nod e1	132 75	N/A	2021-03- 25 (목) 오후 01:13:48 KST	false	base-0.0. 0.0:9836	jms, web, ejb
server 2	SHUTDO WN	nod e1	N/A	N/A	N/A	N/A	N/A	N/A

- 삭제할 서버가 종료되어 있는지 확인 후 삭제

```
[MASTER]domain1.adminServer>remove-server server2
Successfully performed the REMOVE operation for server (server1).
Check the results using "list-servers or remove-server"
```

#### 4.2.4.36. remove-servers-from-cluster

동적으로 특정 클러스터에서 특정 서버를 삭제한다.

- 사용법

```
remove-servers-from-cluster <cluster-name>
    [-servers <server-lists>]
    [-f,--forceLock]
    [-detail]
```

- 파라미터

파라미터	설명
<cluster-name>	클러스터의 이름이다.
[-servers <server-lists>]	이 클러스터에서 제거하려는 서버의 이름이다. 서버가 한 개 이상인 경우는 콤마(,)로 구분한다.
[-f,--forceLock]	설정 변경을 강제로 진행하여 반영한다.
[-detail]	동적 변경 결과에 대한 상세한 정보가 출력된다.

- 예제

```
[MASTER]domain1.adminServer>remove-servers-from-cluster cluster1 -servers server1
Successfully performed the REMOVE operation for The server list for cluster(cluster1)..
Check the results using "list-clusters cluster1 or remove-servers-from-cluster cluster1"
```

#### 4.2.4.37. set-domain-backup

도메인 백업 기능에 대한 정책을 설정한다.

- alias

domain-backup

- 사용법

```
set-domain-backup [-backupOnBoot <backup-on-boot>]
                  [-backupDir <backup-directory>]
                  [-nodeName <nodeName>]
                  [-f,--forceLock]
                  [-detail]
```

- 파라미터

파라미터	설명
[-backupOnBoot <backup-on-boot>]	JEUS 기동 시 백업을 수행할지 여부를 true/false로 설정한다.
[-backupDir <backup-directory>]	백업 파일을 저장할 디렉터리를 지정한다.
[-nodeName <node-name>]	백업 파일이 위치할 노드를 설정한다.  이 값이 설정된 경우에는 backupDir이 해당 노드 내의 디렉터리를 의미하게 된다. 만약 설정된 노드로의 전송에 실패한 경우에는 로컬 노드의 domains home 디렉터리에 위치하게 된다.
[-f,--forceLock]	설정 변경을 강제로 진행하여 반영한다.
[-detail]	동적 변경 결과에 대한 상세한 정보가 출력된다.

- 예제

```
[MASTER]domain1.adminServer>set-domain-backup -backupOnBoot true
Successfully performed the MODIFY operation for Domain Backup Policy.
Check the results using "set-domain-backup".
```



#### 4.2.4.38. set-domain-log-config

도메인 내 log 파일들의 소유자 및 권한을 설정한다. 도메인에 속해 있는 서버들의 로거 설정 중 로그 파일 소유자 및 권한 설정이 없을 경우 이 설정을 따르게 된다.

만약 로거의 소유자 또는 권한 설정이 따로 존재할 경우에는 이 설정보다는 해당 설정을 따르게 된다.

- alias

setdomainlog, setdomainlogcon, setdomainlogconfig

- 사용법

```
set-domain-log-config [-perm,--permission <permission>]
                      [-own,--owner <chown>]
```

- 파라미터

파라미터	설명
[-perm,--permission <permission>]	도메인 내 log 파일의 권한을 설정한다. 형식은 <b>rwxrw-r--</b> 와 같이 입력하면 된다. POSIX 호환 OS에서만 사용가능한 설정이다.
[-own,--owner <chown>]	도메인 내 log 파일의 소유자를 설정한다.  형식은 ' <b>소유자명,그룹</b> '으로 입력한다. 만약 소유자명만 지정할 경우에는 ' <b>소유자명</b> '만 입력한다. 그룹만 지정할 경우 ' <b>그룹</b> '으로 입력한다.

- 예제

```
[MASTER]domain1.adminServer>set-domain-log-config -own user,tmax
Successfully performed the MODIFY operation for Domain Log file configuration for domain1, but
all changes were non-dynamic. They will be applied after restarting.
Check the results using "show-domain-log-config or set-domain-log-config".
[MASTER]domain1.adminServer>set-domain-log-config -own ,tmax -perm rwxrwxrwx
Successfully performed the MODIFY operation for Domain Log file configuration for domain1, but
all changes were non-dynamic. They will be applied after restarting.
Check the results using "show-domain-log-config or set-domain-log-config".
```

#### 4.2.4.39. show-domain-log-config

도메인 내 log 파일들의 소유자 및 권한을 설정을 보여준다.

- alias

showdomainlog, showdomainlogcon, showdomainlogconfig

- 사용법

```
show-domain-log-config
```

- 예제

```
[MASTER]domain1.adminServer>show-domain-log-config
Shows the current configuration.
Domain Log file configuration for domain1
=====
+-----+
| Domain Log file permission          | (not set) |
| Domain Log file owner              | (not set) |
+-----+
=====
```

#### 4.2.4.40. show-dynamic-servers

동적 서버 생성으로 만들어진 클러스터의 설정 정보를 보여준다.

- alias

showdservers

- 사용법

```
show-dynamic-servers <cluster-name>
```

- 파라미터

파라미터	설명
<cluster-name>	동적 설정 서버의 정보를 볼 클러스터 이름을 지정한다.

- 예제

```
[MASTER]domain1.adminServer>show-dynamic-servers cluster1
Shows the current configuration.
Dynamic servers configuration in the cluster (cluster1)
=====
+-----+
| Server Count          | 2      |
| Nodes                 | node1  |
| Base Listen Port      | 9736   |
| HTTP Listen Port      | 8088   |
| JMS Listen Port       | 9741   |
| Server Template Name  | temp1  |
| Listen Port Step      | 100    |
+-----+
=====
```

#### 4.2.4.41. show-server-template

설정된 서버 템플릿의 리스트를 조회한다.

- alias

show-template

- 사용법

```
show-server-template [<server-template-name>]
```

- 파라미터

파라미터	설명
[<server-template-name>]	설정된 템플릿의 정보를 조회한다.

- 예제

```
[MASTER]domain1.adminServer>show-server-template temp1
Shows the current configuration.
server template (temp1)
=====
+-----+-----+
| Action On Resource Leak          | WARNING |
| Stdout to Raw Format             | true    |
| MEJB                             | false   |
| Class FTP                       | false   |
| Server Log Home Directory       | none    |
+-----+-----+
=====
```

#### 4.2.4.42. show-system-clustering-framework

System Clustering Framework 기능과 관련된 설정을 조회한다.

- alias

show-scf

- 사용법

```
show-system-clustering-framework
```

- 예제

```
[MASTER]domain1.adminServer>show-system-clustering-framework
```

```

=====
System clustering framework configurations
+-----+-----+
| transport-address | 224.0.0.1 |
| transport-port    | 12488    |
| transport-type    | HYBRID   |
| leader-discovery-timeout | 3000    |
| failure-detection-timeout | 3000    |
| max-thread-pool-size | 10       |
| min-thread-pool-size | 0        |
+-----+-----+
=====

```

#### 4.2.4.43. unpack-domain

서버에 장애가 생겼을 때, pack-domain에 의해 압축된 파일을 해제하는 명령어이다. 해당 도메인 설정에 JEUS Master Server의 동작 주소와 동작 포트 번호가 설정된 경우와 JEUS Master Server의 노드이름이 노드 설정에 설정된 경우 변경 여부를 설정할 수 있다.

- alias

unpackdomain

- 사용법

```

unpack-domain -p <file-path> | <domain-name>
               [-delete]
               [-overwrite]
               [-masterurl <master-url>]
               [-nodename <node-name>]
               [-nodeurl <node-url>]

```

- 파라미터

파라미터	설명
-p <file-path>	압축을 풀 파일의 경로이다.  (예: JEUS_HOME/<domain-name>_packed.zip)
<domain-name>	압축을 풀 파일의 대상 도메인을 지정한다.  압축을 풀 파일은 반드시 JEUS_HOME/backups 디렉터리 안에 있어야 한다.
[-delete]	Unpack을 하려는 대상 도메인이 DOMAIN_HOME에 이미 존재하면 기존 도메인을 삭제하고 unpack을 진행하도록 한다.
[-overwrite]	Unpack을 하려는 대상 도메인이 DOMAIN_HOME에 이미 존재하면 unpack하려는 도메인이 기존 도메인을 덮어쓰도록 한다.

파라미터	설명
[-masterurl <master-url>]	Unpack을 진행하는 domain.xml에 설정될 JEUS Master Server의 동작 주소를 설정한다.
[-nodename <node-name>]	Unpack을 진행하는 domain.xml에 설정될 JEUS Master Server의 노드 이름을 설정한다.
[-nodeurl <node-url>]	Unpack을 진행하는 nodes.xml에 설정될 Master Server의 노드 주소를 설정한다.

- 예제

```
offline>unpack-domain domain1
The JEUS Master Server listener address is already set to [0.0.0.0]. Do you want to change it?
(y/n): y
Enter the JEUS Master Server base listener address: 192.168.34.56
The JEUS Master Server listener port is already set to [9736]. Do you want to change it? (y/n): n
The JEUS Master Server nodename is already set to [node1]. Do you want to change it? (y/n): y
Enter the JEUS Master Server nodename: node1
The nodemanager of JEUS Master Server listener address is already set to [0.0.0.0]. Do you want
to change it? (y/n): n
The nodemanager of JEUS Master Server listener port is already set to [7730]. Do you want to
change it? (y/n): n
Unpacking the domain [domain1] configuration completed successfully.
```

## 4.2.5. Thread Management 관련 명령어

Thread Management 관련 명령어의 목록은 다음과 같다.

명령어	설명
<a href="#">interrupt-thread</a>	서블릿 또는 EJB RMI Thread에 인터럽트 시그널을 전송한다.
<a href="#">modify-service-thread-pool</a>	동적으로 특정 서버의 특정 서비스의 전용 Thread Pool을 변경한다.
<a href="#">modify-system-thread-pool</a>	동적으로 특정 서버의 System Thread Pool을 변경한다.
<a href="#">print-stack-trace</a>	Servlet Thread 또는 EJB RMI Thread의 Stack Trace를 출력한다.
<a href="#">show-service-thread-pool</a>	System Thread Pool에서 특정 서비스에 할당된 Thread의 정보를 조회한다.
<a href="#">show-system-thread-pool</a>	System Thread Pool의 정보를 조회한다.
<a href="#">thread-info</a>	요청을 처리하는 Servlet Thread와 EJB RMI Thread의 정보를 조회한다.

### 4.2.5.1. interrupt-thread

서블릿 또는 EJB RMI Thread에 인터럽트 시그널을 전송한다. Thread에 인터럽트 시그널을 보냈을 때의 결과에 대한 자세한 설명은 JEUS Server 안내서의 "Thread 모니터링 및 제어"를 참고한다.



이 기능은 인터럽트 신호만 보내는 것이므로 그 Thread가 반드시 동작을 멈춘다고 보장할 수

없다.

- alias

interruptthread, interrupt, intthread

- 사용법

```
interrupt-thread -server <server-name>  
                 -ctx <web-context-name> | -wc <Web Connection Name> | <thread-id>  
                 [-svg <Service Group Name>]
```

- 파라미터

파라미터	설명
-server <server-name>	서버의 이름이다.
-ctx <web-context-name>   -wc <web-connection-name>   <thread-id>	<ul style="list-style-type: none"><li>• -ctx &lt;web-context-name&gt; : interrupt할 thread pool이 Context 레벨의 thread pool일 때 context name을 인자로 설정한다.</li><li>• -wc : interrupt할 thread pool이 Web-Connection 레벨의 thread pool일 때 Web-Connection name을 인자로 설정한다.</li><li>• &lt;thread-id&gt; : 시그널을 전송할 Thread의 ID이다.</li></ul>
[-svg <Service Group Name>]	Context 레벨의 Thread Pool 이름이다. -ctx 옵션이 설정되었을 경우만 옵션을 줄 수 있다. -svg 옵션이 없을 경우 모든 Context 레벨의 Thread Pool을 interrupt한다.

- 예제

```
[MASTER]domain1.adminServer>interrupt-thread -server server1 50  
Sent an interrupt hint signal to the thread [tid=50] on the server server1.
```

#### 4.2.5.2. modify-service-thread-pool

동적으로 서비스의 전용 Thread Pool을 변경한다. 현재 통합 설정을 사용하고 있는 Naming Server, Scheduler, Transaction Thread Pool만 이 명령어로 변경 가능하다. 변경할 설정을 입력하지 않으면 현재 설정을 조회한다.

- alias

modify-svctp, modifysvctp, svctpconfig, svctpconf

- 사용법

```
modify-service-thread-pool -service <service name>
```

```

<server-name>
[-min <min>]
[-max <max>]
[-k,--keepalivetime <keep-alive-time>]
[-q,--queuesize <queue-size>]
[-m,--maxstuckthreadtime <max-stuck-thread-time>]
[-a,--actiononstuckthread <action-on-stuck-thread>]
[-stcp,--stuckthreadcheckperiod <stuck-thread-check-period>]
[-f,--forceLock]
[-detail]

```

#### • 파라미터

파라미터	설명
-service <service name>	System Thread Pool을 사용하는 서비스의 이름이다.  <service name>에 다음 중에 하나를 선택한다. <ul style="list-style-type: none"> <li>• transaction</li> <li>• scheduler</li> <li>• namingserver</li> </ul>
<server-name>	서버의 이름이다.
[-min <min>]	<b>[동적 변경]</b> Thread Pool의 최소 Thread 개수이다.
[-max <max>]	<b>[동적 변경]</b> Thread Pool의 최대 Thread 개수이다.
[-k,--keepalivetime <keep-alive-time>]	<b>[동적 변경]</b> 최소 Thread 개수를 초과하는 경우 입력한 시간 이상 idle 상태인 Thread는 삭제한다.
[-q,--queuesize <queue-size>]	실행 대기하는 worker의 최대 개수이다.
[-m,--maxstuckthreadtime <max-stuck-thread-time>]	<b>[동적 변경]</b> 입력한 시간 이상으로 Thread가 수행되고 있으면 STUCK 상태라고 판단한다.
[-a,--actiononstuckthread <action-on-stuck-thread>]	<b>[동적 변경]</b> STUCK 상태일 경우의 처리 방법이다.  <action-on-stuck-thread>에 다음 중에 하나를 설정한다. <ul style="list-style-type: none"> <li>• IgnoreAndReplace</li> <li>• Interrupt</li> <li>• None</li> </ul>
[-stcp,--stuckthreadcheckperiod <stuck-thread-check-period>]	<b>[동적 변경]</b> Thread들이 STUCK 상태임을 확인하는 주기를 설정한다. (단위: ms)
[-f,--forceLock]	설정 변경을 강제로 진행하여 반영한다.
[-detail]	동적 변경 결과에 대한 상세한 정보가 출력된다.

#### • 예제

```
88% [MASTER]domain1.adminServer>modify-service-thread-pool server1 -service transaction -min 10
-max 20
Successfully performed the MODIFY operation for The transaction thread pool of the
server (server1)., but all changes were non-dynamic. They will be applied after
restarting.
Check the results using "show-service-thread-pool server1 -service transaction or
modify-service-thread-pool server1 -service transaction"
```

#### 4.2.5.3. modify-system-thread-pool

동적으로 System Thread Pool을 변경한다. System Thread Pool을 사용하고 있는 서비스들의 할당된 Thread 개수도 설정할 수 있다.

- alias

modify-systp, modifysystp, systpconfig, systpconf

- 사용법

```
modify-system-thread-pool <server-name>
    [-min <min>]
    [-max <max>]
    [-k,--keepalivetime <keep-alive-time>]
    [-q,--queuesize <queue-size>]
    [-m,--maxstuckthreadtime <max-stuck-thread-time>]
    [-a,--actiononstuckthread <action-on-stuck-thread>]
    [-stp,--stuckthreadcheckperiod <stuck-thread-check-period>]
    [-service <service-name>]
    [-r,--reservednum <reserved-number>]
    [-f,--forceLock]
    [-detail]
```

- 파라미터

파라미터	설명
<server-name>	서버의 이름이다.
[-min <min>]	<b>[동적 변경]</b> Thread Pool의 최소 Thread 개수이다.
[-max <max>]	<b>[동적 변경]</b> Thread Pool의 최대 Thread 개수이다.
[-k,--keepalivetime <keep-alive-time>]	<b>[동적 변경]</b> 최소 Thread 개수를 초과하는 경우 입력한 시간 이상 idle 상태인 Thread는 삭제한다.
[-q,--queuesize <queue-size>]	실행 대기하는 worker의 최대 개수이다.
[-m,--maxstuckthreadtime <max-stuck-thread-time>]	<b>[동적 변경]</b> 입력한 시간 이상으로 Thread가 수행되고 있으면 STUCK 상태라고 판단한다.



파라미터	설명
<code>[-a,--actiononstuckthread &lt;action-on-stuck-thread&gt;]</code>	<b>[동적 변경]</b> STUCK 상태일 경우의 처리 방법이다. <action-on-stuck-thread>에 다음 중에 하나를 선택한다. <ul style="list-style-type: none"> <li>• IgnoreAndReplace</li> <li>• Interrupt</li> <li>• None</li> </ul>
<code>[-stcp,--stuckthreadcheckperiod &lt;stuck-thread-check-period&gt;]</code>	<b>[동적 변경]</b> Thread들이 STUCK 상태임을 확인하는 주기를 설정한다. (단위: ms)
<code>[-service &lt;service-name&gt;]</code>	System Thread Pool을 사용하는 서비스의 이름이다. <service name>에 다음 중에 하나를 선택한다. <ul style="list-style-type: none"> <li>• transaction</li> <li>• scheduler</li> <li>• namingserver</li> </ul>
<code>[-r,--reservednum &lt;reserved-number&gt;]</code>	<b>[동적 변경]</b> System Thread Pool을 사용하는 서비스에 보장되는 최소 Thread 개수이다.
<code>[-f,--forceLock]</code>	설정 변경을 강제로 진행하여 반영한다.
<code>[-detail]</code>	동적 변경 결과에 대한 상세한 정보가 출력된다.

#### • 예제

```

85% [MASTER]domain1.adminServer>modify-system-thread-pool server1 -min 10 -max 60
-service transaction -reservednum 20 -detail
Successfully performed the MODIFY operation for the system thread pool of the server (server1).
=====
+-----+-----+-----+-----+
| servers.server.{? name == 'server1' }.systemThreadPool | MODIFY| ACTIVATED|
| servers.server.{? name == 'server1' }.tmConfig          | MODIFY| ACTIVATED|
+-----+-----+-----+-----+
=====
Check the results using "modify-system-thread-pool server1 or show-system-thread -pool server1"
-----details-----
servers.server.{? name == 'server1' }.tmConfig : ACTIVATED
servers.server.{? name == 'server1' }.tmConfig.pooling : ACTIVATED
servers.server.{? name == 'server1' }.tmConfig.pooling.shared : ACTIVATED
servers.server.{? name == 'server1' }.tmConfig.pooling.shared.reservedThreadNum : ACTIVATED
previous value : 0, edited value : 20, result value : 20
servers.server.{? name == 'server1' }.systemThreadPool : ACTIVATED
servers.server.{? name == 'server1' }.systemThreadPool.min : ACTIVATED
previous value : 0, edited value : 10, result value : 10
servers.server.{? name == 'server1' }.systemThreadPool.max : ACTIVATED
previous value : 100, edited value : 60, result value : 60

```

#### 4.2.5.4. print-stack-trace

Servlet Thread 또는 EJB RMI Thread의 Stack Trace를 출력한다.

- alias

stacktrace, strace

- 사용법

```
print-stack-trace -server <server-name>
                  -ctx <web-context-name> | -wc <Web Connection Name> | <thread-id>
                  [-svg <Service Group Name>]
```

- 파라미터

파라미터	설명
-server <server-name>	서버의 이름이다.
-ctx <web-context-name>   -wc <web-connection-name>   <thread-id>	<ul style="list-style-type: none"><li>• -ctx &lt;web-context-name&gt; : Stack Trace를 보여 줄 thread pool이 Context 레벨의 thread pool일 때 context name을 인자로 설정한다.</li><li>• -wc : Stack Trace를 보여 줄 thread pool이 Web-Connection 레벨의 thread pool일 때 Web-Connection name을 인자로 설정한다.</li><li>• &lt;thread-id&gt; : Thread의 ID이다.</li></ul>
[-svg <Service Group Name>]	Context 레벨의 Thread Pool 이름이다. -ctx 옵션이 설정되었을 경우만 옵션을 줄 수 있다. -svg 옵션이 없을 경우 모든 Context 레벨의 Thread Pool의 Stack Thrace를 보여준다.

- 예제

```
[MASTER]domain1.adminServer>print-stack-trace -server server1 50
servlet thread [tid=50] Stack trace of http1-1 [server1-50] tid=50 java.lang.Thread.State:
WAITING
    at sun.misc.Unsafe.park(Native Method)
    at java.util.concurrent.locks.LockSupport.park(LockSupport.java:156)
    at
java.util.concurrent.locks.AbstractQueuedSynchronizer$ConditionObject.await(AbstractQueuedSynchro
nizer.java:1987)
    at java.util.concurrent.LinkedBlockingQueue.take(LinkedBlockingQueue.java:399)
    at jeus.util.ThreadPoolExecutor.getTask(ThreadPoolExecutor.java:1291)
    at jeus.servlet.engine.WebThreadPoolExecutor.getTask(WebThreadPoolExecutor.java:68)
    at jeus.util.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:1215)
    at
jeus.servlet.engine.WebThreadPoolExecutor$WebRequestWorker.run(WebThreadPoolExecutor.java:332)
    at java.lang.Thread.run(Thread.java:662)
```

#### 4.2.5.5. show-service-thread-pool

특정 서버의 System Thread Pool을 정보를 확인한다.

- alias

show-svctp, showsvctp

- 사용법

```
show-service-thread-pool <server-name>
                        -service <service-name>
```

- 파라미터

파라미터	설명
<server-name>	서버의 이름이다.
-service <service-name>	서비스의 이름이다.  <service-name>에 다음 중에 하나를 선택한다. <ul style="list-style-type: none"><li>• transaction</li><li>• scheduler</li><li>• namingserver</li></ul>

- 예제

```
[MASTER]domain1.adminServer>show-service-thread-pool server1 -service transaction
Shows the current configuration.
=====
+-----+-----+
| Min           | 10    |
| Max           | 20    |
| Keep-Alive Time | 60000 |
| Queue Size    | 4096  |
| Max Stuck Thread Time | 3600000 |
| Action On Stuck Thread | NONE  |
| Stuck Thread Check Period | 300000 |
+-----+-----+
=====
```

#### 4.2.5.6. show-system-thread-pool

특정 서버의 System Thread Pool 정보를 확인한다.

- alias

show-systp, showsystp

- 사용법

```
show-system-thread-pool <server-name>
                        [-service <service-name>]
```

- 파라미터

파라미터	설명
<server-name>	서버의 이름이다.
[-service <service-name>]	서비스의 이름이다.  <service-name>에 다음 중에 하나를 선택한다. <ul style="list-style-type: none"><li>• transaction</li><li>• scheduler</li><li>• namingserver</li></ul>

- 예제

```
[MASTER]domain1.adminServer>show-system-thread-pool server1
Shows the current configuration.
the system thread pool of the server (server1).
=====
+-----+-----+
| Min                | 10  |
| Max                | 60  |
| Keep-Alive Time    | 300000|
| Queue Size         | 4096 |
| Max Stuck Thread Time | 0   |
| Action On Stuck Thread | NONE |
| Stuck Thread Check Period | 300000|
| Reserved Threads for the Service transaction | 0   |
+-----+-----+
=====
```

#### 4.2.5.7. thread-info

요청을 처리하는 Servlet Thread와 EJB RMI Thread의 정보 및 그 외의 JEUS Thread Pool들을 조회한다.

Servlet에서는 리스너의 Thread Pool과 각 작업 Thread(Worker Thread)에 대한 정보를 조회하며, EJB에서는 EJB 리모트 요청을 처리하는 EJB RMI Thread의 정보를 출력한다. 서버에서는 기본적으로 서버 Thread Pool에 대한 정보만을 조회하며 옵션에 따라 모든 Pool 정보를 볼 수도 있다. Batch 애플리케이션의 경우 Thread Pool을 설정했다면, 이 Thread Pool에 대한 정보를 조회하여 출력한다.

- alias

threadinfo, ti

• 사용법

```
thread-info [-server <server-name>]
            [-type <thread-type>]
            [-wc <web-connection-name> | -ctx <web-context-name>]
            [-r,--request | -a,--all]
            [-s <thread-state>]
            [-os,--only-stats]
            [-st,--stacktrace]
```

• 파라미터

파라미터	설명
[-server <server-name>]	서버의 이름이다.
[-type <thread-type>]	<p>&lt;thread-type&gt;에 다음 중에 하나를 입력한다.</p> <ul style="list-style-type: none"><li>• servlet : Servlet Thread</li><li>• ejb : EJB RMI Thread</li><li>• jms : JMS Thread</li><li>• server : Server Thread Pool</li><li>• batch : Batch 애플리케이션에서 사용되는 Thread Pool</li></ul> <p>이외에도 -ctx 옵션이 지정된 경우 다음의 값을 입력한다.</p> <ul style="list-style-type: none"><li>• context : Context Thread Pool</li><li>• webasync : Servlet 3.0 Async Processing을 위한 Thread Pool</li><li>• upgrade : WebSocket Container의 Async Send 또는 Http Upgrade의 NIO Servlet을 위한 Thread Pool</li></ul>
[-wc <web-connection-name>   -ctx <context-name>]	특정 웹 커넥션 또는 웹 컨텍스트를 지정한다. 웹 커넥션일 경우에는, 웹 커넥션의 Thread Pool 정보를 출력하고, 웹 컨텍스트일 경우에는 원하는 스레드 풀을 지정할 수 있고, 옵션이 없을 경우에는 Context, WebAsync, Upgrade Thread Pool 정보를 전부 출력한다.
[-r,--request   -a,--all]	<ul style="list-style-type: none"><li>• -r : Servlet Thread의 요청 정보만을 확인한다. Servlet Thread에 대해서만 유효하다.</li><li>• -a : Servlet에서는 Servlet Thread의 모든 정보를 확인하며 서버에서는 JEUS 내부적인 모든 pool의 정보 확인을 의미한다.</li></ul>

파라미터	설명
[-s <thread-state>]	특정 상태의 Thread만 조회하는 경우 그 상태를 입력한다.  <ul style="list-style-type: none"> <li>• active</li> <li>• idle</li> <li>• blocked</li> <li>• reconn</li> </ul>
[-os,--only-stats]	Servlet에서는 각 리스너별 통계 정보만을 출력한다. 서버에서는 pool의 통계 정보만을 출력한다.
[-st,--stacktrace]	서버에서 조회한 Pool들에 속한 모든 Active Threads(실제 task를 부여받고 수행 중인 Thread)의 trace를 조회한다. 서버에서만 유효하다.

#### • 예제

```
[MASTER]domain1.adminServer>thread-info -server server1

Thread information for the server [server1]
There are no EJB RMI threads for the server [server1].
There is no batch thread pool in server [server1].

=====
The web engine threads for 'ADMIN-HTTP'.

+---+-----+-----+-----+-----+
| tid|          name          | state | elapsed | uri |
+---+-----+-----+-----+-----+
| 45 | ADMIN-HTTP-w00         | waiting| 48521870|    |
+---+-----+-----+-----+-----+

elapsed: Elapsed time (ms)
=====

=====
Thread statistics for the 'ADMIN-HTTP'.

+-----+-----+-----+-----+-----+
|          | total| active| idle| blocked| reconn|
+-----+-----+-----+-----+-----+
| The number of threads. |    1 |    0 |    1 |    0 |    0 |
+-----+-----+-----+-----+-----+

total = active + idle, reconn: reconnecting
=====

=====
The web engine threads for 'http1'.

+---+-----+-----+-----+-----+
| tid |          name          | state | elapsed | uri |
+---+-----+-----+-----+-----+
| 46 | http1-w00              | waiting| 48521873|    |
+---+-----+-----+-----+-----+
```

47	http1-w01	waiting	48521868	
48	http1-w02	waiting	48521868	
49	http1-w03	waiting	48521868	
51	http1-w05	waiting	48521862	
52	http1-w06	waiting	48521858	
53	http1-w07	waiting	48521858	
54	http1-w08	waiting	48521857	
55	http1-w09	waiting	48521857	
21203	http1-w10	waiting	37629	

elapsed: Elapsed time (ms)

=====

=====

Thread statistics for the 'http1'.

	total	active	idle	blocked	reconn
The number of threads.	10	0	10	0	0

total = active + idle, reconn: reconnecting

=====

=====

The threads for the 'threadpool.System' thread pool.

tid	name	thread state	active thread
93	threadpool.System-2	TIMED_WAITING	false
92	threadpool.System-1	TIMED_WAITING	false
94	JMXMP [adminServer-94]	RUNNABLE	true

=====

=====

The statistics for the 'threadpool.System' thread pool.

pool name	minimum pool size	maximum pool size	current pool size	work queue size	remaining work queue size
threadpool	0	100	3	4096	4096
.System					

=====

=====

The threads for the 'chunk\_checkpoint' thread pool.

tid	name	thread state	active thread
(No data available)			

=====

```

=====
The statistics for the 'chunk_checkpoint' thread pool.

+-----+-----+-----+-----+-----+-----+
| pool name | minimum | maximum | current | work | remaining work |
|           | pool size | pool size | pool size | queue size | queue size |
+-----+-----+-----+-----+-----+-----+
| chunk_che | 0 | 10 | 0 | 4096 | 4096 |
| ckpoint   |   |   |   |   |   |
+-----+-----+-----+-----+-----+-----+
=====

```

## 4.2.6. Application 관련 명령어

애플리케이션 관련 명령어의 목록은 다음과 같다.

명령어	설명
<a href="#">add-application-repository</a>	새로운 애플리케이션 저장소를 등록한다.
<a href="#">add-application-target</a>	deploy되어 있는 애플리케이션의 Target에 서버나 클러스터를 추가한다. 해당 서버나 클러스터에 해당 애플리케이션을 deploy한다. JEUS Master Server-Managed Server 구조에서만 수행 가능한 명령어이다.
<a href="#">application-info</a>	도메인에 존재하는 애플리케이션의 정보를 확인한다.
<a href="#">deploy-application</a>	설치한 애플리케이션을 Target 서버로 deploy한다.  Target으로 지정된 서버 중 하나라도 deploy가 실패하면 성공한 서버에도 undeploy를 수행한다.
<a href="#">deploy-library</a>	설치되어 있는 라이브러리 파일을 서버나 클러스터를 대상으로 하여 배포한다.
<a href="#">deployment-plan-info</a>	설치되어 있는 Deployment Plan의 정보를 확인한다.
<a href="#">distribute-application</a>	설치된 애플리케이션을 각 서버나 클러스터로 배포한다.  파일 배포 기능으로 실제 서비스를 위해서는 <a href="#">start-application</a> 명령어로 애플리케이션을 시작해야 한다.
<a href="#">install-application</a>	애플리케이션을 설치한다. 애플리케이션의 ID를 정의할 수 있다.
<a href="#">install-deployment-plan</a>	Deployment Plan을 설치한다. Deployment Plan의 식별자를 정의할 수 있다.
<a href="#">install-library</a>	라이브러리 파일을 설치한다.
<a href="#">library-info</a>	설치 및 배포한 라이브러리들에 대한 정보를 출력한다.
<a href="#">list-application-repositories</a>	등록되어 있는 애플리케이션 저장소들의 정보를 확인한다.
<a href="#">redploy-application</a>	특정 애플리케이션을 다시 deploy한다.
<a href="#">remove-application-repository</a>	등록되어 있는 애플리케이션 저장소를 삭제한다.



명령어	설명
<code>remove-application-target</code>	특정 서버나 특정 클러스터에 deploy 또는 배포된 애플리케이션을 해당 서버나 클러스터에서 제거한다. JEUS Master Server-Managed Server 구조에서만 수행 가능한 명령어이다.
<code>start-application</code>	배포된 애플리케이션의 서비스를 시작한다.
<code>stop-application</code>	서비스 중인 애플리케이션을 정지시킨다.
<code>undeploy-application</code>	deploy된 애플리케이션을 undeploy한다.
<code>uninstall-application</code>	설치되어 있는 애플리케이션을 제거한다.
<code>uninstall-deployment-plan</code>	설치되어 있는 Deployment Plan을 제거한다.
<code>undeploy-library</code>	deploy한 라이브러리를 undeploy한다.
<code>uninstall-library</code>	설치되어 있는 라이브러리 파일을 삭제한다.

#### 4.2.6.1. add-application-repository

새로운 애플리케이션 저장소(Repository)를 추가한다.

- alias

`addapprepo, installapps`

- 사용법

```
add-application-repository [<application-repository-path>]
                           [-f, --forceLock]
                           [-detail]
```

- 파라미터

파라미터	설명
<code>[&lt;application-repository-path&gt;]</code>	추가하려고 하는 애플리케이션 저장소의 절대 경로이다.
<code>[-f, --forceLock]</code>	설정 변경을 강제로 진행하여 반영한다.
<code>[-detail]</code>	정보를 상세히 출력할 때 사용한다.

- 예제

```
[MASTER]domain1.adminServer>add-application-repository /home/user1/apps
Successfully performed the ADD operation for An application repository.
Check the results using "add-application-repository or list-application-repositories"
```

#### 4.2.6.2. add-application-target

deploy되어 있는 애플리케이션의 Target에 서버나 클러스터를 추가한다. 해당 서버나 클러스터에 해당 애플리케이션을 deploy한다. 해당 애플리케이션이 DEPLOYED, DISTRIBUTED, RUNNING 상태일 때만 명령어를 사용할 수 있다. JEUS Master Server-Managed Server 구조에서만 수행 가능한 명령어이다.

- alias

addapptarget, addtarget

- 사용법

```
add-application-target <application-id>
                        [-servers <server-list>]
                        [-clusters <cluster-list>]
                        [-vh,--virtualHost <virtual-host>]
```

- 파라미터

파라미터	설명
<application-id>	Target을 추가하려고 하는 애플리케이션의 ID이다.  해당 애플리케이션이 DEPLOYED, DISTRIBUTED, RUNNING 상태일 때만 명령어를 사용할 수 있다.
[-servers <server-list>]	애플리케이션을 추가할 서버의 이름들이다. 서버 이름은 콤마(,)로 구분한다.
[-clusters <cluster-list>]	애플리케이션을 추가할 클러스터의 이름들이다. 클러스터 이름은 콤마(,)로 구분한다.
[-vh,--virtualHost <virtual-host>]	가상 호스트 이름을 지정한다.

- 예제

```
[MASTER]domain1.adminServer>add-application-target myApp -servers server1
add a target server or cluster to the application for the application [deployment_helloear]
succeeded.
```

#### 4.2.6.3. application-info

도메인에 존재하는 애플리케이션의 정보를 확인한다.

- alias

appinfo, list-applications

- 사용법

```

application-info [-server <server-name>]
                 [-id <application-id> | -state <application-state>]
                 [-type <application-type>]
                 [-module <module-name>]
                 [-bean <bean-name>]
                 [-jndi <jndi-context>]
                 [-mod <jndi-module-name>]
                 [-comp <jndi-component-name>]
                 [-detail,--detail]
                 [-sps,--stateperserver]

```

#### • 파라미터

파라미터	설명
[-server <server-name>]	정보를 확인할 서버의 이름이다. 해당 서버의 애플리케이션 정보를 좀 더 자세히 조회한다.
[-id <application-id>]	정보를 확인할 애플리케이션의 ID이다.
[-state <application-state>]	애플리케이션의 특정 상태를 지정하여 정보를 확인한다.
[-type <application-type>]	애플리케이션의 특정 종류를 지정하여 정보를 확인한다.  <application-type>에 다음 중 하나를 설정한다. <ul style="list-style-type: none"> <li>• EAR</li> <li>• EJB</li> <li>• WAR</li> <li>• CAR</li> <li>• RAR</li> </ul>
[-module <module-name>]	EAR 내부의 모듈 정보를 확인한다.
[-bean <bean-name>]	EJB Bean에 대한 자세한 정보를 조회하기 위해 사용한다.  server, id, module 정보가 필요하다.
[-jndi <jndi-context>]	특정 애플리케이션의 JNDI 정보를 확인한다.  server, id 정보가 필요하다.
[-mod <jndi-module-name>]	특정 모듈의 JNDI Namespace를 확인한다.
[-comp <jndi-component-name>]	특정 컴포넌트의 Namespace를 확인한다.
[-detail,--detail]	stand-alone 모듈의 상세 정보를 조회할 때 사용한다.  웹 모듈의 경우 서블릿, 필터, 리스너, WebSocket Server Endpoints, EJB Bean(EJB in a .war) 목록이 출력된다.

파라미터	설명
[-sps,--stateperserver]	<p>추가로 각 애플리케이션에 대해 서버별 상태를 출력한다.</p> <p>대상 서버가 내려가 있거나 하는 등의 이유로 상태가 없는 경우에는 'NO_STATE'라고 출력한다.</p>

## • 예제

### ◦ 기본적인 출력

다음은 기본적인 출력 결과물이다.

```
[MASTER]domain1.adminServer>application-info
Application information for the domain [domain1].
=====
+-----+-----+-----+-----+-----+-----+
| Applicat | Applicati | State | Target | Target | Application Path |
| ion ID   | on Type   |       | Servers| Clusters|                   |
+-----+-----+-----+-----+-----+-----+
| myApp    | EAR       | RUNNING | adminServer |      | ${INSTALL_HOME}/my |
|          |           |         |             |      | App/myApp.ear      |
+-----+-----+-----+-----+-----+-----+
| myEJB    | EJB       | RUNNING |           | cluster1 | ${INSTALL_HOME}/my |
|          |           |         |           |          | EJB/myEJB.jar      |
+-----+-----+-----+-----+-----+-----+
| myWeb    | WAR       | RUNNING | server1,adm |      | ${INSTALL_HOME}/my |
|          |           |         | inServer   |      | Web/myWeb.war      |
+-----+-----+-----+-----+-----+-----+
=====
```

### ◦ stateperserver 옵션을 사용한 경우

각 애플리케이션에 대한 서버별 상태를 보여주는 것을 확인할 수 있다. 상태를 얻을 수 없는 서버에 대해서는 'NO\_STATE'라고 표시한다.

```
[MASTER]domain1.adminServer>application-info -sps
Application information for the domain [domain1].
=====
+-----+-----+-----+-----+-----+-----+
| Applicat | Applicati | State | Target | Target | Application Path |
| ion ID   | on Type   |       | Servers| Clusters|                   |
+-----+-----+-----+-----+-----+-----+
| myApp    | EAR       | RUNNING | adminServer |      | ${INSTALL_HOME}/my |
|          |           |         |             |      | App/myApp.ear      |
+-----+-----+-----+-----+-----+-----+
| myEJB    | EJB       | RUNNING |           | cluster1 | ${INSTALL_HOME}/my |
|          |           |         |           |          | EJB/myEJB.jar      |
+-----+-----+-----+-----+-----+-----+
| myWeb    | WAR       | RUNNING | server1,adm |      | ${INSTALL_HOME}/my |
|          |           |         | inServer   |      | Web/myWeb.war      |
+-----+-----+-----+-----+-----+-----+
=====
```

Server-specific state information for each application in the domain [domain1]

```
=====
```

Application ID	Application Type	State	Server	Cluster	Application Path
myApp	EAR	RUNNING	adminServer		\${INSTALL_HOME}/myApp/myApp.ear
myEJB	EJB	RUNNING	server2	cluster1	\${INSTALL_HOME}/myEJB/myEJB.jar
myEJB	EJB	NO STATE	server3	cluster1	\${INSTALL_HOME}/myEJB/myEJB.jar
myWeb	WAR	NO STATE	server1		\${INSTALL_HOME}/myWeb/myWeb.war
myWeb	WAR	RUNNING	adminServer		\${INSTALL_HOME}/myWeb/myWeb.war

```
=====
```

#### 4.2.6.4. deploy-application

애플리케이션을 Target 서버로 deploy한다. Target으로 지정된 서버 중 하나라도 deploy가 실패하면 성공한 서버에도 undeploy를 수행한다.

- alias

deployapp, deploy

- 사용법

```
deploy-application [<application-id>]
    [-path,--applicationPath <path>]
    [-servers <server-list>]
    [-clusters <cluster-list>]
    [-all]
    [-exmaster,--excludeMaster]
    [-vh,--virtualHost <virtual-host>]
    [-svh,--systemVirtualHost]
    [-type,--applicationType <application-type>]
    [-cl,--classloading <classloading-mode>]
    [-auto,--autoRedeployCheckInterval <auto-redeploy-check-interval>]
    [-security,--securityDomainName <security-domain-name>]
    [-fast,--fastDeploy]
    [-keep,--keepGenerated]
    [-shared,--sharedDisk]
    [-u,--upgrade]
    [-plan,--deploymentPlanName <deployment-plan-name>]
    [-ctxp,--contextPath <context-path>]
    [-staging]
    [-lib,--libraryId <library-IDs>]
```

```

[-version,--libraryVersion <library-versions>]
[-con,--concurrent]
[-thn <thread-pool-name>]
[-tmin <thread-pool-min>]
[-tmax <thread-pool-max>]
[-tidle <thread-pool-maxIdleTimeout>]
[-qs <thread-pool-maxQueueCount>]
[-b <thread-pool base> | -uri <thread-pool-uri>]
[-jwdp <jeus-web-dd.xml path>]
[-wp <web.xml path>]
[-ss,--sessionScope <scope-name>]

```

## • 파라미터

파라미터	설명
[<application-id>]	애플리케이션을 설치(install-application)할 때 입력한 애플리케이션의 ID이다. 한 개 이상의 애플리케이션을 deploy하려면 콤마(,)를 통해 ID를 구분한다. 파일 경로를 통해 distribute할 때도 애플리케이션에 ID를 부여할 수 있다. 이 경우에는 하나의 경로와 하나의 ID만 올 수 있다.
[-path,--applicationPath <path>]	애플리케이션을 설치하지 않고 파일 경로를 이용할 때 사용한다. JEUS Master Server와 같은 머신에 있는 파일만 가능하다.
[-servers <server-list>]	애플리케이션을 deploy할 서버들이다. 서버 이름은 콤마(,)로 구분한다.
[-clusters <cluster-list>]	애플리케이션을 deploy할 클러스터들이다. 클러스터 이름은 콤마(,)로 구분한다.
[-all]	도메인에 존재하는 모든 서버에 deploy할 때 지정한다.
[-exmaster,--excludeMaster]	all 옵션을 사용하는 경우 JEUS Master Server를 애플리케이션을 deploy할 대상 서버 목록에서 제외하고자 할 때 지정한다.
[-vh,--virtualHost <virtual-host>]	가상 호스트 주소이다. 웹 엔진에 설정한 가상 호스트 이름 중 하나를 선택한다. 웹 모듈에서만 동작한다.
[-svh,--systemVirtualHost]	애플리케이션을 시스템 가상 호스트에 배포한다.
[-type,--applicationType <application-type>]	<p>애플리케이션의 종류를 지정한다.</p> <p>&lt;application-type&gt;에 다음 중 하나를 설정한다.</p> <ul style="list-style-type: none"> <li>• EAR</li> <li>• EJB</li> <li>• WAR</li> <li>• CAR</li> <li>• RAR</li> </ul> <p>옵션을 주지 않더라도 DD나 Annotation을 통해 자동으로 지정될 수 있다.</p>

파라미터	설명
<code>[-cl,--classloading &lt;classloading-mode&gt;]</code>	<p>클래스 로딩 형식을 지정한다.</p> <p>&lt;classloading-mode&gt;에 다음 중 하나를 설정한다.</p> <ul style="list-style-type: none"> <li>• SHARED</li> <li>• ISOLATED (기본값)</li> </ul>
<code>[-auto,--autoRedeployCheckInterval &lt;auto-redeploy-check-interval&gt;]</code>	<p>애플리케이션을 자동으로 redeploy하기 위한 체크 주기이다.</p> <p>(단위: ms)</p>
<code>[-security,--securityDomainName &lt;security-domain-name&gt;]</code>	<p>애플리케이션에 적용될 보안 도메인 이름을 선택한다.</p> <p>기본값으로 보안 도메인 설정에서 default-application-domain으로 설정된 값이 적용된다.</p>
<code>[-fast,--fastDeploy]</code>	<p>deploy할 때 생성되는 파일을 미리 생성해 놓은 경우 미리 생성된 파일을 사용하여 deploy 속도를 향상시킬지 여부를 설정한다.</p>
<code>[-keep,--keepGenerated]</code>	<p>deploy할 때 생성된 파일을 유지할지 여부를 설정한다.</p>
<code>[-shared,--sharedDisk]</code>	<p>NAS를 사용할 경우 설정한다. 애플리케이션 파일을 복사하지 않도록 하기 위한 옵션이다.</p>
<code>[-u,--upgrade]</code>	<p>미리 설치하지 않은 애플리케이션을 배포할 때 호환성을 위해 JEUS 6 이전 버전의 DD를 현재 버전에 맞는 형식으로 바꿔주기 위한 옵션이다.</p>
<code>[-plan,--deploymentPlanName &lt;deployment-plan-name&gt;]</code>	<p>Deployment Plan 파일의 이름을 설정하기 위한 옵션이다.</p>
<code>[-ctxp,--contextPath &lt;context-path&gt;]</code>	<p>웹 애플리케이션의 경우 컨텍스트를 설정하기 위한 옵션이다.</p>
<code>[-staging]</code>	<p>Exploded module 형태의 애플리케이션을 압축하여 다른 머신에 전송하기 위한 옵션이다.</p>
<code>[-lib,--libraryId &lt;library-IDs&gt;]</code>	<p>애플리케이션이 참조할 라이브러리들을 지정한다. 여러 라이브러리를 지정할 경우 콤마(,)로 구분한다.</p>
<code>[-version,--libraryVersion &lt;library-versions&gt;]</code>	<p>참조할 라이브러리들에 대한 버전을 지정한다. 여러 라이브러리를 지정할 경우 콤마(,)로 구분한다.</p>
<code>[-con, --concurrent]</code>	<p>여러 애플리케이션을 deploy하려고 할 때 애플리케이션을 동시에 deploy할지 여부를 설정한다.</p>
<code>[-thn &lt;thread-pool name&gt;]</code>	<p>애플리케이션 관련 요청을 처리할 Thread Pool 이름을 설정한다. command로는 하나의 Context Thread Pool만 설정할 수 있다. 해당 옵션을 적용할 경우 -tmin, -tmax 옵션은 필수 값이다.</p>
<code>[-tmin &lt;thread-pool min&gt;]</code>	<p>Context Thread Pool에서 Worker Thread를 유지해야 하는 최소 개수를 설정한다.</p>
<code>[-tmax &lt;thread-pool max&gt;]</code>	<p>Context Thread Pool에서 Worker Thread를 유지해야 하는 최대 개수를 설정한다.</p>

파라미터	설명
<code>[-tidle &lt;thread-pool max-idle-time&gt;]</code>	Context Thread Pool에서 제거되기 전에 Idle 상태로 머물러 있는 Worker Thread의 최대 시간을 설정한다.
<code>[-qs &lt;thread-pool max-queue&gt;]</code>	Context Thread Pool에서 Queue에 대기할 수 있는 최대 요청값을 설정한다.
<code>[-b &lt;thread-pool base&gt;   -uri &lt;thread-pool uri&gt;]</code>	추가하고자 하는 Thread Pool이 base인지 Service Group인지 구분하고자 옵션을 추가해야 한다. -b 옵션을 줄 경우 base Thread Pool로 인식하고, 아무 옵션도 안 줄 경우에는 Service Group Thread Pool을 생성하고 uri에는 Empty String이 들어간다. 하나 이상의 uri를 설정하려면 스페이스 없이 ','로 구분해서 넣는다.
<code>[-jwdp &lt;jeus-web-dd.xml path&gt;]</code>	<p>사용할 jeus-web-dd.xml의 위치를 지정한다.</p> <p>여러 개의 jeus-web-dd.xml이 있을 경우 우선순위가 높은 하나만 적용된다. 이때 우선순위는 DeploymentPlan, DeployApplicationCommand에 언급된 path, WEB-INF 하위 순으로 적용된다.</p>
<code>[-wp &lt;web.xml path&gt;]</code>	<p>사용할 web.xml의 위치를 지정한다.</p> <p>여러 개의 web.xml이 있을 경우 우선순위가 높은 하나의 web.xml만 적용된다(해당 web.xml은 merge 불가). 이때 우선순위는 DeploymentPlan, DeployApplicationCommand에 언급된 path, WEB-INF 하위, domain/config/servlet 하위 순으로 적용된다.</p>
<code>[-ss,--sessionScope &lt;scope-name&gt;]</code>	application이 포함될 scope를 지정한다.

## • 예제

Target 서버 또는 클러스터는 `[-servers]`, `[-clusters]`, `[-all]` 옵션 중 하나를 선택하여 지정할 수 있다.

### ◦ 설치된 애플리케이션

application-id만 지정해서 배포할 수 있다.

```
[MASTER]domain1.adminServer>deploy-application myApp -servers server1
deploy the application for the application [myApp] succeeded.
```

### ◦ 설치되지 않은 애플리케이션

path만 지정하거나, application-id와 path를 모두 지정해서 배포할 수 있다.

path만 지정

```
[MASTER]domain1.adminServer>deploy-application -path /home/user/apps/sample.war -servers
server1
deploy the application for the application [/home/user/apps/sample.war] succeeded.
```



application-id와 path를 모두 지정

```
[MASTER]domain1.adminServer>deploy-application myApp -path /home/user/apps/sample.war
-servers server1
deploy the application for the application [myApp] succeeded.
```

#### 4.2.6.5. deploy-library

설치되어 있는 라이브러리를 서버나 클러스터에 배포한다.

- alias

deploylib

- 사용법

```
deploy-library <library-id>
    [-version,--libraryVersion <version>]
    [-all]
    [-clusters <cluster-list>]
    [-servers <server-list>]
```

- 파라미터

파라미터	설명
<library-id>	배포할 라이브러리의 식별자를 지정한다.
[-version,--libraryVersion <version>]	설치할 라이브러리의 버전을 지정한다.
[-all]	모든 서버를 대상으로 배포하고자 하는 경우 지정한다.
[-clusters <cluster-list>]	배포 대상이 될 클러스터를 지정한다. 여러 클러스터를 지정할 경우 콤마(,)로 구분한다.
[-servers <server-list>]	라이브러리를 배포할 서버를 지정한다. 여러 서버를 지정할 경우 콤마(,)로 구분한다.

- 예제

```
[MASTER]domain1.adminServer>deploy-library log4j -version 1.2.17 -servers adminServer
deploy the library [log4j] succeeded.
```

#### 4.2.6.6. deployment-plan-info

설치되어 있는 Deployment Plan의 정보를 확인한다.

- alias

dpinfo

- 사용법

```
deployment-plan-info [-name,--deploymentPlanName <deployment-plan-name>]
```

- 파라미터

파라미터	설명
[-name,--deploymentPlanName <deployment-plan-name>]	상세 정보를 확인할 Deployment Plan의 이름(식별자)을 지정한다.

- 예제

```
[MASTER]domain1.adminServer>deployment-plan-info
The list of deployment plans installed in the domain and the applications to which each
deployment plan applies
=====
+-----+-----+
|          Deployment plan          | Applications |
+-----+-----+
| plan1                             |              |
+-----+-----+
=====
```

#### 4.2.6.7. distribute-application

애플리케이션을 각 서버나 클러스터로 배포한다. 파일 배포 기능으로써 실제 서비스를 위해서는 [start-application](#) 명령어로 애플리케이션을 시작해야 한다.

- alias

distributeapp, distapp, distribute

- 사용법

```
distribute-application [<application-id>]
    [-path,--applicationPath <path>]
    [-servers <server-list>]
    [-clusters <cluster-list>]
    [-all]
    [-exmaster,--excludeMaster]
    [-vh,--virtualHost <virtual-host>]
    [-svh,--systemVirtualHost]
    [-type,--applicationType <application-type>]
    [-cl,--classloading <classloading-mode>]
    [-auto,--autoRedeployCheckInterval <auto-redeploy-check-interval>]
```

```

[-security,--securityDomainName <security-domain-name>]
[-fast,--fastDeploy]
[-keep,--keepGenerated]
[-shared,--sharedDisk]
[-u,--upgrade]
[-plan,--deploymentPlanName <deployment-plan-name>]
[-ctxp,--contextPath <context-path>]
[-staging]
[-lib,--libraryId <library-IDs>]
[-version,--libraryVersion <library-versions>]
[-con,--concurrent]

```

#### • 파라미터

파라미터	설명
[<application-id>]	애플리케이션을 설치(install-application)할 때 입력한 애플리케이션의 ID이다. 한 개 이상의 애플리케이션을 distribute하려면 콤마(,)를 통해 ID를 구분한다. 파일 경로를 통해 distribute할 때도 애플리케이션에 ID를 부여할 수 있다. 이 경우에는 하나의 경로와 하나의 ID만 올 수 있다.
[-path,--applicationPath <path>]	애플리케이션을 설치하지 않고 파일 경로를 이용할 때 사용한다. JEUS Master Server와 같은 머신에 있는 파일만 가능하다.
[-servers <server-list>]	애플리케이션을 설치할 서버들이다. 서버 이름은 콤마(,)로 구분한다.
[-clusters <cluster-list>]	애플리케이션을 설치할 클러스터들이다. 클러스터 이름은 콤마(,)로 구분한다.
[-all]	도메인에 존재하는 모든 서버에 설치할 때 지정한다.
[-exmaster,--excludeMaster]	all 옵션을 사용하는 경우 JEUS Master Server를 애플리케이션을 deploy할 대상 서버 목록에서 제외하고자 할 때 지정한다.
[-vh,--virtualHost <virtual-host>]	가상 호스트 주소로 웹 엔진에 설정한 가상 호스트 이름 중 하나를 선택한다. 웹 모듈에서만 동작한다.
[-svh,--systemVirtualHost]	애플리케이션을 시스템 가상 호스트에 배포한다.
[-type,--applicationType <application-type>]	<p>애플리케이션의 종류를 지정한다.</p> <p>&lt;application-type&gt;에 다음 중 하나를 설정한다.</p> <ul style="list-style-type: none"> <li>• EAR</li> <li>• EJB</li> <li>• WAR</li> <li>• CAR</li> <li>• RAR</li> </ul> <p>옵션을 주지 않더라도 DD나 Annotation을 통해 자동으로 지정될 수 있다.</p>

파라미터	설명
<code>[-cl,--classloading &lt;classloading-mode&gt;]</code>	클래스 로딩 형식을 지정한다.  <classloading-mode>에 다음 중 하나를 설정한다. <ul style="list-style-type: none"> <li>• SHARED</li> <li>• ISOLATED(기본값)</li> </ul>
<code>[-auto,--autoRedeployCheckInterval &lt;auto-redeploy-check-interval&gt;]</code>	애플리케이션을 자동으로 Redeploy하기 위한 체크 주기이다. (단위: ms)
<code>[-security,--securityDomainName &lt;security-domain-name&gt;]</code>	애플리케이션에 적용될 보안 도메인 이름을 선택한다. 기본값으로 보안 도메인 설정에서 default-application-domain으로 설정된 값이 적용된다.
<code>[-fast,--fastDeploy]</code>	deploy할 때 생성되는 파일을 미리 생성해 놓은 경우 미리 생성된 파일을 사용하여 Deploy 속도를 향상시킬지 여부이다.
<code>[-keep,--keepGenerated]</code>	deploy할 때 생성된 파일을 유지할지 여부이다.
<code>[-shared,--sharedDisk]</code>	NAS를 사용할 경우 설정한다. 애플리케이션 파일을 복사하지 않도록 하기 위한 옵션이다.
<code>[-u,--upgrade]</code>	미리 설치하지 않은 애플리케이션을 배포할 때 호환성을 위해 JEUS 6 이전 버전의 DD를 현재 버전에 맞는 형식으로 바꿔주기 위한 옵션이다.
<code>[-plan,--deploymentPlanName &lt;deployment-plan-name&gt;]</code>	Deployment Plan 파일의 이름을 설정하기 위한 옵션이다.
<code>[-ctxp,--contextPath &lt;context-path&gt;]</code>	웹 애플리케이션의 경우 컨텍스트를 설정하기 위한 옵션이다.
<code>[-staging]</code>	Exploded module 형태의 애플리케이션을 압축하여 다른 머신에 전송하기 위한 옵션이다.
<code>[-lib,--libraryId &lt;library-IDs&gt;]</code>	애플리케이션이 참조할 라이브러리들을 지정한다. 여러 라이브러리를 지정할 경우 콤마(,)로 구분한다.
<code>[-version,--libraryVersion &lt;library-versions&gt;]</code>	참조할 라이브러리들에 대한 버전을 지정한다. 여러 라이브러리를 지정할 경우 콤마(,)로 구분한다.
<code>[-con, --concurrent]</code>	여러 애플리케이션을 distribute하려고 할 때, 애플리케이션을 동시에 distribute할지 여부를 설정한다.

## • 예제

Target 서버 또는 클러스터는 `[-servers]`, `[-clusters]`, `[-all]` 옵션 중 하나를 선택하여 지정할 수 있다.

### ◦ 설치된 애플리케이션

application-id만 지정해서 배포할 수 있다.

```
[MASTER]domain1.adminServer>distribute-application myApp -servers server1
```

```
distribute the application for the application [myApp] succeeded.
```

- 설치되지 않은 애플리케이션

path만 지정하거나, application-id와 path를 모두 지정해서 배포할 수 있다.

path만 지정

```
[MASTER]domain1.adminServer>distribute-application -path /home/user/apps/sample.war -servers  
server1  
distribute the application for the application [/home/user/apps/sample.war] succeeded.
```

application-id와 path를 모두 지정

```
[MASTER]domain1.adminServer>distribute-application myApp -path /home/user/apps/sample.war  
-servers server1  
distribute the application for the application [myApp] succeeded.
```

#### 4.2.6.8. install-application

애플리케이션을 설치한다. 애플리케이션의 ID를 정의할 수 있다. 별도로 ID를 지정하지 않는 경우 애플리케이션의 파일 이름을 사용하며, 문자 점(.)은 문자 언더바(\_)로 대체된다.

- alias

installapp, install-app

- 사용법

```
install-application <application-source-path>  
                [-id,--applicationId <application-id>]  
                [-f,--force]  
                [-u,--upgrade]
```

- 파라미터

파라미터	설명
<application-source-path>	애플리케이션의 경로를 지정한다.
[-id,--applicationId <application-id>]	애플리케이션의 ID를 지정한다.
[-f,--force]	동일한 ID가 존재하는 경우 현재 애플리케이션으로 덮어쓸지 여부를 선택한다.
[-u,--upgrade]	호환성을 위해 JEUS 6 이전 버전의 DD를 현재 버전에 맞는 형식으로 바꿔주기 위한 옵션이다.

- 예제

```
[MASTER]domain1.adminServer>install-application /home/apps/myApp.ear -id myApp
Successfully installed the application [myApp].
```

#### 4.2.6.9. install-deployment-plan

Deployment Plan을 설치한다. 이때 설치할 Deployment Plan의 식별자를 정의할 수 있다.

- alias

installdp

- 사용법

```
install-deployment-plan -path,--deploymentPlanPath <deployment-plan-path>
                        [-name,--deploymentPlanName <deployment-plan-name>]
                        [-f,--force]
```

- 파라미터

파라미터	설명
-path,--deploymentPlanPath <deployment-plan-path>	설치할 Deployment Plan의 로컬 경로를 지정한다.
[-name,--deploymentPlanName <deployment-plan-name>]	설치할 Deployment Plan의 도메인 상에서의 식별자(이름)를 정의한다.
[-f,--force]	동일한 이름을 가지고 있는 기존 Deployment Plan을 덮어쓰도록 강제한다.

- 예제

```
[MASTER]domain1.adminServer>install-deployment-plan -path /home/plans/jeus-deployment-plan.xml
-name plan1
Installing the deployment plan [plan1] was successful.
```

#### 4.2.6.10. install-library

라이브러리 파일을 설치한다.

- alias

installlib, install-lib

- 사용법

```
install-library <library-id>
    -path <library-source-path>
    [-version,--libraryVersion <version>]
```

- 파라미터

파라미터	설명
<library-id>	설치할 라이브러리에 대한 식별자를 지정한다.
-path <library-source-path>	라이브러리 파일이 위치한 경로를 지정한다.
[-version,--libraryVersion <version>]	설치할 라이브러리의 버전을 지정한다. 지정하지 않은 경우 1.0으로 간주한다.

- 예제

```
[MASTER]domain1.adminServer>install-library log4j -path /home/lib/apache-log4j-1.2.17/log4j-1.2.17.jar -version 1.2.17
Successfully installed the library [log4j] version [1.2.17].
```

#### 4.2.6.11. library-info

현재 설치 및 배포되어 있는 라이브러리에 대한 목록을 출력한다.

- alias

libinfo, list-libraries

- 사용법

```
library-info
```

- 예제

```
[MASTER]domain1.adminServer>library-info
Library information
=====
+-----+-----+-----+-----+-----+-----+
| Library ID| Version| State | Target Servers| Target Clusters | Applications|
+-----+-----+-----+-----+-----+-----+
| log4j     | 1.2.17 | RUNNING| adminServer  |                  |              |
+-----+-----+-----+-----+-----+-----+
=====
```

#### 4.2.6.12. list-application-repositories

등록되어 있는 애플리케이션 저장소들의 정보를 확인한다.

- alias

listapprepos

- 사용법

```
list-application-repositories
```

- 예제

```
[MASTER]domain1.adminServer>list-application-repositories
Application Repositories
=====
+-----+-----+
|                               Path of Application Repository                               |
+-----+-----+
| /home/user1/apps/              |
| //host1/shared/apps            |
+-----+-----+
=====
```

#### 4.2.6.13. redeploy-application

특정 애플리케이션을 다시 deploy한다.

- alias

redapp, redeploy

- 사용법

```
redeploy-application <application-id>
    [-path,--uploadPath <application-upload-path>]
    [-masterPath,--pathManuallyInstalled
    <application-source-path-in-master>]
    [-to <graceful-timeout>]
    [-f,--force]
    [-distonly,--distributeOnly]
    [-plan,-deploymentPlanName <deployment-plan-name>]
    [-lib,--libraryId <library-IDs>]
    [-version,--libraryVersion <library-versions>]
    [-u,--upgrade]
```

- 파라미터



파라미터	설명
<application-id>	애플리케이션의 ID이다. 한 개 이상의 애플리케이션을 redeploy하려면 콤마(,)로 구분한다.
[-path, --uploadPath <application-upload-path>]	변경된 애플리케이션의 파일 경로이다. 파일 재설치 동작을 포함한다.
[-masterPath, --pathManuallyInstalled <application-source-path-in-master>]	JEUS Master Server 머신에서 접근가능한 변경된 애플리케이션의 로컬 파일 경로이다. 설치하는 작업을 포함하지 않는다.
[-to <graceful-timeout>]	Graceful undeploy의 타임아웃 값이다. (단위: 초)
[-f, --force]	기존의 애플리케이션이 계속 서비스되도록 한다.
[-distonly, --distributeOnly]	새 애플리케이션을 서비스하지 않고 각 서버로의 설치까지만 진행한다.
[-plan, --deploymentPlanName <deployment-plan-name>]	Deployment Plan 파일의 이름을 설정하기 위한 옵션이다.
[-lib, --libraryId <library-IDs>]	애플리케이션이 참조할 라이브러리들을 지정한다. 여러 라이브러리를 지정할 경우 콤마(,)로 구분한다.
[-version, --libraryVersion <library-versions>]	참조할 라이브러리들에 대한 버전을 지정한다. 여러 라이브러리를 지정할 경우 콤마(,)로 구분한다.
[-u, --upgrade]	새 애플리케이션이 갖고 있는 DD를 현재 버전에 맞게 업그레이드한다. 이 옵션은 path 옵션이나 masterPath 옵션을 사용하여 새 애플리케이션 경로를 지정하였을 경우에만 동작한다.

- 예제

```
[MASTER]domain1.adminServer>redeploy-application myApp
redeploy application on JEUS Master Server for the application [myApp] succeeded.
```

#### 4.2.6.14. remove-application-repository

등록되어 있는 애플리케이션 저장소를 삭제한다.

- alias

rmapprepo, removeapprepo, uninstallapps

- 사용법

```
remove-application-repository [<application-repository-path>]
                               [-f, --forceLock]
                               [-detail]
```

- 파라미터

파라미터	설명
[<application-repository-path>]	삭제할 애플리케이션 저장소의 경로이다.
[-f, --forceLock]	설정 변경을 강제로 진행하여 반영한다.
[-detail]	정보를 자세히 출력하는 경우 사용한다.

- 예제

```
[MASTER]domain1.adminServer>remove-application-repository /home/user1/apps
Successfully performed the REMOVE operation for An application repository.
Check the results using "remove-application-repository or list-application-repositories"
```

#### 4.2.6.15. remove-application-target

특정 서버나 특정 클러스터에 Deploy 또는 배포된 애플리케이션을 해당 서버나 클러스터에서 제거한다. JEUS Master Server-Managed Server 구조에서만 수행 가능한 명령어이다.

- alias

remove-app-target, remove-target, rm-app-target, rmapptarget

- 사용법

```
remove-application-target <application-id>
                        [-servers <server-list>]
                        [-clusters <cluster-list>]
                        [-vh,--virtualHost <virtual-host>]
```

- 파라미터

파라미터	설명
<application-id>	제거할 애플리케이션의 ID이다.
[-servers <server-lists>]	애플리케이션을 제거할 서버의 이름을 지정한다. 서버 이름은 콤마(,)로 구분한다.
[-clusters <cluster-lists>]	애플리케이션을 제거할 클러스터의 이름을 지정한다. 클러스터 이름은 콤마(,)로 구분한다.
[-vh,--virtualHost <virtual-host>]	가상 호스트 이름을 지정한다.

- 예제

```
[MASTER]domain1.adminServer>remove-application-target myApp -servers server1
remove server or cluster target from the application for the application [myApp] succeeded.
```

#### 4.2.6.16. start-application

배포된 애플리케이션의 서비스를 시작한다.

- alias

startapp, start-app

- 사용법

```
start-application <application-id>
                 [-con, --concurrent]
```

- 파라미터

파라미터	설명
<application-id>	서비스를 시작할 애플리케이션의 ID이다. 한 개 이상의 애플리케이션을 시작하려면 콤마(,)로 구분한다.
[-con, --concurrent]	여러 애플리케이션을 start하려고 할 때, 애플리케이션을 동시에 start할지 여부를 설정한다.

- 예제

```
[MASTER]domain1.adminServer>start-application myApp
start the application for the application [myApp] succeeded.
```

#### 4.2.6.17. stop-application

서비스 중인 애플리케이션을 정지시킨다.

- alias

stopapp, stop-app

- 사용법

```
stop-application <application-id>
                 [-con, --concurrent]
```

- 파라미터

파라미터	설명
<application-id>	서비스를 정지시킬 애플리케이션의 ID이다. 한 개 이상의 애플리케이션을 정지하려면 콤마(,)로 구분한다.

파라미터	설명
[-con, --concurrent]	여러 애플리케이션을 stop하려고 할 때, 애플리케이션을 동시에 stop할지 여부를 설정한다.

- 예제

```
[MASTER]domain1.adminServer>stop-application myApp
stop the application for the application [myApp] succeeded.
```

#### 4.2.6.18. undeploy-application

deploy된 애플리케이션을 undeploy한다.

- alias

undeployapp, undeploy

- 사용법

```
undeploy-application <application-id>
    [-to,--gracefultimeout <graceful-undeploy-timeout>]
    [-f,--force <non-graceful undeploy>]
    [-new | -old | -all]
    [-con,--concurrent]
    [-rss,--removeSessionScope]
```

- 파라미터

파라미터	설명
<application-id>	애플리케이션의 ID이다. 한 개 이상의 애플리케이션을 undeploy하려면 콤마(,)로 구분한다.
[-to,--gracefultimeout <graceful-undeploy-timeout>]	Graceful undeploy할 때 적용되는 타임아웃 값이다. 처리 중인 요청을 입력한 시간만큼 완료되길 기다린다. (단위: 초)
[-f,--force <non-graceful undeploy>]	undeploy할 때 Graceful undeploy를 수행하지 않는다. 처리 중인 요청은 무시되고 애플리케이션은 undeploy된다.
[-new]	Graceful replot할 때 새로운 애플리케이션을 undeploy하고 기존 애플리케이션을 서비스한다.
[-old]	Graceful replot할 때 기존 애플리케이션을 undeploy한다.
[-all]	Graceful redeploy할 때 새로운 애플리케이션과 기존의 애플리케이션 모두를 undeploy한다.
[-con, --concurrent]	여러 애플리케이션을 undeploy하려고 할 때 애플리케이션을 동시에 undeploy할지 여부를 설정한다.
[-rss,--removeSessionScope]	설정된 scope에서 애플리케이션을 제거한다.

- 예제

```
[MASTER]domain1.adminServer>undeploy-application myApp
Undeploying [myApp] (This may take time due to graceful undeployment) .....
undeploy the application for the application [myApp] succeeded.
Successfully undeployed (elapsed = 2822ms)
```

#### 4.2.6.19. undeploy-library

서버나 클러스터를 대상으로 배포했던 라이브러리를 undeploy한다.

- alias

undeploylib

- 사용법

```
undeploy-library <library-id>
                    [-version,--libraryVersion <version>]
```

- 파라미터

파라미터	설명
<library-id>	undeploy할 라이브러리에 대한 식별자를 지정한다.
[-version,--libraryVersion <version>]	undeploy할 라이브러리의 버전을 지정한다.

- 예제

```
[MASTER]domain1.adminServer>undeploy-library log4j -version 1.2.17
undeploy the library [log4j] succeeded.
```

#### 4.2.6.20. uninstall-application

설치되어 있는 애플리케이션을 제거한다.

- alias

uninstallapp, uninstall

- 사용법

```
uninstall-application <application-id> | [-all]
```

- 파라미터

파라미터	설명
<application-id>	애플리케이션의 ID이다. 한 개 이상의 애플리케이션을 제거하려면 콤마(,)로 구분한다.
[-all]	설치되어 있는 모든 애플리케이션을 제거한다.

- 예제

```
[MASTER]domain1.adminServer>uninstall-application myApp
uninstall the application for the application [myApp] succeeded. : Successfully deleted [myApp].
```

#### 4.2.6.21. uninstall-deployment-plan

설치되어 있는 Deployment Plan을 제거한다.

- alias

uninstalldp

- 사용법

```
uninstall-deployment-plan <deployment-plan-name> | [-all]
```

- 파라미터

파라미터	설명
<deployment-plan-name>	제거할 Deployment Plan 식별자(이름)를 지정한다.
[-all]	설치되어 있는 모든 Deployment Plan을 제거한다.

- 예제

```
[MASTER]domain1.adminServer>uninstall-deployment-plan plan1
Uninstalling the deployment plan was successful.
```

#### 4.2.6.22. uninstall-library

설치되어 있는 라이브러리 파일을 삭제한다.

- alias

uninstalllib

- 사용법

```
uninstall-library <library-id>
                    [-version,--libraryVersion <version>]
```

- 파라미터

파라미터	설명
<library-id>	삭제할 라이브러리의 식별자를 지정한다.
[-version,--libraryVersion <version>]	삭제할 라이브러리의 버전을 지정한다.

- 예제

```
[MASTER]domain1.adminServer>uninstall-library log4j -version 1.2.17
uninstall the library [log4j] succeeded. : Successfully deleted [log4j].
```

## 4.2.7. EJB 엔진 관련 명령어

EJB 엔진 관련 명령어의 목록은 다음과 같다.

명령어	설명
<a href="#">cancel-ejb-timer</a>	특정 서버의 EJB 타이머를 취소한다.
<a href="#">ejb-timer-info</a>	특정 서버의 EJB 타이머 정보를 확인한다.
<a href="#">modify-active-management</a>	특정 서버의 Active management를 변경한다.
<a href="#">modify-check-resolution</a>	특정 서버의 Resolution 정보를 변경한다.
<a href="#">show-active-management</a>	특정 서버의 Active management 정보를 확인한다.
<a href="#">show-check-resolution</a>	특정 서버의 Resolution 정보를 확인한다.

### 4.2.7.1. cancel-ejb-timer

특정 서버의 EJB 타이머를 취소한다.

- alias

cancelejbtimer, canceltimer

- 사용법

```
cancel-ejb-timer -server <server-name>
                 -module <module-id> | -timer <timer-id>
```

- 파라미터

파라미터	설명
-server <server-name>	서버의 이름이다.
-module <module-id>	타이머를 취소할 모듈의 ID이다.
-timer <timer-id>	취소할 타이머의 ID이다.

- 예제

```
[MASTER]domain1.adminServer>cancel-ejb-timer -server server1 -timer 502
Successfully canceled.
```

#### 4.2.7.2. ejb-timer-info

특정 서버의 EJB 타이머 정보를 확인한다.

- alias

ejbtimerinfo, timerinfo, scheduler

- 사용법

```
ejb-timer-info -server <server-name>
                [-module <module-id> | -all]
                [-detail]
```

- 파라미터

파라미터	설명
-server <server-name>	서버의 이름이다.
[-module <module-id>]	타이머 정보를 확인할 EJB 모듈의 이름이다.
[-all]	모든 EJB 모듈의 타이머 정보를 확인한다.
[-detail]	타이머의 정보를 상세하게 출력한다.

- 예제

```
[MASTER]domain1.adminServer>ejb-timer-info -server server1 -all -detail
Persistent Timer List: server[server1]
Module ID: Timer
=====
+---+-----+-----+-----+-----+-----+
| ID | BEAN   | METHOD | SCHEDULE | INFO | NEXT |
+---+-----+-----+-----+-----+-----+
| 52 | Persistence | print() | sec=*,min=*,hour=* |  | 2016-08-29 (Mon) PM |
```



```
| | TestBean | | ,dayOfMonth=*,dayOf | | 03:50:34 KST | | | |
| | | | | Week=*,month=*,year | | | |
| | | | | |=* | | | |
+---+-----+-----+-----+-----+-----+
=====
```

#### 4.2.7.3. modify-active-management

특정 서버의 Active management를 변경한다.

- alias

modifyam

- 사용법

```
modify-active-management -server <server-name>
                        [-mbt,--maxBlockedThread <max-blocked-thread>]
                        [-mit,--maxIdleTime <max-idle-time>]
                        [-f,--forceLock]
                        [-detail]
```

- 파라미터

파라미터	설명
-server <server-name>	서버의 이름이다.
[-mbt,--maxBlockedThread <max-blocked-thread>]	<b>[동적 변경]</b> 블록된 Thread 개수의 최댓값이다.
[-mit,--maxIdleTime <max-idle-time>]	<b>[동적 변경]</b> 블록되었다고 판단할 idle 시간의 최댓값이다.
[-f,--forceLock]	설정 변경을 강제로 진행하여 반영한다.
[-detail]	동적 변경 결과에 대한 상세한 정보가 출력된다.

- 예제

```
[MASTER]domain1.adminServer>modify-active-management -server server1 -mbt 5 -mit 300001
Successfully performed the MODIFY operation for active-management for the server (server1)..
Check the results using "show-active-management -server server1"
```

#### 4.2.7.4. modify-check-resolution

특정 서버의 Resolution을 변경한다.

- alias

modifyresolution, ejbengineresolution

- 사용법

```
modify-check-resolution -server <server-name>
                        [-r,--resolution <resolution>]
                        [-f,--forceLock]
                        [-detail]
```

- 파라미터

파라미터	설명
-server <server-name>	서버의 이름이다.
[-r,--resolution <resolution>]	<b>[동적 변경]</b> 변경할 EJB 엔진의 Resolution이다.
[-f,--forceLock]	설정 변경을 강제로 진행하여 반영한다.
[-detail]	동적 변경 결과에 대한 상세한 정보가 출력된다.

- 예제

```
[MASTER]domain1.adminServer>modify-check-resolution -server server1 -r 3000001
Successfully performed the MODIFY operation for The EJB engine resolution of the server
(server1)..
Check the results using "show-check-resolution -server server1"
```

#### 4.2.7.5. show-active-management

특정 서버의 Active management 정보를 확인한다.

- alias

showam

- 사용법

```
show-active-management -server <server-name>
```

- 파라미터

파라미터	설명
-server <server-name>	서버의 이름이다.

- 예제

```
[MASTER]domain1.adminServer>show-active-management -server server1
Shows the current configuration.
active-management for the server (server1).
=====
+-----+-----+
| Max Blocked Thread          | 5      |
| Max Idle Time               | 300001 |
+-----+-----+
=====
```

#### 4.2.7.6. show-check-resolution

특정 서버의 Resolution 정보를 확인한다.

- alias

showresolution

- 사용법

```
show-check-resolution -server <server-name>
```

- 파라미터

파라미터	설명
-server <server-name>	서버의 이름이다.

- 예제

```
[MASTER]domain1.adminServer>show-check-resolution -server server1
Shows the current configuration.
The EJB engine resolution of the server (server1).
=====
+-----+-----+
| Resolution          | 300001 |
+-----+-----+
=====
```

#### 4.2.8. 웹 엔진 관련 명령어

웹 엔진 관련 명령어는 크게 다음과 같이 2가지로 구분할 수 있다.

- 모니터링 및 제어

deploy되어 있는 웹 컨텍스트의 상태를 확인하거나 제어하기 위한 명령어, 웹 엔진의 상태를 확인하거나 제어하기 위한 명령어들이다.

명령어	설명
<a href="#">resume-web-component</a>	일시 중지된 웹 컴포넌트(서블릿)를 재시작한다.
<a href="#">suspend-web-component</a>	웹 컴포넌트(서블릿)를 일시중지한다.
<a href="#">reload-web-context</a>	특정 컨텍스트를 동적으로 reload한다.
<a href="#">show-web-engine-configuration</a>	웹 엔진 관련 설정 정보를 조회한다.
<a href="#">show-web-statistics</a>	웹 엔진 관련 각종 통계 자료를 조회한다.
<a href="#">clear-thread-local</a>	web-connection thread pool에 저장된 thread local 정보의 삭제 여부를 설정한다.
<a href="#">clear-web-statistics</a>	컨텍스트, 서블릿 등의 요청 통계 자료를 초기화한다.
<a href="#">show-request-processing-flow</a>	URL 패턴이나 특정 호스트명으로 들어온 요청의 처리 흐름을 조회한다.
<a href="#">precompile-jsp</a>	deploy된 웹 컨텍스트의 JSP 파일들을 요청처리 전에 미리 컴파일한다.
<a href="#">notify-auto-scale</a>	Web Server나 Jboss Managed Server의 추가/삭제를 알려준다.

- 설정 변경 관련 명령어

웹 엔진에 속한 설정을 추가, 삭제, 수정할 수 있는 명령어들이다. 명령에 따라 동적으로 반영되지 않는 설정이 있다. 이러한 명령의 설정 변경 사항들은 XML에는 적용하지만 실제 운영 중인 서비스(메모리 영역)에 적용하지 않는다. 실제로 서비스에 적용하려면 서버를 재시작해야 한다.

명령어	설명
<a href="#">add-web-cookie-policy</a>	Cookie Policy 설정을 추가한다.
<a href="#">modify-web-cookie-policy</a>	Cookie Policy 설정을 변경한다.
<a href="#">remove-web-cookie-policy</a>	Cookie Policy 설정을 삭제한다.
<a href="#">add-web-encoding</a>	Character Set Encoding 설정을 추가한다.
<a href="#">modify-web-encoding</a>	Character Set Encoding 설정을 변경한다.공통
<a href="#">remove-web-encoding</a>	Character Set Encoding 설정을 삭제한다.
<a href="#">add-web-properties</a>	웹 엔진의 프로퍼티 설정을 추가한다.
<a href="#">modify-web-properties</a>	웹 엔진의 프로퍼티 설정을 변경한다.
<a href="#">remove-web-properties</a>	웹 엔진의 프로퍼티 설정을 삭제한다.
<a href="#">add-response-header</a>	HTTP Response Custom Header 설정을 추가한다.
<a href="#">modify-response-header</a>	HTTP Response Custom Header 설정을 변경한다.
<a href="#">remove-response-header</a>	HTTP Response Custom Header 설정을 삭제한다.
<a href="#">modify-jsp-engine</a>	JSP 엔진 설정을 변경한다.
<a href="#">add-virtual-host</a>	가상 호스트를 추가한다.
<a href="#">modify-virtual-host</a>	가상 호스트의 일부 설정을 변경한다.
<a href="#">remove-virtual-host</a>	가상 호스트를 삭제한다.
<a href="#">add-valve</a>	Valve를 추가한다.

명령어	설명
<code>remove-valve</code>	Valve를 삭제한다.
<code>set-valve-property</code>	Valve의 프로퍼티 설정을 추가한다.
<code>remove-valve-property</code>	Valve의 프로퍼티 설정을 삭제한다.
<code>add-webtob-connector</code>	WebtoB 커넥터를 추가한다.
<code>modify-webtob-connector</code>	WebtoB 커넥터의 일부 설정을 변경한다.
<code>remove-webtob-connector</code>	WebtoB 커넥터를 삭제한다.
<code>show-webtob-connector</code>	WebtoB 커넥터 정보를 조회한다.
<code>add-backup-webtob</code>	Backup WebtoB를 추가한다.
<code>remove-backup-webtob</code>	Backup WebtoB를 삭제한다.
<code>add-tmax-connector</code>	Tmax 커넥터를 추가한다.
<code>modify-tmax-connector</code>	Tmax 커넥터의 일부 설정을 변경한다.
<code>remove-tmax-connector</code>	Tmax 커넥터를 삭제한다.
<code>add-ajp-listener</code>	AJP 리스너를 추가한다.
<code>add-http-listener</code>	HTTP 리스너를 추가한다.
<code>add-tcp-listener</code>	TCP 리스너를 추가한다.
<code>modify-web-listener</code>	HTTP, TCP, AJP 리스너의 일부 설정을 변경한다.
<code>remove-web-listener</code>	HTTP, TCP, AJP 리스너를 삭제한다.
<code>modify-web-engine-configuration</code>	웹 엔진의 일부 설정을 동적으로 변경한다.

#### 4.2.8.1. 공통적용 사항

다음은 웹 엔진 관련 명령어에서 공통적으로 적용되는 옵션에 대한 설명이다. 아래 옵션 중 하나만 설정될 수 있다.

- `[-cluster <cluster-name>]` 옵션
  - 각 명령어가 적용될 클러스터를 지정할 때 사용하는 옵션이다. 옵션을 설정하지 않으면 현재 접속된 서버의 웹 엔진으로 적용된다.
  - 각 명령어의 옵션 설명에서 이 옵션에 대한 설명은 생략한다. 단, `precompile-jsp`는 이 옵션을 제공하지 않는다.
- `[-server <server-name>]` 옵션
  - 각 명령어가 적용될 서버를 지정할 때 사용하는 옵션이다. 옵션을 설정하지 않으면 현재 접속된 서버의 웹 엔진으로 적용된다.
  - 각 명령어의 옵션 설명에서 이 옵션에 대한 설명은 생략한다.



1. 서버에 즉시 반영이 안되는 설정의 경우 즉, Pending된 설정이 최초의 설정으로 재설정되는 명령어를 수행하면, 서버의 재시작이 필요하다는 안내문(Restart the Server)이 출력되지 않는다.

2. JEUS Master Server의 경우는 위 옵션 중 하나는 반드시 필요하다. 그렇지만 JEUS Master Server가 아닌 일반 관리 서버에서 해당 명령어를 실행할 경우에는 아래 옵션들이 없을 경우 접속한 서버를 기반으로 명령어가 수행된다.

#### 4.2.8.2. add-ajp-listener

AJP 리스너를 추가한다.

- 관련 스키마

web-engine.xsd - web-engine/web-connections/ajp13-listener

- alias

addajpl

- 사용법

```
add-ajp-listener -cluster <cluster-name> | -server <server-name>
                  [-f, --forceLock]
                  -name <web-connection-name>
                  -slref <server-listener-ref-name>
                  -tmin <minimum-thread-num>
                  [-tmax <maximum-thread-num>]
                  [-tidle <max-idle-time>]
                  [-qs <max-queue-size>]
```

- 파라미터

파라미터	설명
[-f, --forceLock]	설정 변경을 강제로 진행하여 반영한다.
-name <web-connection-name>	리스너의 이름이다.
-slref <server-listener-ref-name>	서버에 등록된 서비스 리스너의 이름이다.
-tmin <minimum-thread-num>	Thread Pool의 최소 Thread 수이다.
[-tmax <maximum-thread-num>]	Thread Pool의 최대 Thread 수이다. 설정하지 않으면 tmin와 같은 값을 갖는다.
[-tidle <max-idle-time>]	Thread Pool에서 제거되기 전에 Idle 상태로 머물러 있는 Worker Thread의 최대 시간이다.
[-qs <max-queue-size>]	Thread Pool에서 처리할 작업들을 저장할 Queue의 Size이다.

- 예제

```
add-ajp-listener -name ajp1 -server server1 -slref ajp-server -tmin 10
```

### 4.2.8.3. add-backup-webtob

Backup WebtoB을 추가한다.

- 관련 스키마

web-engine.xsd - web-engine/web-connections/webtob-connector/webtob-backup

- alias

addbackupwebtob

- 사용법

```
add-backup-webtob -cluster <cluster-name> | -server <server-name>
                  [-f, --forceLock]
                  -name <web-connection-name>
                  [-addr <server-address>]
                  -port <server-port> | -dsocket
                  [-wbhome <webtob-home>]
                  [-ipcport <ipc-base-port>]
```

- 파라미터

파라미터	설명
[-f, --forceLock]	설정 변경을 강제로 진행하여 반영한다.
-name <web-connection-name>	Backup WebtoB를 추가할 WebtoB 커넥터의 이름이다.
[-addr <server-address>]	Backup WebtoB의 서버 주소이다. -port 사용할 때 필수 옵션이다.
-port <server-port>	Backup WebtoB의 서버 포트 번호이다.
-dsocket	Unix 도메인 소켓(Pipe)을 이용하여 JEUS와 WebtoB 사이에 통신을 한다.
[-wbhome <webtob-home>]	JEUS와 WebtoB가 같은 장비 안에 설치되어 있을 경우 Backup WebtoB가 설치된 절대 경로이다.
[-ipcport <ipc-base-port>]	Windows 환경에서 WebtoB의 내부 프로세스간 통신을 할 때 사용하는 기본 포트 번호이다.

- 예제

```
add-backup-webtob -server server1 -name webtob1 -port 9999 -addr localhost
```

### 4.2.8.4. add-http-listener

HTTP 리스너를 추가한다.

- 관련 스키마

web-engine.xsd - web-engine/web-connections/http-listener

- alias

addhttp

- 사용법

```
add-http-listener -cluster <cluster-name> | -server <server-name>
                  [-f, --forceLock]
                  -name <web-connection-name>
                  -slref <server-listener-ref-name>
                  -tmin <minimum-thread-num>
                  [-tmax <maximum-thread-num>]
                  [-tidle <max-idle-time>]
                  [-qs <max-queue-size>]
                  [-http2]
```

- 파라미터

파라미터	설명
[-f, --forceLock]	설정 변경을 강제로 진행하여 반영한다.
-name <web-connection-name>	리스너의 이름이다.
-slref <server-listener-ref-name>	서버에 등록된 서비스 리스너의 이름이다.
-tmin <minimum-thread-num>	Thread Pool의 최소 Thread 수이다.
[-tmax <maximum-thread-num>]	Thread Pool의 최대 Thread 수이다. 설정하지 않으면 tmin와 같은 값을 갖는다.
[-tidle <max-idle-time>]	Thread Pool에서 제거되기 전에 Idle 상태로 머물러 있는 Worker Thread의 최대 시간이다.
[-qs <max-queue-size>]	Thread Pool에서 처리할 작업들을 저장할 Queue의 Size이다.
[-http2]	HTTP/2를 사용한다.

- 예제

```
add-http-listener -name http1 -server server1 -slref http-server -tmin 5 -tmax 10 -http2
```

#### 4.2.8.5. add-response-header

웹 엔진의 Response Header 설정을 추가한다. 추가되는 Response Header는 대소문자를 구별하지 않는다. 단, XML에만 반영하며 실제로 서비스에 적용하려면 서버를 재시작해야 한다.



- 관련 스키마

web-engine.xsd - web-engine/response-header

- alias

addwebrh

- 사용법

```
add-response-header -cluster <cluster-name> | -server <server-name>
                    [-f, --forceLock]
                    [-n, --name <header-name>]
                    [-v, --value <header-value>]
```

- 파라미터

파라미터	설명
[-f, --forceLock]	설정 변경을 강제로 진행하여 반영한다.
[-n, --name <header-name>]	추가할 HTTP Response Header의 이름을 설정한다. 입력값은 대소문자를 구별하지 않는다. 같은 이름의 Header는 1개만 허용된다.
[-v, --value <header-value>]	추가할 HTTP Response Header의 값을 설정한다. 입력값은 대소문자를 구별하지 않는다.

- 예제

```
add-response-header -server server1 -n testHeader -v testValue
```

#### 4.2.8.6. add-tcp-listener

TCP 리스너를 추가한다. 단, TCP 리스너는 XML에만 반영하며 TCP 리스너를 실제로 서비스에 적용하려면 서버를 재시작해야 한다.

- 관련 스키마

web-engine.xsd - web-engine/web-connections/tcp-listener

- alias

addtcppl

- 사용법

```
add-tcp-listener -cluster <cluster-name> | -server <server-name>
                 [-f, --forceLock]
                 -name <web-connection-name>
```

```
-slref <server-listener-ref-name>
-dcc <dispatcher-config-class>
-tmin <minimum-thread-num>
[-tmax <maximum-thread-num>]
[-tidle <max-idle-time>]
[-qs <max-queue-size>]
```

#### • 파라미터

파라미터	설명
[-f, --forceLock]	설정 변경을 강제로 진행하여 반영한다.
-name <web-connection-name>	리스너의 이름이다.
-slref <server-listener-ref-name>	서버에 등록된 서비스 리스너의 이름이다.
-dcc <dispatcher-config-class>	jeus.servlet.tcp.TCPDispatcherConfig 인터페이스를 구현한 클래스 이름이다.
-tmin <minimum-thread-num>	Thread Pool의 최소 Thread 수이다.
[-tmax <maximum-thread-num>]	Thread Pool의 최대 Thread 수이다. 설정하지 않으면 tmin와 같은 값을 갖는다.
[-tidle <max-idle-time>]	Thread Pool에서 제거되기 전에 Idle 상태로 머물러 있는 Worker Thread의 최대 시간이다.
[-qs <max-queue-size>]	Thread Pool에서 처리할 작업들을 저장할 Queue의 Size이다.

#### • 예제

```
add-tcp-listener -name tcp1 -slref tcp -server server1 -dcc tcp.DispatcherConfigImpl -tmin 5
-tmax 10
```

#### 4.2.8.7. add-tmax-connector

Tmax 커넥터를 추가한다. 단, XML에만 반영하며 실제로 서비스에 적용하려면 서버를 재시작해야 한다.

##### • 관련 스키마

web-engine.xsd - web-engine/web-connections/tmax-connector

##### • alias

addtmaxcon

##### • 사용법

```
add-tmax-connector -cluster <cluster-name> | -server <server-name>
[-f, --forceLock]
```

```

-name <web-connection-name>
-addr <server-address>
-port <server-port>
-svrg <server-group-name>
-svr <server-name>
-dcc <dispatcher-config-class>
-tmin <minimum-thread-num>
[-tmax <maximum-thread-num>]
[-tidle <max-idle-time>]
[-qs <max-queue-size>]

```

#### • 파라미터

파라미터	설명
[-f, --forceLock]	설정 변경을 강제로 진행하여 반영한다.
-name <web-connection-name>	Tmax 커넥터의 이름이다.
-addr <server-address>	Tmax 서버 주소이다.
-port <server-port>	Tmax 서버 포트 번호이다.
-svrg <server-group-name>	Tmax가 속한 서버 그룹 이름이다.
-svr <server-name>	Tmax의 서버 이름이다.
-dcc <dispatcher-config-class>	jeus.servlet.tcp.TCPDispatcherConfig 인터페이스를 구현한 클래스 이름이다.
-tmin <minimum-thread-num>	Thread Pool의 최소 Thread 수이다.
[-tmax <maximum-thread-num>]	Thread Pool의 최대 Thread 수이다. 설정하지 않으면 tmin와 같은 값을 갖는다.
[-tidle <max-idle-time>]	Thread Pool에서 제거되기 전에 Idle 상태로 머물러 있는 Worker Thread의 최대 시간이다.
[-qs <max-queue-size>]	Thread Pool에서 처리할 작업들을 저장할 Queue의 Size이다.

#### • 예제

```

add-tmax-connector -name tmax1 -addr 5.0.1.2 -port 1024
-svrg group1 -svr server1 -server server1 -dcc sevice.DispatcherConfig

```

### 4.2.8.8. add-virtual-host

가상 호스트를 추가한다.

#### • 관련 스키마

web-engine.xsd - web-engine/virtual-host

- alias

addvh

- 사용법

```
add-virtual-host -cluster <cluster-name> | -server <server-name>
                  [-f, --forceLock]
                  <virtual-host-name>
                  -list <host-name-list>
                  [-tmin <minimum-thread-num>]
                  [-tmax <maximum-thread-num>]
                  [-tidle <max-idle-time>]
                  [-qs <max-queue-size>]
```

- 파라미터

파라미터	설명
[-f, --forceLock]	설정 변경을 강제로 진행하여 반영한다.
<virtual-host-name>	가상 호스트의 이름이다.
-list <host-name-list>	가상 호스트에 등록할 도메인 이름 또는 IP 주소이다.
[-tmin <minimum-thread-num>]	Thread Pool의 최소 Thread 수이다.
[-tmax <maximum-thread-num>]	Thread Pool의 최대 Thread 수이다.
[-tidle <max-idle-time>]	Thread Pool에서 제거되기 전에 Idle 상태로 머물러 있는 Worker Thread의 최대 시간이다.
[-qs <max-queue-size>]	Thread Pool에서 Queue에 대기할 수 있는 최대 요청값이다.

- 예제

```
add-virtual-host -server server1 host2 -list www.foo.com,192.168.1.2 -tmin 1 -tmax 10 -tidle
20000
```

#### 4.2.8.9. add-valve

Valve를 추가한다.

- 관련 스키마

web-engine.xsd - pipeline/valve

- alias

addvlv

- 사용법

```
add-valve -cluster <cluster-name> | -server <server-name>
        [-f, --forceLock]
        -cn, --classname <class-name>
        -p, --properties <property>
        [-vh <virtual-host-name>]
```

- 파라미터

파라미터	설명
[-f, --forceLock]	설정 변경을 강제로 진행하여 반영한다.
-cn, --classname <class-name>	서버, 가상호스트에 적용할 valve class를 지정한다. 제우스의 ValveBase를 상속받은 클래스로 지정해야 한다.
-p, --properties <properties>	valve의 프로퍼티를 설정한다. <ul style="list-style-type: none"> <li>• 한 개의 프로퍼티는 "key=value" 형태로 "="를 이용하여 설정한다.</li> <li>• 한 개 이상의 프로퍼티를 수정할 경우에는 각 프로퍼티를 콤마(,)로 구분하여 설정한다.</li> </ul>
[-vh <virtual-host-name>]	valve를 추가할 가상 호스트의 이름이다. 가상 호스트 이름을 설정 안하면 서버 단에서 valve가 추가된다.

- 예제

```
add-valve -server server1 -cn UserValveBase
-p jeus.servlet.jsp.compile-java-source-concurrently=false,userProperty=myValue
```

#### 4.2.8.10. add-web-cookie-policy

웹 엔진의 HTTP Cookie Policy 설정을 추가한다. 단, XML에만 반영하며 실제로 서비스에 적용하려면 서버를 재시작해야 한다.

- 관련 스키마

web-engine.xsd - web-engine/cookie-policy

- alias

addcookie

- 사용법

```
add-web-cookie-policy -cluster <cluster-name> | -server <server-name>
        [-f, --forceLock]
        [-auer, --apply-url-encoding-rule <apply-url-encoding-rule>]
```

```
[ -enc, --charset-encoding < charset-encoding > ]  
[ -vh < virtual-host-name > ]
```

- 파라미터

파라미터	설명
[ -f, --forceLock ]	설정 변경을 강제로 진행하여 반영한다.
[ -auer, --apply-url-encoding-rule < apply-url-encoding-rule > ]	URL Encoding Rule 적용 여부를 설정한다.
[ -enc, --charset-encoding < charset-encoding > ]	URL Encoding Rule을 적용할 때 사용하는 Charset Encoding이다. 설정하지 않을 경우 Request Encoding 값을 따른다.
[ -vh < virtual-host-name > ]	가상 호스트를 지정한다.

- 예제

```
add-web-cookie-policy -server server1 -auer true  
  
add-web-cookie-policy -server server1 -enc EUC-KR
```

#### 4.2.8.11. add-web-encoding

웹 엔진의 Encoding을 추가한다. 단, XML에만 반영하며 실제로 서비스에 적용하려면 서버를 재시작해야 한다(이미 설정되어 있던 Encoding을 삭제하고 추가하며, Encoding이 변한 경우 restart되어야 서버에 적용이 된다).

- 관련 스키마

web-engine.xsd - web-engine/encoding

- alias

addenc

- 사용법

```
add-web-encoding -cluster < cluster-name > | -server < server-name >  
    [ -f, -forceLock ]  
    < charset-encoding >  
    -co | -default | -forced  
    [ -url ]  
    [ -req ]  
    [ -res ]  
    [ -vh < virtual-host-name > ]
```

- 파라미터

파라미터	설명
[-f, --forceLock]	설정 변경을 강제로 진행하여 반영한다.
<charset-encoding>	설정할 Character Set Encoding 이름이다. 잘못된 값이 입력되면 오류가 출력되고 명령이 실패한다.
-co   -default   -forced	입력된 Encoding을 default로 설정할지 forced로 설정할지 결정한다. request encoding인 경우 두 설정 외에 client-override를 설정할 수 있다.
[-url]	Request Url의 Character Set Encoding을 설정한다.
[-req]	Request의 Character Set Encoding을 설정한다.
[-res]	Response의 Character Set Encoding을 설정한다.
[-vh <virtual-host-name>]	가상 호스트를 지정한다.

- 예제

```
add-web-encoding -server server1 -default -url UTF-8

addenc -server server1 UTF-8 -forced -req -res
```

#### 4.2.8.12. add-web-properties

웹 엔진의 프로퍼티 설정을 추가한다. 추가되는 프로퍼티는 대소문자를 구별한다. 단, XML에만 반영하며 실제로 서비스에 적용하려면 서버를 재시작해야 한다.

- 관련 스키마

web-engine.xsd - web-engine/properties

- alias

addwebpr

- 사용법

```
add-web-properties -cluster <cluster-name> | -server <server-name>
                    [-f, --forceLock]
                    -p, --properties <properties>
                    [-vh <virtual-host-name>]
```

- 파라미터

파라미터	설명
[-f, --forceLock]	설정 변경을 강제로 진행하여 반영한다.

파라미터	설명
-p,--properties <properties>	추가할 웹 엔진의 프로퍼티를 설정한다. <ul style="list-style-type: none"> <li>한 개의 프로퍼티는 "key=value" 형태로 "="를 이용하여 설정한다.</li> <li>한 개 이상의 프로퍼티를 추가할 경우에는 각 프로퍼티를 콤마(,)로 구분하여 설정한다.</li> </ul>
[-vh <virtual-host-name>]	가상 호스트를 지정한다.

- 예제

```
add-web-properties -server server1 -p jeus.servlet.jsp.modern=false

add-web-properties -server server1 -p jeus.servlet.jsp.compile-java-source-
concurrently1=false,jeus.servlet.jsp.assure-utf8-file-encoding-detection=true
```

#### 4.2.8.13. add-webtob-connector

WebtoB 커넥터를 추가한다. 추가한 WebtoB 커넥터는 실제 서비스에 사용 가능하다. 단, 추가 후에는 WebtoB 연결 주소, 포트 등을 변경할 수 없으므로 주의깊게 추가하기 바란다.

- 관련 스키마

web-engine.xsd - web-engine/web-connections/webtob-connector

- alias

addwebtobcon

- 사용법

```
add-webtob-connector -cluster <cluster-name> | -server <server-name>
    [-f, --forceLock]
    -name <web-connection-name>
    -conn <connection-count>
    -regid <registration-id>
    -tmin <minimum-thread-num>
    [-tmax <maximum-thread-num>]
    [-tidle <max-idle-time>]
    [-qs <max-queue-size>]
    [-hth <hth-count>]
    [-useNio <true | false>]
    [-ver <wjp-version>]
    [-addr <server-address>]
    [-port <server-port> | -dsocket]
    [-wbhome <webtob-home>]
    [-ipcport <ipc-base-port>]
    [-sndbuf <send-buffer-size>]
```



`[-rcvbuf <receive-buffer-size>]`

- 파라미터

파라미터	설명
<code>[-f, --forceLock]</code>	설정 변경을 강제로 진행하여 반영한다.
<code>-name &lt;web-connection-name&gt;</code>	WebtoB 커넥터의 이름이다.
<code>-conn &lt;connection-count&gt;</code>	WebtoB와 연결할 Connection 개수이다.
<code>-tmin &lt;minimum-thread-num&gt;</code>	Thread Pool의 최소 Thread 수이다.
<code>[-tmax &lt;maximum-thread-num&gt;]</code>	Thread Pool의 최대 Thread 수이다. 설정하지 않으면 tmin와 같은 값을 갖는다.
<code>[-tidle &lt;max-idle-time&gt;]</code>	Thread Pool에서 제거되기 전에 Idle 상태로 머물러 있는 Worker Thread의 최대 시간이다.
<code>[-qs &lt;max-queue-size&gt;]</code>	Thread Pool에서 처리할 작업들을 저장할 Queue의 Size이다.
<code>[-hth &lt;hth-count&gt;]</code>	hth process의 개수이다.
<code>[-useNio &lt;true   false&gt;]</code>	NIO 방식의 사용 여부이다.
<code>[-ver &lt;wjp-version&gt;]</code>	WebtoB와 통신할 때 사용하는 프로토콜 버전을 나타낸다. 1 또는 2를 입력한다.
<code>[-addr &lt;server-address&gt;]</code>	WebtoB 서버 주소이다. -port 사용할 때 필수 옵션이다.
<code>[-port &lt;server-port&gt;]</code>	WebtoB 서버 포트 번호이다.
<code>-regid &lt;registration-id&gt;</code>	WebtoB와의 등록 ID이다.
<code>[-dsocket]</code>	Unix 도메인 소켓(Pipe)을 이용하여 JEUS와 WebtoB 사이에 통신을 한다.
<code>[-wbhome &lt;webtob-home&gt;]</code>	JEUS와 WebtoB가 같은 장비 안에 설치되어 있을 경우 WebtoB가 설치된 절대 경로이다.
<code>[-ipcport &lt;ipc-base-port&gt;]</code>	Windows 환경에서 WebtoB의 내부 프로세스간 통신을 할 때 사용하는 기본 포트 번호이다.
<code>[-sndbuf &lt;send-buffer-size&gt;]</code>	TCP 소켓이나 Unix 도메인 소켓의 SO_SNDBUF를 설정한다. 0이면 OS의 기본값을 사용한다.
<code>[-rcvbuf &lt;receive-buffer-size&gt;]</code>	TCP 소켓이나 Unix 도메인 소켓의 SO_RCVBUF를 설정한다. 0이면 OS의 기본값을 사용한다.

- 예제

```
add-webtob-connector -server server1 -name webtob1 -conn 10 -tmin 10 -hth 2 -regid MyGroup -port 9999 -addr localhost
```

#### 4.2.8.14. clear-thread-local

모든 web-connection thread pool에 저장된 thread local 정보의 삭제 여부를 설정한다. 기본적으로는 삭제하지 않으며, 삭제할 경우 각 thread가 task를 실행하기 전에 매번 체크하고 삭제한다. 해당 기능은 on/off로 조정할 수 있다.

- 관련 스키마

web-engine.xsd

- alias

clrtl, cltl

- 사용법

```
clear-thread-local -cluster <cluster-name> | -server <server-name>
                    [-f, --forceLock]
                    -wc <web-connection-name>
                    -off | -on
```

- 파라미터

파라미터	설명
[-f, --forceLock]	설정 변경을 강제로 진행하여 반영한다.
-wc <web-connection-name>]	web-connection의 이름이다.
-off   -on	<ul style="list-style-type: none"><li>• -off : thread local을 지우지 않는다.</li><li>• -on : thread local을 지운다.</li></ul>

- 예제

```
clear-thread-local -server server1 -wc http-listener -on
```

#### 4.2.8.15. clear-web-statistics

지정한 Entity(컨텍스트, 서블릿)에 대하여 요청 통계 자료를 초기화하여 카운터 기록을 재설정한다.

- 관련 스키마

web-engine.xsd

- alias

clrstat, clst, clrwebstat

- 사용법

```
clear-web-statistics -server <server-name>
                    [-ctx,--context <context-name>]
                    [-servlet <servlet-name>]
```

- 파라미터

파라미터	설명
[-ctx <context-name>]	요청 횟수 및 처리 시간 기록, 작업 스레드에 대한 정보들을 삭제할 컨텍스트이다.
[-servlet <servlet-name>]	요청 횟수 및 처리 시간 기록을 삭제할 서블릿이다.

- 예제

```
clear-web-statistics -server server1 -ctx ctx1
```

#### 4.2.8.16. modify-jsp-engine

웹 엔진의 JSP 엔진 설정을 수정한다. 단, XML에만 반영하며 실제로 서비스에 적용하려면 서버를 재시작해야 한다.

- 관련 스키마

web-engine.xsd - web-engine/jsp-engine

- alias

modjsp

- 사용법

```
modify-jsp-engine -cluster <cluster-name> | -server <server-name>
  [-f, --forceLock]
  [-rjwd,--remove-jsp-work-dir | -jwd,--jsp-work-dir <jsp-work-dir>]
  [-javac,--java-compiler <java-compiler>]
  [-cod,--compile-output-dir <compile-output-dir> |
  -rcod,--remove-compile-output-dir]
  [-rcopt,--remove-compile-option | -copt,--compile-option <compile-option>]
  [-cij,--check-included-jspfile <check-included-jspfile>]
  [-kg,--keep-generated <keep-generated>]
  [-gjr,--graceful-jsp-reloading <graceful-jsp-reloading>]
  [-gjrp,--graceful-jsp-reloading-period <graceful-jsp-reloading-period>]
  [-umc,--use-in-memory-compilation <use-in-memory-compilation>]
```

- 파라미터

파라미터	설명
<code>[-f, --forceLock]</code>	설정 변경을 강제로 진행하여 반영한다.
<code>[-rjwd, --remove-jsp-work-dir   -jwd, --jsp-work-dir &lt;jsp-work-dir&gt;]</code>	<p><code>-jwd, --jsp-work-dir &lt;jsp-work-dir&gt;</code>을 이용하여 JSP를 Java로 생성한 파일과 그 소스 파일을 컴파일한 클래스 파일들이 저장되는 위치를 설정한다. 이때 설정값은 절대 경로로 입력해야 한다.</p> <p><code>-rjwd, --remove-jsp-work-dir</code>을 이용하여 설정을 삭제할 수 있다. 삭제할 경우 엔진의 내부 기본값이 사용된다.</p>
<code>[-javac, --java-compiler &lt;java-compiler&gt;]</code>	<p>JSP의 Java 소스를 서블릿 클래스 파일로 컴파일하기 위한 Java 컴파일러를 지정한다.</p> <p>다음 중에 하나를 설정한다.</p> <ul style="list-style-type: none"> <li>• java6 (기본값)</li> <li>• sun.tools.javac</li> <li>• javac</li> <li>• com.sun.tools.javac</li> <li>• eclipse</li> </ul> <p>각 값들에 대한 설명은 스키마를 참고한다. 기본값으로 설정하는 것이 가장 효율적이기 때문에 별도의 설정을 하지 않는 것을 권장한다.</p>
<code>[-cod, --compile-output-dir &lt;compile-output-dir&gt;   -rcod, --remove-compile-output-dir]</code>	<p><code>-cod, --compile-output-dir &lt;compile-output-dir&gt;</code>를 이용하여 JSP 파일에 의해 생성된 클래스 파일들을 JSP Work Dir 디렉터리가 아닌 다른 디렉터리에 저장할 수 있는 설정할 수 있다. 설정하지 않으면 클래스 파일들은 JSP Work Dir 디렉터리에 위치한다.</p> <p><code>-rcod, --remove-compile-output-dir</code>을 이용하여 설정을 삭제할 수 있다. 삭제할 경우 엔진의 내부 기본값이 사용된다.</p>
<code>[-rcopt, --remove-compile-option   -copt, --compile-option &lt;compile-option&gt;]</code>	<p><code>-copt, --compile-option &lt;compile-encoding&gt;</code>을 이용하여 서블릿 컴파일러로 사용되는 옵션을 설정한다.</p> <p><code>-rcopt, --remove-compile-option</code>을 이용하여 설정한 값을 삭제할 수 있다.</p>
<code>[-cij, --check-included-jspfile &lt;check-included-jspfile&gt;]</code>	JSP 파일의 변경 여부를 확인할 때 include된 JSP 파일들, Tag 파일들의 변경 여부를 확인해서 변경된 경우 해당 JSP 파일을 재컴파일하는 기능의 사용 여부를 설정한다.
<code>[-kg, --keep-generated &lt;keep-generated&gt;]</code>	JSP 페이지로부터 생성된 Java 소스 파일을 컴파일하여 서블릿 클래스 파일을 생성한 이후에 Java 소스 파일의 저장 여부를 결정한다.
<code>[-gjr, --graceful-jsp-reloading &lt;graceful-jsp-reloading&gt;]</code>	<p>JSP 소스와 컴파일된 클래스 파일이 공유되고 있고, 하나의 JVM에서 exclusive하고 JSP 파일을 배치 형태의 Java 파일로 변환하여 컴파일할 것을 설정한다.</p> <p>JSP Work Dir 설정에 의해 공유되는 폴더가 지정되어 있는 경우에 한해서 동작한다.</p>

파라미터	설명
[ <i>-gjr</i> ,--graceful-jsp-reloading-period <i>&lt;graceful-jsp-reloading-period&gt;</i> ]	Graceful JSP reloading이 동작되는 주기를 설정한다.
[ <i>-umc</i> ,--use-in-memory-compilation <i>&lt;use-in-memory-compilation&gt;</i> ]	서비스 중인 JSP 파일을 새로 컴파일해야 할 때 .java 및 .class 파일을 메모리 상에 생성해서 컴파일하는 기능을 설정한다. 단, .class 파일의 경우에는 추후 리부팅할 때에 재컴파일하는 일이 없도록 Background thread를 통해서 파일 시스템에 쓴다.  또한 <i>&lt;keep-generated&gt;</i> 가 true인 경우에는 .java 파일을 Background thread를 통해서 파일 시스템에 쓴다.

- 예제

```
modify-jsp-engine -server server1 -jwd /home/jeus/jsp
```

#### 4.2.8.17. modify-response-header

웹 엔진의 Response Header 설정을 수정한다. 수정되는 Response Header는 대소문자를 구별하지 않는다. 단, XML에만 반영하며 실제로 서비스에 적용하려면 서버를 재시작해야 한다.

- 관련 스키마

web-engine.xsd - web-engine/response-header

- alias

modwebrh

- 사용법

```
modify-response-header -cluster <cluster-name> | -server <server-name>
                        [-f, --forceLock]
                        [-n,--name <header-name>]
                        [-v,--value <header-value>]
```

- 파라미터

파라미터	설명
[ <i>-f</i> , --forceLock]	설정 변경을 강제로 진행하여 반영한다.
[ <i>-n</i> ,--name <i>&lt;header-name&gt;</i> ]	수정할 HTTP Response Header의 이름을 설정한다. 입력값은 대소문자를 구별하지 않는다. 같은 이름의 Header는 1개만 허용된다.
[ <i>-v</i> ,--value <i>&lt;header-value&gt;</i> ]	수정할 HTTP Response Header의 값을 설정한다. 입력값은 대소문자를 구별하지 않는다.

- 예제

```
modify-response-header -server server1 -n testheader -v testvalue3
```

#### 4.2.8.18. modify-tmax-connector

Tmax 커넥터의 설정 등을 변경한다.

- 관련 스키마

web-engine.xsd - web-engine/web-connections/tmax-connector

- alias

modtmaxcon

- 사용법

```
modify-tmax-connector -cluster <cluster-name> | -server <server-name>
                        [-f, --forceLock]
                        -name <web-connection-name>
                        [-tmin <minimum-thread-num>]
                        [-tmax <maximum-thread-num>]
                        [-tidle <max-idle-time>]
                        [-obuf <output-buffer-size>]
                        [-addr <server-address>]
                        [-port <server-port>]
                        [-svrg <server-group-name>]
                        [-svr <server-name>]
```

- 파라미터

파라미터	설명
[-f, --forceLock]	설정 변경을 강제로 진행하여 반영한다.
-name <web-connection-name>	Tmax 커넥터의 이름이다.
[-tmin <minimum-thread-num>]	<b>[동적 변경]</b> Thread Pool의 최소 Thread 수이다.
[-tmax <maximum-thread-num>]	<b>[동적 변경]</b> Thread Pool의 최대 Thread 수이다.
[-tidle <max-idle-time>]	Thread Pool의 Thread별 유후 대기 시간이다.
[-obuf <output-buffer-size>]	서블릿 응답 버퍼 크기다.
[-addr <server-address>]	Tmax 서버 주소이다.
[-port <server-port>]	Tmax 서버 포트 번호이다.
[-svrg <server-group-name>]	Tmax 서버의 서버 그룹 이름이다.
[-svr <server-name>]	Tmax 서버의 이름이다.

- 예제

```
modify-tmax-connector -name tmax1 -tmin 30 -server server1 -obuf 20000
```

#### 4.2.8.19. modify-virtual-host

가상 호스트의 액세스 로그 포맷 설정을 변경한다. 만약 가상 호스트의 액세스 로그가 enable 상태가 아닐 경우 실제로 서비스에 적용하려면 서버를 재시작해야 한다.

- 관련 스키마

web-engine.xsd - web-engine/virtual-host

- alias

modvh

- 사용법

```
modify-virtual-host -cluster <cluster-name> | -server <server-name>
                    [-f, --forceLock]
                    <virtual-host-name>
                    [-tmin <minimum-thread-num>]
                    [-tmax <maximum-thread-num>]
                    [-tidle <max-idle-time>]
                    [-aluph <true | false>]
                    -alf <access-log-format> |
                    -aluse <true | false> |
                    -alhnl <true | false> |
                    -alexext <access-log-excluded-extensions>]
                    [-hnrn <host-name> | -hnadd <host-name>]
                    [-ast <attach-stacktrace-on-error>]
                    [-fhn <access-log-handler-file-handler-name>]
                    [-fhp <access-log-handler-file-handler-permission>]
```

- 파라미터

파라미터	설명
[-f, --forceLock]	설정 변경을 강제로 진행하여 반영한다.
<virtual-host-name>	가상 호스트의 이름이다.
[-tmin <minimum-thread-num>]	<b>[동적 변경]</b> Thread Pool의 최소 Thread 수이다.
[-tmax <maximum-thread-num>]	<b>[동적 변경]</b> Thread Pool의 최대 Thread 수이다.
[-tidle <max-idle-time>]	Thread Pool의 Thread별 유후 대기 시간이다.

파라미터	설명
<code>[-aluph &lt;true   false&gt;]</code>	<b>[동적 변경]</b> 웹 엔진 액세스 로그에 남길지 여부를 지정한다.  (true   false 또는 t   f)
<code>[-alf &lt;access-log-format&gt;]</code>	<b>[동적 변경]</b> 액세스 로그 포맷이다. 중간에 스페이스 문자가 있는 경우 큰따옴표(" ")로 감싸야 한다. 포맷에 관한 자세한 사항은 JEUS Web Engine 안내서의 "액세스 로그 포맷 설정"을 참고한다.
<code>[-aluse &lt;true   false&gt;]</code>	<b>[동적 변경]</b> 액세스 로그 사용 여부를 지정한다.  (true   false 또는 t   f)
<code>[-alhn1 &lt;true   false&gt;]</code>	<b>[동적 변경]</b> %h 포맷에 대해 로깅할 때 IP 주소에 대한 DNS Resolution 여부를 지정한다. (true   false 또는 t   f)
<code>[-alex &lt;access-log-excluded-extensions&gt;]</code>	<b>[동적 변경]</b> 액세스 로그에 남기지 않을 확장자를 설정한다. '.gif', '.jpg'와 같이 여러 개를 설정하는 경우 콤마(,)를 사용한다. 설정하지 않는 경우 콤마만 입력한다.
<code>[-hnrm &lt;host-name&gt;]</code>	가상 호스트에 등록된 호스트 이름을 제거한다.
<code>[-hnadd &lt;host-name&gt;]</code>	가상 호스트에 새로운 호스트 이름을 등록한다.
<code>[-ast &lt;attach-stacktrace-on-error&gt;]</code>	JEUS에서 보내주는 에러 페이지에 Stack Trace를 첨부할지 여부를 설정한다.
<code>[-fhn &lt;access-log-handler-file-handler-name&gt;]</code>	Permission을 수정할 file-handler의 이름이다. -fhp를 사용할 때 필수 옵션이다.
<code>[-fhp &lt;access-log-handler-file-handler-permission&gt;]</code>	Log 파일의 접근 권한 설정인 Permission을 수정한다. -fhn을 사용할 때 필수 옵션이다.

- 예제

```
modify-virtual-host -server server1 vhost1 -tmin 1 -tmax 20 -alf "common %I"
```

#### 4.2.8.20. modify-web-cookie-policy

웹 엔진의 HTTP Cookie Policy 설정을 수정한다. 단, XML에만 반영하며 실제로 서비스에 적용하려면 서버를 재시작해야 한다.

- 관련 스키마

web-engine.xsd - web-engine/cookie-policy

- alias

modcookie

- 사용법



```

modify-web-cookie-policy -cluster <cluster-name> | -server <server-name>
    [-f, --forceLock]
    [-auer, --apply-url-encoding-rule <apply-url-encoding-rule>]
    [-enc, --charset-encoding <charset-encoding>]
    [-vh <virtual-host-name>]

```

- 파라미터

파라미터	설명
[-f, --forceLock]	설정 변경을 강제로 진행하여 반영한다.
[-auer, --apply-url-encoding-rule <apply-url-encoding-rule>]	URL Encoding Rule 적용 여부를 수정한다.
[-enc, --charset-encoding <charset-encoding>]	URL Encoding Rule을 적용할 때 사용하는 Charset Encoding을 수정한다.
[-vh <virtual-host-name>]	가상 호스트를 지정한다.

- 예제

```

modify-web-cookie-policy -server server1 -auer false

modify-web-cookie-policy -server server1 -enc UTF-8

```

#### 4.2.8.21. modify-web-encoding

웹 엔진의 Encoding을 수정한다. 단, XML에만 반영하며 실제로 서비스에 적용하려면 서버를 재시작해야 한다.

- 관련 스키마

web-engine.xsd - web-engine/encoding

- alias

modenc

- 사용법

```

modify-web-encoding -cluster <cluster-name> | -server <server-name>
    [-f, --forceLock]
    <charset-encoding>
    -co | -default | -forced
    [-url]
    [-req]
    [-res]
    [-vh <virtual-host-name>]

```

- 파라미터

파라미터	설명
[-f, --forceLock]	설정 변경을 강제로 진행하여 반영한다.
< charset-encoding >	수정할 Character Set Encoding 이름이다. 잘못된 값이 입력되면 오류가 출력되고 명령이 실패한다.
-co   -default   -forced	입력된 Encoding을 default로 설정할지 forced로 설정할지 결정한다. request encoding인 경우 두 설정 외에 client-override를 설정할 수 있다.
[-url]	Request Url의 Character Set Encoding을 수정한다.
[-req]	Request의 Character Set Encoding을 수정한다.
[-res]	Response의 Character Set Encoding을 수정한다.
[-vh <virtual-host-name>]	가상 호스트를 지정한다.

- 예제

```
modify-web-encoding -server server1 EUC-KR -forced -url
modify-web-encoding -server server1 ISO-8859-1 -default -url -res -req
```

#### 4.2.8.22. modify-web-properties

웹 엔진의 프로퍼티 설정을 수정한다. 수정되는 프로퍼티는 대소문자를 구별한다. 단, XML에만 반영하며 실제로 서비스에 적용하려면 서버를 재시작해야 한다.

- 관련 스키마

web-engine.xsd - web-engine/properties

- alias

modwebpr

- 사용법

```
modify-web-properties -cluster <cluster-name> | -server <server-name>
                        [-f, --forceLock]
                        -p, --properties <properties>
                        [-vh <virtual-host-name>]
```

- 파라미터

파라미터	설명
[-f, --forceLock]	설정 변경을 강제로 진행하여 반영한다.
-p, --properties <properties>	수정할 웹 엔진의 프로퍼티를 설정한다. <ul style="list-style-type: none"> <li>한 개의 프로퍼티는 "key=value" 형태로 "="를 이용하여 설정한다.</li> <li>한 개 이상의 프로퍼티를 수정할 경우에는 각 프로퍼티를 콤마(,)로 구분하여 설정한다.</li> </ul>
[-vh <virtual-host-name>]	가상 호스트를 지정한다.

- 예제

```
modify-web-properties -server server1
-p jeus.servlet.jsp.compile-java-source-concurrently=false
```

#### 4.2.8.23. modify-web-engine-configuration

웹 엔진 설정을 동적으로 변경한다. 이 명령어는 실제로 서비스 운영 중에도 반영되는 설정만 변경을 제공한다. 변경 가능한 정보는 모니터링 주기, 액세스 로그 포맷이다.

- 관련 스키마

web-engine.xsd - web-engine

- alias

setwebcfg, setwebconf, set-web-engine-configuration, modwebcfg, modwebconf

- 사용법

```
modify-web-engine-configuration -cluster <cluster-name> | -server <server-name>
[-f, --forceLock]
[-tpp <monitoring/check-thread-pool>]
[-crp <monitoring/check-class-reload>]
[-sp <monitoring/check-session>]
[-alf <access-log-format>]
[-aluse <true | false>]
[-alhnl <true | false>]
[-alex <access-log-excluded-extensions>]
[-ast <attach-stacktrace-on-error>]
[-att <async-timeout-min-threads>]
[-rerp | -erp <default-error-page>]
[-fhn <access-log-handler-file-handler-name>]
[-fhp <access-log-handler-file-handler-permission>]
```

- 파라미터

파라미터	설명
[-f, --forceLock]	설정 변경을 강제로 진행하여 반영한다.
[-tpp <monitoring/check-thread-pool>]	Thread Pool 체크 주기이다.
[-crp <monitoring/check-class-reload>]	웹 컨텍스트 리로드 체크 주기이다.
[-sp <monitoring/check-session>]	세션 체크 주기이다.
[-alf <access-log-format>]	<b>[동적 변경]</b> 액세스 로그 포맷이다. 중간에 스페이스 문자가 있는 경우 큰따옴표(" ")로 감싸야 한다. 포맷에 관한 자세한 사항은 JEUS Web Engine 안내서의 "액세스 로그 포맷 설정"을 참고한다.
[-aluse <true   false>]	<b>[동적 변경]</b> 액세스 로그 사용 여부를 지정한다.  (true   false 또는 t   f)
[-alhn1 <true   false>]	<b>[동적 변경]</b> %h 포맷에 대해 로깅할 때 IP 주소에 대한 DNS Resolution 여부를 결정한다. (true   false 또는 t   f)
[-alex <access-log-excluded-extensions>]	<b>[동적 변경]</b> 액세스 로그에 남기지 않을 확장자를 설정한다. '.gif', '.jpg'와 같이 여러 개를 설정하는 경우 콤마(,)를 사용한다. 설정하지 않는 경우 콤마만 입력한다.
[-ast <attach-stacktrace-on-error>]	JEUS에서 보내주는 에러 페이지에 Stack Trace를 첨부할지 여부를 설정한다.
[-att <async-timeout-min-threads>]	Servlet 3.0의 Asynchronous Servlet를 사용할 경우 타임아웃 처리를 하기 위한 Thread Pool의 최소 개수를 설정한다.  0이면 타임아웃이 제대로 동작하지 않을 수 있으므로 항상 1 이상으로 설정해야 한다.
[-rerp   -erp <default-error-page>]	-erp <default-error-page>을 이용하여 웹 애플리케이션에 별도의 에러 페이지를 설정하지 않은 경우에 사용하는 에러 페이지를 설정한다. 정적인 페이지(HTML, HTM)만 설정할 수 있으며, 절대 경로여야 한다. -rerp를 이용하여 설정을 삭제할 수 있다.
[-fhn <access-log-handler-file-handler-name>]	Permission을 수정할 file-handler의 이름이다. -fhp를 사용할 때 필수 옵션이다.
[-fhp <access-log-handler-file-handler-permission>]	Log 파일의 접근 권한 설정인 Permission을 수정한다. -fhn을 사용할 때 필수 옵션이다.

## • 예제

- [-alf <access-log-format>] 옵션을 사용한 예

```
modify-web-engine-configuration -server server1 -alf "common %I"
modify-web-engine-configuration -server server1 -alf "%h %l %u %t \"%r\" %s %b"
```

- [-aluse <use-access-log >] 옵션을 사용한 예

```
modify-web-engine-configuration -server server1 -aluse false
```

- [-alex *<access-log-excluded-extensions>*] 옵션을 사용한 예

```
modify-web-engine-configuration -server server1 -alex .gif
```

```
modify-web-engine-configuration -server server1 -alex .gif,.jpg
```

- [-alex *<access-log-excluded-extensions>*] 옵션을 사용해서 기존의 extension 정보를 없애는 예

```
modify-web-engine-configuration -server server1 -alex
```

- [-tpp *<monitoring/check-thread-pool>*], [-crp *<monitoring/check-class-reload>*], [-sp *<monitoring/check-session>*] 옵션을 사용한 예

```
modify-web-engine-configuration -tpp 10000 -crp 20000 -server server1 -sp 25000
```

- [-ast *<attach-stacktrace-on-error>*] 옵션을 사용한 예

```
modify-web-engine-configuration -server server1 -ast true
```

- [-att *<async-timeout-min-threads>*] 옵션을 사용한 예

```
modify-web-engine-configuration -server server1 -att 10
```

- [-rerp | -erp *<default-error-page>*] 옵션을 사용한 예

```
modify-web-engine-configuration -server server1 -erp /home/jeus/error/error.html
```

#### 4.2.8.24. modify-web-listener

HTTP, TCP, AJP 리스너의 설정 등을 변경한다.

- 관련 스키마

web-engine.xsd - web-engine/web-connections/http-listener, ajp13-listener, tcp-listener

- alias

modwebl

- 사용법

```
modify-web-listener -cluster <cluster-name> | -server <server-name>
                    [-f, --forceLock]
                    -name <web-connection-name>
                    [-tmin <minimum-thread-num>]
                    [-tmax <maximum-thread-num>]
                    [-tidle <max-idle-time>]
                    [-obuf <output-buffer-size>]
                    [-http2 <true | false>]
```

- 파라미터

파라미터	설명
[-f, --forceLock]	설정 변경을 강제로 진행하여 반영한다.
-name <web-connection-name>	리스너의 이름이다.
[-tmin <minimum-thread-num>]	<b>[동적 변경]</b> Thread Pool의 최소 Thread 수이다.
[-tmax <maximum-thread-num>]	<b>[동적 변경]</b> Thread Pool의 최대 Thread 수이다.
[-tidle <max-idle-time>]	Thread Pool의 Thread별 유후 대기 시간이다.
[-obuf <output-buffer-size>]	서블릿이 사용하는 응답 버퍼 크기다.
[-http2 <true   false>]	HTTP 리스너에서 HTTP/2의 사용 여부이다.

- 예제

```
modify-web-listener -name http1 -server server1 -obuf 200000 -http2 true
```

#### 4.2.8.25. modify-webtob-connector

WebtoB 커넥터의 Thread Pool 설정 등을 변경한다. 단, 운영 중인 Thread Pool의 Thread 수가 바로 변하는 것은 아니고 Thread Pool 체크 주기에 의해서 변한다.

- 관련 스키마

web-engine.xsd - web-engine/web-connections/webtob-connector

- alias

modwebtobcon

- 사용법

```
modify-webtob-connector -cluster <cluster-name> | -server <server-name>
                        [-f, --forceLock]
                        -name <web-connection-name>
```

```

[-conn <connection-count>]
[-hth <hth-count>]
[-tmin <minimum-thread-num>]
[-tmax <maximum-thread-num>]
[-tidle <max-idle-time>]
[-obuf <output-buffer-size>]
[-ver <wjp-version>]
[-addr <server-address>]
[-port <server-port> | -dsocket]
[-wbhome <webtob-home>]
[-cloud <cloud-mode>]
[-ipcport <ipc-base-port>]
[-regid <registration-id>]
[-sndbuf <send-buffer-size>]
[-rcvbuf <receive-buffer-size>]

```

#### • 파라미터

파라미터	설명
[-f, --forceLock]	설정 변경을 강제로 진행하여 반영한다.
-name <web-connection-name>	WebtoB 커넥터의 이름이다.
[-conn <connection-count>]	<b>[동적 변경]</b> WebtoB와 연결할 Connection 개수이다.
[-hth <hth-count>]	hth process의 개수이다.
[-tmin <minimum-thread-num>]	<b>[동적 변경]</b> Thread Pool의 최소 Thread 수이다.
[-tmax <maximum-thread-num>]	<b>[동적 변경]</b> Thread Pool의 최대 Thread 수이다.
[-tidle <max-idle-time>]	Thread Pool의 Thread별 유후 대기 시간이다.
[-obuf <output-buffer-size>]	서블릿 응답 버퍼 크기다.
[-ver <wjp-version>]	WebtoB와 통신할 때 사용하는 프로토콜 버전을 나타낸다. 1 또는 2를 입력한다.
[-addr <server-address>]	WebtoB 서버 주소이다. -port 사용할 때 필수 옵션이다.
[-port <server-port>]	WebtoB 서버 포트 번호이다.
[-regid <registration-id>]	WebtoB와의 등록 ID 이다.
[-dsocket]	Unix 도메인 소켓(Pipe)을 이용하여 JEUS와 WebtoB 사이에 통신을 한다.
[-wbhome <webtob-home>]	JEUS와 WebtoB가 같은 장비 안에 설치되어 있을 경우 WebtoB가 설치된 절대 경로이다. -dsocket을 사용할 때 설정할 수 있다.
[-cloud <cloud-mode>]	cloud mode를 사용할 것인지에 대한 여부를 나타낸다.
[-ipcport <ipc-base-port>]	Windows 환경에서 WebtoB의 내부 프로세스간 통신을 할 때 사용하는 기본 포트 번호이다. -dsocket을 사용할 때 설정할 수 있다.

파라미터	설명
<code>[-sndbuf &lt;send-buffer-size&gt;]</code>	TCP 소켓이나 Unix 도메인 소켓의 SO_SNDBUF를 설정한다. 0이면 OS의 기본값을 사용한다.
<code>[-rcvbuf &lt;receive-buffer-size&gt;]</code>	TCP 소켓이나 Unix 도메인 소켓의 SO_RCVBUF를 설정한다. 0이면 OS의 기본값을 사용한다.

- 예제

```
modify-webtob-connector -name webtob1 -conn 30 -tmin 30 -sndbuf 1200 -rcvbuf 2400 -server server1
```

#### 4.2.8.26. notify-auto-scale

웹 서버나 JEUS MS(Managed Server)의 추가/삭제를 알려준다.

- 관련 스키마

web-engine.xsd

- alias

nas

- 사용법

```
notify-auto-scale -wscaleout | -wscalein | -wunlink | -jscaleout | -jscalein
                  -address <ip-address>
                  [-port <server-port>]
                  [-group <group-id>]
```

- 파라미터

파라미터	설명
<code>-wscaleout</code>	웹 서버가 추가되었음을 알려준다.
<code>-wscalein</code>	웹 서버가 삭제되었음을 알려준다.
<code>-wunlink</code>	웹 서버가 종료되었음을 알려준다.
<code>-jscaleout</code>	JEUS MS(Managed Server)가 추가되었음을 알려준다.
<code>-jscalein</code>	JEUS MS(Managed Server)가 삭제되었음을 알려준다.
<code>-address &lt;ip-address&gt;</code>	추가/삭제된 서버의 ip-address를 지정한다.



파라미터	설명
[-port <server-port>]	추가/삭제된 서버의 port를 지정한다.  WebtoB를 사용할 경우 WebtoB에 설정된 JSVPORT를 넣어준다. 그 외의 경우에는 사용되지 않는다.
[-group <group-id>]	CWDP 동작을 수행할 그룹의 ID를 지정한다.

- 예제

```
notify-auto-scale -wscaleout -address 192.168.0.2 -port 9900
notify-auto-scale -wscalein -address 192.168.0.2 -port 9900
```

#### 4.2.8.27. precompile-jsp

deploy된 웹 컨텍스트의 JSP 파일들을 미리 컴파일한다.

JSP 파일은 보통 웹 브라우저로부터 JSP 파일에 대한 요청이 웹 엔진에 올 때 컴파일된다. 그리하여 JSP 파일에 대한 초기 요청은 많은 파싱과 컴파일 작업을 통해 많은 리소스를 소모하여 응답시간이 길어진다.

개발된 JSP 파일이 많고, 사용자 요구가 빈번한 웹사이트이면 "초기 요청 수행"에 소요되는 시간이나 리소스가 서비스에 크게 영향을 미칠 수 있다. 이런 경우에는 개발된 JSP 소스 파일들을 웹 엔진이 시작해서 서비스가 시작되기 전에 **precompile-jsp** 명령어로 컴파일을 수행하면 문제를 해결할 수 있다.

precompile-jsp는 JEUS가 부팅된 상태에서 deploy된 모듈에 한해 컴파일을 수행한다. JEUS가 부팅되지 않은 상태에서 사전 컴파일을 수행하기 위해서는 [appcompiler](#)를 사용해야 한다.



1. precompile-jsp는 웹 엔진의 다른 명령어와는 다르게 클러스터를 옵션으로 사용하지 않는다.
2. JEUS 6까지는 jspc라는 명령어로 JEUS\_HOME/bin 밑에 따로 존재한 배치 컴파일러를 JEUS 7부터는 jeusadmin의 내부 명령어로 통합하였다.

- 관련 스키마

web-engine.xsd

- alias

jspc

- 사용법

```
precompile-jsp [-server <server-name>]
               -ctx <context-name> | -id <application-id>
               [-e <excluded-jsp-list-file> | -l <included-jsp-list-file>]
```

- 파라미터

파라미터	설명
-ctx <context-name>	웹 컨텍스트 이름을 설정한다. EAR 안에 있는 경우 ear-name#context-name으로 입력한다.
-id <application-id>	웹 컨텍스트가 포함된 애플리케이션의 ID를 설정한다. EAR 안에 있는 경우엔 설정할 수 없으며, -ctx 옵션을 사용해야 한다.
[-e <excluded-jsp-list-file>]	<p>웹 컨텍스트를 기준으로 컴파일하지 않을 path들을 기술한 텍스트 파일을 지정한다. JSP path들은 Context Root에 대한 상대 경로를 가져야 하며 반드시 슬래시( / )로 시작해야 한다.</p> <p>jsp-list-file의 위치는 시스템 절대 경로로 지정한다.</p>
[-i <included-jsp-list-file>]	<p>웹 컨텍스트를 기준으로 컴파일할 path들을 기술한 텍스트 파일을 지정한다. JSP path들은 Context Root에 대한 상대 경로를 가져야 하며 반드시 슬래시( / )로 시작해야 한다.</p> <p>jsp-list-file의 위치는 시스템 절대 경로로 지정한다.</p>

- 예제

```
precompile-jsp -server server1 -ctx myctx
```

- 참고

본 명령어는 서버에 접속하지 않은 상태에서도 사용할 수 있다.

#### 4.2.8.28. reload-web-context

디스크로부터 지정한 Entity(웹 컨텍스트)를 reload한다. 이미 로드된 서블릿(JSP)들과 클래스들을 Reload한다. 이 명령어는 서블릿이나 클래스의 reload 기능이 비활성화되어 있을 경우 유용하게 사용할 수 있다(enable-reload, check-class-reload).

Managed Server에서 jeusadmin에 연결할 경우 해당 서버가 자동 적용되므로 cluster나 server 옵션을 생략한다.

- 관련 스키마

jeus-web-dd.xsd

- alias

webreload, reloadctx

- 사용법

```
reload-web-context -cluster <cluster-name> | -server <server-name>
```

```
-ctx <context-name>
```

- 파라미터

파라미터	설명
-ctx <context-name>	reload하려는 컨텍스트이다. 값이 반드시 설정되어야 한다.

- 예제

```
reload-web-context -server server1 -ctx servlets-examples
```

#### 4.2.8.29. remove-backup-webtob

Backup WebtoB를 삭제한다.

- 관련 스키마

web-engine.xsd - web-engine/web-connections/webtob-connector/webtob-backup

- alias

rmbackupwebtob

- 사용법

```
remove-backup-webtob -cluster <cluster-name> | -server <server-name>  
                    [-f, --forceLock]  
                    -name <web-connection-name>
```

- 파라미터

파라미터	설명
[-f, --forceLock]	설정 변경을 강제로 진행하여 반영한다.
-name <web-connection-name>	삭제할 backup을 가지고 있는 WebtoB 커넥터의 이름이다.

- 예제

```
remove-backup-webtob -server server1 -name webtob1
```

#### 4.2.8.30. remove-response-header

웹 엔진의 Response Header 설정을 삭제한다. 삭제되는 Response Header는 대소문자를 구별하지 않는다. 단,

XML에만 반영하며 실제로 서비스에 적용하려면 서버를 재시작해야 한다.

- 관련 스키마

web-engine.xsd - web-engine/response-header

- alias

rmwebrh

- 사용법

```
remove-response-header -cluster <cluster-name> | -server <server-name>
                        [-f,--forceLock]
                        [-a,--all | -n,--name <header-name>]
```

- 파라미터

파라미터	설명
[-f, --forceLock]	설정 변경을 강제로 진행하여 반영한다.
[-a,--all   -n,--name <header-name>]	<ul style="list-style-type: none"><li>• -n, --name &lt;header-name&gt;를 이용하여 특정 HTTP Response Header를 삭제한다. 입력값은 대소문자를 구별하지 않는다. 한 개 이상의 Header를 동시에 삭제할 경우 콤마(,)로 구분하여 삭제할 Header의 이름을 설정한다.</li><li>• -a, --all를 이용하면 현재 설정되어있는 HTTP Response Header를 모두 삭제할 수 있다.</li></ul>

- 예제

```
remove-response-header -server server1 -n testheader
remove-response-header -server server1 -n testheader2,testheader4
remove-response-header -server server1 -a
```

#### 4.2.8.31. remove-tmax-connector

Tmax 커넥터를 삭제한다. 단, XML에만 반영하며 실제로 서비스에 적용하려면 서버를 재시작해야 한다.

- 관련 스키마

web-engine.xsd - web-engine/web-connections/tmax-connector

- alias

rmtmaxcon

- 사용법

```
remove-tmax-connector -cluster <cluster-name> | -server <server-name>
                        [-f,--forceLock]
                        <web-connection-name>
```

- 파라미터

파라미터	설명
[-f, --forceLock]	설정 변경을 강제로 진행하여 반영한다.
<web-connection-name>	Tmax 커넥터의 이름이다.

- 예제

```
remove-tmax-connector -server server1 tmax1
```

#### 4.2.8.32. remove-virtual-host

가상 호스트를 삭제한다.

- 관련 스키마

web-engine.xsd - web-engine/virtual-host

- alias

rmvh

- 사용법

```
remove-virtual-host -cluster <cluster-name> | -server <server-name>
                    [-f,--forceLock]
                    <virtual-host-name>
```

- 파라미터

파라미터	설명
[-f, --forceLock]	설정 변경을 강제로 진행하여 반영한다.
<virtual-host-name>	가상 호스트의 이름이다.

- 예제

```
remove-virtual-host -server server1 host2
```

#### 4.2.8.33. remove-valve

Valve를 삭제한다.

- 관련 스키마

web-engine.xsd - pipeline/valve

- alias

rmvlv

- 사용법

```
remove-valve -cluster <cluster-name> | -server <server-name>
               [-f, --forceLock]
               -cn, --classname <class-name>
               [-vh <virtual-host-name>]
```

- 파라미터

파라미터	설명
[-f, --forceLock]	설정 변경을 강제로 진행하여 반영한다.
-cn, --classname <class-name>	서버, 가상호스트에서 삭제할 valve class를 지정한다.
[-vh <virtual-host-name>]	valve를 삭제할 가상 호스트의 이름이다. 가상 호스트 이름을 설정 안하면 서버 단의 valve가 삭제된다.

- 예제

```
remove-valve -server server1 -cn UserValveBase -vh host1
```

#### 4.2.8.34. remove-valve-property

Valve의 프로퍼티 설정을 삭제한다. 단, XML에만 반영하며 실제로 서비스에 적용하려면 서버를 재시작해야 한다.

- 관련 스키마

web-engine.xsd - pipeline/valve/property

- alias

rmvlvprop

- 사용법

```
remove-valve-property -cluster <cluster-name> | -server <server-name>
```

```
[-f, --forceLock]
-cn, --classname <class-name>
-k, --key <key>
[-vh <virtual-host-name>]
```

- 파라미터

파라미터	설명
[-f, --forceLock]	설정 변경을 강제로 진행하여 반영한다.
-cn, --classname <class-name>	삭제할 property를 가지고 있는 valve class를 지정한다. 제우스의 ValveBase를 상속받은 클래스로 지정해야 한다.
-k, --key <key>	삭제할 property의 key를 지정한다.
[-vh <virtual-host-name>]	삭제할 property의 valve를 가지고 있는 가상 호스트의 이름이다. 가상 호스트 이름을 설정 안하면 서버 단에 있는 valve의 property가 삭제된다.

- 예제

```
remove-valve-property -server server1 -cn UserValveBase -k userProperty -vh host1
```

#### 4.2.8.35. remove-web-cookie-policy

웹 엔진의 HTTP Cookie Policy 설정을 삭제한다. 단, XML에만 반영하며 실제로 서비스에 적용하려면 서버를 재시작해야 한다.

- 관련 스키마

web-engine.xsd - web-engine/cookie-policy

- alias

rmcp, rmcookie

- 사용법

```
remove-web-cookie-policy -cluster <cluster-name> | -server <server-name>
[-f, --forceLock]
[-auer, --apply-url-encoding-rule]
[-enc, --charset-encoding]
[-vh <virtual-host-name>]
```

- 파라미터

파라미터	설명
[-f, --forceLock]	설정 변경을 강제로 진행하여 반영한다.
[-auer, --apply-url-encoding-rule]	URL Encoding Rule 적용 여부를 삭제한다.
[-enc, --charset-encoding]	URL Encoding Rule을 적용할 때 사용하는 Charset Encoding을 삭제한다.
[-vh <virtual-host-name>]	가상 호스트를 지정한다.

- 예제

```
remove-web-cookie-policy -server server1 -auer
```

#### 4.2.8.36. resume-web-component

일시 중지된 웹 컴포넌트(서블릿)를 재시작한다.

Managed Server에서 jeusadmin에 연결할 경우 해당 서버가 자동 적용되므로 cluster나 server 옵션을 생략한다.



JEUS 6에서는 명령어 webresume과 webtob -r 명령어로 분리되어있던 것을 JEUS 7부터 통합하였다. 또한 JEUS 7부터 웹 컨텍스트에 대한 시작 명령어는 별도로 제공되지 않는다. JEUS Master Server의 [start-application](#)를 사용해서 제어해야 한다.

- 관련 스키마

web-engine.xsd - web-engine/web-connections

- alias

webresume

- 사용법

```
resume-web-component -cluster <cluster-name> | -server <server-name>
                        -ctx,--context <context>
                        -svl,--servlet <servlet>
```

- 파라미터



파라미터	설명
-ctx, --context <context>	<p>시작할 서블릿이 포함된 웹 컨텍스트를 설정해야 한다.</p> <p>반드시 -svl 옵션과 함께 설정해야 한다.</p> <p>application-information -type 옵션을 사용한 명령어를 사용한 조회 결과 중 Module Name에 해당하는 값을 &lt;context&gt;에 설정한다.</p>
-svl,--servlet <servlet>	중지된 서블릿 중에서 다시 시작할 서블릿이다. 컨텍스트 옵션이 설정되어 있어야 한다.

- 예제
  - -svl,--servlet <servlet> 옵션을 사용한 예

```
resume-web-component -server server1 -ctx servlets -svl CookieExample
```

#### 4.2.8.37. remove-web-encoding

웹 엔진의 Encoding을 삭제한다. 단, XML에만 반영하며 실제로 서비스에 적용하려면 서버를 재시작해야 한다.

- 관련 스키마

web-engine.xsd - web-engine/encoding

- alias

rmenc

- 사용법

```
remove-web-encoding -cluster <cluster-name> | -server <server-name>
                        [-f,--forceLock]
                        [-url]
                        [-req]
                        [-res]
                        [-vh <virtual-host-name>]
```

- 파라미터

파라미터	설명
[-f, --forceLock]	설정 변경을 강제로 진행하여 반영한다.
[-url]	Request URL의 Character Set Encoding을 삭제한다.
[-req]	Request의 Character Set Encoding을 삭제한다.
[-res]	Response의 Character Set Encoding을 삭제한다.
[-vh <virtual-host-name>]	가상 호스트를 지정한다.

- 예제

```
remove-web-encoding -server server1 -url
remove-web-encoding -server server1 -req -res
```

#### 4.2.8.38. remove-web-listener

HTTP, TCP, AJP 리스너를 삭제한다. 단, XML에만 반영하며 실제로 서비스에 적용하려면 서버를 재시작해야 한다. ADMIN-HTTP 리스너는 삭제되지 않는다.

- 관련 스키마

web-engine.xsd - web-engine/web-connections/http-listener, ajp13-listener, tcp-listener

- alias

rmweb

- 사용법

```
remove-web-listener -cluster <cluster-name> | -server <server-name>
                        [-f,--forceLock]
                        <web-connection-name>
```

- 파라미터

파라미터	설명
[-f, --forceLock]	설정 변경을 강제로 진행하여 반영한다.
<web-connection-name>	리스너의 이름이다.

- 예제

```
remove-web-listener -server server1 http1
```

#### 4.2.8.39. remove-web-properties

웹 엔진의 프로퍼티 설정을 삭제한다. 삭제되는 프로퍼티는 대소문자를 구별한다. 단, XML에만 반영하며 실제로 서비스에 적용하려면 서버를 재시작해야 한다.

- 관련 스키마

web-engine.xsd - web-engine/properties

- alias

rmwebpr

- 사용법

```
remove-web-properties -cluster <cluster-name> | -server <server-name>
                        [-f,--forceLock]
                        -a,--all | -k,--keys <keys>
                        [-vh <virtual-host-name>]
```

- 파라미터

파라미터	설명
[-f, --forceLock]	설정 변경을 강제로 진행하여 반영한다.
-a,--all   -k,--keys <keys>	<ul style="list-style-type: none"><li>• -k, --keys &lt;keys&gt;를 이용하여 웹 엔진의 특정 프로퍼티를 삭제한다. 입력값은 대소문자를 구별한다. 한 개 이상의 프로퍼티를 동시에 삭제하고자할 경우 콤마(,)로 구분하여 삭제할 프로퍼티의 key를 설정한다.</li><li>• -a, --all를 이용하면 현재 설정되어있는 프로퍼티를 모두 삭제할 수 있다.</li></ul>
[-vh <virtual-host-name>]	가상 호스트를 지정한다.

- 예제

```
remove-web-properties -server server1 -k jeus.servlet.jsp.compile-java-source-concurrently1
remove-web-properties -server server1 -a
```

#### 4.2.8.40. remove-webtob-connector

WebtoB 커넥터를 삭제한다. 단, XML에만 반영하며 실제로 서비스에 적용하려면 서버를 재시작해야 한다.

- 관련 스키마

web-engine.xsd - web-engine/web-connections/webtob-connector

- alias

rmwebtobcon

- 사용법

```
remove-webtob-connector -cluster <cluster-name> | -server <server-name>
                        [-f,--forceLock]
                        <web-connection-name>
```

- 파라미터

파라미터	설명
[-f, --forceLock]	설정 변경을 강제로 진행하여 반영한다.
<web-connection-name>	WebtoB 커넥터의 이름이다.

- 예제

```
remove-webtob-connector -server server1 webtob1
```

#### 4.2.8.41. set-valve-property

Valve의 프로퍼티 설정을 추가한다. 추가되는 프로퍼티는 대소문자를 구별한다. 단, XML에만 반영하며 실제로 서비스에 적용하려면 서버를 재시작해야 한다.

- 관련 스키마

web-engine.xsd - pipeline/valve/property

- alias

addvprop

- 사용법

```
set-valve-property -cluster <cluster-name> | -server <server-name>
                    [-f, --forceLock]
                    -cn, --classname <class-name>
                    -k, --key <key>
                    -v, --value <value>
                    [-vh <virtual-host-name>]
```

- 파라미터

파라미터	설명
[-f, --forceLock]	설정 변경을 강제로 진행하여 반영한다.
-cn, --classname <class-name>	property를 추가할 valve class를 지정한다. 제우스의 ValveBase를 상속받은 클래스로 지정해야 한다.
-k, --key <key>	valve에 추가할 key를 지정한다.
-v, --value <value>	valve에 추가할 value를 지정한다.
[-vh <virtual-host-name>]	property를 추가할 가상 호스트의 이름이다. 가상 호스트 이름을 설정 안하면 서버 단에 있는 valve에 추가된다.

- 예제

```
set-valve-property -server server1 -cn UserValveBase -k userProperty -v true -vh host1
```

#### 4.2.8.42. show-request-processing-flow

URL 패턴이나 특정 호스트 이름으로 들어온 요청의 처리 흐름을 조회한다.

Managed Server에서 jeusadmin에 연결할 경우 해당 서버가 자동 적용되므로 cluster나 server 옵션을 생략한다.

- 관련 스키마

web-app\_4\_0.xsd

- alias

reqflow

- 사용법

```
show-request-processing-flow -cluster <cluster-name> | -server <server-name>
                             -path <url-pattern>
                             [-hostname <host-name>]
```

- 파라미터

파라미터	설명
-path <url-pattern>	슬래시( / )로 시작하는 URL 패턴이다.
[-hostname <host-name>]	HTTP Host Header에 명시된 호스트 이름이다.

- 예제

```
show-request-processing-flow -server server1 -path /test/index.jsp

show-request-processing-flow -server server1 -path /test -hostname examples.com
```

#### 4.2.8.43. show-web-engine-configuration

웹 엔진에 대한 설정 정보를 조회한다. 이 명령어를 실행하여 조회되는 구성 항목과 필드들에 대한 정보는 XML의 웹 엔진 구성 레퍼런스를 참고한다.

출력 항목 중 빈 칸으로 표시되는 항목은 XML에 실제 설정이 되어 있지 않음을 의미한다. 즉, 값이 표시되어 있는 경우가 실제 설정되어 있거나, 기본값이 있는 설정이다.

- 관련 스키마

web-engine.xsd - web-gine

- alias

webcfg, webconf, showwebcfg, showwebconf

- 사용법

```
show-web-engine-configuration -cluster <cluster-name> | -server <server-name>
                                [-vh,--virtual host]
                                [-cn,--web-connections <connection-type>]
                                [-al,--access-log]
                                [-sc,--session-config]
                                [-rh,--response-header]
                                [-mo,--monitoring]
                                [-enc,--encoding]
                                [-cp,--cookie-policy]
                                [-jsp,--jsp-engine]
                                [-pr,--properties]
                                [-erp,--error-page]
                                [-att,--async-timeout_thread]
                                [-ast,--attach-stacktrace]
```

- 파라미터

파라미터	설명
[-vh,--virtual host]	가상 호스트의 설정 정보를 출력한다.
[-cn,--web-connections <connection-type>]	모든 웹 커넥션의 설정 정보를 출력한다. 인자를 줄 경우 해당 타입의 커넥션 설정 정보만 출력한다.
[-al,--access-log]	액세스 로그의 설정 정보를 출력한다.
[-sc,--session-config]	웹 엔진의 세션 설정 정보를 출력한다.
[-rh,--response-header]	응답 헤더의 설정 정보를 출력한다.
[-mo,--monitoring]	웹 엔진 모니터링의 설정 정보를 출력한다.
[-enc,--encoding]	웹 엔진의 인코딩 설정 정보를 출력한다.
[-cp,--cookie-policy]	쿠키의 정책 설정 정보를 출력한다.
[-jsp,--jsp-engine]	JSP 엔진의 설정 정보를 출력한다.
[-pr,--properties]	웹 엔진 프로퍼티의 설정 정보를 출력한다.
[-erp,--error-page]	웹 엔진의 기본 오류 페이지 설정 정보를 출력한다.
[-att,--async-timeout_thread]	비동기 서블릿의 타임아웃을 처리하는 Thread 설정 정보를 출력한다.
[-ast,--attach-stacktrace]	오류가 발생할 때 Stack Trace의 추가 여부 설정 정보를 출력한다.

- 예제

- 웹 엔진 설정 모두 조회

```
show-web-engine-configuration -server server1
```

- 옵션별 웹 엔진 설정 조회

```
show-web-engine-configuration -erp -att -server server1
```

```
show-web-engine-configuration -server server1 -sc
```

#### 4.2.8.44. show-web-statistics

구동 중인 웹 엔진의 JVM에서 사용하는 메모리 정보, 웹 엔진에 속해 있는 각 리스너/커넥터에 대한 Thread Pool의 상태, 컨텍스트에 대한 요청 횟수 및 처리시간 등의 정보를 파라미터 설정에 따라 조회한다.

- 관련 스키마

web-engine.xsd

- alias

webstats, stat, st

- 사용법

```
show-web-statistics -server <server-name>
                    [-ctx,--context <context-name>]
                    [-t,--thread | -s,--session | -r,--request | -m,--memory]
```

- 파라미터

파라미터	설명
[-t,--thread]	웹 엔진에 속해 있는 각각의 WebContainer와 Context 레벨의 Thread Pool의 상태를 조회한다. 조회되는 정보는 현재 할당된 Worker Thread의 수와 Wait-Queue에 적체된 클라이언트의 수 그리고 Thread Pool을 유지하는 최대 Thread의 개수를 포함한다.
[-s,--session]	웹 엔진에 구성된 세션 서버의 커넥션에 대한 정보를 출력한다.  정보는 현재 연결되어 있는 세션 서버의 이름, 전체 세션의 수, Pool에서 사용된 세션 서버와 커넥션의 수를 포함한다.
[-r,--request]	각 컨텍스트에 대해 처리된 요청의 누적 수치와 평균 처리시간을 출력한다.
[-m,--memory]	구동 중인 웹 엔진에서 현재 JVM에서 사용하고 있는 메모리의 상태를 조회한다.

- 예제

- 스레드 풀 정보 조회

```
show-web-statistics -server server1 -ctx test -t
```

- 세션 서버 정보 확인

```
show-web-statistics -server server1 -s
```

#### 4.2.8.45. show-webtob-connector

WebtoB Connection 정보를 출력한다. 현재 Connection의 상태를 알 수 있다.

- 관련 스키마

web-engine.xsd - web-engine/web-connections/webtob-connector

- alias

webtobcon

- 사용법

```
show-webtob-connector -cluster <cluster-name> | -server <server-name>  
                        [-f, --forceLock]  
                        -name <web-connection-name>
```

- 파라미터

파라미터	설명
[-f, --forceLock]	설정 변경을 강제로 진행하여 반영한다.
-name <web-connection-name>	WebtoB 커넥터의 이름이다.

- 예제

```
show-webtob-connector -server server1 -name webtob1
```

#### 4.2.8.46. suspend-web-component

지정한 서블릿을 중지하여 클라이언트로 에러 페이지를 보여주도록 한다.

Managed Server에서 jeusadmin에 연결할 경우 해당 서버가 자동 적용되므로 cluster나 server 옵션을 생략한다.



JEUS 6에서는 websuspend과 webtob -s 명령어로 분리되어있던 것을 JEUS 7부터



통합하였다. 또한 JEUS 7부터 웹 컨텍스트에 대한 중지 명령어는 별도로 제공되지 않는다. JEUS Master Server의 [stop-application](#)을 사용해서 제어해야 한다.

- 관련 스키마

web-engine.xsd - web-engine/web-connections

- alias

websuspend

- 사용법

```
suspend-web-component -cluster <cluster-name> | -server <server-name>
                        -ctx,--context <context>
                        -svl,--servlet <servlet>
```

- 파라미터

파라미터	설명
-ctx, --context <context>	중지할 서블릿이 포함된 웹 컨텍스트를 설정해야 한다. 반드시 -svl 옵션과 함께 설정해야 한다.  application-information -type 옵션을 이용한 명령어를 사용한 조회 결과 중 Unique Module Name에 해당하는 값을 <context>에 설정한다.
-svl, --servlet <servlet>	시작된 서블릿 중에서 중지할 서블릿이다. 컨텍스트 옵션이 설정되어 있어야 한다.

- 예제

- -svl,--servlet <servlet> 옵션을 사용한 예

```
suspend-web-component -svl CookieExample -server server1 -ctx servlets
```

## 4.2.9. 세션 관련 명령어

세션 명령어를 2가지로 구분하여 설명한다.

- 모니터링 및 제어

세션 상태를 확인하거나 제어하기 위한 명령어들이다.

명령어	설명
<a href="#">list-session</a>	idle 시간 기준으로 정렬된 세션들의 정보를 보여준다.

명령어	설명
<a href="#">remove-session</a>	세션 타임아웃과 무관하게 입력한 타임아웃 동안 access가 없는 세션들을 제거할 때 사용한다.
<a href="#">show-session-server-backup-table</a>	세션 서버의 현재 backup table 정보를 보여준다.

- 설정 변경 관련 명령어

세션 설정을 추가, 삭제, 수정할 수 있는 명령어들이다. 설정 변경 사항들은 XML에는 적용하지만 실제 운영 중인 서비스(메모리 영역)에 적용하지 않는다. XML만 변경되는 것이며 실제로 서비스에 적용하려면 서버를 재시작해야 한다.

명령어	설명
<a href="#">show-session-configuration</a>	세션 관련 설정 정보를 확인한다.
<a href="#">modify-session-configuration</a>	세션 관련 설정 정보를 확인하고 수정한다.

- 세션 서버 관련 명령어

세션 스토리지 설정을 할 수 있는 명령어들이다. 설정 후 서버를 재시작해야 설정내용들이 적용된다.

명령어	설명
<a href="#">list-sessionstorages</a>	현재 존재하는 세션 스토리지의 목록을 보여준다.
<a href="#">add-sessionstorage</a>	세션 서버에 세션 스토리지를 추가한다. 여러 개의 세션 스토리지가 있는 경우 이름은 중복 되서는 안된다.
<a href="#">remove-sessionstorage</a>	세션 서버에서 세션 스토리지를 제거한다.
<a href="#">rename-sessionstorage</a>	세션 스토리지의 이름을 변경한다.
<a href="#">set-sessionstorage-property</a>	세션 스토리지의 프로퍼티를 추가/변경한다.
<a href="#">remove-sessionstorage-property</a>	세션 스토리지의 프로퍼티를 제거한다.
<a href="#">add-sessionstorage-scope</a>	세션 스토리지에 스코프를 추가한다. 여러 개의 스코프가 있는 경우 다른 스코프와 중복되면 안된다. DOMAIN_WIDE 클러스터 모드를 사용하는 경우 DOMAIN_WIDE로 설정하면 해당 스코프를 사용할 수 있다.
<a href="#">set-sessionstorage-scope</a>	스코프에 포함될 application이나 cluster를 변경한다.
<a href="#">remove-sessionstorage-scope</a>	세션 스토리지에서 스코프를 제거한다.
<a href="#">set-sessionstorage-scope-session-config</a>	스코프에 포함된 컨텍스트에서 사용될 세션 관련 설정을 한다.
<a href="#">list-sessionserver-property</a>	세션 서버의 프로퍼티 목록을 보여준다.
<a href="#">set-sessionserver-property</a>	세션 서버의 프로퍼티를 추가/변경한다.
<a href="#">remove-sessionserver-property</a>	세션 서버의 프로퍼티를 제거한다.

- 중앙식 세션 서버 관련 명령어

중앙식 세션 클러스터링의 세션 서버 설정을 위한 명령어들이다. 특정 서버를 중앙식 세션 서버로 동작 하게 하거나 제외 시키는 명령어 들이다. 추가적으로 중앙식 세션 서버의 그룹에 대한 설정 하는 명령어가 존재한다.

설정 후 관련 서버를 재시작해야 적용된다.

명령어	설명
<code>add-central-session-server</code>	특정 서버를 중앙식 세션 서버로 동작 하도록 설정한다.
<code>modify-central-session-server</code>	특정 서버의 중앙식 세션 서버 그룹을 변경한다.
<code>remove-central-session-server</code>	특정 서버를 중앙식 세션 서버로 동작하지 않도록 설정한다.
<code>add-central-session-def</code>	중앙식 세션 서버 정의 안에 그룹의 서버를 새로 추가하거나 변경한다.
<code>remove-central-session-server-definition</code>	중앙식 세션 서버 정의 안의 그룹을 제거 하거나, 그룹내 서버를 제거한다.
<code>list-central-session-server-session</code>	세션 서버에 존재하는 세션의 정보를 보여준다.
<code>show-central-session-server-statistics</code>	구동 중인 중앙식 세션 서버의 메모리 및 세션 정보를 조회한다.

#### 4.2.9.1. 공통 적용 사항

다음은 모니터링 및 제어 명령어에서 공통적으로 적용되는 옵션에 대한 설명이다. 아래 옵션 중 하나만 설정될 수 있다.

- `[-server <server-name> ]` 옵션
  - 각 명령어가 적용될 서버를 지정할 때 사용하는 옵션이다. 옵션을 설정하지 않으면 현재 접속된 서버의 웹 엔진으로 적용된다.
  - 각 명령어의 옵션 설명에서 이 옵션에 대한 설명은 생략한다.



1. 서버에 즉시 반영이 안되는 설정의 경우 즉, Pending된 설정이 최초의 설정으로 재설정되는 명령어를 수행하면, 서버의 재시작이 필요하다는 안내문(Restart the Server)이 출력되지 않는다.
2. JEUS Master Server의 경우는 위 옵션 중 하나는 반드시 필요하다. 그렇지만 JEUS Master Server가 아닌 일반 관리 서버에서 해당 명령어를 실행할 경우에는 아래 옵션들이 없을 경우 접속한 서버를 기반으로 명령이 수행된다.

#### 4.2.9.2. list-session

idle 시간 기준으로 정렬된 세션들의 정보를 보여준다.

- 관련 스키마

`jeus-session-server.xsd`, `jeus-session-domain.xsd`

- alias

lisession, lss

- 사용법

```
list-session -server <server-name>
             [-s,--simple ]
             [-count <count>]
             [-target <manager-name>]
             [-id <target-id>]
```

- 파라미터

파라미터	설명
-server	명령어가 적용될 서버를 지정한다. 해당 옵션은 필수 옵션이다.
[-s,--simple ]	출력되는 결과를 Simple format으로 출력한다.
[-count <count>]	모니터링으로 확인될 세션들의 개수를 설정한다. 생략하면 100개가 기본으로 적용된다.
[-target <manager-name>]	세션 정보를 보고 싶은 타깃 세션 매니저를 설정한다. 옵션을 주지 않을 경우 서버 내 모든 매니저의 정보를 보여준다.
[-id <target-id>]	특정 세션을 검색하기 위해 세션의 ID를 설정한다.

- 예제

- 별도 옵션을 사용하지 않는 예

```
list-session -server server1
```

- [-s,--simple ] 옵션을 사용한 예

```
list-session -server server1 --simple
```

### 4.2.9.3. remove-session

Idle Time 기준으로 설정한 시간이 지난 세션을 제거한다.

- 관련 스키마

jeus-session-server.xsd, jeus-session-domain.xsd

- alias

rmsession, rms

- 사용법

```
remove-session -server <server-name>
               -time,--timeout <timeout(min)>
```

- 파라미터

파라미터	설명
-server	명령어가 적용될 서버를 지정한다. 해당 옵션은 필수 옵션이다.
-time,--timeout <timeout(min)>	세션을 제거할 기준이 되는 타임아웃을 설정한다. 해당 옵션은 필수 옵션이다.

- 예제

- [-time,--timeout <timeout(min)>]을 사용한 예

```
remove-session -server server1 --timeout 3
```

#### 4.2.9.4. show-session-server-backup-table

세션 서버의 현재 backup table 정보를 보여준다.

- 관련 스키마

jeus-session-server.xsd, jeus-session-domain.xsd

- alias

sssb, sbt

- 사용법

```
show-session-server-backup-table -server <server-name>
```

- 파라미터

파라미터	설명
-server	명령어가 적용될 서버를 지정한다. 해당 옵션은 필수 옵션이다.

- 예제

```
show-session-server-backup-table -server server1
```

#### 4.2.9.5. show-session-configuration

세션 관련 설정 정보를 조회한다.

- 관련 스키마

jeus-session-server.xsd, jeus-session-domain.xsd

- alias

getsc, ssc

- 사용법

```
show-session-configuration -server <server-name>
[-h, --help]
```

- 파라미터

파라미터	설명
-server <server-name>	웹 엔진에 설정된 세션 설정을 확인한다.
[-h, --help]	파라미터 없는 명령어와 동일하며 전체적인 설정 구조 및 파라미터들을 설명한다.

- 예제

- [-server <server-name>] 옵션을 사용한 예

```
show-session-configuration -server server1
```

#### 4.2.9.6. modify-session-configuration

세션 관련 설정 정보를 수정한다.

- 관련 스키마

jeus-session-server.xsd, jeus-session-domain.xsd

- alias

setsc, msc

- 사용법

```
modify-session-configuration -server <server-name>
[-f, --forceLock]
[-to, --timeout <int timeout>]
```

```

[-mc,--max-session-count <int limit count>]
[-rp,--reload-persistent <boolean reload-persistent>]
[-tmc,--tracking-mode.cookie <boolean enable-cookie>]
[-tmu,--tracking-mode.url <boolean enable-url>]
[-tms,--tracking-mode.ssl <boolean enable-ssl>]
[-scn,--session-cookie.cookie-name <string cookie-name>]
[-sucn,--session-cookie.url-cookie-name <string url-cookie-name>]
[-scv,--session-cookie.version <int version(1 or 0)>]
[-scd,--session-cookie.domain <string domain-scope>]
[-scp,--session-cookie.path <string path-scope>]
[-scm,--session-cookie.max-age <int max-age(sec)>]
[-scs,--session-cookie.secure <boolean secure>]
[-sch,--session-cookie.http-only <boolean http-only>]
[-sca,--session-cookie.same-site <string same-site>]
[-scc,--session-cookie.comment <string comment>]

```

#### • 파라미터

파라미터	설명
-server <server-name>	웹 엔진에 설정된 세션 설정을 변경한다. 해당 옵션은 필수 옵션이다.
[-f, --forceLock]	설정 변경을 강제로 진행하여 반영한다.
[-to,--timeout <int timeout>]	session-config의 세션 타임아웃을 설정한다.
[-mc,--max-session-count <int maxCount>]	저장할 세션의 최대 개수를 설정한다.
[-rp,--reload-persistent <boolean reload-persistent>]	session-config의 Reload Persistent를 설정한다.
[-tmc,--tracking-mode.cookie <boolean enable-cookie>]	session-config의 트래킹 모드 중 쿠키를 설정한다.
[-tmu,--tracking-mode.url <boolean enable-url>]	session-config의 트래킹 모드 중 URL rewriting을 설정한다.
[-tms,--tracking-mode.ssl <boolean enable-ssl>]	session-config의 트래킹 모드 중 SSL을 설정한다.
[-scn,--session-cookie.cookie-name <string cookie-name>]	세션 쿠키의 이름을 설정한다.
[-sucn,--session-cookie.url-cookie-name <string url-cookie-name>]	url rewriting을 사용하여 세션을 전달할 때 해당 세션 쿠키의 이름을 설정한다.
[-scv,--session-cookie.version <int version(1 or 0)>]	세션 쿠키의 컴파일 버전을 선택한다.
[-scd,--session-cookie.domain <string domain-scope>]	세션 쿠키의 도메인 범위를 결정한다.

파라미터	설명
<code>[-scp,--session-cookie.path &lt;string path-scope&gt;]</code>	세션 쿠키의 path 범위를 결정한다.
<code>[-scm,--session-cookie.max-age &lt;int max-age&gt;]</code>	브라우저에서 세션 쿠키의 유지 시간을 결정한다.
<code>[-scs,--session-cookie.secure &lt;boolean secure&gt;]</code>	브라우저가 쿠키를 SSL에 제한되게 전송할지를 결정한다.
<code>[-sch,--session-cookie.http-only &lt;boolean http-only&gt;]</code>	브라우저가 HTTP 요청에 의해서만 세션 쿠키를 사용할지를 결정한다.
<code>[-sca,--session-cookie.same-site &lt;string same-site&gt;]</code>	세션 ID 쿠키가 사용자 의도와 상관없는 요청에 쓰이는 공격(사이트 간 요청위조)을 방지하는 보안기술을 설정한다.
<code>[-scc,--session-cookie.comment &lt;string comment&gt;]</code>	쿠키 버전이 1일 때에 해당 세션 쿠키의 설명을 설정한다.

- 예제
  - `[-server <server-name>]` 옵션을 사용한 예

```
modify-session-configuration -server server1 -to 40 -scs true
```

#### 4.2.9.7. list-sessionstorages

현재 세션서버에 존재하는 세션 스토리지의 목록을 보여준다.

- 관련 스키마  
jeus-session-server.xsd, jeus-session-domain.xsd
- alias  
listsessionstorages, listss
- 사용법

```
list-sessionstorages
```

#### 4.2.9.8. add-sessionstorage

세션 서버에 세션 스토리지를 추가한다. 다른 스토리지와 이름이 중복되서는 안된다.

- 관련 스키마  
jeus-session-server.xsd, jeus-session-domain.xsd



- alias

addsession-storage, addss, add-session-storage

- 사용법

```
add-sessionstorage <session-storage-name>
    [-provider <provider>]
    [-property <property>]
    [-csm,--centralSessionManager <target-group-name>]
```

- 파라미터

파라미터	설명
<session-storage-name>	추가할 세션 스토리지의 이름이다.
[-provider <provider>]	<p>사용할 Session Manager의 Provider를 설정한다. 예약어를 설정하거나 사용한 패키지를 설정한다.</p> <p>예약어 외 설정이 되지 않는다.</p> <p>예약어는 다음과 같다.</p> <ul style="list-style-type: none"> <li>• DISTRIBUTE : JEUS에서 제공하는 분산식 세션 매니저(서버)를 사용한다.</li> <li>• CENTRAL : JEUS에서 제공하는 중앙식 세션 매니저를 사용한다.</li> <li>• REDIS : JEUS에서 제공하는 REDIS 세션 매니저를 사용한다.</li> <li>• HAZELCAST : JEUS에서 제공하는 HAZELCAST 세션 매니저를 사용한다.</li> <li>• RUNTIME : 다른 프로바이더가 존재할 경우 해당 프로바이더를 우선 사용한다. 없을 경우 JEUS 세션 매니저를 사용한다.</li> </ul>
[-property <property >]	세션 스토리지의 property이다.
[-csm <target-group-name>]	중앙식 세션 서버 그룹의 이름이다.

- 예제

```
add-sessionstorage sessionStorage1 -provider CENTRAL -property connect-timeout=10000,full-search-timeout=20000
```

#### 4.2.9.9. remove-sessionstorage

세션 서버에서 세션 스토리지를 제거한다.

- 관련 스키마

jeus-session-server.xsd, jeus-session-domain.xsd

- alias

removess, rmss, removesessionstor, rmssessionstor, rm,sessionstorage

- 사용법

```
remove-sessionstorage <session-storage-name>
```

- 파라미터

파라미터	설명
<session-storage-name>	제거할 세션 스토리지의 이름이다.
-csm <target-group-name>	제거할 중앙식 세션 서버 그룹의 이름이다.

- 예제

```
remove-sessionstorage sessionStorage1
```

#### 4.2.9.10. rename-sessionstorage

세션 스토리지의 이름을 변경한다.

- 관련 스키마

jeus-session-server.xsd, jeus-session-domain.xsd

- alias

renamesessionstor, renamesessionstor

- 사용법

```
rename-sessionstorage <session-storage-name>  
                        -name <new-session-storage-name>
```

- 파라미터

파라미터	설명
<session-storage-name>	이름을 변경할 세션 스토리지의 이름이다.
-name <new-session-storage-name>	새로운 세션 스토리지의 이름이다. 해당 옵션은 필수 옵션이다.

- 예제

```
rename-sessionstorage sessionStorage1 -name new_sessionStorage
```

#### 4.2.9.11. set-sessionstorage-property

세션 스토리지의 property를 추가하거나 변경할 수 있다.

- 관련 스키마

jeus-session-server.xsd, jeus-session-domain.xsd

- alias

setsessionstorageproperty, setsessionstorprop, setsstorprop

- 사용법

```
set-sessionstorage-property <session-storage-name>
                        [-key <key-of-property> -value <value-of-property>] | [-property <
property>]
```

- 파라미터

파라미터	설명
<session-storage-name>	property를 추가할 세션 스토리지의 이름이다.
[-key <key-of-property>]	추가하거나 변경할 property의 key이다. 이미 key가 설정되어 있다면 property의 value를 변경하고, 없다면 새로 추가한다.  [-property] 옵션과 동시에 쓸 수 없다.
[-value <value-of-property>]	추가하거나 변경할 property의 value이다.  [-key] 옵션이 있다면 필수로 입력해야 한다.
[-property <property>]	추가하거나 변경할 property이다. 이미 key가 설정되어 있다면 property의 value를 변경하고, 없다면 새로 추가한다.  [-key] 옵션과 동시에 쓸 수 없다.



[-key] 또는 [-property] 중 하나는 필수로 입력해야 한다.

- 예제

- [-key <key-of-property> -value <value-of-property>] 옵션을 사용한 예

```
set-sessionstorage-property sessionStorage1 -key full-search-timeout -value 10000
```

- [-property <property>] 옵션을 사용한 예

```
set-sessionstorage-property sessionStorage1 -property full-search-timeout=10000,backup-queue-size=200
```

#### 4.2.9.12. remove-sessionstorage-property

세션 스토리지의 property를 제거할 수 있다.

- 관련 스키마

jeus-session-server.xsd, jeus-session-domain.xsd

- alias

rmsessionstorageproperty, rmsessionstorprop, rmsstorprop

- 사용법

```
remove-sessionstorage-property <session-storage-name>  
-key <key-of-property>
```

- 파라미터

파라미터	설명
<session-storage-name>	property를 제거할 세션 스토리지의 이름이다.
-key <key-of-property>	세션 스토리지에서 제거할 property의 key이다. 해당 옵션은 필수 옵션이다.

- 예제

```
remove-sessionstorage-property sessionStorage1 -key full-search-timeout
```

#### 4.2.9.13. add-sessionstorage-scope

세션 스토리지에 스코프를 추가한다. 다른 스코프와 이름이 중복되서는 안된다.

- 관련 스키마

jeus-session-server.xsd, jeus-session-domain.xsd

- alias

addsessionstoragescope, addssscope, addsessionstorscope, addsss

- 사용법

```
add-sessionstorage-scope <scope-name>
    -target <storage-name>
    [-clusters <target-cluster-list> | -applications <target-application-list>]
```

- 파라미터

파라미터	설명
<scope-name>	추가할 세션 스코프의 이름이다.
-target <storage-name>	스코프를 추가할 세션 스토리지의 이름이다. 해당 옵션은 필수 옵션이다.
[-clusters <target-cluster-list>]	스코프에 포함될 클러스터 리스트이다. [-applications] 옵션 과 동시에 쓸 수 없다.
[-applications <target-application-list>]	스코프에 포함될 애플리케이션 리스트이다. [-clusters] 옵션과 동시에 쓸 수 없다.

- 예제

- [-clusters <target-cluster-list> ] 옵션을 사용한 예

```
add-sessionstorage-scope scope1 -target sessionStorage -clusters cluster1,cluster2
```

- [-applications <target-application-list>] 옵션을 사용한 예

```
add-sessionstorage-scope scope1 -target sessionStorage -applications app1,app2
```

#### 4.2.9.14. set-sessionstorage-scope

스코프에 포함될 클러스터나 애플리케이션을 설정한다. 기존에 설정되어 있던 target-application이나 target-cluster가 제거되고 새롭게 설정된다.

- 관련 스키마

jeus-session-server.xsd, jeus-session-domain.xsd

- alias

setsessionstoragescope, setsessionstorscope, setsss, setssscope

- 사용법

```
set-sessionstorage-scope <scope-name>
                        [-clusters <target-cluster-list> | -applications <target-application-list>]
```

- 파라미터

파라미터	설명
<scope-name>	추가할 세션 스코프의 이름이다.
[-clusters <target-cluster-list>]	스코프에 포함될 클러스터의 리스트이다. [-applications] 옵션과 동시에 쓸 수 없다.
[-applications <target-application-list>]	스코프에 포함될 애플리케이션 리스트이다. [-clusters] 옵션 과 동시에 쓸 수 없다.

- 예제

- [-clusters <target-cluster-list> ] 옵션을 사용한 예

```
set-sessionstorage-scope scope1 -clusters cluster1,cluster2
```

- [-applications <target-application-list>] 옵션을 사용한 예

```
set-sessionstorage-scope scope1 -applications app1,app2
```

#### 4.2.9.15. remove-sessionstorage-scope

세션 스토리지에서 스코프를 제거한다.

- 관련 스키마

jeus-session-server.xsd, jeus-session-domain.xsd

- alias

removessscope, rmssscope, rmsessionstorscope, removesessionstoragescope

- 사용법

```
remove-sessionstorage-scope <scope-name>
```

- 파라미터

파라미터	설명
<scope-name>	제거할 세션 스코프의 이름이다.

- 예제

```
remove-sessionstorage-scope scope1
```

#### 4.2.9.16. set-sessionstorage-scope-session-config

스코프에 포함된 컨텍스트에서 사용될 세션 관련 설정을 할 수 있다.

- 관련 스키마

jeus-session-server.xsd, jeus-session-domain.xsd

- alias

setsessionstoragescopesessionconfig, setssssc, setsessionstorscopesc

- 사용법

```
set-sessionstorage-scope-session-config <scope-name>
    [-timeout <timeout>]
    [-msc, --max-session-count <limit-count>]
    [-rp, --reload-persistent <reload-persistent>]
    [-tm, --tracking-mode <tracking-mode>]
    [-sc, --session-cookie <session-cookie>]
```

- 파라미터

파라미터	설명
<scope-name>	세션 관련 설정을 할 스코프의 이름이다.
[-timeout <timeout>]	session-config의 세션 타임아웃을 설정한다. (기본값: 30)
[-msc, --max-session-count <limit-count>]	저장할 세션의 최대 개수를 설정한다. (기본값: -1(무제한))
[-rp, --reload-persistent <reload-persistent>]	session-config의 Reload Persistent를 설정한다. (기본값: false)

파라미터	설명
<code>[-tm, --tracking-mode &lt;tracking-mode&gt;]</code>	<p>추가하거나 변경할 property이다. 이미 key가 설정되어 있다면 property의 value를 변경하고, 없다면 새로 추가한다.</p> <p>property에 들어갈 수 있는 key는 다음과 같다.</p> <ul style="list-style-type: none"> <li>• <code>cookie</code> : session-config의 트래킹 모드 중 쿠키를 설정한다. value의 type은 Boolean이다. (기본값: true)</li> <li>• <code>url</code> : session-config의 트래킹 모드 중 URL rewriting을 설정한다. value의 type은 Boolean이다. (기본값: false)</li> <li>• <code>ssl</code> : session-config의 트래킹 모드 중 SSL를 설정한다. value의 type은 Boolean이다. (기본값: false)</li> </ul>
<code>[-sc, --session-cookie &lt;session-cookie&gt;]</code>	<p>추가하거나 변경할 property이다. 이미 key가 설정되어 있다면 property의 value를 변경하고, 없다면 새로 추가한다.</p> <p>property에 들어갈 수 있는 key는 다음과 같다.</p> <ul style="list-style-type: none"> <li>• <code>cookie-name</code> : 쿠키를 사용하여 세션을 전달할 때 해당 세션 쿠키의 이름을 설정한다. value의 type은 String이다. (기본값: JSESSIONID)</li> <li>• <code>url-cookie-name</code> : url rewriting을 사용하여 세션을 전달할 때 해당 세션 쿠키의 이름을 설정한다. value의 type은 String이다. (기본값: jsessionid)</li> <li>• <code>version</code> : 쿠키 ID 버전을 설정한다. value의 type은 Integer이다. (기본값: 0)</li> <li>• <code>domain</code> : 세션 쿠키가 적용되는 도메인 이름을 설정한다. value의 type은 String이다. 한 글자 이상의 문자열이어야 한다.</li> <li>• <code>path</code> : 세션 쿠키가 적용되는 경로를 설정한다. value의 type은 String이다. 한 글자 이상의 문자열이어야 한다.</li> <li>• <code>max-age</code> : 브라우저에서 세션 쿠키의 유지 시간을 결정한다. value의 type은 Integer이다. (기본값: -1)</li> <li>• <code>secure</code> : 세션 ID 쿠키의 Secure 속성을 설정한다. true로 설정하면 세션 ID 쿠키는 오직 Secure HTTPS 커넥션으로 보내진다. value의 type은 Boolean이다. (기본값: false)</li> <li>• <code>http-only</code> : 브라우저가 HTTP 요청에 의해서만 세션 쿠키를 사용할지를 결정한다. value의 type은 Boolean이다. (기본값: true)</li> <li>• <code>same-site</code> : 세션 ID 쿠키가 사용자 의도와 상관없는 요청에 쓰이는 공격(사이트 간 요청위조)을 방지하는 보안기술을 설정한다. value의 Type은 [None, Strict, Lax, Disable] 중 하나이다.</li> <li>• <code>comment</code> : 해당 쿠키에 대한 목적 또는 설명을 기록한다. value의 type은 String이다. 한 글자 이상의 문자열이어야 한다.</li> </ul>



- 예제

```
set-sessionstorage-scope-session-config scope1 -timeout 20
-session-cookie secure=true,http-only=true,path=/ -tracking-mode cookie=true
```

#### 4.2.9.17. list-sessionserver-property

세션 서버의 property 목록을 보여준다.

- 관련 스키마

jeus-session-server.xsd, jeus-session-domain.xsd

- alias

listsessionserverproperty, listssp, listsessionserverprop

- 사용법

```
list-sessionserver-property
```

#### 4.2.9.18. set-sessionserver-property

세션 서버의 property를 추가하거나 변경 할 수 있다.

- 관련 스키마

jeus-session-server.xsd, jeus-session-domain.xsd

- alias

setsessionserverproperty, setsessionserverprop, setsservprop

- 사용법

```
set-sessionserver-property [-key <key-of-property> -value <value-of-property>] | [-property <property>]
```

- 파라미터

파라미터	설명
[-key <key-of-property>]	추가하거나 변경할 property의 key이다. 이미 key가 설정되어 있다면 property의 value를 변경하고, 없다면 새로 추가한다.  [-property] 옵션과 동시에 쓸 수 없다.

파라미터	설명
<code>[-value &lt;value-of-property&gt;]</code>	추가하거나 변경할 property의 value이다.  [-key] 옵션이 있다면 필수로 입력해야 한다.
<code>[-property &lt;property&gt;]</code>	추가하거나 변경할 property이다. 이미 key가 설정되어 있다면 property의 value를 변경하고, 없다면 새로 추가한다.  [-key] 옵션과 동시에 쓸 수 없다.



[-key] 또는 [-property] 중 하나는 필수로 입력해야 한다.

- 예제

- `[-key <key-of-property> -value <value-of-property>]` 옵션을 사용한 예

```
set-sessionserver-property -key encoding-rule -value base64
```

- `[-property <property>]` 옵션을 사용한 예

```
set-sessionserver-property -property encoding-rule=raw
```

#### 4.2.9.19. remove-sessionserver-property

세션 서버의 property를 제거할 수 있다.

- 관련 스키마

jeus-session-server.xsd, jeus-session-domain.xsd

- alias

removesessionserverproperty, rmsessionserverprop, rmsservprop

- 사용법

```
remove-sessionserver-property -key <key-of-property>
```

- 파라미터

파라미터	설명
<code>-key &lt;key-of-property&gt;</code>	세션 서버에서 제거할 property의 key이다.

- 예제

```
remove-sessionserver-property -key excluded-servers
```

#### 4.2.9.20. add-central-session-server

특정 서버를 중앙식 세션 서버로 동작하도록 설정할 수 있다.

- 관련 스키마

jeus-session-server.xsd, jeus-session-domain.xsd

- alias

addcss, add-css

- 사용법

```
add-central-session-server <server-name> -g <group-name>
```

- 파라미터

파라미터	설명
<server-name>	중앙식 세션 서버로 동작하고자 하는 대상 서버 이름이다.
-g <group-name>	중앙식 세션 서버 정의에 설정된 대상 그룹 이름이다.

- 예제

```
add-central-session-server server1 -g central1
```

#### 4.2.9.21. modify-central-session-server

특정 서버의 중앙식 세션 서버 대상 그룹의 이름을 변경할 수 있다.

- 관련 스키마

jeus-session-server.xsd, jeus-session-domain.xsd

- alias

mvcss, mv-css

- 사용법

```
modify-central-session-server <server-name> -g <group-name>
```

- 파라미터

파라미터	설명
<server-name>	변경하려는 대상 서버 이름이다.
-g <group-name>	변경하고자 하는 중앙식 세션 서버 그룹 이름이다.

- 예제

```
modify-central-session-server server1 -g central2
```

#### 4.2.9.22. remove-central-session-server

특정 서버를 중앙식 세션 서버로 동작하지 않도록 변경할 수 있다.

- 관련 스키마

jeus-session-server.xsd, jeus-session-domain.xsd

- alias

rmcss, rm-css

- 사용법

```
remove-central-session-server <server-name>
```

- 파라미터

파라미터	설명
<server-name>	중앙식 세션 서버 제외 대상 서버 이름이다.

- 예제

```
remove-central-session-server server1
```

#### 4.2.9.23. add-central-session-server-definition

대상 중앙식 세션 서버 그룹에 서버(primary 및 backup)를 새로 추가할 수 있다.

- 관련 스키마

jeus-session-server.xsd, jeus-session-domain.xsd

- alias

add-central-session-def, addcsd, add-csd

- 사용법

```
add-central-session-server-definition -g <group-name> -p <primary> [-b <backup>]
```

- 파라미터

파라미터	설명
-g <group-name>	서버를 추가하고자 하는 대상 그룹 이름이다.
-p <primary>	Primary 서버 주소이다. IP(hostname):port 형식으로 설정한다.
[-b <backup>]	Backup 서버 주소이다. IP(hostname):port 형식으로 설정한다. 여러 개를 설정할 경우에는 콤마(,)로 구분한다.

- 예제

```
add-central-session-server-definition -g central1 -p localhost:9736 -b  
localhost:19736,localhost:29736
```

#### 4.2.9.24. remove-central-session-server-definition

대상 중앙식 세션 서버 정의에서 그룹을 제거하거나 특정 서버를 제거한다.

- 관련 스키마

jeus-session-server.xsd, jeus-session-domain.xsd

- alias

remove-central-session-def, rmcsd, rm-csd

- 사용법

```
remove-central-session-server-definition -g <group-name> [-p <primary>] [-b <backup>]
```

- 파라미터

파라미터	설명
-g <group-name>	제거하고자 하는 대상 그룹 이름이다.

파라미터	설명
[-p <primary>]	제거하고자 하는 대상 그룹의 Primary 서버 주소이다.  1개 이상의 서버는 설정에서 유지되어야 한다.
[-b <backup>]	제거하고자 하는 대상 그룹의 Backup 서버 주소이다.  'IP(hostname):Port' 형식으로 설정한다. 만약 여러 개를 설정할 경우에는 콤마(,)로 구분한다.

- 예제

```
remove-central-session-server-definition -g central1 -p localhost:9736 -b localhost:9735
remove-central-session-server-definition -g central1 -p localhost:9736
remove-central-session-server-definition -g central2
```

#### 4.2.9.25. list-central-session-server-session

대상 서버의 중앙식 세션 서버에 존재하는 세션의 정보를 보여준다.

- 관련 스키마

jeus-session-server.xsd, jeus-session-domain.xsd

- alias

list-css-session, lscsss

- 사용법

```
list-central-session-server-session [-server <server-name>] [-a,--address <server-address>] [-c,--count <count> ] [-i,--id <expectedToContain>] [-s,--simple]
```

- 파라미터

파라미터	설명
[-server <server-name>]	동일한 도메인 안의 세션 서버의 서버 이름이다.
[-a,--address <server-address>]	조회할 세션 서버의 주소이다.
[-c,--count <count>]	조회할 세션의 개수이다. (기본값: 100)
[-i,--id <expectedToContain>]	문자열이 포함된 세션을 조회할 때 사용한다.
[-s,--simple]	출력 포맷을 간략하게 표시할 경우 사용한다.

- 예제

```
list-central-session-server-session -a localhost:19736
```

#### 4.2.9.26. show-central-session-server-statistics

대상 서버의 중앙식 세션 서버의 메모리 및 세션 정보를 조회한다.

- 관련 스키마

jeus-session-server.xsd, jeus-session-domain.xsd

- alias

cssst, stcss, css-stat

- 사용법

```
show-central-session-server-statistics [-server <server-name>] [-a,--address <server-address>] [-s,--session] [-m,--memory]
```

- 파라미터

파라미터	설명
[-server <server-name>]	동일한 도메인 안의 세션 서버의 서버 이름이다.
[-a,--address <server-address>]	조회할 세션 서버의 주소이다.
[-s,--session]	세션 서버의 정보를 출력한다. 이때 세션 서버의 이름, 세션의 수, 백업 서버를 포함한다.
[-m,--memory]	구성 중인 서버 JVM이 사용하고 있는 메모리 상태를 조회한다.

- 예제

```
show-central-session-server-statistics -a localhost:19736
```

#### 4.2.10. JMS 엔진 관련 명령어

JMS 엔진 명령어는 다음과 같다.

명령어	설명
<a href="#">ban-jms-client</a>	지정된 JMS 클라이언트와의 연결을 강제로 끊는다.
<a href="#">add-jms-connection-factory</a>	새로운 Connection Factory를 추가한다.
<a href="#">add-jms-destination</a>	새로운 Destination을 추가한다.
<a href="#">add-jms-durable-subscription</a>	새로운 Durable Subscription을 추가한다.

명령어	설명
<a href="#">add-jms-message-sort</a>	새로운 Message Sort를 추가한다.
<a href="#">add-jms-quota</a>	새로운 Quota를 추가한다.
<a href="#">add-jms-service-config</a>	새로운 Service Config를 추가한다.
<a href="#">commit-jms-pending-transaction</a>	지정된 pending 트랜잭션을 강제로 commit한다.
<a href="#">control-jms-destination</a>	지정된 Destination의 서비스 상태를 제어한다.
<a href="#">control-jms-durable-subscription</a>	지정된 Durable Subscription의 서비스 상태를 제어한다.
<a href="#">delete-jms-messages</a>	지정된 Destination 안의 지정된 메시지를 삭제한다.
<a href="#">export-jms-messages</a>	지정된 Destination 안의 지정된 메시지를 XML 형태로 내보낸다.
<a href="#">import-jms-messages</a>	지정된 Destination로 내보내진 XML 형태의 메시지를 가져온다.
<a href="#">list-jms-clients</a>	접속 중인 클라이언트 목록을 조회하고 정보를 출력한다.
<a href="#">list-jms-connection-factories</a>	Connection Factory 목록을 조회하거나 지정된 Connection Factory의 정보를 출력한다.
<a href="#">list-jms-destinations</a>	Destination 목록을 조회하거나 지정된 Destination 정보를 출력한다.
<a href="#">list-jms-durable-subscriptions</a>	Durable Subscription 목록을 조회한다.
<a href="#">list-jms-message-sorts</a>	Message Sort 목록을 조회한다.
<a href="#">list-jms-messages</a>	지정된 Destination 안의 메시지 정보를 조회한다.
<a href="#">list-jms-pending-transactions</a>	pending 트랜잭션 목록을 출력한다.
<a href="#">list-jms-quotas</a>	Quota 목록을 조회하거나 지정된 Quota 정보를 출력한다.
<a href="#">list-jms-service-configs</a>	Service Config 목록을 조회하거나 지정된 Service Config 정보를 출력한다.
<a href="#">modify-jms-connection-factory</a>	지정된 Connection Factory 정보를 수정한다.
<a href="#">modify-jms-destination</a>	지정된 Destination 정보를 수정한다.
<a href="#">modify-jms-durable-subscription</a>	지정된 Durable Subscription 정보를 수정한다.
<a href="#">modify-jms-message-sort</a>	지정된 Message Sort 정보를 수정한다.
<a href="#">modify-jms-quota</a>	지정된 Quota 정보를 수정한다.
<a href="#">modify-jms-service-config</a>	지정된 Service Config 정보를 수정한다.
<a href="#">move-jms-messages</a>	지정된 Destination 안의 지정된 메시지를 주어진 Destination으로 옮긴다.
<a href="#">remove-jms-connection-factory</a>	지정된 Connection Factory를 제거한다.
<a href="#">remove-jms-destination</a>	지정된 Destination을 제거한다.
<a href="#">remove-jms-durable-subscription</a>	지정된 Durable Subscription을 제거한다.
<a href="#">remove-jms-message-sort</a>	지정된 Message Sort를 제거한다.
<a href="#">remove-jms-quota</a>	지정된 Quota를 제거한다.
<a href="#">remove-jms-service-config</a>	지정된 Service Config를 제거한다.



명령어	설명
<code>set-jms-engine-quota</code>	JMS 엔진의 Quota를 설정한다.
<code>set-jms-failover-check</code>	Failover 전 JMS 엔진의 생존 여부 확인을 위한 옵션을 설정한다.
<code>set-jms-persistence-store</code>	JMS 엔진의 Persistence Store를 설정한다.
<code>view-jms-message</code>	지정된 메시지의 상세한 정보를 조회한다.

#### 4.2.10.1. 공통적용 사항

다음은 JMS 엔진 관련 명령어에서 공통적으로 적용되는 옵션에 대한 설명이다. 아래 옵션 중 하나만 설정될 수 있다.

- `[-cluster <cluster-name> ]` 옵션
  - 각 명령어가 적용될 클러스터를 지정할 때 사용하는 옵션이다. Master Server에 연결된 상태에서만 사용 가능하다. MS에 접속된 상태에서는 해당 서버에 명령어가 적용된다.
  - Destination에 관련된 명령어에서만 지원한다.
- `[-server <server-name> ]` 옵션
  - 각 명령어가 적용될 서버를 지정할 때 사용하는 옵션이다. Master Server에 연결된 상태에서만 사용 가능하다. MS에 접속된 상태에서는 해당 서버에 명령어가 적용된다.
  - Destination에 관련된 것 이외의 명령어에서만 지원한다.

#### 4.2.10.2. add-jms-connection-factory

동적으로 JMS 엔진에 새로운 Connection Factory를 추가한다. Queue 또는 Topic Connection Factory를 생성할 수 있다.

- alias  
`add-connection-factory, addconf, createconf`
- 사용법

```
add-jms-connection-factory -server <server-name>
                           -type <factory-type>
                           -name <factory-name>
                           [-export <export-name>]
                           [-clientid <client-id>]
```

- 파라미터

파라미터	설명
<code>-server &lt;server-name&gt;</code>	대상이 되는 서버를 설정한다.

파라미터	설명
-type <factory-type>	<p>Connection Factory의 type을 결정한다.</p> <p>다음 중에 하나를 설정한다.</p> <ul style="list-style-type: none"> <li>• nonxa</li> <li>• xa</li> <li>• queue</li> <li>• topic</li> <li>• xaqueue</li> <li>• xatopic</li> </ul>
-name <factory-name>	<p>새로 생성할 Connection Factory의 이름을 입력한다.</p> <p>이 이름은 JMS 엔진 내에서 다른 Connection Factory와 중복되지 않도록 주의한다.</p>
[-export <export-name>]	<p>새로 생성할 Connection Factory의 JNDI 이름을 입력한다.</p> <p>이 이름은 JEUS JNDI 서버 내에서 다른 객체와 중복되지 않도록 주의한다.</p>
[-clientid <client-id>]	<p>Connection Factory로부터 생성한 커넥션에 부여할 클라이언트 ID를 입력한다.</p>

- 예제

```
[MASTER]domain1.adminServer>add-jms-connection-factory -server adminServer -type nonxa -name
MyConnectionFactory
Executed Successfully.
View the results using 'list-jms-connection-factories -server adminServer -name
MyConnectionFactory'.
[MASTER]domain1.adminServer>list-jms-connection-factories -server adminServer
Connection Factory Information
=====
+-----+-----+-----+
|          Factory Name          |          Export Name          |   Type   |
+-----+-----+-----+
| ConnectionFactory              | ConnectionFactory             | nonxa    |
| MyConnectionFactory           | MyConnectionFactory         | nonxa    |
| XAConnectionFactory           | XAConnectionFactory         | xa       |
+-----+-----+-----+
=====
```

#### 4.2.10.3. add-jms-destination

동적으로 JMS 엔진에 새로운 Destination을 추가한다.

- alias

add-destination, adddest, createdest

- 사용법

```
add-jms-destination -cluster <cluster-name> | -server <server-name>
                    -type <destination-type>
                    -name <destination-name>
                    [-export <export-name>]
```

- 파라미터

파라미터	설명
-cluster <cluster-name>   -server <server-name>	Destination이 속할 클러스터 또는 서버를 지정한다.
-type <destination-type>	Destination의 type을 결정한다. (queue 또는 topic)
-name <destination-name>	새로 생성할 Destination의 이름을 입력한다.  이 이름은 JMS 엔진 내에서 다른 Destination과 중복되지 않도록 주의한다.
[-export <export-name>]	새로 생성할 Destination의 JNDI 이름을 입력한다.  이 이름은 JEUS JNDI 서버 내에서 다른 객체와 중복되지 않도록 주의한다. 지정하지 않으면 destination-name과 동일한 값이 지정된다.

- 예제

```
[MASTER]domain1.adminServer>add-jms-destination -server adminServer -type queue -name MyQueue
Executed Successfully.
View the results using 'list-jms-destinations -server adminServer -name MyQueue'.
[MASTER]domain1.adminServer>list-jms-destinations -server adminServer
Destination information in Server adminServer
=====
+-----+-----+-----+-----+-----+-----+-----+
| Name   | Export | Type | Remaining | Dead Letter | Produce | Consume |
|        | Name   |      | Messages  | Destination | Suspended | Suspended |
+-----+-----+-----+-----+-----+-----+-----+
| MyQueue | MyQueue | Queue | 0          | JEUSMQ_DLQ  | false   | false   |
+-----+-----+-----+-----+-----+-----+-----+
| Examples | Examples | Topic | 0          | JEUSMQ_DLQ  | false   | false   |
| Topic    | Topic    |      |            |              |         |         |
+-----+-----+-----+-----+-----+-----+-----+
| Examples | Examples | Queue | 0          | JEUSMQ_DLQ  | false   | false   |
| Queue    | Queue    |      |            |              |         |         |
+-----+-----+-----+-----+-----+-----+-----+
| JEUSMQ_D | JEUSMQ_D | Queue | 0          | JEUSMQ_DLQ  | false   | false   |
| LQ       | LQ       |      |            |              |         |         |
+-----+-----+-----+-----+-----+-----+-----+

For detailed information, use the -name option
```

=====

#### 4.2.10.4. add-jms-durable-subscription

동적으로 JMS 엔진에 새로운 Durable Subscription을 추가한다.

- alias

add-durable, adddur, createdur

- 사용법

```
add-jms-durable-subscription -cluster <cluster-name> | -server <server-name>
                             -clientid <client-id>
                             -name <durable-subscription-name>
                             -topic <topic-name>
                             [-shared]
```

- 파라미터

파라미터	설명
-cluster <cluster-name>   -server <server-name>	Durable Subscription이 속할 클러스터 또는 서버를 지정한다.
-clientid <client-id>	subscribe 하게 될 client의 id를 지정한다.
-name <durable-subscription-name>	새로 생성할 Durable Subscription의 이름을 입력한다.  이 이름은 JMS 엔진 내에서 다른 Durable Subscription과 중복되지 않도록 주의한다.
-topic <topic-name>	Durable Subscription에 대한 Topic의 이름을 입력한다.
[-shared]	Shared 형태의 Durable Subscription을 생성할 경우 사용하는 옵션이다.

- 예제

```
[MASTER]domain1.adminServer>add-jms-durable-subscription -server adminServer -clientid testClient
-name testDurable -topic testTopic
Executed Successfully.
View the results using 'list-jms-durable-subscriptions -server adminServer'.
[MASTER]domain1.adminServer>list-jms-durable-subscriptions -server adminServer
Durable Subscription Information
=====
+-----+-----+-----+-----+-----+
| Durable Name | Client ID | Shared | Message Selector | Remaining Messages |
+-----+-----+-----+-----+-----+
| testDurable | testClient| false  |                  | 0                  |
+-----+-----+-----+-----+-----+
```

=====

#### 4.2.10.5. add-jms-message-sort

JMS 엔진에 새로운 Message Sort를 추가한다.

- alias

add-message-sort, addmsgsort

- 사용법

```
add-jms-message-sort -server <server-name>
                    -name <message-sort-name>
                    -key <key>
                    [-type <type>]
                    -direction <direction>
```

- 파라미터

파라미터	설명
-server <server-name>	Message Sort가 속할 서버를 지정한다.
-name <message-sort-name>	새로 생성할 Message Sort의 이름을 입력한다.  이 이름은 JMS 엔진 내에서 다른 Message Sort와 중복되지 않도록 주의한다.
-key <key>	메시지 정렬의 기준이 되는 Key를 입력한다.
[-type <type>]	Key의 타입을 입력한다.
-direction <direction>	메시지 정렬 순서를 입력한다.

- 예제

```
[MASTER]domain1.adminServer>add-jms-message-sort -server adminServer -name testSort -key
JMSPriority -type Integer -direction Ascending
Executed successfully, but some configurations were not applied dynamically. It might be
necessary to restart the server.
View the results using 'list-jms-message-sorts -server adminServer -name testSort'.
[MASTER]domain1.adminServer>list-jms-message-sorts -server adminServer -name testSort
Detailed Message Sort Information
=====
testSort

+-----+-----+
| Message Sort Name | testSort |
| Key               | JMSPriority |
| Type             | INTEGER   |
| Direction        | ASCENDING  |
```

```
+-----+-----+
=====
```

#### 4.2.10.6. add-jms-quota

JMS 엔진에 새로운 Quota를 추가한다.

- alias

add-quota, addquota

- 사용법

```
add-jms-quota -server <server-name>
               -name <quota-name>
               [-byte <max-bytes>]
               [-msg <max-messages>]
               [-shared <shared>]
```

- 파라미터

파라미터	설명
-server <server-name>	Quota가 속할 서버를 지정한다.
-name <quota-name>	새로 생성할 Quota의 이름을 입력한다. 이 이름은 JMS 엔진 내에서 다른 Quota와 중복되지 않도록 주의한다.
[-byte <max-bytes>]	제한 할 최대 Byte를 입력한다 .
[-msg <max-messages>]	제한 할 최대 메시지 수를 입력한다.
[-shared <shared>]	Quota를 여러 Destination에서 공유하여 사용할지 여부를 입력한다.

- 예제

```
[MASTER]domain1.adminServer>add-jms-quota -server adminServer -name testQuota -byte 128M -msg 1M
-shared false
Executed successfully, but some configurations were not applied dynamically. It might be
necessary to restart the server.
View the results using 'list-jms-quotas -server adminServer -name testQuota'.
[MASTER]domain1.adminServer>list-jms-quotas -server adminServer -name testQuota
Detailed Quota Information
=====
testQuota

+-----+-----+
| Quota Name           | testQuota |
| Max Bytes            | 128M      |
| Max Messages         | 1M        |
| Shared               | false     |
+-----+-----+
```

=====

#### 4.2.10.7. add-jms-service-config

JMS 엔진에 새로운 서비스를 추가한다.

- alias

add-service-config, addjmsservice

- 사용법

```
add-jms-service-config -server <server-name>
                        -name <service-name>
                        [-limit <client-limit>]
                        [-timeout <client-keepalive-timeout>]
                        [-listener <listener>]
                        [-virtual]
                        [-address <address>]
                        [-port <port>]
```

- 파라미터

파라미터	설명
-server <server-name>	서비스가 속할 서버를 지정한다.
-name <service-name>	새로 생성할 서비스의 이름을 입력한다. 이 이름은 JMS 엔진 내에서 다른 서비스와 중복되지 않도록 주의한다.
[-limit <client-limit>]	허용 되는 client의 최대값을 입력한다 .
[-timeout <client-keepalive-timeout>]	비정상 종료 될 때 재연결을 기다릴 시간을 입력한다.
[-listener <listener>]	서비스의 Listener를 입력한다 .
[-virtual]	Virtual Listener를 선택할 경우 사용하는 옵션이다 .
[-address <address>]	Virtual Listener의 address를 입력한다 .
[-port <port>]	Virtual Listener의 port를 입력한다.

- 예제

```
[MASTER]domain1.adminServer>add-jms-service-config -server adminServer -name testService -limit
1000 -timeout 30 -listener base
Executed Successfully.
View the results using 'list-jms-service-configs -server adminServer -name testService'.
[MASTER]domain1.adminServer>list-jms-service-configs -server adminServer -name testService
Detailed Service Config Information
=====
```

```
testService
```

```
+-----+-----+
| Service Config Name          | testService |
| Client Limit                 |      1000   |
| Client Keepalive Timeout     |      30     |
| Listener Name                | base        |
+-----+-----+
=====
```

#### 4.2.10.8. ban-jms-client

JMS 클라이언트와의 연결을 강제로 끊는다.

- alias

killentry, banentry

- 사용법

```
ban-jms-client -server <server-name>
               -name <entry-name>
```

- 파라미터

파라미터	설명
-server <server-name>	대상이 되는 서버를 설정한다.
-name <entry-name>	강제로 연결을 끊을 JMS 클라이언트의 이름을 설정한다. 이 이름은 <a href="#">list-jms-clients</a> 명령어로 확인할 수 있다.

#### 4.2.10.9. commit-jms-pending-transaction

Pending 트랜잭션을 강제로 commit한다.

- alias

jmsptcommit, ptcommit

- 사용법

```
commit-jms-pending-transaction -server <server-name>
                                -id <tx-id>
```

- 파라미터



파라미터	설명
-server <server-name>	대상이 되는 서버를 설정한다.
-id <tx-id>	강제로 commit할 트랜잭션의 ID이다. ID는 <a href="#">list-jms-pending-transactions</a> 명령어로 확인할 수 있다.

#### 4.2.10.10. control-jms-destination

JMS 엔진에서 특정 Destination의 상태를 제어한다.

- alias  
destctrl
- 사용법

```
control-jms-destination -cluster <cluster-name> | -server <server-name>
                        -dest <destination-name>
                        -suspend | -resume
                        [-produce]
                        [-consume]
```

- 파라미터

파라미터	설명
-cluster <cluster-name>   -server <server-name>	Destination이 속한 클러스터 또는 서버를 지정한다.
-dest <destination-name>	제어할 Destination의 이름이다.
-suspend   -resume	제어할 Destination의 상태이다. <ul style="list-style-type: none"> <li>• suspend : 해당 Destination의 서비스가 일시 정지된다.</li> <li>• resume : 해당 Destination의 서비스가 재개된다.</li> </ul>
[-produce]	지정된 Destination에 produce를 제어할지를 지정한다.
[-consume]	지정된 Destination에서의 consume을 제어할지를 지정한다.

- 예제

```
[MASTER]domain1.adminServer>control-jms-destination -server adminServer -dest ExamplesQueue
-suspend -produce
Controlling destination is successful in ExamplesQueue : [production] is suspended

[MASTER]domain1.adminServer>control-jms-destination -server adminServer -dest ExamplesQueue
-resume -produce
Controlling destination is successful in ExamplesQueue : [production] is resumed
```

#### 4.2.10.11. control-jms-durable-subscription

JMS 엔진에서 특정 Durable Subscription의 상태를 제어한다.

- alias

control-durable, durctrl

- 사용법

```
control-jms-durable-subscription -cluster <cluster-name> | -server <server-name>
                                -name <durable-subscription-name>
                                -suspend | -resume
```

- 파라미터

파라미터	설명
-cluster <cluster-name>   -server <server-name>	Durable Subscription이 속한 클러스터 또는 서버를 지정한다.
-name <durable-subscription-name>	제어할 Durable Subscription의 이름이다.
-suspend   -resume	제어할 Durable Subscription의 상태이다. <ul style="list-style-type: none"><li>• suspend : 해당 Durable Subscription의 서비스가 일시 정지된다.</li><li>• resume : 해당 Durable Subscription의 서비스가 재개된다.</li></ul>

- 예제

```
[MASTER]domain1.adminServer>control-jms-durable-subscription -server adminServer -name testDS
-suspend
Executed Successfully
View the results using 'list-jms-durable-subscriptions -server adminServer'.
[MASTER]domain1.adminServer>list-jms-durable-subscriptions -server adminServer
Durable Subscription Information
=====
+-----+-----+-----+-----+-----+-----+-----+
| Durable | Client | Shared | Message | Remaining Messages | Suspended |
| Name   | ID     |        | Selector | (current)           | (current) |
+-----+-----+-----+-----+-----+-----+-----+
| testDS | testDS | false  |          | 0                    | true      |
+-----+-----+-----+-----+-----+-----+-----+
=====
```

#### 4.2.10.12. delete-jms-messages

Queue나 Durable subscriber에 쌓여있는 특정 메시지 또는 메시지 전체를 삭제한다.

- alias

deletemsg, deletemessage, removemsg, removemessage, rmmsg

- 사용법

```
delete-jms-message -cluster <cluster-name> | -server <server-name>
                    -dest <destination-name> | -durable <client-id>_<durable-name>
                    -id <message-id> | -all
```

- 파라미터

파라미터	설명
-cluster <cluster-name>   -server <server-name>	삭제할 메시지가 있는 Destination이나 Durable subscriber가 속한 클러스터나 서버를 지정한다.
-dest <destination-name>   -durable <client-id>_<durable-name>	삭제할 메시지가 있는 Destination이나 Durable subscriber의 이름이다.
-id <message-id>   -all	삭제할 메시지를 지정한다. 개별 ID를 지정하거나 모든 메시지를 지정할 수 있다.

#### 4.2.10.13. export-jms-messages

지정된 Destination의 특정 메시지들을 XML 형태로 내보낸다.

- alias

exmsg

- 사용법

```
export-jms-message -cluster <cluster-name> | -server <server-name>
                   -dest <destination-name> | -durable <client-id>_<durable-name>
                   -path <file-path>
                   -id <message-id> | -all
```

- 파라미터

파라미터	설명
-cluster <cluster-name>   -server <server-name>	내보낼 메시지가 있는 Destination이나 Durable subscriber가 속한 클러스터나 서버를 지정한다.
-dest <destination-name>   -durable <client-id>_<durable-name>	내보낼 메시지가 있는 Destination이나 Durable subscriber의 이름이다.

파라미터	설명
-path <file-path>	내보낼 메시지가 저장될 파일의 경로를 지정한다.
-id <message-id>   -all	내보낼 메시지를 지정한다. 개별 ID를 지정하거나 모든 메시지를 지정할 수 있다.

- 예제

```
[MASTER]domain1.adminServer>export-jms-messages -server adminServer -dest ExamplesQueue -all
-path exported.xml
Successfully exported 5 messages.
```

#### 4.2.10.14. import-jms-messages

Destination에 내보내진 XML 파일로부터 메시지들을 가져온다.

- alias

immsg

- 사용법

```
import-jms-message -cluster <cluster-name> | -server <server-name>
                    -dest <destination-name>
                    -path <file-path>
                    [-overwrite]
```

- 파라미터

파라미터	설명
-cluster <cluster-name>   -server <server-name>	메시지를 가져올 Destination이 속한 클러스터나 서버를 지정한다.
-dest <destination-name>	메시지를 가져올 Destination의 이름이다.
-path <file-path>	가져올 메시지가 저장된 파일의 경로를 지정한다.
[-overwrite]	가져올 메시지의 JMSMessageID를 새로 발급하지 않는다. 같은 ID가 존재하는 경우 덮어 쓴다.

- 예제

```
[MASTER]domain1.adminServer>list-jms-messages -server adminServer -dest ExamplesQueue
There are no messages in ExamplesQueue.

[MASTER]domain1.adminServer>import-jms-messages -server adminServer -dest ExamplesQueue -path
exported.xml
Successfully imported 5 of 5 messages.
```

```
[MASTER]domain1.adminServer>list-jms-messages -server adminServer -dest ExamplesQueue
Messages in Destination ExamplesQueue
```

Message ID	Message Type	Created Time
ID:796886517700001:1:5	Text	Tue Nov 19 19:52:54 KST 2016
ID:796886517700001:1:6	Text	Tue Nov 19 19:52:54 KST 2016
ID:796886517700001:1:7	Text	Tue Nov 19 19:52:54 KST 2016
ID:796886517700001:1:8	Text	Tue Nov 19 19:52:54 KST 2016
ID:796886517700001:1:9	Text	Tue Nov 19 19:52:54 KST 2016

#### 4.2.10.15. list-jms-clients

현재 접속해 있는 모든 클라이언트의 목록을 조회한다.

- alias

jmsclient, jmsentry

- 사용법

```
list-jms-clients -server <server-name>
```

- 파라미터

파라미터	설명
-server <server-name>	클라이언트의 목록을 조회할 서버를 지정한다.

- 예제

```
[MASTER]domain1.adminServer>list-jms-clients -server adminServer
JEUS MQ client information
```

Entry Name	Remote Address	Start Time	Connection Count	Session Count
JMSClient-RE63 675900002	192.168.34.33/192.168.34.33:49490	Tue Nov 07 16:14:32 KST 2016	1	1
JMSClient-RE63 675900001	192.168.34.33/192.168.34.33:49484	Tue Nov 07 16:14:23 KST 2016	1	1

## 4.2.10.16. list-jms-connection-factories

JMS 엔진에 등록되어 있는 모든 Connection Factory들의 정보를 조회하거나 지정된 Connection Factory의 정보를 출력한다.

- alias

jmscf, jmsconf

- 사용법

```
list-jms-connection-factories -server <server-name>
                               [-n, --name <factory-name>]
```

- 파라미터

파라미터	설명
-server <server-name>	Connection Factory가 속한 서버를 지정한다.
[-n, --name <factory-name>]	정보를 출력할 Connection Factory의 이름이다. 지정하지 않으면 모든 Connection Factory에 관한 간략한 정보가 출력된다.

- 예제

```
[MASTER]domain1.adminServer>list-jms-connection-factories -server adminServer
Connection Factory Information
=====
+-----+-----+-----+
| Factory Name | Export Name | Type |
+-----+-----+-----+
| ConnectionFactory | ConnectionFactory | nonxa |
| XAConnectionFactory | XAConnectionFactory | xa |
+-----+-----+-----+
=====

[MASTER]domain1.adminServer>list-jms-connection-factories -server adminServer -name
ConnectionFactory
=====
ConnectionFactory
+-----+-----+
| Factory Name | ConnectionFactory |
| Export Name | ConnectionFactory |
| Type | nonxa |
| Client ID | not-set |
| Max Client Session Threads | 100 |
| Clustered | false |
| Broker Selection Policy | round-robin |
+-----+-----+
=====
=====
+-----+
```

```

|-----+-----+-----+-----+-----+-----+-----+-----+
|                               Addresses                               |
|-----+-----+-----+-----+-----+-----+-----+-----+
| 192.168.0.26:9741(JMSServiceChannel-internal)                       |
|-----+-----+-----+-----+-----+-----+-----+-----+
|=====|

```

#### 4.2.10.17. list-jms-destinations

JMS 엔진에 등록되어 있는 모든 Destination의 목록을 조회하거나 지정한 Destination의 정보를 출력한다.

- alias

jmsdest, dest

- 사용법

```

list-jms-destinations -cluster <cluster-name> | -server <server-name>
                        [-n, --name <destination-name>]

```

- 파라미터

파라미터	설명
-cluster <cluster-name>   -server <server-name>	Destination이 속한 클러스터 또는 서버를 지정한다.
[-n, --name <destination-name>]	정보를 출력할 Destination의 이름이다. 지정하지 않으면 모든 Destination에 대한 간략한 정보가 출력된다.

- 예제

```

[[MASTER]domain1.adminServer>list-jms-destinations -server adminServer
Destination information in Server adminServer
=====
+-----+-----+-----+-----+-----+-----+-----+-----+
|  Name  | Export | Type | Remaining | Dead Letter | Produce | Consume |
|  Name  | Name   |      | Messages  | Destination | Suspended | Suspended |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Examples| Examples| Topic |           | JEUSMQ_DLQ | false  | false  |
| Topic   | Topic   |      |           |           |        |        |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Examples| Examples| Queue |           | JEUSMQ_DLQ | false  | false  |
| Queue   | Queue   |      |           |           |        |        |
+-----+-----+-----+-----+-----+-----+-----+-----+
| JEUSMQ_D| JEUSMQ_D| Queue |           | JEUSMQ_DLQ | false  | false  |
| LQ      | LQ      |      |           |           |        |        |
+-----+-----+-----+-----+-----+-----+-----+-----+

For detailed information, use the -name option
=====

[[MASTER]domain1.adminServer>list-jms-destinations -server adminServer -name ExamplesQueue

```

Detailed destination information in Server adminServer

ExamplesQueue

Export Name	ExamplesQueue
Type	Queue
Dead Letter Destination	JEUSMQ_DLQ
Consumer Count	0
Processed Messages	0
Remaining Messages	0
Pending Messages	0
Dispatched Messages	0
Delivered Messages	0
Expired Messages	0
Moved Messages	0
Memory Usage (current)	0kb
Memory Usage (high mark)	0kb
Produce Suspended	false
Consume Suspended	false

#### 4.2.10.18. list-jms-durable-subscriptions

JMS 엔진에 등록되어 있는 모든 Durable Subscription의 정보를 출력한다.

- alias

durable, dur

- 사용법

```
list-jms-durable-subscriptions -cluster <cluster-name> | -server <server-name>
```

- 파라미터

파라미터	설명
-cluster <cluster-name>   -server <server-name>	Durable Subscription의 목록을 조회할 클러스터나 서버를 지정한다.

- 예제

```
[MASTER]domain1.adminServer>list-jms-durable-subscriptions -server adminServer
Durable Subscription Information
=====
+-----+-----+-----+-----+-----+
| Durable Name | Client ID | Shared | Message Selector | Remaining Messages |
+-----+-----+-----+-----+-----+
| testDurable | testClient | false | | 5 |
+-----+-----+-----+-----+-----+
```



=====

#### 4.2.10.19. list-jms-message-sorts

JMS 엔진에 등록되어 있는 모든 Message Sort의 목록을 조회하거나 지정한 Message Sort의 정보를 출력한다.

- alias

list-message-sorts, msgsort

- 사용법

```
list-jms-destinations -server <server-name>
                        [-name <message-sort-name>]
```

- 파라미터

파라미터	설명
-server <server-name>	Message Sort가 속한 서버를 지정한다.
[-name <message-sort-name>]	정보를 출력할 Message Sort의 이름이다. 지정하지 않으면 모든 Message Sort에 대한 간략한 정보가 출력된다.

- 예제

```
[[MASTER]domain1.adminServer>list-jms-message-sorts -server adminServer
Message Sort Information
=====
+-----+-----+-----+
| Message Sort Name | Key | Type |
+-----+-----+-----+
| testSort          | JMSPriority | INTEGER |
+-----+-----+-----+
=====

[[MASTER]domain1.adminServer>list-jms-message-sorts -server adminServer -name testSort
Detailed Message Sort Information
=====
testSort

+-----+-----+-----+
| Message Sort Name | testSort |
| Key               | JMSPriority |
| Type              | INTEGER   |
| Direction         | DESCENDING |
+-----+-----+-----+
=====
```

#### 4.2.10.20. list-jms-messages

Destination의 메시지들의 정보를 출력한다.

- alias

message, msg

- 사용법

```
list-jms-messages -cluster <cluster-name> | -server <server-name>
                  -dest <destination-name> | -durable <client-id>_<durable-name>
                  [-s,--selector <message-selector>]
                  [-offset <offset>]
                  [-size <size>]
                  [-id <message-id-pattern>]
                  [-type <message-type>]
                  [-from <YYYY:MM:DD:HH:MM:SS>]
                  [-to <YYYY:MM:DD:HH:MM:SS>]
```

- 파라미터

파라미터	설명
-cluster <cluster-name>   -server <server-name>	Destination이나 Durable subscriber가 속한 클러스터나 서버를 지정한다.
-dest <destination-name>   -durable <client-id>_<durable-name>	출력할 메시지가 있는 Destination이나 Durable subscriber의 이름이다.
[-s, --selector <message-selector>]	출력할 메시지를 선택할 수 있는 Message selector를 지정한다.
[-offset <offset>]	메시지를 나누어 출력할 때 몇 번째 메시지부터 출력할지를 지정한다.
[-size <size>]	메시지를 나누어 출력할 때 한 번에 출력할 메시지의 개수를 지정한다.
[-id <message-id-pattern>]	출력할 메시지를 선택할 수 있는 message id의 패턴을 지정한다.
[-type <message-type>]	출력할 메시지를 선택할 수 있는 메시지의 타입을 지정한다.
[-from <YYYY:MM:DD:HH:MM:SS>]	특정 시점 이후에 들어온 메시지를 선택할 때 그 시점을 지정한다.
[-to <YYYY:MM:DD:HH:MM:SS>]	특정 시점 이전에 들어온 메시지를 선택할 때 그 시점을 지정한다.

- 예제

```
[MASTER]domain1.adminServer>list-jms-messages -server adminServer -dest ExamplesQueue
Messages in Destination ExamplesQueue.
```

```
=====
+-----+-----+-----+
|      Message ID      | Message Type |      Created Time      |
+-----+-----+-----+
```

ID:7968865177000001:1:1	Text	Mon Nov 07 20:10:01 KST 2016
ID:7968865177000001:1:2	Text	Mon Nov 07 20:10:03 KST 2016
ID:7968865177000001:1:3	Text	Mon Nov 07 20:10:04 KST 2016
ID:7968865177000001:1:4	Text	Mon Nov 07 20:10:06 KST 2016

+-----+-----+-----+  
=====

#### 4.2.10.21. list-jms-pending-transactions

indoubt 상태의 트랜잭션 목록들을 조회한다.

- alias

jmspt, pt

- 사용법

```
list-jms-pending-transactions -server <server-name>
```

- 파라미터

파라미터	설명
-server <server-name>	indoubt 상태인 트랜잭션을 조회할 서버를 지정한다.

#### 4.2.10.22. list-jms-quotas

JMS 엔진에 등록되어 있는 모든 Quota의 목록을 조회하거나 지정한 Quota의 정보를 출력한다.

- alias

list-quotas, jmsquota

- 사용법

```
list-jms-quotas -server <server-name>
                [-name <quota-name>]
```

- 파라미터

파라미터	설명
-server <server-name>	Quota가 속한 서버를 지정한다.
[-name <quota-name>]	정보를 출력할 Quota의 이름이다. 지정하지 않으면 모든 Quota에 대한 간략한 정보가 출력된다.

- 예제

```

[[MASTER]domain1.adminServer>list-jms-quotas -server adminServer
Quota Information
=====
+-----+-----+-----+
|      Quota Name      | Max Bytes | Shared |
+-----+-----+-----+
| testQuota            | 128M      | false  |
+-----+-----+-----+
=====

[[MASTER]domain1.adminServer>list-jms-quotas -server adminServer -name testQuota
Detailed Quota Information
=====
testQuota

+-----+-----+
| Quota Name      | testQuota |
| Max Bytes       | 128M      |
| Max Messages    | 1M        |
| Shared          | false     |
+-----+-----+
=====

```

#### 4.2.10.23. list-jms-service-configs

JMS 엔진에 등록되어 있는 모든 서비스의 목록을 조회하거나 지정한 서비스의 정보를 출력한다.

- alias

list-service-config, jmsservice

- 사용법

```

list-jms-service-configs -server <server-name>
                        [-name <service-name>]

```

- 파라미터

파라미터	설명
-server <server-name>	서비스가 속한 서버를 지정한다.
[-name <service-name>]	정보를 출력할 서비스의 이름이다. 지정하지 않으면 모든 서비스에 대한 간략한 정보가 출력된다.

- 예제

```

[[MASTER]domain1.adminServer>list-jms-service-configs -server adminServer
Service Config Information
=====

```

```

=====
+-----+-----+-----+
| Service Config Name | Listener Name | Virtual Listener |
+-----+-----+-----+
| default             | jms           |                  |
| testService         | base          |                  |
+-----+-----+-----+
=====

[MASTER]domain1.adminServer>list-jms-service-configs -server adminServer -name testService
Detailed Service Config Information
=====
testService

+-----+-----+
| Service Config Name | testService |
| Client Limit       | 1000        |
| Client Keepalive Timeout | 30          |
| Listener Name      | base        |
+-----+-----+
=====

```

#### 4.2.10.24. modify-jms-connection-factory

지정된 Connection Factory의 정보를 수정한다.

- alias

modify-connection-factory, modifyconf

- 사용법

```

modify-jms-connection-factory -server <server-name>
                               -name <factory-name>
                               [-type <factory-type>]
                               [-service <service-name>]
                               [-export <export-name>]
                               [-clientid <client-id>]
                               [-policy <server-selection-policy>]
                               [-time <request-blocking-time>]
                               [-re <reconnect-enabled>]
                               [-period <reconnect-period>]
                               [-interval <reconnect-interval>]

```

- 파라미터

파라미터	설명
-server <server-name>	대상이 되는 서버를 입력한다.
-name <factory-name>	수정할 Connection Factory의 이름을 입력한다.

파라미터	설명
[-type <factory-type>]	<p>Connection Factory의 type을 입력한다.</p> <p>다음 중에 하나를 설정한다.</p> <ul style="list-style-type: none"> <li>• nonxa</li> <li>• xa</li> <li>• queue</li> <li>• topic</li> <li>• xaqueue</li> <li>• xatopic</li> </ul>
[-service <service-name>]	Connection Factory의 서비스를 입력한다.
[-export <export-name>]	<p>Connection Factory의 JNDI 이름을 입력한다.</p> <p>이 이름은 JEUS JNDI 서버 내에서 다른 객체와 중복되지 않도록 주의한다.</p>
[-clientid <client-id>]	Connection Factory로부터 생성한 커넥션에 부여할 클라이언트 ID를 입력한다.
[-policy <server-selection-policy>]	<p>커넥션을 생성할 때 채널 선택을 결정하는 정책을 입력한다.</p> <p>다음 중에 하나를 설정한다.</p> <ul style="list-style-type: none"> <li>• Round-robin</li> <li>• Random</li> </ul>
[-time <request-blocking-time>]	클라이언트에서 응답을 기다리는 시간을 입력한다.
[-re <reconnect-enabled>]	연결이 끊어질 때 재연결 여부를 입력한다.
[-period <reconnect-period>]	재연결을 시도할 최대 시간을 입력한다.
[-interval <reconnect-interval>]	재연결을 시도할 때 시간 간격을 입력한다.

#### • 예제

```
[MASTER]domain1.adminServer>modify-jms-connection-factory -server adminServer -name
MyConnectionFactory -type xa -service testService -export MyCF -clientid MyId -policy Random
Executed successfully, but some configurations were not applied dynamically. It might be
necessary to restart the server.
View the results using 'list-jms-connection-factories -server adminServer -name
MyConnectionFactory'.
[MASTER]domain1.adminServer>list-jms-connection-factories -server adminServer -name
MyConnectionFactory
Detailed Connection Factory Information
=====
MyConnectionFactory

+-----+-----+
| Factory Name | MyConnectionFactory |
```

Export Name	MyCF	
Type	xa	
Client ID	MyId	
Max Client Session Threads		100
Clustered	false	
Server Selection Policy	Random	
+-----+-----+		

```
=====
+-----+
|                                     |
|                               Addresses                               |
+-----+
| 192.168.13.13:9736(JMSServiceChannel-testService)                  |
+-----+
=====
```

#### 4.2.10.25. modify-jms-destination

지정된 Destination의 정보를 수정한다.

- alias

modify-destination, modifydest

- 사용법

```
modify-jms-destination -cluster <cluster-name> | -server <server-name>
                        -name <destination-name>
                        [-type <destination-type>]
                        [-export <export-name>]
                        [-subLimit <subscription-limit>]
                        [-quota <quota>]
                        [-pendLimit <max-pending-limit>]
                        [-dispatch <resume-dispatch-factor>]
                        [-sort <message-sort>]
                        [-deadLetter <dead-letter-destination>]
                        [-policy <expiration-policy>]
                        [-delay <redelivery-delay>]
```

- 파라미터

파라미터	설명
-cluster <cluster-name>   -server <server-name>	수정할 Destination이 속한 클러스터 또는 서버를 입력한다.
-name <destination-name>	수정할 Destination의 이름을 입력한다.
[-type <destination-type>]	Destination의 type을 입력한다. (queue 또는 topic)

파라미터	설명
[-export <export-name>]	Destination의 JNDI 이름을 입력한다.  이 이름은 JEUS JNDI 서버 내에서 다른 객체와 중복되지 않도록 주의한다.
[-subLimit <subscription-limit>]	Destination에 접근가능한 Consumer의 최대값을 입력한다.
[-quota <quota>]	Destination의 메시지 데이터 제한을 위한 Quota를 입력한다.
[-pendLimit <max-pending-limit>]	ACKNOWLEDGE를 수신하지 않은 메시지의 최대 허용 범위를 입력한다.
[-disptach <resume-dispatch-factor>]	Pending 메시지가 다시 Dispatch 될 수 있는 인수를 입력한다.
[-sort <message-sort>]	Destination에 적용 될 Message Sort를 입력한다.
[-deadLetter <dead-letter-destination>]	처리 되지 못한 메시지들을 보관 할 Destination의 이름을 입력한다.
[-policy <expiration-policy>]	만료된 메시지들을 처리할 방법을 입력한다.  다음 중에 하나를 설정한다. <ul style="list-style-type: none"> <li>• Delete</li> <li>• Redirect</li> </ul>
[-delay <redelivery-delay>]	메시지가 재전송 되기까지의 지연시간을 입력한다.

#### • 예제

```
[MASTER]domain1.adminServer>modify-jms-destination -server adminServer
-name MyQueue -type topic -export MyTopic -deadLetter ExamplesTopic
Executed successfully, but some configurations were not applied dynamically. It might be
necessary to restart the server.
View the results using 'list-jms-destinations -server adminServer -name MyQueue'.
[MASTER]domain1.adminServer>list-jms-destinations -server adminServer -name MyQueue
Detailed destination information in Server adminServer
=====
MyQueue

+-----+-----+
| Export Name           | MyTopic |
| Type                  | Topic   |
| Dead Letter Destination | ExamplesTopic |
| Consumer Count        | 0       |
| Processed Messages     | 0       |
| Remaining Messages (current) | 0       |
| Remaining Messages (high mark) | 0       |
| Pending Messages      | -       |
| Dispatched Messages   | -       |
| Delivered Messages    | 0       |
| Expired Messages      | 0       |
| Moved Messages       | -       |
| Removed Messages      | -       |
| Poisoned Messages     | 0       |
```



Memory Usage (current)	0kb	
Memory Usage (high mark)	0kb	
Production Suspended	false	
Consumption Suspended	false	
+-----+-----+		
=====		

#### 4.2.10.26. modify-jms-durable-subscription

지정된 Durable Subscription의 정보를 수정한다.

- alias

modify-durable-subscription, modify-durable, modifydur

- 사용법

```
modify-jms-durable-subscription -cluster <cluster-name> | -server <server-name>
                                -name <durable-subscription-name>
                                [-clientid <client-id>]
                                [-topic <topic-name>]
                                [-shared <shared>]
                                [-selector <message-selector>]
                                [-sort <message-sort>]
```

- 파라미터

파라미터	설명
-cluster <cluster-name>   -server <server-name>	수정할 Durable Subscription이 속한 클러스터 또는 서버를 입력한다.
-name <durable-subscription-name>	수정할 Durable Subscription의 이름을 입력한다.
[-clientid <client-id>]	subscribe 하게 될 client의 id를 지정한다.
[-topic <topic-name>]	Durable Subscription에 대한 Topic의 이름을 입력한다.
[-shared <shared>]	Durable Subscription의 Shared 여부를 입력한다.
[-selector <message-selector>]	Durable Subscription에 대한 Message Selector의 이름을 입력한다.
[-sort <message-sort>]	Durable Subscription에 대한 Message Sort의 이름을 입력한다.

- 예제

```
[MASTER]domain1.adminServer>modify-jms-durable-subscription -server adminServer
-name testDS -clientid testClientID -shared false -selector TESTNO>=1
Executed successfully, but some configurations were not applied dynamically. It might be
necessary to restart the server.
```

```
View the results using 'list-jms-durable-subscriptions -server adminServer'.
[MASTER]domain1.adminServer>list-jms-durable-subscriptions -server adminServer
Durable Subscription Information
```

```
=====
+-----+-----+-----+-----+-----+-----+
| Durable | Client | Shared | Message | Remaining Messages | Suspended |
| Name   | ID     |        | Selector | (current)          | (current) |
+-----+-----+-----+-----+-----+-----+
| testDS | testCli | false  | TESTNO>=1 | 0                  | false     |
|         | entID  |        |           |                    |           |
+-----+-----+-----+-----+-----+-----+
=====
```

#### 4.2.10.27. modify-jms-message-sort

지정된 Message Sort의 정보를 수정한다.

- alias

modify-message-sort, modmsgsort

- 사용법

```
modify-jms-message-sort -server <server-name>
                        -name <message-sort-name>
                        [-key <key>]
                        [-type <type>]
                        [-direction <direction>]
```

- 파라미터

파라미터	설명
-server <server-name>	수정할 Message Sort가 속한 서버를 지정한다.
-name <message-sort-name>	수정할 Message Sort의 이름을 입력한다.
[-key <key>]	메시지 정렬의 기준이 되는 Key를 입력한다.
[-type <type>]	Key의 타입을 입력한다.
[-direction <direction>]	메시지 정렬 순서를 입력한다.

- 예제

```
[MASTER]domain1.adminServer>modify-jms-message-sort -server adminServer
-name testSort -key JMSMessageID -type String -direction Descending
Executed successfully, but some configurations were not applied dynamically. It might be
necessary to restart the server.
View the results using 'list-jms-message-sorts -server adminServer -name testSort'.
[MASTER]domain1.adminServer>list-jms-message-sorts -server adminServer -name testSort
Detailed Message Sort Information
```

```

=====
testSort

+-----+-----+
| Message Sort Name | testSort |
| Key               | JMSMessageID |
| Type              | STRING      |
| Direction         | DESCENDING  |
+-----+-----+
=====

```

#### 4.2.10.28. modify-jms-quota

지정된 Quota의 정보를 수정한다.

- alias

modify-quota, modquota

- 사용법

```

modify-jms-quota -server <server-name>
                  -name <quota-name>
                  [-byte <max-bytes>]
                  [-msg <max-messages>]
                  [-shared <shared>]

```

- 파라미터

파라미터	설명
-server <server-name>	수정할 Quota가 속한 서버를 지정한다.
-name <quota-name>	수정할 Quota의 이름을 입력한다.
[-byte <max-bytes>]	제한 할 최대 Byte를 입력한다 .
[-msg <max-messages>]	제한 할 최대 메시지 수를 입력한다.
[-shared <shared>]	Quota를 여러 Destination에서 공유하여 사용할지 여부를 입력한다.

- 예제

```

[MASTER]domain1.adminServer>modify-jms-quota -server adminServer
-name testQuota -byte 256M -msg 2M -shared true
Executed successfully, but some configurations were not applied dynamically. It might be
necessary to restart the server.
View the results using 'list-jms-quotas -server adminServer -name testQuota'.
[MASTER]domain1.adminServer>list-jms-quotas -server adminServer -name testQuota
Detailed Quota Information
=====
testQuota

```

```

+-----+-----+
| Quota Name          | testQuota |
| Max Bytes           | 256M      |
| Max Messages        | 2M        |
| Shared              | true      |
+-----+-----+
=====

```

#### 4.2.10.29. modify-jms-service-config

지정된 서비스의 정보를 수정한다.

- alias

modify-service-config, modifyjmsservice

- 사용법

```

modify-jms-service-config -server <server-name>
                        -name <service-name>
                        [-limit <client-limit>]
                        [-timeout <client-keepalive-timeout>]
                        [-listener <listener>]
                        [-virtual]
                        [-address <address>]
                        [-port <port>]

```

- 파라미터

파라미터	설명
-server <server-name>	수정할 서비스가 속한 서버를 지정한다.
-name <service-name>	수정할 서비스의 이름을 입력한다.
[-limit <client-limit>]	허용 되는 client의 최대값을 입력한다 .
[-timeout <client-keepalive-timeout>]	비정상 종료 될 때 재연결을 기다릴 시간을 입력한다.
[-listener <listener>]	서비스의 Listener를 입력한다 .
[-virtual]	Virtual Listener를 선택할 경우 사용하는 옵션이다 .
[-address <address>]	Virtual Listener의 address를 입력한다 .
[-port <port>]	Virtual Listener의 port를 입력한다.

- 예제

```

[MASTER]domain1.adminServer>modify-jms-service-config -server adminServer

```

```
-name testService -limit 2000 -timeout 100
```

Executed successfully, but some configurations were not applied dynamically. It might be necessary to restart the server.

View the results using 'list-jms-service-configs -server adminServer -name testService'.

```
[MASTER]domain1.adminServer>list-jms-service-configs -server adminServer -name testService
Detailed Service Config Information
```

```
=====
testService

+-----+-----+
| Service Config Name | testService |
| Client Limit        |      2000  |
| Client Keepalive Timeout |      100  |
| Listener Name       | base       |
+-----+-----+
=====
```

#### 4.2.10.30. move-jms-messages

지정된 메시지를 클러스터나 서버 내의 다른 Destination으로 이동한다.

- alias

mvmsg

- 사용법

```
move-jms-message -cluster <cluster-name> | -server <server-name>
                  -dest <destination-name> | -durable <client-id>_<durable-name>
                  -id <message-id> | -all
                  -target <destination-name>
```

- 파라미터

파라미터	설명
-cluster <cluster-name>   -server <server-name>	Destination이나 Durable subscriber가 속한 클러스터나 서버를 지정한다.
-dest <destination-name>   -durable <client-id>_<durable-name>	정보를 출력할 Destination이나 Durable subscriber의 이름이다.
-id <message-id>   -all	이동할 메시지를 지정한다. 개별 message id를 지정하거나 전체를 지정할 수 있다.
-target <destination-name>	이동할 메시지가 들어갈 대상 Destination을 지정한다.

- 예제

```
[MASTER]domain1.adminServer>list-jms-destinations -server adminServer
```

Destination information in Server adminServer

```
=====
```

Name	Export Name	Type	Remaining Messages	Dead Letter Destination	Produce Suspended	Consume Suspended
MyQueue	MyQueue	Queue	0	JEUSMQ_DLQ	false	false
ExamplesTopic	ExamplesTopic	Topic	0	JEUSMQ_DLQ	false	false
ExamplesQueue	ExamplesQueue	Queue	5	JEUSMQ_DLQ	false	false
JEUSMQ_DLQ	JEUSMQ_DLQ	Queue	0	JEUSMQ_DLQ	false	false

```
=====
```

For detailed information, use the -name option

```
=====
```

```
[MASTER]domain1.adminServer>move-jms-messages -server adminServer -dest ExamplesQueue -all -target MyQueue
```

Successfully moved all messages from ExamplesQueue to MyQueue is successful

```
[MASTER]domain1.adminServer>list-jms-destinations -server adminServer
```

Destination information in Server adminServer

```
=====
```

Name	Export Name	Type	Remaining Messages	Dead Letter Destination	Produce Suspended	Consume Suspended
MyQueue	MyQueue	Queue	5	JEUSMQ_DLQ	false	false
ExamplesTopic	ExamplesTopic	Topic	0	JEUSMQ_DLQ	false	false
ExamplesQueue	ExamplesQueue	Queue	0	JEUSMQ_DLQ	false	false
JEUSMQ_DLQ	JEUSMQ_DLQ	Queue	0	JEUSMQ_DLQ	false	false

```
=====
```

For detailed information, use the -name option

```
=====
```

#### 4.2.10.31. remove-jms-connection-factory

동적으로 JMS 엔진에서 특정 Connection Factory를 제거한다.

- alias

remove-connection-factory, deleteconf, removeconf

- 사용법

```
remove-jms-connection-factory -server <server-name>
                             -name <factory-name>
```

- 파라미터

파라미터	설명
-server <server-name>	Connection Factory가 속한 서버를 지정한다.
-name <factory-name>	제거할 Connection Factory의 이름이다.

- 예제

```
[MASTER]domain1.adminServer>list-jms-connection-factories -server adminServer
Connection Factory information
=====
+-----+-----+-----+
|          Factory Name          |          Export Name          |   Type   |
+-----+-----+-----+
| ConnectionFactory              | ConnectionFactory              | nonxa    |
| MyConnectionFactory           | MyConnectionFactory           | nonxa    |
| XAConnectionFactory           | XAConnectionFactory           | xa       |
+-----+-----+-----+
=====

[MASTER]domain1.adminServer>remove-jms-connection-factory -server adminServer -name
MyConnectionFactory
Executed Successfully.
View the results using 'list-jms-connection-factories -server adminServer'.
[MASTER]domain1.adminServer>list-jms-connection-factories -server adminServer
Connection Factory information
=====
+-----+-----+-----+
|          Factory Name          |          Export Name          |   Type   |
+-----+-----+-----+
| ConnectionFactory              | ConnectionFactory              | nonxa    |
| XAConnectionFactory           | XAConnectionFactory           | xa       |
+-----+-----+-----+
=====
```

#### 4.2.10.32. remove-jms-destination

동적으로 JMS 엔진에서 특정 Destination을 제거한다.

- alias

remove-destination, deletedest, removedest

- 사용법

```
remove-jms-destination -cluster <cluster-name> | -server <server-name>
```

-name <destination-name>

- 파라미터

파라미터	설명
-cluster <cluster-name>   -server <server-name>	Destination이 속한 클러스터 또는 서버를 지정한다.
-name <destination-name>	제거할 Destination의 이름이다.

- 예제

```
[MASTER]domain1.adminServer>list-jms-destinations -server adminServer
Destination information in Server adminServer
```

```
=====
```

Name	Export Name	Type	Remaining Messages	Dead Letter Destination	Produce Suspended	Consume Suspended
MyQueue	MyQueue	Queue	0	JEUSMQ_DLQ	false	false
Examples Topic	Examples Topic	Topic	0	JEUSMQ_DLQ	false	false
Examples Queue	Examples Queue	Queue	0	JEUSMQ_DLQ	false	false
JEUSMQ_D LQ	JEUSMQ_D LQ	Queue	0	JEUSMQ_DLQ	false	false

```
=====
```

For detailed information, use the -name option

```
[MASTER]domain1.adminServer>remove-jms-destination -server adminServer -name MyQueue
Executed Successfully.
View the results using 'list-jms-destinations -server adminServer'.
[MASTER]domain1.adminServer>list-jms-destinations -server adminServer
Destination information in Server adminServer
```

```
=====
```

Name	Export Name	Type	Remaining Messages	Dead Letter Destination	Produce Suspended	Consume Suspended
Examples Topic	Examples Topic	Topic	0	JEUSMQ_DLQ	false	false
Examples Queue	Examples Queue	Queue	0	JEUSMQ_DLQ	false	false
JEUSMQ_D LQ	JEUSMQ_D LQ	Queue	0	JEUSMQ_DLQ	false	false

```
=====
```

For detailed information, use the -name option



=====

#### 4.2.10.33. remove-jms-durable-subscription

동적으로 JMS 엔진에서 특정 Durable Subscription을 제거한다.

- alias

remove-durable, deletedur, removedur

- 사용법

```
remove-jms-durable-subscription -cluster <cluster-name> | -server <server-name>
                                -name <durable-subscription-name>
```

- 파라미터

파라미터	설명
-cluster <cluster-name>   -server <server-name>	Durable Subscription이 속한 클러스터 또는 서버를 지정한다.
-name <durable-subscription-name>	제거할 Durable Subscription의 이름이다.

- 예제

```
[MASTER]domain1.adminServer>list-jms-durable-subscriptions -server adminServer
Durable Subscription Information
=====
+-----+-----+-----+-----+-----+
| Durable Name | Client ID | Shared | Message Selector | Remaining Messages |
+-----+-----+-----+-----+-----+
| testDurable1 | client1   | false  |                   | 5                  |
| testDurable2 | client2   | false  |                   | 0                  |
+-----+-----+-----+-----+-----+
=====

[MASTER]domain1.adminServer>remove-jms-durable-subscription -server adminServer -name
testDurable2
Executed Successfully.
View the results using 'list-jms-durable-subscriptions -server adminServer'.
[MASTER]domain1.adminServer>list-jms-durable-subscriptions -server adminServer
Durable Subscription Information
=====
+-----+-----+-----+-----+-----+
| Durable Name | Client ID | Shared | Message Selector | Remaining Messages |
+-----+-----+-----+-----+-----+
| testDurable1 | client1   | false  |                   | 5                  |
+-----+-----+-----+-----+-----+
=====
```

#### 4.2.10.34. remove-jms-message-sort

JMS 엔진에서 특정 Message Sort를 제거한다.

- alias

remove-message-sort, removemsgsort

- 사용법

```
remove-jms-message-sort -server <server-name>
                        -name <message-sort-name>
```

- 파라미터

파라미터	설명
-server <server-name>	Message Sort가 속한 서버를 지정한다.
-name <message-sort-name>	제거할 Message Sort의 이름이다.

- 예제

```
[MASTER]domain1.adminServer>list-jms-message-sorts -server adminServer
Message Sort Information
=====
+-----+-----+-----+
|      Message Sort Name      |      Key      |      Type      |
+-----+-----+-----+
| testSort                    | JMSMessageID  | STRING         |
| testSort2                   | JMSPriority    | INTEGER        |
+-----+-----+-----+
=====

[MASTER]domain1.adminServer>remove-jms-message-sort -server adminServer -name testSort2
Executed successfully, but some configurations were not applied dynamically. It might be
necessary to restart the server.
View the results using 'list-jms-message-sorts -server adminServer'.
[MASTER]domain1.adminServer>list-jms-message-sorts -server adminServer
Message Sort Information
=====
+-----+-----+-----+
|      Message Sort Name      |      Key      |      Type      |
+-----+-----+-----+
| testSort                    | JMSMessageID  | STRING         |
+-----+-----+-----+
=====
```

#### 4.2.10.35. remove-jms-quota

JMS 엔진에서 특정 Quota를 제거한다.

- alias

remove-quota, removequota

- 사용법

```
remove-jms-quota -server <server-name>
                  -name <quota-name>
```

- 파라미터

파라미터	설명
-server <server-name>	Quota가 속한 서버를 지정한다.
-name <quota-name>	제거할 Quota의 이름이다.

- 예제

```
[MASTER]domain1.adminServer>list-jms-quotas -server adminServer
Quota Information
=====
+-----+-----+-----+
|      Quota Name      | Max Bytes | Shared |
+-----+-----+-----+
| testQuota            | 256M      | true   |
| testQuota2           | 128M      | false  |
+-----+-----+-----+
=====

[MASTER]domain1.adminServer>remove-jms-quota -server adminServer -name testQuota2
Executed successfully, but some configurations were not applied dynamically. It might be
necessary to restart the server.
View the results using 'list-jms-quotas -server adminServer'.
[MASTER]domain1.adminServer>list-jms-quotas -server adminServer
Quota Information
=====
+-----+-----+-----+
|      Quota Name      | Max Bytes | Shared |
+-----+-----+-----+
| testQuota            | 256M      | true   |
+-----+-----+-----+
=====
```

#### 4.2.10.36. remove-jms-service-config

JMS 엔진에서 특정 Service Config를 제거한다.

- alias

remove-service-config, deletejmservice, removejmservice

- 사용법

```
remove-jms-service-config -server <server-name>
                           -name <service-name>
```

- 파라미터

파라미터	설명
-server <server-name>	서비스가 속한 서버를 지정한다.
-name <service-name>	제거할 서비스의 이름이다.

- 예제

```
[MASTER]domain1.adminServer>list-jms-service-configs -server adminServer
Service Config Information
=====
+-----+-----+-----+
| Service Config Name | Listener Name | Virtual Listener |
+-----+-----+-----+
| default             | jms           |                  |
| testService         | base         |                  |
+-----+-----+-----+
=====

[MASTER]domain1.adminServer>remove-jms-service-config -server adminServer -name testService
Executed successfully, but some configurations were not applied dynamically. It might be
necessary to restart the server.
View the results using 'list-jms-service-configs -server adminServer'.
[MASTER]domain1.adminServer>list-jms-service-configs -server adminServer
Service Config Information
=====
+-----+-----+-----+
| Service Config Name | Listener Name | Virtual Listener |
+-----+-----+-----+
| default             | jms           |                  |
+-----+-----+-----+
=====
```

#### 4.2.10.37. set-jms-engine-quota

JMS 엔진의 Quota를 설정한다.

- alias

set-engine-quota

- 사용법

```
set-jms-engine-quota -server <server-name>
```

```
[ -byte <max-bytes>]
[ -msg <max-messages>]
```

- 파라미터

파라미터	설명
-server <server-name>	설정할 JMS 엔진이 속한 서버를 지정한다.
[ -byte <max-bytes>]	JMS 엔진에서 제한 할 최대 Byte를 입력한다 .
[ -msg <max-messages>]	JMS 엔진에서 제한 할 최대 메시지 수를 입력한다.

- 예제

```
[MASTER]domain1.adminServer>set-jms-engine-quota -server adminServer -byte 256M -msg 1M
Executed successfully, but some configurations were not applied dynamically. It might be
necessary to restart the server.
View the results using 'set-jms-engine-quota -server adminServer'.
[MASTER]domain1.adminServer>set-jms-engine-quota -server adminServer
Engine Quota Information
=====
JMS Engine

+-----+-----+
| Max Bytes           | 256M |
| Max Messages        | 1M   |
+-----+-----+
=====
```

#### 4.2.10.38. set-jms-failover-check

JMS 엔진의 Failover 전 장애 감지를 위한 옵션을 설정한다.

- alias

set-jms-failover, set-failover

- 사용법

```
set-jms-failover-check -server <server-name>
                        [-timeout <timeout>]
                        [-count <count>]
```

- 파라미터

파라미터	설명
-server <server-name>	설정할 JMS 엔진이 속한 서버를 지정한다.

파라미터	설명
[-timeout <timeout>]	Failover 전 JMS 엔진의 생존 여부를 확인 할 시간을 입력한다.
[-count <count>]	Failover 전 JMS 엔진의 생존 여부를 확인 할 최대 횟수를 입력한다.

- 예제

```
[MASTER]domain1.adminServer>set-jms-failover-check -server adminServer -timeout 10 -count 5
Executed successfully, but some configurations were not applied dynamically. It might be
necessary to restart the server.
View the results using 'set-jms-failover-check -server adminServer'.
[MASTER]domain1.adminServer>set-jms-failover-check -server adminServer
Failover check Information
=====
JMS Engine

+-----+-----+-----+
| Failover Check Timeout |      10 |
| Failover Check Count   |      5 |
+-----+-----+-----+
=====
```

#### 4.2.10.39. set-jms-persistence-store

JMS 엔진의 Persistent Object들을 위한 Store 정보를 설정한다.

- alias

set-jms-store, set-persistence-store, setjmsstore

- 사용법

```
set-jms-persistence-store -server <server-name>
                          -none | -journal | -jdbc
                          [-dir <base-dir>]
                          [-init <initial-log-file-count>]
                          [-max <max-log-file-count>]
                          [-size <log-file-size>]
                          [-dest <destination-table>]
                          [-durSub <durable-subscription-table>]
                          [-property <property>]
                          [-source <data-source>]
                          [-msg <message-table>]
                          [-subMsg <subscription-message-table>]
                          [-tx <transaction-table>]
```

- 파라미터

파라미터	설명
-server <server-name>	설정할 JMS 엔진이 속한 서버를 지정한다.
-none   -journal   -jdbc	Store의 타입을 선택하는 옵션이다. 세 가지 중 하나를 선택해야 한다.
[-dir <base-dir>]	Journal Store를 설정할 디렉토리를 입력한다 .
[-init <initial-log-file-count>]	Journal Store를 생성할 때 초기에 생성할 로그 파일의 개수를 입력한다.
[-max <max-log-file-count>]	Journal Store의 최대 로그 파일의 개수를 입력한다.
[-size <log-file-size>]	Journal Store의 로그 파일의 크기를 입력한다.
[-dest <destination-table>]	Journal과 JDBC Store의 Destination 테이블 이름을 입력한다.
[-durSub <durable-subscription-table>]	Journal과 JDBC Store의 Durable Subscription 테이블 이름을 입력한다.
[-property <property>]	Journal Store의 추가적인 설정을 입력한다.
[-source <data-source>]	JDBC Store의 Data Source를 입력한다.
[-msg <message-table>]	JDBC Store의 Message 테이블 이름을 입력한다.
[-subMsb <subscription-message-table>]	JDBC Store의 Subscription 테이블 이름을 입력한다.
[-tx <transaction-table>]	JDBC Store의 Transaction 테이블 이름을 입력한다.

- 예제

```
[MASTER]domain1.adminServer>set-jms-persistence-store -server adminServer
-journal -dir store -init 5 -max 50 -size 256M -dest ADMIN_DEST -durSub ADMIN_DS
Executed successfully, but some configurations were not applied dynamically. It might be
necessary to restart the server.
```

#### 4.2.10.40. view-jms-message

Destination 안의 특정 메시지의 상세한 정보를 출력한다.

- alias

viewmsg

- 사용법

```
view-jms-messages -cluster <cluster-name> | -server <server-name>
                  -dest <destination-name> | -durable <client-id>_<durable-name>
                  -id <message-id>
```

- 파라미터

파라미터	설명
-cluster <cluster-name>   -server <server-name>	Destination이나 Durable subscriber가 속한 클러스터나 서버를 지정한다.
-dest <destination-name>   -durable <client-id>_<durable-name>	정보를 출력할 Destination이나 Durable subscriber의 이름이다.
-id <message-id>	출력할 메시지의 message id를 지정한다.

- 예제

```
[MASTER]domain1.adminServer>list-jms-messages -server adminServer -dest ExamplesQueue
Messages in Destination ExamplesQueue
```

```
=====
+-----+-----+-----+
| Message ID | Message Type | Created Time |
+-----+-----+-----+
| ID:1242976880:1:0 | Text | Tue Nov 19 15:53:17 KST 2016 |
| ID:1242976880:1:1 | Text | Tue Nov 19 15:53:17 KST 2016 |
| ID:1242976880:1:2 | Text | Tue Nov 19 15:53:17 KST 2016 |
| ID:1242976880:1:3 | Text | Tue Nov 19 15:53:17 KST 2016 |
| ID:1242976880:1:4 | Text | Tue Nov 19 15:53:17 KST 2016 |
+-----+-----+-----+
=====
```

```
[MASTER]domain1.adminServer>view-jms-message -server adminServer
-dest ExamplesQueue -id ID:1242976880:1:0 Message ID:1242976880:1:0
```

```
=====
+-----+-----+
| Attribute name | Value |
+-----+-----+
| Message ID | ID:1242976880:1:0 |
| Destination Name | ExamplesQueue |
| Delivery Mode | 2 |
| Message Type | Text |
| Correlation ID | |
| Created Time | Tue Nov 19 15:53:17 KST 2016 |
| Expiration Time | - |
| Time-to-live | 0 |
| Priority | 4 |
| Redelivered | false |
| Delivery Time | Tue Nov 19 15:53:17 KST 2016 |
| Redelivery Limit | 4 |
| Reply To | |
| Message Body | Test Message |
+-----+-----+
=====
```

## 4.2.11. 데이터소스 관련 명령어

데이터소스 관리 및 설정 변경, 설정 확인 등의 기능을 제공한다. 명령어들의 목록은 다음과 같다.



명령어	설명
<code>add-cluster-data-source</code>	클러스터 데이터소스를 도메인에 동적으로 추가한다.
<code>add-data-source</code>	데이터소스를 도메인에 동적으로 추가한다.
<code>list-cluster-data-sources</code>	도메인에 존재하는 모든 클러스터 데이터소스의 목록을 조회한다. 특정 데이터소스의 ID를 명시할 경우 해당 클러스터 데이터소스의 설정을 자세히 조회한다.
<code>list-data-sources</code>	도메인에 존재하는 모든 데이터소스의 목록을 조회한다. 특정 데이터소스의 ID를 명시할 경우 해당 데이터소스의 설정을 자세히 조회한다.
<code>modify-cluster-data-source</code>	클러스터 데이터소스의 설정을 변경한다. 대부분의 설정이 동적으로 변경 가능하다.
<code>modify-data-source</code>	데이터소스의 설정을 변경한다. 일부 설정은 동적으로 변경 가능하다.
<code>remove-cluster-data-source</code>	클러스터 데이터소스를 도메인으로부터 동적으로 삭제한다.
<code>remove-data-source</code>	데이터소스를 도메인으로부터 동적으로 삭제한다.
<code>test-data-source-config</code>	데이터소스의 설정이 올바른지 확인한다.

#### 4.2.11.1. add-cluster-data-source

클러스터 데이터소스를 도메인에 동적으로 추가한다.

- alias

`addcds`

- 사용법

```
add-cluster-data-source [-id,--dataSourceID <data-source-id>]
                        [-en,--exportName <export-name>]
                        [-dss,--dataSourceSelector <data-source-selector>]
                        [-lb,--loadBalance <true | false>]
                        [-ipc,--isPreConn <is-pre-conn>]
                        [-ufb,--useFailback <use-failback>]
                        [-cds,--componentDataSources <component-data-sources>]
                        [-oc,--onsConfig <ons-config>]
                        [-dsa,--dataSourceAffinity <data-source-affinity>]
                        [-f,--forceLock]
                        [-detail]
```

- 파라미터

파라미터	설명
<code>[-id,--dataSourceID &lt;data-source-id&gt;]</code>	클러스터 데이터소스의 ID이다. 하나의 도메인에서 클러스터 데이터소스 ID는 클러스터 데이터소스의 유일한 식별자로서 동작하도록 설정해야 한다.

파라미터	설명
<code>[-en,--exportName &lt;export-name&gt;]</code>	<p>클러스터 데이터소스의 JNDI 이름을 설정한다. 서로 다른 두 데이터소스가 서로 다른 서버에 JNDI 바인드되는 것을 보장할 수 있으면 해당 데이터소스들은 서로 같은 JNDI 이름을 가질 수 있다. 이는 임의의 서버에서 동일한 JNDI 이름을 가지는 서로 다른 데이터소스를 허용하지 않음을 의미한다.</p> <p>설정되지 않으면 클러스터 데이터소스 ID를 JNDI 이름으로 사용한다.</p> <p>콤마(,)를 구분자로 사용하여 여러 개 입력할 수 있으며, 이 경우 구분자로 구분 된 어떠한 값을 사용하여도 JNDI lookup이 가능하다.</p>
<code>[-dss,--dataSourceSelector &lt;data-source-selector&gt;]</code>	<p>클러스터 데이터소스로부터 커넥션을 얻을 때 사용자나 개발자가 특정 컴포넌트 데이터소스 선택에 대한 정책을 직접 정의할 수 있다. <code>jeus.jdbc.helper.DataSourceSelector</code> 추상 클래스를 상속하여 구현하고 그 구현 클래스의 패키지 이름을 포함하는 이름을 적어준다. 이것을 설정하면 <code>--loadBalance</code> 설정은 아무런 기능을 하지 않으며 <code>failover</code>와 <code>failback</code>을 무조건 시도한다.</p> <ul style="list-style-type: none"> <li>• <code>failover</code>는 이 설정을 통하여 최초 선택된 컴포넌트 데이터소스 다음 인덱스의 컴포넌트 데이터소스를 시작으로 하여 한 번만 라운딩하는 방식으로 이루어진다.</li> <li>• <code>failback</code>은 Use Failback을 설정하는 경우 Failback이 수행되는 방식과 유사하게 이루어진다. 정책을 정의할 때는 대체로 동기화를 고려해야 하며 이는 구현자의 몫이다.</li> </ul>
<code>[-lb,--loadBalance &lt;true   false&gt;]</code>	<p>load balancing 여부를 설정한다.</p> <ul style="list-style-type: none"> <li>• <code>true</code> : <code>--useFailback</code> 설정은 아무런 기능을 하지 않게 된다.</li> </ul>
<code>[-ipc,--isPreConn &lt;is-pre-conn&gt;]</code>	<p>클러스터 데이터소스에 속한 컴포넌트 데이터소스들의 Connection Pool을 미리 생성할지 여부를 결정한다. 컴포넌트 데이터소스들의 Connection Pool을 미리 생성해 놓으면 성능상으로 이점이 있으나 리소스가 낭비되어 효율이 떨어진다.</p>
<code>[-ufb,--useFailback &lt;use-failback&gt;]</code>	<p>이전 버전의 JEUS에서는 <code>failover</code>만을 지원했으므로 이에 대한 호환성을 위해 제공하는 옵션이다.</p> <p>보조 데이터소스로 <code>failover</code>한 후에 주 데이터소스로 <code>failback</code>할 것인지의 여부를 설정한다. 기본적으로 <code>failback</code>을 시도한다.</p> <p><code>failback</code>을 위해서는 반드시 주 데이터소스에 대하여 <code>--checkQuery</code> 및 <code>--checkQueryPeriod</code>를 설정해야 한다.</p>
<code>[-cds,--componentDataSources &lt;component-data-sources&gt;]</code>	<p>클러스터 데이터소스에 속한 컴포넌트 데이터소스들의 데이터소스 ID를 명시한다.</p>

파라미터	설명
<code>[-oc,--onsConfig &lt;ons-config&gt;]</code>	<p>ONS와 결합된 클러스터 데이터소를 사용하고자 할 때 설정한다. ONS상의 각 RAC 노드들이 ONS 통신에 사용하는 IP, 포트를 적는다. 클러스터 데이터소스는 설정된 IP, 포트들에 소켓 연결을 맺어 ONS 클라이언트로 동작하게 된다.</p> <p>다음과 같은 형식으로 적는다.</p> <pre>nodes=host1:6200,host2:6200</pre>
<code>[-dsa,--dataSourceAffinity &lt;data-source-affinity&gt;]</code>	트랜잭션 사용 중 데이터소스에 대한 Affinity 설정 여부를 정한다. 이 설정이 켜지면 트랜잭션 처리가 하나의 멤버 데이터소스 인스턴스에 한정하여 이루어져 글로벌 트랜잭션은 처리 성능 향상을 도모할 수 있고, XA Emulation을 설정한 로컬 트랜잭션도 클러스터 데이터소스 사용 간 보장 될 수 있다.
<code>[-f,--forceLock]</code>	설정 변경을 강제로 진행하여 반영한다.
<code>[-detail]</code>	변경 결과에 대한 상세 정보가 출력된다.

- 예제

```
[MASTER]domain1.adminServer>add-cluster-data-source -id cds1 -cds ds1,ds2
Successfully performed the ADD operation for cluster data source [cds1] to domain.
Check the results using "add-cluster-data-source"
```

#### 4.2.11.2. add-data-source

데이터소스를 도메인에 동적으로 추가한다.

- alias

addds

- 사용법

```
add-data-source [-id,--dataSourceID <data-source-id>]
                [-user,--user <user>]
                [-pw,--password <password>]
                [-algorithm,--algorithm <algorithm>]
                [-ac,--autoCommit <auto-commit-value>]
                [-sqt,--stmtQueryTimeout <statement-query-timeout>]
                [-aocl,--actionOnConnectionLeak <action-on-connection-leak>]
                [-min,--min <minimum-size>]
                [-max,--max <maximum-size>]
                [-step,--step <resizing-step>]
                [-period,--period <resizing-period>]
                [-ew,--enableWait <true | false>]
```

```

[-wt,--waitTime <wait-time>]
[-dds,--delegationDataSource <delegation-data-source>]
[-muc,--maxUseCount <max-use-count>]
[-cq,--checkQuery <check-query>]
[-cqt,--checkQueryTimeout <check-query-timeout>]
[-nvi,--nonValidationInterval <non-validation-interval>]
[-cqp,--checkQueryPeriod <check-query-period>]
[-cqrc,--checkQueryRetrialCount <check-query-retrial-count>]
[-dpocq,--destroyPolicyOnCheckQuery <destroy-policy-on-check-query>]
[-scs,--stmtCachingSize <stmt-caching-size>]
[-sfs,--stmtFetchSize <stmt-fetch-size>]
[-ct,--connectionTrace <true | false>]
[-gct,--getConnectionTrace <get-connection-trace>]
[-act,--autoCommitTrace <auto-commit-trace>]
[-ust,--useSqlTrace <use-sql-trace>]
[-kcho,--keepConnectionHandleOpen <keep-connection-handle-open>]
[-en,--exportName <export-name>]
[-dscn,--dataSourceClassName <data-source-class-name>]
[-dst,--dataSourceType <data-source-type>]
[-vendor,--vendor <vendor>]
[-desc,--description <description>]
[-sn,--serverName <server-name>]
[-pn,--portNumber <port-number>]
[-dn,--databaseName <database-name>]
[-lt,--loginTimeout <login-timeout>]
[-il,--isolationLevel <isolation-level>]
[-pdt,--poolDestroyTimeout <pool-destroy-timeout>]
[-prop,--property <property>]
[-sxe,--supportXaEmulation <support-xa-emulation>]
[-ddba,--delegationDb <delegation-dba>]
[-dbat,--dbaTimeout <dba-timeout>]
[-cqc,--checkQueryClass <check-query-class>]
[-is,--initSql <init-sql>]
[-f,--forceLock]
[-detail]

```

## • 파라미터

파라미터	설명
[-id,--dataSourceID <data-source-id>]	데이터소스의 ID이다. 하나의 도메인에서 데이터소스 ID는 데이터소스의 유일한 식별자로서 동작하도록 설정해야 한다.
[-user,--user <user>]	데이터베이스 사용자의 ID를 설정한다. 트랜잭션 처리 등을 위해서는 충분한 권한을 가지고 있어야 한다.
[-pw,--password <password>]	데이터베이스 사용자의 패스워드를 설정한다. 패스워드를 암호화해서 저장할 때에는 '{algorithm:keysize}ciphertext' 와 같은 형식으로 쓴다.

파라미터	설명
[-algorithm,--algorithm <algorithm>]	<p><b>[동적 변경]</b> 데이터베이스 사용자 패스워드의 알고리즘을 설정한다. algorithm 옵션은 password 옵션과 함께 사용될 때에만 의미가 있다.</p> <p>다음의 알고리즘을 사용할 수 있다.</p> <ul style="list-style-type: none"> <li>• base64</li> <li>• DES</li> <li>• DESede</li> <li>• AES</li> <li>• SEED</li> <li>• Blowfish</li> </ul>
[-ac,--autoCommit <TRUE   FALSE   DRIVER> ]	<p>데이터소스의 기본 auto-commit 설정으로 다음 중에 하나를 설정한다.</p> <ul style="list-style-type: none"> <li>• TRUE</li> <li>• FALSE</li> <li>• DRIVER : DRIVER로 설정하면 JEUS는 auto-commit 설정에 관여하지 않으며 JDBC 드라이버의 auto-commit 설정을 따르게 된다. XA 에뮬레이션 기능을 사용하는 Connection Pool 데이터소스 또는 XA 데이터소스의 경우 트랜잭션이 연동되어있지 않을 때만 이 설정값이 유효하다.</li> </ul>
[-sqt,--stmtQueryTimeout <statement-query-timeout> ]	<p>데이터소스로부터 얻은 커넥션을 통해 생성된 java.sql.Statement 객체에 지정될 쿼리 타임아웃을 설정한다.</p> <p>JEUS는 JDBC API를 통해 정의되고 JDBC 드라이버 벤더에서 구현한 쿼리 타임아웃 설정 메소드 (java.sql.Statement#setQueryTimeout)를 호출할 뿐이다. JDBC 드라이버 벤더별로 쿼리 타임아웃 적용에 약간의 차이가 있다. (단위: ms)</p>
[-aocl,--actionOnConnectionLeak <action-on-connection-leak>]	<p>애플리케이션(주로 stateless 컴포넌트 - Servlet/JSP, stateless Session Bean, MDB)에서 사용하고 제대로 반환하지 않은 JDBC 커넥션을 감지했을 때 JEUS가 취하는 액션을 정의한다.</p> <p>다음의 값 중에 하나를 설정한다. 설정하지 않을 경우 서버의 설정을 따른다.</p> <ul style="list-style-type: none"> <li>• NO_ACTION : 아무런 액션을 취하지 않는다.</li> <li>• WARNING : 설정하면 반환되지 않은 JDBC 커넥션 정보를 로그로 남긴다.</li> <li>• AUTO_CLOSE : 반환되지 않은 JDBC 커넥션 정보를 로그로 남기며 해당 JDBC 커넥션을 회수한다.</li> </ul>
[-min,--min <minimum-size>]	<p>Connection Pool에 Pooling되는 커넥션의 최소값을 지정한다.</p>

파라미터	설명
<code>[-max,--max &lt;maximum-size&gt;]</code>	Connection Pool에 Pooling되는 커넥션의 최대값을 지정한다.
<code>[-step,--step &lt;resizing-step&gt;]</code>	Connection Pool에 커넥션이 부족할 때 현재 커넥션의 개수가 최대값 이하인 경우 DB로부터 커넥션을 새로 받아와 채우는 데, 이때 새로 받아오는 커넥션의 개수를 지정한다.
<code>[-period,--period &lt;resizing-period&gt;]</code>	Connection Pool의 크기를 최소값에 맞춰 조정하는 주기를 설정한다. Connection Pool의 크기가 최소값을 초과하는 경우에는 사용하지 않는 커넥션들을 닫아주고 Connection Pool의 크기가 최소값에 미치지 못하는 경우에는 DB로 붙일 커넥션을 새로 받아와 채운다. (단위: ms)
<code>[-ew,--enableWait &lt;true   false&gt;]</code>	<p>Connection Pool에 사용 가능한 커넥션이 없고 커넥션도 더 이상 늘릴 수 없을 때 커넥션 요청을 처리하는 방법을 결정한다.</p> <ul style="list-style-type: none"> <li>• <code>true</code> : 이용 가능한 커넥션을 얻기 위해 기다린다.</li> <li>• <code>false</code> : 새로운 커넥션을 만들어서 제공하지만 그 커넥션은 반환되었을 때 Pooling되지 않고 버려진다. 이를 일회용(disposable) 커넥션이라고도 한다.</li> </ul>
<code>[-wt,--waitTime &lt;wait-time&gt;]</code>	--enable-wait이 true인 경우에만 유효한 설정으로 커넥션을 얻기 위해 대기하는 한계 시간을 나타낸다. 만약 이 시간이 지나도 커넥션을 얻지 못하면 JEUS는 타임아웃 예외를 발생시킨다. (단위: ms)
<code>[-dds,--delegationDataSource &lt;delegation-data-source&gt;]</code>	<p>트랜잭션과 연동하지 않은 상태에서는 XA 데이터소스를 통해 커넥션을 얻기보다 Connection Pool 데이터소스를 통해 커넥션을 얻는 것이 낫다.</p> <p>기능상 차이도 없고, 트랜잭션 연동을 위한 기능을 포함하고 있는 XA 커넥션은 시스템에 부담을 더 주기 때문이다. 이를 위해 XA 데이터소스인 경우 이 설정을 통하여 트랜잭션과 연동하지 않은 상태에서의 커넥션 요청을 위임할 Connection Pool 데이터소스를 지정한다.</p> <p>한편 Oracle, DB2 등에서 XA 커넥션을 트랜잭션 없이도 사용한다. 트랜잭션에 연동하면서 사용하다 보면 XA를 시작할 수 없는 예외가 발생하기도 하는데 정확한 원인은 알 수 없기 때문에 이를 회피하기 위한 방편으로서도 이 설정을 이용한다.</p>
<code>[-muc,--maxUseCount &lt;max-use-count&gt;]</code>	<p>커넥션의 최대 사용 횟수이다. 이 사용 횟수 이상이 되면 새로운 커넥션으로 교체한다.</p> <p>(기본값: 0, 커넥션을 교체하지 않겠다는 의미)</p>
<code>[-cq,--checkQuery &lt;check-query&gt;]</code>	<p>커넥션 상태 점검에 사용될 쿼리(check-query)를 설정한다.</p> <p>보통 DB와의 연결 유효성만을 확인하면 되므로 간단한 select 쿼리를 사용할 것을 권장한다.</p>

파라미터	설명
<code>[-cqt,--checkQueryTimeout &lt;check-query-timeout&gt;]</code>	<p>커넥션 점검을 위해 check-query를 수행했을 때 DB가 응답이 없어 드라이버가 계속 기다리는 상황이 발생할 수 있다. 이런 경우를 피하기 위해 check-query에 대해 쿼리 타임아웃을 적용한다.</p> <p>이것은 JDBC API에서 정의한 <code>java.sql.Statement#setQueryTimeout</code> 메소드 호출을 통해 가능하다.</p> <p>1000ms보다 작을 경우 0으로 설정되므로 주의한다. (단위: ms)</p>
<code>[-nvi,--nonValidationInterval &lt;non-validation-interval&gt;]</code>	<p>커넥션 점검이 너무 잦아서 오버헤드가 발생하는 경우 설정한다.</p> <p>커넥션 점검을 수행하기 직전의 시각과 가장 최근의 커넥션 점검 시각과의 차이가 설정한 시간 간격 이내면 커넥션 점검을 생략한다. (단위: ms)</p> <p>예를 들어 이 설정값이 5000ms인 경우 어떤 커넥션의 마지막 커넥션 점검 시간으로부터 아직 5초가 지나지 않았다면 해당 커넥션에 대한 점검이 생략된 채 애플리케이션에 전달된다.</p>
<code>[-cqp,--checkQueryPeriod &lt;check-query-period&gt;]</code>	<p>Connection Pool의 커넥션들을 설정한 주기마다 체크하여 문제가 있는 커넥션을 제거한다. (단위: ms)</p> <p>클러스터 데이터소스에 속한 데이터소스는 자신의 상태 체크에 사용하므로 반드시 설정해야 한다.</p>
<code>[-cqrc,--checkQueryRetrialCount &lt;check-query-retrial-count&gt;]</code>	<p>커넥션 점검은 기본적으로 <code>--destroyPolicyOnCheckQuery</code>가 <code>FAILED_CONNECTION_ONLY</code>로 설정된 경우 한 번 수행된다.</p> <p><code>--destroyPolicyOnCheckQuery</code>가 <code>ALL_CONNECTIONS</code>으로 설정된 경우에는 최초의 커넥션 점검에서 커넥션 이상이 확인되면 또 다른 커넥션에 대해서 한 번 더 커넥션 점검이 이루어져 총 두 번의 커넥션 점검이 수행될 수 있다.</p> <p>이 설정값이 이러한 기본 커넥션 점검 수행 횟수에 더해져 최종 커넥션 점검 수행 횟수가 정해진다.</p>
<code>[-dpocq,--destroyPolicyOnCheckQuery &lt;destroy-policy-on-check-query&gt;]</code>	<p>커넥션이 유효하지 않은 것으로 확인되었을 때 Connection Pool에 있는 다른 커넥션들에 대한 처리 정책을 설정한다.</p> <ul style="list-style-type: none"> <li>• <code>FAILED_CONNECTION_ONLY</code> : 유효하지 않은 것으로 확인된 커넥션만 제거한다.</li> <li>• <code>ALL_CONNECTIONS</code> : 유효하지 않은 것으로 확인된 커넥션을 제거하고 Connection Pool에 있는 다른 커넥션의 유효성을 한 번 더 확인한다. 그조차 유효하지 않은 것으로 확인되면 Connection Pool의 모든 커넥션을 제거한다.</li> </ul>

파라미터	설명
<code>[-scs,--stmtCachingSize &lt;stmt-caching-size&gt;]</code>	<p>JDBC 드라이버는 애플리케이션에서 prepared statement를 요청할 때마다 파라미터로 넘어온 SQL 문장을 파싱한다. 이 파싱 작업이 성능에 영향을 줄 수 있기 때문에 이를 피하기 위해서 JEUS 내부적으로 prepared statement를 캐시하는 기능을 제공한다. 이 설정은 캐싱할 prepared statement의 개수를 지정한다. 이 기능을 사용하면 커넥션을 항상 열어둔 채로 사용하기 때문에 커넥션을 닫았을 때 드라이버가 수행하는 클리어 작업이 이뤄지지 않는다.</p> <p>예를 들어 Oracle JDBC 드라이버의 경우 auto-commit을 false로 설정해서 사용하다가 commit이나 rollback을 하지 않고 커넥션을 닫으면 무조건 commit을 하도록 되어 있는데 이러한 처리가 되지 않는다.</p> <p>만약 JDBC 드라이버에서 statement 캐시 기능을 제공한다면 그것을 사용하기를 권장한다.</p>
<code>[-sfs,--stmtFetchSize &lt;stmt-fetch-size&gt;]</code>	JDBC 드라이버 statement의 fetch 사이즈를 설정한다.
<code>[-ct,--connectionTrace &lt;true   false&gt;]</code>	<p>커넥션과 관련된 부가 정보 제공 여부를 결정한다.</p> <ul style="list-style-type: none"> <li>• false : --getConnectionTrace와 --autoCommitTrace가 모두 무효하다.</li> </ul>
<code>[-gct,--getConnectionTrace &lt;get-connection-trace&gt;]</code>	애플리케이션이 java.sql.DataSource#getConnection을 호출했을 때의 Stack Trace를 확인할 수 있도록 한다.
<code>[-act,--autoCommitTrace &lt;auto-commit-trace&gt;]</code>	java.sql.Connection#setAutoCommit이 호출되었을 때 관련 로그와 Stack Trace를 서버 로그에 기록한다. 단, jeus.jdbc.connection-trace 로거의 로그 레벨을 FINE으로 설정해야 한다.
<code>[-ust,--useSqlTrace &lt;use-sql-trace&gt;]</code>	<p>커넥션별로 사용하고 있는 SQL 쿼리를 보여주는 기능이다.</p> <p>jeus.jdbc.sql 로거의 레벨을 FINE으로 설정할 경우 서버 로그를 통해서 SQL 쿼리 히스토리를 확인할 수 있다. 이 기능을 사용할 경우 JDBC 드라이버의 statement 구현체를 JEUS의 것으로 감싸게되므로 JDBC 드라이버의 statement 객체를 캐스팅해서 사용하는 애플리케이션은 이 기능을 사용할 수 없다.</p>
<code>[-kcho,--keepConnectionHandleOpen &lt;keep-connection-handle-open&gt;]</code>	<p>true로 설정하면 XA 커넥션을 사용하고 Pool에 반납할 때 그에 대한 핸들(또는 논리적 커넥션)을 닫지 않고 열어둔다.</p> <p>DB2 universal 드라이버의 경우에 사용할 필요가 있다. 이 기능을 사용하면 커넥션 핸들이 닫히지 않으므로 커넥션을 닫을 때 드라이버가 수행하는 클리어 작업이 이뤄지지 않는다.</p> <p>예를 들어 Oracle JDBC 드라이버의 경우 auto-commit을 false로 설정해서 사용하다가 commit이나 rollback을 하지 않고 커넥션을 닫으면 무조건 commit을 하도록 되어 있는데 이러한 처리가 되지 않는다.</p>



파라미터	설명
<code>[-en,--exportName &lt;export-name&gt;]</code>	<p>데이터소스의 JNDI 이름을 설정한다.</p> <p>서로 다른 두 데이터소스가 서로 다른 서버에 JNDI 바인드되는 것을 보장할 수 있으면 해당 데이터소스들은 서로 같은 JNDI 이름을 가질 수 있다. 이는 임의의 서버에서 동일한 JNDI 이름을 가지는 서로 다른 데이터소스를 허용하지 않음을 의미한다.</p> <p>설정되지 않으면 데이터소스 ID를 JNDI 이름으로 사용한다.</p> <p>콤마(,)를 구분자로 사용하여 여러 개 입력할 수 있으며, 이 경우 구분자로 구분된 어떠한 값을 사용하여도 JNDI lookup이 가능하다.</p>
<code>[-dscn,--dataSourceClassName &lt;data-source-class-name&gt;]</code>	JDBC 드라이버 데이터소스 클래스의 이름을 설정한다. 패키지 이름을 포함하는 완전한 형태로 적는다.
<code>[-dst,--dataSourceType &lt;data-source-type&gt;]</code>	<p>데이터소스의 타입을 설정한다.</p> <ul style="list-style-type: none"> <li>• DATA_SOURCE : Connection Pooling 서비스는 제공되지 않는다.</li> <li>• CONNECTION_POOL_DATA_SOURCE : Connection Pooling 서비스가 제공된다.</li> <li>• XA_DATA_SOURCE : Connection Pooling 서비스와 더불어 XA 연동이 지원된다.</li> </ul>
<code>[-vendor,--vendor &lt;vendor&gt;]</code>	JDBC 드라이버 벤더의 이름을 설정한다.
<code>[-desc,--description &lt;description&gt;]</code>	데이터소스에 대한 설명이다.
<code>[-sn,--serverName &lt;server-name&gt;]</code>	데이터베이스가 실행되는 호스트 이름 또는 IP를 설정한다.
<code>[-pn,--portNumber &lt;port-number&gt;]</code>	데이터베이스 리스너의 포트 번호를 설정한다.
<code>[-dn,--databaseName &lt;database-name&gt;]</code>	<p>데이터베이스의 이름을 설정한다.</p> <p>Oracle의 경우 데이터베이스의 SID를 설정한다.</p>
<code>[-lt,--loginTimeout &lt;login-timeout&gt;]</code>	데이터베이스와 커넥션을 맺을 때 로그인 단계에서 기다리는 최대 시간이다. (단위: 초)
<code>[-il,--isolationLevel &lt;isolation-level&gt;]</code>	java.sql.Connection에서 정의하는 트랜잭션 isolation을 설정한다.

파라미터	설명
[-pdt,--poolDestroyTimeout <pool-destroy-timeout>]	<p>Connection Pool destroy 완료를 기다리는 시간이다. (단위: ms)</p> <p>데이터소스를 정의한 애플리케이션을 undeploy하거나 서버가 내려갈 때 Connection Pool을 destroy하게 되는데, 커넥션을 제거하면서 DB와 네트워크 통신을 할 경우 블록될 가능성이 존재하여 Connection Pool destroy를 마냥 기다리는 문제가 발생할 수 있다. 이와 같은 문제를 피하기 위해 이 설정을 사용한다. 설정한 시간 만큼만 Connection Pool destroy 완료를 기다려보고 undeploy 또는 서버 다운을 이어서 진행한다.</p>
[-prop,--property <property>]	<p>JDBC 드라이버별로 제각각일 수 있는 속성들의 설정을 모두 수용하기 위해 제공되는 통일된 설정 방법이다.</p> <p>한 개의 속성에 대하여 "이름:타입=값"의 형태로 입력한다.</p> <p>한 개 이상의 속성이 설정될 경우 콤마(,)로 구분한다.</p>
[-sxe,--supportXaEmulation <support-xa-emulation>]	<p>Connection Pool 데이터소스 타입의 데이터소스에만 유효한 설정으로 이 설정을 적용할 경우 Connection Pool 데이터소스의 커넥션이 글로벌 트랜잭션(XA)에 참여하도록 에뮬레이션한다.</p> <p>JEUS 6까지의 LocalXADataSource의 대체 옵션으로 ConnectionPoolDataSource 타입의 Connection Pool에 사용한다.</p> <p>하나의 트랜잭션에는 하나의 Connection Pool 데이터소스만 참여할 수 있다는 점에 유의해야 한다.</p>
[-ddba,--delegationDb <delegation-dba>]	<p>데이터베이스의 세션을 강제로 죽일 수 있는 권한(DBA 권한)을 가진 데이터소스(이하 DBA 위임 데이터소스)의 JNDI 이름을 설정한다. 이 설정을 한 데이터소스로부터 얻어진 커넥션을 이용한 쿼리 수행이 일정 시간 이상 지체되면 JEUS는 위임 DBA 데이터소스를 통해 해당 커넥션과 연관된 DB 세션을 강제로 제거하도록 하는 쿼리를 DB에 보낸다. 이후 애플리케이션이 사용 불가능해진 커넥션으로 인해 발생한 예외를 처리하고 커넥션을 닫아주게 되면 JEUS는 그 커넥션을 제거하고 DB로부터 새로운 커넥션을 얻어 Connection Pool에 넣는다.</p> <p>현재 Tiberio, Oracle, Sybase에 대해서 이 기능을 지원한다. 이 기능은 JDBC 2.0 이하 JDBC 드라이버에서 쿼리 수행이 지나치게 오래 걸릴 때 그것을 중단시킬 방법으로서 고안된 것이다. 그러나 JDBC 3.0 또는 그 이상의 버전을 구현한 JDBC 드라이버는 java.sql.Statement#setQueryTimeout을 구현하므로 이 기능을 통해 강제로 DB 세션을 제거하기보다는 Statement Query Timeout 설정을 이용하는 것을 권장한다.</p> <p>특히, XA 데이터소스의 경우 XA가 정상적으로 진행되는 도중에 DB 세션이 제거되면 XA 처리에 문제가 발생할 수 있기 때문에 Statement Query Timeout과 트랜잭션 타임아웃을 적절하게 설정하여 사용하도록 한다.</p>

파라미터	설명
<code>[-dbat,--dbaTimeout &lt;dba-timeout&gt;]</code>	<p>위임 DBA 데이터소스는 이 설정으로 지정한 시간 동안만 커넥션의 쿼리 수행을 기다린다. 설정한 시간이 경과하면 해당 커넥션과 연관된 DB 세션을 강제로 제거하도록 하는 쿼리를 DB로 보낸다.</p> <p>--delegationDba가 설정된 경우에만 유효하다. (단위: ms)</p>
<code>[-cqc,--checkQueryClass &lt;check-query-class&gt;]</code>	<p>사용자나 개발자가 커넥션 점검 기능을 커스터마이징할 경우 그것을 위해 구현한 클래스의 패키지 이름을 포함한 이름을 적어준다.</p> <p>이때 그 클래스는 반드시 <code>jeus.jdbc.connectionpool.JEUSConnectionChecker</code> 인터페이스를 구현해야 한다.</p>
<code>[-is,--initSql &lt;init-sql&gt;]</code>	커넥션을 생성한 후 가장 처음으로 수행할 SQL 쿼리를 설정한다.
<code>[-f,--forceLock]</code>	설정 변경을 강제로 진행하여 반영한다.
<code>[-detail]</code>	변경 결과에 대한 상세 정보가 출력된다.

- 예제

```
[MASTER]domain1.adminServer>add-data-source -id ds1 -dst ConnectionPoolDataSource
-dscn oracle.jdbc.pool.OracleConnectionPoolDataSource -sn 61.77.153.4 -pn 1521
-dn orcl -user scott -password tiger -property driverType;java.lang.String;thin
Successfully performed the ADD operation for data source [ds1] to domain.
Check the results using "add-data-source"
```

#### 4.2.11.3. list-cluster-data-sources

도메인에 존재하는 모든 클러스터 데이터소스의 목록을 조회한다. 특정 데이터소스의 ID를 명시할 경우 해당 클러스터 데이터소스의 설정을 자세히 조회한다.

- alias

listcds, lscds

- 사용법

```
list-cluster-data-sources [-id,--dataSourceID <data-source-id>]
```

- 파라미터

파라미터	설명
<code>[-id,--dataSourceID &lt;data-source-id&gt;]</code>	클러스터 데이터소스 ID이다.

- 예제

```
[MASTER]domain1.adminServer>list-cluster-data-sources
```

```
The list of cluster data sources
```

```
=====
```

Data source ID	JNDI export name	Component data sources
cds1	cds1	[ds1, ds2]
cds2	cds2	[ds3, ds4]

```
=====
```

```
[MASTER]domain1.adminServer>list-cluster-data-sources -id cds1
```

```
The configuration of cluster data source [cds1]
```

```
=====
```

Configuration name	Configuration value
id	cds1
export-name	cds1
load-balance	false
is-pre-conn	false
use-failback	true
component-data-sources	[ds1, ds2]

```
=====
```

#### 4.2.11.4. list-data-sources

도메인에 존재하는 모든 데이터소스의 목록을 조회한다. 특정 데이터소스의 ID를 명시할 경우 해당 데이터소스의 설정을 자세히 조회한다.

- alias

listds, lsds

- 사용법

```
list-data-sources [-id,--dataSourceID <data-source-id>]
```

- 파라미터

파라미터	설명
[-id,--dataSourceID <data-source-id>]	데이터소스 ID이다.

- 예제

```
[MASTER]domain1.adminServer>list-data-sources
```

```
The list of data sources
```

```
=====
```

Data source ID	JNDI export name	Data source type
ds1	ds1	ConnectionPoolDataSource
ds2	ds2	ConnectionPoolDataSource
ds3	ds3	ConnectionPoolDataSource

```
[MASTER]domain1.adminServer>list-data-sources -id ds1
```

The configuration of the data source [ds1]

Configuration name	Configuration value
id	ds1
export-name	ds1
data-source-class-name	oracle.jdbc.pool.OracleConnectionPoolDataSource
data-source-type	ConnectionPoolDataSource
vendor	oracle
server-name	61.77.153.4
port-number	1521
database-name	orcl
user	scott
password	tiger
login-timeout	0
auto-commit	DRIVER
stmt-query-timeout	0
pool-destroy-timeout	10000
property	[driverType;java.lang.String;thin]
action-on-connection-leak	Warning
support-xa-emulation	false
min	10
max	50
step	1
period	3600000
enable-wait	false
wait-time	10000
max-use-count	0
dbaTimeout	-1
stmt-caching-size	-1
stmt-fetch-size	-1
connection-trace	false
get-connection-trace	true
auto-commit-trace	false
use-sql-trace	false
keep-connection-handle-open	false

#### 4.2.11.5. modify-cluster-data-source

클러스터 데이터소스의 설정을 변경한다. 대부분의 설정이 동적으로 변경 가능하다.

- alias

## modifycds

### • 사용법

```
modify-cluster-data-source -id,--dataSourceID <data-source-id>
                           [-en,--exportName <export-name>]
                           [-dss,--dataSourceSelector <data-source-selector>]
                           [-lb,--loadBalance <true | false>]
                           [-ipc,--isPreConn <is-pre-conn>]
                           [-ufb,--useFailback <use-failback>]
                           [-cds,--componentDataSources <component-data-sources>]
                           [-oc,--onsConfig <ons-config>]
                           [-dsa,--dataSourceAffinity <data-source-affinity>]
                           [-f,--forceLock]
                           [-detail]
```

### • 파라미터

파라미터	설명
-id,--dataSourceID <data-source-id>]	클러스터 데이터소스의 ID를 설정한다.  하나의 도메인에서 클러스터 데이터소스 ID는 클러스터 데이터소스의 유일한 식별자로서 동작하도록 설정해야 한다.
[-en,--exportName <export-name>]	클러스터 데이터소스의 JNDI 이름을 설정한다. 서로 다른 두 데이터소스가 서로 다른 서버에 JNDI 바인드되는 것을 보장할 수 있으면 해당 데이터소스들은 서로 같은 JNDI 이름을 가질 수 있다. 이는 임의의 서버에서 동일한 JNDI 이름을 가지는 서로 다른 데이터소스를 허용하지 않음을 의미한다.  설정되지 않으면 클러스터 데이터소스 ID를 JNDI 이름으로 사용한다. 설정을 해제하려면 옵션 값으로 "unset"을 입력한다.  콤마(,)를 구분자로 사용하여 여러 개 입력할 수 있으며, 이 경우 구분자로 구분 된 어떠한 값을 사용하여도 JNDI lookup이 가능하다.

파라미터	설명
<code>[-dss,--dataSourceSelector &lt;data-source-selector&gt;]</code>	<p><b>[동적 변경]</b> 클러스터 데이터소스로부터 커넥션을 얻을 때 사용어나 개발자가 특정 컴포넌트 데이터소스 선택에 대한 정책을 직접 정의할 수 있다.</p> <p>jeus.jdbc.helper.DataSourceSelector 추상 클래스를 상속하여 구현하고 그 구현 클래스의 패키지 이름을 포함하는 이름을 적어준다. 이것을 설정하면 --loadBalance 설정은 아무런 기능을 하지 않으며 failover와 fallback을 무조건 시도한다.</p> <ul style="list-style-type: none"> <li>• failover는 이 설정을 통하여 최초 선택된 컴포넌트 데이터소스 다음 인덱스의 컴포넌트 데이터소스를 시작으로 하여 한 번만 라운딩하는 방식으로 이루어진다.</li> <li>• fallback은 Use fallback을 설정할 때 fallback이 수행되는 방식과 유사하게 이루어진다. 정책을 정의할 때는 대체로 동기화를 고려해야 하며 이는 구현자의 몫이다. 설정을 해제하려면 옵션 값으로 "unset"을 입력한다.</li> </ul>
<code>[-lb,--loadBalance &lt;true   false&gt;]</code>	<p><b>[동적 변경]</b> load balancing 여부를 설정한다.</p> <ul style="list-style-type: none"> <li>• true : --useFallback 설정은 아무런 기능을 하지 않게 된다.</li> </ul>
<code>[-ipc,--isPreConn &lt;is-pre-conn&gt;]</code>	<p><b>[동적 변경]</b> 클러스터 데이터소스에 속한 컴포넌트 데이터소스들의 Connection Pool을 미리 생성할지 여부를 결정한다. 컴포넌트 데이터소스들의 Connection Pool을 미리 생성해 놓으면 성능상으로 이점이 있으나 리소스의 절약면에서는 좋지 못하다.</p>
<code>[-ufb,--useFallback &lt;use-fallback&gt;]</code>	<p><b>[동적 변경]</b> 이전 버전의 JEUS에서는 failover만을 지원했으므로 이에 대한 호환성을 위해 제공하는 옵션이다.</p> <p>보조 데이터소스로 failover한 후에 주 데이터소스로 fallback할 것인지의 여부를 설정한다. 기본적으로 fallback을 시도한다. fallback을 위해서는 반드시 주 데이터소스에 대하여 --checkQuery 및 --checkQueryPeriod를 설정해야 한다.</p>
<code>[-cds,--componentDataSources &lt;component-data-sources&gt;]</code>	<p><b>[동적 변경]</b> 클러스터 데이터소스에 속한 컴포넌트 데이터소스들의 데이터소스 ID를 명시한다.</p>
<code>[-oc,--onsConfig &lt;ons-config&gt;]</code>	<p>ONS와 결합된 클러스터 데이터소를 사용하고자 할 때 설정한다. ONS상의 각 RAC 노드들이 ONS 통신에 사용하는 IP, 포트를 적는다. 클러스터 데이터소스는 설정된 IP, 포트들에 소켓 연결을 맺어 ONS 클라이언트로 동작하게 된다.</p> <p>다음과 같은 형식으로 적는다.</p> <div style="border: 1px solid #ccc; padding: 10px; margin-top: 10px;"> nodes=host1:6200,host2:6200 </div>

파라미터	설명
[-dsa,--dataSourceAffinity <data-source-affinity>]	트랜잭션 사용 중 데이터소스에 대한 Affinity 설정 여부를 정한다. 이 설정이 켜지면 트랜잭션 처리가 하나의 멤버 데이터소스 인스턴스에 한정하여 이루어져 글로벌 트랜잭션은 처리 성능 향상을 도모할 수 있고, XA Emulation을 설정한 로컬 트랜잭션도 클러스터 데이터소스 사용 간 보장 될 수 있다.
[-f,--forceLock]	설정 변경을 강제로 진행하여 반영한다.
[-detail]	변경 결과에 대한 상세 정보가 출력된다.

- 예제

```
[MASTER]domain1.adminServer>modify-cluster-data-source -id cds1 -cds ds1
Successfully performed the MODIFY operation for configuration of the cluster data source [cds1].
Check the results using "modify-cluster-data-source -id cds1"
```

#### 4.2.11.6. modify-data-source

데이터소스의 설정을 변경한다. 일부 설정은 동적으로 변경 가능하다.

- alias

modifyds

- 사용법

```
modify-data-source -id,--dataSourceID <data-source-id>
    [-user,--user <user>]
    [-pw,--password <password>]
    [-algorithm,--algorithm <algorithm>]
    [-ac,--autoCommit <TRUE | FALSE | DRIVER> ]
    [-sqt,--stmtQueryTimeout <statement-query-timeout>]
    [-aocl,--actionOnConnectionLeak <action-on-connection-leak>]
    [-min,--min <minimum-size>]
    [-max,--max <maximum-size>]
    [-step,--step <resizing-step>]
    [-period,--period <resizing-period>]
    [-ew,--enableWait <true | false>]
    [-wt,--waitTime <wait-time>]
    [-dds,--delegationDataSource <delegation-data-source>]
    [-muc,--maxUseCount <max-use-count>]
    [-cq,--checkQuery <check-query>]
    [-cqt,--checkQueryTimeout <check-query-timeout>]
    [-nvi,--nonValidationInterval <non-validation-interval>]
    [-cqp,--checkQueryPeriod <check-query-period>]
    [-cqrc,--checkQueryRetrialCount <check-query-retrial-count>]
    [-dpocq,--destroyPolicyOnCheckQuery <destroy-policy-on-check-query>]
```



```

[-scs,--stmtCachingSize <stmt-caching-size>]
[-sfs,--stmtFetchSize <stmt-fetch-size>]
[-ct,--connectionTrace <true | false>]
[-gct,--getConnectionTrace <get-connection-trace>]
[-act,--autoCommitTrace <auto-commit-trace>]
[-ust,--useSqlTrace <use-sql-trace>]
[-kcho,--keepConnectionHandleOpen <keep-connection-handle-open>]
[-en,--exportName <export-name>]
[-dscn,--dataSourceClassName <data-source-class-name>]
[-dst,--dataSourceType <data-source-type>]
[-vendor,--vendor <vendor>]
[-desc,--description <description>]
[-sn,--serverName <server-name>]
[-pn,--portNumber <port-number>]
[-dn,--databaseName <database-name>]
[-lt,--loginTimeout <login-timeout>]
[-il,--isolationLevel <isolation-level>]
[-pdt,--poolDestroyTimeout <pool-destroy-timeout>]
[-prop,--property <property>]
[-sxe,--supportXaEmulation <support-xa-emulation>]
[-ddba,--delegationDbc <delegation-dba>]
[-dbat,--dbaTimeout <dba-timeout>]
[-cqc,--checkQueryClass <check-query-class>]
[-is,--initSql <init-sql>]
[-f,--forceLock]
[-detail]

```

#### • 파라미터

파라미터	설명
-id,--dataSourceID <data-source-id>	데이터소스의 ID를 설정한다. 하나의 도메인에서 데이터소스 ID는 데이터소스의 유일한 식별자로서 동작하도록 설정해야 한다.
[-user,--user <user>]	<p><b>[동적 변경]</b> 데이터베이스 사용자의 ID를 설정한다. 트랜잭션 처리 등을 위해서는 충분한 권한을 가지고 있어야 한다.</p> <p>설정을 해제하려면 옵션 값으로 "unset"을 입력한다.</p>
[-pw,--password <password>]	<p><b>[동적 변경]</b> 데이터베이스 사용자의 패스워드를 설정한다.</p> <p>패스워드는 '{algorithm:keysize}ciphertext' 와 같은 형식으로 암호화해서 저장한다.</p> <p>설정을 해제하려면 옵션 값으로 "unset"을 입력한다.</p>

파라미터	설명
[-algorithm,--algorithm <algorithm>]	<p><b>[동적 변경]</b> 데이터베이스 사용자 패스워드의 알고리즘을 설정한다. algorithm 옵션은 password 옵션과 함께 사용될 때에만 의미가 있다.</p> <p>사용할 수 있는 알고리즘은 다음 중에 선택한다.</p> <ul style="list-style-type: none"> <li>• base64</li> <li>• DES</li> <li>• DESede</li> <li>• AES</li> <li>• SEED</li> <li>• Blowfish</li> </ul>
[-ac,--autoCommit <TRUE   FALSE   DRIVER>]	<p><b>[동적 변경]</b> 데이터소스의 기본 auto-commit 설정으로 다음 중에 하나를 설정한다.</p> <ul style="list-style-type: none"> <li>• TRUE</li> <li>• FALSE</li> <li>• DRIVER : JEUS는 auto-commit 설정에 관여하지 않으며 JDBC 드라이버의 auto-commit 설정을 따른다.</li> </ul> <p>XA 에뮬레이션 기능을 사용하는 Connection Pool 데이터소스 또는 XA 데이터소스의 경우 트랜잭션이 연동되어 있지 않을 때만 이 설정값이 유효하다.</p>
[-sqt,--stmtQueryTimeout <statement-query-timeout>]	<p><b>[동적 변경]</b> 데이터소스로부터 얻은 커넥션을 통해 생성된 java.sql.Statement 객체에 지정될 쿼리 타임아웃을 설정한다.</p> <p>JEUS는 JDBC API를 통해 정의되고 JDBC 드라이버 벤더에서 구현한 쿼리 타임아웃 설정 메소드 (java.sql.Statement#setQueryTimeout)를 호출할 뿐이다. JDBC 드라이버 벤더별로 쿼리 타임아웃 적용에 약간의 차이를 보인다. (단위: ms)</p>
[-aocl,--actionOnConnectionLeak <action-on-connection-leak>]	<p><b>[동적 변경]</b> 애플리케이션(주로 stateless 컴포넌트 - Servlet/JSP, stateless Session Bean, MDB)에서 사용하고 제대로 반환하지 않은 JDBC 커넥션을 감지했을 때 JEUS가 취하는 액션을 정의한다.</p> <p>다음의 값 중에 하나를 설정한다.</p> <ul style="list-style-type: none"> <li>• NO_ACTION : 아무런 액션을 취하지 않는다.</li> <li>• WARNING : 반환되지 않은 JDBC 커넥션 정보를 로그로 남긴다.</li> <li>• AUTO_CLOSE : 반환되지 않은 JDBC 커넥션 정보를 로그로 남기며 해당 JDBC 커넥션을 회수한다.</li> </ul> <p>설정하지 않을 경우 서버의 설정을 따른다. 설정을 해제하려면 옵션 값으로 "unset"을 입력한다.</p>

파라미터	설명
[-min,--min <minimum-size>]	<b>[동적 변경]</b> Connection Pool에 Pooling되는 커넥션의 최소값을 지정한다.
[-max,--max <maximum-size>]	<b>[동적 변경]</b> Connection Pool에 Pooling되는 커넥션의 최대값을 지정한다.
[-step,--step <resizing-step>]	<b>[동적 변경]</b> Connection Pool에 커넥션이 부족할 때 현재 커넥션의 개수가 최대값 이하인 경우 DB로부터 커넥션을 새로 받아와 채우는 데 이때 새로 받아오는 커넥션의 개수를 지정한다.
[-period,--period <resizing-period>]	<b>[동적 변경]</b> Connection Pool의 크기를 최소값에 맞춰 조정하는 주기를 설정한다. Connection Pool의 크기가 최소값을 초과하는 경우에는 사용하지 않는 커넥션들을 닫아주고 Connection Pool의 크기가 최소값에 미치지 못하는 경우에는 DB로 붙여 커넥션을 새로 받아와 채운다. (단위: ms)
[-ew,--enableWait <true   false>]	<b>[동적 변경]</b> Connection Pool에 사용 가능한 커넥션이 없고 커넥션도 더 이상 늘릴 수 없을 때 커넥션 요청을 처리하는 방법을 결정한다. <ul style="list-style-type: none"> <li>• true : 이용 가능한 커넥션을 얻기 위해 기다린다.</li> <li>• false : 새로운 커넥션을 만들어서 제공하지만 그 커넥션은 반환되었을 때 Pooling되지 않고 버려진다. 이를 일회용(disposable) 커넥션이라고도 한다.</li> </ul>
[-wt,--waitTime <wait-time>]	<b>[동적 변경]</b> --enable-wait이 true인 경우에만 유효한 설정으로 커넥션을 얻기 위해 대기하는 한계 시간을 나타낸다. 만약 이 시간이 지나도 커넥션을 얻지 못하면 JEUS는 타임아웃 예외를 발생시킨다. (단위: ms)
[-dds,--delegationDataSource <delegation-data-source>]	<b>[동적 변경]</b> 트랜잭션과 연동하지 않은 상태에서는 XA 데이터소스를 통해 커넥션을 얻기보다 Connection Pool 데이터소스를 통해 커넥션을 얻는 것이 낫다. <p>기능의 차이도 없고, 트랜잭션 연동을 위한 기능을 포함하고 있는 XA 커넥션은 시스템에 부담을 더 주기 때문이다. 이를 위해 XA 데이터소스인 경우 이 설정을 통하여 트랜잭션과 연동하지 않은 상태에서의 커넥션 요청을 위임할 Connection Pool 데이터소스를 지정한다. 한편 Oracle, DB2 등에서 XA 커넥션을 트랜잭션 없이도 사용하고, 트랜잭션에 연동도 하면서 사용하다 보면 XA를 시작할 수 없는 예외가 발생하기도 하는데 정확한 원인은 알 수 없기 때문에 이를 회피하기 위한 방편으로서도 이 설정을 이용한다.</p> <p>설정을 해제하려면 옵션 값으로 "unset"을 입력한다.</p>
[-muc,--maxUseCount <max-use-count>]	<b>[동적 변경]</b> 커넥션의 최대 사용 횟수이다. 이 사용 횟수 이상이 되면 새로운 커넥션으로 교체한다. 기본값은 0으로 이는 커넥션을 교체하지 않겠다는 의미다.

파라미터	설명
<code>[-cq,--checkQuery &lt;check-query&gt;]</code>	<p><b>[동적 변경]</b> 커넥션 상태 점검에 사용될 쿼리(check-query)를 설정한다. 보통 DB와의 연결 유효성만을 확인하면 되므로 간단한 select 쿼리를 사용하는 것을 권장한다.</p> <p>설정을 해제하려면 옵션 값으로 "unset"을 입력한다.</p>
<code>[-cqt,--checkQueryTimeout &lt;check-query-timeout&gt;]</code>	<p><b>[동적 변경]</b> 커넥션 점검을 위해 check-query를 수행했을 때 DB가 응답이 없어 드라이버가 계속 기다리는 상황이 발생할 수 있다. 이런 경우를 피하기 위해 check-query에 대해 쿼리 타임아웃을 적용한다. 이것은 JDBC API에서 정의한 <code>java.sql.Statement#setQueryTimeout</code> 메소드의 호출을 통해 가능하다.</p> <p>1000ms보다 작을 경우 0으로 설정되므로 주의한다. (단위: ms)</p>
<code>[-nvi,--nonValidationInterval &lt;non-validation-interval&gt;]</code>	<p><b>[동적 변경]</b> 커넥션 점검이 너무 잦아서 오버헤드가 발생하는 경우 설정한다. 커넥션 점검을 수행하기 직전의 시각과 가장 최근의 커넥션 점검 시각과의 차이가 설정한 시간 간격 이내면 커넥션 점검을 생략하도록 하는 설정이다. (단위: ms)</p> <p>예를 들어 이 설정값이 5000ms인 경우 어떤 커넥션의 마지막 커넥션 점검 시간으로부터 아직 5초가 지나지 않았다면 그 커넥션에 대한 점검이 생략된 채 애플리케이션에 전달되게 된다.</p>
<code>[-cqp,--checkQueryPeriod &lt;check-query-period&gt;]</code>	<p><b>[동적 변경]</b> Connection Pool의 커넥션들을 설정한 주기마다 체크하여 문제가 있는 커넥션을 제거한다. (단위: ms)</p> <p>클러스터 데이터소스에 속한 데이터소스는 자신의 상태 체크에 사용하므로 반드시 설정해야 한다.</p>
<code>[-cqrc,--checkQueryRetrialCount &lt;check-query-retrial-count&gt;]</code>	<p><b>[동적 변경]</b> 커넥션 점검은 기본적으로 <code>--destroyPolicyOnCheckQuery</code>가 <code>FAILED_CONNECTION_ONLY</code>로 설정된 경우 한 번 수행된다.</p> <p><code>--destroyPolicyOnCheckQuery</code>가 <code>ALL_CONNECTIONS</code>로 설정되어 있을 경우에는 최초의 커넥션 점검에서 커넥션 이상이 확인되면 또 다른 커넥션에 대해서 한 번 더 커넥션 점검이 이루어져 총 두 번의 커넥션 점검이 수행될 수 있다. 이 설정값이 이러한 기본 커넥션 점검 수행 횟수에 더해져 최종 커넥션 점검 수행 횟수가 정해진다.</p>
<code>[-dpocq,--destroyPolicyOnCheckQuery &lt;destroy-policy-on-check-query&gt;]</code>	<p><b>[동적 변경]</b> 커넥션이 유효하지 않은 것으로 확인되었을 때 Connection Pool에 있는 다른 커넥션들에 대한 처리 정책을 설정한다.</p> <ul style="list-style-type: none"> <li>• <code>FAILED_CONNECTION_ONLY</code> : 유효하지 않은 것으로 확인된 커넥션만 제거한다.</li> <li>• <code>ALL_CONNECTIONS</code> : 유효하지 않은 것으로 확인된 커넥션을 제거하고 Connection Pool에 있는 다른 커넥션의 유효성을 한 번 더 확인한다. 그조차 유효하지 않은 것으로 확인되면 Connection Pool의 모든 커넥션을 제거한다.</li> </ul>

파라미터	설명
<code>[-scs,--stmtCachingSize &lt;stmt-caching-size&gt;]</code>	<p><b>[동적 변경]</b> JDBC 드라이버는 애플리케이션에서 prepared statement를 요청할 때마다 파라미터로 넘어온 SQL 문장을 파싱한다. 이 파싱 작업이 성능에 영향을 줄 수 있기 때문에 이를 피하기 위해서 JEUS 내부적으로 prepared statement를 캐시하는 기능을 제공한다. 이 설정은 캐싱할 prepared statement의 개수를 지정한다. 이 기능을 사용하면 커넥션을 항상 열어둔 채로 사용하기 때문에 커넥션을 닫았을 때 드라이버가 해주는 클리어 작업이 이뤄지지 않는다.</p> <p>예를 들어 Oracle JDBC 드라이버의 경우 auto-commit을 false로 설정해서 사용하다가 commit이나 rollback을 하지 않고 커넥션을 닫으면 무조건 commit을 하도록 되어 있는데 이러한 처리가 되지 않는다.</p> <p>만약 JDBC 드라이버에서 statement 캐시 기능을 제공한다면 그것을 사용하기 바란다.</p>
<code>[-sfs,--stmtFetchSize &lt;stmt-fetch-size&gt;]</code>	<p><b>[동적 변경]</b> JDBC 드라이버 statement의 fetch 사이즈를 설정한다.</p>
<code>[-ct,--connectionTrace &lt;true   false&gt;]</code>	<p><b>[동적 변경]</b> 커넥션 관련 부가 정보 제공 여부를 결정한다.</p> <ul style="list-style-type: none"> <li>• false : --getConnectionTrace와 --autoCommitTrace가 모두 무효가 된다.</li> </ul>
<code>[-gct,--getConnectionTrace &lt;get-connection-trace&gt;]</code>	<p><b>[동적 변경]</b> 애플리케이션이 java.sql.DataSource#getConnection을 호출했을 때의 Stack Trace를 확인할 수 있도록 한다.</p>
<code>[-act,--autoCommitTrace &lt;auto-commit-trace&gt;]</code>	<p><b>[동적 변경]</b> java.sql.Connection#setAutoCommit이 호출되었을 때 관련 로그와 Stack Trace를 서버 로그에 기록하도록 한다. 단, jeus.jdbc.connection-trace 로거의 로그 레벨을 FINE으로 설정해야 한다.</p>
<code>[-ust,--useSqlTrace &lt;use-sql-trace&gt;]</code>	<p><b>[동적 변경]</b> Connection별로 사용하고 있는 SQL 쿼리를 보여주는 기능이다. jeus.jdbc.sql 로거의 레벨을 FINE으로 설정할 경우 서버 로그를 통해서 SQL 쿼리 히스토리를 확인할 수 있다. 이 기능을 사용할 경우 JDBC 드라이버의 statement 구현체를 JEUS의 것으로 감싸게되므로 JDBC 드라이버의 statement 객체를 캐스팅해서 사용하는 애플리케이션은 이 기능을 사용할 수 없다.</p>
<code>[-kcho,--keepConnectionHandleOpen &lt;keep-connection-handle-open&gt;]</code>	<p><b>[동적 변경]</b> true로 설정하면 XA 커넥션을 사용하고 Pool에 반납할 때 그에 대한 핸들(또는 논리적 커넥션)을 닫지 않고 열어둔다.</p> <p>DB2 universal 드라이버의 경우에 사용할 필요가 있다. 이 기능을 사용하면 커넥션 핸들이 닫히지 않으므로 커넥션을 닫을 때 드라이버가 해주는 클리어 작업이 이뤄지지 않는다.</p> <p>예를 들어 Oracle JDBC 드라이버의 경우 auto-commit을 false로 설정해서 사용하다가 commit이나 rollback을 하지 않고 커넥션을 닫으면 무조건 commit을 하도록 되어 있는데 이러한 처리가 되지 않는다.</p>

파라미터	설명
<code>[-en,--exportName &lt;export-name&gt;]</code>	<p>데이터소스의 JNDI 이름을 설정한다. 서로 다른 두 데이터소스가 서로 다른 서버에 JNDI 바인드되는 것을 보장할 수 있으면 해당 데이터소스들은 서로 같은 JNDI 이름을 가질 수 있다. 이는 임의의 서버에서 동일한 JNDI 이름을 가지는 서로 다른 데이터소스를 허용하지 않음을 의미한다.</p> <p>설정되지 않으면 데이터소스 ID를 JNDI 이름으로 사용한다. 설정을 해제하려면 옵션 값으로 "unset"을 입력한다.</p> <p>coma(,)를 구분자로 사용하여 여러 개 입력할 수 있으며, 이 경우 구분자로 구분 된 어떠한 값을 사용하여도 JNDI lookup이 가능하다.</p>
<code>[-dscn,--dataSourceClassName &lt;data-source-class-name&gt;]</code>	JDBC 드라이버 데이터소스 클래스의 이름을 설정한다. 패키지 이름을 포함하는 완전한 형태로 적는다.
<code>[-dst,--dataSourceType &lt;data-source-type&gt;]</code>	<p>데이터소스의 타입을 설정한다.</p> <ul style="list-style-type: none"> <li>• DATA_SOURCE : Connection Pooling 서비스는 제공되지 않는다.</li> <li>• CONNECTION_POOL_DATA_SOURCE : Connection Pooling 서비스가 제공되며 XA_DATA_SOURCE로 설정한 경우 Connection Pooling 서비스와 더불어 XA 연동이 지원된다.</li> </ul>
<code>[-vendor,--vendor &lt;vendor&gt;]</code>	<p>JDBC 드라이버 벤더의 이름을 설정한다.</p> <p>설정을 해제하려면 옵션 값으로 "unset"을 입력한다.</p>
<code>[-desc,--description &lt;description&gt;]</code>	<p>데이터소스에 대한 설명이다.</p> <p>설정을 해제하려면 옵션 값으로 "unset"을 입력한다.</p>
<code>[-sn,--serverName &lt;server-name&gt;]</code>	<p>데이터베이스가 실행되는 호스트 이름 또는 IP를 설정한다.</p> <p>설정을 해제하려면 옵션 값으로 "unset"을 입력한다.</p>
<code>[-pn,--portNumber &lt;port-number&gt;]</code>	데이터베이스 리스너의 포트 번호를 설정한다. 설정을 해제하려면 옵션 값으로 "unset"을 입력한다.
<code>[-dn,--databaseName &lt;database-name&gt;]</code>	데이터베이스의 이름을 설정한다. Oracle의 경우 데이터베이스의 SID를 설정한다. 설정을 해제하려면 옵션 값으로 "unset"을 입력한다.
<code>[-lt,--loginTimeout &lt;login-timeout&gt;]</code>	데이터베이스와 커넥션을 맺을 때 로그인 단계에서 기다리는 최대 시간을 설정한다. (단위: 초)
<code>[-il,--isolationLevel &lt;isolation-level&gt;]</code>	<p>java.sql.Connection에서 정의하는 트랜잭션 isolation을 설정한다. 설정하지 않은 경우 JDBC 드라이버의 기본 설정을 따른다.</p> <p>설정을 해제하려면 옵션 값으로 "unset"을 입력한다.</p>

파라미터	설명
<code>[-pdt,--poolDestroyTimeout &lt;pool-destroy-timeout&gt;]</code>	<p>Connection Pool destroy 완료를 기다리는 시간이다. (단위: ms)</p> <p>데이터소스를 정의한 애플리케이션을 undeploy하거나 서버가 내려갈 때 Connection Pool을 destroy하게 되는데, 커넥션을 제거하면서 DB와 네트워크 통신을 할 경우 블록될 가능성이 존재하여 Connection Pool destroy를 마냥 기다리는 문제가 발생할 수 있다. 이와 같은 문제를 피하기 위해 이 설정을 사용한다.</p> <p>설정된 시간 만큼만 Connection Pool destroy 완료를 기다려보고 undeploy 또는 서버 다운을 이어서 진행한다.</p>
<code>[-prop,--property &lt;property&gt;]</code>	<p>JDBC 드라이버별로 제각각일 수 있는 속성들의 설정을 모두 수용하기 위해 제공되는 통일된 설정 방법이다.</p> <ul style="list-style-type: none"> <li>• 한 개의 속성은 "이름:타입=값"의 형태로 입력한다.</li> <li>• 한 개 이상의 속성을 설정할 경우 콤마(,)로 구분한다.</li> </ul>
<code>[-sxe,--supportXaEmulation &lt;support-xa-emulation&gt;]</code>	<p>Connection Pool 데이터소스 타입의 데이터소스에만 유효한 설정으로 이 설정을 적용할 경우 Connection Pool 데이터소스의 Connection이 글로벌 트랜잭션(XA)에 참여하도록 에뮬레이션한다.</p> <p>JEUS 6까지의 LocalXADataSource의 대체 옵션으로 ConnectionPoolDataSource 타입의 Connection Pool에 사용한다. 하나의 트랜잭션에는 하나의 Connection Pool 데이터소스만 참여할 수 있다는 점에 유의해야 한다.</p>
<code>[-ddba,--delegationDb &lt;delegation-dba&gt;]</code>	<p>데이터베이스의 세션을 강제로 죽일 수 있는 권한(DBA 권한)을 가진 데이터소스(이하 DBA 위임 데이터소스)의 JNDI 이름을 설정한다. 이 설정을 한 데이터소스로부터 얻어진 커넥션을 이용한 쿼리 수행이 일정 시간 이상 지체되면 JEUS는 위임 DBA 데이터소스를 통해 해당 커넥션과 연관된 DB 세션을 강제로 제거하도록 하는 쿼리를 DB에 날린다. 이후 애플리케이션이 사용 불가능해진 커넥션으로 인해 발생한 예외를 처리하고 커넥션을 닫아주게 되면 JEUS는 그 커넥션을 제거하고 DB로부터 새로운 커넥션을 얻어 Connection Pool에 넣는다.</p> <p>현재 Tiberio, Oracle, Sybase에 대해서 이 기능을 지원한다. 이 기능은 JDBC 2.0 이하 JDBC 드라이버에서 쿼리 수행이 지나치게 오래 걸릴 때 그것을 중단시킬 방법으로서 고안된 것이다. 그러나 JDBC 3.0 또는 그 이상의 버전을 구현한 JDBC 드라이버는 <code>java.sql.Statement#setQueryTimeout</code>을 구현하므로 이 기능을 통해 강제로 DB 세션을 제거하기보다는 Statement Query Timeout 설정을 이용하는 것을 권장한다. 특히나 XA 데이터소스의 경우 XA가 정상적으로 진행하는 도중에 DB 세션이 제거되면 XA 처리에 문제가 발생할 수 있기 때문에 Statement Query Timeout과 트랜잭션 타임아웃을 적절하게 설정하여 사용하도록 한다.</p> <p>설정을 해제하려면 옵션 값으로 "unset"을 입력한다.</p>

파라미터	설명
<code>[-dbat,--dbaTimeout &lt;dba-timeout&gt;]</code>	위임 DBA 데이터소스는 이 설정으로 지정한 시간 동안만 커넥션의 쿼리 수행을 기다려준다. 설정한 시간이 경과하면 해당 커넥션과 연관된 DB 세션을 강제로 제거하는 쿼리를 DB로 보낸다.  --delegationDb가 설정된 경우에만 유효하다. (단위: ms)
<code>[-cqc,--checkQueryClass &lt;check-query-class&gt;]</code>	사용자나 개발자가 커넥션 점검 기능을 커스터마이징할 경우 그것을 위해 구현한 클래스의 패키지 이름을 포함한 이름을 적어준다. 이때 그 클래스는 반드시 <code>jeus.jdbc.connectionpool.JEUSConnectionChecker</code> 인터페이스를 구현해야 한다.  설정을 해제하려면 옵션 값으로 "unset"을 입력한다.
<code>[-is,--initSql &lt;init-sql/&gt;]</code>	커넥션을 생성한 후 가장 처음으로 수행할 SQL 쿼리를 설정한다.  설정을 해제하려면 옵션 값으로 "unset"을 입력한다.
<code>[-f,--forceLock]</code>	설정 변경을 강제로 진행하여 반영한다.
<code>[-detail]</code>	변경 결과에 대한 상세 정보가 출력된다.

- 예제

```
[MASTER]domain1.adminServer>modify-data-source -id ds1 -min 10 -max 50
Successfully performed the MODIFY operation for configuration of the data source [ds1].
Check the results using "modify-data-source -id ds1"
```

#### 4.2.11.7. remove-cluster-data-source

클러스터 데이터소스를 도메인으로부터 동적으로 삭제한다.

- alias

rmcds

- 사용법

```
remove-cluster-data-source [-id,--dataSourceID <data-source-id>]
                           [-f,--forceLock]
                           [-detail]
```

- 파라미터

파라미터	설명
<code>[-id,--dataSourceID &lt;data-source-id&gt;]</code>	클러스터 데이터소스 ID이다.



파라미터	설명
[-f,--forceLock]	설정 변경을 강제로 진행하여 반영한다.
[-detail]	변경 결과에 대한 상세 정보가 출력된다.

- 예제

```
[MASTER]domain1.adminServer>remove-cluster-data-source -id cds1
Successfully performed the REMOVE operation for cluster data source [cds1] from domain.
Check the results using "remove-cluster-data-source"
```

#### 4.2.11.8. remove-data-source

데이터소스를 도메인으로부터 동적으로 삭제한다.

- alias

rmids

- 사용법

```
remove-data-source [-id,--dataSourceID <data-source-id>]
                  [-f,--forceLock]
                  [-detail]
```

- 파라미터

파라미터	설명
[-id,--dataSourceID <data-source-id>]	데이터소스 ID를 설정한다.
[-f,--forceLock]	설정 변경을 강제로 진행하여 반영한다.
[-detail]	변경 결과에 대한 상세한 정보가 출력된다.

- 예제

```
[MASTER]domain1.adminServer>remove-data-source -id ds1
Successfully performed the REMOVE operation for data source [ds1] from the domain.
Check the results using "remove-data-source"
```

#### 4.2.11.9. test-data-source-config

데이터소스의 설정이 올바른지 확인한다.

- alias

testdsconf, testdsconfig, testdscfg

- 사용법

```
test-data-source-config -id <data-source-id>
                        [-all]
```

- 파라미터

파라미터	설명
-id <data-source-id>	데이터소스 ID이다.
[-all]	JEUS Master Server의 jeusadmin에서만 사용 가능하며, 도메인 내의 기동 중인 모든 서버들 중 해당 데이터소스를 타겟으로 설정한 서버에게 명령을 요청하고 결과를 수집하여 보여준다.

- 예제

```
[MASTER]domain1.adminServer>test-data-source-config -id ds1
The data source [ds1] is properly configured and can be used.
```

## 4.2.12. Connection Pool 제어 및 모니터링 명령어

JDBC나 JCA Connection Pool에 대한 모니터링 및 제어 기능을 제공한다. 명령어들의 목록은 다음과 같다.

명령어	설명
<a href="#">connection-pool-info</a>	JDBC나 JCA Connection Pool을 모니터링한다.  특정 Connection Pool의 ID를 명시할 경우 해당 Connection Pool의 정보를 자세히 조회한다.
<a href="#">control-connection-pool</a>	JDBC나 JCA Connection Pool을 제어한다.  Connection Pool 생성, Connection Pool 활성화, Connection Pool 비활성화, Connection Pool refresh, Connection Pool shrink가 가능하다.
<a href="#">control-cluster-data-source</a>	클러스터 데이터소스를 제어한다. 클러스터 데이터소스에 속한 컴포넌트 데이터소스의 목록을 보여주거나 failback을 수행할 수 있다.
<a href="#">create-connection-pool</a>	JDBC Connection Pool을 생성한다.  JCA Connection Pool에 대해서는 동작하지 않는다.
<a href="#">destroy-connection</a>	Active 상태의 커넥션의 연결을 강제로 끊는다.
<a href="#">disable-connection-pool</a>	JDBC Connection Pool을 비활성화한다.  JCA Connection Pool에 대해서는 동작하지 않는다.

명령어	설명
<code>enable-connection-pool</code>	JDBC Connection Pool을 활성화한다. JCA Connection Pool에 대해서는 동작하지 않는다.
<code>refresh-connection-pool</code>	JDBC Connection Pool의 커넥션들을 새로운 커넥션으로 교체한다. JCA Connection Pool에 대해서는 동작하지 않는다.
<code>return-connection</code>	Active 상태의 커넥션을 Connection Pool로 반환한다.
<code>shrink-connection-pool</code>	JDBC Connection Pool의 커넥션 개수를 설정된 커넥션 최솟값으로 조정한다. JCA Connection Pool에 대해서는 동작하지 않는다.

#### 4.2.12.1. connection-pool-info

JDBC나 JCA Connection Pool을 모니터링한다. 특정 Connection Pool의 ID를 명시할 경우 해당 Connection Pool의 정보를 자세히 조회한다.

- alias
- cpinfo
- 사용법

```
connection-pool-info [-server <server-name>]
                    [-id <connection-pool-id>]
                    [-jca | -jdbc]
                    [-jndi]
                    [-active]
                    [-t]
                    [-stmt]
```

- 파라미터

파라미터	설명
<code>[-server &lt;server-name&gt;]</code>	서버 이름이다.
<code>[-id &lt;connection-pool-id&gt;]</code>	Connection Pool ID이다.
<code>[-jca]</code>	JCA Connection Pool의 정보만 표시한다.
<code>[-jdbc]</code>	JDBC Connection Pool의 정보만 표시한다.
<code>[-jndi]</code>	Connection Pool의 JNDI 이름을 표시한다.
<code>[-active]</code>	현재 서버에 생성된 Connection Pool만 표시한다.
<code>[-t]</code>	커넥션을 가장 최근에 사용한 Thread의 이름을 표시한다. -id 옵션이 설정된 경우에만 유효하다.
<code>[-stmt]</code>	각 커넥션별 Statement Cache의 정보를 표시한다.

- 예제

```
[MASTER]domain1.adminServer>connection-pool-info -server server1 -jndi -jdbc
The connection pool information on the server [server1].
```

```
=====
```

Connection Pool ID	JNDI Expo rt Name	Min	Max	Act ive Max	Act ive	Act ive Aver age	Idle	Dis posa ble	Tot al	Wait se	Ena bled
ds1	ds1	2	30	0	0	0.0	2	0	2	fal se	true
ds2 *	ds2	2	30	0	0	0.0	0	0	0	fal se	fal se

```
=====
```

\* : has not been created, total = active + idle + disposable

```
[MASTER]domain1.adminServer>connection-pool-info -server server1 -id ds1 -t -jdbc
Information about connections in the server [server1]'s connection pool [ds1].
```

```
=====
```

Connection ID	State	State time(sec.)	Use count	Type	Thread name
ds1-1	active	1105.954	1	pooled	http-w1
ds1-2	idle	1105.923	0	pooled	

```
=====
```

```
[MASTER]domain1.adminServer>connection-pool-info -server server1 -id ds1 -stmt -jdbc
Information about statement caches in the server [server1]'s connection pool [ds1].
```

```
=====
```

Connection ID	Access Count	Hit Count	Miss Count	Add Count	Remove Count	Current Size
ds1-1	5	0	5	5	0	5
ds1-2	5	2	3	3	0	3

```
=====
```

#### 4.2.12.2. control-connection-pool

JDBC나 JCA Connection Pool을 제어한다. Connection Pool 생성, Connection Pool 활성화, Connection Pool 비활성화, Connection Pool refresh, Connection Pool shrink가 가능하다.

- alias

controlcp, ctrlcp

- 사용법

```
control-connection-pool -id <connection-pool-id>
                        [-server <server-name>]
                        -enable | -shrink | -disable | -refresh | -create
```

- 파라미터

파라미터	설명
-id <connection-pool-id>	Connection Pool ID이다.
[-server <server-name>]	서버 이름이다.
-enable	Connection Pool을 활성화한다.
-shrink	Connection Pool의 커넥션 개수를 설정된 커넥션 최솟값으로 조정한다.
-disable	Connection Pool을 비활성화한다.
-refresh	Connection Pool의 커넥션들을 새로운 커넥션으로 교체한다.
-create	Connection Pool을 생성한다.

- 예제

```
[MASTER]domain1.adminServer>control-connection-pool -id ds1 -server server1 -enable
Servers that successfully enabled a connection pool : server1
Servers that failed to enable a connection pool : none.

[MASTER]domain1.adminServer>control-connection-pool -id ds1 -server server1 -shrink
Servers that successfully shrank a connection pool : server1
Servers that failed to shrink a connection pool : none.

[MASTER]domain1.adminServer>control-connection-pool -id ds1 -server server1 -disable
Servers that successfully disabled a connection pool : server1
Servers that failed to disable a connection pool : none.

[MASTER]domain1.adminServer>control-connection-pool -id ds1 -server server1 -refresh
Servers that successfully refreshed a connection pool : server1
Servers that failed to refresh a connection pool : none.

[MASTER]domain1.adminServer>control-connection-pool -id ds1 -server server1 -create
Servers that successfully created a connection pool : server1
Servers that failed to create a connection pool : none.
```

### 4.2.12.3. control-cluster-data-source

클러스터 데이터소스를 제어한다. 클러스터 데이터소스에 속한 컴포넌트 데이터소스의 목록을 보여주거나 failback을 수행한다.

- alias

controlcds, ctrlcds

- 사용법

```
control-cluster-data-source -server <server-name>
                             -id <data-source-id>
                             [-failback]
```

- 파라미터

파라미터	설명
-server <server-name>	서버 이름이다.
-id <data-source-id>	클러스터 데이터소스 ID이다.
[-failback]	클러스터 데이터소스에 대하여 failback을 수행한다.

- 예제

```
[MASTER]domain1.adminServer>control-cluster-data-source -server server1 -id cds1
Component data sources of the cluster data source [cds1] : ds1, ds2

[MASTER]domain1.adminServer>control-cluster-data-source -server server1 -id cds1 -failback
The failback for the cluster data source [cds1] succeeded.
```

#### 4.2.12.4. create-connection-pool

JDBC Connection Pool을 생성한다. JCA Connection Pool에 대해서는 동작하지 않는다.

- alias

createcp

- 사용법

```
create-connection-pool -id <connection-pool-id>
                       [-server <server-name>]
```

- 파라미터

파라미터	설명
-id <connection-pool-id>	JDBC Connection Pool ID이다.
[-server <server-name>]	서버 이름이다.

- 예제

```
[MASTER]domain1.adminServer>create-connection-pool -id ds1 -server server1
```

```
Servers that successfully created a connection pool : server1
Servers that failed to create a connection pool : none.
```

#### 4.2.12.5. destroy-connection

Active 상태의 커넥션의 연결을 강제로 끊는다.

애플리케이션에서 커넥션을 요청하여 커넥션이 현재 Active 상태인데 이 커넥션에 문제가 있다고 판단되는 경우 해당 커넥션을 중도에 강제로 파기하기 위해 이 명령어를 사용할 수 있다. 이 명령어는 Active 상태의 커넥션에 대해서만 동작한다. 단, 이 기능은 실제로 물리적 연결을 강제로 끊는 것이므로 사용에 주의를 해야 한다.

- alias

destroyconnection

- 사용법

```
destroy-connection -server <server-name>
                    [-cpid <connection-pool-id> | -cid <connection-id>]
```

- 파라미터

파라미터	설명
-server <server-name>	연결을 강제로 끊고자 하는 커넥션을 가지고 있는 Connection Pool이 있는 서버의 이름이다.
[-cpid <connection-pool-id>]	Connection Pool에 존재하는 모든 커넥션들에 대해 연결 끊기를 시도할 때 입력하는 Connection Pool의 ID이다.
[-cid <connection-id>]	Connection Pool에 존재하는 특정 커넥션의 연결을 강제로 끊을 때 입력하는 Connection ID이다. Connection ID는 <b>connection-pool-info</b> 명령에 -id 옵션을 주어 확인할 수 있다. 한번에 여러 개의 커넥션을 강제로 연결 끊고자 할 땐 여러 개의 Connection ID 값을 콤마(,)로 구분하여 입력한다.

- 예제

```
[MASTER]domain1.adminServer>destroy-connection -server server1 -cpid ds1
Successfully destroyed the connections from the connection pool.

[MASTER]domain1.adminServer>return-connection -server server1 -cid ds1-1,ds1-2
Successfully destroyed the connections from the connection pool.
```

#### 4.2.12.6. disable-connection-pool

JDBC Connection Pool을 비활성화한다. JCA Connection Pool에 대해서는 동작하지 않는다.

- alias

disablecp

- 사용법

```
disable-connection-pool -id <connection-pool-id>
                        [-server <server-name>]
```

- 파라미터

파라미터	설명
-id <connection-pool-id>	JDBC Connection Pool ID이다.
[-server <server-name>]	서버 이름이다.

- 예제

```
[MASTER]domain1.adminServer>disable-connection-pool -id ds1 -server server1
Servers that successfully disabled a connection pool : server1
Servers that failed to disable a connection pool : none.
```

#### 4.2.12.7. enable-connection-pool

JDBC Connection Pool을 활성화한다. JCA Connection Pool에 대해서는 동작하지 않는다.

- alias

enablecp

- 사용법

```
enable-connection-pool -id <connection-pool-id>
                        [-server <server-name>]
```

- 파라미터

파라미터	설명
-id <connection-pool-id>	JDBC Connection Pool ID이다.
[-server <server-name>]	서버 이름이다.

- 예제

```
[MASTER]domain1.adminServer>enable-connection-pool -id ds1 -server server1
```



```
Servers that successfully enabled a connection pool : server1
Servers that failed to enable a connection pool : none.
```

#### 4.2.12.8. refresh-connection-pool

JDBC Connection Pool의 커넥션들을 새로운 커넥션으로 교체한다. JCA Connection Pool에 대해서는 동작하지 않는다.

- alias

refreshcp

- 사용법

```
refresh-connection-pool -id <connection-pool-id >
                        [-server <server-name>]
```

- 파라미터

파라미터	설명
-id <connection-pool-id>	JDBC Connection Pool ID이다.
[-server <server-name>]	서버 이름이다.

- 예제

```
[MASTER]domain1.adminServer>refresh-connection-pool -id ds1 -server server1
Servers that successfully refreshed a connection pool : server1
Servers that failed to refresh a connection pool : none.
```

#### 4.2.12.9. return-connection

Active 상태의 커넥션을 Connection Pool로 반환한다.

애플리케이션에서 커넥션을 close해주지 않았을 때 데이터소스 설정에서 Action On Connection Leak 설정값이 AutoClose로 되어있지 않으면 지속적으로 커넥션이 Active 상태로 남아 있게 되는 Leak이 발생하는데, 이러한 경우에 이 명령어를 사용할 수 있다. 이 명령어는 Active 상태의 커넥션에 대해서만 동작한다. 단, 반환하려는 커넥션이 현재 진행 중인 트랜잭션에 작업 중인 경우 반환되지 않는다. 그리고 명령어가 수행되어 커넥션 반환하는 중에 에러가 발생하면 해당 커넥션은 destroy 된다.

- alias

returnconnection

- 사용법

```
return-connection -server <server-name>
                  [-cpid <connection-pool-id> | -cid <connection-id>]
```

- 파라미터

파라미터	설명
-server <server-name>	연결 반환하고자 하는 커넥션을 가지고 있는 Connection Pool이 있는 서버의 이름이다.
[-cpid <connection-pool-id>]	Connection Pool에 존재하는 모든 커넥션들에 대해 연결 반환을 시도할 때 입력하는 Connection Pool의 ID이다.
[-cid <connection-id>]	Connection Pool에 존재하는 특정 커넥션의 연결을 Connection Pool로 반환할 때 입력하는 Connection ID이다. Connection ID는 <b>connection-pool-info</b> 명령에 -id 옵션을 주어 확인할 수 있다. 한번에 여러 개의 커넥션을 Connection Pool로 연결 반환하고자 할 땐 여러 개의 Connection ID 값을 콤마(,)로 구분하여 입력한다.

- 예제

```
[MASTER]domain1.adminServer>return-connection -server server1 -cpid ds1
Successfully returned the connections to the connection pool.

[MASTER]domain1.adminServer>return-connection -server server1 -cid ds1-1,ds1-2
Successfully returned the connections to the connection pool.
```

#### 4.2.12.10. shrink-connection-pool

JDBC Connection Pool의 커넥션 개수를 설정된 커넥션 최솟값으로 조정한다. JCA Connection Pool에 대해서는 동작하지 않는다.

- alias

shrinkcp

- 사용법

```
shrink-connection-pool -id <connection-pool-id****>
                      [-server <server-name>]
```

- 파라미터

파라미터	설명
-id <connection-pool-id>	JDBC Connection Pool ID이다.
[-server <server-name>]	서버 이름이다.

- 예제

```
[MASTER]domain1.adminServer>shrink-connection-pool -id ds1 -server server1
Servers that successfully shrank a connection pool : server1
Servers that failed to shrink a connection pool : none.
```

## 4.2.13. 트랜잭션 관련 명령어

트랜잭션 관련 명령어의 목록은 다음과 같다.

명령어	설명
<a href="#">modify-transaction-manager</a>	대상 서버의 트랜잭션 매니저 설정을 변경한다.
<a href="#">recover-transactions</a>	대상 서버의 리소스 매니저를 복구한다. 복구할 특정 리소스를 선택할 수 있다.
<a href="#">show-transaction-manager</a>	대상 서버의 트랜잭션 매니저 설정을 확인한다.
<a href="#">transaction-info</a>	대상 서버의 트랜잭션을 상태별로 확인한다.

### 4.2.13.1. modify-transaction-manager

대상 서버의 트랜잭션 매니저 설정을 변경한다.

- alias

modify-tm, modifytm, tmconfig, tmconf

- 사용법

```
modify-transaction-manager <server-name>
    [-at,--activeTimeout <active-timeout>]
    [-pt,--prepareTimeout <prepare-timeout>]
    [-pdt,--preparedTimeout <prepared-timeout>]
    [-ct,--commitTimeout <commit-timeout>]
    [-rt,--recoveryTimeout <recovery-timeout>]
    [-it,--incompleteTimeout <incomplete-timeout>]
    [-d,--txLogDir <transaction-log-directory>]
    [-a,--automaticRecovery <automatic-recovery-enable/disable>]
    [-f,--forceLock]
    [-detail]
```

- 파라미터

파라미터	설명
<server-name>	트랜잭션 설정을 변경할 서버의 이름이다.

파라미터	설명
<code>[-at,--activeTimeout &lt;active-timeout&gt;]</code>	트랜잭션을 시작하고 commit까지의 최대 시간이다. 시간을 초과한 트랜잭션은 강제로 rollback된다.
<code>[-pt,--prepareTimeout &lt;prepare-timeout&gt;]</code>	루트 코디네이터가 prepare를 보내고 하위 코디네이터로부터 그 응답을 기다리는 최대 시간이다.
<code>[-pdt,--preparedTimeout &lt;prepared-timeout&gt;]</code>	하위 코디네이터가 prepare 응답을 보내고 루트 코디네이터의 Global decision을 기다리는 최대 시간이다. 시간을 초과한 경우 루트에 prepare 응답을 다시 보낸다.
<code>[-ct,--commitTimeout &lt;commit-timeout&gt;]</code>	루트 코디네이터가 하위 코디네이터로 commit을 보내고 그 응답을 기다리는 최대 시간이다. 시간을 초과한 경우 incomplete 트랜잭션으로 취급한다.
<code>[-rt,--recoveryTimeout &lt;recovery-timeout&gt;]</code>	recovery 정보를 뽑아내는 데 기다리는 최대 시간이다. 시간을 초과하는 경우 사용자에게 recovery를 맡긴다.
<code>[-it,--incompleteTimeout &lt;incomplete-timeout&gt;]</code>	Incomplete 트랜잭션이 recover되기까지 기다리는 최대 시간이다. 시간을 초과할 경우 Incomplete 트랜잭션이라는 정보를 삭제한다.
<code>[-d,--txLogDir &lt;transaction-log-directory&gt;]</code>	recovery를 위한 트랜잭션 로그를 남기는 디렉터리이다. 자세한 설명은 JEUS Server 안내서의 "복구 관련 로그 파일"을 참고한다.
<code>[-a,--automaticRecovery &lt;automatic-recovery-enable/disable&gt;]</code>	<b>[동적 변경]</b> 서버에 이상이 있을 때 다른 서버가 해당 서버의 Incomplete 트랜잭션을 recovery할지 여부를 설정한다. 자세한 설명은 JEUS Server 안내서의 "트랜잭션 복구"를 참고한다.
<code>[-f,--forceLock]</code>	설정 변경을 강제로 진행하여 반영한다.
<code>[-detail]</code>	결과를 자세하게 조회한다.

## • 예제

### ◦ 현재 설정 확인

```
[MASTER]domain1.adminServer>modify-transaction-manager server1
Shows the current configuration.
transaction of server (server1)
```

```
=====
+-----+-----+
| Active Timeout      | 600000 |
| Prepare Timeout     | 120000 |
| Prepared Timeout    | 60000  |
| Commit Timeout      | 240000 |
| Recovery Timeout     | 120000 |
| Incomplete Timeout  | 86400000 |
| Tx Log Dir          | ${SERVER_HOME}/.workspace/tmlog |
| Automatic Recovery  | false  |
+-----+-----+
=====
```

### ◦ 동적 설정인 트랜잭션 자동 복구 기능 켜기

```
[MASTER]domain1.adminServer>modify-transaction-manager server1 -automaticRecovery true
```

Successfully performed the MODIFY operation for transaction of server (server1).  
Check the results using "show-transaction-manager server1 or modify-transaction-manager server1"

- 서버 재시작으로 반영되는 타임아웃 설정 변경

```
[MASTER]domain1.adminServer>modify-transaction-manager server1 -activeTimeout 20000
Successfully performed the MODIFY operation for transaction of server (server1),
but all changes were non-dynamic. They will be applied after restarting.
Check the results using "show-transaction-manager server1 or modify-transaction-manager server1"
```

- 동적 반영 설정과 서버 재시작으로 반영되는 설정 함께 변경

```
[MASTER]domain1.adminServer>modify-transaction-manager server1 -automaticRecovery true
-activeTimeout 20000
Successfully performed the MODIFY operation for transaction of server (server1),
but some changes were non-dynamic. They will be applied after restarting.
Check the results using "show-transaction-manager server1 or modify-transaction-manager server1"
```

#### 4.2.13.2. recover-transactions

대상 서버의 리소스 매니저를 복구한다. 복구할 리소스를 선택할 수 있다.

- alias

recovertx, tmresync

- 사용법

```
recover-transactions -server <server-name>
                     <resource-type>
                     <export-name>
                     [<connector-module-name>]
```

- 파라미터

파라미터	설명
-server <server-name>	서버의 이름이다.
<resource-type>	복구할 리소스 타입을 지정한다. <ul style="list-style-type: none"><li>• sql : JDBC XADataSource</li><li>• jms : JMS Connection Factory</li><li>• connector : Connector 리소스 어댑터</li></ul>

파라미터	설명
<export-name>	리소스의 <export-name>을 지정한다.
[<connector-module-name>]	리소스가 Connector인 경우에는 해당 Connector Module의 이름을 지정한다.

- 예제

```
[MASTER]domain1.adminServer>recover-transactions -server server1 sql jdbc/derby/XADataSource
The recover-transactions command was executed on the server1 server.
```

### 4.2.13.3. show-transaction-manager

대상 서버의 트랜잭션 매니저 설정을 확인한다.

- alias

show-tm, showtm

- 사용법

```
show-transaction-manager <server-name>
```

- 파라미터

파라미터	설명
<server-name>	서버의 이름이다.

- 예제

```
[MASTER]domain1.adminServer>show-transaction-manager server1
Shows the current configuration.
transaction of server (server1)
=====
+-----+-----+
| Active Timeout      | 600000 |
| Prepare Timeout     | 120000 |
| Prepared Timeout    | 60000  |
| Commit Timeout      | 240000 |
| Recovery Timeout    | 120000 |
| Incomplete Timeout  | 86400000 |
| Tx Log Dir          | ${SERVER_HOME}/.workspace/tmlog |
| Automatic Recovery  | false  |
+-----+-----+
=====
```

#### 4.2.13.4. transaction-info

대상 서버의 트랜잭션을 상태별로 확인한다.

- alias

transactioninfo, txinfo

- 사용법

```
transaction-info -server <server-name>
                 [-i,--inflight]
                 [-d,--indoubt]
                 [-r,--retrying]
```

- 파라미터

파라미터	설명
-server <server-name>	서버의 이름이다.
[-i,--inflight]	진행 중인 트랜잭션을 확인한다. commit 또는 rollback 단계까지 진행되지 않은 메모리 상의 트랜잭션이다.
[-d,--indoubt]	미완료 트랜잭션을 확인한다. 불완전한 상태의 끝나지 않은 것으로 복구가 필요하거나 현재 결정을 받지 못한 트랜잭션이다.
[-r,--retrying]	재시도할 트랜잭션을 확인한다. 리소스 매니저의 문제로 one-phase-commit 또는 two-phase-commit을 제대로 처리하지 못해 JEUS 자체적으로 재시도 중인 트랜잭션이다.

- 예제

```
[MASTER]domain1.adminServer>transaction-info -server server1
<transactions of server1>
```

In-Flight Transaction Information

```
=====
+-----+-----+-----+-----+-----+-----+-----+
|      XID      | Coordina| Extern| Status | Timeout | Elapse | XAResources|
|               | tor     | al XID |        |          |         |             |
+-----+-----+-----+-----+-----+-----+-----+
| 049FE5.00C0A80011 | 192.168. |      | STATUS | 5000(ms) | 7418(ms) | jdbc/derby/ |
| 2608221CC3D6B40000 | 0.17:9736 |      | _MARKED_ |          |          | XADatasource |
| 000000000011.00   |          |      | ROLLBACK |          |          |             |
+-----+-----+-----+-----+-----+-----+-----+
=====
```

In-doubt Transaction Information

```
=====
+---+-----+-----+-----+-----+-----+
| XID | Coordinator | External XID | Status | Decision | Sub-XID |
+---+-----+-----+-----+-----+-----+
(No data available)
```

#### Retrying Transaction Information

```
=====
+---+---+-----+-----+---+---+
|XID| Name|      Failure Count      | Status| Decision|
+---+---+-----+-----+---+---+
(No data available)
=====
```

## 4.2.14. 보안 관련 명령어

본 절에서는 보안과 관련된 명령어에 대해 설명한다. 보안 관련 명령어는 각각 사용자, 그룹 및 Policy 관리 명령어로 분류되고, 목록은 다음과 같다.

- 사용자 관리 명령어

사용자 관리 명령어는 사용자를 추가, 수정, 삭제할 때 사용한다.

명령어	설명
<a href="#">add-user</a>	새로운 사용자를 추가한다.
<a href="#">remove-user</a>	사용자를 삭제한다.
<a href="#">list-user-names</a>	로그인된 모든 사용자 이름 목록과 각 사용자의 Lock 상태와 만료 상태를 조회한다.
<a href="#">show-user</a>	사용자에 대한 정보를 조회한다.
<a href="#">modify-default-password-validator</a>	Default Password Validator를 새로 설정하거나 기존 설정을 변경 또는 삭제한다.
<a href="#">show-default-password-validator</a>	Default Password Validator 설정을 조회한다.
<a href="#">add-custom-password-validator</a>	Custom Password Validator에 새로운 클래스를 등록한다.
<a href="#">remove-custom-password-validator</a>	Custom Password Validator에 등록된 클래스를 삭제한다.
<a href="#">show-custom-password-validator</a>	Custom Password Validator에 등록된 클래스 목록을 조회한다.
<a href="#">show-subject-validation</a>	Subject Validation Service 설정을 조회한다.
<a href="#">set-password</a>	패스워드를 설정한다.
<a href="#">lock-user</a>	특정 사용자에게 Lock을 설정한다.
<a href="#">unlock-user</a>	특정 사용자에게 설정된 Lock을 해제한다.
<a href="#">expire-user</a>	특정 사용자에게 만료시간을 설정한다.
<a href="#">unexpire-user</a>	특정 사용자에게 설정된 만료시간을 해제한다.



- 그룹 관리 명령어

그룹 관리 명령어는 그룹을 추가, 수정, 삭제할 때 사용한다.

명령어	설명
<code>add-group</code>	새로운 그룹을 추가한다.
<code>add-user-to-group</code>	특정 그룹에 사용자를 추가한다.
<code>remove-user-from-group</code>	특정 그룹의 사용자를 삭제한다.
<code>remove-group</code>	그룹을 삭제한다.
<code>list-group-names</code>	그룹의 목록을 조회한다.
<code>show-group</code>	특정 그룹에 대한 정보를 조회한다.
<code>check-group-member</code>	특정 그룹에 사용자 등록 여부를 조회한다.

- Policy 관리 명령어

Policy 명령어는 보안 시스템에서 Policy와 관련된 사항을 관리할 때 사용한다.

명령어	설명
<code>show-policy</code>	특정 Context ID를 포함한 policy를 조회한다.
<code>add-role</code>	새로운 Role을 추가한다.
<code>modify-role</code>	특정 Role의 설정을 변경한다.
<code>remove-role</code>	특정 Role을 삭제한다.
<code>add-resource</code>	새로운 리소스를 추가한다.
<code>modify-resource</code>	특정 리소스의 설정을 변경한다.
<code>remove-resource</code>	특정 리소스를 삭제한다.
<code>assign-role-to-principal</code>	특정 Role에 Principal을 할당한다.
<code>unassign-role-from-principal</code>	특정 Role로부터 Principal을 삭제한다.
<code>assign-resource-to-role</code>	특정 리소스에 Role을 할당한다.
<code>unassign-resource-from-role</code>	특정 리소스로부터 Role을 삭제한다.



본 절에서 소개하는 명령어는 민감한 사안을 다루므로, 이 명령어를 실행하기에 적절한 권한을 가진 사용자로 로그인해야 한다.

#### 4.2.14.1. 공통 적용 사항

다음은 보안 관련 명령어에서 공통적으로 적용되는 옵션에 대한 설명이다.

- `[-domain <domain-name>]` 옵션

- JEUS에서 보안이 적용되는 범위인 보안 도메인을 지정한다(기본값: SYSTEM\_DOMAIN). 도메인에 대한 자세한 내용은 "JEUS Security 안내서"를 참고한다.
- 각 명령어의 옵션 설명에서 위 옵션에 대한 설명은 생략한다.

#### 4.2.14.2. add-custom-password-validator

Custom Password Validator에 새로운 클래스를 추가 등록한다. Password Validation은 JEUS Security 안내서의 "패스워드 보안 설정"을 참고한다.

- alias

add-custom-validator

- 사용법

```
add-custom-password-validator [-class <class-name>]
```

- 파라미터

파라미터	설명
<class-name>	Custom Password Validator로 등록하기 위해 사용자가 DOMAIN_HOME/lib/application에 jar 파일 형태로 넣어놓은 클래스의 이름이다. 해당 클래스는 jeus.util.PasswordValidator 인터페이스를 구현해야 한다.

- 예제

```
[MASTER]domain1.adminServer> add-custom-password-validator -class MyValidator
Custom password validator [MyValidator] is added successfully.
Check the results using show-custom-password-validator.
```

#### 4.2.14.3. add-group

동적으로 보안 시스템에 새로운 그룹을 추가한다. 설정된 그룹 이름은 현재 도메인 내에서 유일해야 한다(list-group-names 명령어로 확인 가능). 그룹 이름은 일련의 호출 과정에서 특정 그룹을 참조하기 위해 사용한다. 즉, 그룹에 대한 ID에 해당된다.

- alias

addgroup

- 사용법

```
add-group <group-name>
        [-domain <domain-name>]
```

`[-f,--forceLock]`

- 파라미터

파라미터	설명
<code>&lt;group-name&gt;</code>	그룹 이름이다.
<code>[-f,--forceLock]</code>	설정 변경을 강제로 진행하여 반영한다.

- 예제

```
[MASTER]domain1.adminServer> add-group group1
The group [group1] has been successfully added.
```

#### 4.2.14.4. add-resource

동적으로 보안 시스템에 새로운 리소스를 추가한다. 주어진 리소스의 이름과 actions의 조합은 현재 도메인과 컨텍스트 내에서 유일해야 한다([show-policy](#) 명령어로 확인 가능).

- alias

addresource

- 사용법

```
add-resource <resource-name>
  <actions>
    [-contextid <context-id>]
    [-classname <class-name>]
    [-role <role-name>]
    [-excluded]
    [-unchecked]
    [-domain <domain-name>]
    [-f,--forceLock]
```

- 파라미터

파라미터	설명
<code>&lt;resource-name&gt;</code>	추가할 리소스의 이름이다.

파라미터	설명
<actions>	<p>리소스에 부여될 Action 목록이다.</p> <p>Actions는 리소스에 대한 Action 목록이며, 각 Action은 공백으로 구분된다. 예를 들어 jeus.server resource의 경우 "boot", "down"과 같은 Action이 있다.</p> <p>-actions 파라미터를 설정한 경우 -classname으로 설정한 클래스에 두 번째 생성자 파라미터로 "actions"를 전달할 때 사용한다. 디폴트 클래스명은 jeus.security.resource.ResourcePermission이고, 이는 "resouce"와 "actions"를 생성자 파라미터로 차례로 받아 들인다.</p>
[-contextid <context-id>]	Policy를 구분하는 컨텍스트 이름이다. contextid는 "default" 이외의 컨텍스트를 지정할 때 사용한다.
[-classname <class-name>]	<p>Permission 클래스 이름이다.</p> <p>-classname으로 설정한 클래스는 java.security.Permission 추상 클래스를 확장한 Java 클래스여야 하며 최소한 하나의 String 타입의 "resource"을 받아 들이는 생성자를 가지고 있어야 한다.</p> <p>Resource Permission에 대한 자세한 정보는 jeus.security.resource.ResourcePermission 클래스를 참고한다. JEUS가 어떤 Resource Permission을 체크하는지에 관한 정보는 JEUS Security 안내서의 "JEUS Server Permissions"를 참고한다.</p>
[-role <role-name>]	해당 리소스를 할당할 Role 목록이다.
[-excluded]	누구도 해당 Permission에 대한 허가를 가질 수 없도록 한다.
[-unchecked]	누구나 해당 Permission에 대한 허가를 가질 수 있도록 한다.
[-f,--forceLock]	설정 변경을 강제로 진행하여 반영한다.

- 예제

```
[MASTER]domain1.adminServer> add-resource resource1 jeus.*
The resource [name=resource1, actions=jeus.*] has been successfully added.
```

#### 4.2.14.5. add-role

동적으로 보안 시스템에 새로운 Role를 추가한다. 주어진 Role 이름은 현재 도메인 내에서 유일해야 한다([show-policy](#) 명령어로 확인 가능).

- alias

addrole

- 사용법

```
add-role <role-name>
    [-principal <principal>]
    [-classname <classname>]
    [-excluded]
    [-unchecked]
    [-actions <actions>]
    [-domain <domain-name>]
    [-f,--forceLock]
```

- 파라미터

파라미터	설명
<role-name>	추가할 Role의 이름이다.
[-principal <principal>]	해당 Role을 할당할 principal을 지정한다.
[-classname <classname>]	Permission 클래스 이름이다.  -classname으로 설정한 클래스는 java.security.Permission 추상 클래스를 확장한 Java 클래스여야 하며 최소한 하나의 String 타입의 "role"을 받아 들이는 생성자를 가지고 있어야 한다.  Role Permission에 대한 자세한 정보는 jeus.security.resource.RolePermission 클래스를 참고한다. JEUS가 어떤 Role Permission을 체크하는지에 관한 정보는 JEUS Security 안내서의 "JEUS Server Permissions"를 참고한다.
[-excluded]	누구도 해당 Permission에 대한 허가를 가질 수 없도록 한다.
[-unchecked]	누구나 해당 Permission에 대한 허가를 가질 수 있도록 한다.
[-actions <actions>]	Role의 action 목록이다.
[-f,--forceLock]	설정 변경을 강제로 진행하여 반영한다.

- 예제

```
[MASTER]domain1.adminServer> add-role role1
The role [role1] has been successfully added.
```

#### 4.2.14.6. add-user

동적으로 보안 시스템에 새로운 사용자를 추가한다. 주어진 사용자 이름은 현재 도메인 내에서 유일해야 한다([list-user-names](#) 명령어로 확인 가능). 사용자 이름은 일련의 호출 과정에서 특정 사용자를 참조하기 위해 사용한다. 즉, 사용자에게 대한 ID에 해당된다.

- alias

adduser

- 사용법

```
add-user <user-name>
        [-domain <domain-name>]
        [-f,--forceLock]
        [<password>]
```

- 파라미터

파라미터	설명
<user-name>	사용자 이름이다.
[-f,--forceLock]	설정 변경을 강제로 진행하여 반영한다.
[<password>]	비밀번호를 지정한다.

- 예제

```
[MASTER]domain1.adminServer> add-user user1
The user [user1] has been successfully added.
```

#### 4.2.14.7. add-user-to-group

동적으로 보안 시스템에 등록된 특정 그룹에 새로운 사용자 멤버를 추가한다.

- alias

addusertogroup

- 사용법

```
add-user-to-group <group-name>
                  <user-name>
                  [-domain <domain-name>]
                  [-f,--forceLock]
```

- 파라미터

파라미터	설명
<group-name>	그룹 이름이다.
<user-name>	사용자 이름이다.
[-f,--forceLock]	설정 변경을 강제로 진행하여 반영한다.

- 예제

```
[MASTER]domain1.adminServer> add-user-to-group group1 user1
The user [user1] has been successfully added to the group [group1].
```

#### 4.2.14.8. assign-resource-to-role

동적으로 Role에 특정 리소스와 Action을 할당한다.

- alias

assignresource, assignres

- 사용법

```
assign-resource-to-role <resource-name>
    <actions>
    <role-name>
    [-contextid <context-id>]
    [-classname <classname>]
    [-domain <domain-name>]
    [-f,--forceLock]
```

- 파라미터

파라미터	설명
<resource-name>	<b>[동적 변경]</b> 리소스 이름이다. 리소스 이름은 보통 "jeus.server"와 같은 리소스를 나타내는 Java 클래스명이다.
<actions>	<b>[동적 변경]</b> 리소스에 부여된 Action 목록이다.  Actions는 리소스에 대한 Action 목록이며, 각 Action은 공백으로 구분된다. 예를 들어 jeus.server resource의 경우 "boot", "down"과 같은 Action이 있다.  actions 파라미터를 설정한 경우 -classname으로 설정한 클래스에 두 번째 생성자 파라미터로 "actions"를 전달할 때 사용한다. 디폴트 클래스명은 jeus.security.resource.ResourcePermission이고, 이는 "resouce"와 "actions"를 생성자 파라미터로 차례로 받아 들인다.
<role-name>	Role 이름이다.
[-contextid <context-id>]	<b>[동적 변경]</b> Policy를 구분하는 컨텍스트 이름이다. contextid는 "default" 이외의 컨텍스트를 지정할 때 사용한다.

파라미터	설명
[-classname <classname>]	<p><b>[동적 변경]</b> Permission 클래스 이름이다.</p> <p>-classname으로 설정한 클래스는 java.security.Permission 추상 클래스를 확장한 Java 클래스여야 하며 최소한 하나의 String 타입의 "resource"를 받아 들이는 생성자를 가지고 있어야 한다.</p> <p>Resource Permission에 대한 자세한 정보는 jeus.security.resource.ResourcePermission 클래스를 참고한다. JEUS가 어떤 Resource Permission을 체크하는지에 관한 정보는 JEUS Security 안내서의 "JEUS Server Permissions"를 참고한다.</p>
[-f,--forceLock]	설정 변경을 강제로 진행하여 반영한다.

- 예제

```
[MASTER]domain1.adminServer> assign-resource-to-role jeus.* deploy-applications deployRole
The resource [name=jeus.*, actions=deploy-applications] is assigned to the role [deployRole].
```

#### 4.2.14.9. assign-role-to-principal

동적으로 특정 Role을 Principal에 할당한다. -classname으로 설정한 클래스는 java.security.Permission 추상 클래스를 확장한 Java 클래스여야 하며 최소한 하나의 String 타입의 "role"을 받아 들이는 생성자를 가지고 있어야 한다. 디폴트 클래스는 jeus.security.resource.RolePermission이다. 단, anonymous의 경우 해당 target role이 unchecked permission을 가지고 있을 경우에만 할당이 가능하다.

- alias

assignrole, assign-role

- 사용법

```
assign-role-to-principal <role-name>
    <principal>
    [-domain <domain-name>]
    [-f,--forceLock]
```

- 파라미터

파라미터	설명
<role-name>	<b>[동적 변경]</b> Role 이름이다.
<principal>	Principal 이름이다.
[-f,--forceLock]	설정 변경을 강제로 진행하여 반영한다.

- 예제



```
[MASTER]domain1.adminServer> assign-role-to-principal AdministratorsRole user1
The role [AdministratorsRole] is assigned to the principal [user1].
```

#### 4.2.14.10. check-group-member

특정 그룹에 해당 사용자가 등록되었는지 여부를 조회한다.

- alias

checkgroupmember, ismember

- 사용법

```
check-group-member <group-name>
                    <user-name>
                    [-domain <domain-name>]
```

- 파라미터

파라미터	설명
<group-name>	그룹 이름이다.
<user-name>	사용자 이름이다.

- 예제

```
[MASTER]domain1.adminServer> check-group-member group1 user1
The user [user1] is a member of the group [group1].
```

#### 4.2.14.11. expire-user

특정 사용자에게 만료시간을 설정한다. SubjectExpirationValidationService가 서버에 제대로 구성되어 있다면, 해당 사용자는 만료시간까지만 유효하며 만료시간 후 로그인 시도는 [unexpire-user](#)가 호출되기 전에 모두 실패로 끝난다. Subject validation은 JEUS Security 안내서의 "SubjectValidationService SPI"를 참고한다.

- alias

expireuser, exuser

- 사용법

```
expire-user <user-name>
            [-domain <domain-name>]
            [-year <year>]
            [-month <month>]
```

```
[-day <day>]
[-hour <hour>]
[-minute <minute>]
```

- 파라미터

파라미터	설명
<user-name>	사용자 이름이다.
[-year <year>]	만료시간의 '연도'이다. yyyy로 입력해야 한다. (기본값: 현재 시간의 연도)
[-month <month>]	만료시간의 '달/월'이다. MM으로 입력해야 한다. (기본값: 현재 시간의 달/월)
[-day <day>]	만료시간의 '날/일'이다. dd로 입력해야 한다. (기본값: 현재 시간의 날/일)
[-hour <hour>]	만료시간의 '시'이다. hh로 입력해야 한다. (기본값: 현재 시간의 시)
[-minute <minute>]	만료시간의 '분'이다. mm으로 입력해야 한다. (기본값: 현재 시간의 분)

- 예제

```
[MASTER]domain1.adminServer> expire-user user1 -year 2020 -month 12 -day 31 -hour 12 -minute 30
The user [user1] is valid only until [2020-12-31 12:30].
```

#### 4.2.14.12. list-group-names

현재 도메인의 모든 그룹에 대한 목록을 조회한다.

- alias

listgroupnames, getgroupnames

- 사용법

```
list-group-names [-domain <domain-name>]
```

- 예제

```
[MASTER]domain1.adminServer> list-group-names
=====
+-----+-----+
|                               Group Name                               |
+-----+-----+
| Administrators                 |
+-----+-----+
| group1                         |
+-----+-----+
```

=====

#### 4.2.14.13. list-user-names

현재 도메인에 로그인된 모든 사용자 이름 목록과 각 사용자의 Lock 상태와 만료 상태를 조회한다.

- alias

listusernames, getusernames

- 사용법

```
list-user-names [-domain <domain-name>]
```

- 예제

```
[MASTER]domain1.adminServer> list-user-names
=====
+-----+-----+-----+
| User Name | Lock State | ExpiryTime State |
+-----+-----+-----+
| jeus      | unlocked  | Unexpired         |
| admin     | locked    | 2020-01-01 12:00  |
+-----+-----+-----+
=====
```

#### 4.2.14.14. lock-user

동적으로 특정 사용자에게 Lock을 설정한다. "SubjectLockoutValidationService"가 서버에 제대로 구성되어 있다면, 해당 사용자에게 대한 모든 로그인 시도는 "unlock-user"가 호출되기 전에 모두 실패로 끝난다.

Subjct validation은 JEUS Security 안내서의 "SubjectValidationService SPI"를 참고한다.

- alias

lockuser

- 사용법

```
lock-user <user-name>
        [-domain <domain-name>]
```

- 파라미터

파라미터	설명
<user-name>	사용자 이름이다.

- 예제

```
[MASTER]domain1.adminServer> lock-user user2
The user [user2] has been locked out.
```

#### 4.2.14.15. modify-default-password-validator

동적으로 Default Password Validator 설정을 추가, 변경 또는 삭제한다. Password Validation은 JEUS Security 안내서의 "패스워드 보안 설정"을 참고한다.

- alias

modify-password-validator

- 사용법

```
modify-default-password-validator [-disable]
    [-min <min-length>]
    [-max <max-length>]
    [-special <true | false>]
    [-digit <true | false>]
    [-capital <true | false>]
    [-small <true | false>]
    [-excludeID <true | false>]
```

- 파라미터

파라미터	설명
[-disable]	<b>[동적 변경]</b> Default Password Validator를 사용 안함으로 설정하고 기존 설정 값들을 모두 삭제한다.
[-min <min-length>]	<b>[동적 변경]</b> 패스워드의 최소 자릿수를 설정한다. (범위: 1 ~ 255, 기본값: 1)
[-max <max-length>]	<b>[동적 변경]</b> 패스워드의 최대 자릿수를 설정한다. (범위: 1 ~ 255, 기본값: 1)
[-special <true   false>]	<b>[동적 변경]</b> 패스워드에 반드시 특수문자가 들어가도록 할지 설정한다. (기본값: false)
[-digit <true   false>]	<b>[동적 변경]</b> 패스워드에 반드시 숫자가 들어가도록 할지 설정한다. (기본값: false)

파라미터	설명
[-capital <true   false>]	<b>[동적 변경]</b> 패스워드에 반드시 대문자가 들어가도록 할지 설정한다. (기본값: false)
[-small <true   false>]	<b>[동적 변경]</b> 패스워드에 반드시 소문자가 들어가도록 할지 설정한다. (기본값: false)
[-excludeID <true   false>]	<b>[동적 변경]</b> 패스워드에 사용자의 ID가 들어가지 못하도록 할지 설정한다. (기본값: false)

- 예제

```
[MASTER]domain1.adminServer> modify-default-password-validator -min 4 -special true
Default password validator is updated successfully.
Check the results using "show-default-password-validator or modify-default-password-validator".
```

#### 4.2.14.16. modify-resource

동적으로 리소스 설정을 변경한다. 다만, Role에 할당된 리소스 설정을 변경하기 위해서는 해당 명령어가 아닌 [assign-resource-to-role](#) 혹은 [unassign-resource-from-role](#) 명령어를 사용해야 한다.

- alias

modifyresource

- 사용법

```
modify-resource <resource-name>
    <actions>
    [-contextid <context-id>]
    [-classname <classname>]
    [-excluded <true | false>]
    [-unchecked <true | false>]
    [-domain <domain-name>]
    [-f,--forceLock]
```

- 파라미터

파라미터	설명
<resource-name>	설정 변경할 리소스의 이름이다.
<actions>	설정 변경할 리소스의 action 목록이다.
[-contextid <context-id>]	Policy를 구분하는 컨텍스트 이름이다. contextid는 "default" 이외의 컨텍스트를 지정할 때 사용한다.

파라미터	설명
<code>[-classname &lt;classname&gt;]</code>	Permission 클래스 이름이다.  -classname으로 설정한 클래스는 java.security.Permission 추상 클래스를 확장한 Java 클래스여야 하며 최소한 하나의 String 타입의 "resource"을 받아 들이는 생성자를 가지고 있어야 한다.  Resource Permission에 대한 자세한 정보는 jeus.security.resource.ResourcePermission 클래스를 참고한다. JEUS가 어떤 Resource Permission을 체크하는지에 관한 정보는 JEUS Security 안내서의 "JEUS Server Permissions"를 참고한다.
<code>[-excluded &lt;true   false&gt;]</code>	누구도 해당 Permission에 대한 허가를 가질 수 없도록 한다. (true false)
<code>[-unchecked &lt;true   false&gt;]</code>	누구나 해당 Permission에 대한 허가를 가질 수 있도록 한다. (true false)
<code>[-f,--forceLock]</code>	설정 변경을 강제로 진행하여 반영한다.

- 예제

```
[MASTER]domain1.adminServer> modify-resource resource1 jeus.* -excluded true -unchecked false
The resource [name=resource1, actions=jeus.*] has been successfully modified.
```

#### 4.2.14.17. modify-role

동적으로 Role 설정을 변경한다. 다만, 사용자에게 할당된 Role 설정을 변경하기 위해서는 해당 명령어가 아닌 [assign-role-to-principal](#) 혹은 [unassign-role-from-principal](#) 명령어를 사용해야 한다.

- alias

modifyrole

- 사용법

```
modify-role <role-name>
    [-classname <classname>]
    [-excluded <true | false>]
    [-unchecked <true | false>]
    [-actions <actions>]
    [-domain <domain-name>]
    [-f,--forceLock]
```

- 파라미터

파라미터	설명
<code>&lt;role-name&gt;</code>	설정 변경할 Role의 이름이다.

파라미터	설명
<code>[-classname &lt;classname&gt;]</code>	Permission 클래스 이름이다.  -classname으로 설정한 클래스는 java.security.Permission 추상 클래스를 확장한 Java 클래스여야 하며 최소한 하나의 String 타입의 "role"을 받아 들이는 생성자를 가지고 있어야 한다.  Role Permission에 대한 자세한 정보는 jeus.security.resource.RolePermission 클래스를 참고한다. JEUS가 어떤 Role Permission을 체크하는지에 관한 정보는 JEUS Security 안내서의 "JEUS Server Permissions"를 참고한다.
<code>[-excluded &lt;true false&gt;]</code>	누구도 해당 Permission에 대한 허가를 가질 수 없도록 한다. (true false)
<code>[-unchecked &lt;true false&gt;]</code>	누구나 해당 Permission에 대한 허가를 가질 수 있도록 한다. (true false)
<code>[-actions &lt;actions&gt;]</code>	Role의 action 목록이다.
<code>[-f,--forceLock]</code>	설정 변경을 강제로 진행하여 반영한다.

- 예제

```
[MASTER]domain1.adminServer> modify-role role1 -excluded true -unchecked false
The role [role1] has been successfully modified.
```

#### 4.2.14.18. remove-custom-password-validator

Custom Password Validator에 등록된 클래스를 삭제한다. Password Validation은 JEUS Security 안내서의 "패스워드 보안 설정"을 참고한다.

- alias

remove-custom-validator

- 사용법

```
remove-custom-password-validator -class <class-name>
```

- 파라미터

파라미터	설명
<code>-class &lt;class-name&gt;</code>	사용자가 DOMAIN_HOME/lib/application에 jar 파일 형태로 넣고 <a href="#">add-custom-password-validator</a> 명령어로 Custom Password Validator에 등록한 클래스의 이름이다.

- 예제

```
[MASTER]domain1.adminServer> remove-custom-password-validator -class MyValidator
Custom password validator [MyValidator] is removed successfully.
Check the results using show-custom-password-validator.
```

#### 4.2.14.19. remove-group

동적으로 현재 도메인에서 이미 존재하고 있는 그룹을 삭제한다. 그룹에 속한 사용자가 존재하지 않을 경우에만 삭제 가능하다.

- alias

removegroup

- 사용법

```
remove-group <group-name>
           [-domain <domain-name>]
           [-f,--forceLock]
```

- 파라미터

파라미터	설명
<group-name>	그룹 이름이다.
[-f,--forceLock]	설정 변경을 강제로 진행하여 반영한다.

- 예제

```
[MASTER]domain1.adminServer> remove-group group1
The group [group1] has been successfully removed.
```

#### 4.2.14.20. remove-resource

동적으로 현재 도메인에 이미 존재하고 있는 리소스를 제거한다.

- alias

rmresource

- 사용법

```
remove-resource <resource-name>
              <actions>
              [-contextid <context-id>]
              [-domain <domain-name>]
```



`[-f,--forceLock]`

- 파라미터

파라미터	설명
<code>&lt;resource-name&gt;</code>	제거할 리소스의 이름이다.
<code>&lt;actions&gt;</code>	제거할 리소스의 action 목록이다.
<code>[-contextid &lt;context-id&gt;]</code>	Policy를 구분하는 컨텍스트 이름이다. contextid는 "default" 이외의 컨텍스트를 지정할 때 사용한다.
<code>[-f,--forceLock]</code>	설정 변경을 강제로 진행하여 반영한다.

- 예제

```
[MASTER]domain1.adminServer> remove-resource resource1 jeus.*  
The resource [name=resource1, actions=jeus.*] has been successfully removed.
```

#### 4.2.14.21. remove-role

동적으로 현재 도메인에 이미 존재하고 있는 Role을 제거한다.

- alias

rmrole

- 사용법

```
remove-role <role-name>  
    [-domain <domain-name>]  
    [-f,--forceLock]
```

- 파라미터

파라미터	설명
<code>&lt;role-name&gt;</code>	제거할 Role의 이름이다.
<code>[-f,--forceLock]</code>	설정 변경을 강제로 진행하여 반영한다.

- 예제

```
[MASTER]domain1.adminServer> remove-role role1  
The role [role1] has been successfully removed.
```

#### 4.2.14.22. remove-user

동적으로 현재 도메인에서 이미 존재하고 있는 사용자를 삭제한다.

- alias

removeuser

- 사용법

```
remove-user <user-name>
           [-domain <domain-name>]
           [-f,--forceLock]
```

- 파라미터

파라미터	설명
<user-name>	사용자 이름이다.
[-f,--forceLock]	설정 변경을 강제로 진행하여 반영한다.

- 예제

```
[MASTER]domain1.adminServer>remove-user user1
The account information about the user [user1] will be permanently removed from accounts.xml. Do
you really want to remove the user [user1]? (y/n)y
The user [user1] has been successfully removed.
```

#### 4.2.14.23. remove-user-from-group

동적으로 현재 도메인에서 존재하는 특정 그룹의 특정 사용자 멤버를 삭제한다.

- alias

removeuserfromgroup

- 사용법

```
remove-user-from-group <group-name>
                       <user-name>
                       [-domain <domain-name>]
                       [-f,--forceLock]
```

- 파라미터

파라미터	설명
<group-name>	그룹 이름이다.
<user-name>	사용자 이름이다.
[-f,--forceLock]	설정 변경을 강제로 진행하여 반영한다.

- 예제

```
[MASTER]domain1.adminServer> remove-user-from-group group1 user1
The user [user1] has been successfully removed from the group [group1].
```

#### 4.2.14.24. set-password

동적으로 주어진 사용자에게 패스워드를 설정한다. 패스워드는 어떤 암호화나 인코딩 없는 일반 문자로 설정해야 한다. - algorithm 옵션을 통해 패스워드를 암호화할 수 있다. 설정 후에는 새 패스워드로 다시 로그인하여야 한다.



이 명령어는 해당 사용자에게 이전에 설정되었던 모든 패스워드를 삭제한다.

- alias

setpassword

- 사용법

```
set-password <user-name>
    <password>
    [-algorithm <algorithm>]
    [-domain <domain-name>]
    [-f,--forceLock]
```

- 파라미터

파라미터	설명
<user-name>	사용자 이름이다.
<password>	<b>[동적 변경]</b> 사용자 패스워드이다.

파라미터	설명
[-algorithm < algorithm>]	<p><b>[동적 변경]</b> 패스워드 암호화에 사용할 알고리즘의 이름이다.</p> <p>다음의 알고리즘을 선택할 수 있다.</p> <ul style="list-style-type: none"> <li>• AES</li> <li>• base64</li> <li>• DES</li> <li>• DESede</li> <li>• Blowfish</li> <li>• SEED</li> <li>• SHA</li> <li>• SSHA</li> </ul>
[-f,--forceLock]	설정 변경을 강제로 진행하여 반영한다.

- 예제

```
[MASTER]domain1.adminServer>set-password user1 pass1 -algorithm AES
※ For accounts without a password, please press enter.
Please enter the current password of user [user1]:
The password is set for [user1].
```

#### 4.2.14.25. show-custom-password-validator

현재의 Default Password Validator 설정을 조회한다. Password Validation은 JEUS Security 안내서의 "패스워드 보안 설정"을 참고한다.

- alias

show-custom-validator

- 사용법

```
show-custom-password-validator
```

- 예제

```
[MASTER]domain1.adminServer> show-custom-password-validator
=====
+-----+
| custom password validator class names |
+-----+
| MyValidator |
+-----+
```

```
=====
```

#### 4.2.14.26. show-default-password-validator

현재의 Default Password Validator 설정을 조회한다. Password Validation은 JEUS Security 안내서의 "패스워드 보안 설정"을 참고한다.

- alias

show-password-validator

- 사용법

```
show-default-password-validator
```

- 예제

```
[MASTER]domain1.adminServer> show-default-password-validator
=====
+-----+-----+
|                property                | value |
+-----+-----+
| min length                             |    4  |
| max length                             |   255 |
| include special characters              | true  |
| include digit characters                | false |
| include capital characters              | false |
| include small characters                | false |
| exclude user id                        | false |
+-----+-----+
=====
```

#### 4.2.14.27. show-subject-validation

현재의 Subject Validation Service 설정을 조회한다. Subject Validation은 JEUS Security 안내서의 "[Subject Validation] 메뉴"를 참고한다.

- alias

ssv

- 사용법

```
show-subject-validation
```

- 예제

```
[MASTER]domain1.adminServer> show-subject-validation
```

```
=====
+-----+-----+
|          Type          |          Class name          |
+-----+-----+
| DefaultSubjectValidation | jeus.security.impl.expiration.SubjectExpirationVa |
|                          | lidationService              |
+-----+-----+
| DefaultSubjectValidation | jeus.security.impl.lockout.SubjectLockoutValidati |
|                          | onService                    |
+-----+-----+
=====
```

#### 4.2.14.28. show-group

주어진 특정 그룹에 대한 자세한 정보를 조회한다.

- alias

showgroup, getgroup

- 사용법

```
show-group <group-name>
          [-domain <domain-name>]
```

- 파라미터

파라미터	설명
<group-name>	그룹 이름이다.

- 예제

```
[MASTER]domain1.adminServer> show-group Administrators
Administrators(members:Principal admin,Principal user1)
```

#### 4.2.14.29. show-policy

특정 Context ID를 포함하고 있는 Policy를 조회한다. 만약 context-id가 설정되지 않았다면, "default"로 설정된다.

- alias

showpolicy, getpolicy

- 사용법

```
show-policy [-contextid <context-id>]
            [-domain <domain-name>]
```

- 파라미터

파라미터	설명
[-contextid <context-id>]	Policy를 구분하는 컨텍스트 이름이다.

- 예제

```
[MASTER]domain1.adminServer> show-policy
_____POLICY_____
ROLE PERMISSIONS:
PERMISSION MAP
-----
Excluded permissions:
(none)

Unchecked permissions:
(jeus.security.resource.RolePermission jndiUser)

Checked permissions:
Permission owner:
Principal user1
Permissions of owner:
(jeus.security.resource.RolePermission AdministratorsRole)

Permission owner:
Principal anonymous
Permissions of owner:
(jeus.security.resource.RolePermission jndiUser)

Unassigned permissions:

RESOURCE PERMISSIONS:
Context id: default
PERMISSION MAP
-----
Excluded permissions:
(none)
Unchecked permissions:
(none)

Checked permissions:
Permission owner:
Role jndiUser
Permissions of owner:
(jeus.security.resource.ResourcePermission jeus.jndi.* lookup)

Permission owner:
Role AdministratorsRole
Permissions of owner:
(jeus.security.resource.ResourcePermission jeus.* *)
```

Unassigned permissions:

#### 4.2.14.30. show-user

주어진 사용자에 대한 자세한 정보를 조회한다.

- alias

showuser, getuser

- 사용법

```
show-user [<user-name>]
          [-domain <domain-name>]
```

- 파라미터

파라미터	설명
<user-name>	사용자 이름이다.

- 예제

```
[MASTER]domain1.adminServer> show-user user1
[SUBJECT]
Description: No description
Domain: SYSTEM_DOMAIN
Main principal: Principal user1
Principals: [Principal user1,Administrators(members:Principal admin,Principal user1)]
Public credentials: []
```

#### 4.2.14.31. unassign-role-from-principal

동적으로 Principal에 할당된 특정 Role을 삭제한다.

- alias

unassignrole, unassign-role

- 사용법

```
unassign-role-from-principal <principal>
                             <role-name>
                             [-domain <domain-name>]
                             [-f,--forceLock]
```



- 파라미터

파라미터	설명
<principal>	Principal 이름이다.
<role-name>	<b>[동적 변경]</b> Role 이름이다.
[-f,--forceLock]	설정 변경을 강제로 진행하여 반영한다.

- 예제

```
[MASTER]domain1.adminServer> unassign-role-from-principal jeus jndiuser
The role [jndiuser] has been successfully unassigned from the principal [jeus].
```

#### 4.2.14.32. unassign-resource-from-role

동적으로 Role로부터 리소스를 삭제할 때 사용한다. Role은 더 이상 해당 리소스에 접근이 불가능하다.

- alias

unassignresource, usassignres

- 사용법

```
unassign-resource-from-role <resource-name>
    <actions>
    <role-name>
    [-contextid <context-id>]
    [-domain <domain-name>]
    [-f,--forceLock]
```

- 파라미터

파라미터	설명
<resource-name>	<b>[동적 변경]</b> 리소스 이름이다.
<actions>	<b>[동적 변경]</b> 리소스의 action 목록이다.
<role-name>	Role 이름이다.
[-contextid <context-id>]	<b>[동적 변경]</b> Policy를 구분하는 컨텍스트 이름이다.
[-f,--forceLock]	설정 변경을 강제로 진행하여 반영한다.

- 예제

```
[MASTER]domain1.adminServer> unassign-resource-from-role jeus.* -actions deploy-applications
deployRole
The resource [name=jeus.*, actions=deploy-applications] has been successfully unassigned from the
```

```
role [deployRole].
```

#### 4.2.14.33. unexpire-user

`expire-user`로 설정했던 ExpiryTime을 해제한다.

- alias

unexpireuser, unexuser

- 사용법

```
unexpire-user <user-name>
               [-domain <domain-name>]
```

- 파라미터

파라미터	설명
<user-name>	사용자 이름이다.

- 예제

```
[MASTER]domain1.adminServer> unexpire-user user2
The expiry time for user [user2] is canceled.
```

#### 4.2.14.34. unlock-user

동적으로 `lock-user`로 설정했던 Lock을 해제한다.

- alias

unlockuser

- 사용법

```
unlock-user <user-name>
            [-domain <domain-name>]
```

- 파라미터

파라미터	설명
<user-name>	사용자 이름이다.

- 예제

```
[MASTER]domain1.adminServer> unlock-user user2
The user [user2] has been successfully unlocked.
```

## 4.2.15. 노드 관리 관련 명령어

본 절에서는 노드 관리와 관련된 명령어에 대해 설명한다. 노드는 하나의 머신에 존재하는 하나의 JEUS 설치본을 의미한다. DAS의 원격 서버 관리를 위한 기능으로 DAS에 접속했을 때만 명령어 사용이 가능하다. 노드에 대한 자세한 설명은 "JEUS Node Manager 안내서"를 참고한다.

노드 관리 관련 명령어의 목록은 다음과 같다.

명령어	설명
<a href="#">add-java-node</a>	동적으로 Java 노드를 추가한다.
<a href="#">add-ssh-node</a>	동적으로 SSH 노드를 추가한다.
<a href="#">apply-patch</a>	해당 노드에 패치를 적용한다.
<a href="#">check-jeus-installed</a>	해당 노드에 JEUS가 설치되어 있는지 확인한다.
<a href="#">check-ssh-node</a>	해당 SSH 노드의 SSH 설정이 제대로 동작하는지 확인한다.
<a href="#">install-jeus</a>	해당 노드에 현재 도메인 구성을 포함하여 JEUS를 설치한다.
<a href="#">list-nodes</a>	노드의 목록을 확인한다.
<a href="#">modify-java-node</a>	저장된 Java 노드를 수정한다.
<a href="#">modify-node</a>	저장된 노드를 수정한다.
<a href="#">modify-ssh-node</a>	저장된 SSH 노드를 수정한다.
<a href="#">patch-info</a>	해당 노드에 적용된 패치정보를 확인한다.
<a href="#">remove-node</a>	저장된 노드를 삭제한다.
<a href="#">remove-patch</a>	해당 노드에 적용된 패치를 삭제한다.
<a href="#">show-node</a>	해당 노드의 설정을 확인한다.
<a href="#">uninstall-jeus</a>	해당 노드에서 JEUS를 삭제한다.

### 4.2.15.1. add-java-node

동적으로 새로운 Java 노드를 추가한다. 각 옵션에 대해 알맞는 값을 입력하여 노드에 설정된 Java NodeManager 접속에 문제가 없도록 한다.

- alias

addjavanode

- 사용법

```
add-java-node <node-name>
    -host <host-name>
    [-port <port-number>]
    [-ssl]
    [-truststore <truststore-path>]
    [-truststorepassword <truststore-password>]
```

- 파라미터

파라미터	설명
<node-name>	추가할 노드의 이름이다. ID로 사용되는 값이므로 유일해야 한다.
-host <host-name>	해당 노드의 노드 매니저에서 사용할 호스트 주소이다.
[-port <port-number>]	해당 노드의 노드 매니저에서 사용할 포트 번호이다. (기본값: 7730)
[-ssl]	SSL을 사용할 지 여부를 설정한다.
[-truststore <truststore-path>]	Truststore file의 경로를 설정한다.
[-truststorepassword <truststore-password>]	Truststore file의 password를 설정한다.

- 예제

```
[MASTER]domain1.adminServer> add-java-node node1 -host 192.168.34.55 -port 7730
The node [node1] was successfully added.
```

## 4.2.15.2. add-ssh-node

동적으로 새로운 SSH 노드를 추가한다. 각 옵션에 대해 알맞는 값을 입력하여 SSH 접속에 문제가 없도록 한다. SSH 접속 가능 여부는 [check-ssh-node](#) 명령어를 통해 가능하며, 생성된 SSH 노드는 [modify-ssh-node](#)를 통해 수정 가능하다.

- alias

addsshnode

- 사용법

```
add-ssh-node <node-name>
    -host <host-name>
    [-dir <install-dir>]
    [-port <port-number>]
```

```
[-user <user-name>]
[-privatekey <file-path>]
```

- 파라미터

파라미터	설명
<node-name>	추가할 원격 노드의 이름이다. ID로 사용되는 값이므로 유일해야 한다.
-host <host-name>	해당 원격 노드의 호스트 주소이다.
[-dir <install-dir>]	해당 원격 노드의 JEUS 설치 디렉터리이다.
[-port <port-number>]	SSH 접속 포트 번호이다. (기본값: 22)
[-user <user-name>]	SSH 접속 사용자명이다.
[-privatekey <file-path>]	SSH 접속을 위한 개인키 파일 경로이다.

- 예제

```
[MASTER]domain1.adminServer> add-ssh-node node1 -host 192.168.34.55 -dir /home/sshUser/jeus -user sshUser
The node [node1] was successfully added.
```

### 4.2.15.3. apply-patch

설정된 노드에 패치 파일을 적용한다.

- alias

applypatch

- 사용법

```
apply-patch [<file-name>]
            [-nodes <node-name>]
            [-rolling]
            [-nomaster]
            [-action <action-on-error>]
```

- 파라미터

파라미터	설명
[<file-name>]	적용할 패치 파일을 설정한다. 설정하지 않은 경우 JEUS_HOME/lib/jext 디렉터리 전체를 전송한다.

파라미터	설명
<code>[-nodes &lt;node-name&gt;]</code>	패치를 적용할 노드 리스트를 설정한다. 설정하지 않은 경우 모든 도메인 내에 존재하는 노드에 적용한다.
<code>[-rolling]</code>	패치 파일 전송 후 서버를 재기동해 적용할지 여부를 설정한다.
<code>[-nomaster]</code>	MASTER는 재기동하지 않는다.
<code>[-action &lt;action-on-error&gt;]</code>	패치 파일 적용에 실패하는 경우 액션에 대해 설정한다.  다음 중에 하나를 설정한다. <ul style="list-style-type: none"> <li>• CONTINUE</li> <li>• ROLLBACK</li> <li>• STOP</li> </ul>

- 예제

```
[MASTER]domain1.adminServer>apply-patch -nodes node2 jext.jar
The patch was successfully applied to the nodes [node1].
```

#### 4.2.15.4. check-jeus-installed

특정 노드에 JEUS가 설치되었는지를 확인한다. 결과값으로 해당 노드의 JEUS 버전 정보가 나온다.

- alias

checkjeusinstalled, isjeusinstalled

- 사용법

```
check-jeus-installed <node-name>
```

- 파라미터

파라미터	설명
<code>&lt;node-name&gt;</code>	노드의 이름이다.

- 예제

```
[MASTER]domain1.adminServer> check-jeus-installed node1
The JEUS version on the node [node1]: JEUS 9.1
```

#### 4.2.15.5. check-ssh-node

특정 SSH 노드에 Java 명령어를 수행해서 SSH 설정이 제대로 되어있는지 확인한다.

- alias

checksshnode, checknode

- 사용법

```
check-ssh-node <node-name>
```

- 파라미터

파라미터	설명
<node-name>	노드의 이름이다.

- 예제

```
[MASTER]domain1.adminServer> check-ssh-node node1  
The Domain Administration Server can execute the "java" process via SSH.
```

#### 4.2.15.6. install-jeus

특정 노드에 현재 도메인의 설정을 포함하여 JEUS를 설치한다. 파일 압축 및 복사 작업이 이루어지므로 환경에 따라 시간이 오래 필요할 수 있다.

- alias

installjeus

- 사용법

```
install-jeus <node-name>
```

- 파라미터

파라미터	설명
<node-name>	노드의 이름이다.

- 예제

```
[MASTER]domain1.adminServer> install-jeus node1  
JEUS was successfully installed on the node [node1].
```

#### 4.2.15.7. list-nodes

현재 구성되어 있는 노드의 목록을 조회한다. 노드의 이름, 종류, 제어 가능 여부, JEUS 설치 버전이 결과값으로 출력된다.

- alias

listnodes, nodelist

- 사용법

```
list-nodes [-onlyname]
```

- 파라미터

파라미터	설명
[-onlyname]	노드의 이름만을 나열하여 조회한다.

- 예제

```
[MASTER]domain1.adminServer> list-nodes
=====
+-----+-----+-----+-----+
| Node name | Type | Under control | JEUS version |
+-----+-----+-----+-----+
| node1     | SSH | Y              | -             |
| node2     | SSH | Y              | JEUS 9.1     |
| node3     | JAVA| N              | -             |
+-----+-----+-----+-----+
=====
```

#### 4.2.15.8. modify-java-node

특정 Java 노드의 설정을 변경한다.

- alias

modifyjavanode

- 사용법

```
modify-java-node <node-name>
                [-host <host-name>]
                [-port <port-number>]
                [-ssl <enable-SSL>]
                [-truststore <truststore-path>]
                [-truststorepassword <truststore-password>]
```



- 파라미터

파라미터	설명
<node-name>	노드의 이름이다.
[-host <host-name>]	해당 노드의 노드 매니저에서 사용할 호스트 주소이다.
[-port <port-number>]	해당 노드의 노드 매니저에서 사용할 포트 번호이다. (기본값: 7736)
[-ssl <enable-SSL>]	SSL을 사용할지 여부를 설정한다.
[-truststore <truststore-path>]	Truststore file의 경로를 설정한다.
[-truststorepassword <truststore-password>]	Truststore file의 password를 설정한다.

- 예제

```
[MASTER]domain1.adminServer> modify-java-node node1 -port 7731
The node [node1] was modified successfully. Check the results using "show-node"
```

#### 4.2.15.9. modify-node

특정 노드의 설정을 변경한다. 노드 매니저 타입을 변경할 수 있다.

- 사용법

```
modify-node <node-name>
    [-type <node-type>]
    [-host <host-name>]
    [-port <port-number>]
    [-ssl <enable-ssl>]
    [-truststore <truststore-path>]
    [-truststorepassword <truststore-password>]
    [-dir <install-dir>]
    [-user <user-name>]
    [-privatekey <file-path>]
```

- 파라미터

파라미터	설명
<node-name>	노드의 이름이다.
[-type <node-type>]	해당 노드의 타입을 변경한다. java 또는 ssh 중 하나의 값을 입력할 수 있다.

파라미터	설명
<code>[-host &lt;host-name&gt;]</code>	해당 노드의 노드 매니저에서 사용할 호스트 주소이다.
<code>[-port &lt;port-number&gt;]</code>	해당 노드의 노드 매니저에서 사용할 포트 번호이다. (기본값: Java 타입의 경우 7730, SSH 타입의 경우 22)
<code>[-ssl &lt;enable-ssl&gt;]</code>	SSL을 사용할지 여부를 설정한다.
<code>[-truststore &lt;truststore-path&gt;]</code>	Truststore file의 경로를 설정한다.
<code>[-truststorepassword &lt;truststore-password&gt;]</code>	Truststore file의 password를 설정한다.
<code>[-dir &lt;install-dir&gt;]</code>	해당 노드에 JEUS가 설치된 경로를 설정한다.
<code>[-user &lt;user-name&gt;]</code>	SSH 접속 사용자명이다.
<code>[-privatekey &lt;file-path&gt;]</code>	SSH 접속을 위한 개인키 파일 경로이다.

- 예제

```
[MASTER]domain1.adminServer> modify-node node1 -type ssh -port 23
The node [node1] was modified successfully. Check the results using "show-node"
```

#### 4.2.15.10. modify-ssh-node

특정 SSH 노드의 설정을 변경한다.

- alias

modifysshnode, modifynode

- 사용법

```
modify-ssh-node <node-name>
    [-host <host-name>]
    [-dir <install-dir>]
    [-port <port-number>]
    [-user <user-name>]
    [-privatekey <file-path>]
```

- 파라미터

파라미터	설명
<node-name>	노드의 이름이다.
[-host <host-name>]	해당 원격 노드의 호스트 주소이다.
[-dir <install-dir>]	해당 원격 노드의 JEUS 설치 디렉터리이다.
[-port <port-number>]	SSH 접속 포트 번호이다. (기본값: 22)
[-user <user-name>]	SSH 접속 사용자명이다.
[-privatekey <file-path>]	SSH 접속을 위한 개인키 파일 경로이다.

- 예제

```
[MASTER]domain1.adminServer> modify-ssh-node node1 -port 23
The node [node1] was modified successfully. Check the results using "show-node"
```

#### 4.2.15.11. patch-info

노드에 적용된 패치를 확인한다.

- alias

patchinfo, patch-list, patchlist, list-patch, listpatch

- 사용법

```
patch-info [<node-names>]
```

- 파라미터

파라미터	설명
<node-names>	노드의 이름이다. 노드가 한 개 이상인 경우는 콤마(,)로 구분한다.

- 예제

```
[MASTER]domain1.adminServer>patch-info
Successfully got patch information for the nodes.
=====
Node Name[node1]

+-----+-----+
|          Patch File Name          | Patch Contents |
+-----+-----+
(No data available)
```

```
Unable to get patch information to SSH NodeManager
```

```
=====
```

```
=====
```

```
Node Name[node2]
```

```
+-----+-----+
|          Patch File Name          | Patch Contents |
+-----+-----+
| jext_patch_test.jar               | patch test    |
+-----+-----+
=====
```

#### 4.2.15.12. remove-node

특정 노드를 삭제한다.

- alias

removenode, rmnode, rm-node

- 사용법

```
remove-node <node-name>
```

- 파라미터

파라미터	설명
<node-name>	노드의 이름이다.

- 예제

```
[MASTER]domain1.adminServer>remove-node node1
The node [node1] was successfully removed.
```

#### 4.2.15.13. remove-patch

설정된 노드에 적용된 패치 파일을 삭제한다.

- alias

removepatch, rmpatch, rm-patch

- 사용법

```
remove-patch [<file-name>]
              [-nodes <node-name>]
```

```
[-rolling]
[-nomaster]
[-action <action-on-error>]
```

#### • 파라미터

파라미터	설명
[<file-name>]	삭제할 패치 파일을 설정한다. 설정하지 않은 경우 JEUS_HOME/lib/jext 디렉터리 전체를 삭제한다.
[-nodes <node-name>]	패치를 삭제할 노드 리스트를 설정한다. 설정하지 않은 경우 모든 도메인 내에 존재하는 노드에서 삭제한다.
[-rolling]	패치 파일 삭제 후 서버를 재기동할 때 삭제 내용을 적용할지 여부를 설정한다.
[-nomaster]	MASTER는 재기동하지 않는다.
[-action <action-on-error>]	<p>패치 파일가 삭제 실패하는 경우의 액션에 대해 설정한다.</p> <p>다음 중에 하나를 설정한다.</p> <ul style="list-style-type: none"> <li>• CONTINUE</li> <li>• STOP</li> <li>• ROLLBACK</li> </ul>

#### • 예제

```
[MASTER]domain1.adminServer>remove-patch -nodes node2 jext.jar
The nodes [node2] have successfully removed patch file.
```

### 4.2.15.14. show-node

특정 노드의 정보를 조회한다. 노드의 기본 정보 외에 해당 노드에 매핑된 서버의 정보도 조회한다.

기본 SSH 노드 정보는 [modify-ssh-node](#) 명령어를 통해 변경 가능하다. 매핑된 서버의 정보는 [modify-server](#) 명령어를 통해 변경 가능하다.

#### • alias

shownode, nodeinfo

#### • 사용법

```
show-node <node-name>
```

#### • 파라미터

파라미터	설명
<node-name>	노드의 이름이다.

- 예제

```
[MASTER]domain1.adminServer> show-node node1
=====
+-----+-----+
| Property | Value |
+-----+-----+
| Node Name | node1 |
| Host      | 192.168.34.55 |
| Mapped Servers | adminServer server1 |
| Node Type | JAVA |
| NodeManager Port | 7730 |
| Use SSL   | false |
+-----+-----+
=====
```

#### 4.2.15.15. uninstall-jeus

특정 노드에 설치된 JEUS를 삭제한다.

- alias

uninstalljeus

- 사용법

```
uninstall-jeus <node-name>
```

- 파라미터

파라미터	설명
<node-name>	노드의 이름이다.

- 예제

```
[MASTER]domain1.adminServer> uninstall-jeus node1
JEUS was successfully uninstalled from the node [node1].
```

#### 4.2.16. 노드 매니저 관련 명령어

본 절에서는 노드 매니저에 접속해서 사용할 수 있는 명령어에 대해 설명한다. 노드는 하나의 머신에 존재하는 하나의 JEUS 설치본을 의미한다. Java 타입의 노드 매니저에 접속해서 서버를 시작시키거나 상태를 확인할 수 있다. Java

타입의 노드 매니저에 대한 자세한 설명은 "JEUS Node Manager 안내서"를 참고한다.

노드 관리 관련 명령어의 목록은 다음과 같다.

명령어	설명
<code>connect-nodemanager</code>	노드 매니저에 접속한다.
<code>disconnect-nodemanager</code>	노드 매니저와 접속을 끊는다. 노드 매니저와 접속한 상태에서만 사용 가능한 명령어이다.
<code>stop-nodemanager</code>	노드 매니저를 종료한다. 노드 매니저와 접속한 상태에서만 사용 가능한 명령어이다.
<code>nm-start-server</code>	노드 매니저를 통해 서버를 시작한다. 노드 매니저와 접속한 상태에서만 사용 가능한 명령어이다.
<code>nm-stop-server</code>	노드 매니저를 통해 시작한 서버를 종료한다. 노드 매니저와 접속한 상태에서만 사용 가능한 명령어이다.
<code>nm-state-server</code>	노드 매니저를 통해 기동한 서버의 상태를 확인한다. 노드 매니저와 접속한 상태에서만 사용 가능한 명령어이다.

#### 4.2.16.1. connect-nodemanager

노드 매니저에 접속한다.

- alias

`connect-nm`, `connectnm`, `nm-connect`, `nmconnect`

- 사용법

```
connect-nodemanager -domain <domain-name>
                    -h,--host <host-address>
                    -p,--port <port>
                    [-t,--type <connect-type>]
                    [-ts,--truststoreFile <truststore-file>]
                    [-tsp,--truststorePass <truststore-password>]
```

- 파라미터

파라미터	설명
<code>-domain &lt;domain-name&gt;</code>	노드 매니저에서 다룰 도메인 이름이다.
<code>-h,--host &lt;host-address&gt;</code>	접속하려고 하는 노드 매니저의 호스트 주소이다.
<code>-p,--port &lt;port&gt;</code>	접속하려고 하는 노드 매니저의 포트 번호이다.

파라미터	설명
<code>[-t,--type &lt;connect-type&gt;]</code>	<p>노드 매니저에 접속할 때 SSL을 사용할지 여부를 설정한다.</p> <p>다음 중에 하나를 설정한다.</p> <ul style="list-style-type: none"> <li>• ssl</li> <li>• plain(기본값)</li> </ul>
<code>[-ts,--truststoreFile &lt;truststore-file&gt;]</code>	SSL을 사용할 때 Truststore file 경로를 설정한다.
<code>[-tsp,--truststorePass &lt;truststore-password&gt;]</code>	SSL을 사용할 때 Truststore file의 password를 설정한다.

- 예제

```
offline>connect-nodemanager -host 192.168.34.55 -port 7730 -domain domain1
The connection to the node manager domain1 has been established.
```

- 참고

본 명령어는 서버에 접속하지 않은 상태에서도 사용할 수 있다.

#### 4.2.16.2. disconnect-nodemanager

노드 매니저와 접속을 끊는다.

- alias

disconnect-nm, disconnectnm, disconnm, nm-disconnect, nmdisconnect

- 사용법

```
disconnect-nodemanager
```

- 예제

```
[NodeManager]domain1>disconnect-nodemanager
disconnect to node manager.
```

- 참고

본 명령어는 서버에 접속하지 않은 상태에서도 사용할 수 있다.



### 4.2.16.3. stop-nodemanager

노드 매니저를 종료한다.

- alias

stop-nm, stopnm, nm-stop, nmstop

- 사용법

```
stop-nodemanager [-verbose]
                  [-p --properties <properties-file>]
                  [-host <host-address>]
                  [-port <port>]
                  [-t, --type <connect-type>]
                  [-ts, --truststoreFile <truststore-file>]
                  [-tsp, --truststorePass <truststore-password>]
                  [-stopServer]
                  [-f]
                  [-g]
                  [-to <shutdowntimeout>]
```

- 파라미터

파라미터	설명
[-verbose]	실행 결과를 상세하게 출력한다.
[-p --properties <properties-file>]	커맨드 실행에 필요한 설정값들을 담은 파일을 설정한다.
[-host <host-address>]	종료하려고 하는 노드 매니저의 호스트 정보이다.
[-port <port>]	종료하려고 하는 노드 매니저의 포트 정보이다.
[-t,--type <connect-type>]	노드 매니저에 접속할 때 SSL을 사용할지 여부를 설정한다.  다음 중에 하나를 설정한다. <ul style="list-style-type: none"><li>• ssl</li><li>• plain(기본값)</li></ul>
[-ts,--truststoreFile <truststore-file>]	SSL을 사용할 때 Truststore file 경로를 설정한다.
[-tsp,--truststorePass <truststore-password>]	SSL을 사용할 때 Truststore file의 password를 설정한다.
[-stopServer]	노드 매니저가 관리하고 있는 서버도 같이 종료하도록 한다.
[-f]	Graceful shutdown을 시도하지 않고 바로 서버를 종료한다. (기본값)
[-g]	Graceful하게 종료할지 여부를 설정한다. 처리 중인 요청이 끝나길 무한히 기다린다.

파라미터	설명
[-to,--shutdowntimeout]	Graceful하게 종료할 때 사용할 타임아웃을 설정한다. 처리 중인 요청을 입력한 시간만큼 완료되길 기다린다. (단위: second)

- 예제

```
[NodeManager]domain1>stop-nodemanager
Succeed to stop the node manager.
```

- 참고

본 명령어는 서버에 접속하지 않은 상태에서도 사용할 수 있다.

#### 4.2.16.4. nm-start-server

노드 매니저를 통해 서버를 시작한다.

- alias

nmstart, nmstartserver

- 사용법

```
nm-start-server [-domain <domain-name>]
                 -server <server-name>
                 -u,--user <user-name>
                 -p,--password <password>
                 [-masterurl <masterurl>]
                 [-f,--force]
                 [-s,--standby]
                 [-host <nm-host>]
                 [-port <nm-port>]
                 [-t,--type <connect-type>]
                 [-ts,--truststoreFile <truststore-file>]
                 [-tsp,--truststorePass <truststore-password>]
```

- 파라미터

파라미터	설명
[-domain <domain-name>]	시작할 서버가 속한 도메인 이름이다.
-server <server-name>	시작할 서버의 이름이다.
-u,--user <user-name>	서버를 기동할 때 필요한 계정정보이다.

파라미터	설명
-p,--password <password>	서버를 기동할 때 필요한 계정의 password이다.
[-masterurl <masterurl>]	MS를 시작할 때 필요한 MASTER의 URL정보이다.
[-f, --force]	MS를 시작할 때 애플리케이션 디플로이가 실패하더라도 RUNNING으로 만들기 위한 설정이다.
[-s, --standby]	MS를 시작할 때 STANDBY 상태까지만 만들기 위한 설정이다.
[-host <nm-host>]	노드 매니저에 접속이 되지 않은 상태라면 노드 매니저의 호스트 정보를 옵션으로 주어 노드 매니저에 접속한 후 서버를 시작할 수 있다.
[-port <nm-port>]	노드 매니저에 접속이 되지 않은 상태라면 노드 매니저의 포트 정보를 옵션으로 주어 노드 매니저에 접속한 후 서버를 시작할 수 있다.
[-t --type <connect-type>]	노드 매니저에 접속할 때 SSL을 사용할지 여부를 설정한다.  다음 중에 하나를 설정한다. <ul style="list-style-type: none"> <li>• ssl</li> <li>• plain(기본값)</li> </ul>
[-ts,--truststoreFile <truststore-file>]	SSL을 사용할 때 Truststore file 경로를 설정한다.
[-tsp,--truststorePass <truststore-password>]	SSL을 사용할 때 Truststore file의 password를 설정한다.

#### • 예제

```
[NodeManager]domain1>nm-start-server -server server1 -u jeus -p jeus
succeed to start server[server1].
RUNNING
```

#### • 참고

본 명령어는 서버에 접속하지 않은 상태에서도 사용할 수 있다.

### 4.2.16.5. nm-stop-server

노드 매니저를 통해 시작한 서버의 상태를 확인한다.

#### • alias

nmstop-server, nmstopserver

#### • 사용법

```
nm-stop-server [-domain <domain-name>]
               -server <server-name>
               -u,--user <user-name>
               -p,--password <password>
               [-to,--timeout <to>]
               [-g,--graceful]
```

#### • 파라미터

파라미터	설명
[-domain <domain-name>]	down시킬 서버가 속한 도메인 이름이다.
-server <server-name>	down시킬 서버의 이름이다.
-u,--user <user-name>	서버를 down시키기 위해 서버에 접속하는 경우 계정 정보이다.
-p,--password <password>	서버를 down시키기 위해 서버에 접속하는 경우 계정의 password이다.
[-to,--timeout <to>]	Graceful하게 종료할 때 사용할 타임아웃을 설정한다. 처리 중인 요청을 입력한 시간만큼 완료되길 기다린다. (단위: second)
[-g, --graceful]	Graceful하게 종료할지 여부를 설정한다. 처리 중인 요청이 끝나길 무한히 기다린다.

#### • 예제

```
[NodeManager]domain1>nm-stop-server -server server1 -u jeus -p jeus
succeed to stop server[server1].
```

#### • 참고

본 명령어는 서버에 접속하지 않은 상태에서도 사용할 수 있다.

### 4.2.16.6. nm-state-server

노드 매니저를 통해 시작한 서버의 상태를 확인한다.

#### • alias

nmstate-server, nmstateserver

#### • 사용법

```
nm-state-server [-domain <domain-name>]
               -server <server-name>
               -u,--user <user-name>
```

```
-p,--password <password>
```

- 파라미터

파라미터	설명
[ <i>-domain &lt;domain-name&gt;</i> ]	상태를 조회할 서버가 속한 도메인 이름이다.
<i>-server &lt;server-name&gt;</i>	상태를 조회할 서버의 이름이다.
<i>-u,--user &lt;user-name&gt;</i>	서버의 상태를 조회하기 위해 서버에 접속하는 경우 계정 정보이다.
<i>-p,--password &lt;password&gt;</i>	서버의 상태를 조회하기 위해 서버에 접속하는 경우 계정의 password이다.

- 예제

```
[NodeManager]domain1>nm-state-server -server server1 -u jeus -p jeus
server[server1] : RUNNING
```

- 참고

본 명령어는 서버에 접속하지 않은 상태에서도 사용할 수 있다.

## 4.2.17. 설정 편집 명령어

본 절에서는 JEUS의 설정을 편집하는 명령어들에 대해 설명한다. 이 명령어들을 사용하면 콘솔을 통해 JEUS의 동적 설정 변경 기능을 이용할 수 있다.



동적 설정 변경 기능에 대한 기본적인 사항은 JEUS Domain 안내서의 "도메인 설정변경"을 참고한다.

다음은 개별 명령어를 설명하기 전에 먼저 알아두어야 할 사항들에 대한 설명이다.

- 설정 편집 명령어로 편집하는 JEUS 설정에 대해

설정 편집 명령어를 사용해서 편집하는 대상은 JEUS 서버가 설정을 저장하기 위해 사용하는 java 객체로, 이 객체는 JEUS 설정을 표현하는 domain.xml을 정의한 XML schema를 JAXB를 사용하여 자바 객체 형태로 변환한 것이다. 그렇기 때문에 객체의 구조나 각 항목의 이름등이 domain.xml의 그것과 유사하나, 몇몇 부분들에 대해서는 다르게 표현될 수도 있다.

- Simple Type

설정 항목 중 자식 항목을 갖지 않는 것을 말한다.

- Complex Type

자식 항목을 갖는 설정 항목을 complex type이라고 한다. 자식으로는 simple type이나 complex type, simple type을 포함하는 목록이나 complex type을 포함하는 목록을 가질 수 있다.

- 목록(List)

simple type이나 complex type을 원소로 갖는 목록(List) 형태의 설정 항목도 존재한다.

- 경로(Path)

설정 편집 명령어에서는 JEUS 설정에서 각 항목의 위치를 경로로 표현한다. 경로는 설정 항목의 이름이나 complex type의 식별자, 목록에서 대상의 위치를 표시하기 위한 색인 번호(index)로 구성된다. 각 항목은 슬래시(/)로 구분한다.

1. 최상위 경로(root)는 슬래시 하나로 표현한다.
2. 현재 위치는 점(dot) 1개로 표현한다.
3. 현재 위치의 부모 항목은 점(dot) 2개로 표현한다. 예를 들어 '/servers'항목의 부모에 대한 경로명은 '/servers/..'으로 표현한다.
4. 대부분의 경우 설정 항목의 이름이 해당 항목의 경로명이 된다. 예를 들어 최상위 경로인 root 아래에 있는 servers라는 항목에 대한 경로는 '/servers'로 표현한다. servers 항목이 자식 항목들을 가지고 있고, 그 항목 중 하나의 이름이 server라고 하면, server 항목을 가리키는 경로는 '/servers/server'이다.
5. complex type을 원소로 갖는 목록의 경우 해당 complex type의 식별자의 값을 경로명으로 사용한다. 여기서 식별자란, 여러 개의 동일한 형식의 complex type이 존재할 때 특정 complex type을 식별할 수 있도록 해주는 각각 고유한 값을 갖는 자식 항목을 말한다. 예를 들어 '/servers/server'는 domain 설정에서 각 서버들의 설정을 담은 여러 개의 complex type으로 이루어진 목록이다. 서버 설정을 담은 complex type에서 식별자는 서버 이름을 사용하도록 지정되어 있다. 따라서 server1에 대한 설정 항목에 대한 경로명은 '/servers/server/server1'로 표현할 수 있다.
6. simple type을 원소로 갖는 목록의 경우 complex type과 달리 식별자가 존재할 수 없기 때문에 해당 목록에서 몇 번째 원소인지를 나타내는 색인 번호(index)를 경로명으로 사용한다. 색인 번호는 대괄호 사이에 0 이상의 정수가 들어간 형태로 지정한다. 예를 들어 '/a/b'라는 경로명이 있고, b가 simple type으로 이루어진 목록이라고 하면 해당 목록의 첫 번째 원소는 '/a/b/[0]'으로 표현한다.



simple type 목록에 대한 경로를 표기할 때 첫 번째 원소는 [1]이 아니라 [0]으로 표기하는 점을 유의하도록 한다.

7. 각 설정 편집 명령어는 현재 경로를 지정하거나 수정하는 경우가 존재한다. 현재 경로는 서버에 저장되기 때문에 만약 jeusadmin을 여러 개 실행하여 동시에 여러 곳에서 편집을 수행할 경우 현재 경로가 의도하지 않게 변경되어 명령어가 오동작할 수 있다. 따라서 설정 편집 시에는 가급적이면 하나의 jeusadmin만 사용하는 것을 권장한다.

아래는 설정 편집을 위해 사용하는 명령어들을 목록으로 나열한 것이다. 모든 명령어는 서버에 접속한 상태에서만 사용할 수 있다.

명령어	설명
<a href="#">activate-configuration</a>	수정한 설정을 서버에 반영하고 결과를 출력한다.

명령어	설명
cancel	동적 설정 변경에 대한 Lock을 해제하고, 서버에 저장되어 있는 수정 내역을 삭제한다. 이 명령어는 Lock을 갖고 있지 않은 상태에서도 사용할 수 있다.
change-current-path	현재 경로를 변경하거나 확인한다.
create-empty-element	지정한 항목에 빈 complex type을 생성하거나, complex type을 원소로 갖는 목록에 새로운 항목을 추가한다.
delete-element	지정한 경로에 위치한 항목을 삭제한다.
set-element-value	지정한 simple type에 값을 변경하거나, simple type을 원소로 갖는 목록에 새로운 항목을 추가한다.
start-configuration-editing	서버에 요청을 보내 동적 설정 변경에 대한 Lock을 요청한다.
show-element	지정한 경로에 위치한 항목에 대한 정보를 출력한다.

#### 4.2.17.1. activate-configuration

수정한 내용을 서버에 반영시키고 결과를 출력한다.

- alias

activate

- 사용법

```
activate-configuration
```

- 예제

```
[MASTER]domain1.adminServer>cd /

[MASTER]domain1.adminServer>ls
description                Not specified (string      )
productionMode              true
id                           982883233
securityManager              complex type
passwordValidator            Not specified (complex type)
adminServerName              adminServer
domainLogHome                Not specified (string      )
systemClusteringFramework   complex type
domainBackup                 complex type
enableWebadmin               true
enableJsonCommand            false
enableToResynchronizeApplications false
servers                      complex type
sessionCluster               complex type
clusters                     Not specified (complex type)
serverTemplates              Not specified (complex type)
lifecycleInvocation           list of [lifecycleInvocation]
```

```

applicationRepositories      Not specified (complex type)
deployedApplications         Not specified (complex type)
deployedLibraries            Not specified (complex type)
resources                    Not specified (complex type)

[MASTER]domain1.adminServer>set description "Hello, world"
The value Hello, world is set at the specified location /description.
[MASTER]domain1.adminServer>ls
description                  Hello, world
productionMode               true
id                           982883233
securityManager              complex type
passwordValidator            Not specified (complex type)
adminServerName              adminServer
domainLogHome                Not specified (string      )
systemClusteringFramework   complex type
domainBackup                 complex type
enableWebadmin               true
enableJsonCommand            false
enableToResynchronizeApplications false
servers                      complex type
sessionCluster               complex type
clusters                     Not specified (complex type)
serverTemplates              Not specified (complex type)
lifecycleInvocation          list of [lifecycleInvocation]
applicationRepositories      Not specified (complex type)
deployedApplications         Not specified (complex type)
deployedLibraries            Not specified (complex type)
resources                    Not specified (complex type)

[MASTER]domain1.adminServer>activate
The activation completed successfully.

[Details]
domain.xml : ACTIVATED
domain : ACTIVATED
domain.description : ACTIVATED

```

#### 4.2.17.2. cancel

수정한 내용을 삭제하고, 동적 설정 변경을 위한 Lock을 해제한다. 명령어를 수행하는 유저가 Lock을 갖고 있지 않아도 사용할 수 있다. 이에 동적 설정 변경을 위한 Lock을 강제로 해제하는 용도로도 사용할 수 있는 명령어이다.

- 사용법

```
cancel
```

- 예제

```

[MASTER]domain1.adminServer>edit
Successfully acquired a configuration lock from Domain Administration Server.

[MASTER]domain1.adminServer>ls

```



```

description                Hello, world
productionMode              true
id                          982883233
securityManager             complex type
passwordValidator           Not specified (complex type)
adminServerName            adminServer
domainLogHome              Not specified (string      )
systemClusteringFramework complex type
domainBackup               complex type
enableWebadmin             true
enableJsonCommand          false
enableToResynchronizeApplications false
servers                   complex type
sessionCluster            complex type
clusters                 Not specified (complex type)
serverTemplates           Not specified (complex type)
lifecycleInvocation       list of [lifecycleInvocation]
applicationRepositories    Not specified (complex type)
deployedApplications      Not specified (complex type)
deployedLibraries         Not specified (complex type)
resources                 Not specified (complex type)

```

```

[MASTER]domain1.adminServer>cancel
Current configuration editing status is cleared, and the configuration lock is released.

```

```

[MASTER]domain1.adminServer>ls
Current user does not have the configuration lock.

```

### 4.2.17.3. change-current-path

지정한 경로를 현재 경로로 변경한다. 변경할 경로를 지정하지 않은 경우 현재 경로를 표시한다.

- alias

cd

- 사용법

```
change-current-path [<path>]
```

- 파라미터

파라미터	설명
[<path>]	변경하고자 하는 경로를 지정한다. 경로 지정은 상대 경로나 절대 경로 방식을 모두 사용할 수 있다.

- 예제

```

[MASTER]domain1.adminServer>change-current-path servers/server/adminServer
Current path is changed to /servers/server/adminServer

```

```
[MASTER]domain1.adminServer>change-current-path ..
Current path is changed to /servers/server

[MASTER]domain1.adminServer>change-current-path .
Current path is changed to /servers/server

[MASTER]domain1.adminServer>change-current-path /
Current path is changed to /

[MASTER]domain1.adminServer>change-current-path servers/server/server1/../../../../server/adminServer
Current path is changed to /servers/server/adminServer

[MASTER]domain1.adminServer>cd /servers/server/adminServer
```

#### 4.2.17.4. create-empty-element

지정한 위치에 새로운 complex type을 생성한다. 생성되는 complex type은 내용이 비어있는 상태로 생성되며, 추가적인 작업을 통해 원하는 값을 채워주어야 한다. complex type을 원소로 갖는 목록인 경우 식별자 값을 항목 이름으로 사용하면 지정한 값을 식별자의 값으로 하는 complex type을 생성하여 목록에 추가한다.

- alias

create

- 사용법

```
create-empty-element <name>
                    [<type>]
```

- 파라미터

파라미터	설명
<name>	생성할 항목의 이름을 지정한다. complex type을 원소로 갖는 목록에 새 complex type을 추가하고자 하는 경우 항목 이름으로 지정한 값을 식별자의 값으로 하는 complex type을 생성한다.
[<type>]	생성할 항목의 형식을 지정한다. 일반적인 경우는 지정하지 않아도 상관없으나, 여러 개의 형식을 원소로 갖는 목록인 경우 가능한 형식 중 생성할 형식을 지정해 주어야 할 필요가 있다.

- 예제

- 빈 complex type을 생성하는 예제이다.

```
[MASTER]domain1.adminServer>cd /
[MASTER]domain1.adminServer>create-empty-element passwordValidator
Created an empty complex type of passwordValidator at path /
```

- complex type을 원소로 갖는 목록에 새로운 complex type을 추가하는 예제이다. 'server2'를 식별자인 서버 이름으로 사용하는 새로운 server 항목이 생성되어 목록에 추가되어 있는 것을 확인할 수 있다.

```
[MASTER]domain1.adminServer>cd /servers/server
Current path is changed to /servers/server

[MASTER]domain1.adminServer>ls
  adminServer
  server1

Current list can contain the following type of elements:
[server]
The name of the identifier element of [server] element is [name].

[MASTER]domain1.adminServer>create-empty-element server2
Added a new element of type server with id [name == server2] to the list at path
/servers/server

[MASTER]domain1.adminServer>ls
  adminServer
  server1
  server2

Current list can contain the following type of elements:
[server]
The name of the identifier element of [server] element is [name].
```

- type 옵션을 사용하여 생성할 complex type 형식을 지정해주는 예제이다. show-element 명령어 결과에 현재 목록이 원소로 여러 형식의 항목을 갖는 것을 확인할 수 있다. 형식을 지정하지 않은 경우 안내 메시지가 출력되는 것도 볼 수 있다.

```
[MASTER]domain1.adminServer>cd /servers/server/adminServer/systemLogging/jeus/handler/
fileHandlerOrSmtphandlerOrSocketHandler
[MASTER]domain1.adminServer>ls
  fileHandler

Current list can contain the following type of elements:
  [fileHandler, userHandler, smtpHandler, socketHandler]
The name of the identifier element of [fileHandler, userHandler,
smtpHandler, socketHandler] element is [name].

[MASTER]domain1.adminServer>create-empty-element handler1
Must specify an element type to create a new element. Available types are fileHandler,
userHandler, smtpHandler, socketHandler.

[MASTER]domain1.adminServer>create-empty-element handler1 userHandler
Added a new element of type userHandler with id [name == handler1] to the list at path
/servers/server/adminServer/systemLogging/jeus/handler/fileHandlerOrSmtphandlerOrSocketHandle
r

[MASTER]domain1.adminServer>ls fileHandler handler1

Current list can contain the following type of elements:
  [fileHandler, userHandler, smtpHandler, socketHandler]
The name of the identifier element of [fileHandler, userHandler,
```

smtpHandler, socketHandler] element is [name].

#### 4.2.17.5. delete-element

지정한 경로에 위치한 항목을 삭제한다.

- alias

delete, rm

- 사용법

```
delete-element <path>
```

- 파라미터

파라미터	설명
<path>	값을 삭제할 항목의 경로를 지정한다.

- 예제

```
[MASTER]domain1.adminServer>cd /
Current path is changed to /

[MASTER]domain1.adminServer>ls
description                Hello,world
productionMode              true
id                           982883233
securityManager              complex type
passwordValidator            Not specified (complex type)
adminServerName              adminServer
domainLogHome                Not specified (string      )
systemClusteringFramework   complex type
domainBackup                 complex type
enableWebadmin               true
enableJsonCommand            false
enableToResynchronizeApplications false
servers                      complex type
sessionCluster               complex type
clusters                     Not specified (complex type)
serverTemplates              Not specified (complex type)
lifecycleInvocation           list of [lifecycleInvocation]
applicationRepositories       Not specified (complex type)
deployedApplications          Not specified (complex type)
deployedLibraries             Not specified (complex type)
resources                    Not specified (complex type)

[MASTER]domain1.adminServer>delete-element description
Successfully delete the target element at /description.

[MASTER]domain1.adminServer>ls
```

```

description                Not specified (string      )
productionMode              true
id                          982883233
securityManager             complex type
passwordValidator           Not specified (complex type)
adminServerName            adminServer
domainLogHome              Not specified (string      )
systemClusteringFramework complex type
domainBackup               complex type
enableWebadmin             true
enableJsonCommand          false
enableToResynchronizeApplications false
servers                    complex type
sessionCluster             complex type
clusters                  Not specified (complex type)
serverTemplates            Not specified (complex type)
lifecycleInvocation        list of [lifecycleInvocation]
applicationRepositories    Not specified (complex type)
deployedApplications       Not specified (complex type)
deployedLibraries          Not specified (complex type)
resources                  Not specified (complex type)

```

#### 4.2.17.6. start-configuration-editing

서버에 요청을 하여 동적 설정 변경 작업에 대한 Lock을 얻는다. Lock을 얻는데 성공한 경우 서버의 현재 경로를 초기화한다. Lock을 얻지 못한 경우 이후에 cancel을 제외한 명령어는 동작하지 않으므로, 설정 편집을 하고자 하는 경우 반드시 이 명령어를 먼저 수행하여 Lock을 얻어야 한다.

- alias

lock-and-edit, edit

- 사용법

```
start-configuration-editing [-f]
```

- 파라미터

파라미터	설명
[-f]	다른 사용자가 동적 설정 변경에 대한 Lock을 갖고 있더라도 이를 무시하고 강제로 Lock을 얻으려고 시도한다.

- 예제

```

[MASTER]domain1.adminServer>start-configuration-editing
Successfully acquired a configuration lock from Domain Administration Server.

```

#### 4.2.17.7. set-element-value

지정한 simple type 항목의 값을 수정하거나, simple type을 원소로 갖는 목록에 값을 추가할 때 사용한다. 이 명령어를 통해 값을 삭제할 수는 없다.

- alias

set

- 사용법

```
set-element-value <path><value>
```

- 파라미터

파라미터	설명
<path>	수정할 항목의 경로를 지정한다.
<value>	수정할 값을 지정한다.

- 예제

- set-element-value 명령어를 사용하여 description 항목에 "Hello, world!"라는 값을 설정하는 예제이다.

```
[MASTER]domain1.adminServer>cd /
Current path is changed to /

[MASTER]domain1.adminServer>ls
description                Not specified (string      )
productionMode             true
id                         982883233
securityManager            complex type
passwordValidator          Not specified (complex type)
adminServerName            adminServer
domainLogHome              Not specified (string      )
systemClusteringFramework complex type
domainBackup               complex type
enableWebadmin             true
enableJsonCommand          false
enableToResynchronizeApplications false
servers                    complex type
sessionCluster             complex type
clusters                   Not specified (complex type)
serverTemplates            Not specified (complex type)
lifecycleInvocation        list of [lifecycleInvocation]
applicationRepositories    Not specified (complex type)
deployedApplications       Not specified (complex type)
deployedLibraries          Not specified (complex type)
resources                  Not specified (complex type)

[MASTER]domain1.adminServer>set-element-value description "Hello,world"
The value Hello,world is set at the specified location /description.

[MASTER]domain1.adminServer>ls
```

description	Hello,world
productionMode	true
id	982883233
securityManager	complex type
passwordValidator	Not specified (complex type)
adminServerName	adminServer
domainLogHome	Not specified (string )
systemClusteringFramework	complex type
domainBackup	complex type
enableWebadmin	true
enableJsonCommand	false
enableToResynchronizeApplications	false
servers	complex type
sessionCluster	complex type
clusters	Not specified (complex type)
serverTemplates	Not specified (complex type)
lifecycleInvocation	list of [lifecycleInvocation]
applicationRepositories	Not specified (complex type)
deployedApplications	Not specified (complex type)
deployedLibraries	Not specified (complex type)
resources	Not specified (complex type)

- simple type을 원소로 갖는 목록에 값을 추가하는 예제이다. 값을 추가할 목록에 대한 경로를 항목 이름으로 사용하는 것을 알 수 있다.

```
[MASTER]domain1.adminServer>start-configuration-editing
Successfully acquired a configuration lock from Domain Administration Server.

[MASTER]domain1.adminServer>cd lifecycleInvocation
Current path is changed to /lifecycleInvocation

[MASTER]domain1.adminServer>create-empty-element com.tmax.sample
Added a new element of type lifecycleInvocation with id [className == com.tmax.sample] to the
list at path /lifecycleInvocation

[MASTER]domain1.adminServer>cd com.tmax.sample
Current path is changed to /lifecycleInvocation/com.tmax.sample

[MASTER]domain1.adminServer>cd invocation
Current path is changed to /lifecycleInvocation/com.tmax.sample/invocation

[MASTER]domain1.adminServer>create-empty-element invocation
Added a new element of type invocation to the list at path
/lifecycleInvocation/com.tmax.sample/invocation

[MASTER]domain1.adminServer>cd [0]
Current path is changed to /lifecycleInvocation/com.tmax.sample/invocation/[0]

[MASTER]domain1.adminServer>ls
  invocationMethod      Not specified (complex type)
  invocationArgument     list of [string]
  invocationType         Not specified (enum[BOOT,
                        BEFORE_DEPLOY, AFTER_DEPLOY, READY,
                        BEFORE_UNDEPLOY, AFTER_UNDEPLOY])

[MASTER]domain1.adminServer>set-element-value invocationArgument arg0
The value arg0 added to the list in the specified location:
```

```

/lifecycleInvocation/com.tmax.sample/invocation/[0]/invocationArgument.
[MASTER]domain1.adminServer>cd invocationArgument
Current path is changed to
/lifecycleInvocation/com.tmax.sample/invocation/[0]/invocationArgument
[MASTER]domain1.adminServer>ls
      [0]                                arg0

Current list can contain the following type of elements:
[string]

[MASTER]domain1.adminServer>set-element-value . arg1
The value arg1 added to the list in the specified location:
/lifecycleInvocation/com.tmax.sample/invocation/[0]/invocationArgument.

[MASTER]domain1.adminServer>ls
      [0]                                arg0
      [1]                                arg1

Current list can contain the following type of elements:
[string]

```

#### 4.2.17.8. show-element

지정한 경로에 위치한 항목에 대한 정보를 표시한다. 출력되는 정보는 항목의 이름, 값, 형식(type) 등이다. 경로를 지정하지 않은 경우 현재 위치에 대한 경로를 표시한다.

- alias

ls

- 사용법

```
show-element [<path>]
```

- 파라미터

파라미터	설명
[<path>]	정보를 조회할 항목의 경로를 지정한다.

- 예제

- complex type에 대한 정보를 출력한 예제이다. 항목의 이름, 값(존재하는 경우), 값의 형식을 출력하는 것을 알 수 있다.

```

[MASTER]domain1.adminServer>show-element

name                adminServer
nodeName            node1
listeners            complex type
logHome              Not specified (string    )
group                Not specified (string    )

```



jvmConfig	complex type
userInterceptor	Not specified (complex type)
userLogging	Not specified (complex type)
systemLogging	list of [systemLogging]
logStdoutToRawFormat	true
actionOnResourceLeak	WARNING (enum[NO_ACTION, WARNING, AUTO_CLOSE])
tmConfig	complex type
externalResource	list of [externalResource]
managedExecutorService	list of [string]
managedScheduledExecutorService	list of [string]
contextService	list of [string]
managedThreadFactory	list of [string]
scheduler	Not specified (complex type)
namingServer	complex type
jmxManager	Not specified (complex type)
systemThreadPool	complex type
dataSourceRemoteLookup	false
engineInitOnStartup	true
useWebEngine	true
useEjbEngine	true
useJmsEngine	true
webEngine	complex type
ejbEngine	complex type
jmsEngine	complex type
useMEJB	false
classFtp	true
enableInterop	Not specified (complex type)
lifecycleInvocation	list of [lifecycleInvocation]
resRef	complex type
jmsResource	complex type
dataSources	Not specified (complex type)
customResourceRefs	Not specified (complex type)
externalResourceRefs	Not specified (complex type)
autoGenerated	Not specified (boolean )

- complex type를 원소로 갖는 목록 항목에 대한 정보를 출력하는 예제이다. 각 항목의 이름은 식별자의 값으로 표시되며, 해당 목록에 들어갈 수 있는 값의 형식과 식별자가 표시되는 것을 알 수 있다.

```
[MASTER]domain1.adminServer>show-element /servers/server
  adminServer
  server1

Current list can contain the following type of elements:
  [server]
The name of the identifier element of [server] element is [name].
```

## 4.2.18. OSGi 관련 명령어

OSGi 기능과 관련한 명령어의 목록은 다음과 같다.

명령어	설명
<a href="#">add-osgi-framework</a>	도메인에 OSGi 프레임워크 설정을 추가한다.

명령어	설명
<a href="#">install-bundle</a>	지정한 위치에 있는 번들을 지정한 OSGi 프레임워크에 설치한다.
<a href="#">list-osgi-frameworks</a>	도메인에 설정되어 있는 OSGi 프레임워크들에 대한 목록을 출력한다.
<a href="#">modify-osgi-framework</a>	OSGi 프레임워크 설정을 수정한다.
<a href="#">remove-osgi-framework</a>	OSGi 프레임워크 설정을 삭제한다.
<a href="#">show-osgi-framework-info</a>	지정한 서버에서 실행 중인 OSGi 프레임워크들에 대한 정보를 출력한다.
<a href="#">start-bundle</a>	지정한 번들을 시작한다.
<a href="#">stop-bundle</a>	지정한 번들을 정지한다.
<a href="#">uninstall-bundle</a>	지정한 번들을 삭제한다.
<a href="#">update-bundle</a>	지정한 번들을 갱신한다.

#### 4.2.18.1. add-osgi-framework

도메인에 OSGi 프레임워크 설정을 추가한다.

- alias

addfwk

- 사용법

```
add-osgi-framework -n,--name <framework-name>
                    [-en,--exportName <jndi-export-name>]
                    [-s,--servers <server-list>]
                    [-iib <true | false>]
                    [-ibd <path>]
                    [-dsl <start-level>]
                    [-foe <true | false>]
                    [-bid <path-list>]
                    [-f,--forceLock]
                    [-detail]
```

- 파라미터

파라미터	설명
<code>-n,--name &lt;framework-name&gt;</code>	OSGi 프레임워크에 대한 고유한 이름을 지정한다.
<code>[-en,--exportName &lt;jndi-export-name&gt;]</code>	OSGi 프레임워크에 대한 JNDI 이름을 지정한다.
<code>[-s,--servers &lt;server-list&gt;]</code>	OSGi 프레임워크를 사용할 서버 목록을 지정한다. 여러 서버를 지정할 경우 각 이름은 콤마(,)를 사용하여 구분한다.

파라미터	설명
<code>[-iib &lt;true   false&gt;]</code>	Initial Bundles Directory에 있는 번들의 설치 여부를 지정한다. (true   false)
<code>[-ibd &lt;path&gt;]</code>	Initial Bundles Directory의 경로를 지정한다. 지정하지 않은 경우 기본 경로로 JEUS_HOME/lib/osgi/Framework_NAME을 사용한다.
<code>[-dsl &lt;start-level&gt;]</code>	초기 번들 디렉터리에 위치한 번들들에 대한 start level 기본값을 지정한다.
<code>[-foe &lt;true   false&gt;]</code>	초기 번들 설치가 제대로 이루어지지 않았을 경우 이를 오류로 간주할지 여부를 설정한다. (true   false)
<code>[-bid &lt;path-list&gt;]</code>	Bundle Installation Descriptor XML 파일들의 위치를 지정한다. 여러 파일을 지정할 경우 콤마(,)를 구분자로 사용한다.
<code>[-f,--forceLock]</code>	설정 변경을 강제로 진행하여 반영한다.
<code>[-detail]</code>	동적 변경 결과에 대한 상세한 정보가 출력된다.

#### • 예제

```
[MASTER]domain1.adminServer>list-osgi-frameworks
No OSGi Frameworks exists.

[MASTER]domain1.adminServer>add-osgi-framework -name framework-0 -servers server1
Successfully performed the ADD operation for OSGi Framework framework-0, but all changes were
non-dynamic. They will be applied after restarting.
Check the results using "list-osgi-frameworks".

[MASTER]domain1.adminServer>list-osgi-frameworks
List of OSGi Frameworks
=====
+-----+-----+-----+
| Framework Name | JNDI Export Name | Target Servers |
+-----+-----+-----+
| framework-0    |                   | server1        |
+-----+-----+-----+
=====
[MASTER]domain1.adminServer>
```

#### 4.2.18.2. install-bundle

지정한 위치에 있는 번들을 지정한 OSGi 프레임워크에 설치한다.

##### • alias

installb

##### • 사용법

```
install-bundle -l,--location <location>
               -f,--framework <framework-name>
```

```
-s,--server <server-name>
```

- 파라미터

파라미터	설명
-l,--location <location>	설치할 번들에 대한 URL을 지정한다.
-f,--framework <framework-name>	번들을 설치할 OSGi 프레임워크의 이름을 지정한다.
-s,--server <server-name>	번들을 설치할 서버를 지정한다.

- 예제

```
[MASTER]domain1.adminServer>install-bundle -location
https://repo.maven.apache.org/maven2/org/ops4j/pax/url/pax-url-mvn/1.3.7/pax-url-mvn-1.3.7.jar
-f,--framework framework-0 -server server1
Installation completed successfully.
[MASTER]domain1.adminServer>
```

#### 4.2.18.3. list-osgi-frameworks

도메인에 설정되어 있는 OSGi 프레임워크들에 대한 목록을 출력한다.

- alias

lsfwks

- 사용법

```
list-osgi-frameworks [-f,--framework <framework-name>]
```

- 파라미터

파라미터	설명
[-f,--framework <framework-name>]	지정한 OSGi 프레임워크 설정에 대한 상세를 출력한다.

- 예제

```
[MASTER]domain1.adminServer>list-osgi-frameworks
List of OSGi Frameworks
=====
+-----+-----+-----+
| Framework Name | JNDI Export Name | Target Servers |
+-----+-----+-----+
| framework-0    |                   | server1        |
+-----+-----+-----+
```

```

+-----+-----+-----+-----+
=====

[MASTER]domain1.adminServer>list-osi-frameworks -framework framework-0
OSGi Framework [framework-0]
=====
+-----+-----+-----+-----+
|                               | Value |
+-----+-----+-----+-----+
| Framework Name               | framework-0 |
| JNDI Export Name             | Not Specified |
| Target Servers               | server1 |
| Install Initial Bundles      | true |
| Initial Bundles Directory Location | Not Specified |
| Default Start Level          | 30 |
| Fail On Error                | true |
+-----+-----+-----+-----+
=====

Framework Configuration Properties
=====
+-----+-----+-----+-----+
| Key | Value |
+-----+-----+-----+-----+
(No data available)
=====

Bundle Installation Descriptors
=====
+-----+-----+-----+-----+
| Location |
+-----+-----+-----+-----+
(No data available)
=====

```

#### 4.2.18.4. modify-osi-framework

OSGi 프레임워크 설정을 수정한다.

- alias

modfwk

- 사용법

```

modify-osi-framework -n,--name <framework-name>
                        [-en,--exportName <jndi-export-name>]
                        [-s,--servers <server-list>]
                        [-iib <true | false>]
                        [-ibd <path>]
                        [-dsl <start-level>]
                        [-foe <true | false>]
                        [-bid <path-list>]
                        [-f,--forceLock]

```

- 파라미터

파라미터	설명
-n,--name <framework-name>	OSGi 프레임워크에 대한 고유한 이름을 지정한다.
[-en,--exportName <jndi-export-name>]	OSGi 프레임워크에 대한 JNDI 이름을 지정한다.
[-s,--servers <server-list>]	OSGi 프레임워크를 사용할 서버 목록을 지정한다. 여러 서버를 지정할 경우, 각 이름은 콤마(,)를 사용하여 구분한다.
[-iib <true   false>]	Initial Bundles Directory에 있는 번들들을 설치 여부를 지정한다. (true   false)
[-ibd <path>]	Initial Bundles Directory의 경로를 지정한다. 지정하지 않은 경우 기본 경로로 JEUS_HOME/lib/osgi/Framework_NAME을 사용한다.
[-dsl <start-level>]	초기 번들 디렉터리에 위치한 번들들에 대한 start level 기본값을 지정한다.
[-foe <true   false>]	초기 번들 설치가 제대로 이루어지지 않았을 경우 이를 오류로 간주할지 여부를 설정한다. (true   false)
[-bid <path-list>]	Bundle Installation Descriptor XML 파일들의 위치를 지정한다. 여러 파일을 지정할 경우 콤마(,)를 구분자로 사용한다.
[-f,--forceLock]	설정 변경을 강제로 진행하여 반영한다.
[-detail]	동적 변경 결과에 대한 상세한 정보가 출력된다.

- 예제

```
[MASTER]domain1.adminServer>modify-osgi-framework -name framework-0 -dsl 45
Successfully performed the MODIFY operation for OSGi Framework framework-0, but some changes were
non-dynamic. They will be applied after restarting.
Check the results using "list-osgi-frameworks".
[MASTER]domain1.adminServer>
```

#### 4.2.18.5. remove-osgi-framework

OSGi 프레임워크 설정을 삭제한다.

- alias

rmfwk

- 사용법

```
remove-osgi-framework -n,--name <framework-name>
```

- 파라미터

파라미터	설명
<code>-n,--name &lt;framework-name&gt;</code>	삭제할 OSGi 프레임워크의 이름을 지정한다.

- 예제

```
[MASTER]domain1.adminServer>remove-osgi-framework -name framework-0
Successfully performed the REMOVE operation for OSGi Framework framework-0.
Check the results using "list-osgi-frameworks".
[MASTER]domain1.adminServer>
```

#### 4.2.18.6. show-osgi-framework-info

지정한 서버에서 실행 중인 OSGi 프레임워크들에 대한 정보를 출력한다.

- alias

osgiinfo, fwkinfo

- 사용법

```
show-osgi-framework-info -s,--server <server-name>
                        [-f,--framework <framework-name>]
                        [-lb]
```

- 파라미터

파라미터	설명
<code>-s,--server &lt;server-name&gt;</code>	대상 서버를 지정한다.
<code>[-f,--framework &lt;framework-name&gt;]</code>	정보를 출력할 OSGi 프레임워크의 이름을 지정한다.
<code>[-lb]</code>	설치되어 있는 번들들에 대한 정보를 출력할지 여부를 지정한다. f 옵션과 함께 사용해야 한다.

- 예제

```
[MASTER]domain1.adminServer>show-osgi-framework-info -server server1 -framework framework-0 -lb
OSGi Framework [framework-0] Information in the server[framework-0]
Framework [framework-0] Overview
=====
+-----+-----+-----+
|               System Bundle Name               | Version |
+-----+-----+-----+
| org.apache.felix.framework                       | 5.4.0   |
+-----+-----+-----+
```

```

+-----+-----+
=====

Installed Bundles in the framework [framework-0].
=====
+---+---+---+---+---+---+
| Id | Symbolic Name | Version | Location | State | Start-level |
+---+---+---+---+---+---+
| 0 | org.apache.felix. | 5.4.0 | System Bundle | ACTIVE | 0 |
| | framework | | | | |
+---+---+---+---+---+---+
| 2 | org.ops4j.pax.url | 1.3.7 | https://repo.maven | ACTIVE | 30 | |
| | .mvn | | .apache.org/maven2/ | | |
| | | | | org/ops4j/pax/url/p | | |
| | | | | ax-url-mvn/1.3.7/pa | | |
| | | | | x-url-mvn-1.3.7.jar | | |
+---+---+---+---+---+---+
=====

[MASTER]domain1.adminServer>

```

#### 4.2.18.7. start-bundle

지정한 번들을 시작한다.

- alias

startb

- 사용법

```

start-bundle -name <bundle-symbolic-name>
              [-version <bundle-version>]
              -f,--framework <framework-name>
              -s,--server <server-name>

```

- 파라미터

파라미터	설명
-name <bundle-symbolic-name>	시작할 번들의 이름을 지정한다.
[-version <bundle-version>]	시작할 번들의 버전을 지정한다.
-f,--framework <framework-name>	OSGi 프레임워크의 이름을 지정한다.
-s,--server <server-name>	대상 서버를 지정한다.

- 예제



```
[MASTER]domain1.adminServer>start-bundle -name org.ops4j.pax.url.mvn -framework framework-0
-server server1
The bundle has been started successfully.
[MASTER]domain1.adminServer>
```

#### 4.2.18.8. stop-bundle

지정한 번들을 정지한다.

- alias

stopb

- 사용법

```
stop-bundle -name <bundle-symbolic-name>
             [-version <bundle-version>]
             -f,--framework <framework-name>
             -s,--server <server-name>
```

- 파라미터

파라미터	설명
-name <bundle-symbolic-name>	정지할 번들의 이름을 지정한다.
[-version <bundle-version>]	정지할 번들의 버전을 지정한다.
-f,--framework <framework-name>	OSGi 프레임워크의 이름을 지정한다.
-s,--server <server-name>	대상 서버를 지정한다.

- 예제

```
[MASTER]domain1.adminServer>stop-bundle -name org.ops4j.pax.url.mvn -framework framework-0
-server server1
The bundle has been stopped successfully.
[MASTER]domain1.adminServer>
```

#### 4.2.18.9. uninstall-bundle

지정한 번들을 삭제한다.

- alias

uninstallb

- 사용법

```
uninstall-bundle -name <bundle-symbolic-name>
                  [-version <bundle-version>]
                  -f,--framework <framework-name>
                  -s,--server <server-name>
```

- 파라미터

파라미터	설명
-name <bundle-symbolic-name>	삭제할 번들의 이름을 지정한다.
[-version <bundle-version>]	삭제할 번들의 버전을 지정한다.
-f,--framework <framework-name>	OSGi 프레임워크의 이름을 지정한다.
-s,--server <server-name>	대상 서버를 지정한다.

- 예제

```
[MASTER]domain1.adminServer>uninstall-bundle -name org.ops4j.pax.url.mvn -framework framework-0
-s-server server1
The bundle has been uninstalled successfully.
[MASTER]domain1.adminServer>
```

## 4.2.18.10. update-bundle

지정한 번들을 갱신한다.

- alias

updateb

- 사용법

```
update-bundle -name <bundle-symbolic-name>
               [-version <bundle-version>]
               -f,--framework <framework-name>
               -s,--server <server-name>
               [-l,--location <location>]
```

- 파라미터

파라미터	설명
-name <bundle-symbolic-name>	갱신할 번들의 이름을 지정한다.
[-version <bundle-version>]	갱신할 번들의 버전을 지정한다.
-f,--framework <framework-name>	OSGi 프레임워크의 이름을 지정한다.
-s,--server <server-name>	대상 서버를 지정한다.
[-l,--location <location>]	갱신할 때 사용할 번들에 대한 URL을 지정한다.

- 예제

```
[MASTER]domain1.adminServer>update-bundle -name org.ops4j.pax.url.mvn -framework framework-0
-server server1
The bundle has been updated successfully.
[MASTER]domain1.adminServer>
```

## 4.2.19. JSON을 사용한 명령어 실행

HTTP POST 요청을 사용하여 명령어 실행을 요청하는 JSON 문장을 보내면, 서버에서 요청을 처리한 후 명령어 실행 결과를 담은 JSON 문장을 HTTP 응답으로 보내주는 구조이다.



JSON에 대한 기본적인 내용은 [www.json.org](http://www.json.org)을 참고한다.

### 4.2.19.1. 일반 명령어 실행

설명을 위해 사용할 예제 명령어는 다음과 같다.

```
serverinfo -server server1 -state
```

위 명령어를 JSON을 사용하여 실행하는 과정은 다음과 같다.

1. 명령어 수행을 요청하는 JSON 문장을 작성한다.

예제 명령어를 실행하기 위해 작성해야 하는 JSON 문장은 다음과 같다.

```
{
  "jeusadmin": {
    "command": "server-info",
    "options": [
      "-server server1",
      "-state"
    ],
    "argument": null
  }
}
```

```
}  
}
```

구분	설명
command	실행할 명령어 이름
options	명령어에 적용할 옵션과 옵션에 대한 인자들
argument	명령어에 적용할 인자들

2. 작성한 JSON 문장을 서버로 전송하여 명령어 실행을 요청한다. 전송할 때 사용하는 프로토콜은 HTTP이며, method는 POST를 사용해야 한다.

JEUS 서버에서 JSON 명령어 실행을 위한 요청을 받는 URL은 다음과 같다.

```
http://${SERVER_HOST}:${SERVER_BASE_PORT}/jsonCommand/command.json
```



JEUS 서버에서 명령어를 실행하려면 사용자 인증이 필요하다. 사용자 인증은 HTTP Basic Authentication을 사용한다.

3. 서버가 요청한 JSON 문장을 해석하여 실행한 후, 실행 결과를 담은 JSON 문장을 HTTP 응답으로 전송한다. 실행 결과는 개별 명령어별로 각각 다르다.

아래 결과는 예제 명령어를 실행한 경우 서버에서 응답으로 올 수 있는 결과물의 예제이다. JEUS 도메인 구성에 따라 결과는 달라질 수 있다.

```
{  
  "jeusadmin-result": {  
    "message": "",  
    "data": [  
      {  
        "title": "Information of Domain (domain1)",  
        "header": null,  
        "column-names": [  
          "Server", "Status", "Node Name", "PID", "Cluster",  
          "Latest StartTime/ShutdownTime", "Need to Restart",  
          "Listen Ports", "Running Engines"  
        ],  
        "rows": [  
          {  
            "row-key": "0",  
            "values": [  
              "server1(*)", "RUNNING (284sec)", "N/A", "2151", "N/A",  
              ... (생략)  
            ]  
          }  
        ],  
        "footer": null  
      }  
    ],  
  }  
},  
[,
```

```

    "post-message": ""
  }
}

```

응답 결과를 담고 있는 JSON 문장은 다음과 같은 요소들로 구성된다.

- message

상단에 출력할 메시지이다. message가 여러 개인 경우 List 형태로 표현한다. 명령어에 따라 존재하지 않을 수도 있다.

- data

실행 결과를 담고 있는 객체로 명령어에 따라 존재하지 않을 수도 있다. data는 기본적으로 표 형태의 자료 구조를 표현하고 있으며, 구성 요소는 다음과 같다.

구분	설명
title	표의 제목이다.
header	표의 머릿말이다.
column-names	표를 구성하는 각 열(column)들의 이름이다.
rows	표를 구성하는 각 행(row)의 내용이다.  다음의 2가지 구성 요소로 이루어져 있다. <ul style="list-style-type: none"> <li>• row-key : 해당 행의 이름이다.</li> <li>• values : 해당 행에 들어갈 값이다.</li> </ul>
footer	표의 꼬릿말이다.

- post-message

하단에 출력할 메시지이다. post-message가 여러 개인 경우 List 형태로 표현한다. 명령어에 따라 존재하지 않을 수도 있다.

#### 4.2.19.2. Application deploy

본 절에서는 JSON을 사용하여 애플리케이션을 설치하거나 배포하는 방법을 설명한다. 애플리케이션을 설치하거나 배포하는 작업은 일반 명령어와 다르게 실행할 JSON 명령어와 배포할 애플리케이션을 함께 전달해야 한다.

다음은 JSON을 사용하여 애플리케이션을 설치하거나 배포하는 과정에 대한 설명이다.

1. 명령어 수행을 요청하는 JSON 문장을 작성한다. Hello.war라는 애플리케이션을 'server1'에 배포하려고 할 경우 JSON 명령어는 다음과 같이 작성한다.

```

{
  "jeusadmin": {

```

```

    "command": "deploy-application",
    "options": [
        "-servers server1",
        "-path Hello.war"
    ],
    "argument": null
  }
}

```

2. 작성한 JSON 문장과 배포할 파일을 Multi-part 형태로 서버로 전송하여 명령어 실행을 요청한다. Multi-part는 1에서 작성한 JSON 명령어를 담은 command 파트와 배포할 애플리케이션 아카이브를 지정한 file 파트로 구성한다.

JEUS 서버에서 JSON 명령어를 통한 애플리케이션 deploy 요청을 받는 URL은 다음과 같다.

```
http://${SERVER_HOST}:${SERVER_BASE_PORT}/jsonCommand/install.json
```



JEUS 서버에서 명령어를 실행하려면 사용자 인증이 필요하다. 사용자 인증은 HTTP Basic Authentication을 사용한다.

3. 서버가 요청한 JSON 문장을 해석하여 실행한 후 실행 결과를 담은 JSON 문장을 HTTP 응답으로 전송한다. 이 부분은 일반 명령어를 실행했을 경우와 다르지 않으므로 [일반 명령어 실행](#)을 참고한다.



애플리케이션 설치 또는 배포에 관련한 명령어는 [install-application](#), [deploy-application](#), [distribute-application](#)이 있다.

## 4.2.20. Script 모드 사용법 및 Script 작성법

본 절에서는 Script 모드 사용법 및 Script 작성법에 대해서 설명한다.

### 4.2.20.1. Script 모드 사용법

Script모드로 jeusadmin을 동작시킨다.

- 예제

다음은 JEUS 사용자 이름은 'administrator', 비밀번호는 'jeus'로 가정하고 test.py 스크립트를 실행하는 jeusadmin script 모드 예제이다.

```

JEUS_HOME/bin$jeusadmin -u administrator -p jeus -script "test.py"
Attempting to connect to 127.0.0.1:9736.
The connection has been established to JEUS Master Server [adminServer] in the domain [domain1].
JEUS 9 Administration Tool

```

To view help, use the 'help' command.

Script 모드로 사용시 추가적으로 JEUS 커맨드 예외를 무시하는 옵션을 사용 할 수 있다. -i 또는 -ignore 옵션을 사용하면 JEUS 커맨드 오류가 발생하더라도 스크립트는 계속 이어서 실행된다. 단, 스크립트 자체의 오류가 있을 경우에는 중단된다.

[-i, --ignore] 옵션은 Script 모드에서만 사용 가능하다.

```
JEUS_HOME/bin$jeusadmin -u administrator -p jeus -script "test.py" -i
```

```
JEUS_HOME/bin$jeusadmin -u administrator -p jeus -script "test.py" --ignore
```

아래와 같이 스크립트에 인자를 넘겨 줄 수도 있다.

```
JEUS_HOME/bin$jeusadmin -u administrator -p jeus -script "test.py arg1 arg2"
```

#### 4.2.20.2. Script 작성법

jeusadmin은 Script 모드로 동작할 때에 JEUS 커맨드 실행을 위한 메서드와 결과 데이터를 제공한다.

- 커맨드 실행

아래와 같은 방법으로 각 스크립트 언어별로 JEUS 커맨드를 실행할 수 있다.

- Python, Ruby 공통

```
result = command("server-info")
```

- Ruby

```
result = command "server-info"
```

선택 사항으로 JEUS 커맨드에 verbose 옵션을 추가적으로 실행할 수 있다. 기본적으로 비활성화(false) 상태이다. JEUS 커맨드에 verbose 옵션을 활성화(true) 한 경우에는 해당 커맨드의 실행 결과를 화면에 상세하게 출력한다.

- Python:

```
result = command("server-info", [True | False])
```

- Ruby:

```
result = command "server-info", [true | false]
```

- 결과 데이터

JEUS 커맨드 실행에 대한 결과로 JeusResult 객체가 리턴되며 결과가 표 형태의 결과를 포함할 경우 JeusResult로부터 JeusTabularData 객체를 얻을 수 있다. 각 데이터 타입은 아래와 같은 메소드를 제공한다.

- JeusResult

메소드	설명
isComplete()	JEUS 커맨드 실행이 정상적으로 완료되었는지 리턴한다.
getMessage()	결과 메시지를 리턴한다.
getData()	결과의 표 데이터인 JeusTabularData의 리스트를 리턴한다.

- JeusTabularData

메소드	설명
getTitle()	표의 타이틀을 리턴한다.
getHeader()	표의 머릿말을 리턴한다.
getFooter()	표의 꼬릿말 리턴한다.
getColumnNames()	표의 각 열의 제목만을 리스트 형태로 리턴한다.
getRows()	표 데이터의 각 행의 리스트를 리턴한다.

- 예제

- SHUTDOWN 상태인 서버들을 찾아 출력한다.

```
result = command("server-info")
tables = result.getData();
table = tables[0];
rows = table.getRows();
shutdown_servers = []
for row in rows:
    if "SHUTDOWN" in row[1]:
        shutdown_servers.append(row[0])
        print("Server : %s, Node : %s" % (row[0], row[2]))
```

- 스크립트의 인자로 애플리케이션 경로와 deploy할 서버 리스트를 받아 애플리케이션을 install, deploy한다.

```
import time
import sys
import os
```



```

path = sys.argv[0]
servers = sys.argv[1]
direc, app = os.path.split(path)
apptype = os.path.splitext(app)[1][1:].upper()

command("undeploy -f %s " % app)
command("uninstall-application %s" % app)
command("install-application %s -id %s" % (path, app))
command("deploy %s -servers %s -type %s" % (app, servers, apptype))

```



현재 jeusadmin에서는 Python과 Ruby 두 가지 스크립트 언어에 대해서만 스크립트 모드를 제공하며 이를 확장자로 구분하고 있다. 각각 .py와 .rb 확장자에 대해서만 스크립트를 인식하여 동작한다.

## 4.3. appcompiler

appcompiler는 EJB 인터페이스 Impl, Skeleton, Stub 클래스, JSP를 컴파일하여 서블릿 클래스, 웹 서비스의 Endpoint 클래스 등을 생성하는 애플리케이션 컴파일러이다.

appcompiler는 EJB 2.1 인터페이스의 Impl 클래스, RMI Stub과 Skeleton 클래스, 웹 모듈의 JSP를 컴파일해서 서블릿 클래스들을 미리 생성하기 위해 사용할 수 있다. EJB 3.0 인터페이스로만 구성된 EJB의 경우나 JAX-WS 클라이언트, 서버 애플리케이션은 이 툴을 실행할 필요가 없다.

이 툴은 다음과 같은 경우에 사용하면 효과적이다.

1. EJB 2.x 표준에 준하는 Bean들이 많아서 최초 디플로이 시간이 오래 걸리는 경우
2. 서비스 타임에 JSP 컴파일할 시간이 오래 걸려서 미리 컴파일을 해두어야 하는 경우

EJB 2.x 모듈의 경우에는 deploy할 때에 애플리케이션에 fast-deploy 설정을 해야 한다. 또는 jeusadmin 콘솔 툴의 **deploy** 명령어에 [-fast] 옵션을 추가한다. 이 옵션을 통해서 모듈의 런타임 deploy하는 경우 appcompiler가 자동으로 수행되는 것을 막을 수 있다. EJB 엔진은 fast-deploy 옵션이 설정된 경우 인터페이스 Impl, Skeleton, Stub 클래스들을 생성하고 컴파일하는 과정을 수행하지 않고, 미리 생성된 클래스들을 사용해서 deploy할 때 시간을 줄일 수 있다. EJB 3.x 모듈의 경우 appcompiler가 수행되지 않는다. 따라서 하나의 모듈안에 EJB 3.x 모듈내에 EJB 2.x가 함께존재하더라도 appcompiler는 수행되지 않는다.

웹 엔진은 JSP에 대한 컴파일된 서블릿 클래스 파일이 있을 경우에는 실제 서비스 타임에 컴파일하지 않는다. 이 툴에는 하나의 파일을 생성한 후 컴파일하는 Each 모드와 모든 파일을 생성한 후 한 번에 컴파일하는 Batch 모드가 있다. Batch 모드는 Each 모드에 비해 속도는 빠르지만 컴파일할 때 에러가 나면 원인이 되는 파일을 찾기 힘들다는 단점이 있다. 기본 동작은 Each 모드이므로 Batch 모드를 원할 경우 [-batch] 옵션을 이용하거나 jeus.app.compiler.mode=batch를 설정한다. 단, Batch 모드는 EJB 2.x 모듈에만 적용된다.



Windows에서 WAR 파일에 대해 appcompiler 수행할 때 임시로 만든 디렉터리가 지워지지 않을 수 있다. 이는 java.net.URLClassLoader가 WAR 파일 내부에 포함된 WEB-INF/lib/\*.jar 파일들을 명시적으로 닫을 수 없기 때문이다. java.net.URLClassLoader의 경우 Java 7부터 close 메소드가 추가되었으므로 이에 기반하는 향후 버전에서는 문제가 생기지 않을 것이다. 그러나 현재는 임시 디렉터리를 수동으로 삭제해야 한다.

다음은 툴 사용법, 파라미터 및 예제에 대한 설명이다.

- 사용법

```
appcompiler [-h]
             [-verbose]
             [-clp <class-path>]
             [-keep]
             [-jspmap]
             [-batch]
             [-q]
             [-client <clientview_filename>]
             [-noaddfile]
             [-deloldgen]
             [-ejbjar <ejb-jar.xml_path>]
             [-jeusejbdd <jeus-ejb-dd.xml_path>]
             [-D <property=value>]
             [-property <file-name>]
             [-target <application-path>]
             [-j concurrency-level]
             [-ejbonly]
             [-genjavaonly]
             [-webonly]
```

- 파라미터

파라미터	설명
[-h]	appcompiler의 도움말을 출력한다.
[-verbose]	verbose 모드를 설정하여 상세한 과정을 출력한다(로그 레벨을 FINEST로 설정한다).
[-clp]	컴파일 과정에서 필요한 라이브러리를 class path로 등록한다. 파일 또는 디렉터리 모두 올 수 있으며 각각을 구분할 때 Windows는 세미콜론( ; ), Unix는 콜론( : )을 쓰도록 한다.
[-keep]	컴파일 과정에서 생성된 소스를 유지할 것인지를 설정한다.
[-jspmap]	servlet-mapping table(jeus_jspmap.xml)을 생성할 것인지를 설정한다. 단, 이 옵션을 사용할 경우에는 JSP 파일이 있는 디렉터리 이름에 Java 예약어(예: if, enum, class 등)가 있으면 안 된다.
[-batch]	모든 파일을 생성한 후 한 번에 컴파일하는 Batch 모드로 동작한다. 이 옵션이 없으면 하나의 파일을 생성한 후 컴파일하는 Each 모드로 동작한다. 이 옵션은 EJB 2.x 모듈에만 적용된다.
[-q]	여러 모듈들을 포함한 EAR 애플리케이션을 컴파일할 때 특정 웹 모듈 컴파일을 실패할 경우 남은 다른 모듈들을 컴파일하지 않고 에러 메시지를 출력하고 중단한다.  이 옵션이 없을 경우에는 특정 웹 모듈 컴파일을 실패해도 에러 메시지만 출력하고 남은 다른 모듈들에 대해 컴파일을 진행한다.

파라미터	설명
[-client <clientview_filename>]	<p>생성될 Stub 클래스들과 인터페이스 클래스들이 포함될 clientview 파일의 이름을 설정한다.</p> <ul style="list-style-type: none"> <li>• &lt;clientview_filename&gt;은 EJB archived 파일 형식인 JAR만 지원하며, 컴파일하는 대상(archived or exploded application)이 존재하는 디렉터리에 생성된다. 이 옵션은 standalone EJB 모듈과 EJB 모듈을 가지고 있는 EAR 애플리케이션에만 적용된다.</li> <li>• &lt;clientview_filename&gt;은 &lt;application_file_or_directory_name&gt;과 동일한 경우에는 지원하지 않는다.</li> <li>• &lt;clientview_filename&gt;과 동일한 이름의 파일이 존재할 경우 clientview를 위한 인터페이스 클래스와 생성된 Stub 클래스를 이미 존재하는 동일한 이름의 파일에 포함시킨다. clientview 파일은 서버에 설치된 EJB를 이용하는 클라이언트 애플리케이션을 개발하기 위해 클라이언트에게 제공한다.</li> </ul>
[-noaddfile]	<p>[-client] 옵션과 함께 사용할 때 유효하며, clientview 파일을 생성하지만, 컴파일 대상이 되는 파일은 컴파일하지 않고 원본 내용을 유지한다.</p> <p>예를 들어 calc.jar를 appcompiler로 수행할 때 clientview 파일을 생성하기 위해서 [-client] 옵션을 사용한다. 여기서 clientview 파일은 Stub과 Remote, Home 인터페이스 클래스들만 모아둔 파일을 의미한다. 이때 calc.jar를 appcompiler로 컴파일한 뒤 clientview 파일을 생성하기 때문에 calc.jar를 컴파일한다. 컴파일된 파일은 원본 파일에서 컴파일 과정을 통해 생성된 Impl, Skeleton, Stub 등 클래스 파일들이 추가되어 있다. 만약 사용자가 calc.jar 파일 자체는 컴파일된 상태로 만들기를 원하지 않고, clientview 파일만 생성하기를 원할 경우에 [-noaddfile] 옵션을 사용할 수 있다.</p>
[-deloldgen]	<p>EJB 모듈을 컴파일할 때 JEUS 이전 버전에서 만들어진 Impl, Skeleton, Stub 파일들을 삭제하고 컴파일을 시도한다.</p> <p><b>[참고]</b></p> <p>JEUS 4와 JEUS 5에서 Impl, Skeleton, Stub 클래스들의 이름을 생성하는 방식이 JEUS 8 이후 버전과는 다르다. JEUS 이전 버전인 JEUS 4와 JEUS 5에서 생성한 EJB 모듈을 appcompiler로 컴파일하여 JEUS 8 이후 버전에서 사용하려고 할 때 기존 버전의 Impl, Skeleton, Stub 파일들이 남아있어 정상적으로 EJB가 수행하지 않는 문제점을 해결할 수 있다.</p>
[-ejbjar <ejb-jar.xml_path>]	컴파일할 때 사용할 ejb-jar.xml 파일을 설정한다.
[-jeusejbdd <jeus-ejb-dd.xml_path>]	컴파일할 때 사용할 jeus-ejb-dd.xml 파일을 설정한다.
[-D <property=value>]	시스템 프로퍼티를 지정한다. 이 옵션은 한 번 이상 이용할 수 있다.
[-property <file-name>]	파일로부터 시스템 프로퍼티를 설정한다.

파라미터	설명
[-target <application-path>]	컴파일할 대상 애플리케이션 파일(archived ear, jar, war) 또는 디렉터리(exploded EAR, JAR, WAR)를 설정한다.  EAR뿐 아니라 standalone JAR, WAR도 하나의 애플리케이션으로 생각한다. 컴파일해서 생성된 Impl, Skeleton, Stub 클래스 파일들은 컴파일하는 대상 애플리케이션의 파일에 추가된다.
[-j concurrency-level]	JSP 컴파일할 때에 사용하는 값이다. JSP 파일들이 많은 경우 이 설정을 통해서 여러 개의 Thread로 동시에 컴파일을 진행하는 것이 효율적이다. (기본값: 1)
[-ejbonly]	EAR 내부에 있는 EJB 모듈만 컴파일한다.
[-genjavaonly]	Java 파일만을 생성하고 싶을 때에 사용한다.
[-webonly]	EAR 내부에 있는 Web 모듈만 컴파일한다.

#### • 예제

appcompiler는 JEUS\_HOME/bin/ 디렉터리에 위치한 일반적인 스크립트 파일이다.

- ejb.jar 모듈의 Home, Remote interface Impl, Skeleton, Stub 클래스를 생성해서 JAR 파일에 포함한다.

```
JEUS_HOME/bin$ appcompiler ejb.jar
```

- web.war 모듈의 모든 JSP 파일로부터 서블릿 클래스를 생성해서 WAR 파일에 포함한다.

```
JEUS_HOME/bin$ appcompiler web.war
```

- Batch 모드로 모듈을 컴파일한다.

```
JEUS_HOME/bin$ appcompiler -batch app.jar
```

- ejb.jar 모듈의 home, remote 인터페이스 Impl, Skeleton, Stub 클래스를 생성해서 ejb.jar 파일에 포함시키며 Home, Remote 인터페이스와 생성된 Stub 클래스를 포함한 clientview.jar 파일을 생성한다.

```
JEUS_HOME/bin$ appcompiler -client clientview.jar ejb.jar
```

## 4.4. ejbddinit

ejbddinit은 ejb-jar.xml 정보와 사전에 생성된 프로퍼티 파일 및 사용자 입력 값 등을 이용해서 JEUS EJB DD(jeus-ejb-dd.xml)를 자동으로 생성하는 툴이다.

EJB를 deploy할 때 필요한 항목들은 기본값으로 설정하므로 JEUS EJB DD는 반드시 존재해야하는 것은 아니다.

따라서 EJB를 새로 개발할 경우 불필요할 수 있지만, JEUS EJB DD를 사용하던 기존 방법으로 애플리케이션을 작성하거나 다른 제품에서 전환(migration)하는 경우 쉽게 템플릿을 작성하는 용도로 사용할 수 있다.



Java EE 6부터 지원하는 annotation-based 방식은 소스 파일 내에 설정할 수 있으므로 지원하지 않는다.

다음은 툴 사용법, 파라미터 및 예제에 대한 설명이다.

- 사용법

```
ejbddinit [-property property_file_path]  
          [-level log_level]  
          [-source application_file_or_directory_name]
```

- 파라미터

파라미터	설명
<code>[-property <i>property_file_path</i>]</code>	<p>프로퍼티 파일의 경로를 지정한다. jeus-ejb-dd.xml을 목적에 맞게 생성하기 위해서는 이 옵션을 반드시 설정해야 한다. 프로퍼티를 JEUS EJB DD의 태그에 대응시켜 값을 설정할 수 있다. 설정 가능한 프로퍼티 목록은 <a href="#">프로퍼티 목록</a>을 참고한다.</p> <ul style="list-style-type: none"><li>• 예1)</li></ul> <div>export-port=40001</div> <p>모든 EJB에 대해서 export-port를 40001로 설정하여 jeus-ejb-dd.xml 파일을 생성한다.</p> <ul style="list-style-type: none"><li>• 예2)</li></ul> <div>SampleBean.export-port=40004</div> <p>ejb.jar에서 EJB 이름이 SampleBean으로 설정된 EJB 대해서 export-port를 40004로 설정하여 jeus-ejb-dd.xml 파일을 생성한다. 이와 같이 각 프로퍼티 이름 앞에 점(.)을 구분자로 하여 특정 EJB 이름을 명시할 수 있다. 동일한 설정에 대하여 특정 EJB 이름을 명시한 프로퍼티와 그렇지 않은 프로퍼티가 존재할 때 전자가 우선 적용된다.</p>
<code>[-level <i>log_level</i>]</code>	ejbddinit를 실행할 때 화면에 출력되는 로그의 레벨을 설정한다.

파라미터	설명
[-source application_file_or_directory_name]	<p>대상 EJB 모듈의 경로를 설정한다. EJB 모듈은 '.jar' 확장자를 갖는 압축 파일이거나 디렉터리이다.</p> <p>성공적으로 JEUS EJB DD를 생성하면, EJB 모듈이 디렉터리일 경우 META-INF 폴더 하위에 jeus-ejb-dd.xml을 생성한다. EJB 모듈이 압축 파일인 경우 본래의 파일 이름 끝에 ".new"를 붙인 이름으로 새로운 압축 파일을 만들고 압축 파일 내부의 META-INF 디렉터리 하위에 jeus-ejb-dd.xml을 생성한다.</p>

## • 예제

- usage를 출력한다.

```
$ ejbddinit
```

- ejbddinit.properties에 정의된 프로퍼티를 바탕으로 EJB 모듈인 ejb.jar에 대하여 ejb.jar.new 파일을 생성한다. 생성된 ejb.jar.new 파일 내부의 META-INF 디렉터리 하위에 jeus-ejb-dd.xml이 위치한다.

```
$ ejbddinit -property ejbddinit.properties -source ejb.jar
```

- ejbddinit.properties에 정의된 프로퍼티를 바탕으로 EJB 모듈인 ejb\_dir의 META-INF 디렉터리 하위에 jeus-ejb-dd.xml을 생성한다.

```
$ ejbddinit -property ejbddinit.properties -source ejb_dir
```

- ejbddinit.properties에 정의된 프로퍼티를 바탕으로 jeus-ejb-dd.xml을 생성한다. source 옵션 값을 입력하지 않았으므로 ejbddinit.properties에 EJB 모듈 경로가 설정되어 있어야 한다.

```
$ ejbddinit -property ejbddinit.properties
```

- ejbddinit.properties에 정의된 프로퍼티를 바탕으로 EJB 모듈인 ejb.jar에 대하여 ejb.jar.new 파일을 생성한다. 생성된 ejb.jar.new 파일 내부의 META-INF 디렉터리 하위에 jeus-ejb-dd.xml이 위치한다. FINE 레벨의 로그가 화면에 출력된다.

```
$ ejbddinit -property ejbddinit.properties -level FINE -source ejb.jar
```



ejbddinit은 Ant Task를 지원한다. ejbddinit ant task는 [ejbddinit](#)에서 설명한다.

## 프로퍼티 목록

다음은 ejbddinit에서 지원하는 프로퍼티 목록이다. 각 프로퍼티에 대한 상세 설명은 대응되는 jeus-ejb-dd.xml

태그의 description을 참고한다.

프로퍼티	Type	비고
source	String	대상 EJB 모듈의 경로이다. jeus-ejb-dd.xml의 태그와 관련 없다.
logging-level	String	ejbddinit를 실행할 때 화면에 출력되는 로그의 레벨로 jeus-ejb-dd.xml의 태그와 관련 없다.
export-name	String	jeus-ejb-dd.xml의 <export-name> 태그에 대응한다. 특정 EJB에 대해서만 사용을 권장한다. 사용하는 경우 패턴을 지원한다. 자세한 내용은 <a href="#">export-name 프로퍼티에 대한 패턴 지원</a> 을 참고한다.
local-export-name	String	jeus-ejb-dd.xml의 <local-export-name> 태그에 대응한다. 특정 EJB에 대해서만 사용을 권장한다.
export-port	int (not negative)	jeus-ejb-dd.xml의 <export-port> 태그에 대응한다.
export-iiop	boolean	jeus-ejb-dd.xml의 <export-iiop> 태그에 대응한다.
thread-max	int (not negative)	jeus-ejb-dd.xml의 <thread-max> 태그에 대응한다.
bean-pool-min	int (not negative)	jeus-ejb-dd.xml의 <bean-pool>/<pool-min> 태그에 대응한다.
bean-pool-max	int (not negative)	jeus-ejb-dd.xml의 <bean-pool>/<pool-max> 태그에 대응한다.
connect-pool-min	int (not negative)	jeus-ejb-dd.xml의 <connect-pool>/<pool-min> 태그에 대응한다.
connect-pool-max	int (not negative)	jeus-ejb-dd.xml의 <connect-pool>/<pool-max> 태그에 대응한다.
capacity	int (not negative)	jeus-ejb-dd.xml의 <capacity> 태그에 대응한다.
passivation-timeout	long (not negative or -1 for disable)	jeus-ejb-dd.xml의 <passivation-timeout> 태그에 대응한다.
disconnect-timeout	long (not negative or -1 for disable)	jeus-ejb-dd.xml의 <disconnect-timeout> 태그에 대응한다.
engine-type	String (defined)	jeus-ejb-dd.xml의 <engine-type> 태그에 대응한다.  다음 중에 하나를 설정한다. <ul style="list-style-type: none"><li>◦ EXCLUSIVE_ACCESS</li><li>◦ SINGLE_OBJECT</li><li>◦ MULTIPLE_OBJECT</li></ul>

프로퍼티	Type	비고
subengine-type	String (defined)	jeus-ejb-dd.xml의 <subengine-type> 태그에 대응한다.  다음 중에 하나를 설정한다.  ◦ ReadLocking ◦ WriteLocking ◦ WriteLockingFind
fetch-size	int (not negative)	jeus-ejb-dd.xml의 <fetch-size> 태그에 대응한다.
init-caching	boolean	jeus-ejb-dd.xml의 <init-caching> 태그에 대응한다.
table-name	String	jeus-ejb-dd.xml의 <table-name> 태그에 대응한다.
creating-table	String (defined)	jeus-ejb-dd.xml의 <creating-table> 태그에 대응한다.  다음 중에 하나를 설정한다.  ◦ none ◦ use-existing ◦ force-create
deleting-table	boolean	jeus-ejb-dd.xml의 <deleting-table> 태그에 대응한다.
db-vendor	String	jeus-ejb-dd.xml의 <db-vendor> 태그에 대응한다.
datasource-name	String	jeus-ejb-dd.xml의 <data-source-name> 태그에 대응한다.
enable-instant-ql	boolean	jeus-ejb-dd.xml의 <enable-instant-ql> 태그에 대응한다.

### export-name 프로퍼티에 대한 패턴 지원

export-name은 EJB 모듈 이름이나 ejb-jar.xml에 정의된 EJB 컴포넌트의 <ejb-name>, <ejb-class> 태그의 값들을 이용하여 지정하는 경우가 대부분이므로 이들 값들을 좀 더 편리하게 이용할 수 있도록 패턴을 지원한다.

패턴 이름	패턴 값
%{module-name}	EJB 모듈 이름이다. JAR 파일인 경우 '.jar' 확장자를 제거한 파일 이름이다.
%{ejb-name}	EJB 컴포넌트 이름이다. ejb-jar.xml에 정의된 해당 EJB 컴포넌트의 <ejb-name> 태그의 값이다.
%{ejb-fqn}	EJB 컴포넌트의 EJB 클래스의 fully qualified name이다. ejb-jar.xml에 정의된 해당 EJB 컴포넌트의 <ejb-class> 태그의 값이다.
%{ejb-class}	%{ejb-fqn} 값에서 패키지 이름을 제외한 클래스 이름이다.

다음은 이름이 'sample-ejb'이고 EJB 클래스의 fully qualified name이 'sample.SampleBean'인 EJB 컴포넌트가 이름이 'myejb'인 EJB 모듈에 패키징되어 있다고 가정하는 경우 패턴 사용의 예이다.

- export-name을 'myejb\_sample-ejb'로 지정한다.



```
export-name=${module-name}_${ejb-name}
```

- export-name을 'PREFIX\_sample.SampleBean'으로 지정한다.

```
export-name=PREFIX_${ejb-fqn}
```

- export-name을 'SampleBean\_POSTFIX'로 지정한다.

```
export-name=${ejb-class}_POSTFIX
```

위의 예에서 알 수 있듯이 export-name을 설정할 때 필요한 기타 문자열(PREFIX나 POSTFIX와 같은)들을 패턴에 붙일 수 있으며 패턴 간의 조합도 가능하다. 패턴 이름은 대소문자 구별 없이 사용할 수 있다.

## 프로퍼티 파일 예제

다음은 ejbdddinit 프로퍼티 파일의 작성 예다. JEUS EJB DD를 생성할 EJB에 대하여 적절히 동작하도록 다음의 예제를 수정해서 사용할 것을 권장한다.

```
# PROPERTY FILE SAMPLE
# JEUS EJB DD Generation Option

# target file or directory path
source=/home/workspace/ejb.jar

# log-level for EJB DD init
logging-level=DEBUG

# JEUS EJB DD configuration tag and value pairs for all EJBs
db-vendor=mysql
datasource-name=jdbc/_default
creating-table=use-existing
deleting-table=false
engine-type=EXCLUSIVE_ACCESS
subengine-type=ReadLocking
fetch-size=1111
enable-instant-ql=true
export-port=9999
export-iiop=false
thread-max=100
bean-pool-min=10
bean-pool-max=100
capacity=10000
passivation-timeout=300000
disconnect-timeout=-1
connect-pool-min=10
connect-pool-max=100
init-caching=false

# JEUS EJB DD configuration tag and value pairs for BookBean EJB
BookBean.export-name=BookBeanFromProperty
```

```
BookBean.local-export-name=LocalBookBeanFromProperty
BookBean.export-port=55555
BookBean.thread-max=100
```

## 4.5. encryption

encryption은 암호화 툴(Encryption Tool)로 문자열을 암호화하거나 그 반대로 복호화하는 기능을 제공한다. 문자열은 Base64 포맷으로 인코딩하거나 Hash 결과를 제공한다.

secret key가 필요한 알고리즘(예: AES, DES, DESEDE, BLOWFISH, SEED)의 경우 security.key 파일을 생성하여 사용한다. security.key 파일은 기본적으로 JEUS\_HOME/domains/<domain name>/config/security에 위치하며, 변경하려면 시스템 프로퍼티 jeus.security.keypath를 이용하여 경로를 지정한다.

encryption은 다음의 디렉터리에 위치한 encryption 스크립트로 실행할 수 있다.

```
JEUS_HOME/bin/
```

다음은 툴 사용법, 파라미터 및 예제에 대한 설명이다.

- 사용법

```
Usage: encryption <algorithm> <text> [-options ...]
Options:
  -algotlist                show available algorithms
  -algorithm <algorithm>   Encryption algorithm
  -convert                  convert old ciphertext format to new ciphertext format
  -decode                   set to decode mode; used for base16/base64 only
  -domain <domain>         domain name
  -h                        help for Encryption
  -keypath <keypath>       security key path
  -protectkey               use protected secret key
  -s                        describe simple result only
  -text <text>              Text to be encrypted
Simple example : encryption AES 1234
```

- 파라미터

파라미터	설명
-algorithm <algorithm>	text를 인코드할 알고리즘을 설정한다. 필수 옵션으로 명시하지 않을 경우 첫 번째 argument를 algorithm 값으로 간주한다.
-algotlist	사용 가능한 알고리즘들을 출력한다.
-convert	입력한 암호문을 새로운 암호문으로 변경한다.
-decode	BASE16/BASE64로 암호화된 암호문을 디코드하여 평문을 얻어낸다.
-domain <domain>	대칭키 알고리즘을 입력한 domain이 가지고 있는 security.key를 사용한다. -domain, -keypath 중 한 가지만 입력해야 한다.

파라미터	설명
-h	사용법을 출력한다.
-keypath <keypath>	대칭키 알고리즘을 입력한 keypath가 가지고 있는 security.key를 사용한다. -domain, -keypath 중 한 가지만 입력해야 한다.
-protectkey	Master Password를 입력받아 security.key 파일을 암호화 한다. 암호화된 security.key를 사용하기 위해서는 입력했던 Master Password를 입력해야 한다.
-s	인코드 결과값만 출력한다.
-text <text>	인코드될 평문을 설정한다. 필수 옵션으로 명시하지 않을 경우 두 번째 argument를 text 값으로 간주한다.

#### • 예제

다음은 JEUS\_HOME 시스템 환경변수가 설정되어 있고, 시스템 Path에 JEUS\_HOME/bin 디렉터리가 설정되어 있다고 가정하는 경우 encryption 호출에 대한 예이다.

호출 후에 툴은 단순히 인코딩 또는 디코딩된 문자열을 조회한다. 만약 툴에 어떤 파라미터도 제공하지 않았거나, 호출 문법이 잘못되었을 경우 단순한 도움말 메시지가 출력된다.

```
$ encryption base64 mypassword
bXlwYXNzd29yZA==
```

다음은 문자열 "bXlwYXNzd29yZA=="는 "mypassword"를 Base 64포맷으로 인코딩한 것이다.

```
$ encryption DESede mypassword -domain domain_name
[DESEDE] : [mypassword] --> [encode:8JLoskMPHkwwLKi+TJeOgQZBD015PBQ=]
[DESEDE] : [8JLoskMPHkwwLKi+TJeOgQZBD015PBQ=] --> [decode:mypassword]
```

문자열 "8JLoskMPHkwwLKi+TJeOgQZBD015PBQ="는 "mypassword"를 DESede cipher algorithm으로 암호화된 패스워드이다. 이 암호화를 진행하기 위해 만들어진 secret key는 JEUS\_HOME/domains/<domain name>/config/security의 security.key 파일에 저장된다(또는 첫 암호화가 아닐 경우 security.key 파일에 저장된 key를 이용).

-protectkey 옵션을 이용하면 Master Password를 입력받아 security.key 파일을 암호화하는 것이 가능하다.

```
$ encryption -protectkey AES mypassword
Input the master password For key encryption>
Confirm the master password For key encryption>
[AES] : [mypassword] --> [encode:qsHQdjQH8aAr3fWPYwBU0/VkFbs9y1lZeRJaK5xSiuo=]
[AES] : [qsHQdjQH8aAr3fWPYwBU0/VkFbs9y1lZeRJaK5xSiuo=] --> [decode:mypassword]
```

위의 예제와 비슷한데 Master Password를 입력 받는 것이 다르다.

여기서 입력받는 패스워드를 이용하여 secret key를 DESede 알고리즘으로 암호화하게 된다. 위의 명령어를 수행한 이후 만들어진 security.key 파일을 사용하기 위해서는 항상 Master Password를 입력받는 작업이

필요하다. 따라서, 암호화된 security.key 파일을 이용해 암호화를 진행할 경우 다음과 같이 수행된다.

```
$ encryption -protectkey AES mypassword
The encryption key file is encrypted. Enter the master password.
Password>
[AES] : [mypassword] --> [encode:qsHQdjQh8aAr3fWPYwbU0/VkFbs9y1lZeRJaK5xSiuo=]
[AES] : [qsHQdjQh8aAr3fWPYwbU0/VkFbs9y1lZeRJaK5xSiuo=] --> [decode:mypassword]
```

JEUS 6 Fix#4부터 encryption 방식이 좀 더 보안적인 측면에서 강화된 버전으로 바뀌었다. 따라서 cipher text의 포맷이 조금 변화하였는데, Fix#4 이전 버전에서 생성한 암호문을 이후 버전에서 그대로 사용할 수 없게 되었다. 이에 따라 JEUS 6 Fix#5부터 encryption에서 ciphertext를 변환할 수 있는 기능이 추가되었고, 이는 -convert 옵션을 이용해 다음과 같이 사용할 수 있다.

```
$ encryption -convert AES i06wYRz3u60/Gqun2sKtXH1u=
Decryption was successful.
[before:i06wYRz3u60/Gqun2sKtXH1u=] --> [after:ET7c/P21Qx1Io8UI6Ss2NvZ0G=]
```

security.key 파일이 암호화되어 있을 경우에는 다음과 같이 실행한다.

```
$ encryption -convert -protectkey AES i06wYRz3u60/Gqun2sKtXH1u=
The encryption key file is encrypted. Enter the master password.
Password>

Decryption succeed.
[before:i06wYRz3u60/Gqun2sKtXH1u=] --> [after:ET7c/P21Qx1Io8UI6Ss2NvZ0G=]
```

## 4.6. java2wsdl

java2wsdl은 JAX-RPC 웹 서비스를 위한 툴로 Java 클래스로부터 WSDL 파일과 JAX-RPC 매핑 파일을 생성한다. 또한 service end-point interface Java 클래스(또는 구현 클래스)로부터 웹 서비스의 WSDL 파일, JAX-RPC 매핑 파일과 같은 리소스를 생성한다.

다음은 툴 사용법, 파라미터 및 예제에 대한 설명이다.

- 사용법

```
Usage: java2wsdl <options> <input file>

where <input file> include:
    configuration file          service configuration xml file (by default)

where <options> include:
    -classpath <path>          specify where to find input class files
    -cp <path>                  same as -classpath <path>
    -d <directory>             specify where to place generated output files
    -level <log-level>         specify a log level
    -verbose                    [optional] turn verbose mode on
```

- 파라미터

파라미터	설명
-classpath <path>	입력 Java 클래스 파일들을 찾기 위한 경로를 기술한다.
-cp <path>	-classpath <path>와 동일하다.
-d <directory>	결과 파일이 생성될 디렉토리를 지정한다.
-level	로그 레벨을 지정한다.
-verbose	Verbose 메시지를 출력한다.

- 예제

java2wsdl는 JEUS\_HOME/bin/ 디렉토리에 위치한 일반적인 스크립트 파일이다.

java2wsdl 툴을 실행하는 예는 다음과 같다.

```
$ java2wsdl -classpath build/classes -d build/web/WEB-INF service-config.xml
```

위의 명령어는 "service-config.xml" 설정 파일과 "build/classes" 디렉토리의 클래스들을 통해 "build/web/WEB-INF" 디렉토리에 WSDL 문서 및 JAX-RPC 매핑 파일을 생성한다.

## 4.7. jeusddupgrade

jeusddupgrade는 사용자 입력한 애플리케이션 내 JEUS DD 파일(jeus-ejb-dd.xml, jeus-web-dd.xml, jeus-application-dd.xml)을 현재 JEUS 버전에 맞도록 수정해 주는 툴이다.

다음은 툴 사용법, 파라미터 및 예제에 대한 설명이다.

- 사용법

```
usage: jeusddupgrade [OPTIONS] <APP_PATH>
Upgrade old version Jeus deployment descriptor in application.
It is internally equivalent to -upgrade option in deploy command.
-b,--backup    Backup original application.
-h,--help      Print this message.
```

- 파라미터

파라미터	설명
-b,--backup	원본 애플리케이션을 백업한다.  백업 파일 혹은 디렉토리는 각 이름 뒤에 '.origin'이 붙는다.
-h,--help	도움말을 출력한다.

- 예제
  - 사용법을 출력한다.

```
$ jeusddupgrade -help
```

- web.war를 백업하고 업그레이드한다.

```
$ jeusddupgrade -b web.war
```

## 4.8. schemagen

schemagen 툴은 Java 클래스들에 나타나는 각각의 Namespace에 대해 하나의 스키마 파일을 생성한다.

다음은 툴 사용법, 파라미터 및 예제에 대한 설명이다.

- 사용법

```
Usage: schemagen [-options ...] <java files>
Options:
  -d <path>           : specify where to place processor and javac generated
                        class files
  -cp <path>           : specify where to find user specified files
  -classpath <path>    : specify where to find user specified files
  -encoding <encoding> : specify encoding to be used for apt/javac invocation
  -episode <file>      : generate episode file for separate compilation
  -version             : display version information
  -fullversion         : display full version information
  -help               : display this usage message
```

- 파라미터

파라미터	설명
-d <path>	프로세서와 클래스 파일들의 위치를 설정한다.
-cp <path>	사용자가 명시하는 파일들을 찾을 곳을 설정한다.
-classpath <path>	-cp <path>와 동일하다.
-encoding <encoding>	apt/javac 호출에서 사용되는 인코딩을 설정한다.
-episode <file>	분리되어 컴파일을 진행하기 위한 episode 파일들을 생성한다.
-version	버전 정보를 출력한다.
-fullversion	전체 버전 정보를 출력한다.
-help	사용법을 출력한다.

- 예제

schemagen는 JEUS\_HOME/bin/ 디렉터리에 위치한 일반적인 스크립트 파일이다.

schemagen 툴을 실행하는 예는 다음과 같다.

```
$ schemagen -d build/classes src/java/cardfile/Address.java
```

위의 명령어는 Root Element에 해당하는 Address.java 소스 파일로부터 "build/classes" 디렉터리에 스키마를 생성한다.

## 4.9. tcpmon

본 절에서는 tcpmon 툴에 대해 설명한다.

### 4.9.1. 실행

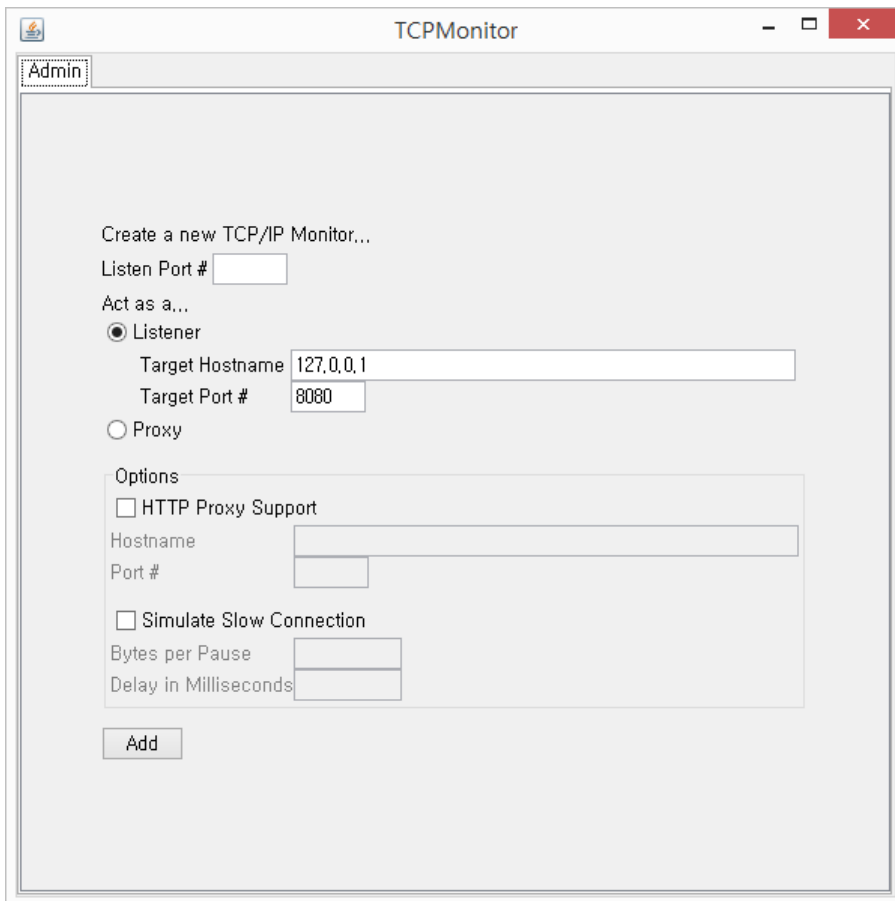
tcpmon 프로그램은 송수신하는 TCP 패킷을 조회하는 프로그램으로 HTTP의 SOAP 메시지를 조회할 때 사용한다.

tcpmon은 다음의 2가지 모드로 동작한다.

모드	설명
Listener	tcpmon이 listen port로 요청을 받아서 지정된 target host와 target port로 받은 요청을 다시 전송하고 응답을 받아서 원래 요청한 커넥션으로 응답을 전송한다.
Proxy	tcpmon을 일반적인 HTTP Proxy처럼 동작하게 한다.

다음과 같이 tcpmon을 실행하면 [tcpmon 실행 화면](#)이 나타난다.

```
$ tcpmon
```



tcpmon 실행 화면

## 4.9.2. Listener 모드의 사용

Listener 모드를 사용하기 위해서 다음과 같은 시나리오를 가정한다.

- 서버(A)는 8000번 Port로 SOAP 요청을 받아들인다.
- 클라이언트(B)의 웹 서비스 클라이언트가 송수신하는 SOAP 메시지를 모니터링하고자 한다.



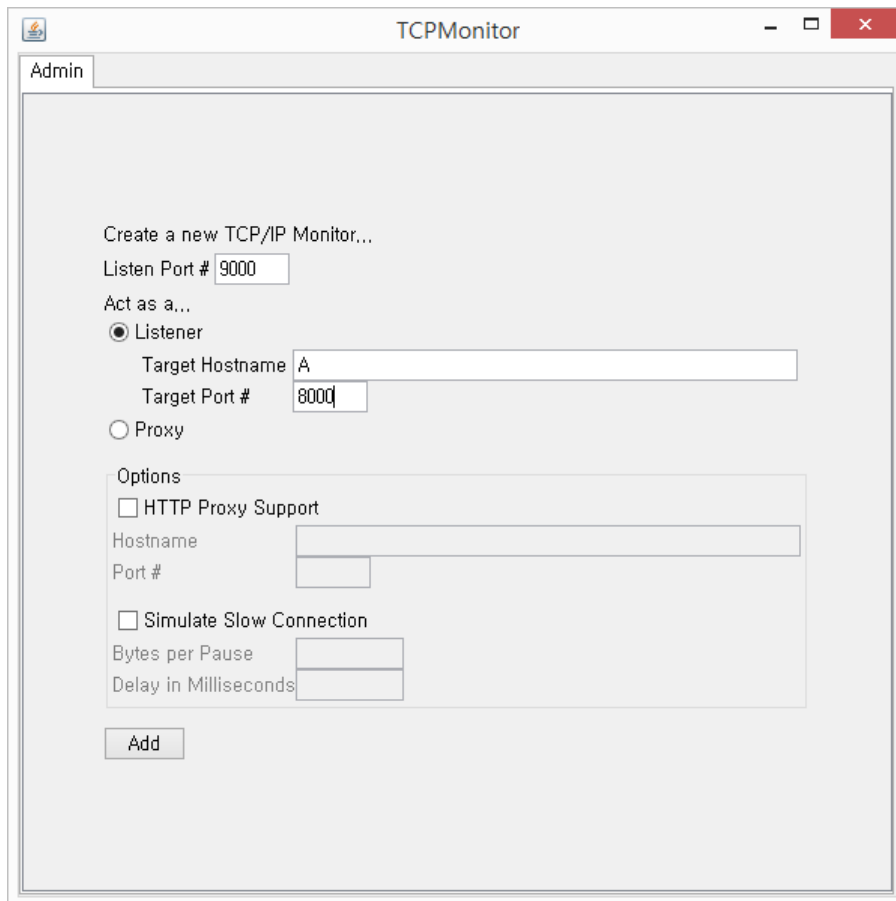
서버(A)가 8000번 포트 SOAP 요청을 받아들이는 시나리오

이 경우 웹 서비스 클라이언트 프로그램에서는 tcpmon을 사용하여 SOAP 메시지를 모니터링할 수 있다.

클라이언트(B)에서 tcpmon을 실행한 뒤 다음과 같이 설정한다.

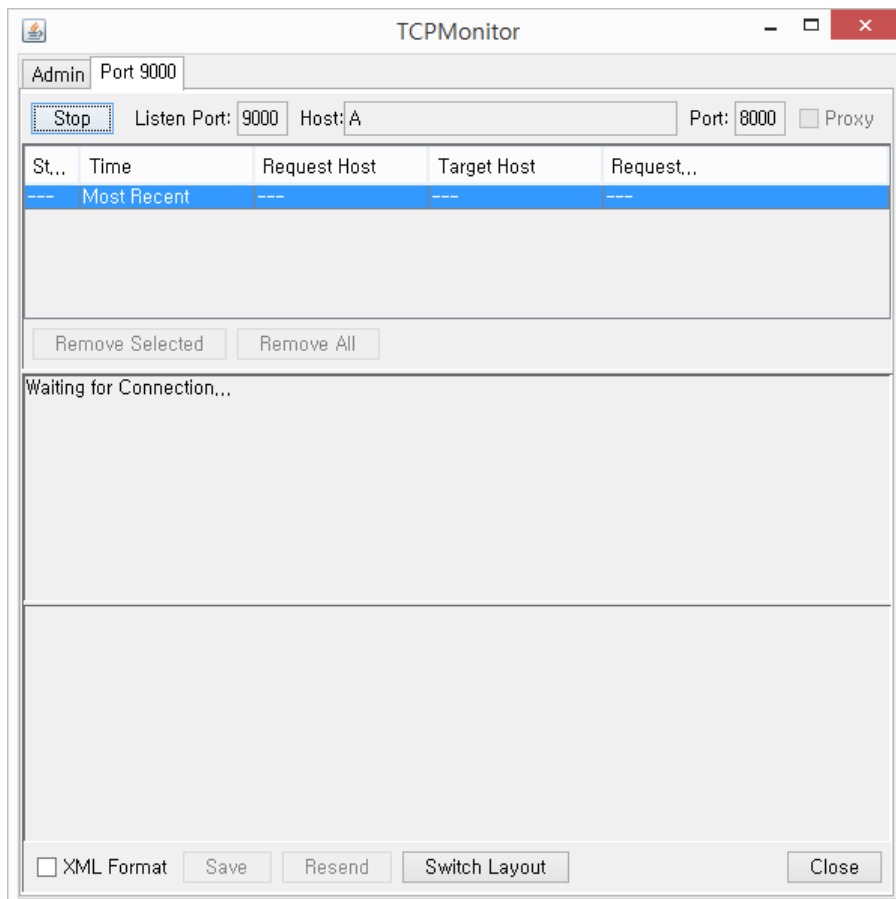
1. **[Admin]** 탭에서 **'Listen Port'**는 '9000'으로 입력하고 **'Act as a'**는 'Listener'를 선택한다. **'Listener'**의 하위 항목인 **'Target Hostname'**는 'A', **'Target Port'**는 '8000'으로 입력한다. 각각의 값을 입력한 후 **[Add]** 버튼을 클릭한다.





Listener 모드 입력 화면

2. 다음과 같이 **[Port 9000]** 탭이 생성된다. 이 탭을 클릭하면 다음과 같은 화면이 나타난다.

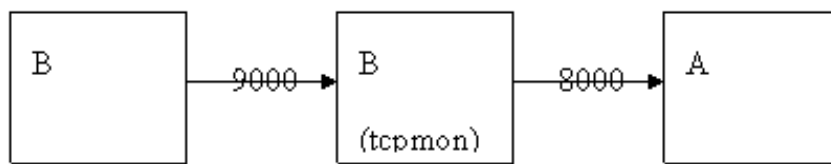


모니터링 화면

화면의 체크박스과 각 버튼에 대한 설명이다.

항목	설명
<b>XML Format</b>	XML을 보기 좋은 형태로 정렬하여 TCPMON 화면에 출력한다.
<b>[Save] 버튼</b>	현재 SOAP 메시지를 저장한다.
<b>[Resend] 버튼</b>	현재 Request SOAP 메시지를 재전송한다.
<b>[Switch Layout] 버튼</b>	메시지 출력 화면의 가로/세로 분할 모드를 바꾼다.

3. 이렇게 설정한 후 클라이언트 프로그램에서 서버(A)의 8000번에 접속하는 대신 클라이언트(B)의 9000번에 접속하도록 한다. 그 후 클라이언트 프로그램을 실행하면 tcpmon이 9000번으로 들어온 요청을 받아서 서버(A)의 8000번 Port로 전송 후 응답을 받아서 클라이언트가 tcpmon에 연결한 커넥션의 응답으로 돌려주고 송수신한 메시지를 tcpmon에 출력한다.



tcpmon이 9000번으로 들어온 요청을 받아서 A의 8000번 Port로 전송

## 웹 서비스의 end-point 주소 변경

tcpmon을 Listener 모드로 사용하는 경우 웹 서비스의 end-point 주소를 바꾸기 위해서 클라이언트 프로그램이나 클라이언트의 설정을 변경할 필요가 있다.

- J2SE JAX-WS 웹 서비스 클라이언트의 경우

접속하려는 웹 서비스의 end-point 주소를 바꾸기 위해서는 다음 예와 같이 코드를 수정한다.

```
(jakarta.xml.ws.BindingProvider)port).getRequestContext().put(
    jakarta.xml.ws.BindingProvider.ENDPOINT_ADDRESS_PROPERTY,
    "http://localhost:9000/ws/AddNumbersService");
```

- port

서비스 end-point 인터페이스에 대한 JAX-WS BindingProvider 객체이다.

- jakarta.xml.ws.BindingProvider.ENDPOINT\_ADDRESS\_PROPERTY

end-point 주소를 지정하기 위한 JAX-WS 스펙에서 정의한 표준 속성이다.

- J2SE JAX-RPC 웹 서비스 클라이언트의 경우

접속하려는 웹 서비스의 end-point 주소를 바꾸기 위해서는 다음 예와 같이 코드를 수정한다.

```
((javax.xml.rpc.Stub)port)._setProperty(
    javax.xml.rpc.Stub.ENDPOINT_ADDRESS_PROPERTY,
    "http://localhost:9000/ws/AddressBookService");
```

- port

서비스 end-point 인터페이스에 대한 JAX-RPC Stub 객체이다.

- javax.xml.rpc.Stub.ENDPOINT\_ADDRESS\_PROPERTY

end-point 주소를 지정하기 위한 JAX-RPC 스펙에서 정의한 표준 속성이다.

- J2EE 웹 서비스 클라이언트의 경우

코드 수정 없이 JEUS 설정 파일(jeus-web-dd.xml 또는 jeus-ejb-dd.xml)을 <service-client>에 <stub-property>를 추가하여 위의 속성을 적용할 수 있다.

```
<service-client>
  <port-info>
    <stub-property>
      <name>javax.xml.rpc.service.endpoint.address</name>
      <value>
        http://localhost:9000/ws/AddressBookService
      </value>
    </stub-property>
  </port-info>
</service-client>
```

### 4.9.3. Proxy 모드의 사용

Proxy 모드는 tcpmon을 일반적인 HTTP Proxy처럼 동작하게 한다. Proxy 모드는 **[Admin]** 탭에서 **'Listen Port'**를 입력하고 **'Act as a'**의 **Proxy**를 선택하고 **[Add]** 버튼을 클릭해서 사용한다.

Proxy 모드의 경우 애플리케이션을 수정하지 않고 애플리케이션을 실행할 때 다음의 옵션을 설정하여 tcpmon을 사용할 수 있다.

옵션	설명
-Dhttp.proxyHost	tcpmon이 수행되고 있는 Host 이름 또는 IP 주소를 입력한다.
-Dhttp.proxyPort	tcpmon의 Listen Port를 입력한다.

클라이언트 애플리케이션이 com.acme.AddressBookClient 클래스인 경우 다음과 같이 옵션을 설정하여 java를 실행한다.

```
java -Dhttp.proxyHost=B -Dhttp.proxyPort=9000 com.acme.AddressBookClient
```

### 4.9.4. 기타 기능

각 Port별 탭에서 SOAP 메시지를 formatting하거나 저장 또는 재전송을 할 수 있다.

## 4.10. webddgen

webddgen은 사용자 입력 값을 바탕으로 웹 애플리케이션 내 WEB-INF 디렉터리에 jeus-web-dd.xml를 생성 또는 업데이트하는 툴이다. 웹 모듈 내부의 WEB-INF 디렉터리에 jeus-ejb-dd.xml을 생성한다. 압축 파일인 경우 그 파일에 jeus-web-dd.xml을 포함시켜준다.

다음은 툴 사용법, 파라미터 및 예제에 대한 설명이다.

- 사용법

```
webddgen -file <war-file-path>
          [-ctx <context-path>]
          [-prop <context-level-property>]
          [-verbose]
```

- 파라미터

파라미터	설명
-file <war-file-path>	웹 모듈의 경로를 설정한다.
[-ctx <context-path>]	jeus-web-dd.xml에 설정할 context-path 정보이다.
[-prop <context-level-property>]	웹 모듈(Web Context) 단위로 설정하는 프로퍼티이다.  예를 들어 JEUS 4에서 작성한 JSP를 수정하지 않고 사용하려면 해당 JSP가 포함된 웹 모듈의 jeus.servlet.jsp.modern 프로퍼티를 false로 설정한다. 프로퍼티 형식은 name=value이다.  (예: jeus.servlet.jsp.modern=false)
[-verbose]	동작에 문제가 있을 때 디버깅 용도로 사용할 수 있다. 이를 위해서 webddgen 내부적으로 생성하는 디렉터리를 삭제하지 않는다. 단, webddgen을 재실행할 때 삭제한다.

- 예제

- 사용법을 출력한다.

```
$ webddgen -help
```

- web.war에 jeus-web-dd.xml을 생성한다.

```
$ webddgen -file web.war
```

- jeus-web-dd.xml을 생성할 때 context-path를 추가한다.

```
$ webddgen -file web.war -ctx /web1
```

- JEUS 4에서 작성한 JSP가 포함된 웹 모듈에 jeus.servlet.jsp.modern 프로퍼티를 false로 설정한다.

```
$ webddgen -file oldjsps.war -prop jeus.servlet.jsp.modern=false
```

## 4.11. wsgen

wsgen 툴은 JAX-WS 웹 서비스를 위한 툴이다. JAX-WS 웹 서비스를 위한 툴로 Java 클래스로부터 WSDL 파일과 JAX-RPC 매핑 파일을 생성하고, 웹 서비스 정책 설정 파일을 생성할 수도 있다.

wsgen 툴은 service endpoint interface 클래스(그리고 임의의 Java로 구현한 클래스)로부터 다음과 같은 것을 생성한다.

- Portable Artifacts
- 웹 서비스의 WSDL 파일(옵션)
- 웹 서비스 정책 파일 `wsit-<endpoint classname>.xml` 파일(옵션)

다음은 툴 사용법, 파라미터 및 예제에 대한 설명이다.

- 사용법

Usage: **wsgen** [options] <SEI>

where [options] include:

- |                     |   |
|---------------------|---|
| -classpath <path>   | specify where to find input class files   |
| -cp <path>          | same as -classpath <path>   |
| -d <directory>      | specify where to place generated output files   |
| -extension          | allow vendor extensions - functionality not specified by the specification. Use of extensions may result in applications that are not portable or may not interoperate with other implementations                   |
| -help               | display help  |
| -keep               | keep generated files  |
| -r <directory>      | resource destination directory, specify where to place resouce files such as WSDLs  |
| -s <directory>      | specify where to place generated source files   |
| -verbose            | output messages about what the compiler is doing  |
| -version            | print version information   |
| -wsdl[:protocol]    | generate a WSDL file. The protocol is optional. Valid protocols are [soap1.1, Xsoap1.2], the default is soap1.1. The non stanadard protocols [Xsoap1.2] can only be used in conjunction with the -extension option. |
| -inlineSchemas      | inline schemas in the generated wsdl. Must be used in conjunction with the -wsdl option.  |
| -servicename <name> | specify the Service name to use in the generated WSDL Used in conjunction with the -wsdl option.  |
| -portname <name>    | specify the Port name to use in the generated WSDL Used in conjunction with the -wsdl option.   |

Examples:

```
wsgen -cp . example.Stock
wsgen -cp . example.Stock -wsdl -servicename {http://mynamespace}MyService
```

## • 파라미터

파라미터	설명
-classpath <path>	입력 Java 클래스 파일들을 찾기 위한 경로를 기술한다.
-cp <path>	-classpath <path>와 동일하다.
-d <directory>	결과 파일이 생성될 디렉터리를 지정한다.
-extension	벤더 특정의 확장성을 명시할 때 사용한다. 상호 운용 및 이식성이 떨어질 수 있다.
-help	도움말을 출력한다.
-keep	생성되는 파일들을 보존한다.
-r <directory>	genwsdl 속성과 함께 사용한다. WSDL 파일을 생성할 위치를 지정한다.
-s <directory>	생성되는 소스 파일들의 위치를 설정한다.
-verbose	Verbose 메시지를 출력한다.
-version	버전 정보를 화면에 출력한다.
-wsdl[:protocol]	기본적으로 wsgen은 WSDL 파일을 생성하지 않는다. 이 옵션을 사용하면 개발자는 deploy되기 전에 WSDL 파일을 볼 수 있다.  [:protocol] 또한 옵션이며 기본값은 soap1.1 이다. Xsoap1.2 또한 사용할 수 있으며 -extension 옵션과 함께 사용한다.
-inlineSchemas	생성된 WSDL의 스키마를 인라인한다.  -wsdl 옵션과 함께 사용해야 한다.
-servicename <name>	-wsdl 옵션과 함께 사용한다. 생성되는 WSDL 파일의 특정 wsdl:service element의 이름을 설정한다.
-portname <name>	-wsdl 옵션과 함께 사용한다. 생성되는 WSDL 파일의 특정 wsdl:portname element의 이름을 설정한다.

## • 예제

wsgen는 JEUS\_HOME/bin/ 디렉터리에 위치한 일반적인 스크립트 파일이다.

wsgen 툴을 실행하는 예는 다음과 같다.

```
$ wsgen -classpath build/classes -d build/classes -wsdl
fromjava.server.AddNumbersImpl
```

위의 명령어는 서비스 end-point 구현 클래스인 fromjava.server.AddNumbersImpl로부터 JAX-WS 웹 서비스를 구성한다. fromjava.server.AddNumbersImpl의 클래스 패스는 "build/classes"이며 WSDL 문서와 함께 JavaBean 파일들이 "build/classes" 디렉터리에 생성된다.

## 4.12. wsimport

wsimport는 JAX-WS 웹 서비스를 위한 툴이다. WSDL 파일로부터 클라이언트 측 Java Stub 소스 파일들과 서버 측 웹 서비스 인터페이스 Java 소스 파일, 웹 서비스 정책 설정 파일을 생성한다.

wsimport 툴은 웹 서비스의 WSDL로부터 다음과 같은 것을 생성한다.

- SEI, 서비스 클래스와 그 밖의 JAXB 관련 Java 클래스들
- 웹 서비스 정책 설정이 담긴 WSDL 파일, wsit-client.xml 파일

다음은 툴 사용법, 파라미터 및 예제에 대한 설명이다.

- 사용법

Usage: **wsimport** [options] <WSDL\_URI>

where [options] include:

- |                          |  |
|--------------------------|--|
| -b <path>                | specify jaxws/jaxb binding files or additional schemas<br>(Each <path> must have its own -b)   |
| -B <jaxbOption>          | Pass this option to JAXB schema compiler   |
| -catalog <file>          | specify catalog file to resolve external entity<br>references supports TR9401, XCatalog,<br>and OASIS XML Catalog format   |
| -d <directory>           | specify where to place generated output files  |
| -extension               | allow vendor extensions - functionality not specified<br>by the specification. Use of extensions may<br>result in applications that are not portable or<br>may not interoperate with other implementations |
| -help                    | display help   |
| -httpproxy:<host>:<port> | specify a HTTP proxy server (port defaults to 8080)  |
| -keep                    | keep generated files   |
| -p <pkg>                 | specifies the target package   |
| -quiet                   | suppress wsimport output   |
| -s <directory>           | specify where to place generated source files  |
| -target <version>        | generate code as per the given JAXWS spec version<br>Defaults to 2.2, Accepted values are 2.0, 2.1 and 2.2<br>e.g. 2.0 will generate compliant code for JAXWS 2.0<br>spec                                  |
| -verbose                 | output messages about what the compiler is doing   |
| -version                 | print version information  |
| -wsdllocation <location> | @WebServiceClient.wsdlLocation value   |
| -clientjar <jarfile>     | Creates the jar file of the generated artifacts along<br>with the WSDL metadata required for invoking the web<br>service.  |
| -generateJWS             | generate stubbed JWS implementation file   |
| -implDestDir <directory> | specify where to generate JWS implementation file  |
| -implServiceName <name>  | local portion of service name for generated JWS<br>implementation  |
| -implPortName <name>     | local portion of port name for generated JWS<br>implementation   |

Extensions:

- |                     |  |
|---------------------|--|
| -XadditionalHeaders | map headers not bound to request or response message |
|---------------------|--|

-Xauthfile	to Java method parameters file to carry authorization information in the format http://username:password@example.org/stock?wsdl
-Xdebug	print debug information
-Xno-addressing-databinding	enable binding of W3C EndpointReferenceType to Java
-Xnocompile	do not compile generated Java files
-XdisableAuthenticator	disable Authenticator used by JAX-WS RI, -Xauthfile option will be ignored if set
-XdisableSSLHostnameVerification	disable the SSL Hostname verification while fetching wsdl

Examples:

```
wsimport stock.wsdl -b stock.xml -b stock.xjb
wsimport -d generated http://example.org/stock?wsdl
```

• 파라미터

파라미터	설명
-b <path>	외부의 JAX-WS 또는 JAXB 바인딩 파일들을 지정한다.
-B <jaxbOption>	이 옵션을 JAXB 스키마 컴파일러에 전달함을 지정한다.
-catalog <file>	TR9401, XCatalog, OASIS XML Catalog 형식과 같은 외부의 Entity 참조값을 설정할 수 있다. 또한 ant xmncatalog 타입을 사용할 수도 있다.
-d <directory>	결과 파일이 생성될 디렉토리를 지정한다.
-extension	벤더 특성의 확장성을 명시할 때 사용한다. 상호 운용 및 이식성이 떨어질 수 있다.
-help	도움말을 출력한다.
-httpproxy:<host>:<port>	HTTP 프록시 서버를 지정한다. (기본값: 8080)
-keep	생성되는 파일들을 보존한다.
-p <pkg>	WSDL의 모든 Namespace URI에 대한 Java 패키지 이름을 지정한다.
quiet	출력 메시지들을 보이지 않게 한다.
-s <directory>	생성된 소스 파일들이 놓일 위치를 지정한다. 이 속성을 지정하면 keep 속성이 자동으로 설정된다.
-target <version>	이곳에 지정된 JAXWS 스펙 버전에 따라 코드를 생성한다.
-verbose	Verbose 출력은 true일 때 가능하다. (기본값: false)
-version	버전 정보를 화면에 출력한다.
-wsdllocation <location>	WSDL URI를 이 속성에 명시하면 그 URI 값이 생성되는 SEI(Service Endpoint Interface)와 서비스 인터페이스의 @WebService.wsdlLocation Annotation과 @WebServiceClient.wsdlLocation Annotation에 설정된다.
-clientjar <jarfile>	웹 서비스 호출에 필요한 WSDL 메타 데이터와 함께 생성된 아티팩트의 jar 파일을 생성한다.
-generateJWS	Stub된 JWS 구현 파일을 생성한다.



파라미터	설명
-implDestDir <directory>	JWS 구현 파일을 생성할 위치를 지정한다.
-implServiceName <name>	생성된 JWS 구현에 대한 서비스 이름의 로컬 부분이다.
-implPortName <name>	생성된 JWS 구현에 대한 포트 이름의 로컬 부분이다.

- 예제

wsimport는 JEUS\_HOME/bin/ 디렉터리에 위치한 일반적인 스크립트 파일이다.

wsimport 툴을 실행하는 예는 다음과 같다.

```
$ wsimport -d build/classes -keep -p fromwsdl.server src/conf/AddNumbers.wsdl
```

위의 명령어는 AddNumbers.wsdl WSDL 문서로부터 패키지명이 fromwsdl.server인 Portable Artifact를 생성한다. [-keep] 옵션에 의해 생성되는 Java 클래스 소스들은 삭제되지 않는다.

## 4.13. wsdl2java

wsdl2java은 JAX-RPC 웹 서비스를 위한 툴로 WSDL 파일로부터 다음의 파일을 생성한다.

- 클라이언트의 웹 서비스 Stub Java 소스 코드들
- 서버의 웹 서비스 인터페이스 Java 소스 코드들

다음은 툴 사용법, 파라미터 및 예제에 대한 설명이다.

- 사용법

```
Usage: wsdl2java mode [options] wsdlURI
```

where mode include:

```
-gen:client      generate all client artifacts
-gen:server      generate all server artifacts
-gen            same as -gen:client
-import:client   generate client JSR-109 portable artifacts only
-import:server   generate server JSR-109 portable artifacts only
-import         same as -import:client
```

where [options] include:

```
* destination directory
-d <directory>    specify where to put output files
-cd <directory>   specify where to put compiled class files
                  If not specified, the compile class files will be
                  put in where '-d' specifies
```

\* WSDL and Java mapping

```
-inputmapping <filename> specify the input JSR-109 JAX-RPC mapping file
                        (used for generating Java artifacts)
-package <package_name> specify the java package name to which
```

```

-ns2pkg NS=PKG      all namespaceURI in the WSDL map
                    specify the namespaceURI and java package name mapping
                    (NS : namespaceURI, PKG : java package name)
                    This option can be used serveral times

* output file
-outputmapping <filename> specify the output JSR-109 JAX-RPC mapping file
                        for the input WSDL
                        This option can not be used with '-inputmapping'

-compile            compile generated Java source files
                    ('tools.jar' must be in the classpath)

-nokeepsrc          delete generated java source files

* artifact generation options
-nowrapped          disable wrapped mode detection for the WSDL
-datahandleronly    force jakarta.activation.DataHandler for MIME types
-nodatabinding      force jakarta.xml.soap.SOAPElement for all WSDL parts
-soapver <VER>      specify SOAP version used in stub/tie class.
                    VER : '11' for SOAP 1.1, '12' for SOAP 1.2
-resolvedir directory specify directory where to be put local copies
                    of include/import schemas and import WSDL

* other options
-username           username to access the WSDL-URI
-password           password to access the WSDL-URI

* verbose options
-level <LEVEL>      specify log level.
                    LEVEL : SEVERE, WARNING, INFO, FINE, FINER, FINEST
-verbose            same as -level FINE

```

## • 파라미터

파라미터	설명
-gen:client	클라이언트 측 Java 클래스를 생성한다.
-gen:server	서버 측 Java 클래스를 생성한다.
-gen	-gen:client와 동일하다.
-import:client	클라이언트 측 Portable Artifact를 생성한다.
-import:server	서버 측 Portable Artifact를 생성한다.
-import	-import:client와 동일하다.
-d <directory>	결과 파일이 생성될 디렉터리를 지정한다.
-cd <directory>	컴파일된 클래스 파일이 생성될 디렉터리를 지정한다.
-inputmapping <filename>	Java 클래스를 생성하기 위하여 사용되는 입력 JAX-RPC 매핑 파일을 지정한다.
-package <package_name>	WSDL의 모든 Namespace URI에 대한 Java 패키지 이름을 지정한다.

파라미터	설명
-ns2pkg NS=PKG	Namespace URI에 대한 Java 패키지 이름을 지정한다. <ul style="list-style-type: none"> <li>• NS: Namespace URI</li> <li>• PKG: Java 패키지 이름</li> </ul>
-outputmapping <filename>	입력 WSDL에 대한 출력 JAX-RPC 매핑 파일을 지정한다.
-compile	생성된 Java 파일을 컴파일한다.
-nokeepsrc	생성된 Java 소스 파일을 유지하지 않고 삭제한다.
-nowrapped	WSDL에 대한 wrapped 모드 탐지를 사용할 수 없게 한다.
-datahandleronly	MIME 타입에 대하여 jakarta.activation.DataHandler를 적용하게 한다.
-nodatabinding	모든 WSDL message part에 대하여 jakarta.xml.soap.SOAPElement를 적용하게 한다.
-soapver <VER>	stub/tie 클래스에서 사용되는 SOAP 버전을 지정한다. <ul style="list-style-type: none"> <li>• 11: SOAP 1.1</li> <li>• 12: SOAP 1.2</li> </ul>
-resolvedir <directory>	원격의 WSDL 파일(스키마 파일 포함)을 접근 가능한 저장소에 저장하고자 할 때 사용한다.
-username	WSDL-URI에 접근하기 위한 사용자명이다.
-password	WSDL-URI에 접근하기 위한 패스워드이다.
-level <LEVEL>	로그 레벨을 지정한다.
-verbose	로그를 화면에 출력한다.

#### • 예제

wsdl2java는 JEUS\_HOME/bin/ 디렉터리에 위치한 일반적인 스크립트 파일이다.

wsdl2java 툴을 실행하는 예는 다음과 같다.

```
$ wsdl2java -gen:client -d build/classes
-outputmapping build/classes/web/WEB-INF/jaxrpcmapping.xml
-compile http://localhost:8088/AddNumbers/addnumbers?wsdl
```

위의 명령어는 원격의 "http://localhost:8088/AddNumbers/addnumbers?wsdl" WSDL 문서를 통해 웹 서비스 클라이언트를 위한 Stub 및 서비스 인터페이스들을 생성한다.

여기에서 Stub 및 서비스 인터페이스들은 "build/classes" 디렉터리에 생성되며 JAX-RPC 매핑 파일은 "build/classes/web/WEB-INF"에 생성된다. [-compile] 옵션으로 생성되는 Stub 및 서비스 인터페이스들은 컴파일이 된다.

## 4.14. wsdl2uddi

WSDL은 추상적인 인터페이스와 임의의 네트워크 서비스의 프로토콜 바인딩들을 제공함으로써 UDDI 표준을 보완할 수 있다. JEUS UDDI WSDL Publishing은 WSDL2UDDI을 제공한다. wsdl2uddi 툴은 웹 서비스의 WSDL 파일로부터 UDDI에 Publish할 수 있다.

다음은 툴 사용법, 파라미터 및 예제에 대한 설명이다.

- 사용법

```
Usage: wsdl2uddi UDDIVersion WSDLURI [wsdl-options]
        -uddiInquiry <UDDIInquiryURI> -uddiPublish <UDDIPublish>
        -uddiUsername <UDDIUsername> -uddiPassword <UDDIPassword> [options]

* wsdl-options
  -wsdlUsername      username to access the WSDL-URI
  -wsdlPassword      password to access the WSDL-URI

* options
  -level <LEVEL>     specify log level.
                     LEVEL : SEVERE, WARNING, INFO, FINE, FINER, FINEST
  -verbose           same as -level FINE
```

- 파라미터

파라미터	설명
UDDIVersion	UDDI의 버전을 설정한다. 입력값은 v2 또는 v3이다.
WSDLURI	실제 WSDL이 존재하는 곳의 URI 값이다.
-uddiInquiry <UDDIInquiryURI>	실제 UDDI의 Inquiry URI 값을 설정한다.
-uddiPublish <UDDIPublish>	실제 UDDI의 publish URI 값을 설정한다.
-uddiUsername <UDDIUsername>	UDDI에 접근할 때 필요한 사용자명을 설정한다.
-uddiPassword <UDDIPassword>	UDDI에 접근할 때 필요한 패스워드를 설정한다.
-wsdlUsername	WSDL URL에 접근할 때 필요한 사용자명을 설정한다.
-wsdlPassword	WSDL URL에 접근할 때 필요한 패스워드를 설정한다.
-level <LEVEL>	로그 레벨을 지정할 경우 사용한다.
-verbose	로그 레벨 'FINE'와 같다.

- 예제

wsdl2uddi는 JEUS\_HOME/bin/ 디렉터리에 위치한 일반적인 스크립트 파일이다.

wsdl2uddi 툴을 실행하는 예는 다음과 같다.

```
$ wsdl2uddi v3 http://localhost:8088/AddNumbers/addnumbers?wsdl
-uddiInquiry http://localhost:8088/uddi/inquiry
-uddiPublish http://localhost:8088/uddi/publish
-uddiUsername jeus
-uddiPassword jeus
-verbose
```

위의 명령어는 원격의 "http://localhost:8088/AddNumbers/addnumbers?wsdl" WSDL 문서를 통해 UDDI에 Publish한다. 여기에서 UDDI의 Inquiry 및 Publish URI와 사용자명, 패스워드를 각각 지정해줄 수 있다.

## 4.15. xjc

xjc 툴은 XML 스키마 파일을 Java 프로그래밍 언어로 된 JAXB Content 클래스들로 변환한다.

다음은 툴 사용법, 파라미터 및 예제에 대한 설명이다.

- 사용법

```
Usage: xjc [-options ...] <schema file/URL/dir/jar> ... [-b <bindinfo>] ...
If dir is specified, all schema files in it will be compiled.
If jar is specified, /META-INF/sun-jaxb.episode binding file will be compiled.
Options:
  -nv                : do not perform strict validation of the input schema(s)
  -extension         : allow vendor extensions
                     - do not strictly follow the Compatibility Rules and App E.2 from the
JAXB Spec
  -b <file/dir>      : specify external bindings files each <file> must have its own -b).
                     If a directory is given, **/*.xjb is searched
  -d <dir>           : generated files will go into this directory
  -p <pkg>           : specify the target package
  -httpproxy <proxy> : set HTTP/HTTPS proxy. Format is [user[:password]@]proxyHost:proxyPort
  -httpproxyfile <f> : Works like -httpproxy but takes the argument in a file to protect
password
  -classpath <arg>   : specify where to find user class files
  -catalog <file>    : specify catalog files to resolve external entity references
                     support TR9401, XCatalog, and OASIS XML Catalog format.
  -readOnly          : generated files will be in read-only mode
  -npa               : suppress generation of package level annotations (**/package-info.java)
  -no-header         : suppress generation of a file header with timestamp
  -target (2.0|2.1)  : behave like XJC 2.0 and generate code that doesnt use any 2.1 features.
  -encoding <encoding> : specify character encoding for generated source files
  -enableIntrospection : enable the correct generation of Boolean getters and setters to enable
Bean Introspection APIs.
  contentForWildcard : disable XML security features when parsing XML documents.
  -xmlschema         : treat input as W3C XML Schema (default)
  -relaxng           : treat input as RELAX NG (experimental,unsupported)
  -relaxng-compact   : treat input as RELAX NG compact syntax experimental,unsupported)
  -dtd               : treat input as XML DTD (experimental,unsupported)
  -wsdl              : treat input as WSDL and compile schemas inside it
(experimental,unsupported)
```

```

-verbose      : be extra verbose
-quiet        : suppress compiler output
-help         : display this help message
-version      : display version information
-fullversion  : display full version information

Extensions:
-Xlocator     : enable source location support for generated code
-Xsync-methods : generate accessor methods with the 'synchronized' keyword
-mark-generated : mark the generated code as @javax.annotation.Generated
-episode <FILE> : generate the episode file for separate compilation
-Xpropertyaccessors : Use XmlAccessType PROPERTY instead of FIELD for generated classes

```

## • 파라미터

파라미터	설명
-nv	스키마 유효성 검사를 할지 여부를 설정한다. (true   false)
-extension	XJC 바인딩 컴파일러는 확장 모드로 실행할지 여부를 설정한다. (true   false, 기본값: false)
-b <file/dir>	처리할 하나 이상의 외부 바인딩 파일들을 설정한다.
-d <dir>	XJC 바인딩 컴파일러는 기본값으로 현재 디렉터리에 Java Content 클래스들을 생성한다. 이 옵션을 사용하면 다른 디렉터리에 클래스들을 생성할 수 있다.
-p <pkg>	이곳에 명시한 Target 패키지 이름은 어떤 다른 Customization도 Overwrite한다.
-httpproxy <proxy>	HTTP/HTTPS 프록시를 설정한다.
-httpproxyfile <f>	-httpproxy 옵션과 유사하지만 암호를 보호하기 위해 파일에서 인수를 사용한다.
-classpath <arg>	<jxb:javaType>와 <xjc:superClass> Customization에 등록된 클라이언트 애플리케이션 클래스들을 찾을 곳을 설정한다.
-catalog <file>	TR9401, XCatalog, OASIS XML Catalog 포맷에 관한 외부 카탈로그 파일을 설정한다.
-readOnly	true일 경우에 Java 소스 파일들은 읽기 가능 모드이다. (기본값: false)
-npa	package-info.java라는 파일에 패키지 수준의 Annotation 생성을 요약한다.
-no-header	시간 소인이 있는 파일 헤더 생성을 억제하도록 지정한다.
-target (2.0 2.1)	XJC 2.0 또는 2.1과 같이 행동하고, 2.2 기능을 사용하지 않는 코드를 생성한다.
-xmlschema	입력은 W3C XML 스키마로 처리하도록 지정한다.  -relaxng, -relaxng-compact, -dtd, -wsdl와 -xmlschema의 5개 중에 -xmlschema이 기본값이다.
-relaxng	입력을 RELAX NG로 처리하도록 지정한다.
-relaxng-compact	입력을 RELAX NG의 간단한 구문으로 처리하도록 지정한다.
-dtd	XML DTD 파일을 컴파일러의 입력 스키마로 사용한다.
-wsdl	WSDL 파일을 입력하여 그 안에 사용되고 있는 스키마 파일들을 컴파일한다.

파라미터	설명
-verbose	Verbose 메시지를 출력한다.
-quiet	진행 정보 또는 경고 메시지와 같은 컴파일러 Output 메시지들을 보여주지 않는다.
-help	컴파일러 사용법을 출력한다.
-version	컴파일러 버전을 출력한다.

- 예제

xjc는 JEUS\_HOME/bin/ 디렉터리에 위치한 일반적인 스크립트 파일이다.

xjc 툴을 실행하는 예는 다음과 같다.

```
$ xjc -d build/classes -p com.tmaxsoft src/conf/ts.xsd
```

위의 명령어는 ts.xsd 스키마 파일로부터 패키지명이 "com.tmaxsoft"인 컴파일된 Java 클래스 파일들을 "build/classes"에 생성한다.

# Part III. Ant Tasks



## 5. Ant Task

본 장에서는 Ant Task에 대해서 설명한다.

### 5.1. 개요

JEUS에서는 애플리케이션을 개발하면서 특정 작업을 자동화하기 위해 다음과 같은 Ant Task를 제공한다. 사용자는 Ant를 통해 실행해야 할 Task를 기술한 Build 파일을 작성해야 한다.

본 절에서는 각 기능들의 Build 파일에 기술되어야 할 내용들에 대해서 설명하고, 이를 통해 각 Ant Task를 실행하는 과정을 설명한다. Ant에 대한 설정법과 사용법은 [Apache Ant 공식 문서](#)를 참조한다.

#### Build 파일 : build.xml

build.xml은 Ant Task를 기술한 XML 형식의 Ant Build 파일이다. Build 파일의 중요한 부분은 <project>에 포함되어 있으며, 일반적으로 1개의 프로젝트에 여러 개의 <target>이 존재한다. <target>은 실제 실행해야 하는 Ant Task 작업을 정의한다.

JEUS에서 제공하는 Ant Task를 사용하기 위해서는 다음과 같이 Task definition을 build.xml에 추가해야 한다.

```
<taskdef resource="jeus/util/ant/jeusant.properties">
  <classpath>
    <path refid="jeus.libraries"/>
  </classpath>
</taskdef>
```

프로젝트의 속성들은 <project>의 하위에 정의하며, "\${property-name}"의 표현식을 통해 프로젝트 내에서 속성 값들을 사용할 수 있다.



JEUS에서 제공하는 Ant Task를 이용할 때는 가급적 Build 파일의 project에 "jeus.home"을 설정해주는 것이 안전하다. 이는 하나의 project에서 다수의 서로 다른 target을 수행할 때 특정 Target은 JEUS\_HOME 정보를 필요로 하는 것이 존재할 수 있기 때문이다.

JEUS\_HOME은 static final 변수여서 하나의 JVM에서 처음 설정되는 값으로 고정되는 특징이 있다. 만약 JEUS\_HOME 정보가 필요없는 Target이 먼저 수행되면, JEUS\_HOME에 null 값이 설정된다. 다음 순서로 JEUS\_HOME이 필요한 Target이 수행될 때 앞의 Target이 설정한 JEUS\_HOME의 값을 이용하게 되므로 원하는 동작과 다르게 동작할 수 있다.

### 5.2. Setup Ant Task

본 절에서는 JEUS 환경을 구성하는 Ant Task를 설명한다.

## 5.2.1. create-domain

create-domain Ant Task를 통해 새로운 도메인 환경을 구성할 수 있다.

JEUS\_HOME/setup 하위에 제공되는 build.xml에 존재하는 Task로써 다음 파일에서 기본 설정 값을 확인할 수 있다.

```
JEUS_HOME/setup/domain-config-template.properties
```

다음은 create-domain Task의 속성에 대한 설명이다.

속성	설명
domain	환경을 구성할 도메인의 이름이다. (기본값: jeus_domain)
servername	JEUS Master Server의 이름이다. (기본값: adminServer)
nodename	JEUS Master Server에서 사용하는 node 이름이다. (기본값: node1)
jeus.address	JEUS Master Server의 기본 동작 주소이다. (기본값: 0.0.0.0 )
jeus.port	JEUS Master Server의 기본 동작 포트이다. (기본값: 9736)
jms.port	JEUS Master Server의 JMS 동작 포트이다. (기본값: 9741)
http.port	JEUS Master Server의 HTTP 동작 포트이다. (기본값: 8808)
jvm.config	JEUS Master Server의 JVM 설정이다.  (기본값: -Xmx1024m -XX:MaxPermSize=128m)
jeus.username	도메인의 관리 계정 이름이다. 도메인의 관리 계정 이름으로 반드시 변경할 것을 권장한다. (기본값: administrator)
jeus.password	도메인의 관리 계정 패스워드로 평문으로는 jeus이며 base64로 인코딩된 값이다. Ant 설정을 통해 변경하거나 jeusadmin의 set-password 명령어를 통해 반드시 변경할 것을 권장한다. (기본값: {base64}amV1cw==)
usevirtualmulticast	SCF(System Clustering Framework)를 사용할 때 TCP 통신을 이용한 가상 멀티캐스트를 사용할지 여부를 설정한다. (true   false, 기본값: false)
heartbeataddress	SCF에 사용되는 heartbeat 주소이다.  usevirtualmulticast 값이 true일 경우 적용되지 않는다. (기본값: 230.30.1.1)
heartbeatport	SCF에 사용되는 heartbeat 포트이다.  usevirtualmulticast 값이 true일 경우 적용되지 않는다. (기본값: 3030)

### 예제

다음은 Ant Task를 통한 도메인 생성 예제이다.

도메인의 이름은 'domain1', JEUS Master Server의 이름은 'adminServer', JEUS Master Server의 jvm 설정은 '-Xmx512m -XX:MaxPermSize=256m'로 설정한 예제이다. 모두 기본값이 존재하므로 굳이 옵션을 모두 줄 필요는

없으나 아래의 예제에는 방법을 설명하기 위해 옵션을 설정하였다.

```
JEUS_HOME/setup$ ant create-domain -Ddomain=domain1 -Dservername=adminServer
-Djvm.config="-Xmx512m -XX:MaxPermSize=256m" -Dheartbeataddress=230.30.1.1
-Dheartbeatport=3030
Buildfile: JEUS_HOME/setup/build.xml

create-domain:
[echo] Creating a domain configuration: domainindir="JEUS_HOME/domains/domain1
domain = domain1, server-name = adminServer, admin password={base64}amV1cw==,
server base port=9736, server base listen address=${jeus.address}
[mkdir] Created dir: JEUS_HOME/domains/domain1
[mkdir] Created dir: JEUS_HOME/domains/domain1/.applications
[mkdir] Created dir: JEUS_HOME/domains/domain1/.deploymentplans
[mkdir] Created dir: JEUS_HOME/domains/domain1/bin
[mkdir] Created dir: JEUS_HOME/domains/domain1/config
[mkdir] Created dir: JEUS_HOME/domains/domain1/lib
[mkdir] Created dir: JEUS_HOME/domains/domain1/lib/application
[mkdir] Created dir: JEUS_HOME/domains/domain1/servers
[mkdir] Created dir: JEUS_HOME/domains/domain1/config/servlet
[mkdir] Created dir: JEUS_HOME/domains/domain1/config/security
[copy] Copying 2 files to JEUS_HOME/domains/domain1/config
[copy] Copying 1 file to JEUS_HOME/domains/domain1/config/servlet
[copy] Copying 1 file to JEUS_HOME/domains/domain1/config/security
[copy] Copied 1 empty directory to 1 empty directory under JEUS_HOME/domains/domain1/config/security
[copy] Copying 3 files to JEUS_HOME/domains/domain1/config/security
[echo] productionmode = true
[java] JVM args ignored when same JVM is used.
[copy] Copying 3 files to JEUS_HOME/domains/domain1/bin
[mkdir] Created dir: JEUS_HOME/domains/domain1/servers/adminServer
[mkdir] Created dir: JEUS_HOME/domains/domain1/servers/adminServer/bin
[mkdir] Created dir: JEUS_HOME/domains/domain1/servers/adminServer/.workspace
[mkdir] Created dir: JEUS_HOME/domains/domain1/servers/adminServer/.workspace/deployed
[mkdir] Created dir:
JEUS_HOME/domains/domain1/servers/adminServer/.workspace/deployed/deploymentplans
[mkdir] Created dir: JEUS_HOME/domains/domain1/servers/adminServer/logs
[mkdir] Created dir: JEUS_HOME/domains/domain1/servers/adminServer/lib
[mkdir] Created dir: JEUS_HOME/domains/domain1/servers/adminServer/lib/application

BUILD SUCCESSFUL
Total time: 2 seconds
JEUS_HOME/setup$
```

## 5.3. EJB Ant Task

JEUS를 기반으로 EJB 컴포넌트를 개발할 때 작업을 편리하게 수행할 수 있도록 몇 가지의 Ant Task를 제공한다.

### 5.3.1. appcompiler

appcompiler Ant Task는 pre-deployment 작업 후 EJB 모듈 또는 개별적인 EJB Bean에 대해서 필요한 RMI Stub과 Skeleton 클래스를 임의로 생성하는 경우 사용한다. 또한 이것은 클라이언트에서 원격지에 있는 디플로이된 Bean과 통신할 때 필요한 클래스 파일들 즉 EJB 클라이언트 JAR를 생성할 수 있다.

다음은 appcompiler Ant Task 속성에 대한 설명이다.

속성	설명
jeusHome	JEUS_HOME을 설정한다. (String 타입)
client	생성될 Stub 클래스들이 포함될 클라이언트 뷰 파일을 설정한다. (String 타입)
keep	컴파일 과정에서 생성된 소스를 유지할 것인지를 설정한다. (기본값: false)
jspmap	servlet-mapping table(jeus_jspmap.xml)을 생성할 것인지를 설정한다. (기본값: false)
ejbJar	컴파일할 때 사용할 ejb-jar.xml 파일을 설정한다. (String 타입)
jeusEjbDd	컴파일할 때 사용할 jeus-ejb-dd.xml 파일을 설정한다. (String 타입)
name	deploy에 사용할 이름을 설정한다. (String 타입)
target	컴파일할 대상 애플리케이션, stand-alone 모듈 파일을 설정한다. (필수 입력항목, String 타입)

## 예제

다음은 build.xml을 작성한 예제와 build.xml을 통해 appcompiler를 실행하는 예제이다.

### • build.xml 작성 예제

appcompiler Ant Task build 파일 예제 : <build.xml>

```
<?xml version="1.0"?>

<project name="example" default="appcompiler" basedir=".>
  <property environment="env"/>
  <!-- jeus.home project property is required
       when you run the various tasks in an ant jvm and
       one of the various tasks requirs JEUS_HOME information -->
  <property name="jeus.home" value="${env.JEUS_HOME}"/>

  <!-- set properties to be needed for appcompiler task -->
  <property name="client" value="client_view.jar"/>
  <property name="keep" value="false"/>
  <property name="jspmap" value="false"/>
  <property name="ejbJar" value="ejb-jar.xml"/>
  <property name="jeusejbdd" value="jeus-ejb-dd.xml"/>
  <property name="targetfile" value="ejb"/>

  <!-- set the library-classpath or to run the task class -->
  <path id="jeus.libraries">
    <fileset dir="${jeus.home}/lib/system" includes="*.jar"/>
  </path>

  <!-- include the task definition resource file -->
  <taskdef resource="jeus/util/ant/jeusant.properties">
    <classpath>
      <path refid="jeus.libraries"/>
    </classpath>
  </taskdef>
```

```

<target name="init">
</target>

<!-- appcompiler task -->
<target name="appcompiler" depends="init">
  <appcompiler jeusHome="${jeus.home}"
    client="${client}"
    keep="${keep}"
    jspmap="${jspmap}"
    ejbJar="${ejbjar}"
    jeusEjbDd="${jeusejbdd}"
    target="${targetfile}"/>
</target>

</project>

```

## • 실행 예제

```

$ ant appcompiler
Buildfile: build.xml

init:

appcompiler:

BUILD SUCCESSFUL
Total time: 3 minutes 7 seconds

```

## 5.3.2. ejbddinit

ejbddinit Ant Task는 JEUS EJB DD를 생성할 때 이용한다. ejbddinit에 대한 자세한 설명은 [ejbddinit](#)을 참고한다.

다음은 ejbddinit Ant Task의 속성에 대한 설명이다.

속성	설명
propertyFile	ejbddinit 실행할 때 참조할 프로퍼티 파일을 지정한다. (String 타입)
logginginglevel	ejbddinit 실행할 때 화면에 출력되는 로그의 레벨을 지정한다.  이 레벨은 Java SE logging API의 레벨을 따른다. (기본값: INFO)
target	ejbddint 실행 대상이 되는 EJB 모듈의 경로를 지정한다. EJB 모듈은 JAR archive 또는 디렉터리일 수 있다.  (필수 입력항목, 프로퍼티 파일이 Target 정보를 설정한 경우 생략 가능)

ejbddinit Ant Task 스크립트에 제한적으로 ejbddinit 프로퍼티를 설정할 수 있다.

ejbddinit 프로퍼티 파일의 프로퍼티는 ejbddinit Ant Task 스크립트에서 다음과 같은 규칙으로 설정되어야 한다. ejbddinit 프로퍼티에 대한 자세한 내용은 [프로퍼티 목록](#)을 참고한다.

ejbddinit 프로퍼티 파일	ejbddinit Ant 스크립트
export-name=%{ejb-class}	<property name="export-name" value="%{ejb-class}"/>
thread-max=100	<property name="thread-max" value="100"/>
HelloBean.export-port=55555	build.xml에서는 특정 EJB 컴포넌트(HelloBean)에 대한 ejbddinit 프로퍼티 설정이 불가능

ejbddinit 프로퍼티 파일과 같이 Ant 스크립트에서도 export-name 프로퍼티에 대하여 "%{ejb-class}"와 같은 패턴을 사용할 수 있다. export-name과 관련된 패턴 사용에 관한 내용은 [export-name 프로퍼티에 대한 패턴 지원](#)을 참고한다.

Ant 스크립트에서는 특정 EJB 컴포넌트에 대한 ejbddinit 프로퍼티 설정은 지원하지 않으므로 특정 EJB 컴포넌트에 대한 ejbddinit 프로퍼티 설정이 필요한 경우에는 ejbddinit 프로퍼티 파일을 따로 작성해야 한다. 특정 EJB 컴포넌트에 대한 설정은 ejbddinit 프로퍼티 파일과 Ant 스크립트의 공통 설정보다 우선한다. ejbddinit 프로퍼티 파일과 ejbddinit Ant 스크립트에 동일한 설정이 존재할 경우에는 Ant 스크립트의 설정을 우선한다.

## 예제

ejbddinit Ant 스크립트 작성 예와 ejbddinit Ant Task를 실행하는 예는 아래와 같다.

### • ejbddinit Ant 스크립트 작성 예

ejbddinit Ant 스크립트 예제

```
<?xml version="1.0"?>
<project name="example" default="ejbddinit" basedir=".">
  <property environment="env"/>
  <!-- jeus.home project property is required
       when you run the various tasks in an ant jvm and
       one of the various tasks requirs JEUS_HOME information -->
  <property name="jeus.home" value="${env.JEUS_HOME}"/>

  <!-- set properties to be needed for ejbddinit task -->
  <property name="targetfile" value="ejb.jar"/>
  <property name="logginglevel" value="FINE"/>
  <property name="propertyfile" value="ejbddinit.properties"/>

  <!-- set properties to be needed for ejbddinit properties -->
  <property name="export-name" value="%{ejb-class}"/>
  <property name="thread-max" value="100"/>

  <!-- set the library-classpath or to run the task class -->
  <path id="jeus.libraries">
    <fileset dir="${jeus.home}/lib/system" includes="*.jar"/>
  </path>

  <!-- include the task definition resource file -->
  <taskdef resource="jeus/util/ant/jeusant.properties">
    <classpath>
      <path refid="jeus.libraries"/>
    </classpath>
  </taskdef>

  <target name="init">
```

```

</target>

<!-- ejbddinit task -->
<target name="ejbddinit" depends="init">
    <ejbddinit loggingLevel="${logginglevel}"
               property="${propertyfile}"
               target="${targetfile}"
               exportName="${export-name}"
               threadMax="${thread-max}">
    </ejbddinit>
</target>
</project>

```

#### • ejbddinit Ant Task 실행 예

```

$ ant ejbddinit
Buildfile: build.xml

init:

ejbddinit:
[ejbddinit] LoadFile: /jeus/sample/ejbddinit.properties
[ejbddinit] Source=/jeus/sample/ejbddinit/ejb.jar
[ejbddinit] Successfully configured the parameters.
[ejbddinit] Deployment descriptor initialization started.
[ejbddinit] Creating JEUS descriptors.
[ejbddinit] Deployment descriptor initialization finished.

BUILD SUCCESSFUL
Total time: 2 seconds

```

## 5.4. 웹 서비스 Ant Task

본 절에서는 JEUS에서 웹 서비스 생성과 웹 서비스 클라이언트를 위해 제공하는 Ant Task에 대해서 설명한다.

### 5.4.1. java2wsdl

java2wsdl Task는 SET(Service Endpoint Interface) 클래스와 임의의 Java로 구현한 클래스로부터 다음의 파일을 생성한다.

- 웹 서비스의 WSDL파일
- JAX-RPC 매핑 파일



java2wsdl Ant Task를 정의하는 클래스는 jeus.util.ant.webservices.Java2WsdTask이다.

다음은 java2wsdl의 속성에 대한 설명이다.

속성	설명
configfilepath	웹 서비스 설정 파일의 경로를 설정한다. (필수 입력항목, String 타입)
classpath	웹 서비스를 구성하는 컴파일된 Java 클래스에 대한 클래스 패스를 설정한다. (필수 입력항목, String 타입)
destDir	생성된 WSDL 파일이 놓일 디렉터리의 절대 경로를 설정한다. (String 타입)
level	로그 레벨을 설정한다. (String 타입)
verbose	Verbose 메시지를 출력 여부를 설정한다. (기본값: false)

## Nested Element

<java2wsdl>은 Ant의 <classpath> element를 가지고 있다.

## 예제

build.xml을 작성한 예제와 이 build.xml을 통해 java2wsdl을 실행시키는 예를 설명한다.

### • build.xml 작성 예제

java2wsdl Ant Task build 파일 예제 : <build.xml>

```
<?xml version="1.0" encoding="UTF-8"?>
<project name="java2wsdl" default="build" basedir=".">
  <property name="is.app-client.module" value="true" />
  <import file="../../common/common-build.xml" />
  <taskdef name="java2wsdl"
    classname="jeus.util.ant.webservices.Java2Wsd1Task">
    <classpath refid="jeus.libraries.classpath" />
  </taskdef>
  <target name="-post-compile">
    <java2wsdl destDir="${build.classes.dir}"
      verbose="true"
      configfilepath="${src.conf}/service-config.xml">
      <classpath refid="classpath" />
    </java2wsdl>
  </target>
</project>
```

### • 실행 예제

```
$ jant
...

[java2wsdl] Building Web Services : DocLitEchoService
[java2wsdl] Generating WSDL File - jeus/build/classes/DocLitEchoService.wsdl
[java2wsdl] Generating JAX-RPC Mapping File - jeus/build/classes/DocLitEchoService-mapping.xml
...
```



## 5.4.2. wsdl2java

wsdl2java Task는 웹 서비스의 WSDL로부터 다음의 2개 중에 하나를 생성한다.

- 클라이언트 측 웹 서비스의 Stub Java 소스 코드들
- 서버 측 웹 서비스의 인터페이스 Java 소스 코드들



wsdl2java Ant Task를 정의하는 클래스는 'jeus.util.ant.webservices.Wsdl2JavaTask'이다.

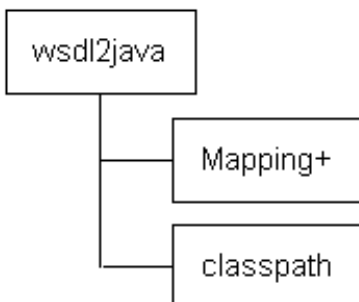
다음은 wsdl2java의 속성에 대한 설명이다.

속성	설명
wsdl	Java 소스 파일을 생성하기 위해서 사용되는 WSDL의 URL이나 절대 경로를 설정한다. (필수 입력항목, String 타입)
mode	Java 소스 파일을 생성하기 위해서 사용되는 MODE를 설정한다. (필수 입력항목, String 타입)  다음 중에 하나를 설정한다. <ul style="list-style-type: none"><li>◦ gen:client</li><li>◦ gen:server</li><li>◦ gen</li><li>◦ import:client</li><li>◦ import:server</li><li>◦ import</li></ul>
destDir	생성된 Java 파일이 놓일 디렉터리의 절대 경로를 설정한다.  (필수 입력항목, String 타입)
classDestDir	컴파일된 클래스 파일이 생성될 디렉터리를 설정한다. (String 타입)
inputMapping	Java 클래스를 생성하기 위하여 사용되는 입력 JAX-RPC 매핑 파일을 설정한다. (String 타입)
package	WSDL의 모든 Namespace URI에 대한 Java 패키지 이름을 설정한다.  (String 타입)
outputMapping	입력 WSDL에 대한 출력 JAX-RPC 매핑 파일을 설정한다. (String 타입)
doCompile	true일 때 생성된 Java 소스 파일들은 컴파일된다. (기본값: true)
username	WSDL의 URL로 접근할 때 필요한 사용자 이름이다. (String 타입)

속성	설명
password	WSDL의 URL로 접근할 때 필요한 패스워드이다. (String 타입)
keepSrc	true로 설정하면 생성된 Java 소스 파일들은 삭제되지 않는다. (기본값: true)
nowrapped	true로 설정하면 WSDL에 대한 wrapped MODE 탐지를 사용할 수 없다. (기본값: false)
dataHandlerOnly	true로 설정하면 MIME 타입에 대하여 jakarta.activation.DataHandler를 적용한다. (기본값: false)
nodatbinding	모든 WSDL message part에 대하여 jakarta.xml.soap.SOAPElement를 적용한다. (기본값: false)
soapver	stub/tie 클래스에서 사용되는 SOAP 버전을 설정한다.  ◦ 11 : SOAP 1.1(기본값) ◦ 12 : SOAP 1.2
resolveDir	원격의 WSDL 파일(스키마 포함)을 로컬 저장소에 저장할 때 값을 설정할 수 있다. (String 타입)
DDGen	JEUS의 기본적인 배치 서술자(web services.xml 파일과 MODE가 web인 경우 web.xml, ejb인 경우 ejb-jar.xml 파일)를 생성할 때 사용한다. (String 타입)
level	로그 레벨을 설정한다. (String 타입)
verbose	Verbose 출력은 true일 때 가능하다. (기본값: false)

## Nested Element

<wsdl2java>는 중첩된 Ant의 <classpath>와 <mapping> element를 가지고 있다. <wsdl2java> Ant Task의 구조는 다음과 같다(+는 한 개 이상의 element를 가질 수 있음을 의미한다).



wsdl2java Task 구성

## <mapping>

Java 패키지과 WSDL Namespace 사이의 매핑이다. 사용자가 <mapping>을 생략하면 모든 Namespace URI는 <wsdl2java> element인 package attribute에 기술된 패키지로 매핑된다.

다음은 <mapping>의 속성에 대한 설명이다.

속성	설명
package	Java 패키지의 이름이다. (필수 입력항목, String 타입)
namespace	WSDL의 Namespace URI이다. (필수 입력항목, String 타입)



만약 사용자가 <wsdl2java> element의 package 속성을 설정하면 이 속성 값이 <mapping> element로 설정된 값에 우선한다. 즉, <mapping> element로 설정된 값은 무의미해진다.

## 예제

build.xml을 작성한 예제와 이 build.xml을 통해 wsdl2java를 실행하는 예제를 설명한다.

### • build.xml 작성 예제

wsdl2java Ant Task build 파일 예제 : <build.xml>

```
<?xml version="1.0" encoding="UTF-8"?>
<project name="wsdl2java" default="build" basedir=". ">
  <property name="is.app-client.module" value="true" />
  <import file="../../common/common-build.xml" />
  <taskdef name="wsdl2java"
    classname="jeus.util.ant.webservices.Wsdl2JavaTask">
    <classpath refid="jeus.libraries.classpath" />
  </taskdef>
  <target name="-pre-compile">
    <mkdir dir="${build.classes.dir}" />
    <wsdl2java destDir="${build.classes.dir}"
      verbose="true"
      mode="import:server"
      doCompile="true"
      noDataBinding="true"
      package="sample.nodatabinding.service"
      outputmapping="${build.classes.dir}/BookQuoteService-mapping.xml"
      wsdl="${src.conf}/BookQuoteService.wsdl">
      <classpath refid="classpath" />
    </wsdl2java>
  </target>
</project>
```

### • 실행 예제

```
$ jant
Buildfile: build.xml
...
[wsdl2java] Compiling generated sources(1)...
...
BUILD SUCCESSFUL
Total time: 6 seconds
```

### 5.4.3. wsgen

wsgen Task는 service endpoint interface 클래스(그리고 임의의 Java로 구현한 클래스)로부터 다음을 생성한다.

- Portable Artifacts
- 웹 서비스의 WSDL파일(옵션)



wsgen Ant Task를 정의하는 클래스는 'jeus.webservices.jaxws.tools.WsGen2' 이다.

다음은 wsgen 속성에 대한 설명이다.

속성	설명
sei	service endpoint interface 클래스 이름을 설정한다. (필수 입력항목, String 타입)
destdir	생성된 클래스 파일이 놓일 디렉터리의 절대 경로를 설정한다. (필수 입력항목, String 타입)
classpath	입력으로 들어오는 클래스 파일들을 찾을 위치를 설정한다. (String 타입)
cp	classpath 속성과 같다. (String 타입)
resourcedestdir	genwsdl 속성과 함께 사용한다. WSDL 파일을 생성할 위치를 설정한다. (String 타입)
sourcedestdir	생성되는 소스 파일들의 위치를 설정한다. (String 타입)
keep	생성되는 파일들을 보존 여부를 설정한다. (기본값: false)
verbose	Verbose 메시지를 출력 여부를 설정한다. (기본값: false)
genwsdl	WSDL 파일이 생성 여부를 설정한다. (기본값: false)
protocol	genwsdl 속성과 함께 사용한다.  wsdl:binding Element에 사용될 프로토콜을 설정한다.  기본값은 soap1.1이며 Xsoap1.2 Xsoap1.2와 같은 값들이 가능하다. Xsoap1.2는 표준은 아니며 extension 속성과 함께 쓰일 수 있다. (String 타입)
servicename	genwsdl 속성과 함께 사용한다. 생성되는 WSDL 파일의 특정 wsdl:service Element의 이름을 설정할 때 사용한다. (String 타입)
portname	genwsdl 속성과 함께 사용한다. 생성되는 WSDL 파일의 특정 wsdl:portname Element의 이름을 설정할 때 사용된다. (String 타입)
extension	벤더 특정의 확장성을 설정할 때 사용한다. 상호 운용 및 이식성이 떨어질 수 있다. (Boolean 타입)
policy	웹 서비스 정책 설정 파일을 읽어 들인다. (String 타입)

#### 예제

build.xml을 작성한 예제와 이 build.xml을 통해 wsgen을 실행하는 예제이다.

- **build.xml 작성 예제**

wsgen Ant Task build 파일 예제 : <build.xml>

```
<?xml version="1.0" encoding="UTF-8"?>
<project name="wsgen" default="build" basedir=".">
  <property name="is.app-client.module" value="true" />
  <import file="../../common/common-build.xml" />
  <taskdef name="wsgen" classname="com.sun.tools.ws.ant.WsGen">
    <classpath refid="jeus.libraries.classpath" />
  </taskdef>
  <target name="-post-compile">
    <wsgen sei="fromjava.server.AddNumbersImpl"
      destdir="${build.classes.dir}"
      classpath="${build.classes.dir}"
      resourcedestdir="${build.classes.dir}"
      sourcedestdir="${build.classes.dir}"
      genwsdl="true" />
  </target>
</project>
```

- **실행 예제**

```
$ jant
Buildfile: build.xml
...

    [echo] Compiling wsgen...
...

BUILD SUCCESSFUL
Total time: 6 seconds
```

## 5.4.4. wsimport

wsimport Task는 웹 서비스의 WSDL로부터 다음의 2개 중에 하나를 생성한다.

- 클라이언트 측 웹 서비스의 Stub Java 소스 코드들
- 서버 측 웹 서비스의 인터페이스 Java 소스 코드들



wsimport Ant Task를 정의하는 클래스는 'jeus.webservices.jaxws.tools.WsImport2'이다.

다음은 wsimport 속성에 대한 설명이다.

속성	설명
wsdl	Java 소스 파일을 생성하기 위해서 사용되는 WSDL의 URL이나 절대 경로를 설정한다. (필수 입력항목, String 타입)
destDir	생성된 Java 파일이 놓일 디렉터리의 절대 경로를 설정한다. (String 타입)
sourcedestdir	생성된 소스 파일들이 놓일 위치를 설정한다. 이 속성을 지정하면 keep 속성이 자동으로 설정된다. (String 타입)
keep	생성되는 파일들을 보존 여부를 설정한다. (기본값: false)
verbose	Verbose 출력은 true일때 가능하다. (기본값: false)
binding	외부의 JAX-WS 또는 JAXB 바인딩 파일들을 설정한다. (String 타입)
extension	벤더 특성의 확장성을 설정할 때 사용한다. 상호 운용 및 이식성이 떨어질 수 있다. (Boolean 타입)
wsdllocation	WSDL URI를 이 속성에 명시하면 그 URI 값이 생성되는 SEI와 서비스 인터페이스의 @WebService.wsdlLocation Annotation과 @WebServiceClient.wsdlLocation Annotation에 설정된다. (String 타입)
catalog	TR9401, XCatalog, OASIS XML Catalog 형식과 같은 외부의 Entity 참조값을 설정할 수 있다. 또한 Ant xmncatalog 타입을 사용할 수 있다. (String 타입)
package	WSDL의 모든 Namespace URI에 대한 Java 패키지 이름을 설정한다. (String 타입)
target	이곳에 지정된 JAXWS 스펙 버전에 따라 코드를 생성한다. (String 타입)
quiet	출력 메시지들을 출력 여부를 설정한다. (기본값: false)
policy	웹 서비스 정책 설정 파일을 읽어 들인다. (String 타입)

## 예제

다음은 build.xml을 작성한 예제와 build.xml을 통해 wsimport를 실행하는 예제이다.

### • build.xml 작성 예제

wsimport Ant Task build 파일 예제 : <build.xml>

```
<?xml version="1.0" encoding="UTF-8"?>
<project name="wsimport" default="build" basedir=".>
  <property name="is.app-client.module" value="true" />
  <import file="../../common/common-build.xml" />
  <taskdef name="wsimport" classname="com.sun.tools.ws.ant.WsImport">
    <classpath refid="jeus.libraries.classpath" />
  </taskdef>
  <target name="-pre-compile">
    <mkdir dir="${build.classes.dir}" />
    <wsimport wsdl="${src.conf}/AddNumbers.wsdl"
      destDir="${build.classes.dir}"
      sourcedestdir="${build.classes.dir}"
      package="fromwsdl.server" />
  </target>
</project>
```

## • 실행 예제

```
$ jant
Buildfile: build.xml
...

[wsimport] Consider using <depends>/<produces> so that wsimport won't do
unnecessary compilation
[wsimport] parsing WSDL...
[wsimport]
[wsimport]
[wsimport] generating code...
...

BUILD SUCCESSFUL
Total time: 6 seconds
```

### 5.4.5. xjc

xjc Task는 XML 스키마 파일을 Java 프로그래밍 언어로 된 JAXB Content 클래스들로 변환한다.

다음은 xjc 속성에 대한 설명이다.

속성	설명
wsdl	컴파일하려는 스키마 파일이다. (필수 입력항목, String 타입)
binding	스키마 파일에 적용될 외부 추가 바인딩 파일이다. (String 타입)
package	이 옵션을 명시하면 생성되는 코드는 이 Java 패키지 하위에 위치한다. 이 옵션은 Command Line의 -p 옵션과 동일한 효과를 갖는다. (String 타입)
destdir	생성된 코드는 이 디렉터리 하위에 생성된다. (필수 입력항목, String 타입)
readonly	true일 경우에 Java 소스 파일들은 읽기 가능 모드이다. (기본값: false)
extension	true일 경우에 XJC 바인딩 컴파일러는 확장 모드로 실행된다. (기본값: false)
catalog	TR9401, XCatalog, OASIS XML Catalog 포맷과 같은 외부 카탈로그 파일을 설정할 때 사용한다. (String 타입)
removeOldOutput	"yes" 값으로 설정되면 produces Element가 가리키는 파일들은 XJC 바인딩 컴파일러가 소스 파일들을 재컴파일하기 전에 모두 삭제된다. (String 타입, "yes"   "no")
source	사용할 스키마 컴파일러의 버전을 명시한다. (String 타입, 1.0   2.0)

## 예제

다음은 build.xml을 작성한 예제와 build.xml을 통해 xjc를 실행하는 예제이다.

### • build.xml 작성 예제

xjc Ant Task build 파일 예제 : <build.xml>

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<project name="xjc" default="build" basedir=".">
  <property name="is.app-client.module" value="true" />
  <import file="../../common/common-build.xml" />
  <taskdef name="xjc" classname="com.sun.tools.xjc.XJCTask">
    <classpath refid="jeus.libraries.classpath" />
  </taskdef>
  <target name="-pre-compile">
    <mkdir dir="${build.classes.dir}" />
    <xjc schema="${src.conf}/ts.xsd"
        package="com.tmaxsoft"
        destdir="${build.classes.dir}">
      <produces dir="${build.classes.dir}/com/tmaxsoft"
        includes="**/*.java" />
      <classpath refid="jeus.libraries.classpath" />
      <classpath refid="classpath" />
    </xjc>
  </target>
</project>
```

## • 실행 예제

```
$ jant
Buildfile: build.xml

...

[xjc] jeus/build/classes/com/tmaxsoft is not found and thus excluded from
the dependency check
[xjc] Compiling file:/jeus/src/conf/ts.xsd
[xjc] Writing output to jeus/build/classes

...

BUILD SUCCESSFUL
Total time: 4 seconds
```

## 5.4.6. schemagen

schemagen Task는 Java 클래스들에 나타나는 각각의 Namespace에 대해 하나의 스키마 파일을 생성한다.

다음은 schemagen의 속성에 대한 설명이다.

속성	설명
destdir	생성되는 스키마 파일을 위치시킬 디렉토리를 설정한다. (String 타입)
classpath	클래스 패스를 설정한다. (String 타입)

## 예제

build.xml을 작성한 예제와 build.xml을 통해 schemagen을 실행하는 예제이다.



- **build.xml 작성 예제**

schemagen Ant Task build 파일 예제 : <build.xml>

```
<?xml version="1.0" encoding="UTF-8"?>
<project name="schemagen" default="build" basedir=".>
  <property name="is.app-client.module" value="true" />
  <import file="../../common/common-build.xml" />
  <taskdef name="schemagen" classname="com.sun.tools.jxc.SchemaGenTask">
    <classpath refid="jeus.libraries.classpath" />
  </taskdef>
  <target name="-pre-compile">
    <mkdir dir="${build.classes.dir}" />
    <schemagen destdir="${build.classes.dir}">
      <src path="${src.dir}" />
      <classpath refid="jeus.libraries.classpath" />
      <classpath refid="classpath" />
    </schemagen>
  </target>
</project>
```

- **실행 예제**

```
$ jant
Buildfile: build.xml

...

[schemagen] Generating schema from 2 source files
[schemagen] Note: Writing jeus/schema1.xsd

...

BUILD SUCCESSFUL
Total time: 5 seconds
```

## 5.5. 웹 서비스 Maven Plugin

본 절에서는 JEUS에서 JAX-WS 기반의 웹 서비스 생성과 웹 서비스 클라이언트를 위해 제공하는 Maven Plugin에 대해서 설명한다.

### 5.5.1. wsgen

wsgen Plugin은 service endpoint interface 클래스(그리고 임의의 Java로 구현한 클래스)로부터 다음을 생성한다.

- Portable Artifacts
- 웹 서비스의 WSDL파일(옵션)

wsgen plugin은 JEUS\_HOME/lib/client/jeus-ws-maven-plugin.jar으로 제공된다.

wsgen plugin을 사용하기 위해서는 다음 작업이 선행되어야 한다.

1. jeus-ws-maven-plugin.jar 파일을 로컬 Repository에 install한다.
2. JDK 7 환경에서는 Maven option에 다음을 추가한다.

```
-Djava.endorsed.dirs=JEUS_HOME/lib/endorsed
```

다음은 wsgen plugin 속성에 대한 설명이다.

속성	설명
sei	SEI 클래스의 이름을 설정한다. (필수 입력항목, String 타입)
destDir	생성된 클래스 파일이 놓일 디렉터리의 절대 경로를 설정한다. (기본값: \${project.build.outputDirectory})
resourceDestDir	genwsdl 속성과 함께 사용한다. WSDL 파일을 생성할 위치를 설정한다. (기본값: \${project.build.directory}/generated-sources/wsdl)
sourceDestDir	생성되는 소스 파일들의 위치를 설정한다. (기본값: \${project.build.directory}/generated-sources/wsgen)
keep	생성되는 파일들을 보존 여부를 설정한다. (기본값: true)
verbose	Verbose 메시지를 출력 여부를 설정한다. (기본값: false)
genWsdl	WSDL 파일이 생성 여부를 설정한다. (기본값: false)
protocol	genwsdl 속성과 함께 쓰이며 wsdl:binding Element에 사용될 프로토콜을 설정한다. 기본값은 soap1.1 이며 Xsoap1.2 Xsoap1.2와 같은 값들이 가능하다. Xsoap1.2는 표준은 아니며 extension 속성과 함께 쓰일 수 있다. (String 타입)
servicename	genwsdl 속성과 함께 사용한다. 생성되는 WSDL 파일의 특정 wsdl:service Element의 이름을 설정할 때 사용한다. (String 타입)
portname	genwsdl 속성과 함께 사용한다. 생성되는 WSDL 파일의 특정 wsdl:portname Element의 이름을 설정할 때 사용된다. (String 타입)
extension	벤더 특정의 확장성을 설정할 때 사용한다. 상호 운용 및 이식성이 떨어질 수 있다. (기본값: false)

## 예제

pom.xml을 작성한 예제와 이 pom.xml을 통해 wsgen plugin을 실행하는 예제에 대한 설명이다.

### • pom.xml 작성 예제

wsgen plugin pom 파일 예제 : <pom.xml>

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
```

```

<modelVersion>4.0.0</modelVersion>

<groupId>jeus.webservices.maven.sample</groupId>
<artifactId>wsngen_sample</artifactId>
<version>0.0.1</version>
<packaging>war</packaging>

<name>${project.artifactId}</name>

<build>
  <plugins>
    <plugin>
      <groupId>jeus.ws</groupId>
      <artifactId>jeus-ws-maven-plugin</artifactId>
      <version>0.0.1</version>
      <executions>
        <execution>
          <id>wsngen_test</id>
          <goals>
            <goal>wsngen</goal>
          </goals>
          <configuration>
            <sei>jeus.webservices.sample.EchoService</sei>
            <extension>true</extension>
            <genWsdL>true</genWsdL>
            <keep>true</keep>
            <inlineSchemas>true</inlineSchemas>
          </configuration>
        </execution>
      </executions>

      <dependencies>
        <dependency>
          <groupId>com.sun</groupId>
          <artifactId>tools</artifactId>
          <version>1.6</version>
          <scope>system</scope>
          <systemPath>${java.home}/../lib/tools.jar</systemPath>
        </dependency>
        <dependency>
          <groupId>com.sun.xml.ws</groupId>
          <artifactId>jaxws-tools</artifactId>
          <version>2.2</version>
          <scope>system</scope>
          <systemPath>${jeus.home}/lib/system/jaxws-tools.jar</systemPath>
        </dependency>
      </dependencies>
    </plugin>

    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-compiler-plugin</artifactId>
      <version>2.3.2</version>
      <configuration>
        <source>1.6</source>
        <target>1.6</target>
        <encoding>UTF-8</encoding>
        <compilerArguments>
          <endorseddirs>${jeus.home}/lib/endorsed</endorseddirs>
        </compilerArguments>
      </configuration>
    </plugin>
  </plugins>
</build>

```

```

        </compilerArguments>
    </configuration>
</plugin>
</plugins>
</build>

</project>

```

## • 실행 예제

```

$ mvn -Djeus.home=${JEUS_HOME} process-classes
[INFO] Scanning for projects...
[INFO]
[INFO] -----
[INFO] Building wsgen_sample 0.0.1
[INFO] -----
[INFO]
[INFO] --- maven-resources-plugin:2.5:resources (default-resources) @ wsgen_sample ---
[INFO] skip non existing resourceDirectory /jeus/wsgen/src/main/resources
[INFO]
[INFO] --- maven-compiler-plugin:2.3.2:compile (default-compile) @ wsgen_sample ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- jeus-ws-maven-plugin:0.0.1:wsgen (wsgen_test) @ wsgen_sample ---
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 3.773s
[INFO] -----

```

## 5.5.2. wsimport

wsimport Plugin은 웹 서비스의 WSDL로부터 다음의 2개 중에 하나를 생성한다.

- 클라이언트 측 웹 서비스의 Stub Java 소스 코드들
- 서버 측 웹 서비스의 인터페이스 Java 소스 코드들

wsimport plugin은 JEUS\_HOME/lib/client/jeus-ws-maven-plugin.jar으로 제공된다. wsimport plugin을 사용하기 위해서는 다음 작업이 선행되어야 한다.

1. jeus-ws-maven-plugin.jar 파일을 로컬 Repository에 install한다.
2. JDK 7 환경에서는 maven option에 다음을 추가한다.

```
-Djava.endorsed.dirs=JEUS_HOME/lib/endorsed
```

다음은 wsimport plugin 속성에 대한 설명이다.

속성	설명
wsdlDirectory	Java 소스 파일을 생성하기 위해서 사용되는 WSDL 파일들을 포함한 디렉터리 경로를 설정한다. (기본값: \${basedir}/src/wsdl)
wsdlFiles	Local WSDL 파일 리스트를 설정한다. (String 타입)
wsdlUrls	External WSDL URL 리스트를 설정한다. (String 타입)
destDir	생성된 Java 파일이 놓일 디렉터리 경로를 설정한다.  (기본값: \${project.build.outputDirectory})
sourceDestDir	생성된 소스 파일들이 놓일 위치를 설정한다. 이 속성을 지정하면 keep 속성이 자동으로 설정된다. (기본값: \${project.build.directory}/generated-sources/wsimport)
keep	생성되는 파일들을 보존 여부를 설정한다. (기본값: true)
verbose	Verbose 출력은 true일 때 가능하다. (기본값: false)
bindingDirectory	JAX-WS 또는 JAXB 바인딩 파일들을 포함한 디렉터리 경로를 설정한다.  (기본값: \${basedir}/src/jaxws)
bindingFiles	JAX-WS 또는 JAXB 바인딩 파일 리스트를 설정한다. (String 타입)
extension	벤더 특성의 확장성을 설정할 때 사용한다. 상호 운용 및 이식성이 떨어질 수 있다. (기본값: false)
wsdlLocation	WSDL URI를 이 속성에 명시하면 그 URI 값이 생성되는 service end-point interface와 서비스 인터페이스의 @WebService.wsdlLocation Annotation과 @WebServiceClient.wsdlLocation Annotation에 설정된다.
catalog	TR9401, XCatalog, OASIS XML Catalog 형식과 같은 외부의 Entity 참조값을 설정할 수 있다. 또한 Ant xmncatalog 타입을 사용할 수 있다.
packageName	WSDL의 모든 Namespace URI에 대한 Java 패키지 이름을 설정한다.  (String 타입)
target	이곳에 지정된 JAXWS 스펙 버전에 따라 코드를 생성한다. (String 타입)
quiet	출력 메시지들을 출력 여부를 설정한다. (기본값: false)

## 예제

다음은 pom.xml을 작성한 예제와 pom.xml을 통해 wsimport plugin을 실행하는 예제에 대한 설명이다.

### • pom.xml 작성 예제

wsimport plugin pom 파일 예제 : <pom.xml>

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>jeus.webservices.maven.sample</groupId>
```

```

<artifactId>wsimport_sample</artifactId>
<version>0.0.1</version>
<packaging>war</packaging>

<name>${project.artifactId}</name>

<build>
  <plugins>
    <plugin>
      <groupId>jeus.ws</groupId>
      <artifactId>jeus-ws-maven-plugin</artifactId>
      <version>0.0.1</version>
      <executions>
        <execution>
          <id>wsimport_test</id>
          <goals>
            <goal>wsimport</goal>
          </goals>
          <configuration>
            <packageName>jeus.webservices.sample.test</packageName>
            <wsdlDirectory>${basedir}/src/wsdl</wsdlDirectory>
            <wsdlFiles>
              <wsdlFile>EchoService.wsdl</wsdlFile>
            </wsdlFiles>
          </configuration>
        </execution>
      </executions>

      <dependencies>
        <dependency>
          <groupId>com.sun</groupId>
          <artifactId>tools</artifactId>
          <version>1.6</version>
          <scope>system</scope>
          <systemPath>${java.home}/../lib/tools.jar</systemPath>
        </dependency>
        <dependency>
          <groupId>com.sun.xml.ws</groupId>
          <artifactId>jaxws-tools</artifactId>
          <version>2.2</version>
          <scope>system</scope>
          <systemPath>${jeus.home}/lib/system/jaxws-tools.jar</systemPath>
        </dependency>
      </dependencies>
    </plugin>

    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-compiler-plugin</artifactId>
      <version>2.3.2</version>
      <configuration>
        <source>1.6</source>
        <target>1.6</target>
        <encoding>UTF-8</encoding>
        <compilerArguments>
          <endorseddirs>${jeus.home}/lib/endorsed</endorseddirs>
        </compilerArguments>
      </configuration>
    </plugin>
  </plugins>
</build>

```

```

    </plugins>
  </build>

</project>

```

## • 실행 예제

```

$ mvn -Djeus.home=${JEUS_HOME} generate-sources
[INFO] Scanning for projects...
[INFO]
[INFO] -----
[INFO] Building wsimport_sample 0.0.1
[INFO] -----
[INFO]
[INFO] --- jeus-ws-maven-plugin:0.0.1:wsimport (wsimport_test) @ wsimport_sample ---
[INFO] Processing: file:/jeus/wsimport/src/wsd1/EchoService.wsd1
[INFO] jaxws:wsimport args: [-keep, -s, /jeus/wsimport/target/generated-sources/wsimport,
-Xnocompile, -p, jeus.webservices.sample.test, file:/jues/wsimport/src/wsd1/EchoService.wsd1]
parsing WSDL...

Generating code...

[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 2.052s
[INFO] -----

```

## 5.6. jeusadmin 명령어 실행

Ant Build 파일(일반적으로 build.xml)을 편집하여 jeusadmin 명령어를 실행하는 방법을 소개한다. 예제를 통해 Ant Target을 작성하는 방법을 설명한다.

### 예제

다음은 Ant를 사용하여 server-info 명령어 수행하는 예제이다.

```

<project name="check-server-state-example" default="check-server-state">
  <property environment="env"/>
  <property name="jeus.home" value="${env.JEUS_HOME}"/>
  <property name="jeus.home.bin" value="${jeus.home}/bin"/>
  <property name="unix.jeusadmin" value="${jeus.home.bin}/jeusadmin"/>
  <property name="windows.jeusadmin" value="${jeus.home.bin}/jeusadmin.cmd"/>
  <property name="jeusadmin.args" value="-u administrator -p jeus -host host1 -port 9736
-verbose"/>
  <property name="server.name" value="adminServer"/>
  <property name="cmd.target" value="server-info -server ${server.name} -state"/>

  <condition property="isWindows">
    <os family="windows"/>

```

```

</condition>
<condition property="isUnix">
  <os family="unix"/>
</condition>

<target name="check-server-state">
  <antcall target="check-server-state-unix"/>
  <antcall target="check-server-state-windows"/>
</target>

<target name="check-server-state-windows" if="isWindows">
  <echo>${windows.jeusadmin} ${jeusadmin.args} ${cmd.target}</echo>
  <exec executable="${windows.jeusadmin}" osfamily="windows" spawn="false"
    failonerror="true">
    <arg line="${jeusadmin.args}"/>
    <arg value="${cmd.target}"/>
  </exec>
</target>

<target name="check-server-state-unix" if="isUnix">
  <echo>${unix.jeusadmin} ${jeusadmin.args} ${cmd.target}</echo>
  <exec executable="${unix.jeusadmin}" osfamily="unix" spawn="false"
    failonerror="true">
    <arg line="${jeusadmin.args}"/>
    <arg value="${cmd.target}"/>
  </exec>
</target>
</project>

```

- jeus.home은 JEUS가 설치되어 있는 위치를 가리킨다. 위 예제에서는 시스템 환경 변수 JEUS\_HOME에 해당 위치가 저장되어 있는 것으로 가정하였다.



JEUS에서 제공하는 jant 스크립트를 사용하여 ant를 실행할 경우 JEUS\_HOME을 별도로 지정하지 않아도 된다.

- unix.jeusadmin과 windows.jeusadmin은 jeusadmin 툴을 실행하기 위한 스크립트의 위치를 가리킨다. 일반적으로 jeusadmin 스크립트는 JEUS\_HOME/bin 아래에 위치한다. UNIX와 Windows 운영체제에서 스크립트 이름이 다르기 때문에 property를 2개 사용하였다.
- jeusadmin.args는 jeusadmin 툴을 실행할 때 사용할 인자값들이다. 사용자 이름, 비밀번호, 호스트 주소, 포트 번호, 상세 출력(verbose) 여부 등 필요한 옵션을 설정한다. 옵션에 대한 자세한 내용은 [jeusadmin](#)을 참고한다.
- cmd.target은 실행할 명령어를 지정한다. 예제에서는 server-info 명령어를 사용하여 서버 상태를 확인하고자 하였으므로, 'server-info -server \${server.name} -state'로 설정하였다. jeusadmin에서 사용할 수 있는 명령어에 대한 자세한 사항은 Part II. 콘솔 명령어와 툴을 참고한다.
- check-server-state-windows와 check-server-state-unix target들에서는 exec task를 사용하여 jeusadmin을 실행한다. osfamily 속성을 사용하여 운영체제의 종류에 맞는 스크립트가 실행될 수 있게 하였다. <exec> 태그에서 spawn 속성을 false로 주어야 Ant를 통해 수행 결과를 확인할 수 있다. 만약 명령어 수행이 실패하였을 때 Ant build가 실패하게 할 경우 failonerror 속성을 true로 설정한다.

예제를 UNIX 운영체제에서 실행한 결과는 다음과 같다.



```

JEUS_HOME/bin$jant check-server-state
Buildfile: JEUS_HOME/bin/build.xml

check-server-state:

check-server-state-unix:
    [echo] JEUS_HOME/bin/jeusadmin -u administrator -p jeus -host host1 -port 9736 -verbose server-
info -server adminServer -state
    [exec] Verbose output is enabled.
    [exec] Attempting to connect to localhost:9736.
    [exec] The connection has been established to Domain Administration Server adminServer in the
domain domain1.
    [exec] RUNNING
    [exec]

check-server-state-windows:

BUILD SUCCESSFUL
Total time: 1 second
JEUS_HOME/bin$

```

## 5.7. Maven Plugin

본 절에서는 JEUS Maven Plugin에 대해 설명한다. JEUS Maven Plugin을 사용하면 Apache Maven을 사용하여 JEUS 서버를 기동하거나, 애플리케이션을 배포하는 등과 같은 작업을 수행할 수 있다.



본 절에서는 Apache Maven에 대한 기본적인 내용은 서술하지 않으며, 독자가 이미 Apache Maven에 대한 충분한 지식을 갖추고 있고, Apache Maven이 이미 정상적으로 설치되어 있다고 가정하고 설명을 진행한다. Apache Maven에 대한 사항은 <http://maven.apache.org>을 참고하도록 한다.

### 5.7.1. 설치

JEUS Maven Plugin을 사용하기 위해서는 플러그인을 설치하는 작업을 먼저 진행해야 한다.

JEUS Maven Plugin은 다음 경로에서 찾을 수 있다.

```
JEUS_HOME/lib/systemapps/jeus-maven-plugin.jar
```

설치 과정은 다음과 같다.

1. jeus-maven-plugin.jar를 임의의 디렉터리에 복사한다(디렉터리 이름은 임의로 설정한다).

```

mkdir /home/user1/jeus-maven-plugin
cd /home/user1/jeus-maven-plugin
cp JEUS_HOME/lib/systemapps/jeus-maven-plugin.jar /home1/user/jeus-maven-plugin

```

2. 임의의 디렉터리로 이동한 후 jeus-maven-plugin.jar에 압축되어 있는 pom.xml을 본다.

```
jar xvf /home/user1/jeus-maven-plugin/jeus-maven-plugin META-INF/maven/com.tmax.jeus/jeus-maven-plugin/pom.xml
```

3. 다음 명령어를 이용하여 JEUS Maven Plugin을 사용자의 maven local repository에 설치한다.

```
mvn install:install-file -Dfile=/home/user1/jeus-maven-plugin/jeus-maven-plugin.jar -DpomFile=META-INF/maven/com.tmax.jeus/jeus-maven-plugin/pom.xml
```

## 5.7.2. Maven Plugin Goals

JEUS Maven Plugin에서 제공하는 goal들에 대해 설명한다.

### 5.7.2.1. start-master-xml

Master Server를 기동한다. 내부적으로는 JEUS\_HOME/bin 아래에 있는 스크립트를 실행한다.

- 사용법

```
com.tmax.jeus:jeus-maven-plugin:start-master-xml
```

- 매개변수

아래 매개변수 중 user와 password, filename은 필수 입력항목이 아니지만, user/password 또는 filename 중 하나는 반드시 설정해야 한다.

매개변수 이름	프로퍼티 이름	타입	설명
jeusHome	jeus.home	java.lang.String	JEUS가 설치되어 있는 경로를 설정한다. (필수 입력항목)
host	jeus.server.host	java.lang.String	Master Server의 host address를 설정한다. (필수 입력항목)
port	jeus.server.port	java.lang.Integer	Master Server의 포트 번호를 설정한다. (필수 입력항목)
user	jeus.user.name	java.lang.String	JEUS 사용자 이름을 설정한다.
password	jeus.user.password	java.lang.String	JEUS 사용자 비밀번호를 설정한다.

매개변수 이름	프로퍼티 이름	타입	설명
domain	jeus.domain.name	java.lang.String	Master Server가 속한 domain 이름을 설정한다.  (필수 입력항목)
server	jeus.server.name	java.lang.String	Master Server 이름을 설정한다. (필수 입력항목)
filename	jeus.auth.file	java.lang.String	JEUS 인증할 때 사용할 계정 정보 파일을 설정한다. startMasterServer 스크립트에서 제공하는 f 옵션과 동일하다.
waitingTime	jeus.boot.timeout	java.lang.Integer	JEUS를 기동시키고 난 후 플러그인에서는 일정 시간 동안 기다리면서 서버가 제대로 기동되었는지 여부를 검사하여 goal이 성공적으로 수행되었는지 여부를 판단한다. 이때 얼마나 기다릴지 설정하는 값이다. (단위: 초, 기본값: 10초)
cachelogin	jeus.server.start.cachedlogin	boolean	기동할 때 사용한 인증 정보를 저장할지 여부를 설정한다. (기본값: false)
force	jeus.server.start.forced	boolean	기동할 때 설정되어 있는 애플리케이션이 모두 RUNNING 상태가 되지 않더라도 부팅이 완료되게 한다. (기본값: false)
verbose	jeus.server.start.verbose	boolean	상세 출력 사용 여부를 설정한다.  (기본값: false)
standby	jeus.server.start.standby	boolean	Standby booting 사용 여부를 설정한다.  (기본값: false)

#### • 예제

매개변수는 pom.xml에 설정하거나 command line에서 실행할 때 지정할 수 있다.

다음은 pom.xml에 설정을 한 예제이다.

```
<!-- JEUS Maven Plugin - start-master.xml -->
<plugin>
  <groupId>com.tmax.jeus</groupId>
  <artifactId>jeus-maven-plugin</artifactId>
  <version>9.0.0.0</version>
  <configuration>
    <jeusHome>/home/user1/jeus</jeusHome>
    <host>host1</host>
    <port>9736</port>
    <user>administrator</user>
    <password>jeus</password>
    <domain>domain1</domain>
    <server>adminServer</server>
  </configuration>
```

```
</plugin>
```

goal을 실행하였을 경우 출력되는 결과는 다음과 같다.

```
$ mvn com.tmax.jeus:jeus-maven-plugin:start-master-xml
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.tmax.jeus:jeus-maven-plugin >-----
[INFO] Building JEUS Maven Plugin 9.0.0.0
[INFO]    from pom.xml
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- jeus-maven-plugin:9.0.0.0:start-master-xml (default-cli) @ jeus-maven-plugin ---
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time:  17.937 s
[INFO] Finished at: 2024-08-20T14:53:46+09:00
[INFO] -----
```

### 5.7.2.2. start-ms-xml

MS를 기동한다. 내부적으로는 JEUS\_HOME/bin 아래에 있는 스크립트를 실행한다.

- 사용법

```
com.tmax.jeus:jeus-maven-plugin:start-ms-xml
```

- 매개변수

아래 매개변수 중 user와 password, filename은 필수 입력항목이 아니지만, user/password 또는 filename 중 하나는 반드시 설정해야 한다.

매개변수 이름	프로퍼티 이름	타입	설명
jeusHome	jeus.home	java.lang.String	JEUS가 설치되어 있는 경로를 설정한다. (필수 입력항목)
host	jeus.server.host	java.lang.String	Master Server의 host address를 설정한다. (필수 입력항목)
port	jeus.server.port	java.lang.Integer	Master Server의 포트 번호를 설정한다. (필수 입력항목)
user	jeus.user.name	java.lang.String	JEUS 사용자 이름을 설정한다.
password	jeus.user.password	java.lang.String	JEUS 사용자 비밀번호를 설정한다.

매개변수 이름	프로퍼티 이름	타입	설명
domain	jeus.domain.name	java.lang.String	Master Server가 속한 domain 이름을 설정한다. (필수 입력항목)
server	jeus.server.name	java.lang.String	Master Server 이름을 설정한다. (필수 입력항목)
filename	jeus.auth.file	java.lang.String	JEUS를 인증할 때 사용할 계정 정보 파일을 설정한다. startManagedServer 스크립트에서 제공하는 f 옵션과 동일하다.
waitingTime	jeus.boot.timeout	java.lang.Integer	JEUS를 기동시키고 난 후 플러그인에서는 일정 시간 동안 기다리면서 서버가 제대로 기동되었는지 여부를 검사하여 goal이 성공적으로 수행되었는지 여부를 판단한다. 이때 얼마나 기다릴지 설정하는 값이다. (단위: 초, 기본값: 10초)
cachelogin	jeus.server.start.cach e login	boolean	기동할 때 사용한 인증 정보를 저장할지 여부를 설정한다. (기본값: false)
force	jeus.server.start.forc ed	boolean	기동할 때 설정되어 있는 애플리케이션이 모두 RUNNING 상태가 되지 않더라도 부팅이 완료되게 한다. (기본값: false)
verbose	jeus.server.start.verb ose	boolean	상세 출력 사용 여부를 설정한다. (기본값: false)
standby	jeus.server.start.stan dby	boolean	Standby booting 사용 여부를 설정한다. (기본값: false)
masterURL	jeus.server.start.mas terurl	java.lang.String	MS가 속한 domain을 관리하는 DAS의 url을 설정한다. (필수 입력항목)

## • 예제

매개변수는 pom.xml에 설정하거나 command line에서 실행할 때 지정할 수 있다.

다음은 pom.xml에 설정을 한 예제이다.

```
<!-- JEUS Maven Plugin - start-ms -->
<plugin>
  <groupId>com.tmax.jeus</groupId>
  <artifactId>jeus-maven-plugin</artifactId>
  <version>9.0.0.0</version>
  <configuration>
    <jeusHome>/home/user1/jeus</jeusHome>
    <host>host1</host>
    <port>9736</port>
    <user>administrator</user>
    <password>jeus</password>
    <domain>domain1</domain>
```

```
<server>server1</server>
<masterURL>host1:9736</masterURL>
</configuration>
</plugin>
```

goal을 실행하였을 경우 출력되는 결과는 다음과 같다.

```
$ mvn jeus.tool:jeus-maven-plugin:start-ms
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.tmax.jeus:jeus-maven-plugin >-----
[INFO] Building JEUS Maven Plugin 9.0.0.0
[INFO] -----[ maven-plugin ]-----
[INFO]
[INFO] --- jeus-maven-plugin:9.0.0.0:start-ms-xml (default-cli) @ jeus-maven-plugin ---
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 6.174 s
[INFO] Finished at: 2022-08-30T08:13:58+09:00
[INFO] -----
```

### 5.7.2.3. stop-jeus

접속한 서버를 종료시킨다. 내부적으로는 콘솔 툴(jeusadmin)에서 제공하는 one-step execution 기능을 사용하여 **local-shutdown** 명령어를 실행한다.

- 사용법

```
com.tmax.jeus:jeus-maven-plugin:stop-jeus
```

- 매개변수

매개변수 이름	프로퍼티 이름	타입	설명
jeusHome	jeus.home	java.lang.String	JEUS가 설치되어 있는 경로를 설정한다. (필수 입력항목)
host	jeus.server.host	java.lang.String	jeusadmin으로 접속할 서버의 주소이다. (필수 입력항목)
port	jeus.server.port	java.lang.Integer	jeusadmin으로 접속할 서버의 포트 번호이다. (기본값: 9736, 필수 입력항목)
user	jeus.user.name	java.lang.String	JEUS 사용자 이름을 설정한다. (필수 입력항목)

매개변수 이름	프로퍼티 이름	타입	설명
password	jeus.user.password	java.lang.String	JEUS 사용자 비밀번호를 설정한다. (필수 입력항목)
args	없음	java.util.List	명령어를 실행할 때 넘겨줄 인자값들을 설정한다. arg와 args 둘 다 설정하였을 경우 arg가 우선한다. pom.xml에 설정할 경우 사용하는 것을 권장한다.
arg	jeus.jeusadmin.com mand.arguments	java.lang.String	명령어를 실행할 때 넘겨줄 인자값들을 설정한다. arg와 args 둘 다 설정하였을 경우 arg가 우선한다.

#### • 예제

매개변수는 pom.xml에 설정하거나 command line에서 실행할 때 지정할 수 있다.

다음은 pom.xml에 설정을 한 예제이다.

```
<!-- JEUS Maven Plugin - stop-jeus -->
<plugin>
  <groupId>com.tmax.jeus</groupId>
  <artifactId>jeus-maven-plugin</artifactId>
  <version>9.0.0.0</version>
  <configuration>
    <jeusHome>/home/user1/jeus</jeusHome>
    <host>host1</host>
    <port>9736</port>
    <user>administrator</user>
    <password>jeus</password>
  </configuration>
</plugin>
```

goal을 실행하였을 경우 일반적으로 출력되는 결과는 다음과 같다.

```
$ mvn jeus.tool:jeus-maven-plugin:stop-jeus
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.tmax.jeus:jeus-maven-plugin >-----
[INFO] Building JEUS Maven Plugin 9.0.0.0
[INFO] -----[ maven-plugin ]-----
[INFO]
[INFO] --- jeus-maven-plugin:9.0.0.0:stop-jeus (default-cli) @ jeus-maven-plugin ---
[INFO] jeus.home = /home/user1/jeus
[INFO] jeusadmin.path = /home/user1/jeus/bin/jeusadmin
[INFO] jeusadmin.one-step-execution-command = [local-shutdown]
[INFO] Attempting to connect to host1:9736.
[INFO] The connection has been established to JEUS Master Server [adminServer] in the domain
[domain1].
[INFO] The server [adminServer] has been shut down successfully.
[INFO] -----
[INFO] BUILD SUCCESS
```

```
[INFO] -----
[INFO] Total time: 4.381 s
[INFO] Finished at: 2022-08-30T08:18:08+09:00
[INFO] -----
```

#### 5.7.2.4. deploy

애플리케이션을 deploy한다. 내부적으로는 콘솔 툴(jeusadmin)에서 제공하는 one-step execution 기능을 사용하여 **deploy-application** 명령어를 실행한다.

- 사용법

```
com.tmax.jeus:jeus-maven-plugin:deploy
```

- 매개변수

매개변수 이름	프로퍼티 이름	타입	설명
jeusHome	jeus.home	java.lang.String	JEUS가 설치되어 있는 경로를 설정한다. (필수 입력항목)
host	jeus.server.host	java.lang.String	jeusadmin으로 접속할 서버의 주소이다. (필수 입력항목)
port	jeus.server.port	java.lang.Integer	jeusadmin으로 접속할 서버의 포트 번호이다. (기본값: 9736, 필수 입력항목)
user	jeus.user.name	java.lang.String	JEUS 사용자 이름을 설정한다. (필수 입력항목)
password	jeus.user.password	java.lang.String	JEUS 사용자 비밀번호를 설정한다. (필수 입력항목)
args	없음	java.util.List	명령어를 실행할 때 넘겨줄 인자값들을 설정한다. arg와 args 둘 다 설정하였을 경우 arg가 우선한다. pom.xml에 설정할 경우 사용하는 것을 권장한다.
arg	jeus.jeusadmin.com mand.arguments	java.lang.String	명령어를 실행할 때 넘겨줄 인자값들을 설정한다. arg와 args 둘 다 설정하였을 경우 arg가 우선한다.

- 예제

매개변수는 pom.xml에 설정하거나 command line에서 실행할 때 지정할 수 있다.

다음은 pom.xml에 설정을 한 예제이다.



```
<!-- JEUS Maven Plugin - deploy -->
<plugin>
  <groupId>com.tmax.jeus</groupId>
  <artifactId>jeus-maven-plugin</artifactId>
  <version>9.0.0.0</version>
  <configuration>
    <jeusHome>/home/user1/jeus</jeusHome>
    <host>host1</host>
    <port>9736</port>
    <user>administrator</user>
    <password>jeus</password>
    <args>
      <param>myWeb</param>
      <param>-servers adminServer</param>
    </args>
  </configuration>
</plugin>
```

goal을 실행하였을 경우 일반적으로 출력되는 결과는 다음과 같다.

```
$ mvn jeus.tool:jeus-maven-plugin:deploy
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.tmax.jeus:jeus-maven-plugin >-----
[INFO] Building JEUS Maven Plugin 9.0.0.0
[INFO]   from pom.xml
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- jeus-maven-plugin:9.0.0.0:deploy (default-cli) @ jeus-maven-plugin ---
[INFO] jeus.home = /home/user1/jeus
[INFO] jeusadmin.path = /home/user1/jeus/bin/jeusadmin
[INFO] jeusadmin.one-step-execution-command = [deploy-application myWeb -servers adminServer ]
[INFO] Attempting to connect to host1:9736.
[INFO] The connection has been established to JEUS Master Server [adminServer] in the domain
[domain1].
[INFO] deploy the application for the application [myWeb] succeeded.
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time:  1.356 s
[INFO] Finished at: 2024-08-20T15:09:26+09:00
[INFO] -----
```

### 5.7.2.5. undeploy

deploy했던 애플리케이션을 undeploy한다. 내부적으로는 콘솔 툴(jeusadmin)에서 제공하는 one-step execution 기능을 사용하여 **undeploy-application** 명령어를 실행한다.

- 사용법

```
com.tmax.jeus:jeus-maven-plugin:undeploy
```

- 매개변수

매개변수 이름	프로퍼티 이름	타입	설명
jeusHome	jeus.home	java.lang.String	JEUS가 설치되어 있는 경로를 설정한다. (필수 입력항목)
host	jeus.server.host	java.lang.String	jeusadmin으로 접속할 서버의 주소이다. (필수 입력항목)
port	jeus.server.port	java.lang.Integer	jeusadmin으로 접속할 서버의 포트 번호이다. (기본값: 9736, 필수 입력항목)
user	jeus.user.name	java.lang.String	JEUS 사용자 이름을 설정한다. (필수 입력항목)
password	jeus.user.password	java.lang.String	JEUS 사용자 비밀번호를 설정한다. (필수 입력항목)
args	없음	java.util.List	명령어를 실행할 때 넘겨줄 인자값들을 설정한다. arg와 args 둘 다 설정하였을 경우 arg가 우선한다. pom.xml에 설정할 경우 사용하는 것을 권장한다.
arg	jeus.jeusadmin.com mand.arguments	java.lang.String	명령어를 실행할 때 넘겨줄 인자값들을 설정한다. arg와 args 둘 다 설정하였을 경우 arg가 우선한다.

- 예제

매개변수는 pom.xml에 설정하거나 command line에서 실행할 때 지정할 수 있다.

다음은 pom.xml에 설정을 한 예제이다.

```

<!-- JEUS Maven Plugin - undeploy -->
<plugin>
  <groupId>com.tmax.jeus</groupId>
  <artifactId>jeus-maven-plugin</artifactId>
  <version>9.0.0.0</version>
  <configuration>
    <jeusHome>/home/user1/jeus</jeusHome>
    <host>host1</host>
    <port>9736</port>
    <user>administrator</user>
    <password>jeus</password>
    <args>
      <param>myWeb</param>
    </args>
  </configuration>
</plugin>

```

goal을 실행하였을 경우 일반적으로 출력되는 결과는 다음과 같다.

```
$ mvn com.tmax.jeus:jeus-maven-plugin:undeploy
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.tmax.jeus:jeus-maven-plugin >-----
[INFO] Building JEUS Maven Plugin 9.0.0.0
[INFO]   from pom.xml
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- jeus-maven-plugin:9.0.0.0:undeploy (default-cli) @ jeus-maven-plugin ---
[INFO] jeus.home = /home/user1/jeus
[INFO] jeusadmin.path = /home/user1/jeus/bin/jeusadmin
[INFO] jeusadmin.one-step-execution-command = [undeploy-application myWeb ]
[INFO] Attempting to connect to host1:9736.
[INFO] The connection has been established to JEUS Master Server [adminServer] in the domain
[domain1].
[INFO] Undeploying [myWeb] (This may take time due to graceful undeployment) .....
[INFO] undeploy the application for the application [myWeb] succeeded.
[INFO] successfully undeployed (elapsed = 160ms)
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time:  1.256 s
[INFO] Finished at: 2024-08-20T15:12:41+09:00
[INFO] -----
```

### 5.7.2.6. start-app

deploy한 애플리케이션의 서비스를 시작한다. 내부적으로는 콘솔 툴(jeusadmin)에서 제공하는 one-step execution 기능을 사용하여 **start-application** 명령어를 실행한다.

- 사용법

```
com.tmax.jeus:jeus-maven-plugin:start-app
```

- 매개변수

매개변수 이름	프로퍼티 이름	타입	설명
jeusHome	jeus.home	java.lang.String	JEUS가 설치되어 있는 경로를 설정한다. (필수 입력항목)
host	jeus.server.host	java.lang.String	jeusadmin으로 접속할 서버의 주소이다. (필수 입력항목)
port	jeus.server.port	java.lang.Integer	jeusadmin으로 접속할 서버의 포트 번호이다. (기본값: 9736, 필수 입력항목)
user	jeus.user.name	java.lang.String	JEUS 사용자 이름을 설정한다. (필수 입력항목)

매개변수 이름	프로퍼티 이름	타입	설명
password	jeus.user.password	java.lang.String	JEUS 사용자 비밀번호를 설정한다. (필수 입력항목)
args	없음	java.util.List	명령어를 실행할 때 넘겨줄 인자값들을 설정한다. arg와 args 둘 다 설정하였을 경우 arg가 우선한다. pom.xml에 설정할 경우 사용하는 것을 권장한다.
arg	jeus.jeusadmin.com mand.arguments	java.lang.String	명령어를 실행할 때 넘겨줄 인자값들을 설정한다. arg와 args 둘 다 설정하였을 경우 arg가 우선한다.

#### • 예제

매개변수는 pom.xml에 설정하거나 command line에서 실행할 때 지정할 수 있다.

다음은 pom.xml에 설정을 한 예제이다.

```
<!-- JEUS Maven Plugin - start-app -->
<plugin>
  <groupId>com.tmax.jeus</groupId>
  <artifactId>jeus-maven-plugin</artifactId>
  <version>9.0.0.0</version>
  <configuration>
    <jeusHome>/home/user1/jeus</jeusHome>
    <host>host1</host>
    <port>9736</port>
    <user>administrator</user>
    <password>jeus</password>
    <args>
      <param>myWeb</param>
    </args>
  </configuration>
</plugin>
```

goal을 실행하였을 경우 일반적으로 출력되는 결과는 다음과 같다.

```
$ mvn com.tmax.jeus:jeus-maven-plugin:start-app
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.tmax.jeus:jeus-maven-plugin >-----
[INFO] Building JEUS Maven Plugin 9.0.0.0
[INFO]   from pom.xml
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- jeus-maven-plugin:9.0.0.0:start-app (default-cli) @ jeus-maven-plugin ---
[INFO] jeus.home = /home/user1/jeus
[INFO] jeusadmin.path = /home/user1/jeus/bin/jeusadmin
[INFO] jeusadmin.one-step-execution-command = [start-application myWeb ]
[INFO] Attempting to connect to host1:9736.
[INFO] The connection has been established to JEUS Master Server [adminServer] in the domain
```

```
[domain1].
[INFO] start the application for the application [myWeb] succeeded.
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 1.089 s
[INFO] Finished at: 2024-08-20T15:11:27+09:00
[INFO] -----
```

### 5.7.2.7. stop-app

서비스 중인 애플리케이션을 정지시킨다. 내부적으로는 콘솔 툴(jeusadmin)에서 제공하는 one-step execution 기능을 사용하여 **stop-application** 명령어를 실행한다.

- 사용법

```
com.tmax.jeus:jeus-maven-plugin:stop-app
```

- 매개변수

매개변수 이름	프로퍼티 이름	타입	설명
jeusHome	jeus.home	java.lang.String	JEUS가 설치되어 있는 경로를 설정한다. (필수 입력항목)
host	jeus.server.host	java.lang.String	jeusadmin으로 접속할 서버의 주소이다. (필수 입력항목)
port	jeus.server.port	java.lang.Integer	jeusadmin으로 접속할 서버의 포트 번호이다. (기본값: 9736, 필수 입력항목)
user	jeus.user.name	java.lang.String	JEUS 사용자 이름을 설정한다. (필수 입력항목)
password	jeus.user.password	java.lang.String	JEUS 사용자 비밀번호를 설정한다. (필수 입력항목)
args	없음	java.util.List	명령어를 실행할 때 넘겨줄 인자값들을 설정한다. arg와 args 둘 다 설정하였을 경우 arg가 우선한다. pom.xml에 설정할 경우 사용하는 것을 권장한다.
arg	jeus.jeusadmin.com mand.arguments	java.lang.String	명령어를 실행할 때 넘겨줄 인자값들을 설정한다. arg와 args 둘 다 설정하였을 경우 arg가 우선한다.

- 예제

매개변수는 pom.xml에 설정하거나 command line에서 실행할 때 지정할 수 있다.

다음은 pom.xml에 설정을 한 예제이다.

```
<!-- JEUS Maven Plugin - stop-app -->
<plugin>
  <groupId>com.tmax.jeus</groupId>
  <artifactId>jeus-maven-plugin</artifactId>
  <version>9.0.0.0</version>
  <configuration>
    <jeusHome>/home/user1/jeus</jeusHome>
    <host>host1</host>
    <port>9736</port>
    <user>administrator</user>
    <password>jeus</password>
    <args>
      <param>myWeb</param>
    </args>
  </configuration>
</plugin>
```

goal을 실행하였을 경우 일반적으로 출력되는 결과는 다음과 같다.

```
$ mvn com.tmax.jeus:jeus-maven-plugin:stop-app
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.tmax.jeus:jeus-maven-plugin >-----
[INFO] Building JEUS Maven Plugin 9.0.0.0
[INFO]   from pom.xml
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- jeus-maven-plugin:9.0.0.0:stop-app (default-cli) @ jeus-maven-plugin ---
[INFO] jeus.home = /home/user1/jeus
[INFO] jeusadmin.path = /home/user1/jeus/bin/jeusadmin
[INFO] jeusadmin.one-step-execution-command = [stop-application myWeb ]
[INFO] Attempting to connect to host1:9736.
[INFO] The connection has been established to JEUS Master Server [adminServer] in the domain
[domain1].
[INFO] stop the application for the application [myWeb] succeeded.
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time:  1.059 s
[INFO] Finished at: 2024-08-20T15:10:32+09:00
[INFO] -----
```

### 5.7.2.8. install-app

애플리케이션을 DAS에 설치한다. 내부적으로는 콘솔 툴(jeusadmin)에서 제공하는 one-step execution 기능을 사용하여 **install-application** 명령어를 실행한다.

- 사용법

```
com.tmax.jeus:jeus-maven-plugin:install-app
```

- 매개변수

매개변수 이름	프로퍼티 이름	타입	설명
jeusHome	jeus.home	java.lang.String	JEUS가 설치되어 있는 경로를 설정한다. (필수 입력항목)
host	jeus.server.host	java.lang.String	jeusadmin으로 접속할 서버의 주소이다. (필수 입력항목)
port	jeus.server.port	java.lang.Integer	jeusadmin으로 접속할 서버의 포트 번호이다. (기본값: 9736, 필수 입력항목)
user	jeus.user.name	java.lang.String	JEUS 사용자 이름을 설정한다. (필수 입력항목)
password	jeus.user.password	java.lang.String	JEUS 사용자 비밀번호를 설정한다. (필수 입력항목)
args	없음	java.util.List	명령어를 실행할 때 넘겨줄 인자값들을 설정한다. arg와 args 둘 다 설정하였을 경우 arg가 우선한다. pom.xml에 설정할 경우 사용하는 것을 권장한다.
arg	jeus.jeusadmin.com mand.arguments	java.lang.String	명령어를 실행할 때 넘겨줄 인자값들을 설정한다. arg와 args 둘 다 설정하였을 경우 arg가 우선한다.

- 예제

매개변수는 pom.xml에 설정하거나 command line에서 실행할 때 지정할 수 있다.

다음은 pom.xml에 설정을 한 예제이다.

```
<!-- JEUS Maven Plugin - install-app -->
<plugin>
  <groupId>com.tmax.jeus</groupId>
  <artifactId>jeus-maven-plugin</artifactId>
  <version>9.0.0.0</version>
  <configuration>
    <jeusHome>/home/user1/jeus</jeusHome>
    <host>host1</host>
    <port>9736</port>
    <user>administrator</user>
    <password>jeus</password>
    <args>
      <param>/home/user/myWeb.war</param>
      <param>-id myWeb</param>
    </args>
  </configuration>
</plugin>
```

goal을 실행하였을 경우 일반적으로 출력되는 결과는 다음과 같다.

```
$ mvn com.tmax.jeus:jeus-maven-plugin:install-app
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.tmax.jeus:jeus-maven-plugin >-----
[INFO] Building JEUS Maven Plugin 9.0.0.0
[INFO]   from pom.xml
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- jeus-maven-plugin:9.0.0.0:install-app (default-cli) @ jeus-maven-plugin ---
[INFO] jeus.home = /home/user1/jeus
[INFO] jeusadmin.path = /home/user1/jeus/bin/jeusadmin
[INFO] jeusadmin.one-step-execution-command = [install-application /home/user1/myWeb.war -id
myWeb ]
[INFO] Attempting to connect to host1:9736.
[INFO] The connection has been established to JEUS Master Server [adminServer] in the domain
[domain1].
[INFO] Successfully installed the application [myWeb].
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time:  1.444 s
[INFO] Finished at: 2024-08-20T15:01:07+09:00
[INFO] -----
```

### 5.7.2.9. uninstall-app

DAS에 설치되어 있는 애플리케이션을 제거한다. 내부적으로는 콘솔 툴(jeusadmin)에서 제공하는 one-step execution 기능을 사용하여 **uninstall-application** 명령어를 실행한다.

- 사용법

```
com.tmax.jeus:jeus-maven-plugin:uninstall-app
```

- 매개변수

매개변수 이름	프로퍼티 이름	타입	설명
jeusHome	jeus.home	java.lang.String	JEUS가 설치되어 있는 경로를 설정한다. (필수 입력항목)
host	jeus.server.host	java.lang.String	jeusadmin으로 접속할 서버의 주소이다. (필수 입력항목)
port	jeus.server.port	java.lang.Integer	jeusadmin으로 접속할 서버의 포트 번호이다. (기본값: 9736, 필수 입력항목)
user	jeus.user.name	java.lang.String	JEUS 사용자 이름을 설정한다. (필수 입력항목)



매개변수 이름	프로퍼티 이름	타입	설명
password	jeus.user.password	java.lang.String	JEUS 사용자 비밀번호를 설정한다. (필수 입력항목)
args	없음	java.util.List	명령어를 실행할 때 넘겨줄 인자값들을 설정한다. arg와 args 둘 다 설정하였을 경우 arg가 우선한다. pom.xml에 설정할 경우 사용하는 것을 권장한다.
arg	jeus.jeusadmin.com mand.arguments	java.lang.String	명령어를 실행할 때 넘겨줄 인자값들을 설정한다. arg와 args 둘 다 설정하였을 경우 arg가 우선한다.

- 예제

매개변수는 pom.xml에 설정하거나 command line에서 실행할 때 지정할 수 있다.

다음은 pom.xml에 설정을 한 예제이다.

```
<!-- JEUS Maven Plugin - uninstall-app -->
<plugin>
  <groupId>com.tmax.jeus</groupId>
  <artifactId>jeus-maven-plugin</artifactId>
  <version>8.0.0.0</version>
  <configuration>
    <jeusHome>/home/user1/jeus</jeusHome>
    <host>host1</host>
    <port>9736</port>
    <user>administrator</user>
    <password>jeus</password>
    <args>
      <param>myWeb</param>
    </args>
  </configuration>
</plugin>
```

goal을 실행하였을 경우 일반적으로 출력되는 결과는 다음과 같다.

```
$ mvn com.tmax.jeus:jeus-maven-plugin:uninstall-app
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.tmax.jeus:jeus-maven-plugin >-----
[INFO] Building JEUS Maven Plugin 9.0.0.0
[INFO]   from pom.xml
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- jeus-maven-plugin:9.0.0.0:uninstall-app (default-cli) @ jeus-maven-plugin ---
[INFO] jeus.home = /home/user1/jeus
[INFO] jeusadmin.path = /home/user1/jeus/bin/jeusadmin
[INFO] jeusadmin.one-step-execution-command = [uninstall-application myWeb ]
[INFO] Attempting to connect to host1:9736.
[INFO] The connection has been established to JEUS Master Server [adminServer] in the domain
```

```
[domain1].
[INFO] uninstall the application for the application [myWeb] succeeded. : Successfully deleted
[myWeb].
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 1.397 s
[INFO] Finished at: 2024-08-20T15:07:09+09:00
[INFO] -----
```

### 5.7.2.10. jeusadmin

지정한 jeusadmin 명령어를 실행한다. 내부적으로는 콘솔 툴(jeusadmin)에서 제공하는 one-step execution 기능을 사용한다.

- 사용법

```
com.tmax.jeus:jeus-maven-plugin:jeusadmin
```

- 매개변수

매개변수 이름	프로퍼티 이름	타입	설명
jeusHome	jeus.home	java.lang.String	JEUS가 설치되어 있는 경로를 설정한다. (필수 입력항목)
host	jeus.server.host	java.lang.String	jeusadmin으로 접속할 서버의 주소이다. (필수 입력항목)
port	jeus.server.port	java.lang.Integer	jeusadmin으로 접속할 서버의 포트 번호이다. (기본값: 9736, 필수 입력항목)
user	jeus.user.name	java.lang.String	JEUS 사용자 이름을 설정한다. (필수 입력항목)
password	jeus.user.password	java.lang.String	JEUS 사용자 비밀번호를 설정한다. (필수 입력항목)
args	없음	java.util.List	명령어를 실행할 때 넘겨줄 인자값들을 설정한다. arg와 args 둘 다 설정하였을 경우 arg가 우선한다. pom.xml에 설정할 경우 사용하는 것을 권장한다.
arg	jeus.jeusadmin.com mand.arguments	java.lang.String	명령어를 실행할 때 넘겨줄 인자값들을 설정한다. arg와 args 둘 다 설정하였을 경우 arg가 우선한다.

매개변수 이름	프로퍼티 이름	타입	설명
command	jeus.jeusadmin.com mand	java.lang.String	실행할 jeusadmin 명령어 이름을 설정한다.  (필수 입력항목)

- 예제

매개변수는 pom.xml에 설정하거나 command line에서 실행할 때 지정할 수 있다.

다음은 pom.xml에 설정을 한 예제이며, 수행할 명령어는 **server-info -server adminServer -state**이다.

```
<!-- JEUS Maven Plugin - jeusadmin -->
<plugin>
  <groupId>com.tmax.jeus</groupId>
  <artifactId>jeus-maven-plugin</artifactId>
  <version>8.0.0.0</version>
  <configuration>
    <jeusHome>/home/user1/jeus</jeusHome>
    <host>host1</host>
    <port>9736</port>
    <user>administrator</user>
    <password>jeus</password>
    <command>server-info</command>
    <args>
      <param>-server adminServer</param>
      <param>-state</param>
    </args>
  </configuration>
</plugin>
```

goal을 실행하였을 경우 출력되는 결과는 다음과 같다.

```
mvn com.tmax.jeus:jeus-maven-plugin:jeusadmin
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.tmax.jeus:jeus-maven-plugin >-----
[INFO] Building JEUS Maven Plugin 9.0.0.0
[INFO]   from pom.xml
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- jeus-maven-plugin:9.0.0.0:jeusadmin (default-cli) @ jeus-maven-plugin ---
[INFO] jeus.home = /home/user1/jeus
[INFO] jeusadmin.path = /home/user1/jeus/bin/jeusadmin
[INFO] jeusadmin.one-step-execution-command = [server-info -server adminServer -state ]
[INFO] Attempting to connect to host1:9736.
[INFO] The connection has been established to JEUS Master Server [adminServer] in the domain
[domain1].
[INFO] RUNNING
[INFO]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time:  1.093 s
```

[INFO] Finished at: 2024-08-20T15:15:08+09:00

[INFO] -----