

JEUS Web Container **안내서**



Copyright © 2005 Tmax Soft Co., Ltd. All Rights Reserved.

Copyright Notice

Copyright©2005 Tmax Soft Co., Ltd. All Rights Reserved.

Tmax Soft Co., Ltd.

대한민국 서울시 강남구 대치동 946-1 글라스타워 18 층 우)135-708

Restricted Rights Legend

This software and documents are made available only under the terms of the Tmax Soft License Agreement and may be used or copied only in accordance with the terms of this agreement. No part of this document may be reproduced, transmitted, or translated in any form or by any means, electronic, mechanical, manual, or optical, without the prior written permission of Tmax Soft Co., Ltd.

소프트웨어 및 문서는 오직 TmaxSoft Co., Ltd.와의 사용권 계약 하에서만 이용이 가능하며, 사용권 계약에 따라서 사용하거나 복사 할 수 있습니다. 또한 이 매뉴얼에서 언급하지 않은 정보에 대해서는 보증 및 책임을 지지 않습니다.

이 매뉴얼에 대한 권리는 저작권에 보호되므로 발행자의 허가 없이 전체 또는 일부를 어떤 형식이나, 사진 녹화, 기록, 정보 저장 및 검색 시스템과 같은 그래픽이나 전자적, 기계적 수단으로 복제하거나 사용할 수 없습니다.

Trademarks

Tmax, WebtoB, WebT, and JEUS are registered trademarks of Tmax Soft Co., Ltd.

All other product names may be trademarks of the respective companies with which they are associated.

Tmax, WebtoB, WebT, JEUS 는 TmaxSoft Co., Ltd.의 등록 상표입니다.

기타 모든 제품들과 회사 이름은 각각 해당 소유주의 상표로서 참조용으로만 사용됩니다.

Document info

Document name: JEUS Web Container 안내서

Document date: 2005-6-6

Manual release version: 3

Software Version: JEUS5

차 례

1	소개.....	31
2	따라하기	33
3	JEUS Web에 대한 개요.....	37
3.1	소개.....	37
3.2	JEUS Web Container	37
3.3	JEUS Web Container와 JEUS 시스템	38
3.4	JEUS Web Container의 기본 컴포넌트들	40
3.5	Web 모듈 툴	41
3.6	Web 디렉토리 구조	42
3.7	JEUS Web 관련된 시스템 변수들	43
3.8	JEUS Web과 관련된 XML 설정 파일들	44
3.9	본 매뉴얼의 구성에 대한 상세 정보.....	45
3.10	결론.....	46
4	Web Container	47
4.1	소개.....	47
4.2	JEUS Web Container의 개요	47
4.2.1	소개	47
4.2.2	Web Container의 구조	47
4.2.3	Web Container의 주요 하위 컴포넌트들과 기능들	48
4.2.4	자동 모니터링	49
4.2.5	stdout 와 stderr redirection	49
4.2.6	Context Group.....	49
4.2.7	Database Connection Pool	50

4.2.8	Session 클러스터링	50
4.2.9	Logging	50
4.2.10	보안 스위치	50
4.2.11	Shutdown Timeout	51
4.2.12	요청 처리와 Worker Thread Pool	51
4.2.13	Web Container의 디렉토리 구조	51
4.2.14	결론	52
4.3	JEUS Web Container 설정	53
4.3.1	소개	53
4.3.2	JEUSMain.xml에 Web Container 추가	53
4.3.3	WEBMain.xml 설정	54
4.3.4	자동 모니터링 설정	55
4.3.5	stdout 과 stderr re-direction 설정	56
4.3.6	Context Group 설정	57
4.3.7	Database Connection Pool 설정	57
4.3.8	Session 클러스터링 설정	58
4.3.9	Logging 설정	59
4.3.10	보안 스위치 설정	60
4.3.11	Startup 과 Shutdown 클래스 설정	60
4.3.12	Shutdown Timeout 설정	60
4.3.13	웹 관리자를 이용한 Web Container 설정	61
4.3.14	결론	61
4.4	JEUS Web Container Logging 설정	61
4.4.1	소개	61
4.4.2	기본 logger file 위치	63
4.4.3	공통 설정 항목	64
4.4.4	access-log 관련 설정 항목	66

4.4.5	access-log <format> 설정	67
4.4.6	user-log 관련 설정 항목	68
4.5	JEUS Web Container 제어	69
4.5.1	소개	69
4.5.2	jeusadmin 콘솔 툴을 이용한 Web Container 제어	69
4.5.3	웹 관리자를 이용한 Web Container 제어 툴	70
4.5.4	결론	70
4.6	JEUS Web Container 모니터링	70
4.6.1	소개	70
4.6.2	콘솔을 통한 Web Container 모니터링	70
4.6.3	웹 관리자를 통한 Web Container 모니터링	71
4.6.4	결론	71
4.7	JEUS Web 성능 향상을 위한 튜닝	71
4.8	결론	71
5	Database Connection Pool.....	73
5.1	소개	73
5.2	Database Connection Pool의 개요	74
5.2.1	소개	74
5.2.2	JEUS Server의 DB connection pool	74
5.2.3	Web Container의 DB Connection Pool	74
5.2.4	Shared Connection Pool 모드	75
5.2.5	Non-shared Connection Pool 모드	75
5.2.6	DBA Override 기능	76
5.2.7	결론	76
5.3	Web Container Database Connection Pool 설정	76
5.3.1	소개	76
5.3.2	기본 DB Pool 설정	76

5.3.3	Non-shared 모드 설정	79
5.3.4	Shared 모드 설정	80
5.3.5	DBA Connection 설정	82
5.3.6	웹 관리자를 이용한 설정	83
5.3.7	결론	83
5.4	Web Container Database Connection Pool 사용	84
5.5	결론	84
6	Context Group	87
6.1	소개	87
6.2	Context Group의 개요	87
6.2.1	소개	87
6.2.2	Context Group 아키텍처	88
6.2.3	Context Group의 주 설정	89
6.2.4	Context Group의 주요 하위 컴포넌트들	89
6.2.5	Virtual Host 설정	90
6.2.6	컨텍스트	90
6.2.7	웹 서버 연결	90
6.2.8	세션 관리 설정	91
6.2.9	인코딩	91
6.2.10	JSP engine	93
6.2.11	로깅	93
6.2.12	클러스터 환경에서 세션 처리	95
6.2.13	Response Header	95
6.2.14	Context Group 디렉토리 구조	95
6.2.15	결론	97
6.3	Context Group 설정	97
6.3.1	소개	97

6.3.2	기본 설정	100
6.3.3	Virtual Host 설정	100
6.3.4	컨텍스트 설정	100
6.3.5	웹 서버 연결 설정	101
6.3.6	세션 관리 설정	101
6.3.7	인코딩 설정	102
6.3.8	JSP engine 설정	102
6.3.9	로깅 설정	104
6.3.10	클러스터된 환경에서의 세션 설정	104
6.3.11	Response Header 설정	104
6.3.12	웹 관리자를 통한 Context Group 설정	106
6.3.13	결론	106
6.4	Context Group 제어	106
6.4.1	소개	106
6.4.2	5 개의 명령어	106
6.4.3	webadmin 콘솔 툴을 사용한 Context Group 제어	107
6.4.4	웹 관리자를 사용한 Context Group 제어	107
6.4.5	결과	108
6.5	Context Group 모니터링	108
6.5.1	소개	108
6.5.2	webadmin을 사용한 Context Group 모니터링	108
6.5.3	웹 관리자를 사용한 Context Group 모니터링	108
6.5.4	결론	108
6.6	Context Group 튜닝	108
6.7	결론	109
7	웹서버 연결과 클러스터링	111
7.1	소개	111

7.2	웹서버 연결의 개요.....	111
7.2.1	소개	111
7.2.2	리스너란 무엇인가?	112
7.2.3	Worker thread와 Worker thread pool	113
7.2.4	쓰레드 풀의 Active-Management와 상태 통보	114
7.2.5	여러 개의 Web Container와 Web Server 클러스터링	114
7.2.6	여덟 가지의 리스너들	116
7.2.7	HTTP 리스너.....	116
7.2.8	보안 리스너	117
7.2.9	TCP 리스너	117
7.2.10	UDP 리스너	117
7.2.11	Apache 리스너	117
7.2.12	WebtoB 리스너	117
7.2.13	AJP13 Listener	118
7.2.14	Tmax Listener.....	118
7.2.15	결론	118
7.3	리스너 설정.....	119
7.3.1	소개	119
7.3.2	Apache, HTTP, TCP, UDP, 보안 리스너 설정	120
7.3.3	자동 쓰레드 풀 관리 설정 (쓰레드 상태 통보)	124
7.3.4	WebtoB 리스너 설정	126
7.3.5	Tmax 리스너 설정	130
7.3.6	웹 관리자를 통한 리스너 설정	133
7.3.7	결론	133
7.4	리스너 연동과 클러스터링을 위한 웹서버 설정.....	133
7.4.1	소개	133
7.4.2	Apache 웹서버 설정과 클러스터링	134

7.4.3	WebtoB 웹서버 설정과 클러스터링	138
7.4.4	결과	144
7.5	TCP 리스너 사용	145
7.5.1	소개	145
7.5.2	용어	146
7.5.3	맞춤 통신 프로토콜 정의	146
7.5.4	Dispatcher 설정 클래스 구현	147
7.5.5	TCP 핸들러 구현	150
7.5.6	맞춤 프로토콜 코드를 위한 TCP 리스너 설정	154
7.5.7	TCP 클라이언트 구현	157
7.5.8	TCP 클라이언트 컴파일과 실행	160
7.5.9	결론	161
7.6	보안(SSL) 리스너 사용	161
7.6.1	소개	161
7.6.2	가정	162
7.6.3	보안 리스너 설정	165
7.6.4	SSL Keystore 설정	166
7.6.5	SSL Truststore 설정	168
7.6.6	보안 리스너 속성 설정	169
7.6.7	보안 리스너 시작하기	171
7.6.8	보안 리스너에 연결하기	171
7.6.9	결론	173
7.7	Web 리스너 튜닝	173
7.8	결론	174
8	Session Tracking	175
8.1	소개	175
8.2	Session Tracking의 개요	175

8.2.1	소개	175
8.2.2	Session Tracking의 기초	176
8.2.3	Web Container에서 지원하는 기본 Session Tracking..	177
8.2.4	클러스터 환경에서 Session Tracking.....	179
8.2.5	세션 라우팅	180
8.2.6	세션 서버	184
8.2.7	Multi Session 모드	187
8.2.8	혼합 모드	187
8.2.9	분산식 세션 서버	188
8.2.10	URL Rewriting 과 쿠키에 관하여	189
8.2.11	Shared 세션 데이터	189
8.2.12	결론	189
8.3	Session Tracking 설정	190
8.3.1	소개	190
8.3.2	세션 라우팅 설정	191
8.3.3	세션 서버 설정	191
8.3.4	Multi Session Mode설정	193
8.3.5	혼합 모드 설정	194
8.3.6	분산식 세션 서버 설정	195
8.3.7	웹 관리자를 통한 Session Tracking 설정.....	196
8.3.8	결론	197
8.4	Apache Web Server를 위한 세션 라우팅 설정	197
8.5	Session Tracking 튜닝	198
8.6	결론.....	199
9	Context(웹 어플리케이션)	201
9.1	소개.....	201
9.2	Web Context의 개요	201

9.2.1	소개	201
9.2.2	Context(웹 어플리케이션) 란 무엇인가?	202
9.2.3	WAR 파일과 그 구조	203
9.2.4	JEUS Web Container의 WAR 파일 등록	205
9.2.5	등록된 Web Application의 디렉토리 구조	206
9.2.6	Context Group, 가상호스트, Context	208
9.2.7	기본Context	208
9.2.8	결론	208
9.3	Web context 등록	208
9.3.1	소개	208
9.3.2	Context 디렉토리 생성과 WAR 언팩킹(Unpacking) ..	209
9.3.3	Deployment Descriptor 파일 설정	209
9.3.4	Web DD의 기본 context 설정	210
9.3.5	보안 롤 매핑	213
9.3.6	심볼릭 레퍼런스 매핑	215
9.3.7	JEUSMain.xml에 새로운 context 등록하기	217
9.3.8	배치 컴파일러를 사용한 JSP 프리컴파일	218
9.3.9	등록 확인	218
9.3.10	웹 관리자를 통한 context 등록	219
9.3.11	결론	219
9.4	Web Context 요청	220
9.5	Web Context 제어	222
9.5.1	소개	222
9.5.2	6 개의 명령어	223
9.5.3	webadmin 콘솔 툴을 사용한 context와 Servlet 제어 ...	223
9.5.4	웹 관리자를 통한 context와 Servlet 제어	224
9.5.5	결과	224

9.6	Web Context 모니터링	224
9.6.1	소개	224
9.6.2	webadmin 콘솔 툴을 사용한 Web context 모니터링...	224
9.6.3	웹 관리자를 통한 Web context 모니터링	224
9.6.4	결론	225
9.7	Web Context 튜닝	225
9.8	결론.....	225
10	가상 호스트	227
10.1	소개.....	227
10.2	가상 호스트의 개요.....	228
10.2.1	소개	228
10.2.2	기본 개념	228
10.2.3	가상 호스트의 규칙과 구조	229
10.2.4	가상 호스트의 주사용	231
10.2.5	기본 가상 호스트	231
10.2.6	javax.servlet.ServletContext.getContext() Method사용시 주의 사항.....	232
10.2.7	결론	232
10.3	가상 호스트 설정.....	232
10.3.1	소개	232
10.3.2	도메인 이름 등록	233
10.3.3	WEBMain.xml에 가상 호스트 설정.....	233
10.3.4	JEUSMain.xml 의 application에 가상호스트 맵핑하기 235	
10.3.5	웹 관리자를 통한 가상 호스트 설정	237
10.3.6	결론	237
10.4	가상 호스트를 통한 context 요청.....	238

10.4.1	소개	238
10.4.2	URL 매칭에 관한 기본 규칙	238
10.4.3	URL 매칭의 예	238
10.4.4	결론	245
10.5	결론.....	245
11	결론.....	247
A Webadmin	콘솔 툴 레퍼런스.....	249
A.1	소개.....	249
A.2	목적.....	249
A.3	실행.....	249
A.4	엔진 리스트와 선택 명령.....	251
A.5	제어 명령.....	251
A.6	상태 정보 명령어.....	252
A.7	구성 리포트 명령어.....	254
A.8	구성 설정 명령어.....	255
A.9	기타 명령어.....	257
B jspc	콘솔 툴 레퍼런스.....	259
B.1	소개.....	259
B.2	목적.....	259
B.3	실행.....	259
B.4	파라미터.....	260
C jspc2	콘솔 툴 레퍼런스.....	263
C.1	소개.....	263
C.2	목적.....	263
C.3	호출.....	263
C.4	파라미터.....	264
D WEBMain.xml XML	구성 레퍼런스	267

D.1	소개.....	267
D.2	XML Schema/XML Tree.....	268
D.3	Element Reference	293
D.4	WEBMain.xml 파일 예제	539
E	jeus-web-dd.xml XML 설정 레퍼런스	553
E.1	소개.....	553
E.2	XML Schema/XML Tree.....	554
E.3	Element 레퍼런스	559
E.4	jeus-web-dd.xml 파일 예제.....	622
F	TCP Listener API.....	625
F.1	소개.....	625
F.2	jeus.servlet.tcp 패키지.....	625
	abstract class TCPDispatcherConfig.....	625
	abstract class TCPServlet.....	628
	abstract class TCPServletRequest.....	629
	abstract class TCPServletResponse	630
G	JEUS WebContainer Properties.....	631
G.1	소개.....	631
G.2	환경 변수 참조.....	632
	색인.....	635

그림 목차

그림 1. JEUS의 다양한 서비스를 바탕으로 운영되는 Web Container	38
그림 2. JEUS Web Container의 기본 컴포넌트들.....	40
그림 3. JEUS Web 모듈에 관련된 기본 디렉토리 구조.....	42
그림 4. 매뉴얼의 계층 구조도.....	46
그림 5. 이 장에서 설명할 Web Container의 컴포넌트들.....	48
그림 6. JEUS Web container 와 관련 있는 디렉토리들.....	52
그림 7 DB connection pool 이 포함된 JEUS Web Container의 구조.....	75
그림 8. Web Container에 연관된 Context Group. 이 장에서 다루는 부분이 강조되어 있다.....	88
그림 9. Context Group의 상세 모습과 그 하위 컴포넌트들.....	89
그림 10. Context Group의 디렉토리 구조.....	96
그림 11. JEUS Web Container 중 Web Server 연결 부분 컴포넌트	112
그림 12. Web Container의 웹서버와 클라이언트 리스너.....	113
그림 13. 두 개의 웹서버가 두 개의 Web Container에 각각 연결되어 있는 작은 클러스터.....	115
그림 14. 두 개의 WebtoB 웹서버 A, B가 두 개의 Web Container들(Context Group)과 연결되어 있다.....	138
그림 15. 두 개의 WebtoB 리스너가 한 개의 WebtoB에 연결됨.....	139
그림 16. 8 개의 리스너, 4 개의 Context Group , 4 개의 Container가 2 개의 WebtoB Server와 연결된 모습.....	144
그림 17. 보안 리스너의 초기 디렉토리의 예.....	163
그림 18. Internet Explorer에서의 인증서 보안 경고창.....	172

그림 19. 보안된 SSL 연결을 통하여 hello.html이 보여지고 있다.(노란 자물쇠가 강조되어 있다).	172
그림 20. JEUS Web Container의 구조 중 Session에 관련된 부분들이 강조되어 있다.(Web Server도 Session Tracking에 관련되어 있음을 주의하라)	176
그림 21. 한 개의 Web Container가 한 개의 Session ID 쿠키를 사용한 세션 초기화.	178
그림 22. 요청 파라미터로 저장된 SessionID쿠키를 사용하여 같은 Web Container에 두 번째 요청을 보내는 과정	179
그림 23. SessionID쿠키를 이용하여 두 개의 Web Container와 Session을 시작한다.	181
그림 24. 세션 라우팅 없이는 클라이언트 Session은 이 클러스터링 환경에서 사라진다.	182
그림 25. 클라이언트의 첫 요청은 ContainerID(CID)를 가진 라우팅 기능을 내재한 Session을 반환한다.	183
그림 26. 세션 라우팅의 작동(CID로 Web Server는 원래의 Web Container에게 요청을 Routing 시킨다.)	183
그림 27. Session ID 쿠키와 연관된 Session 데이터는 Web Container 장애로 인해 소멸되었다.	184
그림 28. 별도의 세션 서버에 Session 데이터와 SessionID를 저장한다.	186
그림 29. Session 데이터는 Container의 내부 메모리에서 가져오지 않고 항상 중앙 세션 서버에서 가져온다.	187
그림 30. JEUS Web Container에 context/Web application과 관련된 것들	202
그림 31. 샘플 WAR파일 내용.	205
그림 32. 기본 context 디렉토리의 구조	207
그림 33. "MyContext" 라는 context의 디렉토리 구조.	209
그림 34. Web 브라우저에서 Servlet요청하기	221
그림 35. 404 오류가 발생하였을 경우 JEUS Web Container에 의해 생성되는 기본 오류 페이지.	222

그림 36. JEUS Web Container의 가상 호스트 컴포넌트.....	228
그림 37. 가상 호스트의 기본적인 개념.....	229
그림 38. context group들과 가상 호스트 그리고 context들간의 유효한 관계의 예	230
그림 39. 두 context 가 같은 context 경로를 가진 경우 같은 Web Container에서 DNS이름으로 구분하고 있다.....	231

표 목차

표 1. JEUS Web을 위한 환경 변수들	43
표 2. JEUS Web 하위 모듈을 위한 XML 설정 파일들	44
표 3. JEUS 보안 리스너 SSL -D 속성들	169
표 4. JEUS WebContainer Properties	632

매뉴얼에 대해서

매뉴얼의 대상

본 매뉴얼은 JEUS Web Container 컴포넌트들(Servlet, JSP, static contents)을 이용하여 J2EE Web 모듈들을 deploy 하고 관리하는 시스템 관리자와 개발자들이 읽어야 한다.

매뉴얼의 전제 조건

본 매뉴얼을 읽기 전에 알아야 할 사항은 다음과 같다.

1. Servlet deploy와 패키징에 관련된 실무적인 지식이 요구된다. 만약에 이런 지식이 없다면 <http://java.sun.com>에 있는 스펙 또는 Servlet에 관련된 서적을 참고한다.
2. JEUS Server 안내서를 읽고 이해한다.

참고: 본 매뉴얼에서는 기본적인 J2EE 또는 Servlet 에 대한 설명을 하지 않는다.

매뉴얼의 구성

JEUS 의 다른 사용 설명서들과 마찬가지로 본 매뉴얼은 다음의 방법 중 하나의 방법으로 읽을 수 있다.

1. 교과서와 같이 처음 장부터 차례대로 읽어 나갈 수 있다.
2. 각 장은 서로 다른 주제에 대하여 이야기 하고 있으므로, 알고 싶은 장들만 선택하여 읽을 수 있다.
3. 부록 부분만을 참고 한다면 단순한 참고서로도 사용 가능하다. 이렇게 하기 위해서는 주요 장들을 정독하고 시스템에 대한 충분한 이해가 얻어진 후에 이 방법을 선택하길 권한다.

본 매뉴얼은 다음과 같이 11 개의 장과 5 개의 부록으로 구성되어 있다.

1. **소개:** 인사말과 개요.
2. **따라하기:** 단계별 튜토리얼로 빠른 시작을 유도.
3. **JEUS Web에 대한 개요:** JEUS Web 모듈에 대한 기본적인 개념과 컴포넌트에 대한 설명.
4. **Web Container:** JEUS Web Container(engine)에 대한 설명. JEUS Web 모듈의 최상위 컴포넌트.
5. **Database Connection Pool:** JEUS Web Container에 내장된 database connection pool에 대한 설명.
6. **Context Group:** JEUS Web Container의 웹 어플리케이션(context) 그룹핑에 대한 설명.
7. **웹서버 연결과 클러스터링:** 어떻게 Context와 가장 앞 단의 Web Server들 또는 직접 HTTP 클라이언트와 연결되는지 설명. 클러스터링에 대해서도 언급.
8. **Session Tracking:** 클라이언트의 요청이 있을 때, Web Server 클러스터에 있는 서로 다른 Web Container들에 어떻게 클라이언트 session 데이터가 라우팅 되고 저장되는지 설명.
9. **Context(웹 어플리케이션):** Web container 의 주요 실행 컴포넌트들에 대한 사용을 설명(즉, J2EE 웹 어플리케이션).
10. **가상 호스트:** 가상 호스트의 개념과 클라이언트 요청에 포함된 도메인 이름에 따라 분기하여 처리하는 방법에 대한 설명
11. **결론.**

부록:

- A. **Webadmin 콘솔 툴 레퍼런스:** webadmin 콘솔 툴에 대한 설명.
- B. **jspc 콘솔 툴 레퍼런스 :** JSP 배치 컴파일러(jspc)에 대한 설명.
- C. **jspc2 콘솔 툴 레퍼런스 :** jspc2 배치 컴파일러에 대한

- D. WEBMain.xml XML 구성 레퍼런스 : Web container의 WEBMain.xml 설정 파일에 대한 상세한 설명.
- E. jeus-web-dd.xml XML 설정 레퍼런스 : JEUS context (웹 어플리케이션) deployment descriptor에 대한 상세한 설명.
- F. TCP Listener API : TCP listener를 사용한 통신 프로토콜을 구현하기 위한 API(7 장에 설명됨).

관련 매뉴얼

관련된 문서들은 다음과 같다:

- J2EE 1.4 스펙.
- Servlet 2.4 스펙.
- JSP 2.0 스펙.
- JEUS Server 안내서.
- JEUS Web Server 안내서.

일러두기

표기 예	내용
텍스트	본문, 12 포인트, 바탕체 Times New Roman
<i>텍스트</i>	본문 강조
CTRL+C	CTRL 와 동시에 C 를 누름
public class myClass { }	Java 코드
<div style="border: 1px solid black; padding: 2px; display: inline-block;"><system-config></div>	XML 문서

표기 예	내용
참조: / 주의:	참조 사항과 주의할 사항
Configuration 메뉴를 연다	GUI 의 버튼 같은 컴포넌트
JEUS_HOME	JEUS 가 실제로 설치된 디렉토리 예)c:\jeus5
j eusadmi n nodename	콘솔 명령어와 문법
[파라미터]	옵션 파라미터
< xyz >	‘<’와 ‘>’ 사이의 내용이 실제 값으로 변경됨. 예)<node name>은 실제 hostname 으로 변경해서 사용
	선택 사항. 예) A B: A 나 B 중 하나
...	파라미터 등이 반복되어서 나옴
?, +, *	보통 XML 문서에 각각 “없거나, 한 번”, “한 번 이상”, “없거나, 여러 번”을 나타낸다.
...	XML 이나 코드 등의 생략
<<FileName.ext>>	코드의 파일명
그림 1.	그림 이름이나 표 이름

OS 에 대해서

본 문서에서는 모든 예제와 환경 설정을 Microsoft Windows™의 스타일을 따랐다. 유닉스같이 다른 환경에서 작업하는 사람은 몇 가지 사항만 고려하

면 별무리 없이 사용할 수 있다. 대표적인 것이 디렉토리의 구분자인데, Windows 스타일인 “\”를 유닉스 스타일인 “/”로 바꿔서 사용하면 무리가 없다. 이외에 환경 변수도 유닉스 스타일로 변경해서 사용하면 된다.

그러나 Java 표준을 고려해서 문서를 작성했기 때문에, 대부분의 내용은 동일하게 적용된다.

용어설명

다음에 나오는 용어들은 본 매뉴얼에서 주로 사용되는 것들이다. 이해하지 못하거나 의미가 불분명한 것들은 이 리스트에서 제공된 설명을 통하여 이해하도록 하자.

용어	정의
가상 호스트 (Virtual Host)	클라이언트 요청을 HTTP 헤더의 도메인 이름에 따라 라우팅 되게 한다. 10 장을 보기 바란다.
웹 어플리케이션	Context 를 참조한다.
쿠키	HTTP 요청에 포함된 적은 양의 헤더 정보이다. HTTP 요청 시에 쿠키는 클라이언트를 Web Container가 식별할 수 있는 정보를 지닌다. 쿠키는 SessionID 와 Session Tracking 을 구현하기 위하여 기본적으로 사용된다.
클라이언트	JEUS Web Container가 이해할 수 있는 프로토콜로 말하고 있는 모든 종류의 소프트웨어 컴포넌트가 “client”로 지칭될 수 있다. 이러한 클라이언트들은 HTTP Web Browser와 지원되는 웹서버들이 있을 수 있다(Apache 와 WebtoB).

용어	정의
클라이언트 요청	클라이언트의 리소스 요청은 웹 서버나 Web Container 에서 받는다. 모든 클라이언트의 요청은 Web 환경일 때 HTTP 프로토콜을 사용한다.
클러스터링	서로 연결된 Web Container 와 Web Server 들의 그룹이다. 성능과 안정성을 증가하기 위해 사용된다.
Apache	Open-source Web Server.
Context	연관 있는 Servlet 들과 JSP 들 등 Web 자원을 지칭하는 용어이다. 그러므로, Context 는 클라이언트에 의해 요청되는 각각의 application 이 된다.
Context Group	여러 개의 Context 들을 묶고 이들에게 운영시의 기반을 제공한다. “Web Container 내의 Web Container”로 볼 수 있다.
HTTP	Hypertext Transfer Protocol. Web 에서 사용되는 프로토콜이다.
HTTPS	HTTP 위에 보안성이 첨가된 SSL 연결을 말한다.
JEUS	Java Enterprise User Solution 의 약자이다. 본 매뉴얼에서 소개되는 JEUS 버전 5.0 은 J2EE 1.4 인증된 WAS 이다.
jeus-web-dd.xml	JEUS 에서만 사용되는 웹 어플리케이션(컨텍스트)을 위한 설정 파일이다.

용어	정의
JEUS Web Server	기능이 제한된 WebtoB Web Server 이다. 이 버전은 JEUS 5.0 에 기본으로 탑재되어 있고 JEUS Web Container 의 내부 HTTP Listener 로 사용된다. 상세한 설명은 JEUS Web Server 안내서를 참조한다.
JSP	<i>Java Server Pages</i> . 자바 형식의 태그가 들어 있는 Web 콘텐츠 파일이다(HTML 또는 다른 markup 언어로 쓰여진다). JSP 는 Servlet 의 "반대"로 볼 수 있다. 실행되기 전에 JSP 는 Servlet 클래스로 전환이 된다.
jspc	JSP 컴파일러이다. JSP파일을 Servlet클래스로 미리 컴파일 하기 위해 사용된다. 부록 B를 참조한다.
Listener	JEUS 에서 사용되는 Web Container 내부의 작은 모듈을 일컫는다. 이 모듈은 Web Container 와 외부 시스템간의 통신을 담당한다. "Listener"라는 용어는 클라이언트로부터의 요청을 듣는다는 의미에서 사용된 것이다. 이 클라이언트에는 Web Server 도 될 수 있고, Web 브라우저도 될 수 있다 (Listener 의 종류에 따라 다름).
Servlet	동적인 Web 콘텐츠(HTML 을 동적으로 생성하는 것 같은)를 제공하는 자바 클래스.
Servlet Container	<i>Web container</i> 를 참조한다.
Servlet engine	<i>Web container</i> 를 참조한다.
Session	동일한 클라이언트에 의해 제한된 시간 내에 수행된 관련 있는 동작들의 집합이다.

용어	정의
Session ID	Session 을 정의하는 ID 이고 따라서 특정 Session 에 소속된 클라이언트도 구별하게 된다.
Session Routing	클라이언트의 요청을 클러스터 내의 해당 Web Container 에게 전달해 줄 수 있는 기능이다.
Session Server	JEUS 에서 사용되는 클라이언트 Session 데이터의 안정적인 공급 저장소.
Session Tracking	한 클라이언트의 Session 데이터(식별 자)를 여러 개의 HTTP 요청 중 식별할 수 있는 기능이다.
SSL	<i>Secure Socket Layer</i> . 소켓 기반 연결(HTTP와 같은)로 전달된 메시지의 인증과 기밀성, 정합성을 보장한다.
URL rewriting	Session Tracking 을 위해 쿠키를 사용하는 대신 사용할 수 있는 방법. HTTP 헤더에 클라이언트 정보를 담는 대신에 클라이언트에 전달되는 Web 문서의 HTTP 링크에 URL 파라미터로 첨가된다.
WAR 파일	<i>Web ARchive</i> 파일. Context(웹 어플리케이션)을 가지고 있는 표준 JAR 파일이다. 이 용어는 J2EE 스펙에 정의되어 있다.
WAS	<i>Web Application Server</i> 의 약자로, 복잡한 웹 어플리케이션을 실행하고 관리하는 미들웨어이다.
Web	본 매뉴얼에서는 HTTP, HTML, Servlet, JSP 등에 관련된 것들을 일컫는다 (실제 정의는 HTTP 프로토콜을 의미한다).

용어	정의
Web Container	웹 어플리케이션을 실행시키기 위한 최상위의 주요 소프트웨어 컴포넌트이다. 이 컴포넌트는 “Servlet engine”과 “Servlet Container”와 동격이다.
webadmin	JEUS Web Container를 위한 관리 툴이다. 이 툴은 명령 창에서 수행된다. 부록 A 를 참조한다.
WEBMain.xml	JEUS Web Container의 주 설정 파일이다. 이 파일은 XML 포맷으로 되어 있다. 상세한 정보는 부록 D를 참조한다.
WebtoB	TmaxSoft 의 고성능 웹서버 (HTTP Server)이다.

연락처

Korea

Tmax Soft Co., Ltd
18F Glass Tower, 946-1, Daechi-Dong, Kangnam-Gu, Seoul 135-708
South Korea
Tel: 82-2-6288-2114
Fax: 82-2-6288-2115
Email: info@tmax.co.kr
Web (Korean): <http://www.tmax.co.kr>

USA

Tmax Soft, Inc.
560 Sylvan Ave, Englewood Cliffs NJ 07632
USA
Tel: 1-201-567-8266
FAX: 1-201-567-7339
Email: info@tmaxsoft.com
Web (English): <http://www.tmaxsoft.com>

Japan

Tmax Soft Japan Co., Ltd.
6-7 Sanbancho, Chiyoda-ku, Tokyo 102-0075
Japan
Tel: 81-3-5210-9270
FAX: 81-3-5210-9277
Email: info@tmaxsoft.co.jp
Web (Japanese): <http://www.tmaxsoft.co.jp>

China

Beijing Silver Tower, RM 1507, 2# North Rd Dong San Huan,
Chaoyang District, Beijing, China, 100027
Tel: 86-10-6410-6148
Fax: 86-10-6410-6144
E-mail : info@tmaxchina.com.cn
Web (Chinese): <http://www.tmaxchina.com.cn>

1 소개

Sun Microsystems 에서 제정한 Servlet 2.4 과 JSP 2.0 스펙은, 간단한 개인 방문록에서 부터 수 백만 사용자가 사용하는 웹 사이트나 완벽한 보안을 보장하는 e-commerce 웹 사이트를 개발할 수 있는 표준을 제공하고 있다.

TmaxSoft JEUS 의 Web 모듈은 위에서 언급한 스펙을 효율적이고 유연하게 구현하였다.

JEUS 웹 어플리케이션 서버를 선택한 것에 대해서 감사를 드린다. JEUS 는 TmaxSoft 의 WebInOne 플랫폼의 하나이다. JEUS 를 사용하면 사용하기 쉽고 신뢰성이 뛰어나다는 것을 곧 느끼게 될 것이라 본다.

우선 JEUS Web 모듈을 간단하게 사용해 보고, 그 후에 각각의 하위 모듈들이 어떻게 설정되며, 컨트롤 및 모니터링 되는지도 살펴보면서, Web 사용자들이 가장 완벽한 Web 을 손 쉽게 사용하는 방법에 대해서 설명한다.

2 따라하기

다음에 소개하는 단계를 따라 하면 JEUS 를 간단하게 체험해 볼 수 있다.

아래의 예에서는 JEUS 의 노드 명이 “johan”이라고 가정한다. 사용자는 이 노드 명을, JEUS 가 설치된 머신의 이름으로 변경해야 한다.

1. 먼저 JEUS 가 시스템에 올바르게 설치되었고 경로와 시스템 변수들이 적절히 설정되어 있는지 확인한다(특히 시스템 경로에 “JEUS_HOME\bin\“ 디렉토리가 포함되어 있는지 확인한다). JEUS 의 설치에는 JEUS 설치 안내서를 참고한다.

참고: 본 매뉴얼에서는 “JEUS_HOME”이라는 지시어가 나올 때마다 실제 JEUS 의 설치 루트 디렉토리를 일컫는다고 생각해야 한다(예, “c:\jeus5”)

2. 적어도 한 개의 Web Container(Servlet engine)가 시스템에 설정되어 있어야 한다. “JEUS_HOME\config\johan\” 디렉토리의 JEUSMain.xml 과 일에는 다음의 굵은 글씨체와 같은 설정을 포함해야 한다.

<<JEUSMain.xml>>

```
<?xml version="1.0"?>
<jeus-system xmlns="http://www.tmaxsoft.com/xml/ns/jeus">
  <node>
    <name>johan</name>
    <sequential-start>>false</sequential-start>
    <engine-container>
      <name>mycontainer</name>
      <sequential-start>>false</sequential-start>
      <engine-command>
        <type>servlet</type>
        <name>engine1</name>
      </engine-command>
    </engine-container>
  </node>
</jeus-system>
```

3. JEUSMain.xml 파일에 선언되어 있는 Web Container 와 대응하는 Web Container 디렉토리가 존재하는지도 확인한다. 즉, 위에서 제시한 “engine1”으로 명명된 Web Container 가 “JEUS_HOME\config\johan\johan_servlet_engine1” 와 같이 존재해야 한다.
4. “JEUS_HOME\config\johan\johan_servlet_engine1” 디렉토리에는 WEBMain.xml 파일이 존재해야 한다. 이 파일은 각 Web Container 의 설정 파일이다. 이 파일은 다음과 같은 항목들을 포함해야 한다.

<<WEBMain.xml>>

```
<?xml version="1.0"?>
<web-container xmlns="http://www.tmaxsoft.com/xml/ns/jeus">
  <context-group>
    <group-name>MyGroup</group-name>
    <group-dochbase>webapps</group-dochbase>
    <webserver-connection>
      <http-listener>
        <port>8088</port>
        <thread-pool>
          <min>25</min>
          <max>30</max>
          <step>2</step>
        </thread-pool>
      </http-listener>
    </webserver-connection>
  </context-group>
</web-container>
```

5. 위에서 보여준 WEBMain.xml 의 한 부분은 “MyGroup”이라는 Context Group 을 Web Container 에 추가하고 내부 HTTP Listener 가 8088 포트를 사용하도록 설정한다(굵은 글씨 참조). “Context Group” 과 “Listener”에 대한 개념은 본 매뉴얼의 후반부에서 설명한다. 당분간 이 의미들에 대해서는 넘어가도록 한다.
6. 위와 비슷한 설정들이 존재한다고 했을 때 명령창을 하나 열고 ‘jeus’라고 명령한다. JEUS manager 가 실행될 것이다.
7. 다른 명령창을 하나 더 열고 이번에는 ‘jeusadmin johan’이라고 명령한다(여기서 또한 번 “johan”은 사용자 운영 시스템의 네트워크 ID 이름으로 바꾼다).

8. 관리자 이름과 패스워드를 입력한다(JEUS 설치 시에 설정한 정보와 동일하다).
9. jeusadmin 이 시작된다. ‘boot’ 라고 입력한다. JEUS 노드가 그 Engine Container 들의 engine 들과 함께 부팅된다. 이렇게 해서 Web Container (JEUSMain.xml 에 설정된 “engine1”)도 같이 시작된다.
10. ‘help’를 명령하면 Web Container 를 모니터링하고 컨트롤 할 수 있는 명령어들의 리스트를 볼 수 있다.
11. 다른 명령창을 열어 ‘webadmin johan_engine1’이라고 명령한다.
12. 사용자 이름과 패스워드를 입력한다.
13. webadmin 의 프롬프트에서 ‘help’라고 명령한다. Web Container 를 컨트롤하고 모니터링 하기 위한 명령어들의 리스트들이 출력된다. 예로 info 명령을 수행해 보기 바란다.
14. 웹관리자에서 위와 동일한 작업을 하고자 한다면 JEUS 웹 관리자 안내서의 웹 관리자의 시작 및 종료 에 대한 설명을 살펴보도록 한다.

이상으로 간단하게나마 JEUS 의 Web Container 를 경험해 보았다. 본 매뉴얼의 나머지 부분에서는 위에서 설명한 단계와 툴 그리고 설정들에 대한 상세한 설명을 하겠다.

위의 과정에서 문제가 발생했다면, JEUS 설치 안내서와 JEUS Server 안내서를 참고하여 올바르게 JEUS 환경을 설정해야 한다. 문제점의 원인을 좀 더 자세히 알아보려면 JEUS 관리자 콘솔 윈도우 로그에 남겨진 정보를 참고해 보는 것도 좋은 방법이다.

다음 장에서는 JEUS Web 에 대한 개요를 통하여 Web 에 대하여 좀 더 자세히 알아본다.

3 JEUS Web 에 대한 개요

3.1 소개

JEUS Web 모듈은 복잡하지만 JEUS 시스템 중에서 실용성 있는 컴포넌트 중의 하나이다. 이 모듈은 사용자에게 다이나믹하고 고성능이며 안정적인 Java 기반의 Web 콘텐츠를 제공하기 위해 만들어진 수십 개의 하위 컴포넌트들로 구성되어 있다(JSP, Servlet, 정적 HTML).

이 장의 나머지 부분에서는 Web 모듈의 주요 컴포넌트들에 대하여 살펴보고 어떻게 이들이 연계되어 있는지 알아본다. 나머지 장에서는 운영환경에서 어떻게 이 하위 컴포넌트들을 설정하고 제어하며 모니터링 하는지에 대하여 기술적으로 상세히 설명한다.

참고: 이 장과 나머지 장에서도 J2EE, Servlet, JSP 등에 대한 기본적인 것에 대해서는 다루지 않고 있다. 이들에 대한 실전 지식과 경험이 선행 조건인 만큼 해당 스펙이나 서적들을 우선 참고하기 바란다. 이 조건을 만족시키고 본 매뉴얼을 본다면 쉽게 이해할 것으로 본다.

3.2 JEUS Web Container

본 매뉴얼에 나오는 컴포넌트 중 가장 기본적이면서도 최상위 레벨의 컴포넌트가 바로 JEUS Web Container 이다. JEUS Web Container 는 J2EE 1.4, Servlet 2.4, JSP 2.0 을 준수하고 Servlet 과 JSP 뿐만 아니라 HTML 과 같은 정적 내용까지도 효과적으로 수행할 수 있다.

다음 장(4 장)에서는 JEUS Web Container에 대하여 자세히 다루고 5 장부터는 Web Container의 주요 하위 컴포넌트들에 대하여 다룬다.

다음 절(3.3 절)에서는 Web Container가 JEUS의 전체적인 시스템 구성에 어떻게 조합되는지 설명하고, 그 다음 절(3.4 절)에서는 Web Container 자체의 주요 하위 컴포넌트들과 그 기능들에 대해 살펴보겠다.

참고: JEUS 시스템의 설정 파일들과 툴에서는 "Web Container"를 "Web engine"이라고 부르기도 한다.

3.3 JEUS Web Container 와 JEUS 시스템

다음 [그림 1]은 JEUS 시스템 구조체 내에서 JEUS Web Container가 어떻게 연관되어 있는지 보여주고 있다. [그림 1]의 컴포넌트들은 Web Container들의 외부에 존재하는 것이고 웹 어플리케이션과 Web Container를 운영하기 위한 환경과 기본 인프라를 제공한다.

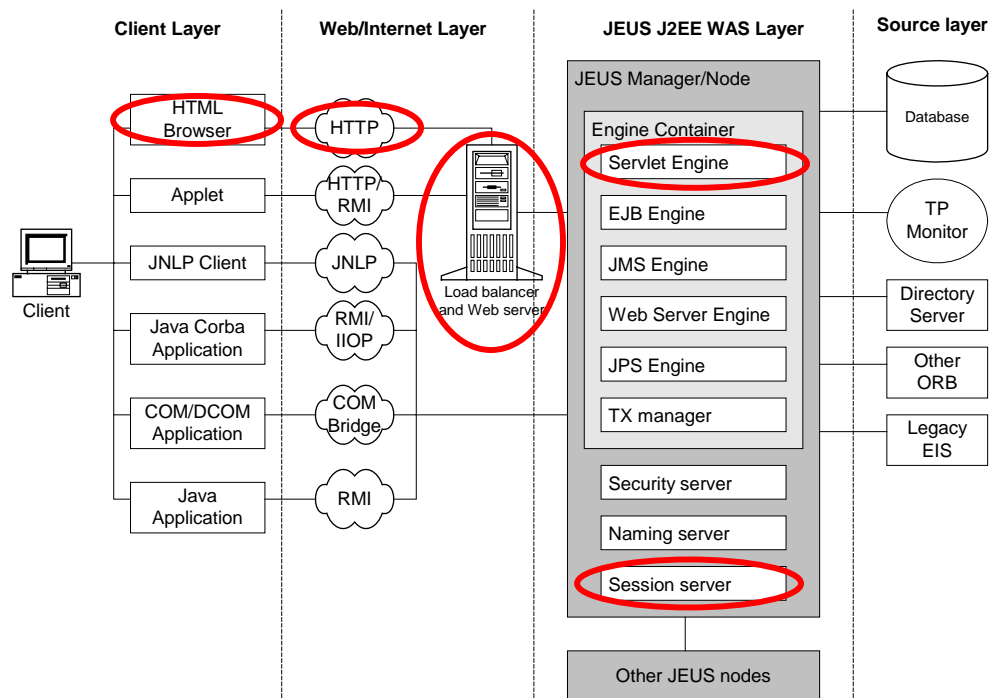


그림 1. JEUS 의 다양한 서비스를 바탕으로 운영되는 Web Container

[그림 1]의 주요 컴포넌트들은 다음과 같다.

- 클라이언트는 JEUS 시스템에 접근하는 객체이고 Web Container 의 서비스를 요청한다. Web 컴포넌트를 접근한다면 이 클라이언트는 사용자에게 의해 조작되는 HTTP 기반의 Web 브라우저가 된다.
- 부하 분산기(Load Balancer)는 클라이언트의 HTTP 요청을 웹서버 간에 나누어 전달하여 일정한 요청 수가 시스템에 전달될 수 있도록 한

다(예를 들어, WebtoB Server가 부하 분산기로 사용될 수 있다. 본 매뉴얼의 7 장을 참조한다).

- **웹서버**들은 클라이언트의 HTTP 요청을 받아 필요한 경우에 JEUS의 Web Container에 전달한다 (7 장을 참조한다).
- **JEUS 관리자**와 **노드**는 Web Container 가 동작하는 환경을 제공한다. 성능과 안정성을 극대화하기 위해서 여러 개의 노드를 클러스터링으로 묶기도 한다 (JEUS Server 안내서를 참조한다).
- **Web Container**(그림에서는 “Servlet engine”으로 표기됨)는 웹서버나 HTTP 클라이언트로부터 서비스 요청을 받고 웹 어플리케이션을 실행시켜 궁극적으로는 HTML 응답 페이지를 통해 응답을 준다. (4 장을 참조한다)
- **Session Server**는 분산된 환경에서 클라이언트의 Session을 관리하고 이 Session 데이터가 모든 Web Container에서 사용되도록 해준다(8 장을 참조한다)
- **JNDI Naming Server** 는 Servlet 과 JSP 와 같은 웹 어플리케이션이 시스템 내부의 객체나 리소스를 접근할 수 있도록 해준다 (EJB 나 Data Source 가 그 예라고 할 수 있다. JEUS Server 안내서의 JNDI 부분을 참조한다).
- **EJB** 와 **JMS engine** 은 웹 어플리케이션에 EJB 와 JMS 서비스를 제공한다. 이 engine 들은 JNDI naming Server 를 통하여 접근된다.
- **Security Server** 는 웹 어플리케이션들에게 보안 정책을 적용시킨다 (JEUS Server 안내서의 보안 부분을 참조한다).
- **TX (트랜잭션) 관리자**는 트랜잭션을 관리한다(JEUS Server 안내서의 JTS 부분을 참조한다).
- **맨 끝 단의 데이터베이스**는 많은 양의 데이터를 저장하는, 가장 일반적인 방법이다. 데이터베이스는 **database connection pool** 을 통하여 접근 가능하다(그림에서는 보이지 않았다).
- **Tmax Server** 는 TP 모니터링 서비스를 제공한다. Tmax Server 는 JEUSMain.xml 에 설정된 WebT connection pool 을 통하여 접근된다 (그림에서는 보이지 않는다. WebT 안내서를 참조한다).

3.4 JEUS Web Container 의 기본 컴포넌트들

지금까지 Web Container 가 JEUS 시스템 내부의 다른 컴포넌트들과 어떻게 연계되는지 살펴보았고, 지금부터는 각각의 컴포넌트들을 구성하며 Web Container 자체에 포함된 컴포넌트들을 자세히 살펴보겠다.

물리적으로 JEUS Web Container는 [그림 2]에 있는 컴포넌트들로 구성되어 있다고 할 수 있다.

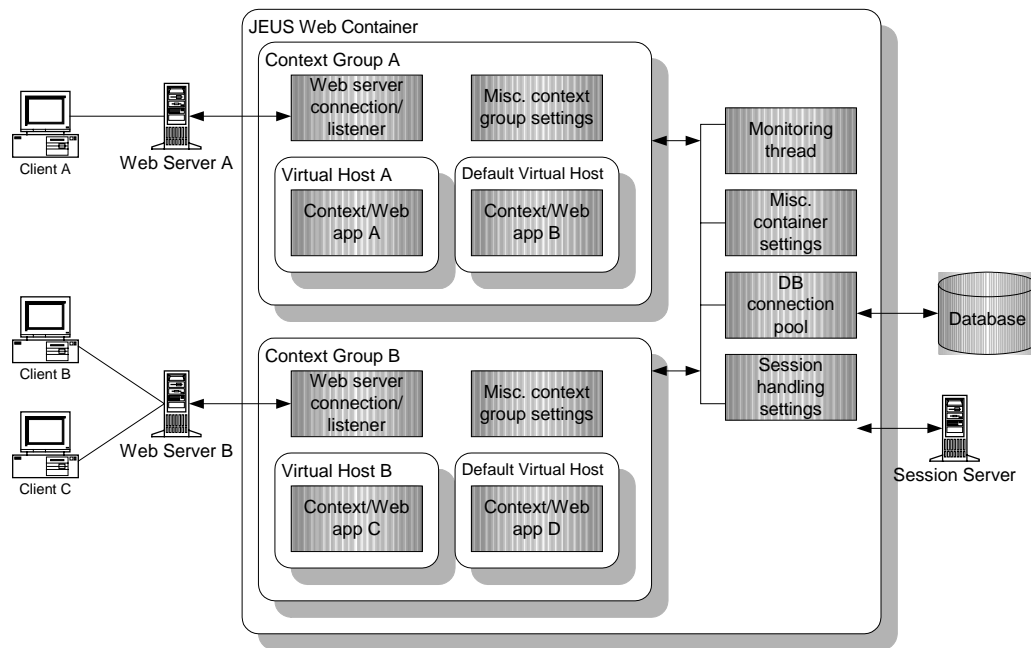


그림 2. JEUS Web Container 의 기본 컴포넌트들.

주요 컴포넌트들은 다음과 같다.

- **JEUS Web Container** 자체는 J2EE기반의 Web Container이다. 4 장에 이 컴포넌트에 대하여 자세하게 설명되어 있다.
- **Monitoring Thread** 는 Web Container 의 다른 컴포넌트들을 감시하는 역할을 한다.
- Container에 내장된 **database connection pool**은 웹 어플리케이션이 데이터베이스에 효율적으로 접근하기 위해 필요한 것이다. 5 장을 참조한다.

- 맨 앞 쪽의 **Web server connections**는 외부의 웹서버(WebtoB와 Apache)와 Container가 연결되기 위한 컴포넌트이다. Web server connection은 여러 개의 웹서버와 Web Container들이 서로 연결되어 성능과 안정성을 향상시킨 커다란 클러스터링으로 구성되도록 한다. 이에 대한 내용은 7 장을 참조한다. 여기서 중요한 점은 Web server connectivity가 *context group* 레벨에서 설정된다는 것이다.
- **Session handling**은 분산 환경에서만 사용되고, Session Routing 기술이나 Session Server를 사용하거나, 또는 두 기술을 모두 적용시켜 활용 가능하다(8 장을 참조한다).
- **context group**은 JEUS Web Container에 포함된 것이다. 이것은 여러 개의 context들과 가상 호스트들을 관리 가능한 단위로 묶고, 이들에게 운영환경을 제공해 준다. 6 장을 참조한다.
- **가상 호스트**는 특정한 domain name을 호출 했을 때, 해당 context에 보다 강력한 제어능력을 주기 위해 만들어 졌다. Context는 가상 호스트 내에 명시적으로 설정될 수도 있고, context group 바로 아래에 묵시적인 기본 가상호스트로도 설정될 수 있다. 10 장을 참고한다.
- **Context** (또는 웹 어플리케이션) 는 J2EE 스펙에 준하고 Container 안에서 운영된다. 이들은 본래 WAR(Web ARchive) 안에 포함되며, Web Container 내의 context group 또는 가상 호스트에 deploy 된다. 9 장을 참조한다.

다음에 나오겠지만 위에서 제시한 모든 기능들이 본 매뉴얼의 전체적인 구성이 된다.

3.5 Web 모듈 툴

JEUS Web 모듈에 관련된 툴들은 다음과 같다.

- **웹관리자**는 JEUS Web Container 를 대상으로 작업할 수 있는 가장 좋은 툴이다. 이 툴의 사용법은 JEUS 웹 관리자 안내서에 잘 설명되어 있다.
- **Webadmin** 콘솔 툴은 Web Container에 대한 기본 제어와 모니터링 기능을 제공한다. 부록 A 에 이 툴에 대한 사용법이 설명되어 있다.

- **JSP batch compiler (jspc)** 는 처음 접근하는 JSP에 대하여 빠른 수행 능력을 주기 위해 JSP를 미리 컴파일 할 수 있는 기능을 제공한다. 부록 B에 이 툴에 대한 사용법이 설명되어 있다.
- **jeusadmin** 콘솔 툴은 명령창에서 Web engine 을 제어할 수 있는 툴이다. 이 툴은 JEUS Server 안내서에 설명되어 있다.

3.6 Web 디렉토리 구조

이 절에서는 JEUS Web Container를 대상으로 작업할 때 염두해야 할 JEUS 설치 시의 디렉토리를 설명한다. [그림 3]

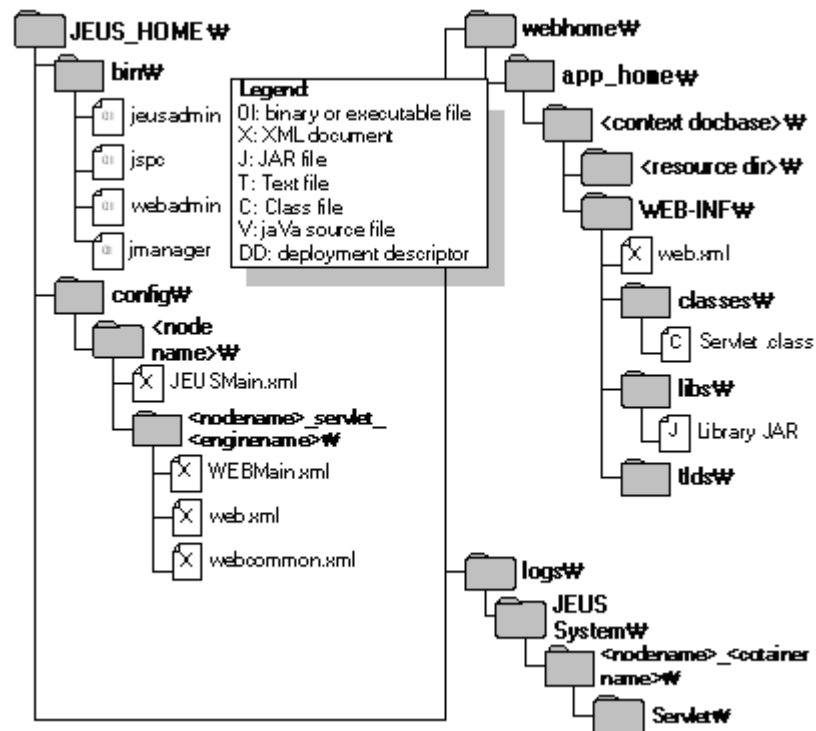


그림 3. JEUS Web 모듈에 관련된 기본 디렉토리 구조

주의 깊게 살펴봐야 할 디렉토리는 다음과 같다.

- **JEUS_HOME** 디렉토리는 JEUS 제품을 설치할 때 선택된다 (예. c:\jeus5\).
- **bin** 디렉토리는 jeusadmin, webadmin, jspc 들을 포함하고 있는 디렉토리이다.

- **config\<nodename>** 디렉토리는 JEUS 시스템에 설정되어 있는 각 Web Container 에 해당하는 하위 디렉토리들을 가지고 있다.
 - **JEUS_HOME\config\<nodename>\<nodename>_servlet_<engine name>**: 이 디렉토리는 Web Container 의 XML 기반 설정 파일을 가지고 있다. (WEBMain.xml)

이 디렉토리는 “web.xml”과 “webcommon.xml”파일들을 가지고 있다. 전자는 자신만의 web.xml 파일을 가지고 있지 않은 웹 어플리케이션을 위한 기본 web.xml 파일이다. 후자는 모든 웹 어플리케이션들에 적용되는 “공통되는” 설정 파일이다. (이 파일은 표준을 따르는 web.xml 과 동일한 파일이다. J2EE XSD 인 “web-app_2_4.xsd” 에는 이 파일들에 대한 정의가 들어있다.)
- **logs\JeusSystem\<nodename>_<containername>\servlet** 디렉토리는 Web Container 의 로깅 데이터를 저장하고 있다.

3.7 JEUS Web 관련된 시스템 변수들

[표 1]은 JEUS Web 모듈에 관련된 시스템 환경 변수들을 보여주고 있다. 이 환경 변수들은 수정 가능하지만 일반적으로 권장하지는 않는다. 꼭 필요한 경우에만 XML 설정 파일에서 이 환경 변수들을 수정하는 것이 더 일반적이라고 하겠다.

이 환경 변수들은 JEUS JVM 에 시스템 설정을 전달하기 위하여 사용된다. 이 시스템과 JVM 사이의 변수 매핑은 JEUS_HOME\bin\ 디렉토리에 위치한 다양한 JEUS 스크립트에 존재한다.

표 1. JEUS Web 을 위한 환경 변수들

환경변수	의미	예제
JEUS_HOME	JEUS 가 설치된 기본 위치.	C:\Jeus5 (반드시 설정되어야 한다.)

환경변수	의미	예제
JEUS_WSDIR	JEUS Web Server (WebtoB)의 홈 디렉토리	C:\Jeus5\webserver (기본값)

이 시스템 값들을 설정하는 방법은 OS에 따라 각기 다르다. 설정 방법에 대한 내용은 해당 OS 매뉴얼을 참고하기 바란다.

참고: 이 값들의 대부분은 JEUS의 설치 시에 자동으로 설정된다. 일반적인 경우라면 이 설정은 그대로 두고 사용하는 것이 좋다.

중요: 본 매뉴얼을 집필할 당시, 어떤 시스템 변수들은 JEUS_HOME\bin\ 디렉토리 내의 스크립트에 직접 반영이 되지 않은 것도 있다. 이 의미는 이 값들을 OS 환경 변수 값을 수정함으로 또는 스크립트 외부에서 수정하는 것이 실제로 적용되지 않을 수도 있다는 것이다. 그러므로, 이 값들은 항상 스크립트 파일 내에서 설정하도록 한다.

3.8 JEUS Web 과 관련된 XML 설정 파일들

[표 2]는 JEUS Web과 관련된 설정 파일들을 보여주고 있다.

표 2. JEUS Web 하위 모듈을 위한 XML 설정 파일들.

파일명 (XSD 파일명은 괄호 안에 명시)	위치	목적	상세정보 위치
JEUSMain.xml (jeus-main-config.xsd)	JEUS_HOME\config\<node>\	JEUS 시스템에 새로운 Web Container를 추가할 때 사용.	JEUS Server 안내서

파일명 (XSD 파일명은 괄호 안에 명시)	위치	목적	상세정보 위치
WEBMain.xml (web-main-config.xsd)	JEUS_HOME\config\ <nodename>\<nodename>_servlet_<engine name>\<="" td=""> <td>Web Container 주요 설정.</td> <td>본 매뉴얼</td> </nodename>\<nodename>_servlet_<engine>	Web Container 주요 설정.	본 매뉴얼
jeus-web-dd.xml (jeus-web-dd.xsd)	<context name>\WEB-INF\	웹 어플리케이션 (context) deployment descriptor.	본 매뉴얼.
web.xml (web-app_2_4.xsd)	<context name>\WEB-INF\	J 2 E E 표준 Web 모듈 deployment descriptor.	Servlet 2.4 스펙
web.xml (web-app_2_4.xsd)	JEUS_HOME\config\ <nodename>\<nodename>_servlet_<engine name>\<="" td=""> <td>Web.xml 을 개별적으로 가지고 있지 않은 context 의 기본 “web.xml”파일</td> <td>Servlet 2.4 스펙</td> </nodename>\<nodename>_servlet_<engine>	Web.xml 을 개별적으로 가지고 있지 않은 context 의 기본 “web.xml”파일	Servlet 2.4 스펙
webcommon.xml (web-app_2_4.xsd)	JEUS_HOME\config\ <nodename>\<nodename>_servlet_<engine name>\<="" td=""> <td>Web Container 의 모든 context 에 적용되는 공통되는 설정</td> <td>Servlet 2.4 스펙</td> </nodename>\<nodename>_servlet_<engine>	Web Container 의 모든 context 에 적용되는 공통되는 설정	Servlet 2.4 스펙

3.9 본 매뉴얼의 구성에 대한 상세 정보

[그림 4]에서는 본 매뉴얼의 구조를 이 장에서 주어진 정보를 기반으로 하여 보여주고 있다.

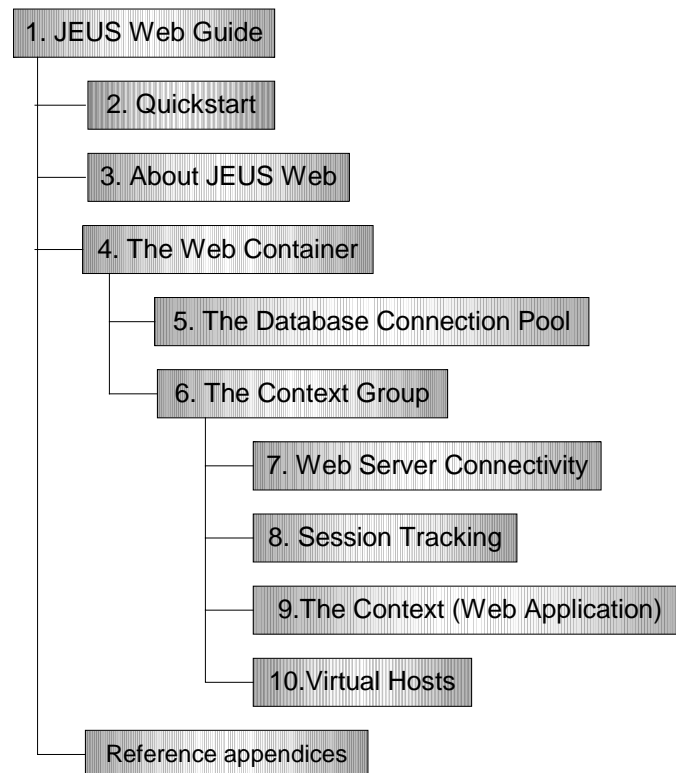


그림 4. 매뉴얼의 계층 구조도

위의 그림에서는 본 매뉴얼의 계층 구조를 확실히 보여주고 있다. 가장 위에 위치한 Web Container 를 시작으로 각 하위 컴포넌트들을 살펴보기로 한다.

3.10 결론

이것으로 JEUS Web 모듈에 대한 가장 상위의 구조적 체험이 마무리 되었다.

우리는 JEUS Web Container 가 어떻게 JEUS 시스템과 전체적으로 융화되어 있는지 살펴보았고, 어떤 하위 컴포넌트들이 존재하는지도 살펴보았다.

JEUS Web 모듈을 관리할 수 있는 툴들에 대해서도 간단히 살펴보았고, Web 모듈과 연관된 디렉토리 구조도 알아보았다.

다음 장에서는 JEUS Web Container 에 대하여 좀 더 자세히 살펴보겠다. 이 내용은 본 매뉴얼의 다른 부분을 이해하는데 기본이 되는 내용이다.

4 Web Container

4.1 소개

JEUS 시스템의 Web Container 는 Servlet 2.4 와 JSP 2.0 스펙으로 만들어진 자바 기술 기반의 다양한 Web 페이지의 수행을 위한 환경과 기반, 서비스를 제공한다. 물론, Web Container 는 HTML 과 이미지 파일들과 같은 정적 콘텐츠들도 운영하고 관리할 수 있다.

Web Container 는 JEUS Web 모듈의 최상위 컴포넌트이다. 여기에는 많은 수의 하위 모듈들과 설정할 수 있는 기능들이 존재한다. 예를 들어 여러 개의 Web Container 들은 부하 분산과 failover 기능을 제공하기 위하여 클러스터로 구성될 수도 있고, 웹을 통해 사용자의 요청을 처리하기 위하여 WebtoB 나 Apache Web Server 와 같은 Web 단에 연결될 수도 있다.

이러한 모든 개념들과 모듈들은 이 장에서 고루 다루겠다. 이 장은 JEUS Web 모듈의 하위 모듈과 기능들을 소개하는 시작점으로 보면 될 것이다. Context group 이나 Web application 과 같이 중요한 개념들은 실제 Web Container 의 하위 컴포넌트들이지만 차지하는 분량이 많은 관계로 별도의 장에서 따로 설명한다.

4.2 JEUS Web Container 의 개요

4.2.1 소개

이 장에서는 JEUS Web Container 의 구조와 그 하위 컴포넌트 및 중요한 기능들에 대해서 살펴보겠다.

4.2.2 Web Container 의 구조

이미 Web Container 의 가장 대략적인 그림은 보았지만 그 보다 더 명확한 개념의 전달을 위하여 3.4 절에서 나온 [그림 2]를 다시 한 번 보자. [그림 5]에서는 이 장에서 주로 설명할 부분을 타원으로 표시하고 있고 나머지 부분들은 나중의 장들에서 설명하겠다.

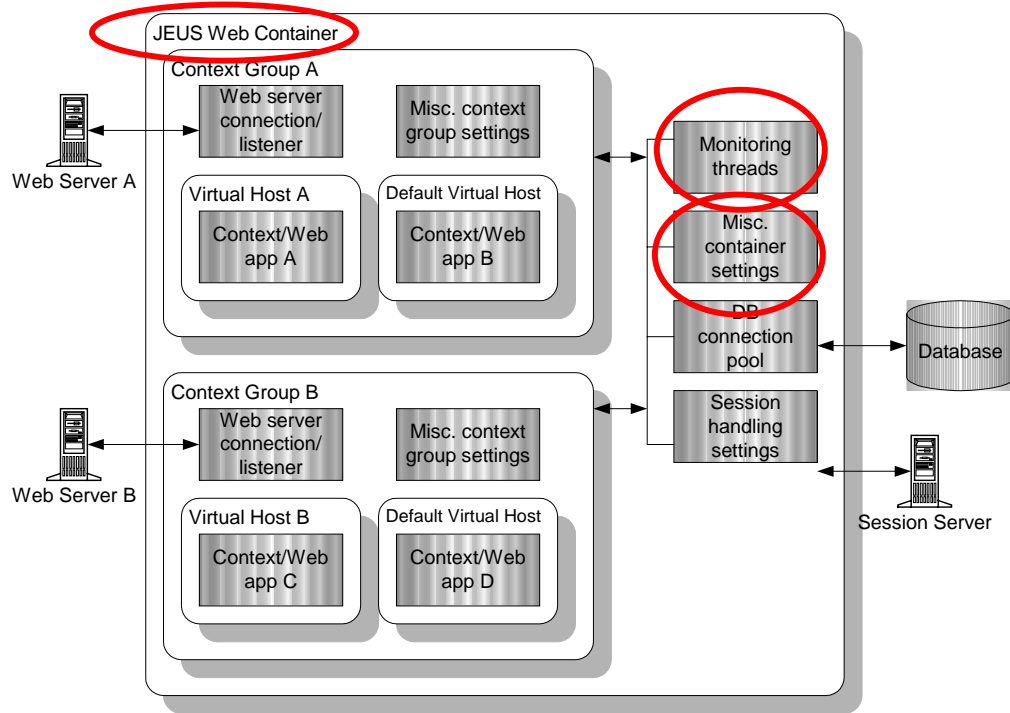


그림 5. 이 장에서 설명할 Web Container 의 컴포넌트들.

4.2.3 Web Container 의 주요 하위 컴포넌트들과 기능들

다음에 설명하는 일곱 개의 하위 절에서는 Web Container 의 가장 중요한 하위 모듈들을 설명한다. 이 모듈들은 각자가 많은 기능들을 포함하고 있지만 여기서 모두 설명하지 않고 별도의 장들에서 설명하도록 한다.

이 하위 모듈들과 기능들은 다음과 같다:

- 자동 모니터링
- stdout 와 stderr redirection.
- Context Group
- Database Connection Pool
- Session 클러스터
- 보안 스위치
- shutdown timeout

위에서 열거된 리스트는 JEUS Web Container 의 최상위 뷰를 표현하고 있다. 이들은 Web Container 의 주요 하위 모듈인 context group 과 context 가 사용할 수 있는 모듈들과 서비스들을 의미한다.

4.2.4 자동 모니터링

Web Container 자체 감시의 가장 중요한 부분은 자동 모니터링을 이용하는 것이다. 이 기능은 Container 내의 모든 자원과 풀들의 상태를 감시하고 문제가 발생하였을 때 대응할 수 있는 기능을 한다.

현재 4 가지의 모니터링이 존재한다.

- **thread pool monitor**는 Container의 worker thread pool을 모니터링 한다.(7 장 참조)
- **class-reloading monitor**는 Web Container에 등록된 Servlet클래스들의 변경을 모니터링 한다. (9 장 참조)
- **session monitor**는 클라이언트 Session 기한 만료를 모니터링하고 사용되지 않는 Session은 제거한다. (8 장 참조)
- **database connection monitor**는 Web Container가 관리하는 database connection pool을 모니터링하고 필요에 따라 database connection을 추가 또는 삭제 한다. (5 장 참조)

4.2.5 stdout 와 stderr redirection

설정 여부에 따라 stdout 과 stderr 를 사용하면 JEUS_HOME\logs\JeusSystem\<node name>_<container name>\servlet\stdout_<date>.log 와 JEUS_HOME\logs\JeusSystem\<node name>_<container name>\servlet\stderr_<date>.log 에 각각 로그를 남기게 할 수 있다.

이 redirection 기능이 지정되지 않으면 Web Container 를 관리하는 JEUS 관리자의 콘솔 윈도우에 출력된다(이것이 기본 설정이다).

4.3 절에 설정 정보가 설명되어 있다.

4.2.6 Context Group

각 Web Container 는 하나 이상의 context group 을 포함할 수 있다. Context group 은 여러 개의 context 와 가상호스트를 그룹화 할 수 있는 JEUS 에서만 제공하는 기능이다. Context 는 Container 에서 수행되는 실제 Web application 과 같은 개념이다. Context group 은 그 안에 포함된 모든 context 와 가상 호스

트에 공통적으로 적용되는 설정과 서비스들을 가지고 있다. Context group 의 또 한 가지 중요한 기능은 Web Server 연결과 클러스터 환경에서의 Session Handling 그리고 active management 이다.

Context group은 다소 기초적이고 많은 양의 설정을 담고 있으므로 6 장에서 따로 상세히 설명한다. context 또한 같은 경우이므로 9 장에서 상세히 다루고 가상 호스트는 10 장에 설명한다.

4.2.7 Database Connection Pool

DB connection pool 은 이미 초기화된 database connection 의 풀을 유지하여 database connection 의 사용을 효과적으로 하기 위하여 사용된다. 이 connection 들은 연결을 클라이언트가 직접 만들 필요 없이 EJB 나 Servlet, application 클라이언트들에게 제공된다. 클라이언트가 database connection 을 다 사용하면 다른 클라이언트가 사용할 수 있도록 풀에 반환된다. (여기서 의미하는 “클라이언트”는 데이터베이스에 접근하려는 모든 종류의 프로그램을 의미한다.)

Web Container 에서는 JEUS Server 에서 제공하는 DB connection pool 대신에 몇 가지의 전용 DB connection pool 들이 추가되어 사용될 수 있다.

5 장에 더 상세한 설명이 되어 있다.

4.2.8 Session 클러스터링

여러 개의 Web Container 가 여러 개의 Web Server 들과 클러스터링이 구성되어 있을 때 클라이언트 요청이 적절한 Web Container 로 전달되려면 어떻게 Session 객체(cookie)를 관리 해야 하는지 지정해야 한다.

클러스터링된 환경에서의 Session에 대한 사항은 8 장에서 설명한다.

4.2.9 Logging

Web Container 에서 생성되는 로그 메시지는 system log, user log, access log 세 가지 이다. 이 중 system log 는 JEUSMain.xml 에서 system-logging 을 통해 설정한다. user-log 및 access-log 는 WEBMain.xml 의 logging 을 통해 설정한다. logging 설정은 Web Container 전체 혹은 하위 Context Group 별로 따로 설정할 수 있다.

4.2.10 보안 스위치

각 Web Container 에는 단 하나의 보안 스위치가 존재한다. 이 스위치는 인증 과정을 거칠 것인지 아닌지를 설정한다. 이 스위치가 꺼져 있으면 이는 web.xml 에 설정된 모든 보안성 제약들이 무시된다.

이 기능은 주로 테스트나 개발 목적으로 사용되고 또는 타 소프트웨어 제품이 보안 관련 설정을 담당할 때 사용된다. 이 기능을 꺼 놓으면 어느 정도의 성능 향상 효과를 볼 수 있다.

4.2.11 Shutdown Timeout

이 설정은 관리자가 “down” 명령을 내렸을 때 Web Container 가 실제로 종료 되기까지 Web Container 가 기다리는 시간을 지정한다. 이 설정은 관리자가 Web Container 는 worker thread 들이 작업을 모두 마치기까지 기다리도록 설정할 수 있는 것이다. “down” 명령을 수행했을 때 어떤 worker thread 도 실행되고 있지 않으면 Container 는 이 설정을 무시하고 바로 종료를 수행한다.

이것으로써 Web Container 의 주요 하위 컴포넌트들의 설명을 마친다.

4.2.12 요청 처리와 Worker Thread Pool

지금쯤 대부분의 독자들은 JEUS Web Container 가 실제로 어떻게 클라이언트나 Web Server HTTP 요청을 받고 처리하는지 궁금해 할 것이다.

기본적으로, 이것은 WEBMain.xml 의 이른바 리스너를 등록함으로써 가능해진다. 이 리스너는 클라이언트와 Web Server 의 요청을 듣고 특정한 worker thread(worker thread pool 에서 가져옴)에게 요청을 처리하고 수행하여 HTTP/HTML 응답을 하도록 한다.(주로 Servlet 을 호출함으로써 가능해진다.)

그러나, 아직은 리스너, worker thread 그리고 Client/Web Server 연결에 대한 이슈를 논할 때가 아니다. 왜냐하면 이 컴포넌트들은 실제로 context group 레벨에서 설정이 되기 때문이다. 6 장과 7 장에서 이 정보에 대해 상세히 알아본다.

4.2.13 Web Container 의 디렉토리 구조

[그림 6]는 JEUS Web Container를 다룰 때 가장 많이 사용되는 디렉토리들이다.

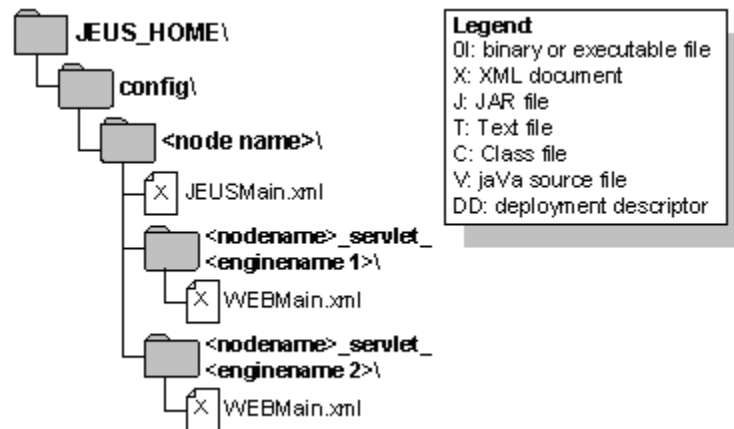


그림 6. JEUS Web container 와 관련 있는 디렉토리들.

[그림 6]의 디렉토리들은 다음과 같다.

- **JEUS_HOME** JEUS 제품을 설치하면 생성되는 디렉토리이다.
- **JEUS_HOME\config\<nodename>** 디렉토리는 JEUS Web Container의 존재를 선언하는 JEUSMain.xml 을 가지고 있다.
- **\config\<nodename>\<nodename>_servlet_<enginename>** 디렉토리는 Web Container 의 홈 디렉토리이다. JEUS 노트에 있는 각 Web Container 설정마다 이와 같은 디렉토리가 존재해야 한다. 각각의 디렉토리에는 WEBMain.xml 파일이 포함되어 있다. WEBMain.xml 파일에는 Web Container 의 모든 설정이 들어있다. 이 파일의 내용은 이 문서의 대부분에서 다루고 있다.

참고: 로그는 Web Container를 통하여 직접적으로 생성되지 않고 context group 레벨에서 설정하고 생성된다. (6 장 참조)

4.2.14 결론

이것으로 Web Container 의 주요 기능과 하위 컴포넌트들에 대한 개요를 살펴해보았다. 이제부터는 이 하위 컴포넌트와 Web Container 를 어떻게 설정하는지에 대하여 살펴보자.

4.3 JEUS Web Container 설정

4.3.1 소개

Web Container 는 복잡한 구조를 가지고 있고 Container 가 시작되어 운영 환경으로 옮겨지기 전에 설정되어야 할 여러 기능들을 가지고 있다. 이 장에서는 한 Web Container 를 설정하기 위해 필요한 모든 것들을 설명한다.

참고: 모든 변경된 설정들을 새롭게 적용시키기 위하여 Web Container 를 재기동 시켜야 한다. 이는 설정이 Web Container 시작 시에 단 한 번만 읽히기 때문이다. 4.5 절 (JEUS Web Container 제어) 에는 JEUS Web Container 를 어떻게 시작, 종료 그리고 제어하는지에 대해 설명한다.

Web Container 를 설정하는 두 단계 과정은 다음과 같다.

1. 먼저 JEUSMain.xml(JEUS 시스템의 주 설정 파일)에 Web Container (Servlet engine) 설정을 추가한다.
2. 새로운 Web Container 홈 디렉토리에 WEBMain.xml 을 만들고 편집한다. (디렉토리의 이름은 “JEUS_HOME\config\<nodename>\<nodename>_servlet_<engine name>”와 같은 형식을 가진다.

위의 두 과정은 다음에 상세히 설명한다.

4.3.2 JEUSMain.xml 에 Web Container 추가

JEUSMain.xml 에 Web Container 를 추가하는 방법은 JEUS Server 안내서에 상세히 설명되어 있다. 여기서는 Servlet “engine”을 추가하기 위한 태그의 예를 보여준다.

<<JEUSMain.xml>>

```
<?xml version="1.0"?>
<jeus-system xmlns="http://www.tmaxsoft.com/xml/ns/jeus">
  ...
  <node>
    ...
    <engine-container>
      ...
      <engine-command>
```

```

                <type>servlet</type>
                <name>engine1</name>
            </engine-command>
            ...
        </engine-container>
        ...
    </node>
    ...
</jeus-system>

```

위의 예에서는 “engine1”이라는 Web Container(= Servlet engine)을 추가하는 것을 보여주고 있다. 다른 태그들에 대한 정보는 JEUS Server 안내서를 참고하라.

또한 위의 예에서는 필수 XML 파일 헤더가 생략되어 있다. 3장에서 반드시 포함해야 할 헤더 정보에 대해 설명하고 있으니 참고 하기 바란다.

<engine-command> 태그가 JEUS boot 과정 중 사용되기 위해서는 JEUS_HOME\config\<nodename>\<nodename>_servlet_<enginename> 디렉토리가 존재해야 한다. 여기에서 “nodename”은 JEUS 의 노드 명이고, “enginename”은 JEUSMain.xml 의 <engine-command>태그의 하위 태그인 <name>에 설정된 값이어야 한다. 위의 예제에서 “hostname”이 “johan”이라면 engine 이름은 “johan_servlet_engine1”이 된다.

JEUS_HOME\config\<nodename>\<nodename>_servlet_<enginename> 아래에는 반드시 WEBMain.xml 이라는 파일이 존재해야 한다. 이 파일은 다음 절에서 설명한다.

중요: 각 Engine Container 에는 단 한 개의 Web Container 가 존재할 수 있다.

4.3.3 WEBMain.xml 설정

JEUSMain.xml 에 추가된 JEUS Web Container 를 설정하기 위해서는 Container 의 홈 디렉토리 아래에 있는 WEBMain.xml 이 반드시 추가되어야 한다. XML 파일을 직접 편집하거나 GUI 환경에서 제공하는 기능을 사용할 수도 있다. (웹 관리자 또는 Web 기반의 Web Admin application 을 사용할 수 있다.)

물론, WEBMain.xml 을 직접 편집할 때는 텍스트 편집기나 XML 전용 편집기를 이용할 수도 있다.

WEBMain.xml 파일에는 Web Container 설정의 여러 부분을 대표하는 태그들이 존재한다. 각 태그들을 다음 하위 절에서 설명하겠다.

1. 자동 모니터링
2. stdout 와 stderr re-direction.
3. Context Group
4. Database connection pool.
5. Session 클러스터링.
6. 보안 스위치.

3, 4, 5 는 각기 다른 장에서 따로 자세히 설명하고 나머지는 여기에서 자세히 설명한다.

4.3.4 자동 모니터링 설정

위에서 설명했듯이, 자동 모니터링은 Web Container 의 자동 관찰 시스템이다. 여기에는 Container 의 리소스를 체크하고 문제 발생시에 적절한 대응을 하기 위한 네 가지의 모니터링 작업이 존재한다.

모니터링에 대하여 관리자가 해야 할 것은 상태를 파악하기 위한 시간 간격을 설정하는 것이다. 완전히 이 기능을 사용하지 않을 수도 있지만 권장하지는 않는다.

모니터링 thread가 사용하는 시간 주기를 설정하기 위해서는 WEBMain.xml 의 <monitoring> 태그와 그 하위 태그들을 직접 편집한다. (부록 D를 참조한다.)

XML 예제는 다음과 같다.

<<WEBMain.xml>>

```
<?xml version="1.0"?>
<web-container xmlns="http://www.tmaxsoft.com/xml/ns/jeus">
  <monitoring>
    <check-thread-pool>60000</check-thread-pool>
    <check-class-reload>60000</check-class-reload>
```

```

        <check-session>60000</check-session>
        <check-db-connection-pool>
            60000
        </check-db-connection-pool>
    </monitoring>
    . . .
</web-container>

```

문제가 생겼을 때 취하는 액션은 각 모니터링 thread 의 하위 컴포넌트에 해당하는 XML 설정 부분에서 따로 설정된다.

그래서 예를 들어, DB connection pool 에 생긴 문제에 대한 액션은 DB connection pool 설정에 지정된다. DB pool 모니터링 thread 는 이 특정한 문제가 발생했을 경우에만 미리 설정된 시간 간격과 대응 액션으로만 작동된다.

4.3.5 stdout 과 stderr re-direction 설정

WEBMain.xml에는 두 개의 태그들이 stdout과stderr redirect을 위하여 설정되고 JEUS_HOME\logs\JeusSystem\<node name>_<container name>\servlet 디렉토리에 로그를 남긴다. (4.2.5 절에 설명됨)

<redirect-stdout>과 <redirect-stderr> 태그가 여기에 해당된다. 설정 값으로는 Boolean 값인 “true”또는 “false”가 설정될 수 있다.

다음의 예는 로그 파일에 stdout 과 stderr 데이터가 남도록 하기 위한 설정이다.

<<WEBMain.xml>>

```

<?xml version="1.0"?>
<web-container xmlns="http://www.tmaxsoft.com/xml/ns/jeus">
    <monitoring>
        . . .
    </monitoring>
    <redirect-stdout>true</redirect-stdout>
    <redirect-stderr>true</redirect-stderr>
    . . .
</web-container>

```


4.3.6 Context Group 설정

전에 설명했듯이, context group 은 여러 개의 Web application(= context) 또는 가상 호스트를 그룹화 하기 위한 JEUS 만의 설정법이다. 각 Web Container 에는 하나 이상의 context group 이 존재할 수 있고, WEBMain.xml 의 최상위 태그인 <context-group>을 이용하여 설정될 수 있다.

예:

<<WEBMain.xml>>

```
<?xml version="1.0"?>
<web-container xmlns="http://www.tmaxsoft.com/xml/ns/jeus">
    <monitoring>
        ...
    </monitoring>
    ...
    <context-group>
        ...<!-- See chapter 6 -->
    </context-group>
    <context-group>
        ...
    </context-group>
    ...
</web-container>
```

주시해야 할 중요한 점은 Web Container 레벨에서 정의된 설정들은 모든 context group 과 Web Container 의 하위 컨텍스트에 글로벌하게 적용된다는 것이다. (예, Container 의 모든 context 는 Web Container 와 같은 레벨에 정의된 DB connection pool 을 사용할 수 있다.)

context group에 대한 내용은 상당히 크고 중심적인 것이기 때문에 이에 대한 심도 있는 설명을 위해 따로 한 개의 장을 할애한다. 6 장을 보길 바란다.

가상 호스트에 대한 설명은 10 장에 나와 있다.

4.3.7 Database Connection Pool 설정

앞에서 봤듯이, 여러 개의 database connection pool 이 추가될 수 있고 설정될 수 있다고 했다. 이 설정 정보는 WEBMain.xml 의 최상위 레벨 태그인 <db-connection-pool>에 설정된다.

<<WEBMain.xml>>

```
<?xml version="1.0"?>
<web-container xmlns="http://www.tmaxsoft.com/xml/ns/jeus">
    ...
    <db-connection-pool>
        ... <!-- See chapter 5 -->
    </db-connection-pool>
    ...
</web-container>
```

DB connection pool에 대한 주제는 상당히 크고, 중요한 것이기 때문에 5장에서 따로 설명하기로 한다.

4.3.8 Session 클러스터링 설정

JEUS Web Container는 크게 두 가지 방식의 Session 클러스터링 방식을 지원한다. 첫번째는 중앙 집중식 방식이고 두번째는 분산식 방식이다. 클러스터를 설정하는 법과 Session 정보를 관리하는 환경에 대한 보다 자세한 내용은 7장에서 설명한다.

여기서는 중앙 집중식 Session 클러스터링 설정이 Web Container 레벨에서도 가능하고 각 context group 에서도 가능하다는 것을 주시하기 바란다. 각 Session 설정이 context group 에 추가되었을 때는 Web Container 레벨에서 설정된 설정보다 우선한다.

주의: 중앙 집중식 Session 클러스터링은 context group 레벨에서 보다는 Web Container 레벨에서 하기를 권장한다.

중앙 집중식 Session 클러스터링 설정은 WEBMain.xml의 <session-cluster> 태그를 이용하며, (부록 D를 참조한다.) 분산식 Session 클러스터링 설정은 JEUSMain.xml의 <session-router-config> 태그를 사용한다.

중앙 집중식 예:

<<WEBMain.xml>>

```
<?xml version="1.0"?>
<web-container xmlns="http://www.tmaxsoft.com/xml/ns/jeus">
    ...
    <context-group>
        ...
    </context-group>
    ...
</web-container>
```

```

    <session-cluster>
        ...
        <!--Takes precedence over "outer" session config.-->
        <!--Use of this element here is discouraged-->
    </session-cluster>
    ...
</context-group>
...
<session-cluster>
    ... <!-- See chapter 8-->
</session-cluster>
...
</web-container>

```

분산식 예:

<<JEUSMain.xml>>

```

<?xml version="1.0"?>
<jeus-system xmlns="http://www.tmaxsoft.com/xml/ns/jeus">
    ...
    <node>
        ...
        <session-router-config>
            ...
        </session-router-config>
    . . .
    </node>
</jeus-system>

```

4.3.9 Logging 설정

앞서 언급하였듯이 Web Container system-log 설정은 JEUSMain.xml 의 system-logging 설정을 통해 이루어진다. 따라서, 이에 대한 자세한 사항은 JEUSServer 안내서 JEUS Logging 절을 참조하기 바란다. user-log 와 access-log 는 WEBMain.xml 에서 설정한다. user-log 와 access-log 는 system-log 와 설정하는 방법이 아주 유사하다. 따라서, 이 설정을 하기 전에 JEUSServer 안내서 JEUS Logging 절을 미리 숙지하여 JEUS system 전반적인 logging 구조를 이해할 것을 권장한다.

access-log, user-log Logging 설정은 내용이 많은 관계로 4.4 절에서 자세히 설명한다.

4.3.10 보안 스위치 설정

보안 스위치를 설정하는 방법은 간단하다.

WEBMain.xml을 편집하여 설정하기 위해서는 <security-switch> 태그를 부록 D에 나와 있는 WEBMain.xml과 같이 true 또는 false로 값을 설정한다.

보안 스위치를 설정하는 WEBMain.xml 의 예는 다음과 같다.

<<WEBMain.xml>>

```
<?xml version="1.0"?>
<web-container xmlns="http://www.tmaxsoft.com/xml/ns/jeus">
  . . .
  <security-switch>true</security-switch>
  . . .
</web-container>
```

4.3.11 Startup 과 Shutdown 클래스 설정

이 설정은 더 이상 WEBMain.xml 에서 지원하지 않는다. 동일한 기능을 JEUSMain.xml lifecycle-invocation 태그를 사용하여 얻을 수 있다. 자세한 내용은 JeusServer Guide 를 참조하기 바란다.

4.3.12 Shutdown Timeout 설정

<shutdown-timeout>태그는 Web Container 가 종료되기 전에 실행중인 worker thread 가 대기하는 시간을 천분의 일초로 지정하는 것이다.

예:

```
<?xml version="1.0"?>
<web-container xmlns="http://www.tmaxsoft.com/xml/ns/jeus">
  . . .
  <shutdown-timeout>10000</shutdown-timeout>
</web-container>
```

4.3.13 웹 관리자를 이용한 Web Container 설정

JEUS 웹 관리자 안내서의 엔진 컨테이너와 서블릿 엔진 장에서 JEUSMain.xml 에 Web Container 를 어떻게 추가 하는지와 WEBMain.xml 과 일을 어떻게 설정하는지 참고하도록 한다.

4.3.14 결론

여기까지 우리는 Web Container 의 주요 하위 컴포넌트들인 자동 모니터링, 보안 스위치 등을 어떻게 설정하는지에 대해 살펴보았다.

다음 장에서는 context group 과 database connection pool, Session 클러스터링 등 Web Container 설정에 대해서 더 자세히 설명하겠다.

그러나, 먼저 Web Container 가 어떻게 제어 되는지에 대해 알아본다.

4.4 JEUS Web Container Logging 설정

4.4.1 소개

Web Container 에서 별도로 설정 가능한 logger 는 access-log 와 user-log 이다. access-log 는 Web Container 에 요청된 request 및 그 처리 결과에 대한 log 이며 user-log 는 `javax.servlet.ServletContext.log(String msg)` 또는 `javax.servlet.ServletContext.log(String msg, Throwable t)` 등의 API 를 사용하여 Servlet application 내에서 생성되는 메시지를 기록하는 log 이다.

access-log 및 user-log 는 logging 설정의 하위 요소이며 context-group 단위로 설정이 가능하다. logging 설정은 아래 예제에서 보듯이 web-container 또는 context-group 의 하위 요소로 존재한다. web-container 설정의 하위 요소로 logging 을 설정하면 web-container 내의 모든 context-group 에 공통으로 적용된다. context-group 의 하위 요소로 설정된 logging 은 해당 context-group 에만 적용되며 web-container 의 logging 설정보다 우선한다.

<<WEBMain.xml>>

```
<?xml version="1.0"?>
<web-container xmlns="http://www.tmaxsoft.com/xml/ns/jeus">
  <context-group>
    ...
    <logging>
      <user-log>
        <level>FINE</level>
```

```
        <use-parent-handlers>
            true
        </use-parent-handlers>
        <handler>
            <smtp-handler>
                <name>smtpHandler</name>
                <level>SEVERE</level>
                <smtp-host-address>
                    mail.com
                </smtp-host-address>
                <from-address>
                    jeus@mail.com
                </from-address>
                <to-address>
                    admin@mail.com
                </to-address>
                <send-for-all-messages>
                    false
                </send-for-all-messages>
            </smtp-handler>
        </handler>
    </user-log>
    <access-log>
        <enable>true</enable>
        <format>
            [%{yyyy.MM.dd HH:mm:ss}t] %a
        </format>
    </access-log>
</log>
```

```

        %m %U%q" %s %Dms
    </format>
    <handler>
        <file-handler>
            <name>fileHandler</name>
            <valid-hour>1</valid-hour>
        </file-handler>
    </handler>
</access-log>
</logging>
</context-group>
...
</web-container>

```

4.4.2 기본 logger file 위치

별도의 설정이 없을 경우 Web Container 당 1 개의 access-log 및 user-log file 이 생성된다. 그 기본 위치는 다음과 같다.

- **JEUS log home (JEUS_LOGHOME):** \$JEUS_HOME/logs/JeusSystem 디렉토리가 기본이다.
- **access-log 기본 위치 :** <JEUS_LOGHOME> 밑의 <node_name>_<container_name>/servlet/accesslog/access.log 가 기본 log file 이다.
- **access-log 기본 위치 예 :** \$JEUS_HOME 이 "c:\jeus5"이고 node_name 이 johan, container_name 이 container1 이라면, c:\jeus5\johan_container1\servlet\accesslog\access.log 가 기본 로그 파일이다.
- **user-log 기본 위치 :** <JEUS_LOGHOME> 밑의 <node_name>_<container_name>/servlet/userlog/user.log 가 기본 log file 이다.
- **user-log 기본 위치 예 :** \$JEUS_HOME 이 "c:\jeus5"이고 node_name 이 johan, container_name 이 container1 이라면, c:\jeus5\johan_container1\servlet\userlog\user.log 가 기본 로그 파일이다.

4.4.3 공통 설정 항목

access-log, user-log 설정에 공통적으로 사용할 수 있는 하위 요소를 소개하면 다음과 같다.

- **<level>** : logger 의 level 을 설정한다. 이 level 이하의 message 만 logger 를 통해 출력될 수 있다. 이 level 의 값으로는 logging API 의 level 인 SEVERE, WARNING, INFO, CONFIG, FINE, FINER, FINEST 가 올수 있다. 기본 설정은 INFO 이다.
- **<use-parent-handlers>** : logger 가 자신의 handler 이외에 상위 logger 의 handler 를 사용할 것인지의 여부를 지정한다. 기본값은 false 이다.
- **<filter-class>** : logger 가 log message 를 handler 에게 보내기 전에 수행 하는 filtering 에 사용할 class 를 지정한다. 여기에 지정된 class 는 lib/application 디렉토리의 jar 파일 내에 포함되어야 한다.
- **<handler>** : logger 가 사용할 handler 를 지정한다. 이 항목이 설정되어 있지 않다면 console handler 가 기본적으로 사용된다.

<handler> 에는 다음과 같은 하위 항목들이 있다.

- **<console-handler>** : 화면으로 log message 를 출력하는 handler 이다. 이 handler 는 다음과 같은 기본적인 handler 설정만 가지고 있다.
 - **name** : 이 handler 가 tool 에서 보여질 때 사용할 이름을 지정한다. 만약 지정되어 있지 않으면 class name 과 hash code 로 이름이 대체된다.
 - **level** : 이 handler 가 출력할 message 의 level 을 지정한다. 즉, logger 를 통과한 log message 가 이 logger 가 사용하는 각각의 handler 에게 전달되는데 이 handler 의 level 에 부합하는 log message 만 이 handler 에 의해 출력된다. 기본 값은 FINEST 로 logger 를 통과하는 모든 log message 가 handler 에 의해 출력되도록 되어 있다.
 - **encoding** : 이 handler 가 출력하는 문자열의 encoding 을 지정한다. 기본은 system encoding 으로 설정되어 있다.
 - **filter-class** : 이 handler 가 log message 를 출력하기 전에 수행할 filtering 에 이용되는 class 이다. Logger 의 filter-class 와 마찬가지로

지로 lib/application 에 이 클래스를 포함한 jar 파일이 존재해야 한다.

- **<file-handler>** : 파일로 log message 를 출력하는 handler 이다. 이 handler 는 <console-handler>의 설정 이외에 다음과 같은 설정을 가지고 있다.
 - **file-name** : 이 handler 가 출력할 file 의 이름을 지정한다. 절대 경로로 되어 있다면 그 경로로 file 이 생기고 상대경로라면 각 logger 의 기본 경로를 기준으로 한 상대경로로 인식한다. 이 설정을 하지 않으면 각 logger 별로 지정된 path 로 file 을 생성해서 log message 를 출력한다.
 - **valid-day 와 valid-hour** : 이 handler 가 출력할 file 을 시간마다 따로 생성할 경우에 사용한다. 둘 중 하나만 사용할 수 있는데, valid-day 는 날짜별로, valid-hour 는 시간별로 file 을 변경한다. Valid-hour 는 24 의 약수이거나 (ex. 3, 6) 24 로 나눈 나머지가 약수(ex. 27, 30)의 이름을 지정한다. File 이름의 형식은 valid-day 의 경우 file 끝에 _YYYYMMDD 가 붙거나 valid-hour 의 경우 _YYYYMMDD_HH 가 붙는다. 이때 HH 는 file 로그의 시작 시간이다.
 - **buffer-size** : file 로 출력할 때 사용할 buffer 의 크기를 지정한다. buffer 가 클수록 logging 의 성능은 좋아지지만 예상치 못한 상황으로 JEUS 가 종료될 때에는 그 buffer 크기만큼 log 가 손실된다. 기본값은 20KB 이다.
 - **append** : file 로 출력할 때 이미 file 이 존재하면 덮어쓸지 file 끝에 추가할지를 결정한다. 기본값은 true 이다.
- **<smtp-handler>** : log message 를 email 로 전송하는 handler 이다. 하나의 log message 가 하나의 email 로 전송된다. 이 handler 는 <console-handler>의 설정 이외에 다음과 같은 설정을 가지고 있다.
 - **smtp-host-address** : email 을 보낼 host 의 주소를 지정한다.
 - **from-address** : email 을 보내는 사람의 주소를 지정한다.
 - **to-address** : email 을 받는 사람의 주소를 지정한다.
 - **cc-address** : email 을 참조하는 사람의 주소를 지정한다.

- **bcc-address** : email 을 숨은 참조하는 사람의 주소를 지정한다.
- **send-for-all-messages** : 모든 message 를 smtp-handler 로 보낼지 를 결정한다. 만약 false 이면 JEUS system 에서 email 로 전송하 기로 결정되어 있는 message 만 이 handler 를 사용해서 보내진 다. 현재 이 설정은 jeus.systemuser logger 에만 유효하다.
- **<socket-handler>** : log message 를 socket 으로 전송하는 handler 이다. 이 handler 는 <console-handler>의 설정 이외에 다음과 같은 설정을 가 지고 있다.
 - **address** : 이 handler 가 접속할 machine 의 IP 주소를 지정한다.
 - **port** : 이 handler 가 접속할 machine 의 port 를 지정한다.
- **<user-handler>** : user 가 만든 handler class 를 지정하는 항목이다. 이 handler 는 <console-handler>의 설정 이외에 다음과 같은 설정을 가지 고 있다.
 - **handler-class** : user 가 만든 handler 의 class 를 지정한다. 이 cla ss 는 lib/application 디렉토리의 jar 파일에 포함되어 있어야 한 다. 또한 이 class 는 logging API 의 java.util.logging.Handler 를 상속받고 jeus.util.logging.JeusHandler 를 구현해야 한다.
 - **handler-property** : jeus.util.logging.JeusHandler 의 setProperty() 에 사용되는 Map 객체에 들어갈 property 를 지정한다.
 - **formatter-class** : 이 handler 가 사용할 formatter class 를 지정한 다. 이 class 도 lib/application 디렉토리의 jar 파일에 포함되어 있어야 한다. 또한 jeus.util.logging.JeusFormatter interface 를 구 현해야 한다. 기본값은 JEUS 에서 사용하는 jeus.util.logging.Si mpleFormatter 이다.
 - **formatter-property** : jeus.util.logging.JeusFormatter 의 setProper ty()에 사용되는 Map 객체에 들어갈 property 를 지정한다.

주의: access-log 설정에서 <level>설정은 무의미하다.

4.4.4 access-log 관련 설정 항목

access-log 에만 사용되는 설정으로는 다음과 같은 것이 있다.

- **<enable>** : access-log 관련하여 아무것도 설정하지 않을 경우 기본 파일에 기본 형식으로 access log 를 남긴다. 이 설정은 access-log 를 남기는 것을 원치 않을 경우 사용한다.
- **<format>** : access-log 에 남길 로그의 형식을 지정한다. 설정하지 않을 경우 기본 형식으로 로그를 남긴다. 자세한 설명은 다음 절에서 따로 하겠다.

4.4.5 access-log <format> 설정

access-log 의 하위 요소인 <format>을 사용하여 사용자 정의 access-log format 을 지정하는 것이 가능하다. <format>에 넣을 값은 임의의 스트링 값이 가능하다. 단 “%” 기호는 특수 기호로 인식된다. 즉, “%”를 접두사로 사용하는 아래에 열거하는 단어들은 runtime 에 여러 가지 request/response property 로 대체되어 access-log 에 남게 된다.

- **Runtime 에 대체되는 특수 단어들**
 - **%a** : Remote IP address
 - **%A** : Local IP address
 - **%b** : HTTP header 를 제외한 response body 의 총 길이. ‘-’는 0 을 나타낸다.
 - **%B** : HTTP header 를 제외한 response body 의 총 길이
 - **%h** : Remote host name 또는 IP address
 - **%H** : Request protocol
 - **%m** : Request method (GET, POST...)
 - **%p** : Local port number
 - **%q** : Query string(앞에 ‘?’가 붙음)
 - **%r** : method 와 request URI
 - **%s** : HTTP response status code
 - **%S** : User session ID

- **%t** : Date and time(기본 시간 형식으로. 후술함)
- **%u** : Remote user name
- **%U** : Request URL
- **%v** : Local server name
- **%D** : processing time (milli 초 단위)
- **%T** : processing time (초 단위)
- **Runtime** 에 대체되는 특수 단어들 2 : request cookie, header, attribute, session attribute 등 에서 특정 값을 가져와 표시할 수 있다.
 - {xxx}I : request header 에서 key 가 “xxx”인 값을 표시
 - {xxx}c : request cookie 에서 key 가 “xxx”인 값을 표시
 - {xxx}r : request attribute 에서 key 가 “xxx”인 값을 표시
 - {xxx}s : 세션 정보에서 key 가 “xxx”인 값을 표시
 - {xxx}t : “xxx”를 JDK standard DateFormat 으로 기술하면 access-log 의 시간 형식을 바꿀 수 있다.
- 기본 <format>값은 다음과 같다.
 - [%{yyyy.MM.dd HH:mm:ss}t] %a "%m %U%q" %s %Dms

4.4.6 user-log 관련 설정 항목

user-log 에만 해당하는 하위 요소는 따로 없다. user-log 는 다음에 따로 설명할 context deployment descriptor 에서 설정하여 특정 context 단독으로 사용하는 것이 가능하다. 그러한 경우에는 context deployment descriptor 안에 있는 user-log 설정이 우선 순위를 갖는다.

4.5 JEUS Web Container 제어

4.5.1 소개

Web Container 를 제어 한다는 것은 Web Container 의 시작과 종료를 제어한다는 이야기와 같다. 이 두 가지 작업은 웹 관리자를 통하여 또는 jeusadmin 콘솔 툴을 사용하여 가능하다.

이 장의 내용은 JEUS Server 안내서에 있는 정보와 중복된 것이지만 완벽을 기하기 위하여 여기서 다시 반복한다.

이 절에서는 Web Container 에 대한 정보만 포함하며 context group 과 context(Web application)에 대해서는 언급하지 않는다.

4.5.2 jeusadmin 콘솔 툴을 이용한 Web Container 제어

여기서는 jeusadmin 콘솔 툴을 이용한 Web Container 제어의 간단한 예를 들어 보이겠다. 아래의 예는 “johan_servlet_engine1”라는 Web Container 가 적절히 JEUSMain.xml 과 WEBMain.xml 에 설정되어 있다고 가정한다.

먼저, jeusadmin 을 시작하고 JEUS 노드에 연결시킨다.

```
C: \> jeusadmin johan
```

사용자와 암호를 입력하면 jeusadmin 명령창이 나타난다.

JEUS 노드가 시작되어 있지 않으면 다음과 같이 한다.

```
johan> boot
```

설정된 Web Container 와 모든 다른 engine 들이 자동으로 시작된다.

“johan_servlet_engine1”이라는 Web Container 가 시작되었다고 가정하고 다음 명령으로 종료한다.

```
johan> downeng johan_servlet_engine1
```

Web Container 가 종료되었다.

종료된 Web Container 를 시작하려면

```
j ohan> starteng j ohan_servl et_engi ne1
```

Web Container 가 다시 시작되었다.

위의 예에서는 Web Container 에 관련된 명령을 몇 가지 알아보았다. JEUS Server 안내서를 통하여 모든 정보를 참고하기 바란다.

4.5.3 웹 관리자를 이용한 Web Container 제어 툴

JEUS 웹 관리자 안내서의 서블릿 엔진 제어 장에 설명을 참조하도록 한다.

4.5.4 결론

여기까지 우리는 터미널 윈도우에서 또는 웹 관리자를 이용하여 Web Container 를 제어하는 방법을 알아보았다.

다음은 어떻게 수행되고 있는 Web Container 를 모니터링 하는지에 대해 알아보자

4.6 JEUS Web Container 모니터링

4.6.1 소개

모니터링은 근본적으로 특정 Web Container 의 수행 시 데이터와 상황 정보를 수집하는 것을 의미한다.

주의: Web Container 를 모니터링 하기 위해서는 콘솔 툴 보다는 보다 상세하고 완전한 engine 상태 정보를 제공하는 웹 관리자를 사용하도록 권고한다. 하지만 여기에서는 두 가지 방법이 모두 설명될 것이다.

4.6.2 콘솔을 통한 Web Container 모니터링

‘jeusadmin’과 ‘webadmin’ 콘솔에서는 Web Container 에 대한 기초 정보를 얻을 수 있는 기능을 제공한다.

JEUS Server 안내서와 부록 A 를 통하여 이 툴에 대하여 알아보라.

4.6.3 웹 관리자를 통한 Web Container 모니터링

웹 관리자를 통하여 Web Container 를 모니터링 하는 방법에 대한 자세한 내용은 JEUS 웹 관리자 안내서의 엔진 컨테이너 통계 장과 세션 추적 장의 세션 서버 통계 내용을 참조한다.

4.6.4 결론

이 절에서는 Web Container engine 모니터링에 관련된 기본적인 것에 대하여 살펴보았다.

특히, 이것에 관련하여 웹 관리자가 권장되는 툴이라는 것도 설명하였다.

다음 장에서는 Web Container 의 튜닝 작업에 대해 알아보겠다.

4.7 JEUS Web 성능 향상을 위한 튜닝

WEBMain.xml 의 최상위 레벨에서 최적 성능을 위해 할 수 있는 것이 두 가지가 있다.

- 모니터 thread 의 체크 시간 간격을 길게 하여 덜 안정적인 운영을 유도한다. 이 thread 를 사용하지 않도록 설정하면 가장 좋은 성능을 낼 것이다. (권장하지 않는 방법이다.)
- Web application 이 인증과 권한부여 메커니즘을 사용하지 않거나 다른 제품에서 보안관련 서비스를 제공하면 보안 스위치를 끈다. 성능을 약간 높일 수 있다.

다음에 나오는 장에서는 JEUS Web 모듈 튜닝에 대한 많은 설명을 할 것이다. 위에서 설명한 팁은 WEBMain.xml 의 최상위에서만 적용되는 것이다.

4.8 결론

이것으로써 JEUS Web 모듈인 JEUS Web Container 에 대한 최상위 레벨 설명을 마친다.

Web Container 는 JEUS 노드 내에서 실행된다는 것을 보았고, WEBMain.xml 을 통하여 설정되며 많은 수의 하위 컴포넌트들을 가진다는 것을 알았다.

어떻게 Web Container 를 설정, 제어, 모니터링, 튜닝하는지에 대하여도 설명하였다.

이 문서의 나머지 부분에서는 Web Container 의 하위 컴포넌트들에 대해 설명한다. 바로 다음에서는 Web Container 가 주관하는 database connection pool 에 대한 설명을 하겠다.

5 Database Connection Pool

5.1 소개

전에 소개했듯이, DB connection pool 은 이미 초기화된 database connection 들의 pool 을 관리하여 더 효과적으로 database connection 을 사용하도록 하는 것이다. 이 연결들은 요청이 있을 때 클라이언트들이 각자 연결을 맺는 작업을 하지 않고 클라이언트(EJB, Servlet, application 클라이언트 등)에 바로 전달된다. 클라이언트가 database connection 의 사용을 마쳤다면 다른 클라이언트가 사용할 수 있도록 pool 에 반환된다. (여기서 클라이언트는 데이터베이스 연결을 필요로 하는 모든 종류의 프로그램을 말한다.)

Web Container 에는 여러 개의 database connection pool 이 추가되고 설정될 수 있다. 이 설정 정보는 WEBMain.xml 의 최상위 태그 중 하나인 <db-connection-pool> 태그에 설정된다. 그러므로, DB connection pool 은 같은 Container 에 설정된 context group 과 context 에서 모두 접근 가능한 Container 의 범용 기능이라 할 수 있다. 각 DB connection pool 은 하나의 물리적 데이터베이스와의 연결을 관리한다.

DB connection pool 을 사용할 수 있는 다른 방법은 JEUS Server 에서 JNDI lookup 을 통하여 제공받는 방법이다. 그러나, 이 문서에서는 Web Container 가 제공하는 connection pool 에 대해서만 고려한다. JEUS Server 에서 제공하는 DB connection pool 은 “JEUS Server 안내서”를 참고한다.

경고: Web Container 의 DB connection pool 을 사용하는 것이 이식성 있는 방법은 아니다. 이식성이 고려되어야 하는 Servlet 을 작성할 때에는 JNDI 기반의 방법을 사용한다.

먼저 Web Container 에서 실행할 수 있는 database connection pool 을 설정하는 법을 알아보고, Servlet 에서 어떻게 pool 을 접근하고 사용하는지에 대해서도 알아보자.

5.2 Database Connection Pool 의 개요

5.2.1 소개

database connection pool 은 Servlet 에서 두 가지 방법으로 접근 가능하다. JEUS Server 의 DB connection pool 을 이용할 수 있고 Web Container 에서 설정된 DB connection pool 을 이용할 수도 있다.

Web Container 의 DB connection pool 을 이용할 때에는 두 가지 주된 설정 방법이 있다. 그 설정 방법은 “shared” 와 “non-shared”이다. Pool 제어와 DBA 접근에 관련된 몇 가지의 기능들이 더 존재한다.

앞으로 이 컴포넌트들과 기능들에 대하여 좀 더 살펴보겠다.

5.2.2 JEUS Server 의 DB connection pool

JEUS Server 의 DB connection pool 을 사용하기 위해서는 JNDI lookup 을 먼저 해야 한다. JEUS Server 안내서에 이 방법에 대해 설명되어 있다.

5.2.3 Web Container 의 DB Connection Pool

JEUS Server 에서 제공하는 DB connection pool 을 사용하는 대신에 사용할 수 있는 방법은 Web Container 가 자체적으로 DB connection pool 을 가지도록 하는 것이다. 이 방식을 사용하면 pool 을 JNDI 의 lookup 을 통해서가 아니라, DriverManager 클래스를 통해서 직접적으로 가져온다.

즉, Container 기반의 lookup 은 JNDI 를 통해서가 아니라 Driver Manager 클래스를 통한 메커니즘을 통해 이루어진다.

이 문서에서는 DB connection pool 이 Web Container 에서 운영되고 있을 경우만을 고려하고 Server 기반의 DB pool 은 설명하지 않는다.

독자들에게 좀 더 명료한 개념을 주기 위해 구조적 개요를 보여주는 그림을 다음에 소개 한다[그림 7]. 원으로 강조한 부분(DB connection pool)은 여기서 소개하고 다른 부분은 이 문서의 다른 장에서 소개한다.

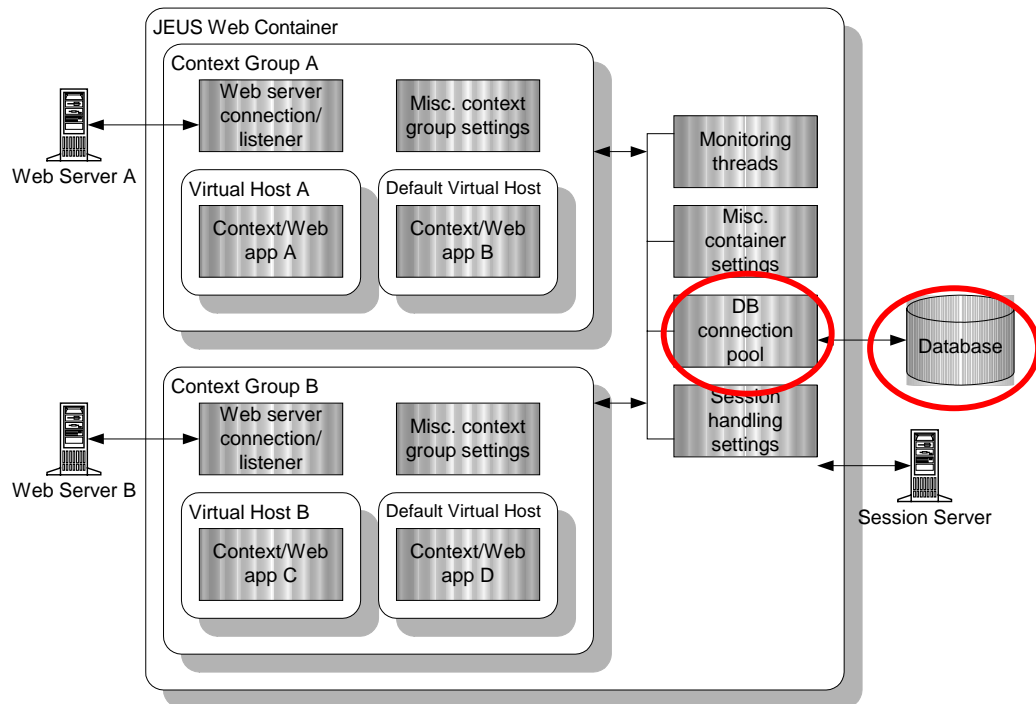


그림 7 DB connection pool 이 포함된 JEUS Web Container 의 구조.

5.2.4 Shared Connection Pool 모드

Web Container 기반 DB connection pool 의 shared 모드는 pool 내의 모든 연결은 worker thread 들(즉, 모든 Servlet)에 의해 공유된다는 의미이다.

가용한 connection 은 쓰레드에 의해 요청이 있을 때 pool 에서 꺼내지고 사용이 끝나면 다른 클라이언트의 사용을 위하여 pool 로 반환된다. 이 모드는 기본 모드이고 일반적인 pool 과 같이 작동한다.

여기에는 특수한 “shared:non-jeus” 하위 모드가 존재한다. 이 모드는 이전 버전에서부터의 마이그레이션과 같은 목적으로만 사용된다.

Shared 모드가 설정되어 있으면 “DB pool control”이라는 추가 설정이 필요하다. 이 정보는 pool 의 크기를 결정한다. 이 기능은 non-shared 모드에서는 사용되지 않는다.

5.2.5 Non-shared Connection Pool 모드

DB connection pool 의 non-shared 모드는 DB connection 을 요청하는 모든 클라이언트가 전용 connection 을 할당 받아야 하는 경우에만 사용된다. 달리 말해서 각 worker thread 는 고유의 connection 을 가진다는 의미이다.

이 경우 각 쓰레드는 connection 을 얻기 위해 경쟁할 필요가 없기 때문에 메모리 자원만 충분하다면 DB connection pool 의 성능 향상을 꾀할 수 있다.

5.2.6 DBA Override 기능

Connection 과 연관된 심각한 문제가 생겼을 경우(일반적인 방법으로 connection 이 끊기지 않는 경우)에 Web Container 는 데이터베이스로부터 특수한 connection 얻어서 이 문제를 해결한다. 이 connection 은 관리자 레벨의 권한으로 모든 접근 권한을 갖는다. 이 모드의 작업을 “DBA override”라고 한다.

예를 들어, 일반적인 “close()” method 로 끊기지 않는 connection 이 있으면 Web Container 는 데이터베이스로부터 특수한 DBA connection 얻기를 시도하고 시스템을 정리하기 위해 낮은 레벨의 “kill”과 같은 명령을 수행한다.

5.2.7 결론

Servlet 에서 두 가지 방법으로 DB connection pool 을 사용하는 방법을 보았다. 각각 JEUS Server 기반과 Container 기반의 연결이었다. 그리고 Container 기반의 pool 이 두 가지 모드를 가진다는 것도 알았다. 마지막으로 “DBA override”라는 기능도 알아보았다.

지금부터 Web Container 기반의 DB connection pool 에서 shared 와 non-shared 모드가 어떻게 설정되는 지를 살펴 보겠다.

5.3 Web Container Database Connection Pool 설정

5.3.1 소개

이 절에서는 어떻게 Web Container DB connection pool 의 설정과 기능에 접근 및 변경하는 지에 대하여 설명한다.

5.3.2 기본 DB Pool 설정

각 Web Container DB connection pool 은 WEBMain.xml 의 <db-connection-pool> 태그로 설정 된다. 그러므로, 각 DB connection pool 은 동일한 Container 내의 모든 context 에서 사용 가능하게 된다.

Shared 와 non-shared 모드에 모두 적용 가능한 기본적인 설정 태그들이 다음 리스트에 설명되어 있다.

- **pool name** 은 소스 코드 내에서 DB connection pool 을 식별하기 위해 사용한다.
- **pool type** 은 shared 또는 non-shared pool 모드를 설정한다.
- **pooling rule** 은 pool 에서 connection 을 얻기 위한 알고리즘을 설정한다. (stack 기반 또는 queue 기반이 적용 가능하다).
- **connection URL** 과 **connection arguments** 은 이 DB connection pool 이 데이터베이스에 연결될 때 필요하다.
- **DB connection pool driver** 는 데이터베이스 제공자로부터 제공되는 자바 클래스 풀링 데이터베이스 드라이버이다.
- **timeout settings** 은 오랜 시간 동안 connection 을 사용하고 있을 경우 정해진 시간이 지나면 pool 로 connection 을 반환하도록 하는 설정이다. (“close long active connection” 과 “max active time”) 또한 대체되기 전까지 사용될 수 있는 횟수를 설정(“max use count”)하고 “get connection” 요청이 모든 connection 이 사용되고 있을 상황(“get connection timeout”)에서 도착한 후 connection 이 가용해질 때까지 기다려야 하는 시간에 대한 설정을 한다.
- **Delay settings** 은 새로운 DB connection 이 설정되기까지 기다리는 시간인 login delay 과 DB pool 이 connection 을 닫을 때 사용하는 “closing-delay”를 설정한다.
- **fetch size** 는 DB connection 을 통하여 매번 ResultSet 을 가져올 때마다 가져오는 레코드의 숫자이다.
- **encryption** 태그는 connection 인자로 주어진 password 가 base64 암호화가 되어 있는지 plain-text 포맷인지 결정한다.
- **check query** 태그는 Web Container 가 DB connection 이 살아있는지 확인하기 위한 SQL 질의를 지정하는 태그이다. 기본적으로, 내부적으로 지정된 SQL 질의문이 사용되지만 특정 DBMS 에서는 이 내부 질의문이 작동하지 않을 수도 있다. 이 경우에는 이 태그에 부하가 적은 질의문을 지정해 놓는다. 이 태그는 필요한 경우에만 사용하도록 한다.

- **read timeout** 태그는 DB 에서 읽기 작업을 완료하기 위해 대기하는 최대 시간을 설정한다.

이 설정에 대한 정보는 부록 D 에 자세히 설명되어 있다.

WEBMain.xml 에 직접 DB connection pool 들을 설정하기 위해서는 파일을 열고 각각의 database 에 대한 설정을 <db-connection-pool> 태그에 해준다.

DB connection pool 설정을 위한 간단한 WEBMain.xml 의 예제가 다음에 나와있다.

<<WEBMain.xml>>

```
<?xml version="1.0"?>
<web-container xmlns="http://www.tmaxsoft.com/xml/ns/jeus">
    ...
    <db-connection-pool>
        <pool-name>oraclePool</pool-name>
        <pool-type>non-shared</pool-type>
        <pooling-rule>stack</pooling-rule>
        <connection-url>
            jdbc:oracle:thin:@111.111.111.111:1521:ORA805
        </connection-url>
        <driver-class-name>
            oracle.jdbc.driver.OracleDriver
        </driver-class-name>
        <connection-argument>
            user=scott;password=tiger
        </connection-argument>
        <!-- For non-shared mode, listener-id elements go here.
            See next sub-section.
        -->
        <!-- For shared mode, dynamic-increment goes here.
            See later sub-section.
        -->
        <get-connection-timeout>30000</get-connection-timeout>
        <close-long-active-connection>
            true
        </close-long-active-connection>
        <max-active-time>300000</max-active-time>
        <max-use-count>30</max-use-count>
```

```

<login-delay>100</login-delay>
<close-delay>10000</close-delay>
<fetch-size>100</fetch-size>
    <!-- Settings for shared mode and DBA control goes
         here. See below.
    -->
    <encryption>false</encryption>
    <check-query>select date from dual</check-query>
</db-connection-pool>
...
</web-container>

```

다음의 두 절에서는 어떻게 shared 와 non-shared 운영 모드로 pool 을 설정하는지에 대해 설명한다.

5.3.3 Non-shared 모드 설정

위에서 설명한 것과 같이 non-shared 모드는 모든 클라이언트에게 전용 DB connection (pool)을 지정하여 공유를 금지하고 다른 클라이언트와의 경쟁을 없앤다.

Pool type 을 “non-shared”(또는 “non-shared:non-jues”)로 설정하여 non-shared 모드를 선택하였을 때 다음과 같은 항목들이 위에서 설명한 항목 외에 추가로 설정되어야 한다.

- **listener ID**는 <webserver-connection><xyz-listener><listener-id>에 지정된 Web Server의 listener ID와 같아야 한다. 이 태그들은 여러 번 사용될 수 있다. ID들은 access control list처럼 worker thread pool(Servlet을 수행하는 thread pool)들을 걸러내고, 전용의 non-shared connection을 얻는다. 7 장에 설명되어 있는 Web listener ID들에 대해 살펴보기 바란다.

Non-shared pool 설정을 위한 WEBMain.xml 의 예가 다음에 나와 있다. (일반 글씨체는 전 절에서 설명한 것들이고 굵은 글씨는 non-shared 에 관련된 것이다.)

<<WEBMain.xml>>

```
<?xml version="1.0"?>
```

```
<web-container xmlns="http://www.tmaxsoft.com/xml/ns/jeus">
  ...
  <db-connection-pool>
    <pool-name>oraclePool</pool-name>
    <pool-type>non-shared</pool-type>
    <pooling-rule>stack</pooling-rule>
    <connection-url>
      jdbc:oracle:thin:@111.111.111.111:1521:ORA805
    </connection-url>
    <driver-class-name>
      oracle.jdbc.driver.OracleDriver
    </driver-class-name>
    <connection-argument>
      user=scott;password=tiger
    </connection-argument>
    <listener-id>WebListener1</listener-id>
    <listener-id>WebListener2</listener-id>
    <get-connection-timeout>30000</get-connection-timeout>
    <close-long-active-connection>
      true
    </close-long-active-connection>
    <max-active-time>300000</max-active-time>
    <max-use-count>30</max-use-count>
    <login-delay>100</login-delay>
    <close-delay>10000</close-delay>
    <fetch-size>100</fetch-size>
    <!-- Settings for DBA control goes
      here. See below.
    -->
    <encryption>false</encryption>
  </db-connection-pool>
  ...
</web-container>
```

5.3.4 Shared 모드 설정

위에서 언급했듯이, shared 모드는 모든 클라이언트 쓰레드들이 connection 들을 공유한다. (즉, connection 을 얻기 위해 경쟁한다는 이야기다.)

이 모드가 사용될 때에는 DB pool 제어 태그인 <db-pool-control>에 다음과 같은 설정들이 포함되어 있어야 한다.

- **Pool size settings:** DB connection pool의 최소, 최대수 지정 및 pool의 크기 증가값 지정.
- **Maximum alive 와 idle times:** 이 두 시간 설정 값이 초과 하면 connection이 제거되고 새로운 것으로 대체된다. “Alive”는 초기 connection 생성 시각에서부터 시간이고 “idle”은 connection이 사용되지 않을 때부터의 시간이 된다.

Shared 모드에서는 <db-connection-pool> 태그 아래에 <dynamic-increase> 태그를 선택적으로 설정할 수 있다.

- **dynamic increment** 설정이 “true”로 지정되어 있으면 모든 connection이 사용 중이고 추가의 클라이언트 요청이 들어오면 제거 가능한 DB connection을 더 생성할 수 있게 된다. 제거 가능한 connection은 클라이언트가 사용을 마쳤을 때에는 pool로 반환하지 않는다. 기본 값인 “false”인 경우에는 이와 같은 추가의 connection을 생성하지 않고 클라이언트는 connection이 제공 될 때 까지 기다려야 한다.

이 설정에 대한 자세한 정보는 부록 D에 나와 있다.

다음은 그 예이다.

<<WEBMain.xml>>

```
<?xml version="1.0"?>
<web-container xmlns="http://www.tmaxsoft.com/xml/ns/jeus">
    . . .
    <db-connection-pool>
        <pool-name>oraclePool</pool-name>
        <pool-type>shared</pool-type>
        <pooling-rule>stack</pooling-rule>
        <connection-url>
            jdbc:oracle:thin:@111.111.111.111:1521:ORA805
        </connection-url>
        <driver-class-name>
            oracle.jdbc.driver.OracleDriver
        </driver-class-name>
        <connection-argument>
```

```

        user=scott;password=tiger
    </connection-argument>
    <dynamic-increment>true</dynamic-increment>
    <get-connection-timeout>30000</get-connection-timeout>
    <close-long-active-connection>
        true
    </close-long-active-connection>
    <max-active-time>300000</max-active-time>
    <max-use-count>30</max-use-count>
    <login-delay>100</login-delay>
    <close-delay>10000</close-delay>
    <fetch-size>100</fetch-size>
    <db-pool-control>
        <min>10</min>
        <max>100</max>
        <step>10</step>
        <max-idle-time>30000</max-idle-time>
        <max-alive-time>60000</max-alive-time>
    </db-pool-control>
    <encryption>false</encryption>
    </db-connection-pool>
    ...
</web-container>

```

5.3.5 DBA Connection 설정

문제 있는 DB connection 을 강제로 닫기 위한 방법인 DBA connection 은 반드시 “close long active connection” 설정이 된 상황에서 설정해야 한다. (WEBMain.xml 의 <db-connection-pool> 태그의 하위 태그인 <close-long-active-connection>에서 설정)

DBA connection은 database pool 설정의 하위 태그인 <dba-connection>에 한다. 부록 D 를 살펴보자.

이 태그는 다음과 같은 하위 설정 태그들을 가진다.

- **connection URL** 은 database 에 연결하기 위해 사용된다.
- **connection arguments** 은 database 에 DBA 권한을 주기 위해 제공한다.

- **force closing** Boolean 옵션은 문제 있는 connection 을 “kill” 명령어로 강제로 닫을 것인지에 대한 설정이다.

다음은 그 예이다.

```
<<WEBMain.xml>>

<?xml version="1.0"?>
<web-container xmlns="http://www.tmaxsoft.com/xml/ns/jeus">
    ...
    <db-connection-pool>
        ...
        <close-long-active-connection>
            true
        </close-long-active-connection>
        ...
        <dba-connection>
            <forced-close>true</forced-close>
            <connection-url>
                jdbc:oracle:thin@111.111.111.111:1521:ORA805
            </connection-url>
            <connection-argument>
                user=system;password=manager
            </connection-argument>
        </dba-connection>
    </db-connection-pool>
    ...
</web-container>
```

5.3.6 웹 관리자를 이용한 설정

JEUS 웹 관리자 안내서를 참고하도록 한다.

5.3.7 결론

지금까지 기본적인 데이터베이스 설정인 shared, non-shared 모드, DBA 특정 연결에 대해 살펴보았다.

제어와 모니터링 JDBC connection pool 의 튜닝에 대한 더 자세한 정보는 JEUS Server 안내서를 참고 한다.

다음에는 Servlet 코드에서 사용되는 DB pool 의 사용 예를 설명하겠다.

5.4 Web Container Database Connection Pool 사용

다음에는 Web Container 기반의 DB connection pool 을 사용한 JDBC connection 을 어떻게 Servlet 에서 사용할 수 있는지 보여주는 예를 설명한다.

이 예에서는 WEBMain.xml 에 pool name 으로 “oraclePool”이 설정되어 있다.

1.

```
Class.forName("jeus.jdbc.pool.Driver");
Connection con =
    DriverManager.getConnection ("jdbc:jeus:pool:oraclePool",
    null);
```

2.

```
Properties prop = new Properties();
prop.put("jeus.jdbc.pool.ConnectionPoolID", "oraclePool");
Connection con = DriverManager.getConnection ("jdbc:jeus:pool",
prop);
```

3.

```
Driver myDriver
    = (Driver)Class.forName("jeus.jdbc.pool.Driver")
        .newInstance();
Connection conn
    = myDriver.connect("jdbc:jeus:pool:oraclePool", null);
```

중요: 위의 세 가지 방법 중 세 번째 것을 권장한다. 앞의 두 가지 방법은 DriverManager 클래스 내에서 많은 synchronized method 의 사용으로 어플리케이션을 블록킹시킬 수 있다.

5.5 결론

이것으로 Web Container 기반의 DB connection pool 에 대한 설명을 마치려 한다. Shared, non-shared, DBA connection 에 대한 설명을 하였다. 그리고 어

떻게 DB connection pool 을 설정하고 Servlet 코드 내에서 사용하는지도 살펴 보았다.

더 자세한 정보는 부록 D 를 참조한다.

다음 장에서는 중요한 개념인 context group 에 대하여 설명한다.

6 Context Group

6.1 소개

JEUS 내부에서 웹 어플리케이션 (또는 컨텍스트)은 Context Group으로 그룹핑 되어있다. 또한, 여러 개의 Context Group들도 JEUS Web Container에 존재할 수 있다 (4 장에 설명되어 있다).

Context Group 은 웹 어플리케이션에 많은 중요한 서비스들을 제공해 준다. 이러한 서비스들의 예는 Web Server 연결, JSP 컴파일, 로깅, active management, , response header 설정 등이 있다.

즉, Context Group 의 설정과 서비스들은 종속되어 있는 모든 웹 어플리케이션(컨텍스트)에 적용된다. JEUS 시스템에 웹 어플리케이션을 성공적으로 deploy 하기 위해서는 이 개념을 반드시 이해해야 한다. 이 장에서는 Context Group 에 대한 기초 지식을 제공하고 상세한 설명을 할 것이다.

6.2 Context Group 의 개요

6.2.1 소개

개념적으로, Context Group 은 “Web Container 안의 Web Container”로 생각할 수 있다. 그 뿐만 아니라 Context Group 은 복수 개의 웹 어플리케이션(컨텍스트)를 호스팅 할 수 있는 Virtual Server 라고 생각할 수 있다.

각 Context Group 에는 그에 등록된 웹 어플리케이션들이 사용할 별도의 설정과 하위 컴포넌트들이 존재한다.

많은 서비스들과 설정들이 최상위 Web Container 로 부터 상속 된다는 사실도 매우 중요하다. 앞에서 설명했듯이, 이러한 서비스들에는 DB Connection Pool 과 세션 처리 설정(세션 처리 설정이 Context Group 에 설정되어 있으면 Web Container 의 설정은 무시된다)이 있다.

6.2.2 Context Group 아키텍처

명료한 설명을 위하여, 3장에서 보여준 구조적 개요가 들어있는 그림을 다시 보자[그림 8].

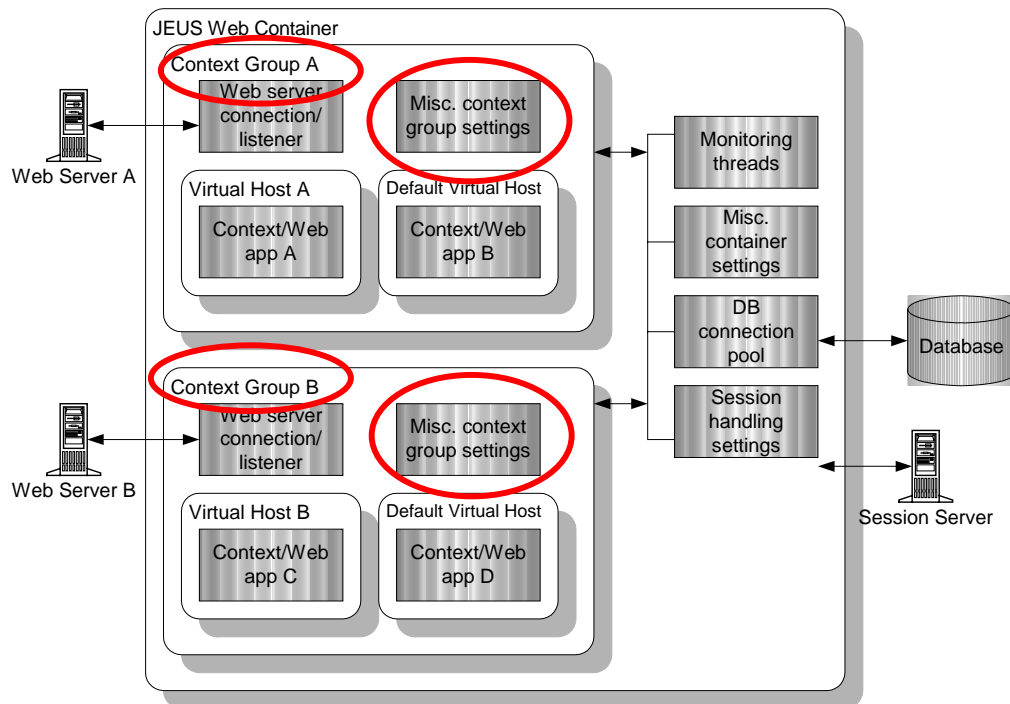


그림 8. Web Container에 연관된 Context Group. 이 장에서 다루는 부분이 강조되어 있다.

[그림 9]은 Context Group의 상세 확대한 모습을 보여주고 있다. 이 장에서 다룰 부분을 밝게 표시했다. 어두운 네모들은 나중의 장에서 다룰 부분들을 표시한다.

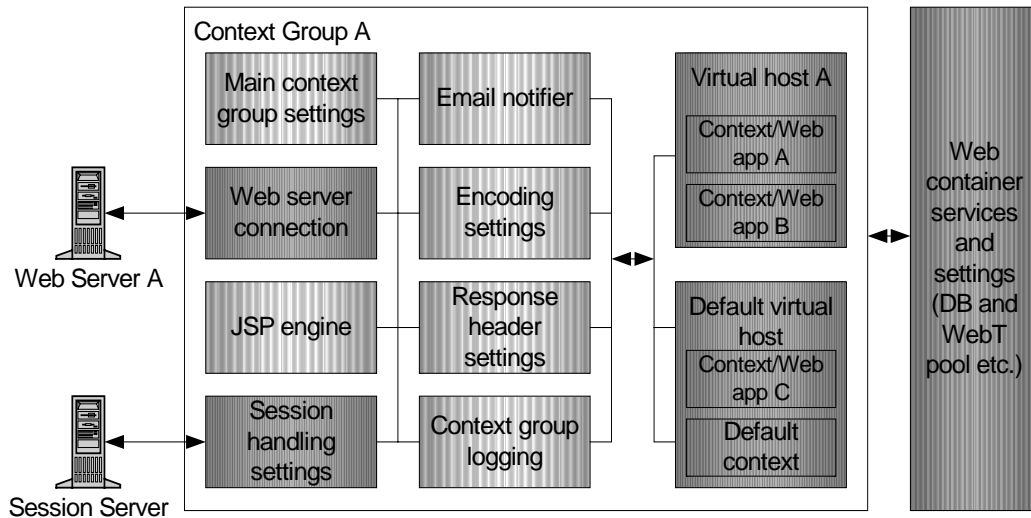


그림 9. Context Group 의 상세 모습과 그 하위 컴포넌트들.

6.2.3 Context Group 의 주 설정

각 Context Group 은 몇가지 주요 설정 항목을 가지고 있다. 이 설정들은 아주 중요하다. 왜냐하면 그 이유는 이들이 Context Group 이 전체적인 Web Container 의 구조에 어떻게 연관되어 있는지 알려주기 때문이다.

여기서 가장 유의해야 할 설정은 Context Group 을 초기화 하고 사용할 때 Web Container 가 바라보아야 할 물리적인 파일 시스템 디렉토리를 지정하고 있는 Document Base 디렉토리이다.

6.2.4 Context Group 의 주요 하위 컴포넌트들

앞에서 보았듯이, Context Group 에 관계된 주요 하위 컴포넌트들과 기능들은 다음과 같다.

- Virtual Host
- 컨텍스트
- 웹 서버 연결
- 세션 관리 설정
- 인코딩 설정
- JSP engine 설정
- 로깅

- 클러스터된 환경에서 세션 처리 (Web Container 에서의 설정을 재설정).
- response header 설정
- Email notifier

각 항목들은 다음 하위 절들에서 자세하게 설명한다.

6.2.5 Virtual Host 설정

Virtual Host 는 Context Group 레벨에서 설정한다. 여러 개의 Virtual Host 들이 추가 될 수 있으며 각 Virtual Host 들은 여러 개의 웹 컨텍스트를 가질 수 있다.

Virtual Host에 대해서는 10 장을 참고하라.

6.2.6 컨텍스트

우리가 이미 잘 알고 있듯이 컨텍스트는 Context Group 과 Web Container 에서 실행되는 웹 어플리케이션과 동일한 개념이다. 여러 개의 컨텍스트가 한 개의 Context Group 에 디플로이 될 수 있으며 Context Group 과 Web Container 의 서비스들을 모두 사용할 수 있다.

여기서 주시해야 할 것은 모든 Context Group에는 “default” 컨텍스트가 존재한다는 것이다. 컨텍스트에 대해서는 9 장에서 자세히 다룬다.

또한 컨텍스트는 Context Group 바로 아래에 또는 Context Group 내의 Virtual Host 에 바로 설정 가능하다. 앞의 경우에는 컨텍스트가 묵시적으로 기본 Virtual Host 에 속한다고 볼 수 있다.

6.2.7 웹 서버 연결

요청을 받아들이고 적당한 웹 어플리케이션에 전달하기 위해서는, Context Group 이, 클라이언트의 HTTP 요청을 받아 적절한 Context Group 에 전달하는 웹 서버와의 연결을 설정해야 한다.

그러므로, Context Group 내의 컨텍스트들은 그 Context Group 에 설정된 웹 서버를 통해서만 요청을 받을 수 있다. 이 의미는 두 가지 다른 Context Group 에 등록된 각기 다른 컨텍스트는 서로 다른 웹 서버의 서비스를 받을 수 있다

는 것이다. 이것이 각 Context Group 이 논리적으로 각기 다른 Virtual Host 로 등록될 수 있는 주요한 이유이다.

웹 서버 연결에 대한 이야기는 Context Group 레벨에서 설정되지만 중요하고 큰 주제이다. 그러므로 이것에 대한 설명은 7 장에서 진행한다.

6.2.8 세션 관리 설정

이 설정들은 Context Group 이 어떻게 생성된 HttpSession 객체를 처리하는 지 지정할 수 있게 한다.

6.2.9 인코딩

Context Group 은 등록된 모든 컨텍스트에 의해 사용될 수 있는 인코딩 설정을 가지고 있다.

여기에는 세 가지의 인코딩 설정이 존재한다.

- **Request encoding:** HTTP 요청 헤더를 위한 인코딩.
- **Postdata encoding:** HTTP 요청의 postdata 블록을 위한 인코딩.
- **Response encoding:** Web Container 로부터 받은 전체 응답 HTTP 메시지에 적용되는 인코딩.

Request Encoding

Request encoding 은 웹 브라우저로부터 받은 요청에 적용된다. 구체적으로 인코딩은 HTTP 헤더 byte 배열(HTTP URL, 질의문과 쿠키 포함)을 Web Container 의 java.lang.String 객체로 변환할 때 적용된다. Post data 를 위한 모든 인코딩은 아래(“Postdata Encoding”)에 설명한 것처럼 따로 처리한다.

요청 URI, 질의문, 쿠키에 대한 인코딩은 아래 우선순위 리스트에 따라 결정된다.

1. WEBMain.xml 에 정의된 “forced” 인코딩 .
2. 요청의 “Accept-Language” HTTP 요청 필드.
3. WEBMain.xml 에 정의된 “default”인코딩.
4. 위의 어떤 것도 적용되지 않으면 기본적으로 “ISO-8859-1”로 설정된다.

위의 리스트에서 첫 번째 것이 가장 높은 우선순위를 가지며 그것이 없으면 두 번째 것이 적용되며, 이런 식으로 순차적으로 우선순위를 갖게 된다.

따라서 관리자는 WEBMain.xml에 “default”와 “forced” 요청 인코딩을 정의할 수 있다. 6.3.7 에서 더 자세하게 설명한다.

Postdata Encoding

HTTP 요청의 post data 에 사용되는 인코딩은 HTTP 헤더 태그들(URL, 질의문, 쿠키 등)에 설정되는 것과 달리 설정된다. 전자의 인코딩은 postdata-encoding 을 통하여 설정되고 후자는 request-encoding 을 통하여 설정된다.

Postdata 에 대한 인코딩은 아래의 우선순위 리스트에 따라 결정된다.

1. WEBMain.xml 에 정의된 “forced” 인코딩.
2. HTTP body 의 “Content-Type “ 헤더 필드에 정의된 인코딩.
3. WEBMain.xml 에 정의된 “default”인코딩.
4. 위의 어떤 것도 적용되지 않으면 기본적으로 “ISO-8859-1”로 설정된다.

위의 리스트에서 첫 번째 것이 가장 높은 우선순위를 가지며 그것이 없으면 두 번째 것이 적용되며, 이런 식으로 순차적으로 우선순위를 갖게 된다.

Response Encoding

Response encoding 은 PrintWriter.println()을 byte 배열로 변환할 때나 HTTP 헤더의 “Content-Type:text/html;charset=XXX” 부분의 “XXX” 값을 설정하고 Web Container 의 응답에 어떤 인코딩을 사용할지 결정한다.

Response encoding 에 대한 인코딩은 아래의 우선순위 리스트에 따라 결정된다.

1. WEBMain.xml 에 정의된 “forced” 인코딩.
2. Servlet 에서의 세팅 (서블릿에서는 response.setContentType (“text/html;charset=XXX”), JSP에서는 <%@ page contentType=“text/html;charset=XXX”%>로 프로그래머가 설정한 XXX 값의 인코딩).
3. WEBMain.xml 에 정의된 “default”인코딩.

4. 위의 어떤 것도 적용되지 않으면 기본적으로 “ISO-8859-1”로 설정된다.

위의 리스트에서 첫 번째 것이 가장 높은 우선순위를 가지며 그것이 없으면 두 번째 것이 적용되며, 이런 식으로 순차적으로 우선순위를 갖게 된다.

위에서처럼 관리자는 WEBMain.xml에 “default”와 “forced” response encoding을 정의할 수 있다. 이는 아래 6.3.7 절에서 다시 다룬다.

인코딩 값의 예

다음 리스트는 Servlet/JSP 프로그래머나 JEUS 관리자가 WEBMain.xml에 흔히 설정할 수 있는 인코딩 값들이다.

- ISO-8859-1 (Web Container에서 기본으로 사용하고 있는 인코딩, ISO Latin)
- UTF-8 (UCS 변환 포맷)
- EUC-KR (한국어)
- EUC-JP (일본어)

6.2.10 JSP engine

각 Context Group은 JSP engine을 가지고 있다고 생각할 수 있다. JSP engine은 기본적으로 “jsp” 자원을 클라이언트가 요청하였을 때 JSP 페이지들을 컴파일 하여 Servlet 코드로 만드는 작업을 한다.

JSP engine의 설정은 컨텍스트가 속하는 Context Group의 설정에 적용받는다.

JSP에 대한 자세한 정보는 JSP 2.0 스펙을 참고한다.

6.2.11 로깅

logging 설정을 <web-container> 하위 요소로 설정하면 모든 Context Group에 공통으로 적용되는 logging이 설정된다. Logging 설정은 <context-group>의 하위 요소로도 설정할 수 있으며 그렇게 하면 소속된 Context Group에만 적용되는 logging을 설정할 수 있다. <context-group>의 하위 요소로 설정된 logging 설정은 <web-container>의 하위 요소로 설정된 logging 설정보다 우선한다. logging 설정 방법은 <web-container>의 하위로 설정하는 logging과 동일하다.

다시 한번 언급하면 system-log 의 설정은 JEUSMain.xml <system-logging> 설정을 통해 이루어진다.

User 로깅

Servlet 에서 생성된 메시지와 Servlet Context.log() method 에 의해 생성된 내용들이 별도의 “user”로그에 남겨진다.

Access 로깅

Context Group 에 대한 모든 요청과 사용자 접근이 별도의 로그 파일에 남는다.

주의: 접근이 많이 빈번한 사이트에서는 access 로그의 양이 클 수가 있으므로, access 로깅 기능을 사용하지 않는 것이 좋다.

로깅 설정

<context-group>의 하위 요소로 설정하는 <logging>은 <web-container>의 하위 요소로 설정하는 <logging>과 그 구성이 동일하다. 따라서, 자세한 설정은 4.4 절을 참조하기 바란다.

기본 logger 파일 위치

별도의 설정이 없을 경우 Web Container 당 1 개의 access-log 및 user-log file 이 생성된다. 그 기본 위치는 다음과 같다.

- **JEUS log home (JEUS_LOGHOME):** \$JEUS_HOME/logs/JeusSystem 디렉토리가 기본이다.
- **GroupName :** <context-group>의 하위 요소인 <group-name>으로 설정된 값이다.
- **access-log 기본 위치 :** <JEUS_LOGHOME> 밑의 <node_name>_<container_name>/servlet/accesslog/<GroupName>/access.log 가 기본 log file 이다.
- **access-log 기본 위치 예 :** \$JEUS_HOME 이 "c:\jeus5"이고 node_name 이 johan, container_name 이 container1, GroupName 이 MyGroup 이라면, c:\jeus5\johan_container1\servlet\accesslog\MyGroup\access.log 가 기본 로그 파일이다.

- **user-log** 기본 위치 : <JEUS_LOGHOME> 밑의 <node_name>_<container_name>/servlet/userlog/<GroupName>/user.log 가 기본 log file 이다.
- **user-log** 기본 위치 예 : \$JEUS_HOME 이 "c:\jeus5"이고 node_name 이 johan, container_name 이 container1, GroupName 이 MyGroup 이라면, c:\jeus5\johan_container1\servlet\userlog\MyGroup\user.log 가 기본 로그 파일이다.

6.2.12 클러스터 환경에서 세션 처리

클러스터된 환경에서 세션 처리는 원래 클라이언트의 세션을 생성한 Web Container 로 세션 data 를 전달하기 위해서 필요한 것이다.

Context Group 에서 Session 을 처리하는 것은 Web Container 에서 처리하는 것과 별반 다를 것이 없다. 그러나, Context Group 에서 설정한 설정은 Web Container 에서 설정한 것보다 우선 순위를 가지고 적용된다.

8 장에 더 많은 정보가 설명되어 있다.

6.2.13 Response Header

사용자의 세션을 추적하는 기본 기술은 모든 클라이언트 응답에 반환되는 session ID 쿠키라는 것이다 (8 장에 더 상세한 정보가 설명되어 있다).

Context Group 레벨에서는 session ID 쿠키와 함께 어떤 데이터(HTTP 헤더 파라미터)를 보낼 것인지 좀 더 세밀하게 지정할 수 있다. 이 데이터는 Context Group 의 response header 부분에 세션 ID **cookie version**, 쿠키 헤더의 **domain**, **path**, **max-age** 와 **secure attributes**, **jsessionid-name** 을 설정하여 보낼 수 있다.

또한, 이름과 값의 짝으로 된 임의의 HTTP 헤더를 정의할 수도 있다.

여기에 대해서는 6.3.11 에서 설명하겠다.

6.2.14 Context Group 디렉토리 구조

[그림 10]는 Context Group과 관계된 JEUS 시스템 디렉토리의 부분을 보여주고 있다.

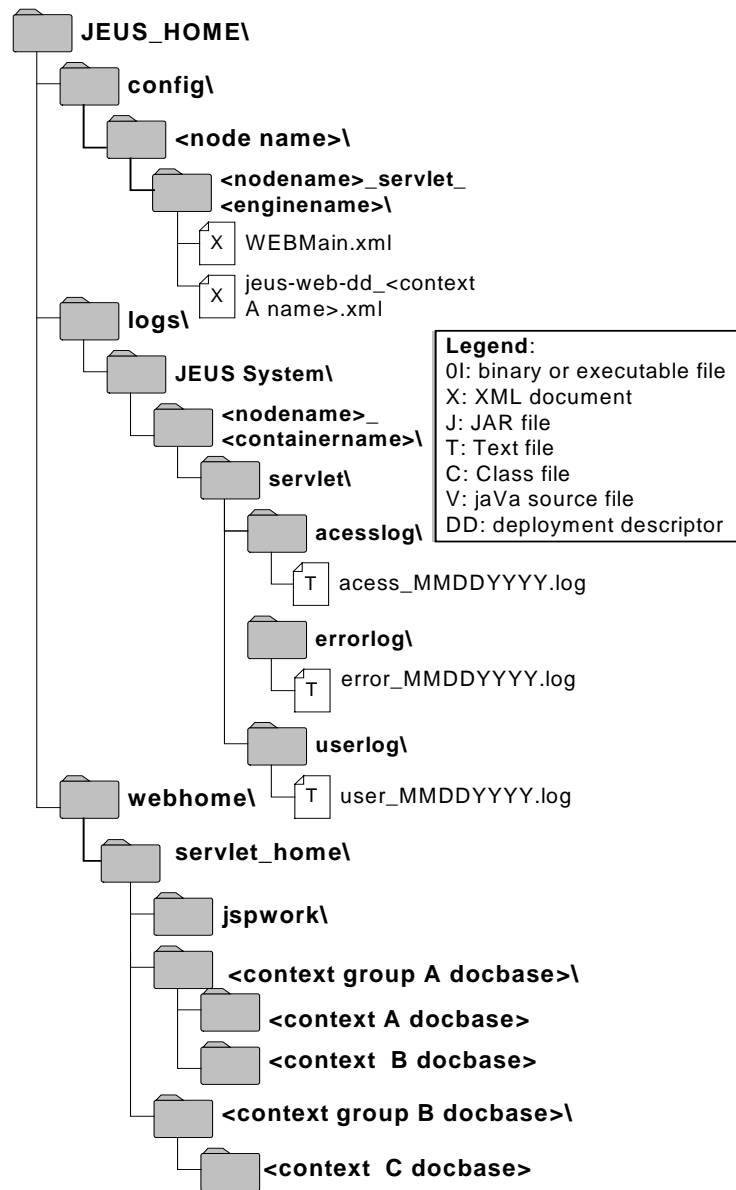


그림 10. Context Group 의 디렉토리 구조.

[그림 10]의 설명

- **JEUS_HOME\config\<node name>\<node name>_servlet_<engine name>:** 이 디렉토리는 Context Group 을 설정하기 위한 WEBMain.xml 을 가지고 있다. 이 파일은 Context Group 과 컨텍스트의 실제 매핑 정보를 가지고 있다.
- **JEUS_HOME\logs\JeusSystem\<node name>_<container name>\servlet:** 이 디렉토리는 Web Container 를 위한 로그 파일을 보

관하는 최상위 디렉토리이다. 사용자가 Context Group 별 로그를 설정하는 경우 다음과 같은 구조로 로그 파일이 생성된다.

- **userlog**\<context group name>: user 로깅 메시지가 여기에 저장된다. 이 디렉토리는 “valid day” 시간 설정이 되어 있지 않으면 JEUS_HOME\logs\JeusSystem\<node name>_ <container name>\ servlet\userlog\<context group name>\ 디렉토리 아래에 “user.log”이 생성된다. “valid day”가 사용되면, 이 디렉토리 아래의 파일들은 주어진 날들 동안 유효하고, “user_MMDDYYYY.log”와 같이 파일명이 생성된다.
- **errorlog**: error 로깅 메시지 파일은 Context Group 별로 따로 설정할 수 없다. 또한, errorlog 설정은 JEUSMain.xml 을 통해서만 설정할 수 있다(제우스 서버 지침서 참조). 따라서, error 로깅 파일은 Context Group 별로 구분하지 않는다.
- **accesslog**\<context group name>: access 로깅 메시지가 여기에 저장된다. “valid day” 시간 설정이 되어 있지 않으면 JEUS_HOME\logs\JeusSystem\<node name>_ <container name>\ servlet\accesslog\<context group name>\ 디렉토리 아래에 “access.log”이 생성된다. “valid day”가 사용되면, 이 디렉토리 아래의 파일들은 주어진 날들 동안 유효하고 “access_MMDDYYYY.log”와 같이 파일명이 생성된다.

6.2.15 결론

여기까지 우리는 Context Group 의 전체적인 구조를 살펴보았고 컨텍스트, 웹 서버 연결과 같은 주요 하위 컴포넌트와 기능들에 대해서도 살펴보았다.

다음 절에서는 어떻게 이 항목들을 설정하는지 살펴보겠다.

6.3 Context Group 설정

6.3.1 소개

Context Group 을 추가하고 설정하는 것은 아주 직관적이다. 모든 설정은 engine 설정 디렉토리 아래의 WEBMain.xml 파일 안에 존재한다.

수작업으로 WEBMain.xml 을 편집하려면 파일을 열고 <web-container>태그 아래에 한 개 이상의 <context-group>태그를 추가하고 수정하면 된다.

예:

<<WEBMain.xml>>

```
<?xml version="1.0"?>
<web-container xmlns="http://www.tmaxsoft.com/xml/ns/jeus">
    ...
    <context-group>
        ...
    </context-group>
    <context-group>
        ...
    </context-group>
    ...
</web-container>
```

다음의 하위 절들은 바로 아래의 하위 컴포넌트들이 어떻게 설정되는지에 대해 설명한다.

1. 기본 Context Group 설정.
 2. Virtual Host.
 3. 컨텍스트.
 4. 웹 서버 연결.
 5. 세션 관리 설정.
 6. 인코딩.
 7. JSP engine.
 8. 로깅.
 9. 클러스터된 환경에서의 세션.
 10. response header.
 11. Active management
- 2, 3, 4, 9 는 별도의 장에서 설명한다.

다음은 위에서 언급한 설정 사항들이 포함된 WEBMain.xml 파일의 예이다. 이 예는 XML 규칙과 순서만 보여주지만, 다음에 나오는 하위 절에서는 각 XML 부분들이 구체적으로 어떻게 설정되는지 보여줄 것이다.

```
<?xml version="1.0"?>
<web-container xmlns="http://www.tmaxsoft.com/xml/ns/jeus">
  ...
  <context-group>
    <group-name>MyGroup</group-name>
    <virtual-host>
      <!-- See chapter 10 -->
    </virtual-host>
    <context>
      <!-- See chapter 9 -->
    </context>
    <webserver-connection>
      <!-- See chapter 7-->
    </webserver-connection>
    <group-docbase>...</group-docbase>
    <session-config>
      <!-- See sub-section 6.3.6 -->
    </session-config>
    <servlet-home>...</servlet-home>
    <print-error-to-browser>...</print-error-to-browser>
    <encoding>
      <!-- See sub-section 6.3.7-->
    </encoding>
    <jsp-engine>
      <!-- See sub-section 6.3.8-->
    </jsp-engine>
    <logging>
      <!-- See sub-section 6.3.9-->
    </logging>
    <session-cluster>
      <!-- See chapter 8 -->
    </session-cluster>
    <response-header>
      <!-- See sub-section 6.3.11 -->
    </response-header>
```

```

    </context-group>
    ...
</web-container>

```

6.3.2 기본 설정

Context Group의 기본 설정은 WEBMain.xml의 <context-group> 태그 바로 아래에 설정되고 아래와 같이 정의된다.

- **context group name**은 내부적으로 통용되는 Context Group의 식별자이다.
- “**print error to browser**” Boolean 설정은 Server 쪽의 문제가 발생했을 때 오류의 상세 내역을 브라우저로 보여줄 것인지에 대한 설정이다. 이 메시지는 개발 시에는 유용하지만 운영 시에는 제거하는 것이 바람직하다.

기본적인 설정은 다음과 같이 WEBMain.xml에 할 수 있다.

```

<<WEBMain.xml>>

<?xml version="1.0"?>
<web-container xmlns="http://www.tmaxsoft.com/xml/ns/jeus">
    ...
    <context-group>
        <group-name>MyGroup</group-name>
        ...
        <group-docbase>webapps</group-docbase>
        ...
        <print-error-to-browser>true</print-error-to-browser>
        ...
    </context-group>
    ...
</web-container>

```

6.3.3 Virtual Host 설정

10장에서 Virtual Host와 그 설정에 대해 자세히 다루겠다.

6.3.4 컨텍스트 설정

9장에서 컨텍스트와 그 설정에 대해 자세히 다루겠다.

6.3.5 웹 서버 연결 설정

7 장에서는 웹 서버 연결에 대해 자세히 다루겠다.

6.3.6 세션 관리 설정

세션 설정은 어떻게 Context Group 과 Web Container 가 HttpSession 객체를 내부적으로 처리할 것인지에 대한 정책을 설정하는 것이다.

WEBMain.xml 의 <session-config> 태그에 다음과 같은 설정을 할 수 있다.

- **Session timeout** 설정은 Servlet Session 의 유효주기에 대한 글로벌 정책을 정의한다. 이 설정은 해당 web.xml 에 Session 유효주기가 설정되어 있지 않으면 사용된다. 값은 분(min) 값을 사용한다.
- **shared Session Boolean** 설정은 같은 세션 객체가 다른 컨텍스트들에 공유될 것인지에 대한 것을 결정 짓는다 (즉, 컨텍스트 A 에서 만들어진 HttpSession 객체가 컨텍스트 B 에서도 같은 사용자에게 의해 사용될 수 있는지에 대한 것이다).
- **persistent Session** 옵션은 이 태그가 “false”로 설정되어 있고 Session Server 가 설정(JEUSMain.xml 의 session-cluster 태그에서)되어 있지 않으면 세션 객체들과 Servlet 컨텍스트 속성들이 Servlet 컨텍스트가 변경되어 리로딩 될 때 삭제된다. 그러나, 만약에 이 태그(persistent)가 “true”로 설정되어 있으면 Session Server 가 정의되어 있지 않더라도 세션 객체와 Servlet 컨텍스트 속성은 Servlet 컨텍스트 리로딩 시에 없어지지 않는다. 그러나 클래스가 변경된 객체나 속성은 삭제된다. Session Server 가 정의되어 있으면 세션 객체와 Servlet 컨텍스트 속성은 이 태그의 값이 어떤 것이든 하더라도 삭제되지 않는다.

다음은 XML 예이다.

```
<?xml version="1.0"?>
<web-container xmlns="http://www.tmaxsoft.com/xml/ns/jeus">
  ...
  <context-group>
    ...
    <session-config>
      <timeout>20</timeout>
      <shared>true</shared>
      <persistent>true</persistent>
    </session-config>
```

```

    ...
    <context-group>
    ...
</web-container>

```

6.3.7 인코딩 설정

앞의 6.2.9 에서 설명하였듯이, WEBMain.xml에 **request, response, postdata encoding**을 설정할 수 있다. 각 설정은 “**default**”나 “**forced**”로 설정 가능하다. 전자는 어떤 인코딩도 없을 경우에 기본 값으로 사용하고 후자는 모든 경우에 항상 강제적으로 사용하도록 한다.

예

```

<?xml version="1.0"?>
<web-container xmlns="http://www.tmaxsoft.com/xml/ns/jeus">
    ...
    <context-group>
        ...
        <encoding>
            <request-encoding>
                <default>EUC-KR</default>
                <forced>UTF-8</forced>
            </request-encoding>
            <response-encoding>
                <default>EUC-KR</default>
                <forced>UTF-8</forced>
            </response-encoding>
            <postdata-encoding>
                <default>EUC-KR</default>
                <forced>UTF-8</forced>
            </postdata-encoding>
        </encoding>
    </context-group>
</web-container>

```

6.3.8 JSP engine 설정

각 context group 에는 어떻게 JSP engine(JSP 컴파일러)이 Web application 에 대하여 어떻게 동작해야 하는지에 대한 설정이 들어 있다.

JSP 설정은 WEBMain.xml 의 <context-group>태그 내의 <jsp-engine>에 정의되어 있으며 다음과 같이 정의된다.

- **Java compiler options** 은 JSP 에서 생성된 java 소스코드를 Servlet 클래스 파일로 컴파일 할 때 필요한 컴파일러를 설정한다. 여기에는 다음과 같은 옵션을 포함한다.
 - 사용할 **Java compiler** (예: “javac”).
 - **compiler output dir** (JSP workdir 을 번복한다).
 - **Compiler options** (예: “-verbose -g:none”).
 - **Compiler encoding** (예: “UTF8”). 이 설정은 Web Container 가 적당한 인코딩을 알아서 지정하므로 많이 사용되지 않는다.
- **keep generated** Boolean 옵션은 JSP 포맷에서 변환을 거친 후 Servlet 소스 코드를 저장해 둘지를 결정한다. 디버깅을 위해서 이 옵션을 “true”로 해 놓으면 유용하다.
- **check included JSP file** 설정이 “true”로 설정되면 요청한 JSP 페이지 뿐만 아니라 <%@ include file=“xxx.jsp” %> directive 로 include 된 모든 JSP 들에 대하여 변경되었는지 확인한다. 이 설정이 “true”로 설정되고 include 된 JSP 파일이 변경되었음이 확인되면 변경된 파일은 재컴파일 된다. 기본 설정인 “false”는 변경 확인을 하지 않는다. 즉, 요청된 JSP 만 변경확인 된다.

예:

<<WEBMain.xml>>

```
<?xml version="1.0"?>
<web-container xmlns="http://www.tmaxsoft.com/xml/ns/jeus">
  ...
  <context-group>
    ...
    <jsp-engine>
      <keep-generated>true</keep-generated>
      <java-compiler>javac</java-compiler>
      <jsp-work-dir>c:\MyJSPWorkdir</jsp-work-dir>
      <compile-output-dir>
        c:\MyJSPWorkdir\
```

```

        </compile-output-dir>
        <compile-option>-g:none -verbose</compile-option>
        <compile-encoding>8859_1</compile-encoding>
        <check-included-jspfile>true</check-included-jspfile>
    </jsp-engine>
    ...
</context-group>
...
</web-container>

```

6.3.9 로깅 설정

Context Group 에 관련하여 마지막으로 설정할 수 있는 것은 로깅이다. 앞에서 보았듯이, 여기에는 두 가지 종류의 로깅 메시지가 있다.

- User 메시지: ServletContext.log() method 호출로 Servlet 프로그래머에 의해 지정된 로깅 데이터.
- Access 로그: Context Group 과 그 컨텍스트의 모든 클라이언트 요청을 레코딩 한다.

이 설정들은 WEBMain.xml 의 <context-group><logging> 태그의 각기 다른 태그로 설정된다. <user-log> , <access-log> .

<context-group> <logging> 설정은 <web-container> <logging> 과 완전히 동일하다. 따라서, 자세한 설정은 4.4 절을 참조하기 바란다. <context-group> 밑의 <logging> 설정이 <web-container> 밑의 <logging> 보다 우선 순위가 높다.

6.3.10 클러스터된 환경에서의 세션 설정

8 장에 이 설정에 대한 설명이 자세히 되어 있다.

Context Group 레벨에서 정의된 클러스터링 설정은 Web Container 레벨에서 정의된 설정을 반복한다.

6.3.11 Response Header 설정

response header 설정은 세션 ID 쿠키 HTTP header 에 전달될 값들을 지정한다. 이 설정들은 Context Group 의 모든 Session ID 쿠키에 적용된다.

세션 ID 쿠키 response header 는 <response-header><session-id-cookie> 태그에 설정한다.

설정들은 다음과 같다.

- 세션 ID **cookie version** (0 = NS 쿠키 또는 1 = RFC 스펙 쿠키)
- 쿠키 헤더의 **domain, path, max-age** 와 **secure attributes**. 부록 D 또는 쿠키 스펙에서 더 상세한 설명을 참고하기 바란다.
- **jsessionId-name** : 세션 ID 이름으로 표준 이름인 “JSESSIONID” 을 사용하지 않고 다른 이름을 사용하고자 할 때 이 설정을 사용한다.

또한 *custom response header* 라는 것을 설정할 수 있다. 이 헤더들은 밖으로 나가는 모든 HTTP 메시지에 이름/값 쌍으로 표현되어 첨부된다.

Custom response header 는 각 Context Group 아래의 <response-header><custom-header>태그에 설정된다. 이 태그가 존재하면 다음 두 항목과 함께 복수 개의 <header-field> 태그가 포함 될 수 있다.

- **field name** 태그는 custom header 필드의 이름을 지정한다.
- **field value** 태그는 field name 과 함께 전달될 값을 지정한다.

다음 XML 예제는 response header 의 설정을 표현하고 있다.

<<WEBMain.xml>>

```
<?xml version="1.0"?>
<web-container xmlns="http://www.tmaxsoft.com/xml/ns/jeus">
  ...
  <context-group>
    ...
    <response-header>
      <session-id-cookie>
        <jsessionId-name>JSESSIONID</jsessionid-name>
        <version>1</version>
        <domain>.foo.com</domain>
        <path>/examples</path>
        <max-age>3600</max-age>
        <secure>true</secure>
      </session-id-cookie>
      <custom-header>
        <header-field>
          <field-name>Test</field-name>
```

```

        <field-value>TestValue</field-value>
    </header-field>
</custom-header>
</response-header>
...
</context-group>
...
</web-container>

```

참고: 앞으로 추가적으로 더 많은 response header 설정들을 사용할 수 있게 될 것이다.

6.3.12 웹 관리자를 통한 Context Group 설정

웹 관리자를 이용하여 WEBMain.xml 에서 Context Group 의 설정에 대한 설명은 JEUS 웹 관리자 안내서의 컨텍스트 그룹 설정을 참고하도록 한다.

6.3.13 결론

지금까지 Context Group 의 설정 가능한 모든 컴포넌트들에 대하여 살펴보았다. 더 상세한 설명은 <context-group> 태그에 대한 설명이 있는 부록 D 를 살펴보자.

다음의 내용은 Context Group 을 제어하는 방법을 설명한다.

6.4 Context Group 제어

6.4.1 소개

사용자는 소수의 명령어들로 Context Group 을 제어할 수 있다(reload, restart, suspend, resume, undeploy). 이 명령어들은 webadmin 콘솔 툴이나 웹 관리자를 통하여 수행할 수 있다.

실제, Context Group 레벨에서 수행되는 명령어는 각 하위 컨텍스트에 전달된다 (9 장을 참고하기 바란다).

6.4.2 5 개의 명령어

5 개의 명령어는 다음과 같이 풀이될 수 있다.

- **reload:** Context Group 과 컨텍스트 데이터를 디스크로부터 다시 읽어 들인다.

- **restart:** Context Group 을 재시작 한다. 즉, 모든 컨텍스트들을 재시작 한다. 이 명령어는 ‘undeploy’를 수행하고 이어서 ‘deploy’명령을 수행 한다.
- **suspend:** 임시적으로 Context Group 과 그의 모든 컨텍스트들을 종료 한다. 임시 종료된 Context Group 은 클라이언트 요청을 서비스하지 못한다.
- **resume:** 임시적으로 종료된 Context Group 을 다시 서비스 상태로 돌 린다.
- **terminate:** Web Container 의 운영환경에서 Context Group 과 그 컨텍 스트를 제거한다. 모든 파일들과 설정들은 그대로 보존 된다.

6.4.3 webadmin 콘솔 툴을 사용한 Context Group 제어

webadmin 콘솔 툴에서는 Context Group 을 제어하기 위해 다음과 같은 명령 어들을 사용할 수 있다.

```
reload <context group name>
restart <context group name>
suspend <context group name>
resume <context group name>
terminate <context group name>
```

참고: “terminate”는 위에서 설명한 “undeploy”와 같은 의미를 가진다.

예:

```
C:\> webadmin johan_container1
```

인증 정보를 입력한다. 명령 프롬프트가 나타난다.

```
$$$ [johan:johan_container1] suspend MyGroup
```

“MyGroup”이라 명칭된 Context Group 이 임시 중지되고 서비스도 중지될 것 이다. 이 명령의 결과로 다음과 같은 메시지가 나타날 것이다.

```
suspend successful
```

6.4.4 웹 관리자를 사용한 Context Group 제어

웹 관리자를 사용하여 Context Group 을 제어하기 위한 방법에 대한 설명은 JEUS 웹 관리자 안내서의 컨텍스트 그룹 제어 장을 참고 하도록 한다.

6.4.5 결과

지금까지 Context Group 을 제어할 수 있는 5 개의 기본 명령어들을 살펴보았다. Webadmin 과 웹 관리자에서 같은 명령을 수행할 수 있다는 것도 알았다.

다음 절에서는 어떻게 Context Group 을 모니터링 할 수 있는지 설명하겠다.

6.5 Context Group 모니터링

6.5.1 소개

webadmin 과 웹 관리자 기능은 Context Group 을 모니터링 할 수 있는 강력한 기능을 가지고 있다. 지금부터 어떤 것들이 있는지 살펴보도록 하자.

6.5.2 webadmin 을 사용한 Context Group 모니터링

webadmin 에서 ‘info’를 수행하여 deploy 되어 있는 모든 Context Group 을 리스트해 보자. ‘info’<context group name>은 특정한 Context Group 에 대한 정보를 제공한다.

6.5.3 웹 관리자를 사용한 Context Group 모니터링

웹 관리자를 사용하여 Context Group 을 모니터링하는 방법에 대한 설명은 JEUS 웹 관리자 안내서의 컨텍스트 그룹 통계 장을 참고한다.

6.5.4 결론

webadmin 콘솔 툴과 웹 관리자에서 어떻게 Context Group 을 모니터링 하는 지에 대해 살펴보았다.

다음 절에서는 최적의 성능을 내기 위해 Context Group 을 설정하는 방법에 대해 알아보자.

6.6 Context Group 튜닝

Context Group 의 최적화된 성능을 주기 위해서는 다음의 두 가지를 반드시 고려해야 한다.

- 일반적으로 JSP engine 의 컴파일러는 변경 할 필요가 없다. “sun.tools.javac” 설정으로 그대로 두어 또 다른 외부 프로세스가 JSP 페이지를 컴파일 할 때 필요하지 않도록 한다.
- <check-included-jspfile>는 include 된 JSP 들이 자주 변경되지 않는다면 설정을 “false”로 그대로 둔다. 이것은 include 된 JSP 파일들에 대한 변경 점검을 하지 않기 때문에 성능 향상에 도움이 된다.
- 모든 종류의 로그는 성능향상을 위하여 다음의 설정을 사용한다. “fatal” 로그 레벨과 “file” 타겟과 큰 버퍼 사이즈

6.7 결론

지금까지 Context Group 의 가장 중요한 사항들에 대하여 알아보았다.

Context Group 이 컨텍스트들을 그룹화 하는데 사용된다는 것을 알았고 웹 서버와 연결을 설정한다는 것도 알았다.

이 장의 후반부에서는 Context Group 을 어떻게 설정, 제어 모니터, 튜닝하는 지에 대해서도 알아보았다.

다음 장에서는 어떻게 Context Group 과 그 컨텍스트들을 앞 단의 웹 서버와 (또는 직접 HTTP 클라이언트와) 연결시키는지에 대해 알아보겠다.

7 웹서버 연결과 클러스터링

7.1 소개

Web Container 를 사용하기 위해서는 HTTP 클라이언트와 Web Container 사이에서 중간자 역할과 코디네이터 역할을 하는 한 개 이상의 웹서버를 설정해야 한다. 이런 관점에서 봤을 때 웹서버는 클라이언트와 Web Container 구조에서 중간 계층 역할을 수행한다.

웹서버의 기능은 기본적으로 클라이언트의 HTTP 요청을 받고 분석하고, 뒤단의 Web Container 에 있는 Context (Web application)에 전달해야 할 요청이라고 판단되면 Container 로 요청을 전달하는 것이다. Container 는 그 요청을 분석하고 수행하여 클라이언트에게 응답을 전달할 수 있는 웹서버에게 돌려보낸다.

다음에서 살펴보겠지만 그러한 중간 계층으로서 웹서버는 두 가지 종류가 있다. WebtoB 웹서버와 Apache 웹서버가 그것이다. 이 두 가지의 웹서버 외에 Container 자체에는 개발과 테스트 용도로 사용할 수 있는 두 개의 간단한 웹서버와 “TCPListener”라는 특수 목적을 가진 웹서버가 존재한다.

이 장에서는 Web Container 의 앞 단에서 사용할 수 있는 한 개 이상의 웹서버를 설정하는데 알아야 할 사항들과 자체적으로 가지고 있는 웹서버를 최대한 이용하는 방법에 대하여 살펴 보겠다. 고급 수준의 사용자들은 해당 웹서버에 대한 매뉴얼을 찾아서 공부할 것을 권장한다.

7.2 웹서버 연결의 개요

7.2.1 소개

이 절에서는 Web Container, 웹서버, 클라이언트 간의 기본적인 구조를 설명하고 Web Container 와 연결할 수 있는 여덟 가지 종류의 웹서버 리스너들에 대하여 알아보겠다.

이 장의 하위 절에서는 Web Container 의 앞에 웹서버를 어떻게 설정하는지에 대하여 알아보겠다.

[그림 11]은 웹서버 연결에 관련된 Web Container의 주요 부분을 보여주고 있다.

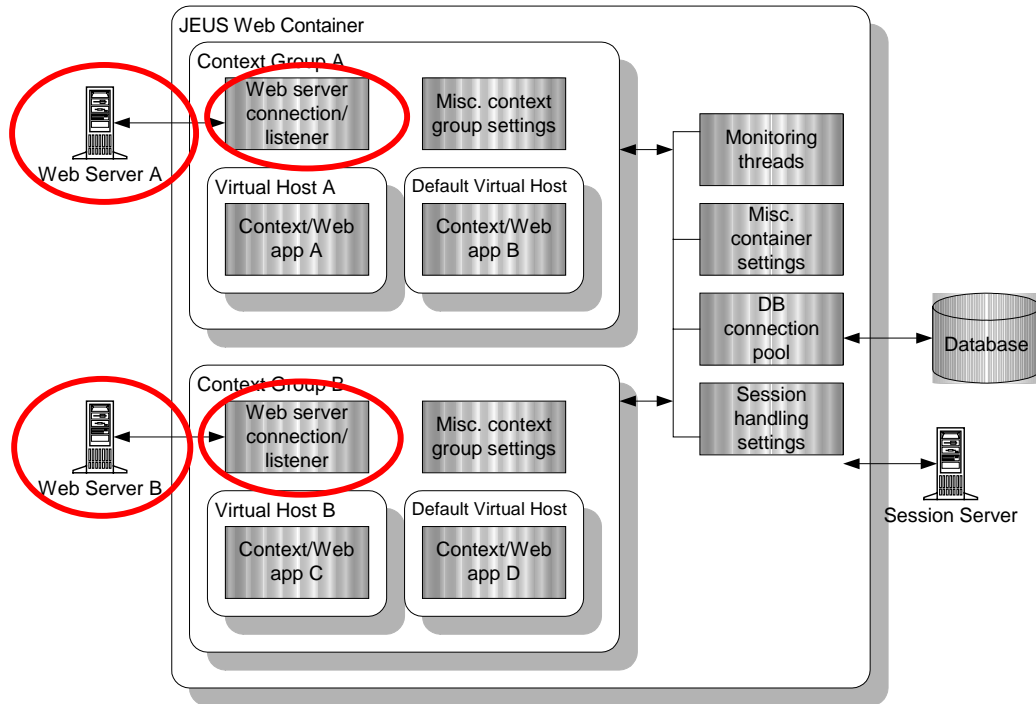


그림 11. JEUS Web Container 중 Web Server 연결 부분 컴포넌트

7.2.2 리스너란 무엇인가?

우선 “리스너”가 어떤 의미를 가지고 있는지에 대하여 분명히 이해 해야 한다. 리스너는 일반적으로 웹서버나 HTTP 클라이언트가 직접 접근 할 수 있는 Web Container 쪽의 소켓이라고 생각할 수 있다. 이 리스너(소켓)는 웹서버(또는 HTTP 클라이언트)로부터 요청을 받고 Web Container 에서 처리한 static 또는 dynamic content 를 반환한다 (여기서 dynamic content 는 JSP 나 Servlet 과 같은 Java 기반의 콘텐츠를 의미하고 static content 는 HTML 페이지나 이미지 파일과 같은 이미 생성된 데이터를 의미한다.).

리스너에는 두 가지 종류가 존재한다. 웹서버 리스너와 클라이언트 리스너가 그것이다[그림 12].

1. 웹서버 리스너는 외부의 웹서버들과 연결되고 이들과 통신을 위해서는 맞춤 프로토콜을 사용한다. 클라이언트는 이 웹서버를 통하여 Web Container 와 통신한다.

2. 클라이언트 리스너는 주로 클라이언트와 직접 연결되고 HTTP 프로토콜을 사용한다.

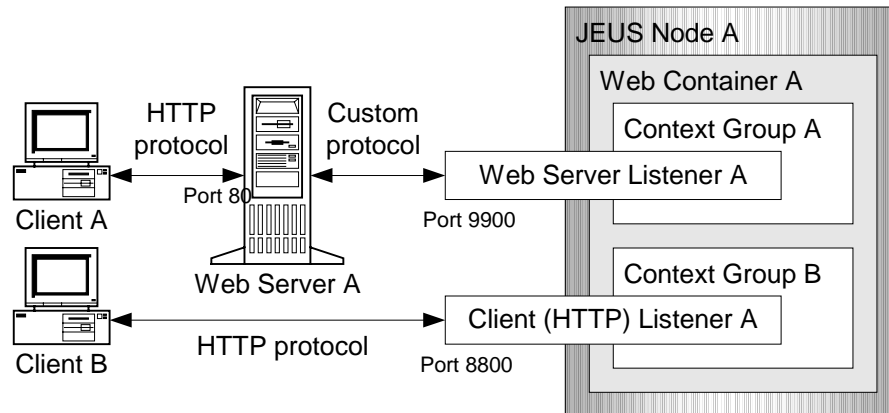


그림 12. Web Container 의 웹서버와 클라이언트 리스너.

이 리스너들은 종류에 상관없이 Container 레벨에서 설정되지 않고, Context Group 레벨에서 설정된다 [그림 12].

주목할 점은 Context Group 리스너의 또 다른 사항은 이론적으로는 다수의 리스너가 각 Context Group 에 설정될 수 있다는 것이다. 단 한 가지의 제약 조건은 각 리스너들이 각기 다른 리스닝 포트를 지정 받아 설정되어야 한다는 것이다.

7.2.3 Worker thread 와 Worker thread pool

웹서버 리스너와 연관된 중요한 개념 중 하나가 “Worker thread”에 관한 것이다. 각 리스너에는 풀(Worker thread pool)이라는 것이 포함되어 있다. 이것은 Worker thread 들을 관리한다. 리스너의 포트로 요청이 도착했을 때 한 개의 Worker thread 가 이 풀에서 꺼내어지고, 요청을 처리하기 위해 지정 받은 후 응답을 만들어 낸다. 여기에서의 “처리”라는 개념은 static content 를 가져오는 것로부터 JSP 나 Servlet 을 실행하는 것까지 모두를 포함한다.

“Worker thread”라는 개념은 이 문서의 많은 부분에서 거론된다. 예를 들어 Context Group 의 Active-Management 설정은 직접적으로 이 Worker thread pool 에 연관되어 있다. 그러므로, “쓰레드 풀 포트” 또는 “쓰레드 풀 주소”라는 개념이 사용될 때에는 Worker thread 를 주관하는 리스너의 포트 번호와 IP 주소를 의미하게 된다(그리고 간접적으로 Worker thread 도 의미한다).

“Worker thread pool” 대신에 “쓰레드 풀”을 사용하기도 한다.

다음에 설명하겠지만, 웹서버 리스너를 설정할 때에는 쓰레드 풀의 일정한 사양도 같이 설정해야 한다.

7.2.4 쓰레드 풀의 Active-Management 와 상태 통보

각 리스너에 설정되는 쓰레드 풀은 Active-Management 에 관한 설정도 포함되어 있다.

Active-Management 는 관리자가 설정된 상태가 되면 Web Container 가 경고 email 통지 또는 Web Container 가 자동으로 재시작 하도록 설정할 수 있게 한다.

설정되어야 할 조건은 언제 Servlet Worker thread 가 블록 되었다고 할 수 있는지 그리고, 특정 작업(경고 email 통보 또는 Container 재시작)이 시작되기 전에 몇 개까지의 블록 된 쓰레드들이 존재할 수 있는 지들을 설정한다.

경고: 재시작 옵션은 임시적인 해결책이고, 재시작이 발생할 경우에는 관리자가 원인을 찾아 반드시 문제를 해결해야 한다. 그리고 Active-Management 기능을 설정할 때 너무 자주 발생하지 않도록 유의해야 한다.

7.2.5 여러 개의 Web Container 와 Web Server 클러스터링

여러 개의 웹서버와 리스너들을 연결시키는 것을 클러스터링이라고 부르고 이 과정에서 만들어진 조직체를 클러스터라고 부른다. 거대한 사이트에서는 많은 양의 클라이언트 요청을 처리하기 위하여 “부하 분산”이라는 기술과 함께 반드시 사용되어야 하는 기술이다.

부하 분산은 기본적으로 클러스터 내의 어떤 Server 에게 요청을 처리하도록 지정할 것인지 정의하여 클러스터 내의 Server 들에게 골고루 요청이 전달되어 처리되도록 한다. 부하 분산기는 모든 클라이언트 요청을 받아 그 순간 가장 여유로운 웹서버에게 지정해 주는 소프트웨어이다.

이 개념들은 [그림 13]에 잘 설명되어져 있고, 여기서는 두 개의 웹서버 (WebtoB 또는 Apache)가 각각 두 개의 Web Container에 연결되어 있는 작은 클러스터 구조를 보여주고 있다.

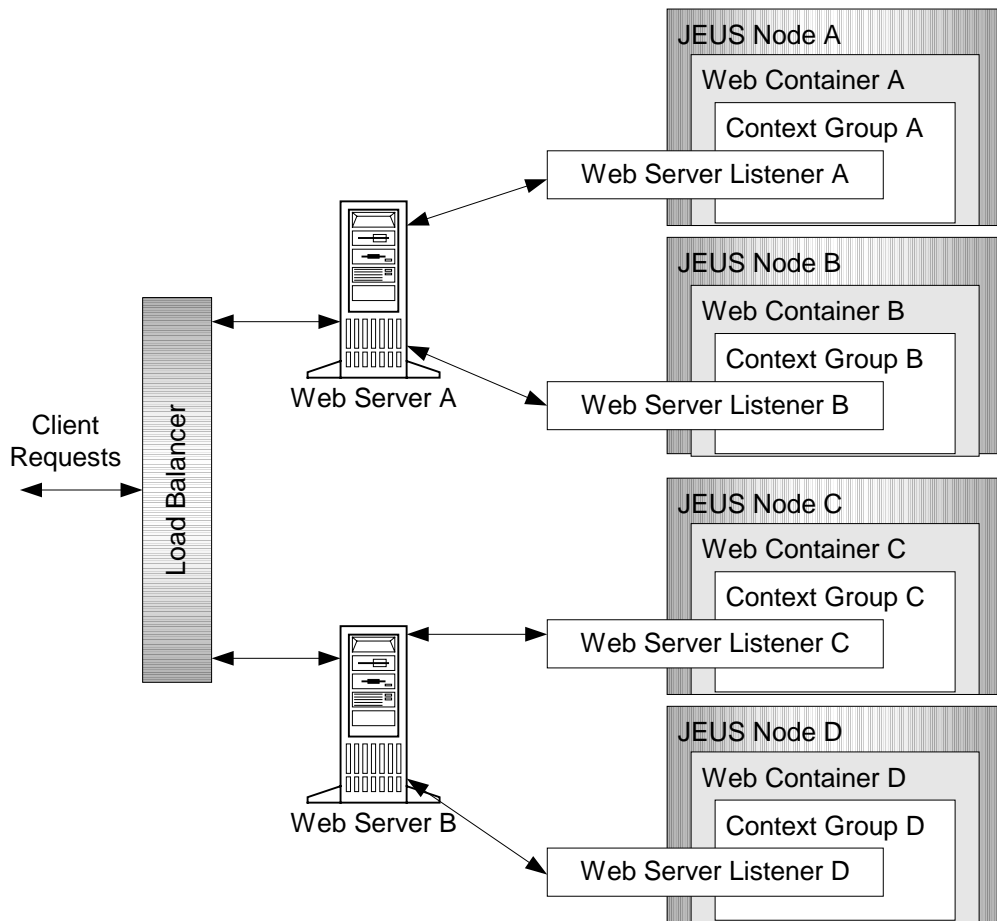


그림 13. 두 개의 웹서버가 두 개의 Web Container 에 각각 연결되어 있는 작은 클러스터

중요:[그림 13]와 같은 설정은 각 Web Container에 동일한 Context Group 과 Context를 가지고 있어야 한다 (즉, Context A, B, C, D는 모두 같은 것이어야 한다).

또한, 웹서버와 리스너의 연동과 클러스터링에서 중요한 것은 어떻게 클라이언트 Session 을 Tracking 하는지에 대한 것이다. 클라이언트 Session 데이터가 여러 개의 Web Container(Context)에 정확하게 분배되는지에 대한 사항은 클러스터 규모가 커질수록 더 중요해 진다. 이 중요한 사항에 대해서는 8장에서 따로 다룬다. 이 장에서는 Session 데이터의 존재에 대하여 무시하고 진행한다.

이러한 Web Server 클러스터를 실제로 어떻게 설정하는지에 대하여 이 장의 7.4 절에서 자세히 다루겠다.

7.2.6 여덟 가지의 리스너들

각 Context Group 에 설정될 수 있는 여덟 종류의 리스너에 대하여 자세히 알아보자. 여기에는 다음과 같은 것들이 있다.

- HTTP 리스너
- HTTPS 리스너 (“보안” 요소 추가)
- TCP 리스너
- UDP 리스너
- Apache 리스너
- WebtoB 리스너
- AJP13 리스너
- Tmax 리스너

처음의 네 리스너들은 클라이언트 리스너들이고 다음 세 개는 웹서버 리스너들이며, 마지막으로 Tmax 리스너는 Tmax 와의 연동을 위한 특별한 리스너이다. 아래의 하위 절에서는 각각에 대한 설명을 하고 있다.

7.2.7 HTTP 리스너

HTTP 리스너는 Web Container 가 자체적으로 가지고 있는 최소 규모의 “웹 서버”라고 할 수 있다. 이것은 static content 와 JSP/Servlet 에 대한 기본 HTTP 요청을 받고 처리할 수 있는 기능을 가진다. 하지만, CGI, PHP, SSI, 그리고 SSL/HTTPS 와 같은 보안 기능은 지원하지 않는다.

그렇기 때문에 실 운영환경에서는 내부 HTTP 리스너의 사용을 권장하지 않는다. 내부 Server 는 개발과 테스트 목적으로 적합하다.

따라서, 운영환경에서는 WebtoB 또는 Apache 를 사용한다.

작은 규모의 운영환경(100~200 동시 사용자)에서는 HTTP 리스너 사용을 고려할 수도 있다. 기능면에서 좀 부족함이 있지만 이러한 제한된 환경 내에서는 WebtoB 나 Apache 보다 좋은 성능을 보여준다.

7.2.8 보안 리스너

보안 리스너는 HTTP 리스너와 동일하지만 SSL(HTTPS)도 지원한다는 것이 다르다. 7.3 과 7.6 절에 상세한 설명이 있다.

7.2.9 TCP 리스너

TCP 리스너는 일반적인 리스너에 맞춤 프로토콜(HTTP 를 사용하지 않음)을 사용하는 리스너이다.

7.3 절과 7.5 절에서 상세히 설명한다.

7.2.10 UDP 리스너

UDP 리스너는 TCP 리스너와 동일하나 TCP 대신 UDP 프로토콜을 사용한다는 점만 다르다. 따라서 이후에는 특별한 언급이 없을 경우 TCP 리스너와 동일하다고 보면 된다.

7.2.11 Apache 리스너

Apache 리스너는 앞단의 Apache Server와 연동을 가능하게 한다. Apache Open Source Server는 <http://www.apache.org/>에서 다운 받을 수 있다.

Apache 리스너는 위에서 설명한 세 개의 리스너와 같은 방식으로 작동한다. 다른 점은 웹서버가 내부적으로 운영되는 반면 Apache 웹서버가 Web Container 의 앞 단에서 외부적으로 실행되고 있다는 것이다.

7.3 절과 7.4 절에서 상세히 설명한다.

7.2.12 WebtoB 리스너

WebtoB 는 JEUS Web 어플리케이션 Server 의 기본 웹서버다. WebtoB 는 static 페이지 전송, CGI, SSI, PHP 등 기본적인 웹서버 기능들을 모두 지원한다. JEUS Web Container 와 인터페이스 할 때에는 Servlet/JSP 서비스도 제공한다.

WebtoB 리스너는 위에서 언급한 리스너와 조금 다른 종류의 리스너라고 할 수 있다.

WebtoB 리스너는 다른 리스너와 달리 리스너가 WebtoB Server 의 위치를 찾아서, 접속하고자 하는 특징을 가진다. 그러므로, WebtoB 리스너를 사용할 때에는 WebtoB Server 가 리스닝 모드로 대기하고, WebtoB 리스너(즉, Web Container)가 연결을 시도한다. 이러한 연결방식을 Reverse Connection Pooling 이라 한다.

중요: 위 문장은 WebtoB Server 가 Web Container 보다 먼저 구동 중에 있어야 한다는 것을 뜻한다.

이런 특징의 결과로 방화벽 밖에 WebtoB Server 를 위치하고 안에서 리스너를 이용하여 연결을 맺을 수 있다. 이것은 방화벽이 주로 밖으로부터의 연결시도를 억제하고 안으로부터의 연결은 가능하게 하여 방화벽의 장점을 그대로 살릴 수 있는 특징을 부여한다. 다른 종류의 리스너와 웹서버를 방화벽과 사용할 경우에는 이러한 구성이 어렵다.

둘째, WebtoB 와 Web Container 가 같은 머신 내에 존재하면 둘 간의 통신은 Pipe 통신을 사용한다. 일반적인 소켓 방식을 사용하는 것보다 Pipe 통신을 함으로써 월등한 성능 향상을 기대할 수 있다.

참고: WS 엔진은 노드 당 하나만을 사용할 수 있다.

7.3 절과 7.4 절에서 상세히 설명한다.

7.2.13 AJP13 Listener

이 리스너는 AJP 1.3 protocol 을 사용하는데, Apache listener 가 AJP 1.2 protocol 을 사용하는 것을 제외하고는 Apache listener 와 동일하게 동작한다.

AJP 1.3 listener 는 Apache listener 보다 강력한 기능을 가진다. 따라서 Apache 서버가 AJP 1.3(mod_jk module 을 통해서)을 지원한다면 AJP 1.3 listener 를 사용할 것을 권장한다.

7.2.14 Tmax Listener

Tmax 는 분산 환경에서 이질적인 자원을 통합해 주는 시스템 소프트웨어이다. Tmax 리스너는 Tmax 와 연동하기 위한 특수한 리스너로 WebtoB 리스너와 마찬가지로 활동적인 리스너이기 때문에 Web Container 를 구동시키기 전에 Tmax 가 먼저 구동되어 있어야 한다.

Tmax 리스너는 Jeus 와 Tmax 간의 정보를 주고 받거나, http 요청을 Tmax 의 Gateway 를 통해 받음으로써 통신 채널을 일원화 하는 등의 용도로 사용될 수 있다.

7.2.15 결론

이것으로 Web Container/Context Group 의 구조에 대한 설명을 마친다. 웹서버 리스너와 클라이언트 리스너의 개념에 대하여 알아보았고, “Worker thread”와 “Worker thread pool”에 대해서도 알아보았다. 여덟 가지 종류의 리스너가 Context Group 에 사용될 수 있음을 확인하였으며, 그 중 세 개는 웹서

버 리스너이고, 다른 네 개는 클라이언트 리스너이며, 나머지 하나는 Tmax와의 연동을 위한 특별한 리스너이다.

다음 장에서는 이 리스너들을 Context Group의 관점에서 어떻게 설정할 수 있는지에 대하여 알아보겠다.

7.3 리스너 설정

7.3.1 소개

리스너를 설정하는 작업은 의외로 간단하다. 그러나, 웹서버 리스너를 실제 웹서버(WebtoB 또는 Apache)와 연동 시킬 경우에는 웹서버 쪽에서도 설정이 필요하다. 그러나 클라이언트 타입 리스너는 그 자체가 완전한 웹서버 역할을 하기 때문에 별도의 설정이 필요하지 않다.

이 절에서는 Web Container 쪽의 Context Group 리스너의 설정에 대해서만 알아보겠다. 여기에서는 두 개의 주제로 나누어 설명하겠다. 이는 WebtoB 리스너, Tmax 리스너에 관련된 사항들과 다른 여섯 종류의 리스너에 관련된 사항들이다. 그 이유는 WebtoB 리스너, Tmax 리스너가 다른 여섯 종류의 리스너 설정과는 좀 다른 면들이 있기 때문이다.

WebtoB와 Apache 웹서버의 해당 설정은 7.4 절을 참고하기 바란다.

모든 리스너는 WEBMain.xml의 <context-group><webserver-connection> 태그 아래에 설정된다. 이는 각 Context Group에는 단 하나의 <webserver-connection> 태그가 존재할 수 있지만 이 아래에는 여러 개의 리스너 설정이 존재할 수 있다는 의미이다. 각 Context Group은 단 한 개의 <webserver-connection> 설정을 가져야 한다.

예:

<<WEBMain.xml>>

```
<?xml version="1.0"?>
<web-container xmlns="http://www.tmaxsoft.com/xml/ns/jeus">
  ...
  <context-group>
    ...
    <webserver-connection>
      <webtob-listener>
        ...
```

```
        </webtob-listener>
        <apache-listener>
            ...
        </apache-listener>
        <http-listener>
            ...
        </http-listener>
        <tcp-listener>
            ...
        </tcp-listener>
        <secure-listener>
            ...
        </secure-listener>
        ...
    </webserver-connection>
    ...
</context-group>
...
</web-container>
```

7.3.2 Apache, HTTP, TCP, UDP, 보안 리스너 설정

Apache, HTTP, TCP, UDP, 보안 리스너의 설정은 단 한 가지의 예외를 제외하고 모두 동일하다. 그러므로 여기에서는 이들은 모두 같은 방법으로 설명한다.

다음의 XML 태그들은 다섯 종류의 리스너를 설정하는데 사용되는 것들이다.

- Apache 리스너: <apache-listener>
- HTTP 리스너: <http-listener>
- TCP 리스너: <tcp-listener>
- UDP 리스너: <udp-listener>
- 보안/SSL 리스너: <secure-listener>

위 태그들의 하위 태그(설정)들은 다음과 같다.

- **listener ID:** 리스너의 유일한 식별자이다. 이름은 WEBMain.xml 파일 내부에서 유일한 것이어야 한다. 여기에서의 ID는 “non-shared” 모드의 DB풀에서 리스너가 작업하고 있는 쓰레드 풀(Servlet 풀)을 식별하기 위해 사용된다. (5 장 참조)
- **port number:** 클라이언트(또는 Apache Server)가 연결되어 있는 포트 번호이다. HTTP를 위해서는 기본적으로 80 번을 사용하고 SSL을 위해서는 443 을 사용한다 (IANA, <http://www.iana.org/> 에서 정의). JEUS 설정에서는 어떤 포트도 기본으로 설정되어 있지 않으므로 이 태그는 모든 리스너에서 반드시 지정해 줘야 한다.
- **input buffer size:** UDP 리스너에만 해당되는 것으로 client 로부터 요청을 받을 때 수용 가능한 버퍼 크기를 설정한다. 설정하지 않으면 디폴트 값인 8192 로 설정이 되며, 0 보다 작은 값이나 문자로 잘못 설정하는 경우에도 8192 로 설정이 된다.
- **output buffer size:** Servlet 결과가 임시적으로 저장되는 내부 캐시 버퍼의 크기를 결정한다. 버퍼가 꽉 찼을 때에는 한 번에 클라이언트에게 모두 보내진다. 이 옵션은 성능향상을 위해 사용되지만 일반적으로 사용되지 않는다.
- **worker-thread pool:** 이 설정은 각 리스너의 Worker thread pool 의 크기와 행동 방식을 결정짓는다. Worker thread 가 클라이언트 요청을 처리하기 위해서 사용되며 풀 크기가 크면 클수록 더 많은 요청이 동시에 처리될 수 있다 (시스템 리소스를 많이 사용한다). 쓰레드 풀의 설정을 다음과 같다.
 - **min** 과 **max** 는 풀에서 관리하고자 하는 최소 및 최대 Worker thread 의 개수이다.
 - **Step** 설정은 풀의 크기가 증가할 때 몇 개 씩의 쓰레드가 증가할 것인지를 지정한다.
 - **maximum idle time** 은 풀 내에 존재하던 쓰레드가 제거되기 전까지의 사용되고 있지 않은 시간을 지정한다(결과로 시스템 자원은 늘어난다).

각 Worker thread pool 은 *request wait queue* 를 가지고 있다. 이 큐는 실제 가용한 Worker thread 보다 많은 요청이 들어 올 때 사용된다. 이 큐는 소켓 리스너에 의해 유지되는 낮은 레벨의

backlog 큐보다 상위 레벨의 큐이다. 다음의 두 태그는 이 큐와 관련된 것들이다.

- **max wait queue** 설정은 더 많은 Worker thread 가 풀에 생성되기 전에 얼마나 많은 요청들이 request wait queue 에 존재할 수 있는지에 대해 결정한다 (몇 개의 쓰레드가 증가될 지는 step 설정에서 정의된다).
- 마지막으로 **max queue** 설정은 큐에 대기할 수 있는 최대 요청 값을 설정한다. 이 큐가 꽉 찬 후에 더 많은 요청이 도착하면 busy 페이지가 클라이언트에게 반환된다.

각 쓰레드 풀은 장애가 발생하였을 때 취하는 액션을 정의하는 <thread-state-notify> 태그를 가지고 있다. 이것에 대해서는 다음 하위 절들을 참조한다.

- **postData read timeout** 설정은 웹서버나 Web 클라이언트에서 post-data 를 읽어 들일 때 기다릴 수 있는 최대 시간 값이다. 읽기는 request.getInputStream().read() 메소드로 한다. 단위는 천분의 일초이고 기본 값은 30000 이다.
- **Scheme** 에 설정하는 값은 javax.servlet.http.HttpServletRequest.getScheme() 메소드에 의해 반환되는 프로토콜의 값을 정의한다. 보안 리스너 또는 WebtoB 나 Apache 에서 SSL 기능을 사용할 경우에는 “https” 값이 지정되어야 한다.
- **Backlog** 는 요청이 매우 빈번하고 리스너가 더 이상 지탱하지 못할 경우에 큐잉될 수 있는 최대 클라이언트 요청 값을 지정한다. 이 값은 리스너 Server 소켓이 인스턴스화 되었을 때에 java.net.ServerSocket(int port, int backlog)에 전달된다. 큐잉된 클라이언트 연결 요청 값이 backlog 크기를 초과할 경우에는 클라이언트는 “connection refused” 메시지를 받는다.
- **busy notify page** 는 request wait queue 크기가 초과되었을 때(또는 리스너가 busy 일 때) 반환되는 static-content 파일(html 파일)의 경로이다. 이 옵션은 Apache 리스너에만 해당된다.
- **server access control** Boolean 스위치는 클라이언트 또는 웹서버 접근 제한을 켜고 끈다.

- **allowed servers list** 는 위의 server access control 이 켜져 있을 때만 사용할 수 있다. 이 리스트는 어떤 클라이언트 또는 Server 들이 이 리스너에 접근할 수 있는지 지정한다. 이 리스트에 있는 클라이언트들은 그들의 IP 주소를 이용하여 식별한다.
- TCP/UDP 리스너에서는 정식명의 “dispatcher config class” 클래스 이름을 지정하는 **dispatcher config class** 태그가 존재한다. TCP/UDP 리스너는 HTTP 프로토콜을 사용하지 않는다. 대신, TCP/UDP 리스너와 그 클라이언트 사이에 한정된 맞춤 프로토콜을 제공한다. 이 맞춤 프로토콜은 dispatcher-config-class를 이용하여 정의 된다. dispatcher-config-class 태그는 jeus.servlet.tcp.TCPDispatcherConfig로 구현한 정식 클래스 명을 정의 한다. 구현된 클래스는 반드시 JEUS_HOME\lib\application아래에 위치해야 한다. TCP/UDP 클라이언트를 서비스 하기 위해서는 jeus.servlet.tcp.TCPServlet을 구현한 클래스를 만들어야 하며 이를 web.xml에 매핑 시켜줘야 한다. 7.5 절에서 상세한 설명 하겠다.

아래에는 XML 예가 나와 있다. 이 예에서는 HTTP 리스너의 설정만 보여주고 있고 다른 리스너는 비슷한 모습으로 다음에 이어 계속 설명한다.

<<WEBMain.xml>>

```
<?xml version="1.0"?>
<web-container xmlns="http://www.tmaxsoft.com/xml/ns/jeus">
  ...
  <context-group>
    ...
    <webserver-connection>
      <http-listener>
        <listener-id>WebListener1</listener-id>
        <port>8007</port>
        <output-buffer-size>16384</output-buffer-size>
        <thread-pool>
          <min>10</min>
          <max>20</max>
          <step>2</step>
          <max-idle-time>60000</max-idle-time>
          <max-wait-queue>10</max-wait-queue>
          <max-queue>50</max-queue>
          <thread-state-notify>
```

```

        ... <!-- see next sub-section -->
        </thread-state-notify>
    </thread-pool>
    <postdata-read-timeout>
        40000
    </postdata-read-timeout>
    <scheme>http</scheme>
    <back-log>100</back-log>
    <server-access-control>
        true
    </server-access-control>
    <allowed-server>127.0.0.1</allowed-server>
</http-listener>
</webserver-connection>
...
</context-group>
...
</web-container>

```

7.3.3 자동 쓰레드 풀 관리 설정 (쓰레드 상태 통보)

<thread-state-notify> 태그는 email 통지나 자동 재시작의 실행을 위해 필요한 최소의 Worker thread 수와 관련된 “오류” 조건을 정의한다.

Thread-state-notify 태그는 웹서버 리스너의 <thread-pool> 태그 아래에 설정된다.

다음과 같은 항목들이 반드시 설정되어야 한다.

- **max thread active time** 은 블록되기 전까지 Worker thread 가 사용될 수 있는 최대 시간 값을 말한다. 이 시간은 Worker thread 가 클라이언트 요청을 서비스 할 때부터 측정된다 (Servlet 이 수행될 때). 만료는 여기서 정한 시간이 초과되었거나 또는 쓰레드가 자유롭게 되어 worker 풀로 되돌아 갈 때 된다. (쓰레드가 블록되지 않는 상황)
- **notify threshold** 는 존재할 수 있는 블록 된 쓰레드의 최대 값을 설정한다. 그 값이 초과되면, 오류 조건으로 결정되어 email 알림자를 통하여 email 통보가 보내지게 된다.

- **restart threshold** 는 존재할 수 있는 블록 된 쓰레드의 최대 값을 설정한다. 그 값이 초과되면, 오류 조건으로 결정되어 email 알림자를 통하여 email 통보가 보내지게 된다. 그 후에 Web Container 가 재시작 된다.
- **notifier ID** 는 경고 통보 또는 Container 재시작 통보를 알리기 위해 사용되는 email 알림자를 식별하기 위해 사용된다. 이 값은 email 알림자의 ID 태그의 값과 일치해야 한다. (JEUSServer 안내서의 6 장의 mail-source 설정을 참조한다.)
- **notify subject** 은 경고 email 을 보낼 때 사용된다.
- **restart subject** 은 Container 재시작을 통보할 때 사용하는 email 에 사용된다.

Email 알림자를 포함하고 있는 예가 다음에 나와 있다.

<<WEBMain.xml>>

```
<?xml version="1.0"?>
<web-container xmlns="http://www.tmaxsoft.com/xml/ns/jeus">
  ...
  <context-group>
    ...
    <webserver-connection>
      <http-listener>
        <listener-id>WebListener1</listener-id>
        <port>8007</port>
        <output-buffer-size>16384</output-buffer-size>
        <thread-pool>
          <min>10</min>
          <max>20</max>
          <step>2</step>
          <max-idle-time>60000</max-idle-time>
          <max-wait-queue>10</max-wait-queue>
          <max-queue>50</max-queue>
          <thread-state-notify>
            <max-thread-active-time>
              150000
            </max-thread-active-time>
            <notify-threshold>100</notify-threshold>
            <restart-threshold>
```

```

        18
        </restart-threshold>
        <notifier-id>MyNotifier</notifier-id>
        <notify-subject>
            JEUS WEB CONTAINER THREAD STATE WARNING
        </notify-subject>
        <restart-subject>
            JEUS WEB CONTAINER RESTART WARNING
        </restart-subject>
    </thread-state-notify>
</thread-pool>
<postdata-read-timeout>
    40000
</postdata-read-timeout>
<scheme>http</scheme>
<back-log>100</back-log>
<server-access-control>
    true
</server-access-control>
    <allowed-server>127.0.0.1</allowed-server>
</http-listener>
</webserver-connection>
    ...
</context-group>
    ...
</web-container>

```

7.3.4 WebtoB 리스너 설정

7.2.12 절에서 설명하였듯이, WebtoB리스너는 다른 종류의 리스너들과는 다르다. 그래서 여기서 별도로 설명한다.

각 WebtoB 리스너는 WEBMain.xml 파일 내 <webserver-connection> 태그 아래의 <webtob-listener> 태그에서 설정한다.

설정해야 할 것들은 다음과 같다.

- **listener ID** (7.3.2 절에서 설명한 유일한 이름).
- WebtoB Server 와 연결할 **Port**. 이 포트 값은 WebtoB 설정 파일내의 JSVPORT 설정 값과 일치해야 한다.

- **output buffer size** 는 7.3.2 절에서 설명한 것과 같다.
- **thread pool settings** 는 7.3.2 절에서 설명한 것과 같다. WebtoB에서 다른 점은 min, max 값이 WebtoB 설정 파일의 *SERVER 절에 지정된 MinProc, MaxProc 값과 일치해야 한다.

중요: min과 max 값을 설정하기 전에 7.4 절을 먼저 읽어 보길 바란다. 또한, max queue 크기 설정은 사용하지 않지만 WebtoB 설정 파일에서 MaxQCount 값으로 설정될 수 있다.

- **postdata read timeout** 설정은 7.3.2 에서 설명하였다.
- **scheme** 설정은 7.3.2 에서 설명하였다.
- **HTH count**는WebtoB 설정 파일의 *NODE 절에 지정된 HTH 프로세스 개수와 일치해야 한다. 7.4.절을 참조한다.
- **Request prefetch** : Web Container 측에서 요청을 처리하는 동안 다음 요청을 WebtoB 로부터 미리 받아 올 것인지 정한다.(Boolean 값). 이 기능이 활성화 되면, Web Container 는 각 WebtoB Worker Thread 마다 하나의 큐를 할당한다. 큐는 Worker Thread 가 현재 요청을 처리하는 동안 WebtoB 로 부터오는 요청을 버퍼링한다. 따라서 Web Container 는 WebtoB 로부터 다음 요청을 받는 시간을 단축 시킬 수 있다. 이 기능의 단점은 요청을 처리하는 도중 Web Container 에 심각한 문제가 발생하면, 큐에 축적된 요청들을 잃어버릴 수 있다는 점이다.
- **disable pipe** 설정이 "true"로 설정되어 있으면 WebtoB 와 Web Container 의 효율적인 Pipe 통신을 불가능하게 한다. WebtoB Server 와 Web Container 가 다른 머신에서 수행되고 있으면 필요한 항목이다.
- **WebtoB (IP) address.** WebContainer 와 연결을 맺을 WebtoB 의 IP 주소 항목이다.
- **registration ID** 는 WebtoB 설정파일의 *SERVER 값과 일치해야 한다. 설정하지 않은 경우 기본값으로 이 값은 Context Group 의 이름과 같은 값으로 설정된다.
- **WebtoB home directory** 는 JEUS_WSDIR 이라는 환경 변수로 정의된 WebtoB 의 기본 WebtoB Home 과 다를 경우에 설정한다 (또는 그 환경변수가 설정되어 있지 않을 때). 이 설정은 두 개 이상의 WebtoB 인

스턴스가 로컬 머신에서 운영되고 있고 Web Container 가 이 두 개 이상의 WebtoB 인스턴스에 연결될 필요가 있을 때 유용하다.

- **read timeout** 설정: WebtoB 웹서버는 지속적으로 Web Container 에게 WebtoB 의 설정 파일에 정의된 “svrchktime” 변수 값의 간격으로 “ping” 메시지를 보낸다. Web Container 는 여기에서 정의한 시간 간격으로 WebtoB 의 ping 을 체크한다. WebtoB 의 ping 이 여기에서 설정된 시간 간격 내에서 발견되지 않으면 통신 연결은 끊어진 것으로 간주되어 다시 설정된다. 그러므로, 여기의 값은 “svrchktime”보다 커야 한다.
- **reconnect timeout** 값: WebtoB Server 와 WebtoB 리스너 사이의 연결들이 운영도중 끊어질 경우가 생길 수 있다. Reconnect-timeout 은 이러한 경우에 재연결 시도를 위한 제한 시간이다. 제한된 시간이 지나면 현재의 모든 WebtoB 연결은 끊어지고, 만약 WebtoB 백업이 지정되어 있으면 Web Container 는 다음 WebtoB Server 에 Fail-over 를 시도한다. 다음 WebtoB 백업 마저 장애가 난다면 그 다음의 것을 시도한다. 최후의 WebtoB 마저 장애가 나면 주 WebtoB 리스너에 시도하게 된다. “-1”은 무한대 반복 시도, “0”은 재연결 시도를 하지 않음을 의미한다.
- **WebtoB backup Server** 설정. 이 설정은 위에서 설명한 WebtoB 리스너의 설정과 비슷하다. WebtoB backup 은 Automatic Connection Recovery 기능을 수행한다. 주 Server 가 심각한 장애 상태에 들어갔을 때 Backup WebtoB Server 에 연결하기 위한 연결 정보를 정의한다. 따라서 Backup WebtoB 의 설정은 Main WebtoB 장애시 Backup WebtoB 로 연결하기 때문에 무장애 시스템을 구현 가능하게 한다. 여기서 주의할 것은 백업 Server 설정시 쓰레드 풀에 대한 설정이 없다면 주 WebtoB 리스너의 설정에서 상속 받는다는 것이다.
- **Request wait queuing** : Web Container 측에서 요청들을 다루기 위한 새로운 worker thread 를 생성하기 전에 해당 request 를 queue 에 관리하고자 하는 설정이다..

WebtoB 리스너의 XML 예가 다음에 나와 있다.

<<WEBMain.xml>>

```
<?xml version="1.0"?>
<web-container xmlns="http://www.tmaxsoft.com/xml/ns/jeus">
. . .
```



```
<context-group>
. . .
<webserver-connection>
  <webtob-listener>
    <listener-id>WebListener2</listener-id>
    <port>9900</port>
    <output-buffer-size>16384</output-buffer-size>
    <thread-pool>
      <min>10</min>
      <max>10</max>
      <step>4</step>
      <max-idle-time>60000</max-idle-time>
      <max-wait-queue>2</max-wait-queue>
      <thread-state-notify>
        <max-thread-active-time>
          150000
        </max-thread-active-time>
        <notify-threshold>
          100
        </notify-threshold>
        <restart-threshold>
          18
        </restart-threshold>
        <notifier-id>MyNotifier</notifier-id>
        <notify-subject>
          JEUS WEB CONTAINER THREAD STATE WARNING
        </notify-subject>
        <restart-subject>
          JEUS WEB CONTAINER RESTART WARNING
        </restart-subject>
      </thread-state-notify>
    </thread-pool>
    <postdata-read-timeout>
      40000
    </postdata-read-timeout>
    <scheme>http</scheme>
    <hth-count>2</hth-count>
    <request-prefetch>true</request- prefetch>
```

```

        <disable-pipe>false</disable-pipe>
        <webtob-address>
            111.111.111.111
        </webtob-address>
        <registration-id>MyGroup</registration-id>
        <webtob-home>c:\WebtoB\</webtob-home>
        <read-timeout>120000</read-timeout>
        <reconnect-timeout>60000</reconnect-timeout>
        <webtob-backup>
            <port>9901</port>
            <output-buffer-size>
                16384
            </output-buffer-size>
            <thread-pool>
                ...
            </thread-pool>
            ...
            <webtob-address>
                111.111.111.112
            </webtob-address>
            ...
        </webtob-backup>
    </webtob-listener>
</webserver-connection>
    ...
</context-group>
    ...
</web-container>

```

다음 절에서는 어떻게 실제의 WebtoB Server가 설정되어 이 리스너에게 연결되는지 설명한다(7.4.3 절 참조). 또한 리스너 설정과 WebtoB의 설정파일 간의 관계도 알아보겠다.

7.3.5 Tmax 리스너 설정

Tmax 리스너 역시 특별한 리스너이므로 따로 설명하겠다.

각 Tmax 리스너는 WEBMain.xml 파일 내 <webserver-connection> 태그 아래의 <tmax-listener> 태그에서 설정한다.

설정해야 할 것들은 다음과 같다.

- **listener ID** (7.3.2 절에서 설명한 유일한 이름).
- **Tmax Server** 와 연결할 **Port**. 이 포트 값은 Tmax 설정 파일내의 Jeus 에 해당하는 Server 에 대한 설정 값과 일치해야 한다.
- **output buffer size** 는 7.3.2 에서 설명한 것과 같다.
- **thread pool settings** 는 7.3.2 절에서 설명한 것과 같다. Tmax 리스너의 경우도 WebtoB 리스너와 동일하게 동작한다.
- **postdata read timeout** 설정은 7.3.2 에서 설명하였다.
- **Tmax (IP) address**: Web Container 와 연결을 맺을 Tmax Server 의 IP 주소항목이다.
- **reconnect timeout** 값: Tmax Server 와 Tmax 리스너 사이의 연결 중 일부가 운영도중 끊어질 경우가 생길 수 있다. Reconnect-timeout 은 이러한 경우에 재연결 시도를 위한 제한 시간이다. 역시 WebtoB 와 동일하나 Tmax 리스너의 경우는 Backup 을 하나만 설정할 수 있다는 것이 차이점이다.
- **Tmax server group name** : Tmax 리스너를 설정하여 Tmax 와 연결할 때에 Tmax 에 어떤 서버와 연결할 것인지를 설정해야 한다. Tmax Server Group Name 은 연결하고자 하는 서버가 속해 있는 그룹을 설정한다.
- **Tmax server name** : 이 설정은 실제로 Tmax 리스너가 연결할 Tmax 의 서버 이름을 설정한다. 이 설정은 반드시 해야 하는 설정이다.
- **Tmax backup address** : 이 설정은 Tmax Server 에 장애가 발생했을 때 backup 으로 동작하는 server 의 address 이다. WebtoB Backup Server 설정과 동일하나 Tmax 리스너의 경우는 address 와 port 만 설정하고 나머지는 primary server 의 설정을 그대로 가져간다.
- **Tmax backup port** : Tmax Server 에 장애가 발생했을 때 backup 으로 동작하는 server 에 연결할 port 번호를 설정한다. Tmax backup address 가 설정되어 있다면 Tmax backup port 도 함께 설정되어야 한다.

Tmax 리스너의 XML 예가 다음에 나와 있다.

<<WEBMain.xml>>

```
<?xml version="1.0"?>
<web-container xmlns="http://www.tmaxsoft.com/xml/ns/jeus">
  ...
  <context-group>
    ...
    <webserver-connection>
      <tmax-listener>
        <listener-id>WebListener3</listener-id>
        <port>9901</port>
        <output-buffer-size>16384</output-buffer-size>
        <thread-pool>
          <min>10</min>
          <max>10</max>
          <step>4</step>
          <max-idle-time>60000</max-idle-time>
          <max-wait-queue>2</max-wait-queue>
          <thread-state-notify>
            <max-thread-active-time>
              150000
            </max-thread-active-time>
            <notify-threshold>100</notify-threshold>
            <restart-threshold>
              18
            </restart-threshold>
            <notifier-id>MyNotifier</notifier-id>
            <notify-subject>
              JEUS WEB CONTAINER THREAD STATE WARNING
            </notify-subject>
            <restart-subject>
              JEUS WEB CONTAINER RESTART WARNING
            </restart-subject>
          </thread-state-notify>
        </thread-pool>
        <postdata-read-timeout>
          40000
        </postdata-read-timeout>
      </tmax-listener>
    </webserver-connection>
  </context-group>
</web-container>
```

```
<tmax-address>111.111.111.111</tmax-address>
<reconnect-timeout>60000</reconnect-time>
<server-group-name>TmaxGroup</server-group-name>
<server-name>SVR1</server-name>
<tmax-backup-address>
    111.111.111.112
</tmax-backup-address>
<tmax-backup-port>9902</tmax-backup-port>
</tmax-listener>
</webserver-connection>
...
</context-group>
...
</web-container>
```

7.3.6 웹 관리자를 통한 리스너 설정

웹 관리자를 사용하여 Context Group 의 Web 리스너를 설정하고자 한다면 JEUS 웹 관리자 안내서의 웹 서버 리스너 장을 참고한다.

7.3.7 결론

지금까지 Context Group 에 해당 리스너들을 WEBMain.xml 에 추가하는 것을 보았다.

다음에는 WebtoB 와 Apache 리스너를 사용하기 위해 어떻게 앞 단의 웹서버들을 설정하는지에 대하여 설명하겠다.

7.4 리스너 연동과 클러스터링을 위한 웹서버 설정

7.4.1 소개

이 절에서는 어떻게 Apache 웹서버와 WebtoB 를 설정하여 해당 리스너와 연동할 수 있는지에 대해 설명하겠다.

여기서는 여러 개의 웹서버와 리스너들이 서로 연결되어 클러스터를 형성하는 상황 및 클러스터를 설정하는 방법에 대해 살펴보도록 하겠다(클러스터링이 무엇인지는 7.2.5 절에서 이미 살펴보았다).

앞에서 언급했듯이, 어떻게 Session이 관리되는지는 설명하지 않겠다. 이에 대한 설명은 8장에서 다시 다룬다.

먼저 Apache 웹서버를 살펴보자.

7.4.2 Apache 웹서버 설정과 클러스터링

Apache 웹서버를 Web Container 앞단에서 사용하기 전에 다음과 같은 것들이 선행되어야 한다.

1. Container 리스너와 Apache Server 간의 통신을 위하여 JServ 라이브러리를 설치한다.
2. WEBMain.xml 에서 Apache 리스너를 설정한다.(앞 절에서 설명했다)
3. jeus_apache.conf 파일을 생성하고 편집한다.
4. Apache 웹서버의 httpd.conf 파일에 연동 구절을 삽입한다.

각 과정을 아래에서 좀 더 자세히 알아보자.

JServ 라이브러리 설치하기

Apache 웹서버를 Web Container 의 앞 단에서 사용하기 위해서는 “mod_jserv”라는 모듈을 Apache 설치 모듈에 추가해 줘야 한다. “mod_jserv” 모듈은 Server 와 Container 간의 통신 프로토콜을 구현한 것이다. 이 프로토콜은 Apache JServ Protocol 1.2(AJP 1.2)이다.

다음은 JServ 라이브러리를 Unix 환경에서 설치하는 방법이다.

```
. /configure \  
--prefix=/usr/local/apache  
--activate-module=src/modules/jserv/libjserv.a  
make  
make install
```

libjserv.a 는 JServ 라이브러리의 아카이브이고 “prefix”는 실행파일들이 저장되는 디렉토리들이다.

Apache AJP1.3 프로토콜 사용

기본적인 Apache listener 는 위에서 기술했듯이 AJP 1.2 protocol 을 사용한다. 그러나 AJP 1.3 protocol 을 사용하기를 원한다면 “mod_jserv” 대신 “mod_jk”

module 을 설치하고 WEBMain.xml 파일에 <apache-listener>대신에 <ajp13-listener>를 사용한다.

WEBMain.xml 설정

<apache-listener> 태그를 원하는 Context Group 에 추가한다. 이를 어떻게 하는지는 앞에서 설명하였다. WEBMain.xml 에서 포트 번호를 주의하기 바란다. 이 포트 번호는 jeus_apache.conf 에서도 지정되어야 한다.

다음은 그 예이다.

<<WEBMain.xml>>

```
<?xml version="1.0"?>
<web-container xmlns="http://www.tmaxsoft.com/xml/ns/jeus">
  ...
  <context-group>
    ...
    <webserver-connection>
      <apache-listener>
        <listener-id>WebListener3</listener-id>
        <port>8887</port>
        <thread-pool>
          <min>10</min>
          <max>20</max>
          <step>2</step>
          <max-idle-time>60000</max-idle-time>
          <max-wait-queue>10</max-wait-queue>
          <max-queue>50</max-queue>
        </thread-pool>
        <postdata-read-timeout>
          40000
        </postdata-read-timeout>
        <scheme>http</scheme>
        <back-log>100</back-log>
        <busy-notify-page>
          c:\web\errorpages\busypage.html
        </busy-notify-page>
        <server-access-control>
          true
        </server-access-control>
      </apache-listener>
    </webserver-connection>
  </context-group>
</web-container>
```

```

        </server-access-control>
        <allowed-server>127.0.0.1</allowed-server>
    </apache-listener>
</webserver-connection>

    ...
</context-group>

    ...
</web-container>

```

위의 예에서는 8887 번 포트가 설정되어 있고 server access control 이 설정되어 있으며 127.0.0.1(="localhost", 두 번째 굵은줄)에서만 연결을 받는다. 이는 Apache 웹서버는 Web Container 가 있는 같은 머신에서만 실행되어야 한다는 것을 의미한다.

Jeus_apache.conf 설정

jeus_apache.conf 파일은 JServ 모듈을 위한 설정 파일이다. 이는 일상적으로 Apache 웹서버의 conf 디렉토리에 위치시킨다 (즉, “/home/jeus/apache/conf /jeus_apache.conf”).

다음은 그 예이다.

```

<<jeus_apache.conf>>

#LoadModule jeus_module modules/ApacheModuleJServ.dll
<IfModule mod_jserv.c>
    ApJServDefaultHost localhost
    ApJServDefaultPort 8887
    ApJServMount /examples /examples
    ApJServMount /test /test
</IfModule>

```

위의 설정파일에서 첫 번째 굵은 글씨는 JEUS 가 동작하는 서버의 IP 주소를 명시하고, 두 번째 것은 Apache Server 가 연결해야 할 포트 번호를 명시한다 (이것이 앞에서 언급한 Apache 리스너의 포트 번호이다).

세 번째 것은 HTTP request URL을 Web Container의 Context Group 의 요청 URL로 바인딩 시킨다. (포맷은 다음과 같다. “ApJServMount URL ContextGroupURL”) 위의 예에서 세 번째 굵은 글씨 줄은 도메인 이름 후의 “/examples”를 포함하는 모든 HTTP 요청은 “/examples” 요청 URL Context

Group으로 전달되어야 한다는 것을 의미한다. (6 장의 Context Group 과 URL 요청 경로에 대한 설명을 참조한다)

참고: jeus_apache.conf 의 첫 줄은 윈도우 시스템에서만 사용된다. 이 줄은 ApacheModuleJServ.dll 이 로드 되어야 한다는 것을 의미한다. 이 예에서는 Unix 환경에서 사용되도록 이 줄이 주석처리 되어 있지만 윈도우 환경에서는 주석을 없애도록 한다 (“#” 문자를 없앤다).

고정된 URL 경로를 이용하는 대신 파일 확장자를 redirection의 기반으로 사용할 수도 있다. 예를 들어, 일반적으로 “jsp”를 포함하는 HTTP요청은 항상 Web Container로 보낸다 (예, www.foo.com/examples/hello.jsp). 이렇게 하기 위해서는 다음과 같은 설정을 jeus_apache.conf에 해야 한다.

<<jeus_apache.conf>>

```
#LoadModule jeus_module modules/ApacheModuleJServ.dll
<IfModule mod_jserv.c>
    ApJServDefaultHost localhost
    ApJServDefaultPort 8887
    AddType application/jsp .jsp
    AddHandler jserv-servlet .jsp
    ApJServMount /examples /jeus_dependent
    ApJServMount /test /jeus_dependent
</IfModule>
```

위의 예에서는 “AddType”을 이용하여 MIME-type 을 지정하고 해당 확장자 (“.jsp”)를 지정한다. 두 번째 굵은 글씨는 이 파일 확장자에 핸들러(“jserv-servlet”이어야 한다)를 지정한다. 이렇게 함으로서 “jsp”를 확장자를 포함하는 모든 요청은 Web Container 에게 전달된다.

마지막으로 주의해야 할 것은 파일 확장자 기반의 redirection 이 사용될 때에는 ContextGroup URL 경로가 항상 “jeus_dependent”로 설정되어야 한다는 것이다 (마지막 예에서 세 번째 굵은 글씨를 참고 한다).

httpd.conf 설정

httpd.conf 파일에는 JServ 모듈을 이용하기 위해 다음 줄을 추가 시켜줘야 한다. 실제 경로는 환경마다 다르므로 적절하게 설정하기 바란다.

```
Include /home/jeus/apache/conf/jeus_apache.conf
```

7.4.3 WebtoB 웹서버 설정과 클러스터링

아래에서는 두 개의 예를 통하여 간단한 WebtoB 클러스터 환경을 구성해 보겠다. 각 Container 가 각각의 WebtoB Server 를 가지는 간단한 평행구조 설정과 각 Context Group 과 Container 가 클러스터 내의 모든 WebtoB Server 와 연결되는 설정이다.

간단한 구성의 경우

첫 번째의 간단한 예에서는 두 개의 WebtoB Server가 각각 두개씩의 전용 Web Container에 각각 연결된다 (총 4 개의 Container). 이 간단한 평행 구조 설정은 [그림 14]에 나타나 있다.

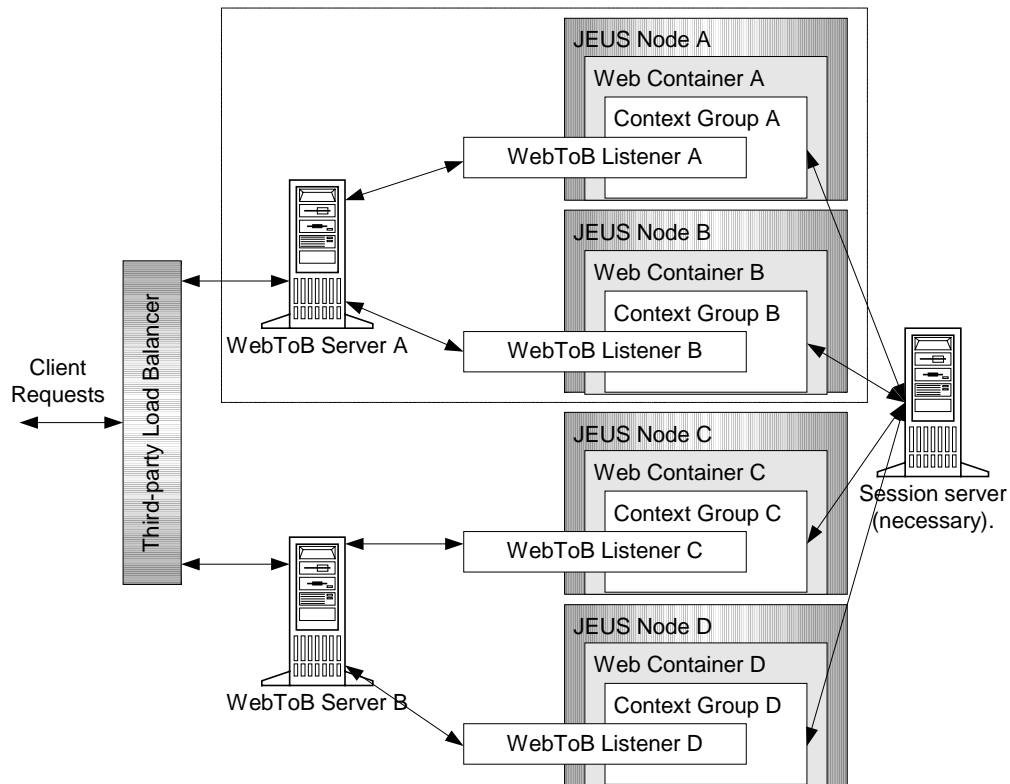


그림 14. 두 개의 WebtoB 웹서버 A, B 가 두 개의 Web Container 들(Context Group)과 연결되어 있다.

따라서, 두 개의 JEUS 노드의 Context Group 들은 해당 그룹들이 동일하도록 설정해야 한다 (즉, Context Group A, B, C, D 는 모두 동일하다).

그리고, 이런 구성에서는 Session Server를 사용해야 한다[그림 14]. 왜냐하면 간단한 Session Routing 기술은 모든 웹서버와 Context Group 들이 서로 연

결되어 있을 경우에만 작동되기 때문이다 (좀 더 복잡한 시나리오는 나중에 소개한다). 그러나, WebtoB Server가 부하 분산기로 사용되면(그림에서는 타사제품의 부하 분산기 사용되고 있음), SessionServer가 없더라도 Session Routing이 가능하다 (기본적으로 SessionServer의 사용은 권장 사항이다).

다음은 점선 안의 모습을 좀 더 자세히 보여주는 그림이다[그림 15]. 이 안에서는 두 개의 Context Group /리스너들(리스너 A, B)가 두 개의 서로 다른 Web Container에 설정되고 한 개의 WebtoB Server(A)에 연결되어 있는 모습을 보여준다.

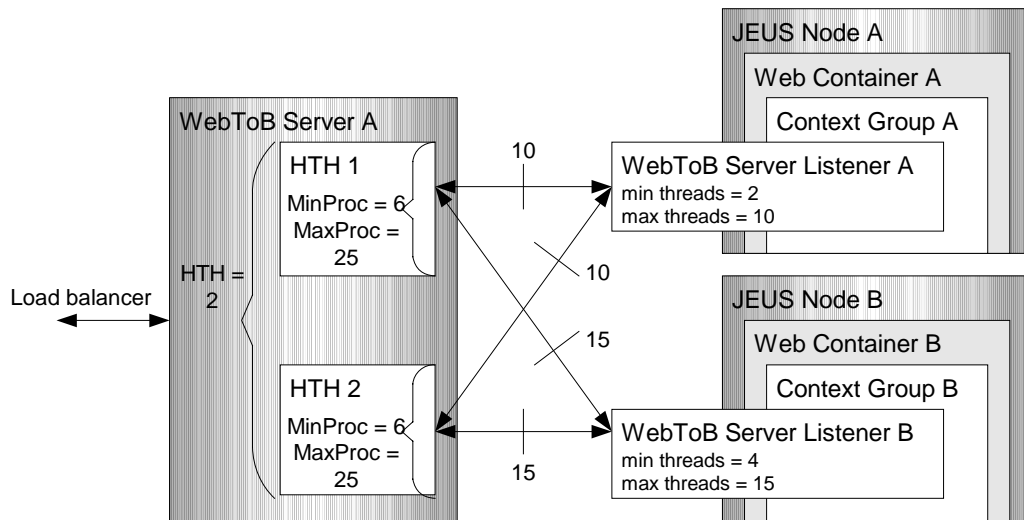


그림 15. 두 개의 WebtoB 리스너가 한 개의 WebtoB 에 연결됨.

WebtoB Server 가 HTH 프로세스를 가지고 있음을 주시해 보자. 각 HTH 프로세스는 몇 개의 하위 프로세스를 가지고 있는 Container 처럼 행동한다. 이러한 하위 프로세스는 WebtoB 리스너의 한 “Worker thread”에 연결되는데 사용된다. 이것은 HTH 의 수, HTH 의 최소 및 최대 값, 연동되는 WebtoB 리스너의 수가 리스너들의 최대 및 최소 Worker thread pool 값 설정과 관계가 있음을 나타낸다.

주의해서 설정해야 할 파라미터는 WebtoB 리스너의 worker thread pool max 값과 WebtoB Server 설정의 MaxProc 값이다. HTH 프로세스의 개수는 WebtoB Server 설정에 존재하는 HTH 프로세스 개수 값을 그대로 WebtoB 리스너의 hth-count 설정에 반영하기만 하면 된다. WebtoB Server의 MaxProc 값과 WebtoB 리스너의 thread-pool max 값은 다음과 같은 공식으로 표현될 수 있다[그림 15].

MinProc = Listener A min threads + Listener B min threads + ... + Listener X min threads setting.

MaxProc = Listener A max threads + Listener B max threads + ... + Listener X max threads setting.

WebtoB Server 의 연결은 이 값들이 제대로 설정되어 있지 않아도 된다. 그러나 성능에 악 영향을 끼칠 수 있음을 감안해야 한다.

또한 WEBMain.xml 의 <hth-count> 태그는 WebtoB 설정 파일의 *NODE 요소 설정 중 HTH 프로세스 개수와 동일해야 한다.

그리고 지금부터 어떻게 WebtoB 리스너 A와 B가 해당 WEBMain.xml에 설정되어야 하는지와 WebtoB Server A가 [그림 15]에 있는 간단한 예와 같이 설정되기 위하여 http.m 파일에 어떻게 설정되는지에 대하여 알아본다.

<<WEBMain.xml for Web container A>>

```
<?xml version="1.0"?>
<web-container xmlns="http://www.tmaxsoft.com/xml/ns/jeus">
  ...
  <context-group>
    ...
    <webserver-connection>
      <webtob-listener> <!--Listener A-->
        <listener-id>WebListenerA</listener-id>
        <port>9900</port>
        <output-buffer-size>16384</output-buffer-size>
        <thread-pool>
          <min>2</min>
          <max>10</max>
          <step>2</step>
          <max-idle-time>60000</max-idle-time>
          <max-wait-queue>5</max-wait-queue>
        </thread-pool>
        <postdata-read-timeout>
          40000</postdata-read-timeout>
        <scheme>http</scheme>
        <hth-count>2</hth-count>
        <disable-pipe>false</disable-pipe>
        <webtob-address>
```

```

        111.111.111.111
    </webtob-address>
    <registration-id>MyGroup</registration-id>
    <webtob-home>c:\WebtoB\</webtob-home>
    <read-timeout>120000</read-timeout>
    <reconnect-timeout>60000</reconnect-timeout>
    <webtob-backup>
        . . .
    </webtob-backup>
</webtob-listener>
</webserver-connection>
...
</context-group>
...
</web-container>

```

<<WEBMain.xml for Web container B>>

```

<?xml version="1.0"?>
<web-container xmlns="http://www.tmaxsoft.com/xml/ns/jeus">
    ...
    <context-group>
        ...
        <webserver-connection>
            <webtob-listener> <!--Listener B-->
                <listener-id>WebListenerB</listener-id>
                <port>9900</port>
                <output-buffer-size>16384</output-buffer-size>
                <thread-pool>
                    <min>4</min>
                    <max>15</max>
                    <step>2</step>
                    <max-idle-time>60000</max-idle-time>
                    <max-wait-queue>5</max-wait-queue>
                </thread-pool>
                <postdata-read-timeout>
                    40000</postdata-read-timeout>
                <scheme>http</scheme>
                <hth-count>2</hth-count>
            </webtob-listener>
        </webserver-connection>
    </context-group>
</web-container>

```

```

        <disable-pipe>false</disable-pipe>
        <webtob-address>
            111.111.111.111
        </webtob-address>
        <registration-id>MyGroup</registration-id>
        <webtob-home>c:\WebtoB\</webtob-home>
        <read-timeout>120000</read-timeout>
        <reconnect-timeout>60000</reconnect-timeout>
        <webtob-backup>
            . . .
        </webtob-backup>
    </webtob-listener>
</webserver-connection>
    ...
</context-group>
    ...
</web-container>

```

<<http.m of WebtoB 웹서버 A (IP address 111.111.111.111)>>

```

*NODE
foo      ...
    HTH = 2,
    JSVPORT = 9900,
    ...
*SVRGROUP
jsvg     NODENAME = foo, SVRTYPE = JSV
*SERVER
MyGroup   SVGNAME = jsvg, MinProc = 6, MaxProc = 25
*URI
uri1    Uri = "/examples/", Svrtype = JSV
uri2    Uri = "/test/", Svrtype = JSV
*EXT
jsp     MimeType = "application/jsp", SvrType = JSV

```

위의 예에서는 리스너 A 의 포트 = 리스너 B 의 포트 = WebtoB Server 의 JSVPORT(=9900). 또한 Context Group A 와 B 의 이름이 “MyGroup”으로 동일하다. 이 이름은 등록 ID 로 사용되고, http.m 의 *SERVER 요소에도 존재한다.

또한 두 WEBMain.xml 파일의 <hth-count> 태그의 값은 http.m 의 *NODE 아래의 HTH 수와 동일함을 주목한다 (값은 모든 파일에서 동일하게 2로 주어졌다).

마지막으로 중요한 것은 min 과 max 값 설정이다. 샘플 설정 파일들에서 앞에서 제시한 공식에 따라 $HTH\ MinProc = 6 = 2\ \text{쓰레드} + 4\ \text{쓰레드}$, $HTH\ MaxProc = 25 = 10\ \text{쓰레드} + 15\ \text{쓰레드}$ 로 설정되었다.

그리고 HTH의 수가 2 (http.m에서)이고 MinProc/MaxProc의 값이 HTH당 값이기 때문에 Worker thread/연결 값은 두 배가 됨을 짐작할 수 있다. 그러므로, 이 설정의 Worker thread의 최대 값은 $MaxProc * HTH\ 수 = 25 * 2 = 50$ 이 된다.

WebtoB 리스너는 자동으로 현재 HTH 프로세스 수에 맞춰진다. 그러므로 WebtoB 리스너를 설정할 때에는 HTH 프로세스가 한 개인 것처럼 생각해서 설정한다.

마지막으로, 동일한 설정 안이 또 다른 Context Group이 추가 될 때마다 적용됨을 명심하기 바란다.

복잡한 구성의 경우

이 예에서는 네 개의 Context Group을 가지고 있고 따라서 여덟 개의 WebtoB 리스너를 가진 네 개의 Web Container가 두 개의 WebtoB Server에 연결된다(즉, 모든 Container와 Context Group은 모든 WebtoB Server에 연결된다.) [그림 16].

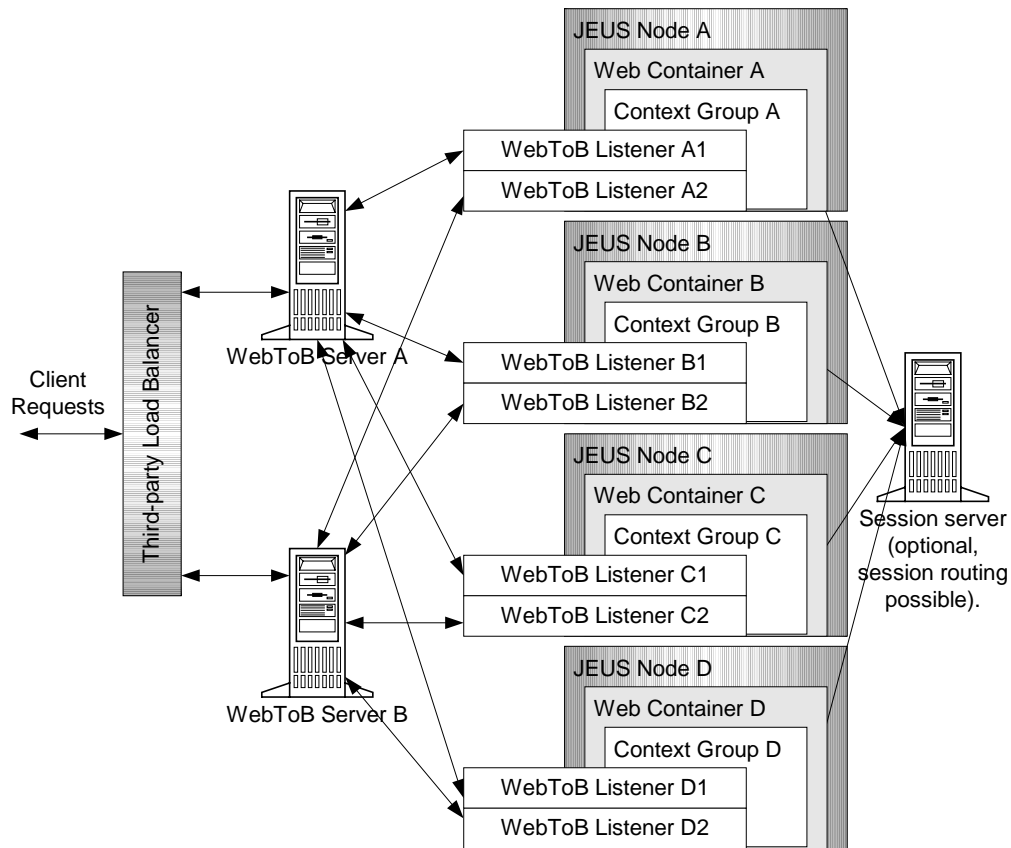


그림 16. 8 개의 리스너, 4 개의 Context Group, 4 개의 Container 가 2 개의 WebtoB Server 와 연결된 모습.

[그림 16]에서는 특별한 설정 사항을 볼 수 없다. 설정은 각 웹서버와 각 웹서버 리스너에서 위의 예에서 한 것과 동일한 방법으로 하면 되는 것이다.

한 가지 다른 점은 각 Context Group 이 모든 웹서버에 연결되어 있고 8 장에서 설명한 것과 같이 Session Routing을 이용할 수 있다는 것이다. 이런 환경에서 Session Server를 사용하는 것은 선택 사항이다. 하지만, 높은 안정성을 얻기 위해서는 Session Server의 사용을 권장한다. (Session Routing과 함께) Session Server는 Web Container의 장애 발생시 기본적으로 Session 데이터를 복구하는 기능을 가진다.

7.4.4 결과

지금까지 Apache 와 WebtoB 웹서버를 Web Container 와 연동시키기 위해 필요한 설정 방법을 살펴보았다.

7.5 TCP 리스너 사용

7.5.1 소개

TCP 리스너는 내부 HTTP 리스너가 동작하는 것과 같은 방식으로 동작한다. 그러나, 주목해야 할 점은 TCP 리스너는 HTTP 프로토콜을 지원하지 않는다는 것이다. 대신에, 개발자는 표준 HTTP 프로토콜 대신에 맞춤 통신 프로토콜을 구현할 수 있다.

“TCP”라는 개념은 이 리스너가 통신 프로토콜 구조의 낮은 레벨에서 작업할 수 있다는 것을 강조하게 된다. 즉, 이것은 어플리케이션 레벨에서 보다 전송 레벨에서 동작한다. (HTTP 리스너는 어플리케이션 레벨의 리스너이다.) 그러므로, TCP 리스너가 동작하기 위해서는 어플리케이션 레벨의 통신 프로토콜의 구현이 요구된다.

이 절에서는 어떻게 이러한 맞춤 프로토콜 제공자를 구현하는지와 어떻게 그것을 TCP 리스너와 함께 등록하는지, 그리고 어떻게 TCP 리스너에 접속하는 클라이언트를 작성하고 맞춤 프로토콜을 사용하여 통신하는지에 대하여 알아본다.

TCP 리스너는 HTTP 또는 HTTPS 프로토콜을 가지고 사용할 수 없는 통신 요구사항이 있을 경우에만 사용한다.

다음의 과정과 절차는 이 절에서 다룰 것이다.

1. 맞춤 통신 프로토콜 정의
2. dispatcher 설정 클래스 구현 (프로토콜 설정 정보)
3. TCP handler Servlet 구현 (프로토콜 구현)
4. 맞춤 프로토콜 구현을 위한 TCP 리스너 설정
5. TCP 클라이언트 구현
6. TCP 클라이언트 컴파일과 구동

TCP 리스너 API는 부록 F 에도 설명되어 있다.

먼저, 이 절에서 사용할 용어에 관하여 정의해 보자.

7.5.2 용어

다음 용어들의 의미를 알아보자.

- **프로토콜** (통신 프로토콜): TCP 클라이언트와 TCP 핸들러 사이에서 (TCP 리스너가 중간역할) 교환되는 메시지의 구조와 내용을 정의한다
- **메시지**: TCP 클라이언트와 TCP 핸들러 사이에서 교환되는 데이터. 프로토콜에서 정의한 구조에 맞아야 한다.
- **TCP 클라이언트**: TCP 리스너와 교신하며 메시지를 주고 받는 외부의 어플리케이션 (TCP 핸들러에 의해 처리된다). 메시지를 주고 받는 것은 표준 소켓을 사용한다.
- **TCP 핸들러**: TCP 리스너를 통하여 메시지를 받고 구현된 방법대로 메시지를 처리한다. TCP 핸들러는 `jeus.servlet.tcp.TCPServlet` 의 하위 클래스의 형태로 구현되고 Web Container 에 보통 Servlet 처럼 등록된다. TCP 핸들러는 TCP 프로토콜 위에 존재하는 맞춤 프로토콜의 “제공사” 또는 “구현체” 처럼 동작한다고 말할 수 있다.
- **TCP dispatcher configuration class**: TCP 설정 “dispatcher” 클래스는 `jeus.servlet.tcp.TCPDispatcherConfig`를 확장한 클래스이다. 이 클래스는 맞춤 프로토콜에 대한 정보를 TCP 리스너로 전달하여 이 non-HTTP 메시지를 해석하여 처리하도록 한다. 이 클래스는 `JEUS_HOME\lib\application` 디렉토리 아래에 놓아두고 `WEBMain.xml` 의 TCP 리스너 설정 항목 중 `dispatcher-config-class` 태그에 설정한다. (7.3 절 참조)
- **TCP listener**: 이것은 맞춤 메시지에 대한 해석과 라우팅 인프라를 제공한다. TCP 클라이언트와 TCP 핸들러 사이에서 non-HTTP 중간 매개체로 역할을 한다. 이것은 다른 HTTP 리스너와 마찬가지로 Web Container 안에 존재한다.

7.5.3 맞춤 통신 프로토콜 정의

TCP 리스너는 TCP 클라이언트와 TCP 핸들러 간에 통신되는 모든 메시지들을 두 부분으로 나누는데, 헤더와 바디가 그것이다. 일반적으로 헤더는 기본적인면서도 표준 정보를 전달하고 크기도 정해져 있다. 바디는 전송될 임의의 사용자 데이터가 포함된다. (예, HTTP 응답의 HTML 코드)

TCP 리스너가 어떻게 사용되는지 설명하기 위하여 여기에서 간단한 프로토콜(메시지 구조)을 정의하겠다. 이것은 나중에 사용할 것이다.

이 맞춤 통신 프로토콜(맞춤 메시지 구조)은 다음과 같은 내용을 가진다.

헤더 내용:

1. 헤더는 4 바이트의 **magic** 수로 시작한다. 이는 사용되는 프로토콜을 식별한다. “7777”로 설정한다.
2. 1 바이트의 **type** 필드가 magic 수 다음에 쓰인다. “0”은 “요청”을 “1”은 “응답”을 의미한다.
3. 세 번째 항목은 4 바이트의 **body length** 이다. 이 항목은 메시지의 바디 부분의 바이트 수를 기록한다. 여기의 예에는 크기가 128 바이트로 고정되어 있다.
4. 마지막 네 번째의 항목은 32 바이트의 string 으로 **service name** 을 기록한다. 여기서는 그 값으로 요청을 처리하는 TCPServlet 의 이름을 넣는다.

바디 내용:

메시지의 바디 부분은 128 바이트 길이를 가진 문자 데이터를 넣으면 된다. 여기서는 이 128 바이트 블록을 두 개의 임의의 64 바이트 텍스트 string 으로 넣는다.

이제부터 어떻게 TCP 리스너를 통하여 맞춤 프로토콜을 적용하는지에 대하여 살펴본다. 먼저 설정 “dispatcher” 클래스를 구현하고 그런다음 TCP 핸들러 클래스(TCPServlet 의 하위 클래스)를 구현한다.

7.5.4 Dispatcher 설정 클래스 구현

Dispatcher 설정 클래스는 `jeus.servlet.tcp.TCPDispatcherConfig` 클래스의 하위 클래스이다. 이 클래스의 **abstract** 메소드는 TCP 리스너가 알맞은 TCP 핸들러에게 메시지를 전달하기 위해 필요한 여러 가지 정보들을 전달되도록 구현되어야 한다. 이 메소드들은 부록 F에 설명되어 있다.

아래에 있는 코드(SampleConfig.java)는 전 절에서 정의된 프로토콜에 기반하여 어떻게 구현하였는지 보여준다. 주석을 참고하여 특정 메소드들이 어떻게 사용되었는지 살펴보라.

<<SampleConfig.java>>

```
package sample.config;

import javax.servlet.*;
import javax.servlet.http.*;
import jeus.servlet.tcp.*;

/*
   This class extends the abstract class jeus.servlet.tcp.
   TCPDispatcherConfig class. This implementation provides
   routing and handling information to the TCP listener
   according to our defined communications protocol.
*/
public class SampleConfig extends TCPDispatcherConfig {
    /*
       Any init code goes into this method. This method will
       be
       Called once after starting the TCP listener.
       We leave this method empty for our simple example.
    */
    public void init() {}

    /*
       This method returns the fixed-length of the header so
       that the TCP listener knows when to stop
       reading the header. The header length for
       our example is 41 (bytes): 4 (magic) + 1
       (type) + 4 (body length) + 32 (service name).
    */
    public int getHeaderLength() {
        return 41;
    }
}
```

```
/*
    This method must return the length of the body.
    For our example, this length is represented as an
    "int" in the header, starting at position "5"
    (see the protocol definition above).
*/
public int getBodyLength(byte[] header) {
    return getInt(header, 5);
}

/*
    This method must return the context path so that the
    request can be routed by the TCP listener to the
    context that contains the TCP handler (TCPServlet
    implementation).
    For our example, we always use the context path
    "/tcpptest".
*/
public String getContextPath(byte[] header) {
    return "/tcpptest";
}

/*
    This method must return the name (path) of the TCP
    handler(TCPServlet) relative to the context path.
    For our example, we fetch this name from the 9th position
    in the header.
*/
public String getServletPath(byte[] header) {
    return "/" + getString(header, 9, 32);
}

/*
    This method returns some path info from the header.
    It is not used in our example and thus returns
    "null".
*/
public String getPathInfo(byte[] header) {
    return null;
}
```

```

    }

    /*
       This method returns any session ID embedded in the
       header.
       It is not used in our example and thus returns
       "null".
    */
    public String getSessionId(byte[] header) {
        return null;
    }

    /*
       This method determines whether the TCP listener
       should keep the socket connection open after the TCP
       handler has delivered its response. If it returns
       "false", the connection will be dropped like in HTTP
       communications.
       If it returns "true" the connection will be kept open
       like in the Telnet or FTP protocols. For our example,
       we choose to make it persistent (connection not
       closed by the TCP listener).
    */
    public boolean isPersistentConnection() {
        return true;
    }
}

```

다음은 어떻게 TCP 핸들러를 구현하는지 살펴보자.

7.5.5 TCP 핸들러 구현

TCP 핸들러는 항상 `jeus.servlet.tcp.TCPServlet` 의 하위 클래스이다. 여기에는 항상 overridden 되는 `abstract void service(TCPServletRequest req, TCPServletResponse res)` 메소드가 존재한다.

이 서비스 메소드는 맞춤 프로토콜에 준하는 메시지를 처리할 수 있도록 구현되어야 한다. 메시지는 `TCPServletRequest` 객체에 전달되고 TCP 핸들러로부터 전달된 응답은 `TCPServletResponse` 객체의 `output stream` 으로 출력된다.

아래의 샘플은 여기서 만든 맞춤 프로토콜에 준하는 메시지를 처리하는 TCP 핸들러의 구현 코드이다. 이는 다소 직설적인 구현으로 TCPServletRequest 에서 헤더와 바디가 꺼내어져 오고 System.out 에 헤더와 바디의 각 항목들이 출력 된다. 그 후에 service() 메소드는 두 개의 응답 메시지를 생성하고 TCPServletResponse 객체의 output stream 에 출력 한다.

<<SampleServlet.java>>

```
package sample.servlet;

import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import jeus.servlet.tcp.*;

/**
 * Sample TCPServlet implementation
 *
 * Protocol scheme:
 *
 * common header (for request and response) : total 41 byte
 *
 *   magic field: length = 4 byte, value = 7777
 *   type field : length = 1 byte, 0 : request, 1:response
 *   body length field : length = 4, value = 128
 *   service name field : length = 32
 *
 * request and response body
 *
 *   message1 field : length = 64
 *   message2 field : length = 64
 *
 */

public class SampleServlet extends TCPServlet {

    public void init(ServletConfig config) throws
        ServletException {
    }
}
```

```
    public void service(TCPServletRequest req,
        TCPServletResponse res) throws ServletException,
        IOException {
        byte[] header = req.getHeader();
        byte[] body = req.getBody();

        String encoding = res.getCharacterEncoding();
        if (encoding == null)
            encoding = "euc-kr";

        DataInputStream in = new DataInputStream(new
            ByteArrayInputStream(header));
        int magic = in.readInt();
        System.out.println("[SampleServlet] received magic = " +
            magic);

        byte type = (byte)in.read();
        System.out.println("[SampleServlet] received type = " +
            type);

        int len = in.readInt();
        System.out.println("[SampleServlet] received body length
            = " + len);

        byte[] svcname = new byte[32];
        in.readFully(svcname);
        System.out.println("[SampleServlet] received service name
            = " + (new String(svcname)).trim());

        String rcvmsg = null;
        rcvmsg = (new String(body, 0, 64)).trim();
        System.out.println("[SampleServlet] received msg1 = " +
            rcvmsg);
        //rcvmsg = (new String(body, 64)).trim();
        try {
            rcvmsg = (new String(body, 64, 64, encoding)).trim();
        } catch (Exception e){}
```



```
System.out.println("[SampleServlet] received msg2 = " +
rcvmsg);

String msg1 = "test response";
String msg2 = "test response2";

byte[] result1 = null;
byte[] result2 = null;
if (encoding != null) {
    try {
        result1 = msg1.getBytes(encoding);
        result2 = msg2.getBytes(encoding);
    } catch (UnsupportedEncodingException uee) {
        result1 = msg1.getBytes();
        result2 = msg2.getBytes();
    }
} else {
    result1 = msg1.getBytes();
    result2 = msg2.getBytes();
}

header[4] = (byte)1; // mark as response

ServletOutputStream out = res.getOutputStream();
out.write(header);

byte[] buf1 = new byte[64];
System.arraycopy(result1, 0, buf1, 0, result1.length);
out.write(buf1);

byte[] buf2 = new byte[64];
System.arraycopy(result2, 0, buf2, 0, result2.length);
out.write(buf2);

out.flush();
//out.close();
}

public void destroy() {
```

```
}
}
```

이제는 위의 코드들을 기반으로 하여 TCP 리스너가 설정되는 것에 대하여 알아보자. (SampleConfig.java 와 SampleServlet.java).

7.5.6 맞춤 프로토콜 코드를 위한 TCP 리스너 설정

위에서 구현한 것을 기반으로 하여 TCP 리스너를 설정해 보자

1. SampleConfig.java 와 SampleServlet.java 를 컴파일 한다.

```
C:\tcptest>j avac -classpath d:\jeus\lib\system\jeus.jar SampleConfig.java SampleServlet.java
```

2. SampleConfig.class 파일을 JEUS_HOME\lib\application\sample\config 디렉토리에 넣는다.
3. SampleServlet.class를 JEUS_HOME/webhome/app_home/tcpcontext/WEB-INF/classes/sample/servlet” 디렉토리에 넣는다. (“tcpcontext” 디렉토리 이름을 알맞은 것으로 대체한다.). Context Group에 대한 설명은 6장을 Context 등록은 9장을 참고 한다)
4. 간단한 web.xml 을 생성하고 SampleServlet 에 대한 정보를 아래와 같이 넣는다.

<<web.xml>>

```
<?xml version="1.0"?>
<web-app xmlns="http://www.tmaxsoft.com/xml/ns/jeus">
  <display-name>TCPHandlerApp</display-name>
  <distributable/>
  <servlet>
    <servlet-name>sample</servlet-name>
    <servlet-class>
      sample.servlet.SampleServlet
    </servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>sample</servlet-name>
    <url-pattern>/sample</url-pattern>
  </servlet-mapping>
</web-app>
```

위의 `web.xml` 파일에서 어떻게 TCP 핸들러인 `sample.servlet.SampleServlet` 을 간략하게 “sample”이라는 Servlet 이름과 URL 경로인 “/sample”로 매핑하는지 살펴보자. 이 맞춤 프로토콜 구현 코드를 이용하려는 TCP 클라이언트는 헤더에 “sample”이라는 서비스 이름을 명시한다. (앞의 프로토콜 정의 참고)

서비스 이름은 TCP 리스너에 의해 호출되는 `SampleConfig` 클래스의 `getServletPath()`에 의해 추출된다. 여기서는 클라이언트가 헤더에 “sample”이라고 지정해 주었고, TCP 리스너는 “sample”서비스 이름을 `web.xml`에 정의된 `sample.servlet.SampleServlet` `TCPServlet` 클래스를 통하여 매핑하게 된다.

5. `web.xml` 파일을 `JEUS_HOME/webhome/app_home/tcpcontext/WEB-INF` 디렉토리 안에 넣는다.
6. `JEUS_HOME/config/<node name>/<node name>_servlet_<engine name>` 디렉토리의 `WEBMain.xml` 을 수정한다. `WEBMain.xml` 파일은 다음 정보들을 포함해야 한다. 새로운 Context Group (여기서는 “TCPGroup”), TCP 리스너의 설정. 아래의 예는 여기서의 예가 동작하기 위한 `WEBMain.xml`의 설정을 보여주고 있다.

`<<WEBMain.xml>>`

```
<?xml version="1.0"?>
<web-container xmlns="http://www.tmaxsoft.com/xml/ns/jeus">
. . .
<context-group>
  <group-name>TCPGroup</group-name>
  <webserver-connection>
    <tcp-listener>
      <listener-id>TCPListener1</listener-id>
      <port>5555</port>
      <thread-pool>
        <min>25</min>
        <max>30</max>
        <step>2</step>
        <max-idle-time>1000</max-idle-time>
      </thread-pool>
      <dispatcher-config-class>
        sample.config.SampleConfig
      </dispatcher-config-class>
```

```

        </tcp-listener>
    </webserver-connection>
    <group-docbase>tcpcontextgroup</group-docbase>
</context-group>
. . .
</web-container>

```

TCP 리스너 포트 (5555), sample.config.SampleConfig 을 사용하기 위한 dispatcher 클래스의 설정, 앞에서 추가한 새로운 Context Group 의 이름(“tcpcontextgroup”)을 잘 살펴보라.

7. JEUS_HOME/config/<node name> 디렉토리의 JEUSMain.xml 을 수정한다. JEUSMain.xml 파일은 새로운 Context (“TCPSample”) 정보들을 포함해야 한다.

<<JEUSMain.xml>>

```

<?xml version="1.0"?>
<jeus-system xmlns="http://www.tmaxsoft.com/xml/ns/jeus">
    <node>
        <name>tmax1</name>
        . . .
    </node>
    <application>
        <name>TCPSample</name>
        <path>
            /home/miere/jeus50/webhome/app_home/tcpcontext
        </path>
        <deployment-target>
            <target>
                <engine-container-name>
                    tmax1_container1
                </engine-container-name>
                <web-context-group>
                    <name>MyGroup</name>
                </web-context-group>
            </target>
        </deployment-target>
        <deployment-type>COMPONENT</deployment-type>
        <web-component />
    </application>

```

```
</jeus-system>
```

TCPContext 의 공식적인 이름도 “TCPSample”이라고 주었다. 다음 과정에서는 이 이름의 용도를 잘 살펴보자.

8. 마지막으로, 이 TCP Context 를 위해 deployment descriptor 를 만들어야 한다. “jeus-web-dd.xml”이라는 파일을 만들고 JEUS_HOME/webhome/app_home/tcpcontext/WEB-INF 에 넣는다.

```
<<jeus-web-dd.xml>>
```

```
<?xml version="1.0"?>
<jeus-web-dd xmlns="http://www.tmaxsoft.com/xml/ns/jeus">
    <context-path>/tcptest</context-path>
    <docbase>tcpcontext</docbase>
</jeus-web-dd>
```

위의 예에서 어떻게 Context 경로를 SampleConfig.java 의 getContextPath() 메소드에 의해 반환되는 값과 일치하도록 설정하는지 유의하여 살펴보기 바란다(“/tcptest”).

참고: 3 부터 7 의 과정은 Web Context와 deploy에 대한 내용을 다루는 9 장에서 자세히 다룬다.

9. 위의 과정이 모두 끝나면 모든 파일들을 저장하고 4 장에서 설명한 것과 같이 Web Container를 재시작 한다.
10. TCPGroup Context Group , TCPSample Context 와 “sample” TCPServlet 의 deploy 를 확인하기 위하여 위와 같이 설정을 하면, TCP 리스너에 연결되는 TCP 클라이언트의 구현과 구동에 대하여 알아보고, SampleServlet TCP 핸들러의 service() 메소드 를 호출해 보자.

7.5.7 TCP 클라이언트 구현

여기의 TCP 클라이언트는 아주 간단 명료하다. TCP리스너와 소켓 연결을 맺고 이 연결을 통하여 메시지를 보낸다. 이 메시지는 7.5.3 절에서 정의한 맞춤 통신 프로토콜에 준하는 바이트 스트림이다.

설정된 TCP 리스너는 메시지를 받을 것이고 SampleConfig 클래스에 dispatch 정보를 물을 것이다. 이 정보는 SampleServlet 구현 코드의 service() 메소드의 호출을 암시하고 있다. SampleServlet 은 클라이언트로부터 전달된 데이터를 출력하고 응답을 생성하여 전송한다. 이 응답은 클라이언트가 받아 System.out 으로 출력한다.

아래에 완전한 코드가 나와 있다.

<<Client.java>>

```
package sample.client;

import java.io.*;
import java.net.*;

public class Client {
    private String address;
    private int port;

    private int magic = 7777;
    private byte type = 0;
    private int bodyLength = 128;
    private byte[] serviceName="sample".getBytes();

    public Client(String host, int port) {
        this.address = host;
        this.port = port;
    }

    public void test() throws IOException,
        UnsupportedEncodingException {
        Socket socket = new Socket(address, port);
        DataOutputStream out = new DataOutputStream(new
            BufferedOutputStream(socket.getOutputStream()));
        DataInputStream in = new DataInputStream(new
            BufferedInputStream(socket.getInputStream()));

        out.writeInt(7777);
        out.write(type);
        out.writeInt(bodyLength);
        byte[] buf = new byte[32];
        System.arraycopy(serviceName, 0, buf, 0,
            serviceName.length);
        out.write(buf);
        byte[] msg1 = "test request".getBytes();
        byte[] msg2 = "test request2".getBytes();
```

```
buf = new byte[64];
System.arraycopy(msg1, 0, buf, 0, msg1.length);
out.write(buf);
buf = new byte[64];
System.arraycopy(msg2, 0, buf, 0, msg2.length);
out.write(buf);

out.flush();

// rx msg
int magic = in.readInt();
System.out.println("[Client] received magic = " + magic);

byte type = (byte)in.read();
System.out.println("[Client] received type = " + type);

int len = in.readInt();
System.out.println("[Client] received body length = " +
len);

byte[] svcname = new byte[32];
in.readFully(svcname);
System.out.println("[Client] received service name = " +
(new String(svcname)).trim());

byte[] body = new byte[128];
in.readFully(body);
String rcvmsg = null;
rcvmsg = (new String(body, 0, 64)).trim();
System.out.println("[Client] received msg1 = " + rcvmsg);
rcvmsg = (new String(body, 64, 64, "euc-kr")).trim();
System.out.println("[Client] received msg2 = " + rcvmsg);

out.close();
in.close();
socket.close();
}
```

```

    public static void main(String[] argv) throws Exception
    {
        Client client = new Client("localhost", 5555);
        client.test();
    }
}

```

위의 클라이언트 코드에서 프로토콜에 필요한 다양한 헤더의 필드들을 어떻게 설정하는지 살펴보자: “magic “ 수는 7777 로, “type”은 0(요청), “body length”는 128 바이트(고정 길이), “service name”은 “sample”로 되어 있다. (SampleServlet 의 이름은 web.xml 에 설정되어 있다.) 그리고 두 개의 메시지들을 생성하여 헤더 정보를 보낸 후에 TCP 리스너에게 그들을 보내게 된다.

마지막으로 hostname 을 “localhost”로 포트 번호는 “5555”로 하였다.

이제 어떻게 이 클라이언트를 컴파일하고 실행하는지 살펴보자.

7.5.8 TCP 클라이언트 컴파일과 실행

이 절에서는 TCP 리스너가 설정되어 “localhost”에 5555 포트를 이용하여 운영되고 있다고 가정한다. 이런 환경에서 다음 과정을 거쳐 TCP 클라이언트를 컴파일 하고 실행시켜보자.

1. Client.java 를 컴파일 한다.

```

C:\tcptest>j avac -classpath d:\jeus\lib\system\jeus.jar Client.java

```

2. Client.class 파일을 현재 작업 디렉토리(c:\tcptest) 아래의 “sample\client” 디렉토리 아래로 옮긴다.

3. Client.class 를 다음과 같이 실행시킨다.(모두 한 줄에)

```

C:\tcptest>j ava -classpath d:\jeus\lib\system\jeus.jar;. sample.client.Client

```

4. JEUS 관리자의 콘솔 윈도우는 다음과 같이 TCP 핸들러의 결과 값을 보여줘야 한다. (SampleServlet 클래스)

```

[SampleServlet] received magic = 7777
[SampleServlet] received type = 0
[SampleServlet] received body length = 128
[SampleServlet] received service name = sample

```



```
[SampleServlet] received msg1 = test request
[SampleServlet] received msg2 = test request2
```

참고: 만약 WEBMain.xml 의 <redirect-stdout> 태그가 “true”로 설정되어 있으면 “JEUS_HOME\logs\<node name>_servlet_<engine name>\stdout_<date>.log”로 결과가 남겨진다.

5. 다음의 것들은 클라이언트의 실행 윈도우에 남겨질 것이다.

```
[Client] received magic = 7777
[Client] received type = 1
[Client] received body length = 128
[Client] received service name = sample
[Client] received msg1 = test response
[Client] received msg2 = test response2
```

7.5.9 결론

이제 JEUS Web Container 의 TCP 리스너를 어떻게 사용하는지에 대해 알아보았다.

맞춤 프로토콜의 정의, dispatcher 설정 클래스의 구현, TCP 핸들러 클래스의 구현(TCPServlet 하위 클래스), dispatcher 클래스와 TCP 핸들러를 JEUS Web Container 에 deploy 가 주된 내용이었다.

마지막으로 TCP 리스너를 통하여 TCP 핸들러와 통신하는 TCP 클라이언트를 구현해 보고 실행도 해 보았다.

다음 절에서는 보안 리스너의 사용 방법을 알아보겠다.

7.6 보안(SSL) 리스너 사용

7.6.1 소개

JEUS Web Container 의 보안 리스너는 직접적인 클라이언트 대 Container 의 HTTPS 요청을 지원한다. HTTPS 프로토콜은 SSL(Secure Socket Layer) 기술을 사용하여 인증, 메시지 기밀, 메시지 무결성 서비스를 제공한다. 오늘날, 민감한 정보(신용카드 번호)가 교환되는 대부분의 온라인 서비스들은 SSL 을 사용한다.

JEUS 보안 리스너는 JSSE (Java Secure Socket Extensions) API 기반의 HTTPS/SSL 연결을 지원한다. 이 기능은 사용은 가능하지만, 실제 많은 운영 환경이 Web Container 보다는 웹서버에서 지원하는 SSL 기능에 의존한다.

<http://java.sun.com/products/jsse>에 SSL과 JSSE에 대한 상세한 정보가 있다.

이 절에서는 JEUS Web Container 의 보안 SSL 리스너를 설정하고 사용하는 기본적인 것에 대하여 설명한다.

7.6.2 가정

시작하기 전에, 우리는 다음과 같은 것들이 설정되어 있다고 가정한다.

- WEBMain.xml에 Servlet engine이 설정되어 시스템에 추가되어 있다 (4 장 참조).
- WEBMain.xml에 “MyGroup”이라는 Context Group 이 추가되어 있다. 이의 docbase는 “webapps”이다. (6 장 참조)
- “MyGroup”은 “TestContext”라는 Context를 가지고 “/Test” 라는 context path와 “MyContext”라는 docbase를 가진다. “MyContext” 디렉토리는 “hello.html”을 가진다(9 장 참조). 이 파일은 우리가 추가할 보안 리스너를 통하여 접근될 것이다. “MyContext\WEB-INF” 디렉토리에는 내용이 없는 web.xml파일이 존재한다.

[그림 17]는 디렉토리 구조를 보여준다.

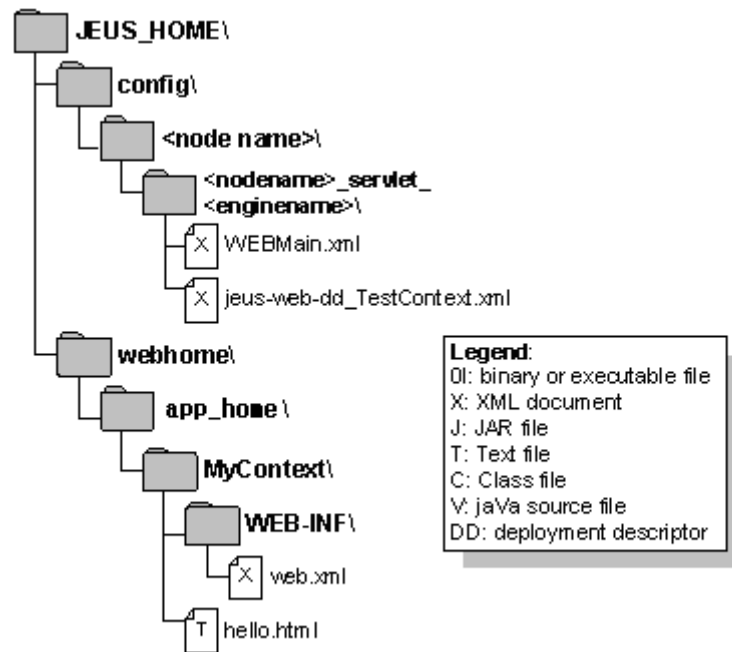


그림 17. 보안 리스너의 초기 디렉토리의 예.

파일들에는 다음과 같은 항목들이 들어 있다.

<<JEUS_HOME/config/<node name>/<Servlet engine dir>/WEBMain.xml>>

```

<?xml version="1.0"?>
<web-container xmlns="http://www.tmaxsoft.com/xml/ns/jeus">
  . . .
  <context-group>
    <group-name>MyGroup</group-name>
    . . .
    <group-docbase>webapps</group-docbase>
    . . .
  </context-group>
  . . .
</web-container>
  
```

<<JEUS_HOME/config/<node name>/<Servlet engine dir>/JEUSMain.xml>>

```

<?xml version="1.0"?>
<jeus-system xmlns="http://www.tmaxsoft.com/xml/ns/jeus">
  <node>
    <name>tmax1</name>
    . . .
  
```

```

</node>
<application>
  <name>TestContext</name>
  <path>
    /home/miere/jeus50/webhome/app_home/MyContext
  </path>
  <deployment-target>
    <target>
      <engine-container-name>
        tmax1_container1
      </engine-container-name>
      <web-context-group>
        <name>MyGroup</name>
      </web-context-group>
    </target>
  </deployment-target>
  <deployment-type>COMPONENT</deployment-type>
  <web-component/>
</application>
</jeus-system>

```

<< *JEUS_HOME*/config/<node/<eng dir>/jeus-web-dd.xml>>

```

<?xml version="1.0"?>
<jeus-web-dd xmlns="http://www.tmaxsoft.com/xml/ns/jeus">
  <context>
    <context-path>/Test</context-path>
    <docbase>MyContext</docbase>
  </context>
</jeus-web-dd>

```

<< *JEUS_HOME*/webhome/app_home/webapps/MyContext\WEB-INF/web.xml>>

```

<?xml version="1.0"?>
<web-app xmlns="http://www.tmaxsoft.com/xml/ns/jeus">
  <description>A test module.</description>
  <display-name>Test Module</display-name>
  . . .
</web-app>

```

<<JEUS_HOME/webhome/app_home/webapps/MyContext/hello.html>>

```
<html>
<head>
<title>Hello</title>
</head>
<body>
<h4><b>Hello, World</b>
(This is hello.html file)</h4>
</body>
</html>
```

참고: 여기서는 위의 모든 파일과 디렉토리를 전부 설명하지 않았다. 더 자세한 정보는 context와 관련 파일들에 대한 설명이 있는 9 장(jeus-web-dd.xml과 web.xml)을 살펴보고 오길 바란다.

Context에 대해서 알지를 못하면, 여기서 나오는 내용을 모두 이해하지 못할 수도 있다. 여기에서 설명을 목적으로 할 뿐이며, 여기 나오는 예제를 실행시키기 위해서는 위에서 제시한 설정이 필요하다.

7.6.3 보안 리스너 설정

전 절에서 주어진 예제 설정에 보안 리스너를 추가해 주면 된다.

첫 번째로 할 것은 WEBMain.xml의 “MyGroup”이라는 Context Group에 보안 리스너를 추가하는 것이다. 아래에는 그 예제이다.

<<JEUS_HOME/config/<node name>/<Servlet engine dir>/WEBMain.xml>>

```
<?xml version="1.0"?>
<web-container xmlns="http://www.tmaxsoft.com/xml/ns/jeus">
  . . .
  <context-group>
    <group-name>MyGroup</group-name>
    <webserver-connection>
      <secure-listener>
        <listener-id>SecureListener1</listener-id>
        <port>443</port>
        <output-buffer-size>8192</output-buffer-
size>
```

```

        <thread-pool>
            <min>2</min>
            <step>1</step>
            <max-idle-time>300000</max-idle-time>
            <max-wait-queue>4</max-wait-queue>
            <max-queue>-1</max-queue>
        </thread-pool>
        <postdata-read-timeout>
            30000</postdata-read-timeout>
        <scheme>https</scheme>
        <back-log>50</back-log>
        <server-access-control>
            false</server-access-control>
    </secure-listener>
</webserver-connection>
<group-docbase>webapps</group-docbase>
    . . .
</context-group>
    . . .
</web-container>

```

추가된 부분은 <webserver-connection><secure-listener> 태그이다. 7.3 절에 이 태그의 내용에 대한 상세한 설명이 나와 있다. 특히, 리스너의 포트를 443 으로 설정하는 것을 주목한다. 이 번호는 Web 브라우저에서 기본적으로 사용하는 HTTPS/SSL 포트다.

<scheme> 부분을 “https”로 설정하는 것도 주목한다.

참고: “443” 포트의 사용이 허락되지 않을 수도 있다. 이런 경우에는 “3456”과 같은 다른 포트를 선택한다. 윈도우에서는 “netstat -an”으로 사용중인 포트의 리스트를 볼 수 있다.

7.6.4 SSL Keystore 설정

SSL 프로토콜은 최소한 Server 를 클라이언트에게 인증해야만 한다. 이는 Server 쪽의 보안 리스너는 클라이언트에게 X.509 인증서를 제공할 수 있어야 한다는 것을 뜻한다.

참고: 현재, 보안 리스너는 Server 인증만 지원하고 상호인증(클라이언트)은 지원하지 않는다.

JSSE는 J2SDK 1.4에서 제공하는 “JKS” keystore를 사용하여 Server 인증서를 저장한다. 더불어, 보안 리스너는 “JEUS_HOME\config\<node name>\sslkeystore”로 저장된 JKS keystore를 기본으로 사용한다. 그러므로 우리가 “sslkeystore”파일을 자체적으로 생성할 필요가 있다. 이는 J2SDK 1.4에서 제공하는 “keytool”을 사용하여 아래와 같이 생성한다. (굵은 글씨는 사용자가 넣어야 할 것들이다.)

```
C: \>keytool -genkey -alias jeusssl -keyalg RSA -validity 7 -keystore d:\jeus\config\johan\sslkeystore
```

```
Enter keystore password: jeus123
```

```
What is your first and last name?
```

```
[Unknown]: localhost
```

```
What is the name of your organizational unit?
```

```
[Unknown]: RND
```

```
What is the name of your organization?
```

```
[Unknown]: Tmax Soft
```

```
What is the name of your City or Locality?
```

```
[Unknown]: Seoul
```

```
What is the name of your State or Province?
```

```
[Unknown]: seoul
```

```
What is the two-letter country code for this unit?
```

```
[Unknown]: KR
```

```
Is <CN=localhost, OU=RND, O=Tmax Soft, L=Seoul, ST=Kyeonggi, C=KR> correct?
```

```
[no]: yes
```

```
Enter key password for <jeusssl>
```

```
(RETURN if same as keystore password):
```

```
C: \>
```

위의 과정은 JEUS_HOME\config\johan\sslkeystore에 keystore를 생성한다 (여기서 “<node-name>”은 “johan”이고 로컬 머신의 이름과 동일하다). 개인 키와 인증서(자가 사인한 공개 키와 다른 정보들)는 만들어져 이 파일에 저장된다. 이 인증서는 보안 리스너(실제로는 JSSE를 구현한 프로그램)에 의해 사용되어 클라이언트에게 자신을 인증하게 된다.

참고: “first and last name”에 “localhost”를 넣은 이유는 인증서는 개인보다는 머신(Server)를 대상으로 발급되기 때문이다. 실제 환경에서는 IP 주소나 DNS 이름을 넣는다.

참고 2: keystore의 위치는 바꿀 수 있다. 7.6.5 에 이에 대한 정보가 나와 있다.

7.6.5 SSL Truststore 설정

SSL 상호(클라이언트) 인증이 발생할 때에는 클라이언트가 Server 에게 자신의 인증서를 보내줘야 한다. 이 인증서는 Server 의 신뢰된 인증서 리스트에 등록되어 신뢰가 이루어진다. 이 인증서들은 “truststore”라는 곳에 저장된다.

현재의 보안 리스너는 상호 인증 방법을 사용하여 설정할 수 없지만 리스너에게는 truststore 를 제공해줘야 한다. JEUS 에서 보안 리스너에 의해 사용되는 truststore 는 JEUS_HOME\config\<node name>\ssltruststore 에 위치한다.

일반적으로 이 파일은 Verisign 과 같은 공인 인증기관(CA)에서 발급된 인증서들을 포함한다. J2SDK 1.4 와 함께 제공되는 JRE_HOME\lib\security\cacerts 은 널리 알려진 CA 들로부터 발급된 신뢰된 인증서들의 긴 리스트이다. 실제 개발 환경에서는 이 파일을 JEUS_HOME\config\<node name>\ssltruststore 에 복사한다. 그 다음에 클라이언트를 위한 인증서들을 생성, 서명하여 이 truststore 에 포함시킨다. 이렇게 생성된 상호 인증서를 클라이언트들에게 배포하여 JEUS SSL 리스너와 함께 인증과정을 거치도록 한다.

여기의 간단한 예에서는 truststore 를 처음부터 생성하여 자체적으로 서명한 Server 인증서(sslkeystore 로부터)에 포함시킨다. 이것은 우리가 만든 인증서를 신뢰하겠다는 의미이다. 이를 하기 위해서는 다음의 명령어들을 실행시켜야 한다.

먼저, sslkeystore 로부터 자체적으로 서명한 인증서를 “jeusssl.cer”라는 파일에 export 한다.

```
C: \>keytool -export -alias jeusssl -keystore d:\jeus\config\johan
\sslkeystore -rfc -file jeusssl.cer
Enter keystore password: jeus123
Certificate stored in file <jeusssl.cer>
```

그 인증서를 JEUS_HOME\config\<node name>\ssltruststore 의 JEUS SSL truststore 로 import 한다.


```
C: \>keytool -import -alias jeussslcert -file c:\jeusssl.cer -keystore d:\jeus\config\johan\ssltruststore
Enter keystore password: jeus123
Owner: CN=local host, OU=RND, O=Tmax Soft, L=Seoul, ST=Kyeonggi, C=KR
Issuer: CN=local host, OU=RND, O=Tmax Soft, L=Seoul, ST=Kyeonggi, C=KR
Serial number: 3e447270
Valid from: Sat Feb 08 11:58:56 KST 2004 until: Sat Feb 15 11:58:56 KST 2004
Certificate fingerprints:
    MD5: B4: 53: FE: B6: 00: EB: FB: 0F: 04: 7F: D2: F6: FA: 9A: A0: 3B
    SHA1: DE: C8: 26: 5F: D0: 06: 9B: 3C: F8: E2: 7E: 3A: 26: B7: 78: 83: 93: 2D: 5E: 1C
Trust this certificate? [no]: yes
Certificate was added to keystore
```

C: \>

위의 과정이 완성되면 새로운 JEUS_HOME\config\<node name>\ssltruststore 파일이 생성된다. 이것은 한 개의 신뢰된 인증서를 가지고 있고 클라이언트 인증서가 상호 인증 과정에서 신뢰될 수 있을 것인지를 결정하는데 사용된다. 이렇게하면 실제 상황에서는 우리의 인증서를 제출하는 클라이언트는 자동적으로 신뢰되지 않을 것이다.

참고: 앞에서 말했듯이, 보안 리스너는 상호 인증을 지원하지 않기 때문에 위의 과정을 실제로 구현할 수는 없다.

참고 2: truststore 의 위치를 변경할 수 있다.

7.6.6 보안 리스너 속성 설정

보안 리스너를 시작하기 전에 SSL keystore 와 truststore 파일의 위치, 그리고 이 파일에 접근할 때 사용하는 암호들에 대한 정보를 제공해 줘야한다.

[표 3]은 -D 옵션을 사용하여 설정할 수 있는 JEUS특정 속성들을 보여주고 있다.

표 3. JEUS 보안 리스너 SSL -D 속성들.

JEUS 속성	의미	비고
---------	----	----

JEUS 속성	의미	비고
jeus.ssl.keystore	SSL keystore 를 포함하고 있는 파일의 경로	불필요. 설정되지 않으면 기본으로 JEUS_HOME\config\<node name>\sslkeystore 가 설정됨
jeus.ssl.keypass	SSL keystore 의 암호	설정되지 않으면 기본으로 “jeuskeypass” 가 설정됨
jeus.ssl.truststore	SSL truststore 를 포함하고 있는 파일의 경로	불필요. 설정되지 않으면 기본으로 JEUS_HOME\config\<node name>\ssltruststore 가 설정됨
jeus.ssl.trustpass	SSL truststore 의 암호.	설정되지 않으면 기본으로 “jeustrustpass” 가 설정됨

위의 속성들은 JEUSMain.xml 의 <engine-container><command-option> 태그에 지정된다.

<<JEUS_HOME\config\<node name>\JEUSMain.xml>>

```
<?xml version="1.0"?>
<jeus-system xmlns="http://www.tmaxsoft.com/xml/ns/jeus">
  <node>
    <name>[node name]</name>
    . . .
    <engine-container>
      <name>mycontainer</name>
      <command-option>
        -Djeus.ssl.keypass=jeus123
        -Djeus.ssl.trustpass=jeus123
      </command-option>
      <engine-command>
        <type>servlet</type>
        <name>engine1</name>
```

```
</engine-command>
<!-- Other engine start commands go here -->
. . .
```

command-option 값은 한 줄에 설정 하는것에 주의하고, JEUSMain.xml 의 편집이 끝나면 저장한다.

7.6.7 보안 리스너 시작하기

위의 모든 과정을 거친 후에 JEUS 노드를 재시작한다. (down 을 하고 boot 을 한다. 이 과정은 JEUS Server 안내서에 설명되어 있다.) 이렇게 하면 보안 리스너가 시작된다.

7.6.8 보안 리스너에 연결하기

로컬 호스트의 Web 브라우저를 열고 <https://localhost/Test/hello.html>를 요청해보자. (만약에 443 외의 다른 포트를 설정했다면 <https://localhost:<Port>/Test/hello.html> 와 같이 해줘야 한다.) 또한 “https”의 “s”를 유의하자.

잠시 후, 브라우저에서는 [그림 18]과 같이 **Security Alert** 대화창이 나타난다. 이 대화창은 SSL연결은 시도되었으나 보안 리스너에서 제공되는 인증서가 신뢰되지 않았다는 것을 알려준다. (왜냐하면, 브라우저가 그 자체의 truststore에 우리가 자체적으로 서명한 인증서를 포함하고 있지 않기 때문이다.) Yes버튼을 눌러 진행하고 연결과 인증서를 수용한다.



그림 18. Internet Explorer 에서의 인증서 보안 경고창.

이제 보안SSL 연결을 사용하는 보안 리스너를 통하여 “hello.html”을 받을 수 있다.[그림 19] 이 그림의 작은 자물쇠 모양은 SSL 보안 연결이 수행되고 있다는 것을 나타낸다.

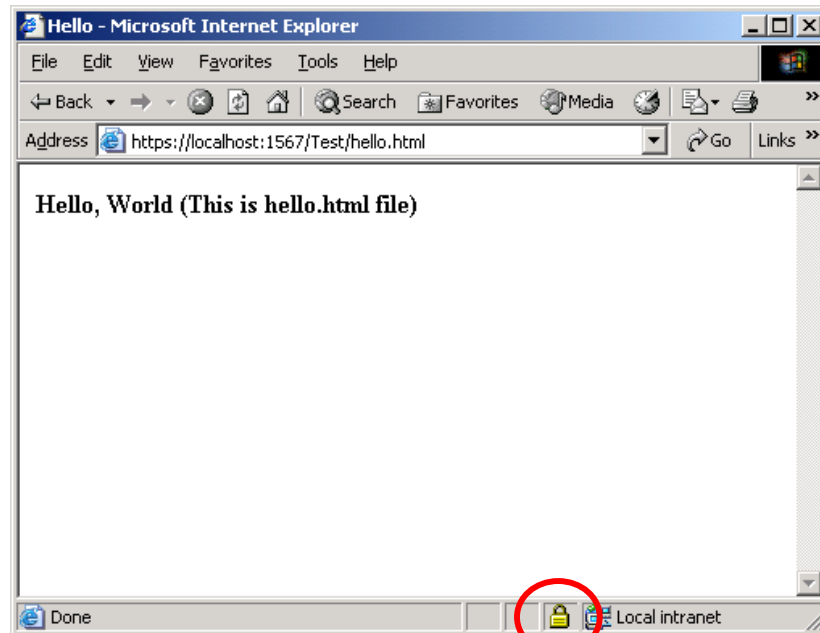


그림 19. 보안된 SSL 연결을 통하여 hello.html 이 보여지고 있다.(노란 자물쇠가 강조되어 있다).

주의: [그림 19]에서는 “1567” 포트를 이용하여 보안 리스너에 접근하고 있다. 일반적으로 “443” 포트가 사용됨을 유의 하자.

7.6.9 결론

지금까지 JEUS Web Container 에서 보안 리스너를 설정하는 방법에 대해 알아보았다.

어떻게 WEBMain.xml 에 기본 리스너를 설정하는지에 대해 알아보았고, 필수 keystore, truststore 파일을 설정하고 보안 리스너에게 위치나 이 파일들의 암호를 알려주는지, 마지막으로 보안된 SSL 연결을 통하여 HTML 파일을 가져오도록 보안 리스너를 시작하고 연결하는 방법에 대하여도 알아보았다.

SSL과 JSSE에 대한 더 상세한 정보는 <http://java.sun.com/products/jsse> 를 참조한다.

다음 절에서는 일반적인 Web Container 리스너의 튜닝 방법에 대해 알아보겠다.

7.7 Web 리스너 튜닝

최적의 성능을 내기 위하여 리스너를 설정할 때, 다음과 같은 것들을 고려한다.

- 시스템 자원을 많이 사용하거나 대기 시간을 길게하여 출력 버퍼의 크기를 증가한다.
- 일반적으로 Worker thread pool 의 min, max, step 값을 크게 부여하면 Web Container 에 많은 클라이언트가 접근할 때 풀이 좋은 성능을 가지게 된다. 시스템 메모리를 적게 사용하기 위해서는 이들의 값을 낮게 설정한다.
- Server Access Control 옵션을 꺼 놓으면 성능 개선을 기대할 수 있다.
- WebtoB 리스너에서는 앞에서 언급했던 공식에 따라 WEBMain.xml 과 http.m 의 값들을 맞춘다. 이렇게 해야 가장 좋은 성능을 기대할 수 있다.
- WebtoB 와 Web Container 사이의 통신은 항상 Pipe 통신을 이용하는 것을 권장한다. WebtoB Server 와 Web Container 가 동일 머신에 존재하면 여기에 관련된 옵션을 기본값인 “false”로 둔다. 다른 머신에 존

재하면 Pipe 통신을 할 수 없으므로, disable-pipe 태그를 “true”로 한다 (Web Container 에 의해 필요한 설정이 자동적으로 설정된다. 그러므로 설정이 꼭 필요하지는 않다).

7.8 결론

이것으로 웹서버 연결에 대한 장을 마무리 짓는다.

지금까지 Apache, WebtoB, AJP13, HTTP, HTTPS, TCP,UDP, TMAX 등 모두 여덟 종류의 웹서버 연결이 존재함을 보았다. 앞의 세 개는 앞 단의 웹서버를 통하여 클라이언트의 요청을 Container 에게 중재 시켜주는 역할을 하고, 다음 네 개는 Web Container 에 통합된 형태의 클라이언트 리스너들이고 마지막으로 Tmax 리스너는 Tmax 와의 연동을 위한 특별한 리스너이다.

각기 다른 연결 방법에 따라 어떻게 설정하는지와 Web Container 와 연결하기 위해 웹서버에 어떤 것들을 설정해 줘야 하는지도 살펴보았다. 그리고 간단히 Web Server 클러스터를 구성하는 방법도 알아보았다.

다음 장에서는 분산/클러스터된 환경의 Session Tracking 개념에 대하여 설명 하겠다.

8 Session Tracking

8.1 소개

이 장은 Session Tracking에 관련된 기본적인 개념에 대한 소개부터 시작한다. 각 주제들은 Session, Session ID, Session ID 쿠키, URL rewriting 등에 대한 정의부터 내린다.

후에는 JEUS Web Container에 어떻게 Session Tracking을 구현하는지에 대해 설명한다. 또한 보다 복잡한 분산환경에서는 어떻게 Session Tracking이 구현되고 설정되는지도 살펴보겠다(Web Server 클러스터링 환경).

이러한 클러스터에서 Session Tracking을 지원하는 메커니즘에는 세션 라우팅과 세션 서버의 사용 두 가지가 있다. 세션 서버를 사용하는 방법에는 동작 방식에 따라 중앙 집중식, 분산식 두 가지 방식이 있다.

8.2 Session Tracking의 개요

8.2.1 소개

이 절에서는 Session Tracking이 무엇이고 JEUS Web Container에서 어떻게 지원되는지에 대하여 알아본다.

여기에서 제공되는 설명은 매우 간단하기 때문에 익숙하지 않은 사용자들은 이 절의 내용을 이해하려고 시도하기 전에 Servlet과 Session Tracking에 대한 서적을 읽어보길 바란다.

[그림 20]은 Session Tracking과 Session 관리에 관련된 Web Container의 컴포넌트를 보여주고 있다.

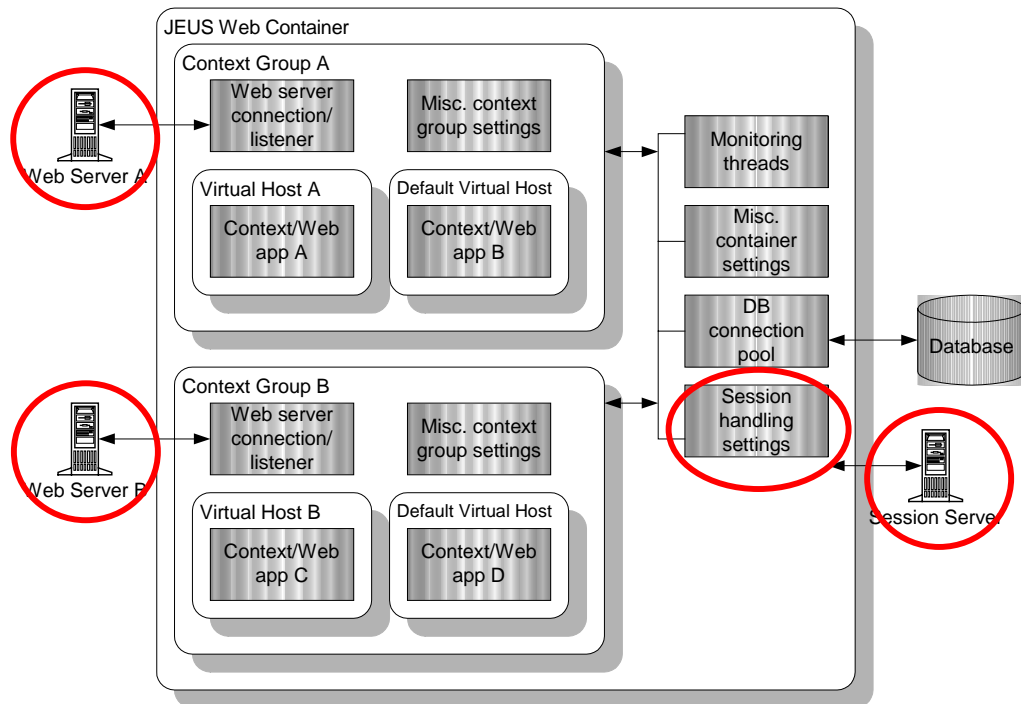


그림 20. JEUS Web Container 의 구조 중 Session 에 관련된 부분들이 강조되어 있다.(Web Server 도 Session Tracking 에 관련되어 있음을 주의하라)

8.2.2 Session Tracking 의 기초

(HTTP) Session 은 동일한 클라이언트(예를 들면 Web 브라우저)의 HTTP 요청과 연관된 일련의 작업들이다.

여러 클라이언트가 이런 요청을 표준 HTTP 를 통하여 보낼 때 Web Server 는 이 클라이언트들을 구별할 수 없다. 이 문제는 HTTP 헤더에 유일한 “client ID”가 기본적으로 들어가 있지 않기 때문이다. 따라서 Web Server 는 관련된 사용자 요청을 하나의 Session 으로 Tracking 할 수 없는 것이다. 이것은 HTTP 가 stateless(무상태)와 connectionless(무연결) 프로토콜이기 때문이다. 이것은 기본적으로 다음과 같이 작동한다.

1. 클라이언트는 Web Server 에 연결을 한다.
2. 클라이언트는 무상태 HTTP 요청을 만든다.
3. 클라이언트는 응답을 받는다.
4. HTTP 연결이 끊어진다.

Client ID 나 지속적인 Session 의 개념은 HTTP 프로토콜에 포함되어 있지 않고 따라서 Web Server 는 HTTP 연결이 끊어지거나 요청에 대한 응답 수신을 종료한 순간 각 요청에 대한 사항들을 잊어버린다(단일 요청 시 발생). 복잡한 Web 어플리케이션에서 사용자가 연달아 서로 연관된 요청을 수행하는 경우에는 사용할 수 없는 것이다.

이 문제를 극복하기 위하여, SessionID 라는 특수 string 을 각 HTTP 요청에 추가 하였다. 이 유일한 ID 는 클라이언트가 최초 요청을 할 때 필요에 따라 생성되어 클라이언트에 전달된다. 이 후 클라이언트의 요청에 이 SessionID 가 지속적으로 붙여진다. 이렇게 함으로써, Web Container 는 각 요청의 근원을 파악할 수 있기 때문에 사용자가 거래를 하는 동안에 대화성 상태 (Conversational state)를 유지할 수 있게 된다. 따라서 우리는 선천적으로 Session 의 기능이 없는 HTTP 프로토콜을 이용하여 Session 의 개념을 지원할 수 있는 것이다.

SessionID 는 쿠키에 또는 클라이언트에게 반환되는 HTML 페이지의 각 URL 링크의 파라미터로 자동으로 추가(이것을 URL rewriting 이라고 한다)된다. 또 다른 옵션은 hidden field 로 HTML 폼에 SessionID 를 지니게 하는 방법이 있다.

물론, Servlet 프로그래머의 관점에서 봤을 때, 이 논점들은 강력한 Servlet API 에 의해 모두 구현되는 것들이다.

8.2.3 Web Container 에서 지원하는 기본 Session Tracking

JEUS Web Container 는 URL rewriting 과 쿠키를 모두 지원하여 Session Tracking 을 가능하게 한다. 기본적으로 쿠키가 사용된다. Session ID 를 운반하는데 사용되는 쿠키는 Session ID 쿠키라고 한다.

Web Container 에서 한 Session 은 하나의 Servlet API 인 HttpSession 클래스의 인스턴스로 표현된다. 이 인스턴스는 SessionID 쿠키(또는 URL rewriting 의 결과로 생긴 URL 파라미터)의 SessionID 와 연관되어 있다. HttpSession 객체는 기본적으로 그 것을 생성하는 Web Container 에 존재하고 이것은 사용자에게 대한 데이터를 가지고 있다. (사용자 선호 성향이나 사용자가 전자 상거래 사이트에서 구매하고 싶은 물품의 리스트 등)

지금부터 JEUS 의 Web Container 에서 어떻게 Session ID 쿠키와 HttpSession 객체가 사용되어 Session 을 Tracking 하는지에 대하여 알아보자. URL rewriting 은 여기서 설명한 기술의 개념과 유사한 것임을 알아두자. (단, Session ID 가 HTML 페이지의 URL 링크에 포함되어 있다는 것이 별도의 쿠키 헤더에 담겨진다는 것과 다르다.)

다음의 여섯 과정은 JEUS Web Container 가 관리하는 리소스를 요청하는 한 클라이언트가 거치는 과정을 설명하고 있다.

1. 클라이언트는 Web Server 에 초기화 요청을 한다.
2. Web Server 는 Web Container 에게 요청을 전달한다.
3. Web Container 는 그 클라이언트를 위한 HttpSession 객체와 SessionID 쿠키를 생성한다. 이 ID 는 같은 클라이언트가 지속적인 요청을 할 때 메모리에서 생성한 HttpSession 객체를 가져오기 위해 사용된다.
4. 응답 데이터와 SessionID 쿠키는 Web Server 에 전달된다.
5. Session ID 쿠키는 클라이언트의 Web 브라우저에 응답과 함께 전달된다. HTTP 연결은 이제 끊겼다.
6. SessionID 를 포함하는 SessionID 쿠키는 클라이언트의 Web 브라우저에 저장된다.

위의 과정은 [그림 21]와 같이 묘사될 수도 있다.

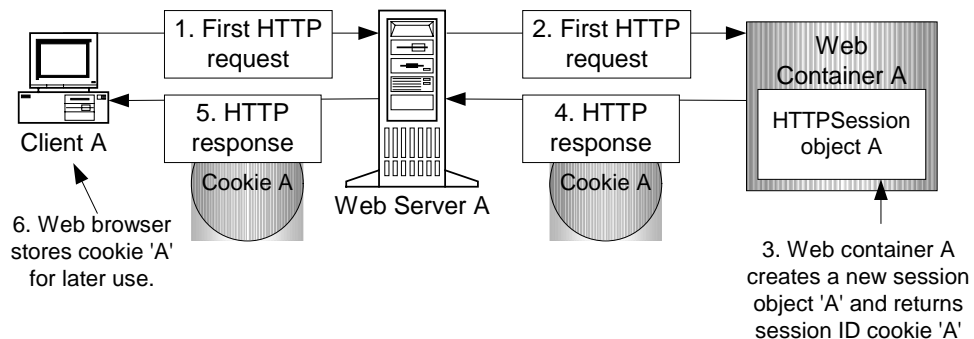


그림 21. 한 개의 Web Container 가 한 개의 Session ID 쿠키를 사용한 세션 초기화.

이제 클라이언트는 SessionID 쿠키를 지니고 있고, 같은 Web Container 에 또 다른 요청을 할 때 쿠키를 끼워보낼 수 있다. Web Container 는 클라이언트를 인식할 수 있고(쿠키의 SessionID 를 보고), 클라이언트의 HttpSession 객체를 가져올 수 있다. 이 과정이 다음에 설명되어 있다.

1. 클라이언트는 같은 Web Server 에게 또 다른 요청을 보낸다. 이 번에는 전에 받은 Session ID 쿠키를 Web 브라우저에서 받아 요청에 첨부한다.

2. Web Server 는 요청을 받아 처음의 요청과 같이 같은 Web Container 에 요청(쿠키도 포함하여)을 전달한다.
3. Web Container 는 요청과 SessionID 쿠키를 받는다. Session 쿠키에서 발견된 Session ID 에 해당하는 HttpSession 객체를 자신의 메모리 영역에서 찾아온다. Web Container 는 그 HttpSession 데이터를 사용하여 요청을 처리한다.
4. 응답 데이터와 SessionID 쿠키는 Web Server 에 전달된다.
5. HTTP 응답이 Web 브라우저에게 전달된다. 그리고 HTTP 연결은 끊어진다. 이 응답에 쿠키가 같이 전달될 필요는 없다. 쿠키는 처음 연결이 생성될 때만 응답에 포함되어 전달되면 된다.

위의 과정이 [그림 22]에 묘사되어 있다.

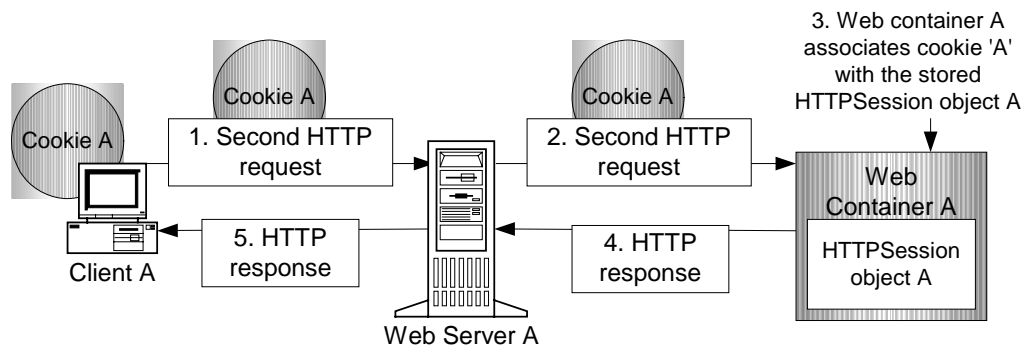


그림 22. 요청 파라미터로 저장된 SessionID 쿠키를 사용하여 같은 Web Container 에 두 번째 요청을 보내는 과정.

8.2.4 클러스터 환경에서 Session Tracking

전 절에서는 단 하나의 클라이언트, 하나의 Web Server, 하나의 Web Container가 연계된 간단한 상황에서 Session Tracking이 이루어지는 것을 살펴보았다. 그러나, 실제 운영 환경에서는 이러한 간단한 구조는 충분하지 않은 경우가 많다. 많은 양의 사용자 요청을 수용하기 위해서는 부하 분산과 Web Server 클러스터링이 구현되어야 한다. 어떻게 클러스터가 구성되는지는 7장에서 자세히 다뤘다.

Web Server 클러스터에 있어서 Session Tracking 메커니즘을 구성하고 설정할 때에는 특별한 주의가 필요하다. 분산된 클러스터에서 Session을 관리할 때에는 세 가지의 주요 사항들이 쟁점이 된다.

1. SessionID 쿠키를 가지는 요청을 처음 요청 했던 Web Container 에게 어떻게 전달되게 할까?
2. Session 으로 제한적인 요청을 모든 Web Container 가 처리할 수 있게 하기 위해서 어떻게 한 Container 에서 생성된 HttpSession 객체를 다른 Web Container 에서도 사용할 수 있게 할까?
3. 내부 또는 외부적의 장애로 Web Container 가 다운되었을 때 어떻게 Session 데이터를 백업할까?

첫 번째 논점은 세션 라우팅이라는 기능으로 대처 가능하고 나머지 두 논점들은 세션 서버로 대처 가능하다. 지금부터 이 세 가지의 문제와 두 가지 해결 방법에 대하여 설명하겠다.

참고: 위의 1, 2 번은 근본적으로 같은 문제이지만 두 가지 방법으로 해결한다.

8.2.5 세션 라우팅

세션 라우팅(Session Routing)은 클러스터된 환경에서 Client ID 뿐만 아니라 Web Container ID 와 함께 Session ID 쿠키로 꼬리표를 붙이기 위해 사용된다. Container ID 는 SessionID 쿠키를 처음 생성했던 Container 에 의해 추가되고, Web Server 가 원래의 Web Container 에게 요청을 전달해 줄 수 있게 한다.

예를 들면, 두 개의 Web Container A, B 가 하나의 Web Server 에 연결된 클라이언트 요청 상황을 고려해보자.

1. 클라이언트는 Web Server 에게 초기 요청을 한다.
2. Web Server 는 임의적으로 요청 전달을 위해 한 개의 Web Container 를 선택한다. 여기서는 Web Container A 가 선택되었다.
3. 요청은 Web Container A 로 전달된다.
4. Web Container A 는 HttpSession 객체를 생성하고 응답과 함께 SessionID 쿠키를 돌려 보낸다. 이 ID 는 다음에 오는 같은 클라이언트의 요청을 처리할 HttpSession 객체를 메모리에서 불러 올 때 사용된다.
5. Web Container 는 응답을 하고 Web Server 에 SessionID 쿠키가 반환된다.

6. SessionID 쿠키는 응답과 함께 클라이언트의 Web 브라우저로 전달된다. HTTP 연결은 이제 끊겼다.

이 과정은 [그림 23]에 설명되어 있다.

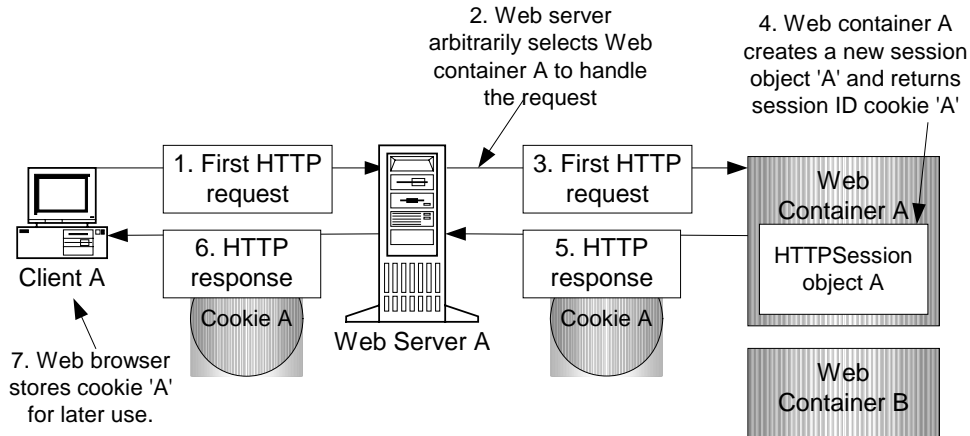


그림 23. SessionID 쿠키를 이용하여 두 개의 Web Container 와 Session 을 시작한다.

지금까지 모든 것이 정상적이었다. 심각한 문제는 두 번째 요청이 같은 클라이언트로부터 만들어질 때 생기기 시작한다.

1. 클라이언트는 또 다른 요청을 같은 Web Server 에 한다. 이 번에는 이전에 반환된 SessionID 쿠키가 Web 브라우저로부터 가져와져서 요청에 붙여진다.
2. Web Server 는 요청을 받아 들인다. Web Server 는 임의로 두 개 중 한 개의 Web Container 를 선택한다. 이번에는 Web Container B 가 선택되었다.
3. 그 요청과 SessionID 쿠키는 Web Container B 에 전달된다.
4. Web Container B 는 요청과 SessionID 쿠키를 받는다. 이것은 자신의 메모리 영역에서 쿠키에 대응하는 HttpSession 을 가져오려고 시도한다. Web Container B 는 그러한 HttpSession 객체를 가져올 수 없다. 왜냐하면, Web ContainerB 에는 그런 HttpSession 객체가 없기 때문이다. 따라서 Web ContainerB 는 클라이언트 Session 을 유지할 수 없고 새로운 Session 을 강제로 생성하거나 또는 오류 메시지를 반환한다. (물론 두 옵션 모두 권장하지 않는다.)

이 문제의 시나리오는 [그림 24]에 표현되어 있다.

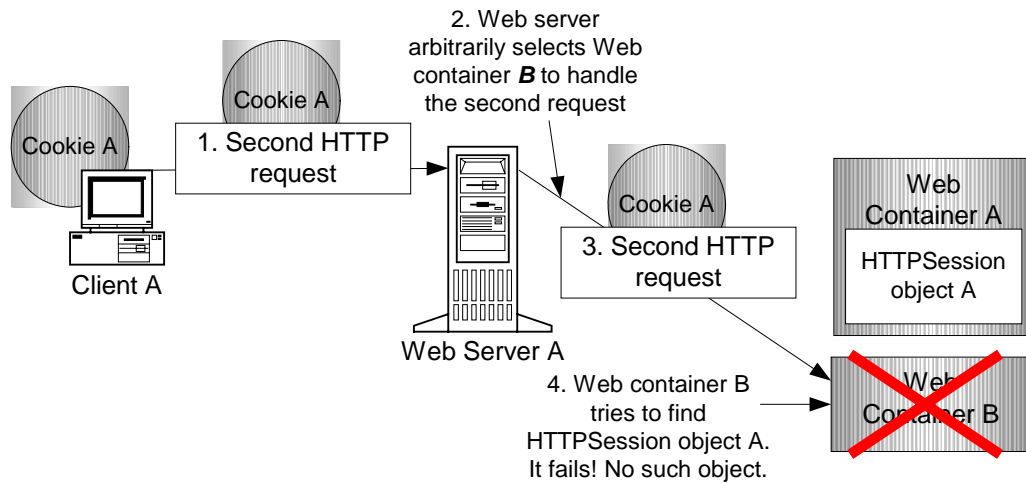


그림 24. 세션 라우팅 없이는 클라이언트 Session 은 이 클러스터링 환경에서 사라진다.

이 문제를 해결하기 위하여 처음에 HttpSession 객체를 생성했던 동일한 Web Container 에게 Session 으로 제한된 요청을 올바르게 라우팅 해 줄 방법이 필요하다. 이것은 SessionID 쿠키에게 추가의 Web Container ID 를 부여함으로써 해결된다.

1. 클라이언트는 Web Server 에 초기 요청을 맺는다.
2. Web Server 는 임의의 Web Container 를 선택한다. 여기서는 Web Container A 가 선택된다.
3. 요청은 Web Container A 에 전달된다.
4. Web Container A 는 HttpSession, SessionID 쿠키를 생성하고 Web Container ID(CID)를 쿠키에 삽입한다.
5. Web Server 에 응답과 SessionID 쿠키를 반환한다.
6. Web 브라우저에 응답과 SessionID 쿠키를 반환한다.
7. Web ContainerID(CID)를 포함하는 SessionID 쿠키는 Web 브라우저에 저장된다.

이것은 [그림 25]에 표현되어 있다.

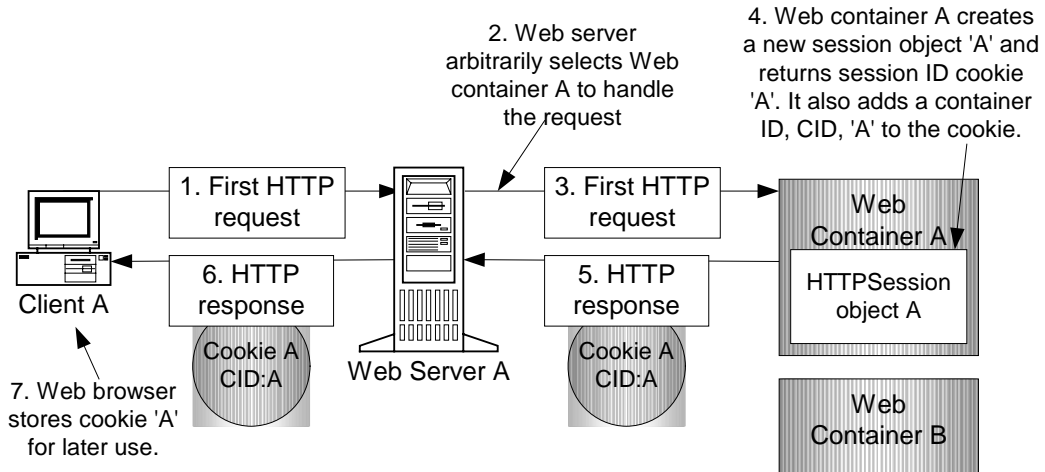


그림 25. 클라이언트의 첫 요청은 ContainerID(CID)를 가진 라우팅 기능을 내재한 Session을 반환한다.

이렇게 함으로써 어떻게 라우팅 문제가 쉽게 풀리는지 살펴보자. 두 번째 요청에 Web Server는 Session ID쿠키와 Container ID값(CID)을 찾아낸다. Web Server는 해당 HttpSession이 존재하는 원래의 Web Container로 요청을 Routing 시킨다. [그림 26]

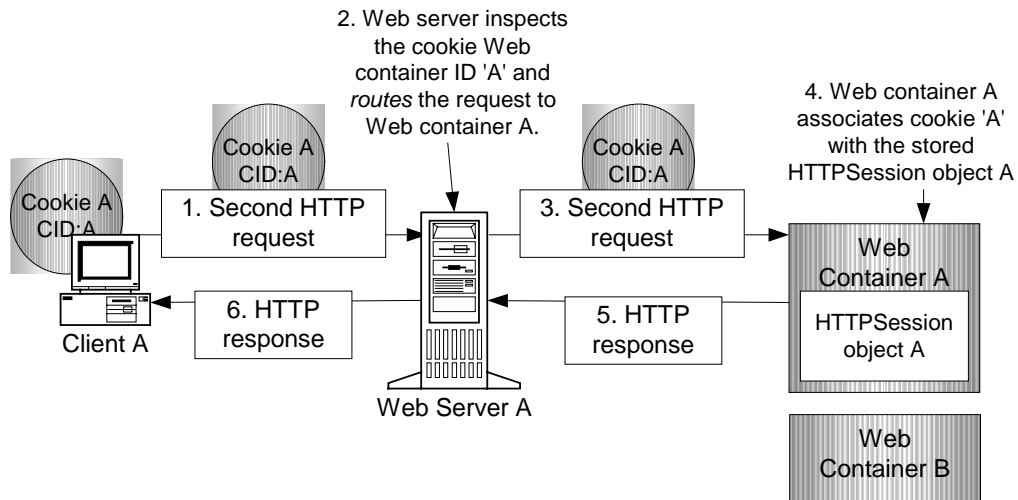


그림 26. 세션 라우팅의 작동(CID로 Web Server는 원래의 Web Container에게 요청을 Routing 시킨다.)

그렇다면 웹 서버는 표준적인 개념이 아닌데도 불구하고 어떻게 Container ID를 식별 할 수 있을까? WebtoB Web Server에게는 이 기능은 내장되어 있지만 Apache Web Server의 경우에는 8.4에서 설명하는 것과 같이 별도의 설정이 필요하다.

중요: 세션 라우팅 기능을 위해서는 모든 웹 서버가 모든 Web Container 와 연결을 맺고 한다. 왜냐하면 부하 분산기를 사용할 경우, 여러 대의 웹 서버 중에서 요청을 받은 웹 서버가 해당 Web Container 에 접속을 하지 못하면 세션이 끊길 수 있기 때문이다.

그러나 부하 분산기가 세션 라우팅을 자체적으로 지원하고 있다면, 위치럼 서로간에 모두 연결될 필요가 없다. 예를 들어 WebtoB 를 부하 분산기로 사용하고 있다면, 세션 라우팅은 클러스터링이 완벽하게 연결되어 있지 않아도 사용할 수 있다.

8.2.6 세션 서버

간단한 세션 라우팅을 사용하는 것보다 한 층 더 강력한 것이 세션 서버를 사용하는 것이다. 세션 서버를 사용할 때에는, 모든 Web Container 가 클러스터 내의 모든 Web Server 에 연결될 필요는 없다. 또한, 세션 서버는 클러스터 내의 모든 Session 데이터에게 신뢰적인 백업 기능을 제공한다. 따라서 임의의 수의 Web Container 에 장애가 발생하더라도 Session 데이터는 저장되고 다른 정상적인 Web Container 가 장애가 발생한 Web Container 의 요청을 대신 처리한다.

장애가 발생한 Web Container의 문제는 [그림 27]에 표현되어 있다. 이 시나리오에서는 Container에 존재하는 모든 사용자 Session 데이터가 소멸된다. 이런 상황은 운영 환경에서는 반드시 피해야 하는 것이다.

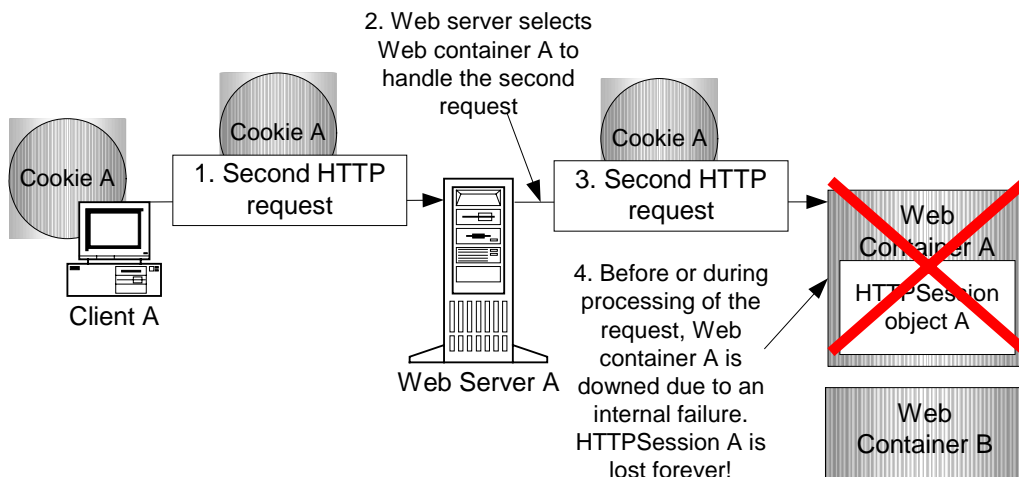


그림 27. Session ID 쿠키와 연관된 Session 데이터는 Web Container 장애로 인해 소멸되었다.

이와 같이 특정 Container 에 장애가 발생하는 상황에서도 세션을 지속시키기 위해서 세션 서버가 클러스터에 추가 되었다. 이런 세션 서버가 사용될 경우에는 클라이언트의 첫 HTTP 요청이 다음과 같은 방법으로 처리된다.

1. 클라이언트는 Web Server 에게 요청을 보낸다.
2. Web Server 는 Web Container A 를 클러스터 내에서 선택하여 요청을 처리하게 한다.
3. Web Server 는 Web Container A 에게 요청을 전달한다.
4. Web Container 는 HttpSession 객체와 SessionID 쿠키를 생성한다. 이 ID 는 다음에 요청이 같은 클라이언트로부터 왔을 때 세션 서버로부터 생성된 HttpSession 객체를 꺼내기 위해 사용된다.
5. 요청에 대한 처리가 완료되면 Web Container 는 HttpSession 객체와 SessionID 를 세션 서버에 저장한다.
6. 응답 데이터와 SessionID 쿠키는 Web Server 로 전달된다.
7. SessionID 쿠키는 응답과 함께 Web 브라우저로 전달된다. HTTP 연결이 끊긴다.
8. Web 브라우저는 이 SessionID 쿠키를 저장한다.

위의 과정이 아래의 [그림 28]에 표현되어 있다.

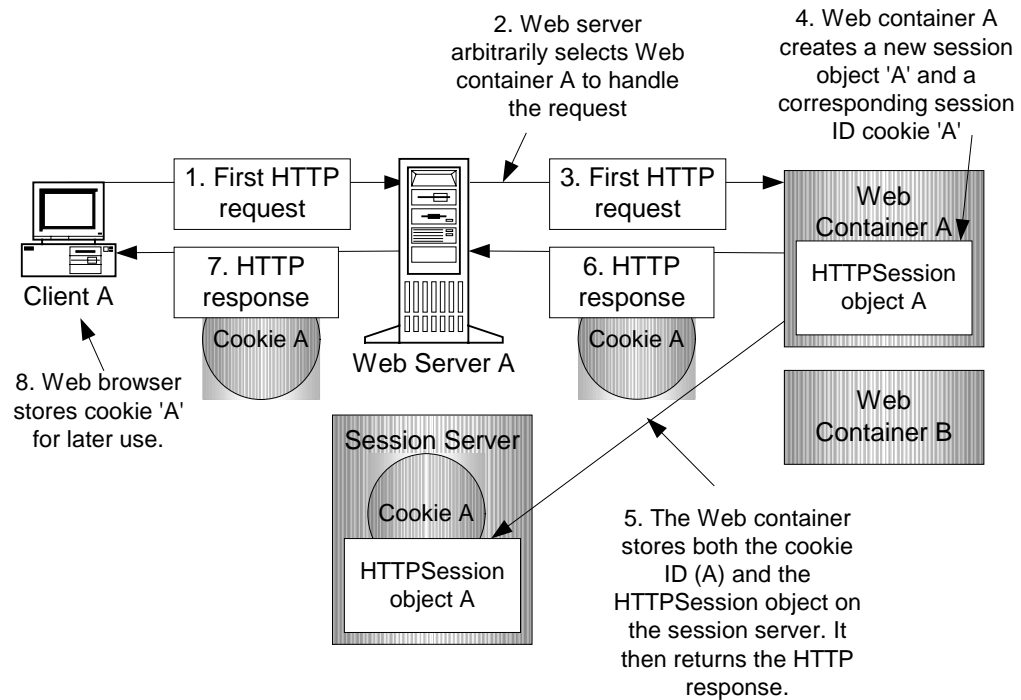


그림 28. 별도의 세션 서버에 Session 데이터와 SessionID 를 저장한다.

나중에 Web Container B가 무작위로 Web Server에 의해 클라이언트 A의 요청을 처리하도록 선택되거나 Web Container A에 장애가 발생하더라도 Session 데이터는 세션 서버에서 가져올 수 있다. [그림 29]

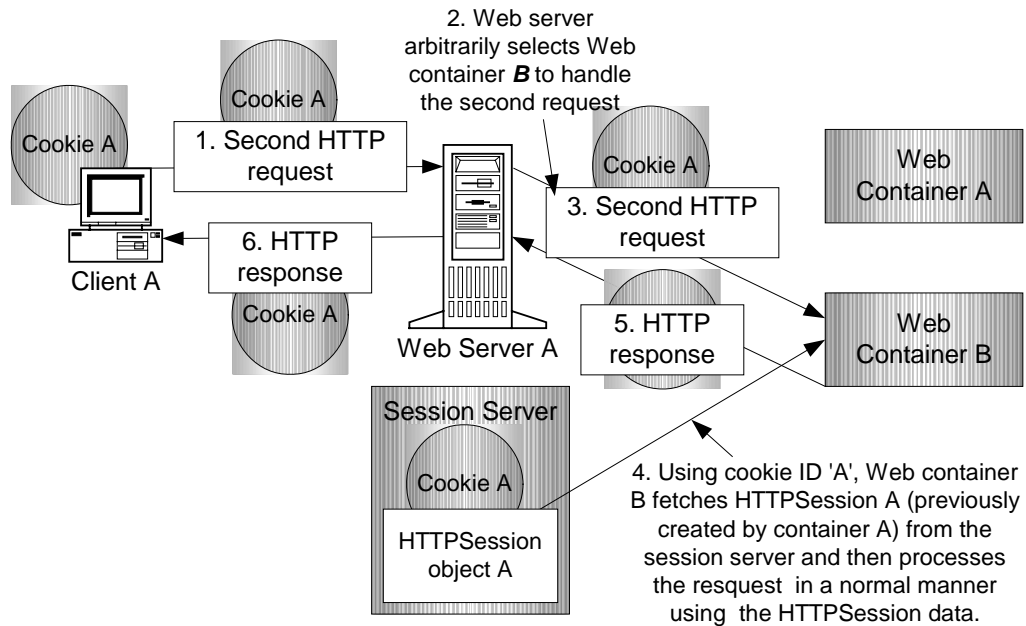


그림 29. Session 데이터는 Container의 내부 메모리에서 가져오지 않고 항상 중앙 세션 서버에서 가져온다.

게다가, Session 데이터 저장소를 좀 더 안정적으로 만들기 위하여 백업 세션 서버가 설정될 수 있다. (8.3에서 간략하게 다시 다룬다.)

실제로 세션 서버는 JEUSMain.xml에서 설정하며 Container에서는 JNDI lookup으로 접근한다. JEUS Server 안내서에 세션 서버 설정에 대한 더 많은 정보가 나와 있다.

8.2.7 Multi Session 모드

Multi Session 모드는 Web Container가 많은 요청을 수용해야 할 경우에 높은 성능을 보장 받을 수 있다. 이것은 한 개의 Web Container가 다수의 Session Manager(세션 서버의 서브 컴포넌트)를 동시에 사용함으로써 가능한 것이다. Session Manager는 SessionID를 사용하여 구성된 hash 값을 바탕으로 선택하는데, 이는 각 Session Manager가 클라이언트 Session 데이터의 일부분만을 관리한다는 것을 뜻한다.

JEUS Server 안내서에서 언급했듯이 다중 세션 모드는 JEUSMain.xml에서 설정하며, 자세한 정보는 JEUS Server 안내서를 참조하기 바란다.

8.2.8 혼합 모드

위에서 설명한 두 개의 방법이 혼합된다.

세션 라우팅 방법은 Web Container 의 Session 객체를 접근하므로 빠르다. 그러나, 문제가 발생할 경우에 Web Container 의 모든 Session 데이터는 소멸되고 복구가 불가능하게 될 것이다.

세션 서버를 사용하면 세션 서버에 모든 Session 객체가 서버에 저장되므로 특정 Web Container 에 문제가 생기더라도 Session 객체의 소멸은 없을 것이다. 특정 Web Container 에 문제가 발생하더라도 세션 서버는 Session 을 유지할 수 있다. 그러나 이 기술은 Session 이 필요한 요청이 들어왔을 경우에 Session 객체를 세션 서버로부터 가져와야 하는 부담이 있다.. 또한 HTTP Session 객체가 변경되었을 경우 세션 서버에 다시 저장되어야 한다. 그러므로, 세션 서버를 사용하는 단점은 세션 라우팅의 방법보다 처리 속도가 늦다는 것이다.

혼합 방식을 사용하면으로써 두 가지 방식의 장점을 모두 살릴 수 있다. 만약에 두 방법이 혼합되면 Session 객체는 세션 서버와 Session 객체를 생성한 Web Container 에 모두 존재한다.

혼합 방식을 사용하여 Web Container 는 필요할 때만 세션 서버에서 HTTP Session 객체를 꺼내서 변경한다. 따라서, 혼합 방식을 사용할 경우에는 사용되는 네트워크 대역폭도 세션 서버만을 사용하는 방식에 비해 거의 절반 가량 줄이고 모든 Session 데이터의 안전한 백업도 보장 받을 수 있다.

이 때문에 클러스터 환경에서는 혼합 방식의 Session 관리의 사용을 권장하는 것이다.

8.2.9 분산식 세션 서버

앞 절에서 살펴 보았듯이 세션 서버 또는 Multi Session 모드 세션 서버를 사용하면 특정 Web Container 에 장애가 발생하더라도 지속적으로 세션을 유지할 수 있는 장점이 있다. 그러나, 세션 서버를 사용하는 경우 소규모의 클러스터링 환경에서는 좋은 성능을 유지 할 수 있으나 클러스터링 규모가 커질수록 세션 서버에 부담이 가중되어 선형적인 성능 향상을 얻을 수 없는 단점이 있다. 분산식 세션 서버는 대규모 클러스터링 환경에서 성능 향상을 꾀할 수 있도록 고안된 세션 서버이다.

분산식 세션 서버란 각 Web Container 마다 세션 서버를 두는 방식을 말한다. 기본적으로 세션 라우팅 기술을 사용하며(물론 세션 라우팅이 적용되지 않는 환경에서도 이상 없이 동작한다.) 세션 서버와 마찬가지로 세션 데이터 백업을 설정할 수 있어 Web Container 에 장애가 발생하여도 지속적으로 세션을 유지 할 수 있다.

소규모 클러스터링 환경에서는 중앙 집중식 세션 서버를 사용하고 대규모 클러스터링 환경에서는 분산식 세션 서버를 사용하기를 권장한다.

분산식 세션 서버 설정은 JEUSMain.xml 를 통해서만 설정한다..

8.2.10 URL Rewriting 과 쿠키에 관하여

기본적으로 Web Container 는 클라이언트의 SessionID 를 지속되는 요청간에 유지하기 위하여 쿠키를 사용한다. 한 가지 문제점은 대부분의 Web 브라우저가 새로운 요청에 대하여 쿠키가 원래 생성된 곳의 도메인 이름과 다른 도메인 이름을 가지고 요청을 하면 SessionID 쿠키를 보내지 않는다는 것이다.

이 이유로, jeus-web-dd.xml deployment descriptor 에 특수한 태그가 존재한다. url-rewriting 태그가 그것이다. 이 태그가 사용되면, SessionID 는 Web 브라우저가 쿠키를 지원하여도 URL rewriting 을 사용하여 유지된다. 이런 방법으로, Session Tracking 은 다른 도메인 이름이 몇 개의 요청에 거쳐 불려도 작동한다.

9 장에 더 상세한 설명이 되어 있다.

8.2.11 Shared 세션 데이터

일반적으로 세션 서버 없이 세션 라우팅을 사용하는 경우, 서로 다른 컨텍스트 그룹간에 세션 데이터를 공유하는 것은 불가능하다. 그러나 이러한 것이 필요할 경우가 있는데, WEBMain.xml 에 “shared”를 설정하면, 세션 서버가 존재하지 않더라도 컨텍스트 그룹간에 세션을 공유할 수 있다. 만약 세션 서버가 존재하고 세션 라우팅을 사용하지 않는다면 이 기능은 필요 없다.

이 기능의 사용에 대한 좀더 많은 정보는 다음 절의 환경 설정을 보라.

8.2.12 결론

이상으로 간단하게 Session Tracking 에 대해서 살펴 보았고, JEUS Web Container 가 Session Tracking 을 어떻게 지원하고 있는지 알아 보았다. 여기에는 두 가지 주요 기술이 적용되는데, 이들이 세션 라우팅과 세션 서버라는 것도 알았다.

두 기술을 혼합하여 사용하여 더 좋은 효과를 만들어 낼 수 있다는 것도 알았다.

다음 절에서는 WEBMain.xml 에서 이 방법들을 설정하는 것에 대해 알아보자.

8.3 Session Tracking 설정

8.3.1 소개

이 절에서는 각 Web Container의 WEBMain.xml 설정 파일을 이용하여 어떻게 세션 라우팅, 세션 서버, 혼합 모드, 그리고 분산식 세션 서버가 설정되는 지에 대하여 설명한다.

또한 Apache Web Server는 어떻게 설정이 되어야 세션 라우팅이 지원되는 지에 대해서도 알아보겠다.

먼저 Session Tracking의 설정은 <web-container> 태그 레벨과 각 context group 태그 아래에서 설정할 수 있다는 것을 주목하기 바란다. (부록 D를 참조하라) 설정이 두 레벨에서 모두 존재하면 context group에서 설정한 것이 Container 레벨에서 한 것보다 우선한다. 이 부분에서는 Container 레벨의 설정만 설명한다.

Session Tracking은 분산식 세션 서버를 사용하는 경우를 제외하고는 <session-cluster> 태그 내에서 설정된다. 예제 템플릿은 다음과 같다.

<<WEBMain.xml>>

```
<?xml version="1.0"?>
<web-container xmlns="http://www.tmaxsoft.com/xml/ns/jeus">
    . . .
    <context-group>
        . . .
        <session-cluster>
            . . .
        </session-cluster>
        . . .
    </context-group>
    . . .
    <session-cluster>
        . . .
    </session-cluster>
    . . .
</web-container>
```

8.3.2 세션 라우팅 설정

WEBMain.xml 세션 라우팅은 매우 간단하다. <session-routing> 태그에 “true” 값만 지정한다.

예:

<<WEBMain.xml>>

```
<?xml version="1.0"?>
<web-container xmlns="http://www.tmaxsoft.com/xml/ns/jeus">
  . . .
  <session-cluster>
    <session-routing>true</session-routing>
    . . .
  </session-cluster>
  . . .
</web-container>
```

만약 세션 서버를 사용하지 않고, 세션 라우팅을 통해 서로 다른 컨텍스트 그룹간에 세션 데이터를 공유하려 한다면, <shared>태그를 “true”로 설정하면 된다(기본값은 “true”).

주의: <shared> 설정은 <context-group> 내에서 사용하지 않고 최상위 레벨인 <web-container><session-cluster> 태그에 사용해야 한다.

Example:

<<WEBMain.xml>>

```
<?xml version="1.0"?>
<web-container xmlns="http://www.tmaxsoft.com/xml/ns/jeus">
  . . .
  <session-cluster>
    <session-routing>true</session-routing>
    <shared>true</shared>
  </session-cluster>
  . . .
</web-container>
```

8.3.3 세션 서버 설정

세션 서버를 사용하고자 할 때에는 JEUSMain.xml 에 설정을 하고 JEUS 노드를 재시작 한다. 그리고, 각 Web Container 의 WEBMain.xml 에 <session-

server> 태그를 설정한다. (Web Server 설정은 세션 서버가 바로 Web Container 에 접근하기 때문에 필요하지 않다.)

일반적으로 외부의 리소스(이 경우에는 세션 서버)가 접근될 때에는 성능을 향상시키기 위하여 Web Container 에서 유지하는 Connection Pool 을 사용한다.

세션 서버 연결을 위한 설정은 다음과 같다.

- 세션 서버 **name**: JEUSMain.xml 의 <session-manager><name>에 설정된 외부 세션 서버 **name**.
- 세션 서버 **connection pool** 설정: Connection Pool 에서 유지해야 할 열린 세션 서버 연결의 최소 및 최대 값을 결정짓는다.
- **timeout settings**:
 - **connection timeout** 세션 서버로 새로운 연결이 맺어질 때까지 기다려야 하는 시간 값(풀 크기가 증가 할 때)이다.
 - **read timeout** 값은 세션 서버로부터 데이터 읽기를 시도할 때 기다려야 하는 시간 값이다.
- 외부 **backup** 세션 서버 **name**: 이 Server 는 주 세션 서버가 접근 불가능 일 때 사용된다. 이것은 JEUSMain.xml 에 설정된다.

여기에 그 예가 있다.

<<WEBMain.xml>>

```
<?xml version="1.0"?>
<web-container xmlns="http://www.tmaxsoft.com/xml/ns/jeus">
    . . .
    <session-cluster>
        <session-server>
            <server-name>session1</server-name>
            <max>10</max>
            <min>2</min>
            <step>3</step>
            <backup-server-name>
                session2
            </backup-server-name>
        </session-server>
    </session-cluster>
</web-container>
```



```

        </backup-server-name>
        <connect-timeout>120000</connect-timeout>
        <read-timeout>120000</read-timeout>
    </session-server>
</session-cluster>
. . .
</web-container>

```

8.3.4 Multi Session Mode 설정

Multi Session Mode의 설정은 위의 8.3.3 과 거의 비슷하다. 세션 서버를 JEUSMain.xml에 Multi Session Mode로 설정을 하고 JEUS 노드를 재시작한다. 그리고 WEBMain.xml의 설정을 위의 8.3.3 과 동일하게 설정한다 (JEUSMain.xml의 Multi Session Mode의 설정은 JEUS Server 안내서의 9.3.4 ‘Multi Session Mode설정’ 부분 참조).

단, <session-server><server-name>과 <session-server><backup-server-name> 이 두 태그의 의미가 Multi Session Mode 의 경우 약간 차이가 있다.

- **세션 서버 name** : Session Server 를 모니터링시 primary 세션 서버 그룹을 대표하는 display name (Description 의 의미이지 JEUSMain.xml 의 session manager name 의 설정과 관계가 없다).
- **backup 세션 서버 name**: Session Server 를 모니터링시 backup 세션 서버 그룹을 대표하는 display name (Description 의 의미이지 JEUSMain.xml 의 session manager name 의 설정과 관계가 없다).

예:

<<WEBMain.xml>>

```

<?xml version="1.0"?>
<web-container xmlns="http://www.tmaxsoft.com/xml/ns/jeus">
    . . .
    <session-cluster>
        <session-server>
            <server-name>PrimarySession</server-name>
            <max>10</max>
            <min>2</min>
            <step>3</step>
            <backup-server-name>
                BackupSession
            </backup-server-name>
        </session-server>
    </session-cluster>
</web-container>

```

```

        </backup-server-name>
        <connect-timeout>120000</connect-timeout>
        <read-timeout>120000</read-timeout>
    </session-server>
</session-cluster>
. . .
<web-container>

```

8.3.5 혼합 모드 설정

혼합 모드의 설정은 다른 작업이 필요하지 않다. 세션 라우팅이 설정되어 있는지와 세션 서버가 설정되어 있는지만 확인하면 된다.

예:

<<WEBMain.xml>>

```

<?xml version="1.0"?>
<web-container xmlns="http://www.tmaxsoft.com/xml/ns/jeus">
    . . .
    <session-cluster>
        <session-routing>true</session-routing>
        <session-server>
            <server-name>session1</server-name>
            <max>10</max>
            <min>2</min>
            <step>3</step>
            <backup-server-name>
                session2
            </backup-server-name>
            <connect-timeout>120000</connect-timeout>
            <read-timeout>120000</read-timeout>
        </session-server>
    </session-cluster>
    . . .
</web-container>

```

다시 말하지만 <session-cluster> 설정은 Container 레벨과 context group 레벨에서 모두 설정 가능하다.

8.3.6 분산식 세션 서버 설정

분산식 세션 서버 설정은 오직 JEUSMain.xml 에만 존재한다. WEBMain.xml 에는 별도의 설정이 필요 없다.

분산식 세션 서버 설정은 다른 모든 세션 Tracking 설정보다 우선한다. 즉, JEUSMain.xml 에 <session-server> 설정이 있고, WEBMain.xml 에 <session-cluster> 설정이 있다 하더라도, JEUSMain.xml 에 분산식 세션 서버를 설정하면 분산식 세션 서버를 이용한 Session Tracking 이 설정된다.

분산식 세션 서버는 모든 Web Container 에 각각의 세션 서버를 가지고 있는 형태이며 모든 설정은 JEUSMain.xml 의 session-router-config 설정을 통해 이루어진다. 분산식 세션 서버를 위한 설정들은 다음과 같다.

- 노드내의 모든 Web Container 에 적용되는 설정
 - **connect-timeout** 값은 Web Container 내에 존재하는 세션 서버 사이에 필요한 경우 연결을 맺게 되는데, 그 때 적용되는 socket connection timeout 값이다.
 - **read-timeout** 값은 세션 서버 사이에서 상대방으로부터 메시지를 받을 때 적용되는 timeout 값이다.
 - **backup-trigger** 값은 백업(local file backup or remote in-memory backup)이 설정되었을 때 백업의 빈도를 조절하는 값이다. 이 값으로 지정된 횟수만큼 세션 객체가 변경되면 백업을 수행하게 된다.
 - **check-to** 값은 백업(local file backup or remote in-memory backup)이 설정되었을 때 백업이 필요한 세션 객체가 존재 여부를 조사하는 주기를 조절하는 값이다. 이 값으로 지정된 시간(msec 단위) 주기로 백업 여부를 조사하여 백업이 필요한 객체가 존재하면 백업을 수행한다.
 - **check-level** 값은 세션 객체가 변경되어 백업이 필요한지를 판단하는 기준을 제공한다. 유효한 값은 “set”, “get”, “all”이다. 설정 값이 “set”이면 HttpSession.setAttribute(), HttpSession.removeAttribute(), HttpSession.invalidate()등의 API 를 사용하여 세션 객체를 변경한 경우에만 백업이 필요한 세션 객체로 간주한다. 설정 값이 “get”이면 HttpSession.getAttribute() API 가 호출된 세션 객체는 백업이 필요한 세션 객체로 간주한다. 설정 값이

“all”이면 Servlet/JSP 에서 HttpSessionRequest.getSession()을 호출한 경우 백업이 필요한 세션 객체로 간주한다. 기본 값은 “set”이다. 이 설정은 엔진의 성능에 많은 영향을 주는 설정이다. 되도록 기본 값으로 설정하기를 권장하며, Servlet/JSP application 에서 세션 객체를 변경할 필요가 있을 경우 반드시 HttpSession.setAttribute(), HttpSession.removeAttribute(), HttpSession.invalidate() 등의 API 를 호출하기를 권장한다.

- **default-file-db** 설정은 persistent storage(즉, 파일 시스템)로 세션 객체를 백업하기 위한 설정이다. 뒤에서 설명할 backup-session-router 설정과 병행해서 사용할 수 있다. session-router 설정 안에서도 file-db 설정을 할 수 있는데 두개의 설정이 동시에 존재하면 session-router 설정 안의 것이 우선한다.
- **session-router** 설정: 분산식 세션 클러스터링에 참여할 Web Container 를 지정하고 Web Container 내부에서 기동될 세션 서버에 대한 상세한 속성을 지정하는 설정이다. 하위 설정은 다음과 같다.
 - **servlet-engine-name** 분산식 세션 클러스터링에 참여할 Web Container 의 이름을 적는다. 이 이름은 같은 노드의 engine-container 설정에 존재하는 engine-command 내의 name 의 값으로부터 얻어온다. 단, 이 때 engine-command 내의 type 설정은 “servlet”이어야 한다.
 - **backup-session-router** in-memory backup 으로 사용할 Web Container 를 지정하는 설정이다. 하위 속성으로 servlet-engine-name, node-name 이 있어서 backup 으로 지정할 Web Container 의 engine-name 과 그 Web Container 가 속한 node 의 name 을 지정한다.
 - **file-db** persistent storage(즉, 파일 시스템)로 세션 객체를 백업하기 위한 설정이다. backup-session-router 설정과 병행해서 사용할 수 있다. default-file-db 설정도 존재하는 경우에는 file-db 설정이 우선한다.

8.3.7 웹 관리자를 통한 Session Tracking 설정

웹 관리자를 이용하여 Session Tracking 에 대한 설명은 JEUS 웹 관리자 안내서의 세션 추적 장을 참고하도록 한다.

8.3.8 결론

지금까지 Web Container 에서 세션 라우팅과 세션 서버 연결을 설정하는 방법을 알아보았다.

다음 절에서는 Apache Web Server 에서 세션 라우팅을 설정하는 방법에 대하여 알아보자.

8.4 Apache Web Server 를 위한 세션 라우팅 설정

WebtoB Web Server 는 세션 라우팅을 특수한 설정 없이 지원하지만 Apache 의 경우에는 다르다. 이 절에서는 두 개의 Web Container 로 구성되는 작은 클러스터 구조에서 Apache 가 세션 라우팅을 구현하는 방법에 대하여 설명한다.

Apache Web Server 와 세션 라우팅을 구현하기 위해서는 Server 는 서비스를 제공해야 하는 모든 Web Container 와 물리적인 연결을 맺고 있어야 한다.

아래에서는 webdev_servlet_engine1 과 webdev_servlet_engine2 라고 불리는 Web Container 에 세션 라우팅을 하기 위한 Apache JServ 설정을 보여주고 있다.

<<jeus_apache.conf>>

```
#LoadModule jeus_module modules/ApacheModuleJServ.dll
<IfModule mod_jserv.c>
    ApJServDefaultHost localhost
    ApJServDefaultPort 8887
    ApJServMount /examples balance://set1/jeus_dependent
    ApJServBalance set1 PC1
    ApJServBalance set1 PC2
    ApJServHost PC1 ajpv12://61.77.153.1:10001
    ApJServHost PC2 ajpv12://61.77.153.1:10002
    ApJServRoute webdev_servlet_engine1 PC1
    ApJServRoute webdev_servlet_engine2 PC2
    ApJServShmFile /var/log/apache/jserv_shm
</IfModule>
```

- **ApJServMount /examples balance://set1/jeus_dependent:** 마운트할 URI 와 JServ 경로를 지정한다. 많은 Web Container 가 존재하여 동일

한 URI(/examples)를 서비스할 경우 이것은 “balance://set1” 라고 표현된다. “set1”은 “ApJServBalance”라고 명명한 항목의 값이다.

- **ApJServBalance set1 PC1:** 클러스터할 Web Container 들을 지정한다. “PC1”은 “ApJServHost”라고 명명된 항목의 선택된 동일명이다. 따라서 이 태그는 ApJServHost 를 “PC1”으로 지정하고 “set1”(/examples)라고 정의되어 마운트된 URI 를 처리한다.
- **ApJServHost PC1 ajpv12://61.77.153.1:10001:** 전에 “set1”이라고 지정한 Web Container 리스너(“PC1”)의 실제 주소와 포트를 지정한다. 여기서 사용한 프로토콜은 “ajpv12” 프로토콜이다.
- **ApJServRoute webdev_servlet_engine1 PC1:** Web Container 이름을 지정하고 실제 리스너(“PC1”)에 매핑 시킨다. 실제 등록된 Web Container 이름이 입력되어야 한다. 앞에서 설명하였듯이, Session 객체를 생성할 Web Container 의 식별자는 쿠키 ID 의 마지막 부분에 삽입 되어 세션 라우팅이 가능하도록 한다. ApJServRoute 에 지정된 이름은 Web Container 가 쿠키 ID 에 삽입할 이름이 된다.
- **ApJServShmFile /var/log/apache/jserv_shm:** 이것은 Apache Server 가 세션 라우팅을 위해 내부적으로 사용하는 내용이다.

참고: Apache JServ version 1.1 또는 그 위의 버전이 안정적인 세션 라우팅 동작을 위해 권장된다.

Apache Web Server 의 설정에 관한 정보는 Apache 의 문서를 찾아보도록 한다.

8.5 Session Tracking 튜닝

클러스터된 환경에서 최적의 성능을 내기 위해 다음과 같이 한다.

- 항상 세션 라우팅과 세션 서버를 혼합하여 사용하도록 노력한다. 이는 좋은 성능과 안정적인 운영을 보장한다.
- 세션 서버 연결 설정은 min, max, step 값을 증가시켜 시스템 리소스를 더 사용하는 조건으로 성능을 향상 시킬 수 있다.
- Web Server 연결들이 잘 설정되고 튜닝 되어 있어야 한다.

- 사용자의 요청이 폭주하는 사이트에서는 Multi Session 모드 세션 서버나 분산식 세션 서버를 사용할 것을 권장한다. 이는 JEUSMain.xml에서 설정하고 JEUS Server 안내서에 설명되어 있다.

8.6 결론

지금까지 Session Tracking에 대한 설명을 하였다.

Web Server 클러스터에서는 세션 라우팅을 하는 것과 세션 서버를 사용하는 두 가지의 주된 방법으로 세션을 라우팅 한다는 것을 알았다.

어떻게 두 방법이 설정되는지 보았고 Web Server에서는 어떤 변경을 해줘야 하는지도 살펴보았다.

마지막으로 Session Tracking 메커니즘의 성능 튜닝도 다뤘었다.

다음 장에서는 가장 중심인 Web 모듈(Context, Web Application)에 대해 알아보겠다.

9 Context(웹 어플리케이션)

9.1 소개

이 장에서는 JEUS Web Container 내의 J2EE Web 모듈(Web application, Context)를 조립, 등록, 또는 제어하는 모든 방법에 대하여 알아본다.

이 장과 나머지 장들에서는 “Web Application”과 “context”의 의미가 완벽하게 같은 의미로 통하지만 후자를 주로 사용할 것이다.

앞에서도 그랬듯이, 여기서는 JEUS Web Container에 관련된 사항들만 이야기하고 J2EE 표준에 대한 사항들은 이야기하지 않겠다. J2EE에 관련된 개념이 더 필요하다면 J2EE 1.4. 스펙, Servlet 2.4 스펙 JSP 2.0 스펙을 참고하라. 이들은 모두 <http://sun.java.com>에서 찾을 수 있다. 그러나, 가장 좋은 방법은 Servlet 프로그래밍에 대한 좋은 참고서를 보는 것이다.

9.2 Web Context 의 개요

9.2.1 소개

이 절에서는 context 에 관련된 기본 개념과 구조를 살펴본다.

[그림 30]은 context/Web application에 관련된 Web Container의 그림이다.

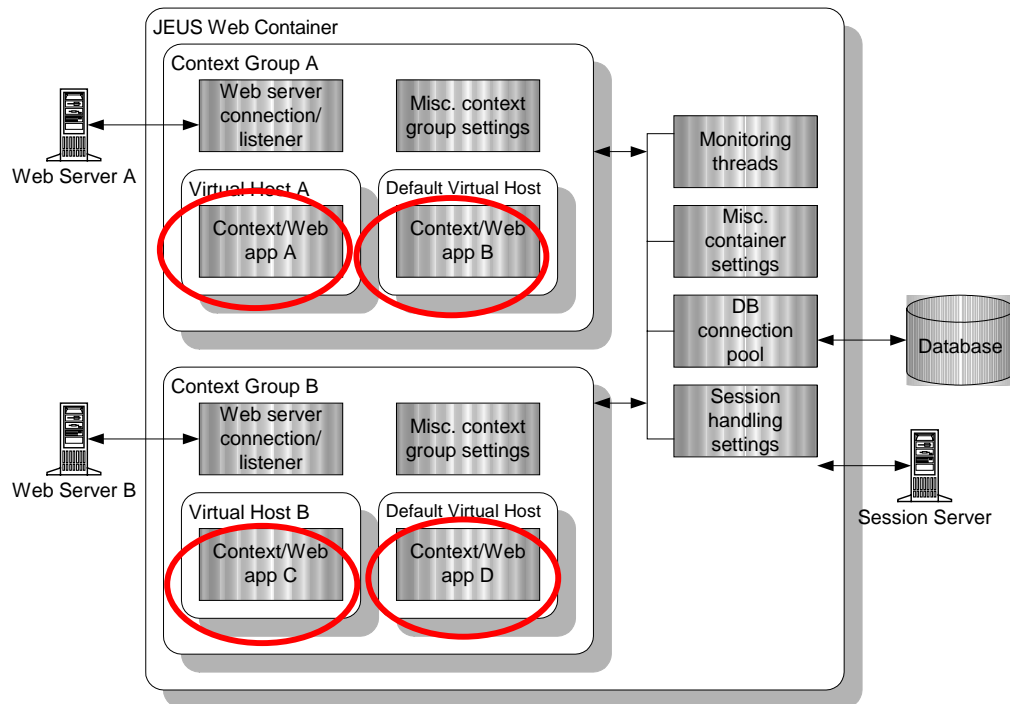


그림 30. JEUS Web Container 에 context/Web application 과 관련된 것들.

9.2.2 Context(웹 어플리케이션) 란 무엇인가?

Web Application 은 클라이언트의 요청에 의한 Web 기반의 서비스(예를 들면, 장바구니에 물품을 추가하거나, 장바구니 안의 물품을 구매하거나 Web 기반의 경매 사이트에서 물건을 사기 위해 브라우징 하는 등)를 수행하기 위한 static 과 dynamic content 의 집합이라고 할 수 있다.

JEUS Web Container 를 설명할 때 J2EE Web Application 에 초점이 맞춰질 것이다. static 인 부분 뿐만 아니라 이런 어플리케이션의 패키징과 deployment 규칙들까지 모두 J2EE 1.4 스펙, Servlet 2.4 스펙, JSP 2.0 스펙에 정의되어 있고 기술되어 있다.

Static Content

static content 는 사전에 제작되어서, 더 이상 어떤 처리도 하지 않고 단순히 반환되는 모든 종류의 데이터들을 의미한다. 다음은 static content 의 예이다.

- HTML 페이지
- 단순 텍스트 파일
- 이미지 파일 (GIF, JPEG)

- 비디오(MPEG)
- 기타

Dynamic Content

변하지 않는 정적과는 반대로 dynamic content 는 클라이언트 요청에 대한 응답으로 지금 즉시 만들어 낸 모든 종류의 데이터를 말한다. 이러한 dynamic content 는 클라이언트와 상호 연동하기 위해 또는 사용자의 선택과 선호에 따라 응답 페이지를 만들어 내는데 사용한다. dynamic content 는 따라서, 저장되기 보다는 생성된다고 표현한다. 다음의 리스트는 J2EE Web application 에서 dynamic 인 content 를 만들어 내는 세 가지 종류를 열거한 것이다.

- Servlet 클래스
- Java help 클래스와 라이브러리
- JSP 페이지

9.2.3 WAR 파일과 그 구조

J2EE Web application 은 J2EE Web Container 에 배포되고 등록되기 전에 특수한 아카이브(라이브러리) 파일에 패키지 해 놓아야 한다. 이 어플리케이션들을 패키지 하는데 사용하는 아카이브들은 JAR/ZIP 파일들로 .war 확장자를 가지고 있다. 이들은 JAR 유틸리티를 사용하여 다른 JAR 아카이브를 만드는 방법과 동일하게 생성된다.

전반적인 WAR 파일의 구조는 다음과 같다.

- **root directory** 모든 종류의 데이터(JSP, HTML, image 등)를 포함하고 있는 디렉토리.
- HTML, JSP, 이미지 파일들과 같은 content 들을 포함하고 있는 임의의 이름의 **sub-folders**.
- Root 디렉토리 아래의 META-INF\ 하위 폴더. 이 폴더는 선택 사항이고, 존재하면 JAR 포맷에 정의된 디스크립터 파일인 MANIFEST.MF 란 파일을 포함한다.
- 필수 WEB-INF\ 디렉토리에는 Web application 의 Servlet, Java 유틸리티 클래스들, JAR/ZIP 라이브러리들이 포함되어 있다. 이 디렉토리 바로 아래에는 다음과 같은 컴포넌트들이 존재한다.

- J2EE 표준 Web application deployment descriptor 인 **web.xml** 파일. 이 파일은 Web application 의 모든 메타 정보를 가지고 있다. 그리고, 클라이언트에 의해 접근 가능한 모든 Servlet 과 JSP 들의 리스트를 가지고 있어야 한다.(그렇지 않으면 Web Container 는 파일들의 위치를 알 수 없다.)
- 새로운 JEUS Web application deployment descriptor 인 jeus-web-dd.xml 파일.(이에 대해서는 9.3.4 절을 참고한다.)
- Servlet 클래스들과 유틸리티 클래스들은 **classes** 하위 디렉토리에 포함되어 있다. (표준 Java 패키지 구조로 정리되어 있다.)
- Web application 에 필요한 Java 라이브러리들을 포함하고 있는 **lib** 하위 디렉토리. 이 라이브러리들은 JAR 와 ZIP 파일들로 패키지 되어 있다. 이 파일들의 내용들은 자동적으로 모든 Servlet 의 클래스 경로에 추가된다.
- JSP 페이지들을 위한 custom tag library descriptor 들을 포함하고 있는 **tlds** 디렉토리.

WAR파일의 샘플 구조가 [그림 31]에 나와 있다.

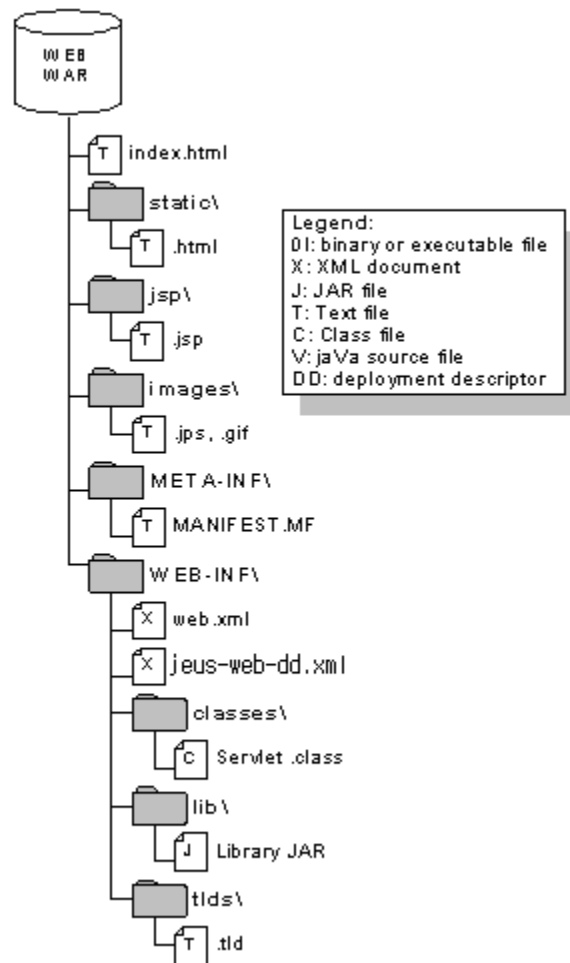


그림 31. 샘플 WAR 파일 내용.

JEUS에서는 각 J2EE WAR파일은 다음 절에서 보는 것과 같이 WAR 등록이 끝나면 동일하게 될 것이다. 그러므로, 몇 개의 WAR/context는 context group 안에 그룹화된다(6 장 참조).

9.2.4 JEUS Web Container 의 WAR 파일 등록

모든 J2EE 호환 WAR 파일은 JEUS Web Container 에 등록(즉, 설치 또는 deploy 라고도 한다)될 수 있다. WAR 파일이 정상적으로 등록되면 Web Container를 재 시작 한 후 context (WAR 파일을 가진)는 클라이언트 요청을 서비스할 준비가 되어 있는 것이다.

이 등록에는 다음과 같은 하위 작업들이 포함된다.

1. 원하는 context 디렉토리를 만든다. 이 디렉토리는 context 의 *document base* (docbase) 디렉토리로 불린다. Web Container 는 클라이언트가 요청하면 이 디렉토리에 접근하여 context 자원을 가져간다.
2. 새로운 context 디렉토리에 WAR 파일의 내용을 풀어놓는다. 새롭게 생성된 context 의 디렉토리 구조는 WAR 파일의 구조와 동일하다.
3. 새로운 JEUS Web application deployment descriptor 파일 작성. 이 파일은 “jeus-web-dd.xml”로 명명된다.
4. web.xml 에서 발견할 수 있는 특정 레퍼런스(EJB 레퍼런스 또는 보안 역할)를 실제 시스템 자원에 매핑하기. 예를 들어, 이것은 symbolic EJB 레퍼런스(web.xml 에서 발견할 수 있는)를 실제의 EJB JNDI 이름에 프로그래머가 정의한 보안 역할을 실제 시스템의 사용자로 매핑하는 것을 포함한다. 모든 이러한 매핑 정보는 “jeus-web-dd.xml” 파일에 입력된다.
5. 다른 Web Container 특정 설정을 새로운 context 에 맞춰 설정하기. 이런 설정의 예에는 사용자 로그 작동 방식 등이 있다. (“jeus-web-dd.xml”에도 설정됨)
6. JEUS.xml 에 새로운 application 으로 context 등록하기. 여기에는 context 의 이름과 경로, 그리고 디플로이 시킬 컨테이너와 context group 을 설정한다.
7. 선택적으로 처음 JSP 를 호출할 때, 보다 빠르게 실행하기 위해서 미리 컴파일한 JSP 파일을 사용할 수도 있다.
8. Web Container 를 재 시작 한다.

이 모든 작업들은 편집기를 이용하여 할 수도 있지만 웹 관리자를 이용할 것을 권장한다. Web application/context 등록은 9.3 절에서 다시 다룬다.

9.2.5 등록된 Web Application 의 디렉토리 구조

Web Container에 새로운 context가 등록된 후, 새 디렉토리로 표현되는 context document base(“docbase”)는 해당 context group 디렉토리 아래에 생성된다. 이 디렉토리는 Web application의 모든 자원과 데이터를 가지고 있다. Context와 그 상위 컴포넌트들의 기본 디렉토리 구조가 [그림 32]에 표현되어 있다.

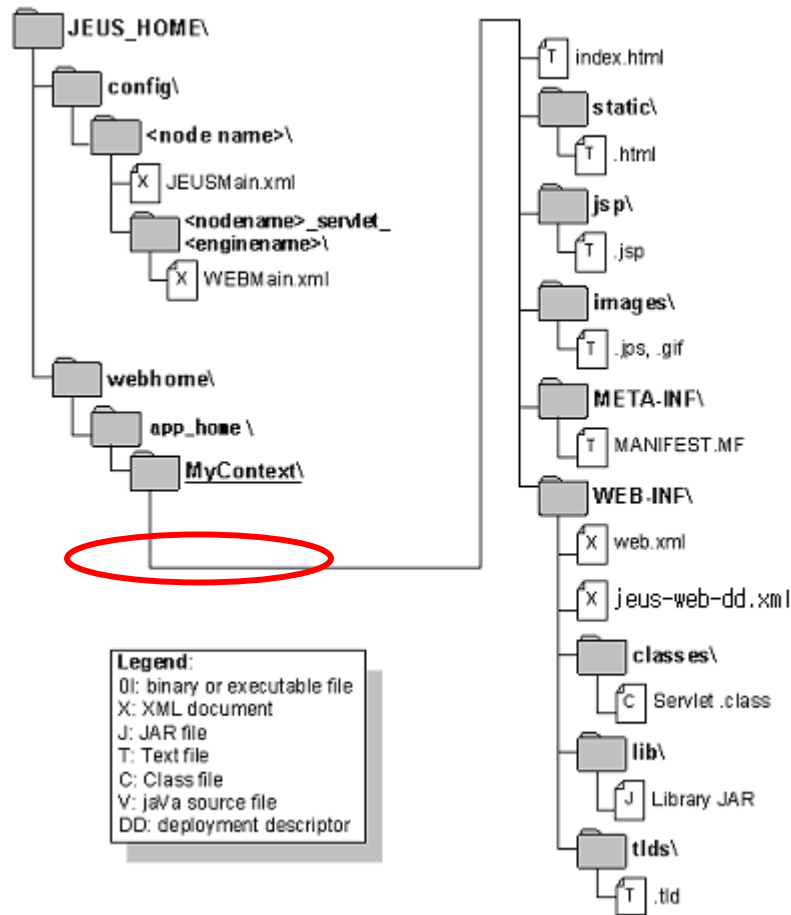


그림 32. 기본 context 디렉토리의 구조

등록된 context 는 다음과 같은 디렉토리와 파일들을 가지고 있다.

- **context deployment descriptor**("jeus-web-dd.xml")는 WEB-INF 디렉토리에 저장된다. Web Container 에 등록된 context/WAR 파일에는 한 개의 DD 파일이 존재한다.
- **context document base directories**(docbase, [그림 32]의 두 번째 원). 이 디렉토리의 내용과 구조는 WAR파일의 구조와 동일하다.(위의 9.2.3 참조)

참고: context document base directory 의 이름과 WAR 파일의 이름, JEUS Web DD 파일의 이름은 서로 연관성이 없어도 된다. Context 의 docbase 디렉토리에는 임의의 이름이 지정되어도 된다.

9.2.6 Context Group, 가상호스트, Context

Context 는 논리적으로 context group 또는 가상호스트 아래에 존재한다. (가상 호스트는 context group 의 서브 컴포넌트이다.)

등록된 Context 는 묵시적 기본 가상 호스트로 작동한다. (즉, WEBMain.xml 설정파일에서는 보이지 않음)

이 장에서는 간략하게 기본 가상 호스트의 일부로 context group 아래에 context 를 설정한다고 가정한다. 가상 호스트의 더 자세한 사항은 10 장에서 다룬다.

9.2.7 기본 Context

Web Container 의 각 context group 은 기본 context 를 가지고 있다. Context 경로를 “/”(실제로 “/”)로 하는 모든 Web application 은 묵시적으로 기본 context 라고 여기면 된다.

9.3 절에 이에 대해 더 자세히 설명한다.

9.2.8 결론

이것으로 context, WAR 파일, WAR 파일의 등록에 대한 설명을 마친다.

WAR 파일을 조립하는 과정에 대한 설명은 JEUS Builder 안내서를 참고하도록 한다.

다음 절에서는 어떻게 JEUS Web Container 에 J2EE WAR 파일을 deploy 하는지에 대해 설명한다.

9.3 Web context 등록

9.3.1 소개

이 부분에서는 어떻게 J2EE Web context (WAR 파일)가 JEUS Web Container 에 등록되는지 알아본다. 여기에서 “등록”은 “deploy”와 “설치”라는 의미와 동일하다.

등록은 앞에서 간단히 설명한 바와 같이, 모든 J2EE web application 을 JEUS Web Container 에서 요청을 받고 수행할 수 있도록 준비하는 과정을 말한다.

나중에 Web application 이 실제로 어떻게 접근되고 제어되는지 알아본다.

아래의 몇 개의 절에서는 수동으로 등록하는 방법을 알아본다.

9.3.2 Context 디렉토리 생성과 WAR 언팩킹(Unpacking)

새로운 WAR 파일을 등록하기 위해서는 WAR 파일이 위치할 새로운 context 디렉토리를 만들어야 한다. 새로 만든 context 디렉토리나 WAR 파일은 JEUSMain.xml 에 등록 시켜야 한다.

Context 디렉토리 경로의 예는 다음과 같다.(임의의 경로이므로 원하는 위치로 옮겨서 사용하면 된다.)

JEUS_HOME\webhome\app_home\MyContext

다음으로는 디렉토리에 WAR파일을 언패킹한다. 따라서 결과적으로 구조는 [그림 33]과 같이 된다.

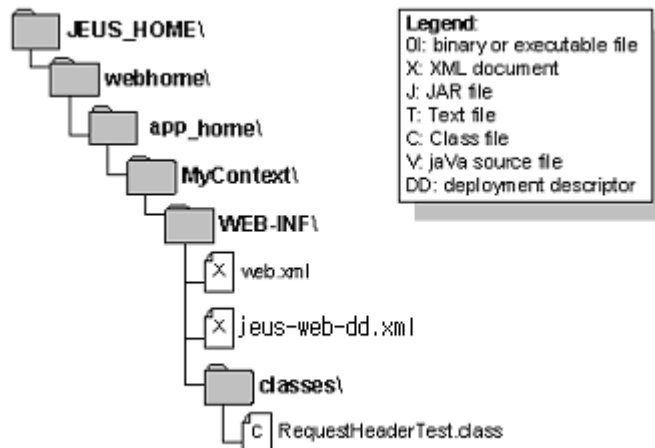


그림 33. "MyContext" 라는 context 의 디렉토리 구조.

9.3.3 Deployment Descriptor 파일 설정

다음으로 할 일은 그 context 를 위해 JEUS 전용의 Deployment Descriptor(이하 DD)를 만드는 것이다. 이것을 위해서는 "jeus-web-dd.xml 의 포맷에 맞도록 파일을 생성한다. 이 파일은 JEUS_HOME\webhome\app_home\ MyContext \WEB-INF 디렉토리에 위치해야 한다.

이제 Web DD 파일에 내용을 넣어보자. 다음에는 이 파일에 포함될 설정 요소들에 대해서 설명되어 있다.

<<jeus-web-dd.xml>>

```
<?xml version="1.0"?>
<jeus-web-dd xmlns="http://www.tmaxsoft.com/xml/ns/jeus">
    <context-path>/Test</context-path>
    <docbase>MyContext</docbase>
    <enable-jsp>true</enable-jsp>
</jeus-web-dd>
```

위의 XML Web DD 는 “/Test” 라는 새로운 context path 와 document base 디렉토리 “MyContext”를 포함하고 있다.

9.3.4 Web DD 의 기본 context 설정

여기에서는 어떻게 새로운 Web application이 JEUS 특정 Web DD(Deployment Descriptor)에 설정되는지 알아본다. 이 파일의 기본 context 설정은 무수히 많다. 다음 리스트에 모든 것들을 소개하고 있지만, 좀더 자세한 정보를 위해서는 부록 E 를 참조한다.

- 모든 설정은 WEBMain.xml 이 들어 있는 디렉토리의 “jeus-web-dd.xml”에 설정되어야 한다. 각 context 마다 이런 DD 파일을 가진다. 이 파일의 설정은 다음과 같다.
- **context path:** 이는 WEBMain.xml에서 해당 context에 설정된 context path와 동일하다. 이 속성은 jeus-web-dd파일이나 WEBMain.xml중에 어디든 설정할 수 있다. 두 군데 모두 설정되어 있고 설정 값이 다를 경우 jeus-web-dd파일에 설정한 path가 우선한다. context path에 대한 자세한 정보는 9.3.7 절을 참조하라.
- **context 의 document base directory** (“docbase”)는 Web application 자원(파일들)이 위치하는 곳이다.
- **user log settings**은 context group의 user log 설정과 같다.(6.2.13 절을 참조하라)
- **auto reload feature** 는 디스크의 Servlet 클래스가 변경되었을 때 자동적으로 로딩이 되는지에 대한 설정이다. (이는 Container 의 클래스 리로딩을 모니터링 하는 쓰레드에 설정된 시간 주기에 따라 정기적으로 확인된다.) 이 하위 태그에는 클래스 리로딩이 새로운 요청이 들어왔을 때만 발생하도록 하는 것도 있다.
- **JSP switch** 는 false 로 설정했을 때, JSP 기능을 완전히 사용하지 않는다(기본은 JSP 를 지원한다).

- **maximum instance pool size** SingleThreadedModel 를 implement 한 Servlet 은 한 번에 한 쓰레드에 의해서만 수행된다. 이러한 Servlet 을 여러 개 동시에 수행 하려면 SingleThreadedModel 의 풀이 생성되어야 한다. 이 설정은 그 풀의 최대 크기를 설정한다. (생성될 최대 Servlet 인스턴스 수)
- **class path additions** 은 Servlet 들이 컴파일되고 실행될 때 접근 가능하게 해야할 Java 클래스 디렉토리 또는 JAR 파일들의 리스트들이다. 기본적으로 “classes\”디렉토리의 클래스 파일들 ,”WEB-INF\”디렉토리의 “lib\” 디렉토리의 JAR 파일들은 접근 가능한 위치들이다.
- **client indexing control** 는 클라이언트가 요청했을 때, 실제 디렉토리 목록이 출력되는 URL 경로의 목록이다. 디렉토리의 목록이 출력되는 경우는 클라이언트에서 요청한 파일이 디렉토리에 없을 때이다. 대신에 디렉토리 목록의 파일을 하나 선택하면, 그 파일의 내용이 보여진다.
- **download denial settings** 는 어떤 경우에라도 직접적으로 다운로드 받거나 접근할 수 없는 자원들을 규정하기 위한 필터이다. 이 필터들은 파일이름의 확장자로, 파일이름과 함께 경로로 지정할 수 있다. 필터의 파일이름과 디렉토리들은 context document base 디렉토리의 상대적인 경로로 주어진다.
- **aliasing** 은 Custom URL 경로에 대한 디렉토리 매핑을 의미한다. 어떤 경우에는 다른 위치에 있는 디렉토리를 URL 요청 경로에 매핑을 시킬 필요가 있다. 이런 때 aliasing 기능을 사용한다.
- **file cache** 설정은 Container 의 응답 속도 향상을 위해 런타임 메모리에 어떤 정적 content(HTML, GIF, JPEG 등)가 캐싱이 되어야 할지 결정한다. 이 항목의 하위 설정들은 캐시 메모리의 최대 크기(단위: Megabytes), 요청되지 않을 때 캐시 안에 머무를 수 있는 정적 자원의 최대 크기, 캐싱 되어야 할 content 가 있는 context docbase 의 상대경로가 있다.
- **URL rewriting**: 기본적으로 Web Container 는 클라이언트의 Session ID 를 여러 요청 중에도 지속적으로 유지하기 위하여 쿠키를 사용한다. 문제는 만약에 요청과 함께 쿠키가 처음 만들어진 곳의 도메인 이름이 요청이 만들어진 곳의 이름과 다르면 대부분의 Web 브라우저는 Session ID 쿠키 정보를 보내지 않는다는 것이다. 이 옵션(url-rewriting)이 “true”로 설정되어 있으면 쿠키에 의존하는 대신에 URL

rewriting 을 강제로 대신 사용하게 한다. 이렇게 함으로써, Session Tracking 이 다른 도메인 이름이 사용되어 몇 번의 요청이 들어와도 가능하게 된다. 이것은 context 에 의해 반환되는 모든 URL 에 유일한 JSESSIONID URL 파라미터가 붙게 한다. 만약 “false”로 설정되어 있으면 이 기능은 사용되지 않고, 기본적인 쿠키 기반의 작동이 사용된다.

- **JSP engine settings:** 현재 컨텍스트의 JSP 페이지에 대한 설정이다. 이 element는 WEBMain.xml의 JSP engine 설정과 동일하다. 만약에 jeus-web-dd 파일에 JSP engine 설정이 되어 있다면, 여기에 설정된 내용이 WEBMain.xml에 설정된 해당 컨텍스트의 JSP engine 설정보다 우선한다. 이 element에 대한 설정 정보는 6.3.8 절을 참고하라.

context XML 설정 파일의 예가 다음에 있다. 몇몇 항목들은 생략되었지만 다음에 이들에 대해서 설명한다.

<<jeus-web-dd.xml>>

```
<jeus-web-dd xmlns="http://www.tmaxsoft.com/xml/ns/jeus">
  <context-path>/examples</context-path>
  <docbase>examples</docbase>
  <enable-jsp>true</enable-jsp>
  <auto-reload>
    <enable-reload>true</enable-reload>
    <check-on-demand>true</check-on-demand>
  </auto-reload>
  <max-instance-pool-size>20</max-instance-pool-size>
  <added-classpath>
    <class-path>c:\mylib\lib.jar</class-path>
  </added-classpath>
  <allow-indexing>
    <directory>/images/</directory>
  </allow-indexing>
  <deny-download>
    <file>/data/secret.dat</file>
    <extension>dat</extension>
    <directory>/data/</directory>
  </deny-download>
  <aliasing>
    <alias>
```

```

        <alias-name>/images/</alias-name>
        <real-path>c:\web\images\</real-path>
    </alias>
</aliasing>
<file-caching>
    <max-idle-time>1800</max-idle-time>
    <max-cache-memory>10</max-cache-memory>
    <directory>/images/</directory>
</file-caching>
. . .
<url-rewriting>true</url-rewriting>
<jsp-engine>
    . . .
</jsp-engine>
</jeus-web-dd>

```

9.3.5 보안 롤 매핑

이 절에서는 web.xml 에 정의된 보안 롤을 실제 시스템 사용자와 사용자 그룹에 어떻게 지정하는지 살펴보겠다. 이 매핑은 “jeus-web-dd.xml”에 기술된다.

개발자가 다음과 같은 것을 deployment descriptor(web.xml)에 정의하였다고 가정하자.

<<web.xml>>

```

<web-app>
    <security-role>
        <role-name>manager</role-name>
    </security-role>
    <security-role>
        <role-name>developer</role-name>
    </security-role>
    <servlet>
        . . .
    </servlet>
    <security-constraint>
        <web-resource-collection>
            <web-resource-name>
                MyResource
            </web-resource-name>
        </web-resource-collection>
    </security-constraint>

```

```

        </web-resource-name>
        <url-pattern>/jsp/*</url-pattern>
        <http-method>GET</http-method>
        <http-method>POST</http-method>
    </web-resource-collection>
    <auth-constraint>
        <role-name>manager</role-name>
    </auth-constraint>
</security-constraint>
</web-app>

```

위의 리스트에서는 굵은 글씨로 두 개의 보안 롤이 선언 되어 있다. 세 번째 굵은 글자 라인은 manager 롤이 <security-constraint>태그에 어떻게 사용되어 지는지를 보여주고 있다.

Web application 이 실제의 시스템에 deploy 될 때, manager 와 developer 롤을 시스템의 특정한 사용자로 바인딩 시켜줘야 한다. 이 매핑은 <context><role-mapping> 에 정의 되어 있다. <role-mapping> 태그는 0 개 이상의 <role-permission> 태그들을 가지고 있다. 이 태그를 이용하여서 principal-to-role 매핑을 정의한다. 다음의 하위 태그들과 함께 바인딩된다.

- **role** (1 개, 필수적): web.xml 에 정의된 **role name** 이며, 위의 예에서는 **role name** 이 “manager”이다.
- **principal** (0 개 이상): role name 과 연계되어야 할 JEUS 가 관리하는 사용자의 이름이다. 더 자세한 사항은 JEUS Security 안내서 4.4 웹 모듈에서 보안 설정하기 부분을 참조하기 바란다.

두 개의 롤 “manager”와 “developer”가 “Peter”와 “Linda”로 각각 매핑되어 있다.

```

<<jeus-web-dd.xml>>

<jeus-web-dd xmlns="http://www.tmaxsoft.com/xml/ns/jeus">
    ...
    <role-mapping>
        <role-permission>
            <principal>Peter</principal>
            <role>manager</role>
        </role-permission>
        <role-permission>

```

```

        <principal>Linda</principal>
        <role>developer</role>
    </role-permission>
</role-mapping>
...
</jeus-web-dd>

```

9.3.6 심볼릭 레퍼런스 매핑

롤이 실제 사용자에게 매핑 되듯이 EJB 레퍼런스, 자원 레퍼런스, 관리되는 객체의 레퍼런스를 실제 시스템 자원에 매핑할 필요가 있다.

아래 예를 보기 바란다.

<<web.xml>>

```

<web-app>
...
  <ejb-ref>
    <ejb-ref-name>ejb/account</ejb-ref-name>
    <ejb-ref-type>Entity</ejb-ref-type>
    <home>com.mycompany.AccountHome</home>
    <remote>com.mycompany.Account</remote>
  </ejb-ref>
  <resource-ref>
    <res-ref-name>jdbc/EmployeeAppDB</res-ref-name>
    <res-type>javax.sql.DataSource</res-type>
    <res-auth>Container</res-auth>
    <res-sharing-scope>Shareable</res-sharing-scope>
  </resource-ref>
  <resource-env-ref>
    <resource-env-ref-name>
      jms/StockQueue
    </resource-env-ref-name>
    <resource-env-ref-type>
      javax.jms.Queue
    </resource-env-ref-type>
  </resource-env-ref>
...
</web-app>

```

이 Web application 을 등록하기 위하여, web.xml 에 있는 <ejb-ref>, <resource-ref>, <resource-env-ref> 아래의 모든 심볼릭 레퍼런스 이름들이 실제 의도하는 자원의 JNDI 이름과 매핑되어야 한다. 이 매핑은 “jeus-web-dd”에 해당 <ejb-ref>, <res-ref>, <res-env-ref>를 추가해 줌으로써 완성된다. 이 태그들은 다음과 같은 하위 태그들을 가지는 <jndi-info> 태그를 가진다.

- **reference name** (ref name)은 web.xml 과 Servlet 코드에 정의된 것과 같은 것이다.
- **JNDI export name** 은 레퍼런스 이름이 뜻하는 실제의 자원 export-name 이다.

다음은 JNDI 이름이 위의 세 개의 레퍼런스와 매핑된 jeus-web-dd.xml 의 예이다.

<<jeus-web-dd.xml>>

```
<jeus-web-dd xmlns="http://www.tmaxsoft.com/xml/ns/jeus">
  <context>
    ...
    <ejb-ref>
      <jndi-info>
        <ref-name>ejb/account</ref-name>
        <export-name>AccountEJB</export-name>
      </jndi-info>
    </ejb-ref>
    <res-ref>
      <jndi-info>
        <ref-name>jdbc/EmployeeAppDB</ref-name>
        <export-name>EmployeeDB</export-name>
      </jndi-info>
    </res-ref>
    <res-env-ref>
      <jndi-info>
        <ref-name>jms/StockQueue</ref-name>
        <export-name>StockQueue</export-name>
      </jndi-info>
    </res-env-ref>
    ...
  </context>
</jeus-web-dd>
```


9.3.7 JEUSMain.xml 에 새로운 context 등록하기

- 새로운 컨텍스트를 등록하기 위해 마지막으로 해야 할 것은 JEUSMain.xml 에 Application 으로 context 를 등록하는 일이다. 해당 context 가 디플로이 될 context group 또는 가상 호스트를 application 태그에 추가함으로써 이 등록이 완성된다.

예:

```
<<JEUSMain.xml>>
<jeus-system xmlns="http://www.tmaxsoft.com/xml/ns/jeus">
  <node>
    <name>tmax1</name>
    ...
  </node>
  <application>
    <name>Examples</name>
    <path>
      /home/miere/jeus50/webhome/app_home/examples
    </path>
    <deployment-target>
      <target>
        <engine-container-name>
          tmax1_container1
        </engine-container-name>
        <web-context-group>
          <name>MyGroup</name>
          <virtual-host-name>
            www.foo.com
          </virtual-host-name>
        </web-context-group>
      </target>
    </deployment-target>
    <deployment-type>COMPONENT</deployment-type>
    <web-component />
  </application>
  <application>
    <name>MyApp</name>
    <path>
      /home/miere/jeus50/webhome/app_home/myapp
```

```

        </path>
        <deployment-target>
            <target>
                <engine-container-name>
                    tmax1_container1
                </engine-container-name>
                <web-context-group>
                    <name>MyGroup</name>
                    <virtual-host-name>
                        www.foo.com
                    </virtual-host-name>
                </web-context-group>
            </target>
        </deployment-target>
        <deployment-type>COMPONENT</deployment-type>
        <web-component />
    </application>
</jeus-system>

```

위의 예는 “Examples”와 “MyApp”라는 두 개의 context 를 등록한다.

두 context의 요청 URL은 “/examples”과 “/myapp”라고 지정되어 있다. 그러므로, 예를 들어, <http://www.foo.com/examples/index.html>를 요청하면 “Examples” context의 “index.html”을 반환한다. (Web Container의 context group의 HTTP 리스너의 포트가 80으로 설정되어 있다고 가정한다)

9.3.8 배치 컴파일러를 사용한 JSP 프리컴파일

등록의 마지막 과정으로 Web application에 있는 JSP페이지들을 프리컴파일 할 수 있다. 이것은 JEUS_HOME\bin\에 있는 JSP 배치 컴파일러 jspc 툴로 한다. 이것은 부록 B에 모두 설명되어 있다.

JSP 배치 컴파일러를 이용하면, JSP가 처음 요청되었을 때의 성능을 향상시킬 수 있다. 이 툴은 새 Web application이 등록되었을 때 각 JSP를 수동으로 요청해야 하는 번거로움을 없애주므로 시스템 관리자에게는 편리성을 제공하는 것이다.

9.3.9 등록 확인

위의 과정을 거친 후 등록이 정확히 되었는지 수동으로 확인하기 위해서 jeusadmin 툴을 사용하여 Web Container를 재시작 한다. (아래의 예에서는 JEUS 노드 이름이 “johan”이라고 가정한다.

```
C: \>j eusadmi n j ohan
```

사용자 이름과 암호를 입력한다.

```
j ohan> downeng j ohan_servl et_engi ne1
```

잠시 후:

```
j ohan> starteng j ohan_servl et_engi ne1
```

Web Container 가 시작하고 운영 환경에 새롭게 등록한 context 가 반영되는 것을 확인 할 수 있을 것이다.

webadmin 을 시작한다.

```
C: \webadmi n j ohan_contai ner1
```

사용자 이름과 암호를 입력한다. 'info' 명령을 수행한다.

```
$$1 [j ohan] i nfo
```

context group 과 context 에 대한 정보가 출력된다. 새롭게 추가한 context ("TestContext")가 리스트에 보여질 것이다. 이것으로 등록이 성공적이었음을 알 수 있다.

9.3.10 웹 관리자를 통한 context 등록

웹 관리자를 사용하여 WAR 모듈을 등록하는 방법에 대한 설명은 JEUS 웹 관리자 안내서를 참고하도록 한다.

9.3.11 결론

이것으로 JEUS 에 Webcontext 를 등록하는 방법을 마친다.

그 과정이 수동으로 또는 웹 관리자를 사용하여 가능하다는 것을 알았다. 후자의 방법을 권장한다.

수동으로 등록하는 방법은 여러 일을 해야 하는 어려움이 있다. Context 디렉토리를 만들고, 이 디렉토리에 WAR 파일을 풀고, JEUS Web DD 파일을 만들고, WEBMain.xml 에 context 를 등록시키며 마지막으로 Web Container 를 재시작 해야 한다.

다음 절에서는 새롭게 등록한 context 와 Servlet 을 어떻게 요청하고 실행시키는지 알아본다.

9.4 Web Context 요청

다음과 같이 예를 들어보자.

- WEBMain.xml 의 context group 에 “/Test” 경로를 가진 context 를 등록하였다. (여기서는 명시적인 가상호스트가 사용되지 않는 것으로 가정한다.)
- 이 context 는 web.xml 에 “HeaderTest”라는 Servlet 을 가진다.
- 이 context 가 소속된 context group 은 HTTP 리스너가 설정되어 있다. 이 리스너는 8088 포트를 로컬 머신에서 사용하고 있다.

위의 것을 가정하고 context 의 HeaderTest Servlet 을 다음과 같은 URL 을 Web 브라우저에서 실행해 보자.

<http://localhost:8088/Test/HeaderTest>

여기서는 다음 [그림 34]과 같은 페이지가 나온다.

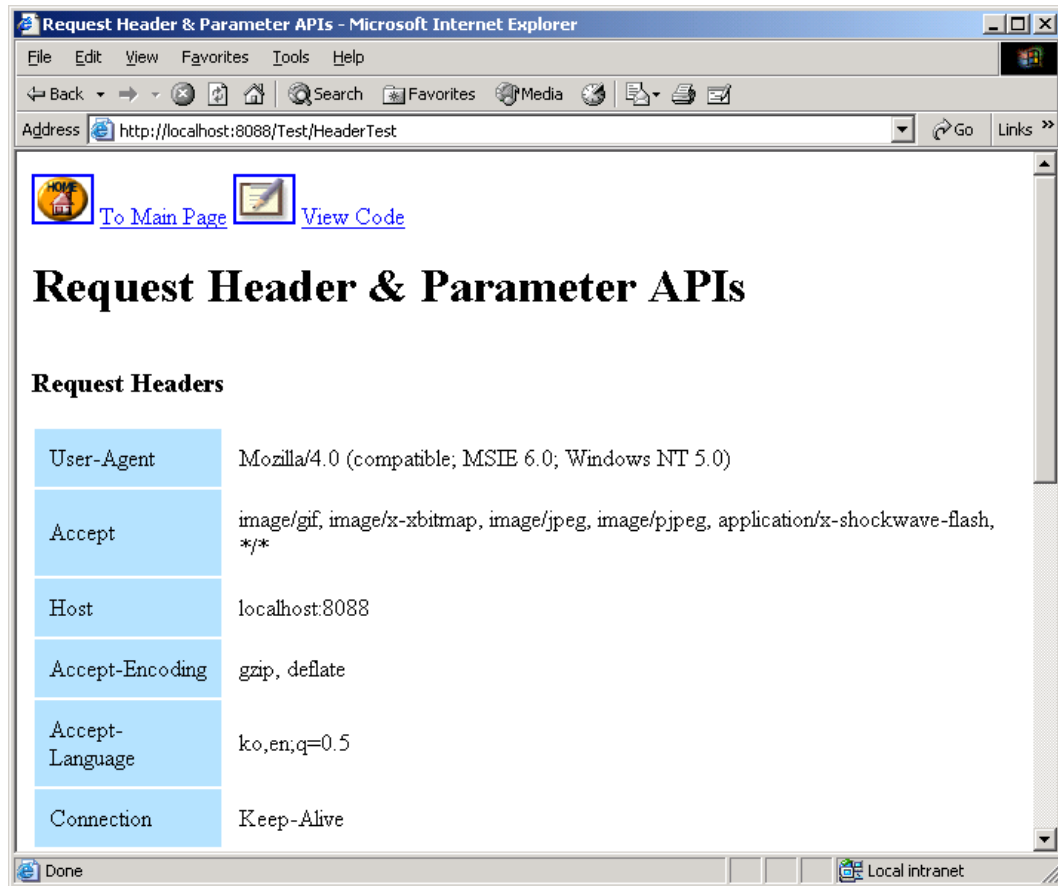


그림 34. Web 브라우저에서 Servlet 요청하기.

그러므로, Servlet(또는 JSP)를 요청할 때에는 다음과 같은 URL 포맷을 이용한다.

```
http://<hostname>:<port>/<context request
path>/<Servlet/JSP name>
```

context docbase 디렉토리의 하위 디렉토리에 저장되어 있는 정적 content에 대해서는 다음과 같은 URL 포맷을 사용한다.

```
http://<hostname>:<port>/<context request path>/<directory
path>/<file name>
```

예: "/Test" 요청 경로를 가진 context 의 "static" 디렉토리 아래에 있는 "hello.html"을 호출하기 위해서는 다음과 같은 URL 을 사용한다.

```
http://localhost:8088/Test/static/hello.html
```

URL 경로의 파일 이름을 지정하지 않으면 기본 페이지인 "index.html"이 사용된다.

요청된 파일 또는 자원이 발견되지 않을 경우에는 다음과 같은 기본 오류 페이지가 반환된다.[그림 35]

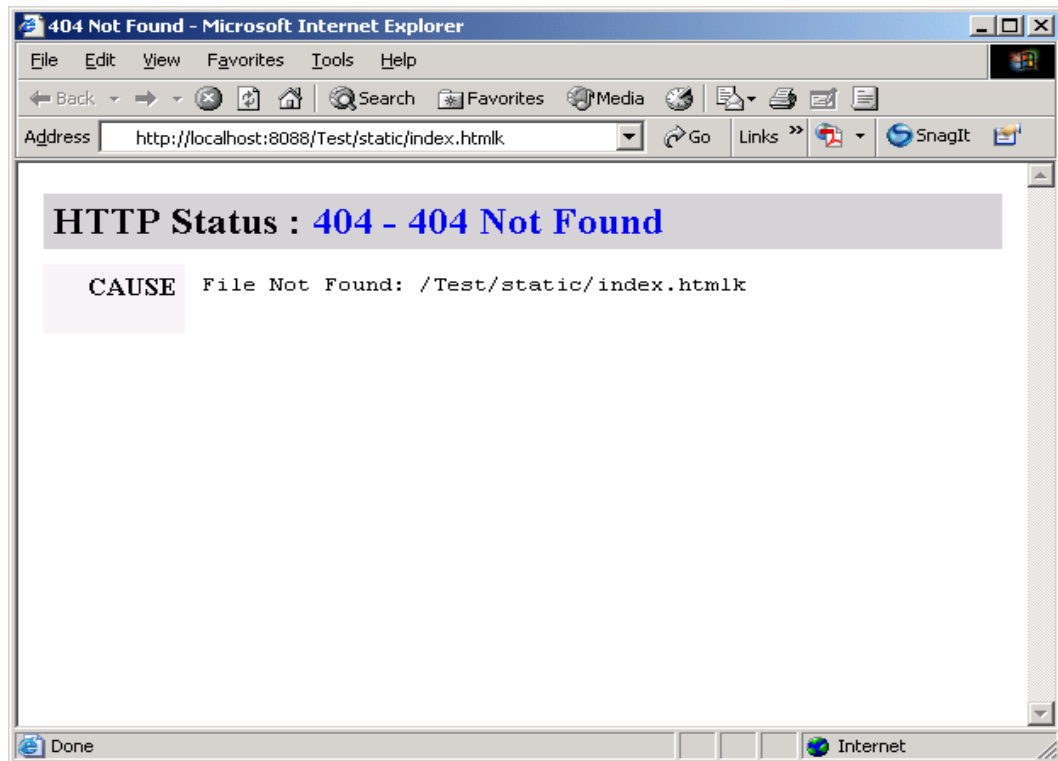


그림 35. 404 오류가 발생하였을 경우 JEUS Web Container 에 의해 생성되는 기본 오류 페이지.

참고: 10 장에는 context 요청에 대해서 더 소개되어 있다. 그 곳에서는 가상 호스트의 사용도 설명하고 있다.

9.5 Web Context 제어

9.5.1 소개

Web 컨텍스트와 Servlet을 제어하기 위한 수단은 6 장에 설명한 context group의 제어와 같다.

명령어는 reload, restart, suspend, resume, terminate, undeploy 가 있다.

webadmin 또는 웹 관리자를 사용하여 이 명령어들을 수행한다. 두 방법 모두를 아래에서 설명한다.

9.5.2 6 개의 명령어

6 개의 기본 명령어의 해석은 다음과 같다.

- **Reload:** 디스크에서부터 context/Servlet 데이터를 다시 읽는다.
- **Restart:** context/Servlet 를 재시작한다. 이 명령어는 기본적으로 “undeploy”다음에 “deploy”를 수행한다.
- **Suspend:** 임시적으로 context/Servlet 를 중지 시킨다. 중지된 context/Servlet 는 클라이언트 요청을 서비스하지 못한다.
- **Resume:** 임시 중지된 context/Servlet 서비스를 가동 시킨다.
- **Terminate:** context 또는 Servlet 의 서비스를 중지 시키고, Web Container 의 운영 환경에서 context/Servlet 를 제거한다.
- **Undeploy:** Web Container 에서 context 를 완전히 제거하고 deploy 되기 전의 상태로 되돌린다. 이 명령어는 각각의 Servlet 에는 적용하지 못한다.

9.5.3 webadmin 콘솔 툴을 사용한 context 와 Servlet 제어

webadmin 콘솔 툴에서 context 와 각각의 servlet 을 제어하기 위해 다음과 같은 명령어들을 사용할 수 있다.

```
reload <context group name> [<context name> [<Servlet name>]]
restart <context group name> [<context name> [<Servlet name>]]
suspend <context group name> [<context name> [<Servlet name>]]
resume <context group name> [<context name> [<Servlet name>]]
terminate <context group name> [<context name> [<Servlet name>]]
```

예:

```
C: \> webadmin johan_container1
```

인증정보를 입력하고, 프롬프트가 나오면 다음과 같이 입력한다.

```
$$1 [johan] suspend MyGroup TestContext
```

“TestContext” 라는 context 가 임시 중지되고 서비스 불가의 상태가 된다. (Servlet 도 포함) 다음의 메시지가 응답으로 출력된다.

...
suspend successful

9.5.4 웹 관리자를 통한 context 와 Servlet 제어

웹 관리자를 통하여 context/Servlet 를 제어하기 위한 설명은 JEUS 웹 관리자 안내서의 J2EE 모듈 장의 웹 모듈 절을 참고하도록 한다.

9.5.5 결과

지금까지 context 와 각 servlet 을 제어할 수 있는 5 개의 기본 명령어들을 살펴보았다. webadmin 과 웹 관리자에서 수행할 수 있는 일은 동일하다.

다음 절에서는 어떻게 context 와 servlet 을 모니터링 하는지에 대해 알아보자.

9.6 Web Context 모니터링

9.6.1 소개

webadmin 또는 웹 관리자를 통하여 context 와 Servlet 을 모니터링 할 수 있다. 후자의 툴을 더 권장한다.

두 가지의 방법을 아래에서 모두 소개하겠다.

9.6.2 webadmin 콘솔 툴을 사용한 Web context 모니터링

webadmin 에서 deploy 된 context group 의 정보를 보려면 'info'를 수행한다. 'info <context group name>'은 특정 context group 에 대한 정보만 제공한다.

'info <context group name> <context name>'은 특정 context 에 대한 정보만 보여준다. (특정 servlet 에 대한 정보는 보지 못한다.)

더 자세한 정보는 부록 A 를 참조하라.

9.6.3 웹 관리자를 통한 Web context 모니터링

- 웹 관리자에서 Context 모니터링과 servlet 모니터링 관련 내용을 살펴보고자 한다면 JEUS 웹 관리자 안내서의 웹 모듈 통계 장을 참고하도록 한다.

9.6.4 결론

지금까지 webadmin 과 웹 관리자를 사용하여 context 와 servlet 을 모니터링 하는 방법에 대해 알아보았다.

이 장의 마지막 부분에서는 Web context 설정을 튜닝하는 방법에 대해 이야기 한다.

9.7 Web Context 튜닝

최대한의 성능을 내기 위해 context 의 설정(jeus-web-dd.xml)을 튜닝할 때 다음 같은 것들을 고려해 보자.

- User log 은 버퍼의 크기를 증가시키고 “stdout”의 사용을 자제한다.
- JSP engine 은 사용하지 않으면 OFF 시킨다.
- “enable-reload”와 “check-on-demand” 옵션은 항상 OFF 시킨다. 이 옵션은 클래스 변경이 잦은 개발환경에서만 사용한다.
- max-instance-pool-size 를 “-1”로 하여 SingleThread 된 servlet 에게 최적의 동시성을 제공한다(그러나 잠재적으로 시스템 자원을 낭비하게 된다).
- 가능하면 파일 캐싱 기능을 사용한다. 최대한 많은 정적 content 디렉토리를 파일 캐싱 태그에 설정한다. 캐시의 max-idle-time 와 max-cache-memory 의 값을 크게 정한다(무한대로 정해도 된다).

9.8 결론

이 장에서는 Web context 와 그 안의 Servlet/JSP 들을 조립, 등록, 요청, 제어, 모니터링 하는 방법에 대해 알아보았다.

다음 장에서는 JEUS Web 환경에서의 가상호스트에 대하여 알아보겠다.

10 가상 호스트

10.1 소개

JEUS Web Container 의 새로운 기능은 가상 호스팅이다.

기본적으로, 가상 호스트는 Web 사이트가 다른 context 에게 주어진 context 경로 뿐만 아니라 요청 주소에 있는 도메인 이름에 의해서도 연결될 수 있도록 한다.

이렇게 함으로써 두 개 이상의 DNS 이름(예, “www1.foo.com” and “www2.foo.com”) 을 하나의 Web Container 에 설정하여 다른 Web context 를 서비스할 수 있다.

가상 호스트는 보기에 따라서 별도의 Web Container 로 보일 수 있다.

아래 [그림 36]은 Web Container에서 가상 호스트에 관련된 컴포넌트들을 보여주고 있다.

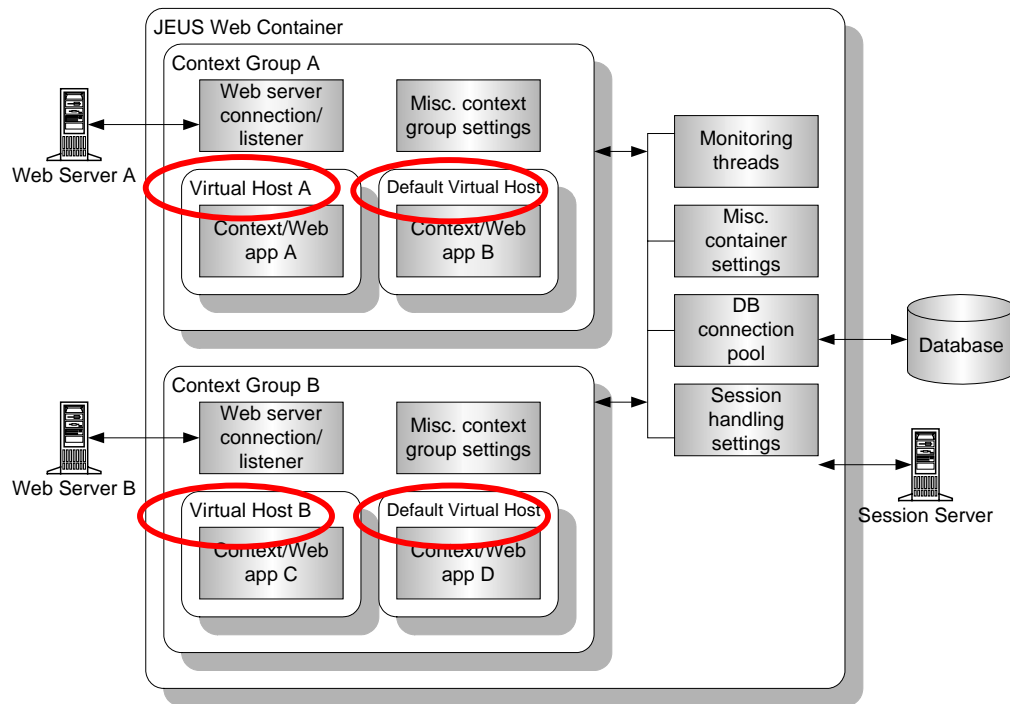


그림 36. JEUS Web Container 의 가상 호스트 컴포넌트.

10.2 가상 호스트의 개요

10.2.1 소개

이 절에서는 간단하게 가상 호스트에 대한 개념을 설명한다. 다음은 설명할 내용이다.

- 가상 호스트의 기본 개념.
- 가상 호스트의 규칙과 구조.
- 기본 가상 호스트(Default virtual host)
- ServletContext 객체와 가상 호스트에 대한 주의 사항.

10.2.2 기본 개념

가상 호스트의 기본적인 개념은 사용자 요청에 포함된 도메인 이름들(예, www.foo.com, www.bar.com)이나 IP 주소(예, 111.111.111.1, 111.111.111.2)에 따라 사용자 요청을 분기하는 것이다.

DNS 이름을 사용한 가상 호스트와 IP 주소를 사용 가상 호스트의 차이점은 DNS 이름을 통해서 라우팅이 되느냐, IP 주소를 통해서 라우팅이 되느냐의 차이점 뿐이다. 궁극적인 동작 방법은 동일하다.

가상 호스트 개념의 도입으로 하나의 Web Container 는 여러 도메인 이름을 가진 요청을 처리할 수 있고, 이 요청을 원하는 context 로 보낼 수 있다.

[그림 37]에서 이에 대한 기본 개념을 보여준다.

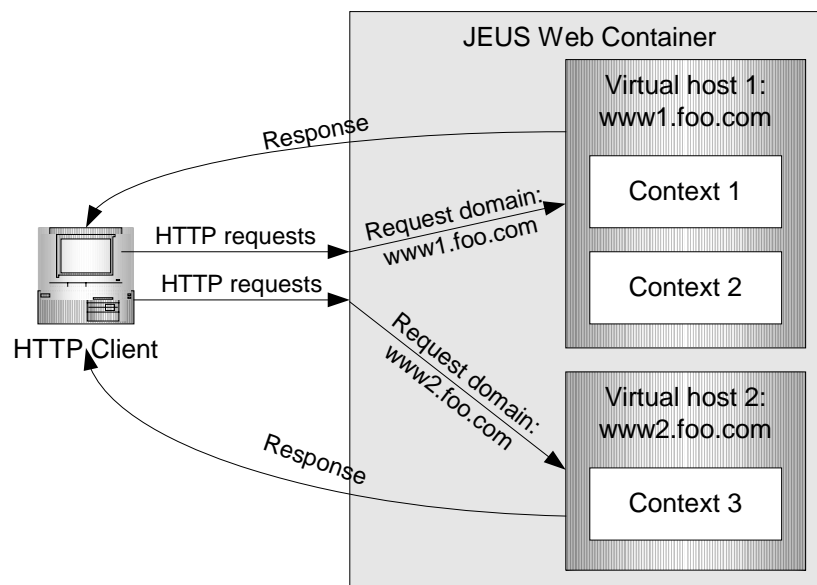


그림 37. 가상 호스트의 기본적인 개념.

10.2.3 가상 호스트의 규칙과 구조

이 부분에서는 가상 호스트를 구성할 때 적용되어야 할 기본적인 규칙에 대하여 알아보겠다.

먼저, 하나의 가상 호스트는 한 개 이상의 DNS 이름이나 IP 주소를 가질 수 있다. 이것을 우리는 가상 호스트의 네트워크 이름이라고 한다. 각 가상 호스트에는 유일한 값의 가상 호스트 ID 이름을 부여한다. 이 이름은 설정 파일 내에서 가상 호스트를 참조하기 위해서 내부적으로 사용되는 이름이다.

가상 호스트는 Web Container 의 context group 에 설정된다. Context group 당 여러 개의 가상 호스트가 존재할 수 있지만, 한 context group 의 모든 가상 호스트들은 유일한 네트워크 이름(DNS 이름이나 IP 주소)을 가져야 한다.

각 가상 호스트는 반드시 하나 이상의 context 와 연계되어 있어야 한다. 같은 가상 호스트 내의 모든 context 들은 각각 유일한 context 이름과 경로(URI 요청 경로)를 가지고 있어야 한다. 그러나, 하나의 context 가 다른 여러 개의 가상 호스트에 존재하면 그 경로는 그 모든 가상호스트에서 동일하게 사용할 수도 있다.

위의 규칙은 [그림 38]을 보기 바란다.

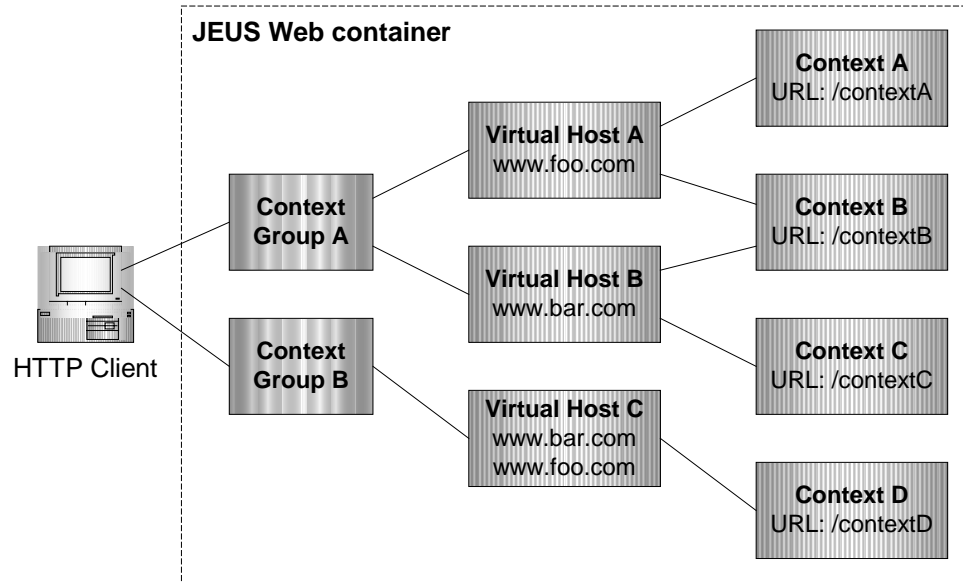


그림 38. context group 들과 가상 호스트 그리고 context 들간의 유효한 관계의 예

[그림 38]에서 주의해서 봐야 할 것은 다음과 같다.

- 두 개의 유일한 context group이 설정되어 있다. 클라이언트는 이 두 group을 소켓 포트 번호로 구분한다. (즉, group “A”는 “80”포트를 group “B”는 “9999”포트를 사용한다.) context group의 Web Server 리스너는 유일한 포트 번호를 가진다는 것을 명심한다(7 장 참조).
- context group “A”는 두 개의 가상 호스트가 추가되어 있다. 두 개의 가상 호스트는 유일한 네트워크 이름을 가지고 있다. Group “B”에는 한 개의 가상 호스트(“C”)가 추가되어 있다. 이 가상 호스트는 두 개의 네트워크 이름을 가지고 있다. www.bar.com과 www.foo.com이다. 이것은 context group내의 다른 가상 호스트가 같은 네트워크 이름을 가지고 있지 않기 때문이다.

- 가상 호스트 “A”는 가상 호스트 “B”와 함께 context “B”를 공유한다. 이것은 같은 가상 호스트 내에서, context 들이 같은 URI 경로를 가지고 있지 않으면 되기 때문에 올바른 설정이다.

위의 규칙에서 우리가 얻을 수 있는 것은, 각 context 는 서로 구분되어 설정되었다는 것이다.

10.2.4 가상 호스트의 주사용

가상 호스트는 context 가 context 경로에 의해서가 아닌 실제 DNS 이름이나 IP 주소에 의해 구분되어 사용될 때 사용되어야 한다.

예를 들어, 두 개의 다른 context 가 한 Web Container 에 있고 모두 기본 context 경로인 “/”에 의해서만 접근 가능해야 한다면, 가상 호스트를 사용하여 한 DNS 이름을 첫 번째 context 로, 다른 DNS 이름을 두 번째 context 로 설정하면서, 같은 Web Container 가 두 컨텍스트를 관리하게 할 수 있다.

[그림 39]에서 이 개념을 설명하고 있다.

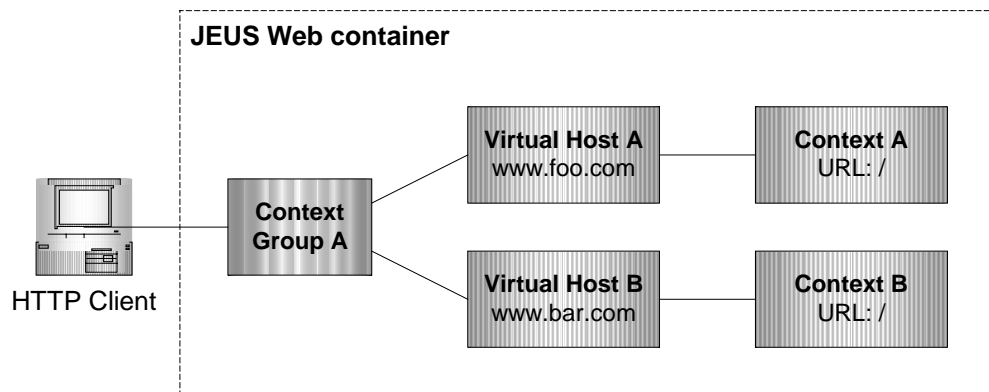


그림 39. 두 context 가 같은 context 경로를 가진 경우 같은 Web Container 에서 DNS 이름으로 구분하고 있다.

위의 그림에서는, 두 개의 context A, B를 가지고 있다. 두 개 모두 한 Web Container에 존재하고 같은 context group과 리스너 그리고 경로("/")를 가지고 있다. 다른 점은 context A는 URL <http://www.foo.com/>이 요청될 때 context B는 URL <http://www.bar.com/>이 요청될 때 사용된다는 것이다.

10.2.5 기본 가상 호스트

기본 가상 호스트(Default virtual host)라는 묵시적인 가상 호스트가 존재한다. 이것은 비록 WEBMain.xml 에 설정되어 있지 않지만 항상 존재한다. 명시적

으로 context 를 가상 호스트에 할당하지 않으면, 이 기본 가상 호스트로 할당된다.

9 장에서 미리 봤듯이, 가상 호스트의 관계를 설정하지 않고 context를 선언하는 것은 일반적인 일이다. 이런 모든 context들은 기본 가상 호스트에 속한다고 생각할 수 있다.

기본 가상 호스트의 존재는 JEUS 사용자에게는 보이지 않는다.

10.2.6 javax.servlet.ServletContext.getContext() Method 사용시 주의 사항

Servlet API 를 보면 javax.servlet.ServletContext.getContext(java.lang.String contextPath)라는 메소드를 볼 수 있다. 이것은 “contextPath” URI에 의해 주어진 Servlet context 를 리턴한다.

이 메소드는 그 Servlet 이 속하는 현재 가상 호스트에 존재하는 context 만을 리턴한다. 유효한 context 가 이 가상 호스트 내에서 존재하지 않으면 기본 가상 호스트에서 찾는다.

이 사항은 Servlet 스펙의 요구 사항이다.

10.2.7 결론

지금까지 가상 호스트에 대한 기본적인 것을 알아보았다.

기본 개념, 기본 규칙과 구조, 기본 가상 호스트의 개념 등에 대해 알아보았다. 마지막으로 가상 호스트 환경에서의 ServletContext 객체에 대한 특수 고려 사항에 대해서도 알아보았다.

다음 장에서는 어떻게 Web Container 에 가상 호스트를 설정하는지에 대해 알아보겠다.

10.3 가상 호스트 설정

10.3.1 소개

가상 호스트는 WEBMain.xml 에 설정된다. 그러나, 이것을 작동시키기 위해서는 도메인 이름을 등록해 줘야 한다.

여기서는 그 두 가지에 대해 설명한다.

10.3.2 도메인 이름 등록

복수의 DNS 이름 또는 IP 주소를 가상 호스트에 사용하기 위해서는 DNS 이름이 어딘가에 등록되어야 한다. 여기에는 두 가지 선택 사항이 있다.

1. DNS Server에 Web Container의 리스너 또는 Web Server IP주소 아래의 모든 DNS이름을 등록한다(예, www.foo.com과 www.bar.com).
2. Server의 “hosts”파일에 네트워크 이름과 IP 주소를 추가한다. Windows 2000에서는 c:\WINNT\system32\drivers\etc\hosts에 hosts 파일이 있다. Unix의 경우는 /etc에 hosts파일이 있다. 이 파일에 www.foo.com과 www.bar.com 호스트를 다음과 같이 등록한다.

<<hosts>>

111.111.111.1	www.foo.com
111.111.111.2	www.bar.com

10.3.3 WEBMain.xml 에 가상 호스트 설정

WEBMain.xml 에 가상 호스트를 설정하기 위하여 <context-group>에 <virtual-host> 태그를 추가 한다. 이 태그는 다음과 같은 하위 태그들을 포함한다.

- **virtual host name** 은 가상 호스트를 참조하기 위해 내부적으로 사용하는 이름이다. 여기에 __DEFAULT_HOST__ 를 사용해서는 안 된다. 이것은 기본 가상 호스트를 위해 사용되는 것이기 때문이다.
- DNS 이름 또는 IP 주소를 포함하는 복수 개의 **host list** 태그. 같은 context group 에는 유일한 hostname 만이 올 수 있다.
- 다음을 포함하는 복수의 **context** 태그
 - o application 에 설정된 context 에서 어느 가상호스트를 사용할 것인지 지정한다. 여기에 __DEFAULT_CONTEXT__ 를 사용해서는 안 된다. 이것은 기본 context 를 위해 사용되는 것이기 때문이다
 - o context를 접근 할 때 사용하는 URL 경로인 **context path**. (<http://www.foo.com/examples/index.html>에서는 “/examples”가 context 경로가 된다.) 한 가상 호스트 내에서는 이 경로가 유일한 것이야 한다. 경로는 반드시 “/”로 시작해야 한다.

아래의 예에서는 어떻게 10.2 절의 [그림 74]의 예가 설정되는지 보여주고 있다.

```
<<WEBMain.xml>>
<web-container xmlns="http://www.tmaxsoft.com/xml/ns/jeus">
    . . .
    <context-group>
        <group-name>A</group-name>
        <virtual-host>
            <virtual-host-name>A</virtual-host-name>
            <host-list>www.foo.com</host-list>
        </virtual-host>
        <virtual-host>
            <virtual-host-name>B</virtual-host-name>
            <host-list>www.bar.com</host-list>
        </virtual-host>
        <webserver-connection>
            . . .
        </webserver-connection>
        . . .
    </context-group>
    <context-group>
        <group-name>B</group-name>
        <virtual-host>
            <virtual-host-name>C</virtual-host-name>
            <host-list>www.foo.com</host-list >
            <host-list>www.bar.com</host-list >
        </virtual-host>
        <webserver-connection>
            . . .
        </webserver-connection>
        . . .
    </context-group>
    . . .
</web-container>
```

여기서 어떻게 context group “B”의 가상 호스트 “C”가 두 개의 hostname 을 가지는지 보아야 한다.

참고: 이름 “A”, “B”, “C”는 간략성을 위해 표현되었다. 실제 환경에서는 의미 있는 이름이 사용되어야 한다.

10.3.4 JEUSMain.xml 의 application 에 가상호스트 맵핑 하기

JEUSMain.xml 에 가상 호스트를 설정하기 위하여 <application>에 <virtual-host-name> 태그를 추가 한다.

```
<<JEUSMain.xml>>
<jeus-system xmlns="http://www.tmaxsoft.com/xml/ns/jeus">
  <node>
    <name>tmax1</name>
    ...
  </node>
  <application>
    <name>A</name>
    <path>
      JEUS_HOME/webhome/app_home/contextA
    </path>
    <deployment-target>
      <target>
        <engine-container-name>
          tmax1_container1
        </engine-container-name>
        <web-context-group>
          <name>A</name>
          <virtual-host-name>A</virtual-host-name>
        </web-context-group>
      </target>
    </deployment-target>
    <deployment-type>COMPONENT</deployment-type>
    <web-component/>
  </application>

  <application>
    <name>B</name>
    <path>
      JEUS_HOME/webhome/app_home/contextB
    </path>
    <deployment-target>
```

```

        <target>
            <engine-container-name>
                tmax1_container1
            </engine-container-name>
            <web-context-group>
                <name>A</name>
                <virtual-host-name>A</virtual-host-name>
                <virtual-host-name>B</virtual-host-name>
            </web-context-group>
        </target>
    </deployment-target>
    <deployment-type>COMPONENT</deployment-type>
    <web-component />
</application>

<application>
    <name>C</name>
    <path>
        JEUS_HOME/webhome/app_home/contextC
    </path>
    <deployment-target>
        <target>
            <engine-container-name>
                tmax1_container1
            </engine-container-name>
            <web-context-group>
                <name>A</name>
                <virtual-host-name>B</virtual-host-name>
            </web-context-group>
        </target>
    </deployment-target>
    <deployment-type>COMPONENT</deployment-type>
    <web-component />
</application>

<application>
    <name>D</name>
    <path>
        JEUS_HOME/webhome/app_home/contextD

```

```
</path>
<deployment-target>
  <target>
    <engine-container-name>
      tmax1_container1
    </engine-container-name>
    <web-context-group>
      <name>B</name>
      <virtual-host-name>C</virtual-host-name>
    </web-context-group>
  </target>
</deployment-target>
<deployment-type>COMPONENT</deployment-type>
<web-component/>
</application>
</jeus-system>
```

여기서 context 로 등록된 B 가 어떻게 context group” A 에 설정된 가상 호스트 “A”와 “B” 를 공유하는지 주의 깊게 보아야 한다.

10.3.5 웹 관리자를 통한 가상 호스트 설정

웹 관리자를 사용하여 WEBMain.xml 에 가상 호스트를 설정하려면 JEUS 웹 관리자 안내서의 컨텍스트 그룹 설정 장을 참고하도록 한다

10.3.6 결론

이 절에서는 어떻게 JEUS Web Container 에 가상 호스트가 설정되는지 살펴 보았다.

다음 절에서는 가상 호스트 환경에서 어떻게 context 를 요청하는지에 대해 알아보겠다.

10.4 가상 호스트를 통한 context 요청

10.4.1 소개

이 절에서는 가상 호스트 내에 존재하는 context 를 접근하기 위해 어떻게 URL 을 요청하는지에 대해 알아보겠다.

먼저 URL 매칭에 대한 일반적인 규칙에 대해 알아보고 예를 들어 이를 설명 하겠다.

10.4.2 URL 매칭에 관한 기본 규칙

다음의 네 가지 규칙은 사용자가 제공하는 요청 내의 URL 이 가상 호스트가 존재하는 환경에서의 특정 context 와 매칭되는지 결정한다.

1. 사용자의 URL 요청 경로가 실제 context에 매칭되는 기본적인 규칙은 (가상 호스트의 사용 여부에 상관없이) 다음과 같다.: 도메인 이름(예, www.foo.com) 이후의 URI 부분을, 등록된 context 중에서 가장 긴 context URI 경로에 매칭한다. 만약 매칭이 발견되면 그 context를 사용한다.
2. 가상 호스트가 설정되어 있으면, 먼저 사용자의 URL 에 있는 도메인 이름을 등록된 모든 가상 호스트의 네트워크 이름에 매칭시킨다. 그 후에 이 가상 호스트에 설정된 모든 context 들을 찾는다. 마지막으로, 위의 규칙 1 을 이 context 에 적용시킨다.
3. 규칙 2 에서 context 가 발견되지 않았으면 기본 가상 호스트에 규칙 1 을 적용시킨다.
4. 기본 가상 호스트에도 context 가 발견되지 않으면, 기본 context 가 규칙 1 에 적용된다.

10.4.3 URL 매칭의 예

WEBMain.xml 에 있는 다음과 같은 가상 호스트의 설정을 고려해 보자:

```
<<WEBMain.xml>>

<web-container xmlns="http://www.tmaxsoft.com/xml/ns/jeus">
    . . .
    <context-group>
        <group-name>MyGroup</group-name>
        <virtual-host>
```

```

        <virtual-host-name>vhost1</virtual-host-name>
        <host-list>www1.foo.com</host-list>
    </virtual-host>
    <virtual-host>
        <virtual-host-name>vhost2</virtual-host-name>
        <host-list >www2.foo.com</host-list>
        <host-list >www3.foo.com</host-list>
    </virtual-host>
    <webserver-connection>
        . . .
    </webserver-connection>
    . . .
</context-group>
. . .
</web-container>

```

JEUSMain.xml 에 있는 다음과 같은 가상 호스트의 설정을 고려해 보자:

```

<<JEUSMain.xml>>
<jeus-system xmlns="http://www.tmaxsoft.com/xml/ns/jeus">
  <node>
    <name>tmax1</name>
    ...
  </node>
  <application>
    <name>vhost1ctx1</name>
    <path>
      JEUS_HOME/webhome/app_home/vhost1ctx1
    </path>
    <deployment-target>
      <target>
        <engine-container-name>
          tmax1_container1
        </engine-container-name>
        <web-context-group>
          <name>MyGroup</name>
          <virtual-host-name>vhost1</virtual-host-name>
        </web-context-group>
      </target>
    </deployment-target>
  </application>
</jeus-system>

```

```

        </target>
    </deployment-target>
    <deployment-type>COMPONENT</deployment-type>
    <web-component/>
</application>

<application>
    <name>vhost1ctx2</name>
    <path>
        JEUS_HOME/webhome/app_home/vhost1ctx2
    </path>
    <deployment-target>
        <target>
            <engine-container-name>
                tmax1_container1
            </engine-container-name>
            <web-context-group>
                <name>MyGroup</name>
                <virtual-host-name>vhost1</virtual-host-name>
            </web-context-group>
        </target>
    </deployment-target>
    <deployment-type>COMPONENT</deployment-type>
    <web-component/>
</application>

<application>
    <name>vhost2ctx1</name>
    <path>
        JEUS_HOME/webhome/app_home/vhost2ctx1
    </path>
    <deployment-target>
        <target>
            <engine-container-name>
                tmax1_container1
            </engine-container-name>
            <web-context-group>
                <name>MyGroup</name>
                <virtual-host-name>vhost2</virtual-host-name>

```



```
        </web-context-group>
    </target>
</deployment-target>
<deployment-type>COMPONENT</deployment-type>
<web-component/>
</application>

<application>
    <name>vhost2ctx2</name>
    <path>
        JEUS_HOME/webhome/app_home/vhost2ctx2
    </path>
    <deployment-target>
        <target>
            <engine-container-name>
                tmax1_container1
            </engine-container-name>
            <web-context-group>
                <name>MyGroup</name>
                <virtual-host-name>vhost2</virtual-host-name>
            </web-context-group>
        </target>
    </deployment-target>
    <deployment-type>COMPONENT</deployment-type>
    <web-component/>
</application>

<application>
    <name>default1</name>
    <path>
        JEUS_HOME/webhome/app_home/default1
    </path>
    <deployment-target>
        <target>
            <engine-container-name>
                tmax1_container1
            </engine-container-name>
            <web-context-group>
                <name>MyGroup</name>
```

```

        </web-context-group>
    </target>
</deployment-target>
<deployment-type>COMPONENT</deployment-type>
<web-component/>
</application>

<application>
    <name>default2</name>
    <path>
        JEUS_HOME/webhome/app_home/default2
    </path>
    <deployment-target>
        <target>
            <engine-container-name>
                tmax1_container1
            </engine-container-name>
            <web-context-group>
                <name>MyGroup</name>
            </web-context-group>
        </target>
    </deployment-target>
    <deployment-type>COMPONENT</deployment-type>
    <web-component/>
</application>

<application>
    <name>default3</name>
    <path>
        JEUS_HOME/webhome/app_home/default3
    </path>
    <deployment-target>
        <target>
            <engine-container-name>
                tmax1_container1
            </engine-container-name>
            <web-context-group>
                <name>MyGroup</name>
            </web-context-group>
        </target>
    </deployment-target>
    <deployment-type>COMPONENT</deployment-type>
    <web-component/>
</application>

```

```

        </target>
    </deployment-target>
    <deployment-type>COMPONENT</deployment-type>
    <web-component/>
</application>
</jeus-system>

```

마지막으로 각각의 application 별로 jeus-web-dd.xml 파일에 대해서 알아보자.

<< vhost1ctx1 >/ jeus-web-dd.xml>>

```

<?xml version="1.0"?>
<jeus-web-dd xmlns="http://www.tmaxsoft.com/xml/ns/jeus">
    <context-path>/vpath1</context-path>
    <enable-jsp>true</enable-jsp>
</jeus-web-dd>

```

<< vhost1ctx2 >/ jeus-web-dd.xml>>

```

<?xml version="1.0"?>
<jeus-web-dd xmlns="http://www.tmaxsoft.com/xml/ns/jeus">
    <context-path>/vpath2</context-path>
    <enable-jsp>true</enable-jsp>
</jeus-web-dd>

```

<< vhost2ctx1 >/ jeus-web-dd.xml>>

```

<?xml version="1.0"?>
<jeus-web-dd xmlns="http://www.tmaxsoft.com/xml/ns/jeus">
    <context-path>/vpath3</context-path>
    <enable-jsp>true</enable-jsp>
</jeus-web-dd>

```

<< vhost2ctx2 >/ jeus-web-dd.xml>>

```

<?xml version="1.0"?>
<jeus-web-dd xmlns="http://www.tmaxsoft.com/xml/ns/jeus">
    <context-path>/vpath3/subdir</context-path>
    <enable-jsp>true</enable-jsp>
</jeus-web-dd>

```

<< default1 >/ jeus-web-dd.xml>>

```

<?xml version="1.0"?>
<jeus-web-dd xmlns="http://www.tmaxsoft.com/xml/ns/jeus">
    <context-path>/default1</context-path>

```

```
<enable-jsp>true</enable-jsp>
</jeus-web-dd>
```

<< default2 의 jeus-web-dd.xml>>

```
<?xml version="1.0"?>
<jeus-web-dd xmlns="http://www.tmaxsoft.com/xml/ns/jeus">
  <context-path>/default2</context-path>
  <enable-jsp>true</enable-jsp>
</jeus-web-dd>
```

<< default3 의 jeus-web-dd.xml>>

```
<?xml version="1.0"?>
<jeus-web-dd xmlns="http://www.tmaxsoft.com/xml/ns/jeus">
  <context-path>/vpath3</context-path>
  <enable-jsp>true</enable-jsp>
</jeus-web-dd>
```

위의 설정을 기반으로 하였을 때 어떤 context (“default1”, “default2”, “default3”, “vhost1ctx1”, “vhost1ctx2”, “vhost2ctx1”, “vhost2ctx2”, default context)가 주어진 URL 요청에 매칭될지 살펴보기로 하자.

http://www1.foo.com/vpath2/test.jsp →

vhost1ctx2

http://www1.foo.com/default2/test.jsp →

default2 (in default virtual host)

http://www.foo.com/vpath3/test.jsp →

default3 (in default virtual host)

http://www2.foo.com/vpath3/test.jsp →

vhost2ctx1

http://www3.foo.com/vpath3/subdir/test.jsp →

vhost2ctx2

`http://www.foo.com/default1/test.jsp` →

`default1` (in default virtual host)

참고: 위의 예는 단지 개념을 설명하기 위한 것이다. 실제 설정에서는 좀 더 주의해야 한다.

10.4.4 결론

지금까지 가상 호스트가 설정되어 있을 때, URL 이 어떤 **context** 에 매칭되는지를 관장하는 기본 규칙에 대하여 살펴보았다. 또한 이런 예가 적용된 예도 들어 보았다.

10.5 결론

이것으로 JEUS Web Container 의 가상 호스트에 대한 설명을 마무리 한다.

가상 호스트가 설정된 환경에서 이것이 무엇인지, 어떻게 설정하고 어떻게 **context** 를 요청하는지에 대해 알아보았다.

11 결론

이것으로 JEUS Web 모듈에 대한 설명을 마친다.

지금까지 다음과 같은 것에 대해 알아보았다.

- JEUS Web 의 개요.
- JEUS Web Container 의 개요, 설정, 운영과 튜닝.
- Web Container 의 DB Connection Pool.
- JEUS Web context group.
- Web Server 연결 (WebtoB, Apache, HTTP 리스너, TCP 리스너, 보안 리스너).
- 클러스터링된 환경에서의 Session 데이터 Tracking.
- JEUS Webcontext (Web application/WAR 모듈).
- 가상 호스트.

JEUS Web 기술에 대한 더 자세한 정보가 필요하다면 JEUS Server 안내서, JEUS Web Server 안내서 또는 필요에 따라 JEUS Web Service 안내서를 참고하라.

이 문서의 나머지 부분은 JEUS Web 기술의 주된 참고 문서로의 역할을 할 수 있는 부록 문서로 구성되어 있다.

A Webadmin 콘솔 툴 레퍼런스

A.1 소개

해당 부록은 webadmin Console Tool 에 대하여 보다 상세히 다룬다.

A.2 목적

webadmin Console Tool 의 목적은 관리자(administrator)가 명령 프롬프트(터미널창)에서 Web Container 를 직접 다루는데 있다.

Webadmin tool 로 다음과 같은 작업을 수행할 수 있다:

- 구동중인 Web Container 에서 Context group, Context, 개별 Servlet deploy 를 제어한다.
- Web Container, Context group, Context 그리고 Servlet 의 상태를 모니터링한다.
- 동적으로 구성 정보를 가져오거나 설정할 수 있다(즉, XML 설정파일 수정 없이).

참조: 해당 툴의 사용으로 Web Container 자체를 구동하거나 중지할 수는 없다. Web Container 의 구동과 중지는 jeusadmin Tool(또는 웹 관리자)을 사용해야 한다. 보다 많은 정보는 JEUS Server 안내서를 참조한다.

A.3 실행

webadmin 은 제우스 인스톨 디렉토리 밑의 bin 디렉토리에 위치한 독립 콘솔 프로그램이다. 보통 해당 프로그램(bin 디렉토리 밑에 있는 모든 실행 가능한 프로그램들)은 환경변수의 PATH 에 경로가 추가되어 있으므로 OS 내의 어떤 위치에서도 실행되어 진다.

webadmin 은 다음과 같이 실행을 할 수 있으며, []안 부분은 옵션을 나타낸다.

```
webadmin nodename | web_container_name [-U username -P password]  
[command [arguments]]
```

파라미터는 다음과 같다.

node_name

구동 중인 Web Container 를 관리하기 위해 연결한 JEUS 노드를 지정한다. 엔진(들)을 제어하기 위해서는 tool 내에서 엔진선택을 해야 한다(특정엔진 선택 명령어는 이후로 논하기로 함).

web_container_name

연결할 단일 Web Container 의 JNDI 이름을 지정한다.

-U username

만일 직접 webadmin 명령을 실행하기를 원하거나 Tool 의 인증 프롬프트를 거치지 않고 로그인을 하려고 할 때 적합한 관리자의 사용자명을 지정한다.

-P password

만일 -U username 을 사용했다면, 해당 사용자명에 대응되는 패스워드를 지정한다.

Command [arguments]

Tool 의 시작과 함께 수행하고 싶은 명령어를 입력하면 된다(프롬프트가 나타나지 않음). 여기서 사용 가능한 명령어들은 다음 절에서 기술된다.

예제:

```
webadmin mynode_container1
```

Tool 이 시작된 후 사용자를 확인하는 절차를 거치게 되는데, 이 때 적합한 사용자명과 패스워드를 넣어 로그인을 하면된다. 여기까지 수행을 하고 나면 사용자의 명령을 받아 들일 수 있는 프롬프트 상태가 되는데, 이 상태에서 본 절에서 기술된 명령어를 입력하면 된다.

‘help’를 치면 도움말 리스트가 나온다.

A.4 엔진 리스트와 선택 명령

본 카테고리의 명령어들은 현재 webadmin Console Tool 로 연결한 JEUS 노드에 연결되어 있는 Web Container 들의 리스트를 보거나 제어할 엔진을 선택할 수 있다.

```
englist [all | current]
```

현재 JEUS 노드에서 구동중인 “all” 또는 “current”(현재 선택) Web Container(들)를 나열한다. 디폴트는 “all”이다.

```
engselect all | X [Y [Z [. . .]]]
```

Web Container 내의 여러 엔진들 중 상태를 살펴보거나 제어할 엔진을 설정하는 명령어로 “all”은 Web Container 내 모든 엔진을 선택하는 옵션이며, 각각의 엔진을 선택하기 위해서는 ‘englist’명령어를 수행했을 때 보여지는 엔진의 고유 번호들을 나열하면 된다.

A.5 제어 명령

다음 명령어들은 webadmin 프롬프트상에서 현재 실행중인 Web Container(들)의 Context group, Context, 각각의 Servlet 을 제어하기 위해 사용된다.

```
reload <context group name> [<context name> [<Servlet name>]]
```

디스크로부터 지정한 엔티티를 reload 한다. 이미 로드된 Servlet(JSP)들과 클래스들을 reload 한다. 이 명령어는 Servlet 이나 class 의 reload 기능이 비활성화되어 있을 경우 유용하게 사용할 수 있다(enable-reload, check-class-reload).

```
restart <context group name> [<context name> <Servlet name>]]
```

지정한 엔티티를 restart 한다. 이 명령어는 본질적으로 ‘terminate’명령 다음에 오는 “deploy”명령이다. Runtime 시에 추가된 웹 모듈의 경우에는 해당 command 가 적용되지 않는다.

```
suspend <context group name> [<context name> [<Servlet name>]]
```

지정한 엔티티를 중지하여 클라이언트가 이용할 수 없게 한다.

```
resume <context group name> [<context name> [<Servlet name>]]
```

중지된 엔티티를 재개한다.

```
terminate <context group name> [<context name> [<Servlet name>]]
```

“undeploy”를 수행함으로써 지정한 엔티티를 종료한다. 이것은 엔티티가 실행중인 메모리로부터 완전히 제거되는걸 의미한다.

설정파일이나 다른 파일들은 해당 작업에 영향을 주지 않는다. Runtime 시에 추가된 웹 모듈의 경우에는 해당 command가 적용되지 않는다.

```
clr_stat <context group name> [<context name> [<Servlet name>]]
```

지정한 엔티티에 대하여 요청 카운터를 재설정한다.

A.6 상태 정보 명령어

다음 명령어들은 webadmin 프롬프트상에서 현재 운영중인 Web Container에 대하여 자세한 실행 정보를 가져오기 위해 쓰여진다.

```
info [<context group name>] [<context name>]
```

deploy된 Context group과 Context들에 대하여 일반적인 정보를 출력한다.

```
st -m | -t | -r | -s | -d
```

지정한 파라미터에 해당하는 정보를 가져온다.

-m

구동중인 Web Container에서 현재 JVM에서 사용하고 있는 메모리의 상태를 보여준다.

-t

Web Container에 속해있는 각각의 listener에 대한 thread pool의 상태를 출력한다. 보여지는 정보는 현재 할당된 worker thread의 수와 wait-queue에 적체된 클라이언트의 수, 그리고 thread pool을 유지하는 최대 thread의 개수를 포함한다.

-r

각 Context 에 대해 처리된 request 의 누적 수치와 평균 처리시간을 출력한다.

-s

Web Container 에 구성된 세션서버 커넥션 풀에 대한 정보를 출력한다. 정보는 현재 연결되어 있는 세션서버의 이름, 전체 세션의 수, pool 에서 사용된 세션서버와의 커넥션 수를 포함한다.

-d

Web Container 에 구성된 DB 커넥션 풀에 대한 정보를 출력한다.

-w

WebT 커넥션 풀에 대한 정보를 출력한다.

ti -l <listener_id> | -r | -a

listener 의 thread pool 과 각 worker thread 에 대한 정보를 가져온다.

-l

지정된 listener(WEBMain.xml 에 설정된 listener ID 를 기입한다)에 대한 정보를 가져온다. 이 명령어는 주어진 listener 의 thread pool 내에 있는 모든 worker thread 들의 현재 상태와 축적된 idle-time(wt)을 나열한다.

-r

Web Container 에 구성되어 있는 listener 내의 모든 worker thread 가 처리한 정보를 출력한다. 그 정보는 이전 “ti” 명령 이후 해당 thread 의 요청 개수(reqs), 각 worker thread 에서 처리한 요청에 대한 평균 실행 시간(avgTime), thread 에 의해 처리된 요청의 전체 누적치(“total”)를 포함한다.

-a

-l 과 -r 옵션을 합친 결과를 출력한다. 또한 worker thread 의 상태를 바꾼 지난 시간을 time-stamp 로 표시한다(예를 들어 thread 의 상태가 “idle”에서 “active”로 바뀌거나 그 반대의 경우).

di -p <pool_id>

특정 <pool-id>의 DB 커넥션 풀에 대한 자세한 정보를 출력한다.

A.7 구성 리포트 명령어

다음 명령어들은 webadmin console tool 의 프롬프트 상에서 Web Container 에 대한 설정 정보를 가져오기 위해 쓰여진다.

다음 명령어들이 실행될 때 보여지는 구성 항목과 필드들에 대한 정보는 부록 D 에 WEBMain.xml 구성 레퍼런스를 참조한다.

```
cfg -cg: <context group name> |
    -c: <context name> |
    -vh: <Virtual Host> |
    -cn: <listener_id> |
    -db: <DB con. pool name> |
    -log |
    -sc |
    -rh
```

“cfg”명령은 주어진 엔티티에 대한 최근 설정을 가져온다.

-cg

주어진 Context group 에 대한 모든 설정내용을 출력한다.

-vh

주어진 가상호스트에 대한 모든 설정내용을 출력한다.

-c

주어진 Context 에 대한 모든 설정내용을 출력한다.

-cn

주어진 listener 에 대한 모든 설정내용을 출력한다. listener 는 WEBMain.xml 에 설정한 listener ID 를 통해 식별된다.

-db

주어진 Web Container 에 종속된 DB 커넥션 풀에 대한 설정정보가 출력된다. Pool 은 WEBMain.xml 에 설정된 pool name 을 통해 식별된다.

-log

로그 설정에 대한 정보를 출력한다.

-sc

세션 클러스터링(세션서버등)에 대한 정보를 출력한다.

-rh

응답 헤더 구성들의 설정을 출력한다.

참조: 위에서 열거한 옵션들은 서로 조합하여 사용할 수 있다(예: "cfg -cg:MyGroup -log" 명령은 "MyGroup" 이름의 Context group 에 대한 로그 데이터를 출력한다).

A.8 구성 설정 명령어

다음 명령어들은 webadmin 프롬프트 상에서 Web Container 에 대한 설정을 변경하기 위해 사용된다. 주의할 점은 여기서 변경된 정보는 운영되고 있는 상태에만 적용되며 설정 파일에는 반영이 되지 않는다. 그러므로 다음 구동 시에도 적용이 되게 하기 위해서는 반드시 설정파일을 변경하여야 한다.

```
set -cg: <context group name> [-c: <context name>] -<setting> <value>
```

Context group 또는 Context 항목을 설정한다. 다음은 <항목>/<값>을 공백으로 구분하여 하나의 짝으로 설정한다.

-ar true | false autoreload 기능을 작동시키거나 끄

```
set -cg: <context group name> -<setting> <value>
```

Context group 에서만 사용되는 항목을 설정한다. 다음은 <항목>/<값>을 공백으로 구분하여 하나의 짝으로 설정한다.

-stm <minutes> session timeout 설정

-ell <level> error log level 설정

```
set -<setting> <value>
```

Context group/context 항목을 설정한다. 다음은 <설정>/<값>을 공백으로 구분하여 하나의 짝으로 설정한다.

-to webtob-listener 인 경우 해당 웹서버(index - main: 0, backup1: 1, bakcup2: 2, ...)로 서비스 전환 설정을 한다.

set -cn: <listener id> -<setting> <value>

thread pool/Web listener 항목을 설정한다. 다음은 <설정>/<값>을 공백으로 구분하여 하나의 짝으로 설정한다.

-tmin <number>	최소 thread 를 설정한다.
-tmax <number>	최대 thread 를 설정한다.
-trate <number>	thread 의 변환비율을 설정한다.
-tidle <msec>	thread 의 idle-time 을 설정한다.
-tmwq <number>	대기하는 queue 의 최대 사이즈를 설정한다.
-tmq <number>	queue 의 최대 사이즈를 설정한다.
-tmtat <msec>	thread 의 최대 액티브 시간을 설정한다.
-tnt <number rate>	notify threshold 를 설정한다.
-trt <number rate>	restart threshold 를 설정한다.

set -db: <DB pool name> -<setting> <value>

DB 풀 항목을 설정한다. 다음은 <설정>/<값>을 공백으로 구분하여 하나의 짝으로 설정한다.

-dmin <number>	최소 connection 을 설정한다.
-dmax <number>	최대 connection 을 설정한다.
-drate <number>	connection 의 증가비율을 설정한다.
-diidle <msec>	connection 의 idle-time 을 설정한다.
-clac <true false>	CloseLongActiveConnection 을 설정한다.
-dmat <msec>	MaxActiveTime 을 설정한다.
-di <true false>	Dynamic Increment 을 설정한다.
-muc <count>	MaxUseCount 를 설정한다.
-ctos <seconds>	ConnectionTimeoutSecs 를 설정한다.

set -<setting> <value>

Web Container 레벨에서(최상위 레벨) 적용되는 설정을 정의한다. 모니터링, 세션서버를 밑에 기술한 대로 설정할 수 있다.

모니터링 설정:

-tmi <msec>	Thread pool 모니터링 주기 설정.
-smi <msec>	세션 모니터링 주기 설정.
-dmi <msec>	DB pool 모니터링 주기 설정.
-rmi <msec>	Servlet Reload 주기 설정.

```
set -vh: <virtual host name> -<setting> <value>
```

앞에서 기술한 “set -cg :<context group name>”을 참조한다.

A.9 기타 명령어

다음과 같은 여러 명령어가 존재한다.

hel p | h

도움말 프린트.

p | !

이전에 수행한 명령을 수행한다.

qui t | q

webadmin 을 종료한다.

B jspc 콘솔 툴 레퍼런스

B.1 소개

JSP 배치 컴파일러는 “처음 접속”시 성능향상을 위해 JSP 파일을 사전에 컴파일 하는데 사용되는 Tool 이다.

B.2 목적

JSP 파일은 보통 웹 브라우저로부터 JSP 파일에 대한 요청이 Web Container 에 올 때 컴파일된다. 그리하여 JSP 파일에 대한 초기 요청은 많은 파싱과 컴파일 작업을 통해 많은 리소스를 소모하여 응답시간이 길어진다. 만일 개발된 JSP 파일이 많고 사용자 요구가 빈번한 웹사이트이면 “초기 요청 수행”에 소요되는 시간이나 리소스가 서비스에 크게 영향을 미칠 수 있다.

이러한 경우 JSP 배치 컴파일러를 사용함으로써, 개발된 JSP 소스 파일들을 Web Container 가 시작하기에 앞서 컴파일 함으로써 이러한 문제를 해결할 수 있다.

JSP 배치 컴파일러는 JEUS_HOME\bin 디렉토리 아래에 위치한 “jspc”명령어를 통해 수행된다.

B.3 실행

호출 문법은 다음과 같다:

```
j spc ( -e engine_name -U user id -P password [-g grpname  
        [-c ctxname [-p path | -l filename]] [-noxml] [-q]  
        [-f container_xml_path] [-co javac_compiler_options] )  
| ( -h | -v )
```

해당 명령어의 파라미터는 다음 섹션에서 설명된다.

참조:괄호”()”,브라켓”[]” 그리고 파이프”|”는 오직 그룹화와 문법을 목적으로 있다.

B.4 파라미터

“jspc”명령에 파라미터는 다음과 같다.

-e engi ne_name

JSP 파일들이 사전컴파일 되어야 할 Web Container 의 이름.

-g context_group_name

JSP 파일들이 컴파일 되어야 할 Context group 의 이름.

-c context_name

JSP 파일들이 컴파일 될 Context 의 이름.만일 해당 파라미터와 Context group 이름 파라미터가 빠지면 지정한 Web Container 의 모든 JSP 파일들이 컴파일된다.만일 해당 파라미터가 빠지고 Context group 이름 파라미터가 포함되면 지정한 Context group 의 모든 JSP 파일들이 컴파일된다.

-p path

컴파일 할 하나의 Context 경로.(‘/’로 반드시 시작한다).해당 옵션은 -l 옵션과 결합할 수 없다.

-l filename

지정한 파일 이름은 컴파일 할 JSP 파일 리스트를 포함하고 있다. Linefeed 문자는 목록에 나열된 각각의 JSP 파일을 분리해야 한다. JSP 파일 이름들은 context doabase 에 관해 절대 경로를 가져야 하며 반드시 “/”로 시작해야 한다.

예:

```
/jsp/directive/directive.jsp <linefeed>
/jsp/forward/forward.jsp.
```

해당 옵션은 -p 옵션과 결합할 수 없다.

-q

이 옵션을 지정하게 되면 컴파일 에러가 발생할 경우에 수행을 멈추게 된다.

-co javac_compiler_options

javac 컴파일러에 관련된 파라미터. 설정 값은 구분자를 ‘,’로 하여 설정하며 실제 컴파일시에 공백으로 대체된다.

-h

도움말을 보여준다.

-v

버전정보를 보여준다.

예제:

```
jspc -e johan_servlet_engine1 -g MyGroup -c Examples
```

위 명령은 “johan_servlet_engine1” Web Container 에 “MyGroup” Context group 아래에 “Examples” Context 에 모든 JSP 파일들을 사전컴파일한다.

C jspc2 콘솔 툴 레퍼런스

C.1 소개

jspc2 배치 컴파일러는 JSP 파일을 미리 컴파일 해 둬으로써, JSP 파일을 처음 요청할 때 성능을 향상시키기 위한 툴이다.

이 툴은 앞서 설명한 jspc 툴과 비슷하나, 몇 가지 추가 기능을 가지고 있다.

C.2 목적

JSP 파일은 보통 Web Container가 웹브라우저로부터 JSP 파일의 요청을 처음 받았을 때 컴파일된다. 이러한 JSP 파일의 첫번째 요청은 파싱과 컴파일 과정을 거치므로, 많은 양의 리소스를 사용하고 응답시간도 길어지게 된다. 만일 개발된 JSP 파일이 많고 사용자 요구가 빈번한 웹사이트이면 “초기 요청 수행”에 소요될 시간은 상당히 크게 될 것이다.

JSP 배치 컴파일러 2를 사용하면 JSP 소스 파일을 Web Container가 시작하기 전에 미리 컴파일 함으로써 이러한 문제를 없앨 수 있다.

이 JSP 배치 컴파일러 2는 JEUS_HOME\bin\ 디렉토리 아래에 있는 jspc2 명령을 통해서 수행된다.

jspc2는 미리 컴파일 된 JSP 파일을 기본으로 하는 Servlet-mapping 테이블을 생성한다. Servlet-mapping 테이블은 jeus_jspmap.xml 파일에 저장된다. jeus_jspmap.xml 파일은 WEB-INF 디렉토리에 위치하고, 생성된 Servlet 클래스 파일(미리 컴파일된 JSP 파일)은 WEB-INF/classes에 위치하거나 또는 WEB-INF/lib에 압축되어 저장된다. jeus_jspmap.xml 파일은 Web Container에 미리 컴파일된 JSP 파일에 대한 Servlet mapping을 자동으로 추가한다.

C.3 호출

호출 문법은 다음과 같다.

```
j spc2 (-mode {normal | war | map}
        -ctxroot {<context root> | <war file path>}
        [-srcdir <srcdir or srcfile>]
        [-workdir <working directory>]
        [-outdir <output directory>]
        [-mapdir <jeus_jspmap.xml output dir>]
        [-q] [-co "javac options"]
        [-classpath <extra classpath>]) |
        ([-h] | [-v])
```

해당 명령어의 파라미터는 다음 절에서 설명된다.

참고: 위의 문법 중 괄호”()”, 사각 브라켓”[]”, 삼각 브라켓”<>” 그리고 파이프”|”는 오직 그룹화와 문법 설명을 위한 것이다. 그것들은 명령어에 포함되지 않는다.

C.4 파라미터

“jspc2”명령 파라미터는 다음과 같다.

-mode {normal | war | map}

normal : 해당 context 의 JSP 파일들을 미리 컴파일한다. 만약 -srcdir 옵션인 생략되어 있다면 <ctxroot> 내의 모든 JSP 파일들을 컴파일 한다. -mapdir 옵션과 -outwar 옵션은 사용되지 않는다. 만약 workdir 옵션값이 생략되어 있으면 디폴트 ./jspwork 디렉토리 밑에 해당 서블릿 소스가 생성되면 -outdir 옵션값이 생략되어 있으면 디폴트 ./jspwork 디렉토리 밑에 해당 컴파일 클래스들이 생성된다.

war: 해당 context 의 JSP 파일들을 미리 컴파일한다. 만약 -outwar 옵션인 생략되어 있다면 디폴트 ./jspcout.war 파일로 생성된다. -srcdir, -mapdir, -workdir 과 -outdir 옵션은 사용되지 않는다

map: 해당 context 의 JSP 파일과 jeus_jspmap.xml 에 servlet-mapping 정의된 JSP 에 대해서 컴파일 작업을 수행한다.. 만약 -srcdir 옵션인 생략되어 있다면 <ctxroot> 내의 모든 JSP 파일들을 컴파일 한다. -mapdir 옵션이 생략되어 있다면 <ctxroot>/WEB-INF 가 default map 디렉토리 경로이며, -workdir 옵션이 생략되어 있다면 디폴트 working 디렉토리 경로는 <ctxroot>/WEB-INF/classes 이다. -outdir 경로가 생략

되어 있으면 output 디렉토리 경로는 <ctxroot>/WEB-INF/classes 이다.
-outwar 옵션은 사용되지 않는다.

-ctxroot <context root>

JSP 파일들이 미리 컴파일 되어야 할 컨텍스트의 root 디렉토리 경로이다.

-srcdir <srcdir or srcfile>

JSP 파일들이 포함되어 있는 디렉토리 또는 JSP 파일의 이름을 지정한다. 이 옵션으로 미리 컴파일 할 JSP 파일을 알 수 있다. 만약 이 옵션을 설정하지 않으면 해당하는 컨텍스트의 모든 JSP 파일들이 컴파일 된다. srcdir 과 srcfile 은 “/”로 시작하여 컨텍스트에 상대 경로로 지정한다.

-workdir <working directory>

이는 java 파일들이 위치할 디렉토리의 경로이다. 이 옵션이 없을 경우 <context root>/WEB-INF/classes 가 사용된다.

-outdir <output directory>

컴파일된 JSP 파일의 class 파일들이 위치할 디렉토리이다. 이 옵션을 사용하지 않으면 <context root>/WEB-INF/classes 가 사용된다.

-mapdir <jeus_jspmap.xml output dir>

jeus_jspmap.xml 파일이 위치할 디렉토리를 지정한다. 만약 이 옵션을 설정하지 않으면 <context root>/WEB-INF/ 가 사용된다.

-q

“-q” 옵션을 설정하면 컴파일 에러가 발생했을 경우 jspc2 실행이 중지된다.

-co javac_compiler_options

javac 컴파일러에 적용할 인자를 정한다. 공백은 ‘,’(콤마)로 대체한다.

-h

Help 메시지를 보여준다.

-v

version 정보를 보여준다.

`-classpath <extra classpath>`

JSP/Servlet 파일들을 컴파일 하는데 필요한 `classpath` 를 추가하는 속성이다. 형식은 `javac` 의 `-classpath` 옵션과 동일하다.

예제:

```
jspc -ctxroot c:\myapp\myctx -sourcedir /jsps -workdir c:\myapp\src
```

위의 명령은 `c:\myapp\myctx` 컨텍스트 아래있는 `jsps` 디렉토리의 모든 JSP 파일들을 미리 컴파일 한다. 이때 `c:\myapp\src` 를 작업 디렉토리로 이용한다(여기에는 `.java` 파일들이 위치한다). 생성된 Servlet 클래스 파일들은 `c:\myapp\myctx\WEB-INF\classes` 에 위치하고, `jeus_jspmap.xml` 파일은 `c:\myapp\myctx\WEB-INF` 에 위치한다.

D WEBMain.xml XML 구성 레퍼런스

D.1 소개

본 부록의 레퍼런스는 WEBMain.xml(핵심 컨테이너 구성파일)의 모든 XML 태그에 대해서 설명하고 있다. 이 파일의 XSD 파일은 “JEUS_HOME\config\xsds\”디렉토리의 “web-main-config.xsd”파일이다.

본 레퍼런스는 3 부분으로 나뉘어 있다.

1. **XML Schema/XML 트리:** XML 설정 파일의 모든 태그 리스트를 정리했다. 각 노드의 형식은 다음과 같다.
 - a. 태그 레퍼런스로 빨리 찾아보기 위해서 각 태그마다 인덱스 번호(예 (11))를 붙여놓았다. 태그 레퍼런스에서는 이 번호 순서로 설명한다.
 - b. XML Schema 에서 정의한 XML 태그명을 <tag name> 형식으로 표시한다.
 - c. XML Schema 에서 정의한 Cardinality 를 표시한다. “?” = 0 개나 1 개의 element, “+” = 1 개 이상의 element, “*” = 0 개 이상의 element, (기호가 없음) = 정확히 1 개의 element
 - d. 몇몇 태그에는 “P” 문자를 붙여놓았는데, 해당 태그는 성능에 관계되는 태그라는 것을 뜻한다. 이 태그는 설정을 튜닝할 때 사용된다.
2. **태그 레퍼런스:** 트리에 있는 각 XML 태그를 설명한다.
 - a. **Description:** 태그에 대한 간단한 설명
 - b. **Value Description:** 입력하는 값과 타입
 - c. **Value Type:** 값의 데이터 타입. 예: String
 - d. **Default Value:** 해당 XML 을 사용하지 않았을 때 기본적으로 사용되는 값.

- e. **상수:** 이미 정해져 있는 값.
- f. **Example:** 해당 XML 태그에 대한 Example.
- g. **Performance Recommendation:** 성능 향상을 위해서 추천하는 값.
- h. **Child Elements:** 자신의 태그 안에 사용되는 태그.

3. **Example XML** 파일: “WEBMain.xml”에 대한 완전한 예제.

D.2 XML Schema/XML Tree

- (1) <web-container>
- (2) <monitoring>?
- (3) <check-thread-pool>? P
- (4) <check-class-reload>? P
- (5) <check-session>? P
- (6) <check-db-connection-pool>? P
- (7) <redirect-stdout>? P
- (8) <redirect-stderr>? P
- (9) <shutdown-timeout>? P
- (10) <context-group>+
- (11) <group-name>
- (12) <group-docbase>?
- (13) <servlet-home>?
- (14) <virtual-host>*
- (15) <virtual-host-name>

- (16) <host-list>+
- (17) <webserver-connection>
- (18) <webtob-listener>
- (19) <listener-id>
- (20) <port>
- (21) <output-buffer-size>? P
- (22) <thread-pool>
- (23) <min>
- (24) <max>?
- (25) <step>? P
- (26) <max-idle-time>? P
- (27) <max-wait-queue>? P
- (28) <max-queue>? P
- (29) <thread-state-notify>?
- (30) <max-thread-active-time>?
- (31) <notify-threshold>?
- (32) <restart-threshold>?
- (33) <notifier-id>?
- (34) <notify-subject>?
- (35) <restart-subject>?
- (36) <postdata-read-timeout>? P
- (37) <scheme>? P

- (38) <hth-count>? P
- (39) <request-prefetch>? P
- (40) <disable-pipe>? P
- (41) <webtob-address>? P
- (42) <registration-id>? P
- (43) <webtob-home>?
- (44) <read-timeout>? P
- (45) <reconnect-timeout>? P
- (46) <webtob-backup>*
- (47) <port>
- (48) <output-buffer-size>? P
- (49) <thread-pool>?
- (50) <min>
- (51) <max>?
- (52) <step>? P
- (53) <max-idle-time>? P
- (54) <max-wait-queue>? P
- (55) <max-queue>? P
- (56) <thread-state-notify>?
- (57) <max-thread-active-time>?
- (58) <notify-threshold>?
- (59) <restart-threshold>?

- (60) <notifier-id>?
- (61) <notify-subject>?
- (62) <restart-subject>?
- (63) <postdata-read-timeout>? P
- (64) <scheme>? P
- (65) <hth-count>? P
- (66) <disable-pipe>? P
- (67) <webtob-address>? P
- (68) <registration-id>? P
- (69) <webtob-home>?
- (70) <read-timeout>? P
- (71) <reconnect-timeout>? P
- (72) <apache-listener>
 - (73) <listener-id>
 - (74) <port>
 - (75) <output-buffer-size>? P
 - (76) <thread-pool>
 - (77) <min>
 - (78) <max>?
 - (79) <step>? P
 - (80) <max-idle-time>? P
 - (81) <max-wait-queue>? P

- (82) <max-queue>? P
- (83) <thread-state-notify>?
 - (84) <max-thread-active-time>?
 - (85) <notify-threshold>?
 - (86) <restart-threshold>?
 - (87) <notifier-id>?
 - (88) <notify-subject>?
 - (89) <restart-subject>?
- (90) <postdata-read-timeout>? P
- (91) <scheme>? P
- (92) <back-log>? P
- (93) <busy-notify-page>?
- (94) <server-access-control>? P
- (95) <allowed-server>*
- (96) <http-listener>
 - (97) <listener-id>
 - (98) <port>
 - (99) <output-buffer-size>? P
 - (100) <thread-pool>
 - (101) <min>
 - (102) <max>?
 - (103) <step>? P

- (104) <max-idle-time>? P
- (105) <max-wait-queue>? P
- (106) <max-queue>? P
- (107) <thread-state-notify>?
 - (108) <max-thread-active-time>?
 - (109) <notify-threshold>?
 - (110) <restart-threshold>?
 - (111) <notifier-id>?
 - (112) <notify-subject>?
 - (113) <restart-subject>?
- (114) <postdata-read-timeout>? P
- (115) <scheme>? P
- (116) <back-log>? P
- (117) <server-access-control>? P
- (118) <allowed-server>*
- (119) <http-version>?
- (120) <read-timeout>? P
- (121) <tcp-listener>
 - (122) <listener-id>
 - (123) <port>
 - (124) <output-buffer-size>? P
 - (125) <thread-pool>

- (126) <min>
- (127) <max>?
- (128) <step>? P
- (129) <max-idle-time>? P
- (130) <max-wait-queue>? P
- (131) <max-queue>? P
- (132) <thread-state-notify>?
 - (133) <max-thread-active-time>?
 - (134) <notify-threshold>?
 - (135) <restart-threshold>?
 - (136) <notifier-id>?
 - (137) <notify-subject>?
 - (138) <restart-subject>?
- (139) <postdata-read-timeout>? P
- (140) <scheme>? P
- (141) <back-log>? P
- (142) <server-access-control>? P
- (143) <allowed-server>*
- (144) <dispatcher-config-class>?
- (145) <udp-listener>
 - (146) <listener-id>
 - (147) <port>

(148) <input-buffer-size>? P

(149) <output-buffer-size>? P

(150) <thread-pool>

 (151) <min>

 (152) <max>?

 (153) <step>? P

 (154) <max-idle-time>? P

 (155) <max-wait-queue>? P

 (156) <max-queue>? P

 (157) <thread-state-notify>?

 (158) <max-thread-active-time>?

 (159) <notify-threshold>?

 (160) <restart-threshold>?

 (161) <notifier-id>?

 (162) <notify-subject>?

 (163) <restart-subject>?

(164) <postdata-read-timeout>? P

(165) <scheme>? P

(166) <back-log>? P

(167) <server-access-control>? P

(168) <allowed-server>*

(169) <dispatcher-config-class>?

(170) <secure-listener>

(171) <listener-id>

(172) <port>

(173) <output-buffer-size>? P

(174) <thread-pool>

(175) <min>

(176) <max>?

(177) <step>? P

(178) <max-idle-time>? P

(179) <max-wait-queue>? P

(180) <max-queue>? P

(181) <thread-state-notify>?

(182) <max-thread-active-time>?

(183) <notify-threshold>?

(184) <restart-threshold>?

(185) <notifier-id>?

(186) <notify-subject>?

(187) <restart-subject>?

(188) <postdata-read-timeout>? P

(189) <scheme>? P

(190) <back-log>? P

(191) <server-access-control>? P

- (192) <allowed-server>*
- (193) <http-version>?
- (194) <read-timeout>? P
- (195) <ajp13-listener>
 - (196) <listener-id>
 - (197) <port>
 - (198) <output-buffer-size>? P
 - (199) <thread-pool>
 - (200) <min>
 - (201) <max>?
 - (202) <step>? P
 - (203) <max-idle-time>? P
 - (204) <max-wait-queue>? P
 - (205) <max-queue>? P
 - (206) <thread-state-notify>?
 - (207) <max-thread-active-time>?
 - (208) <notify-threshold>?
 - (209) <restart-threshold>?
 - (210) <notifier-id>?
 - (211) <notify-subject>?
 - (212) <restart-subject>?
- (213) <postdata-read-timeout>? P

- (214) <scheme>? P
- (215) <back-log>? P
- (216) <busy-notify-page>?
- (217) <server-access-control>? P
- (218) <allowed-server>*
- (219) <tmax-listener>
 - (220) <listener-id>
 - (221) <port>
 - (222) <output-buffer-size>? P
 - (223) <thread-pool>
 - (224) <min>
 - (225) <max>?
 - (226) <step>? P
 - (227) <max-idle-time>? P
 - (228) <max-wait-queue>? P
 - (229) <max-queue>? P
 - (230) <thread-state-notify>?
 - (231) <max-thread-active-time>?
 - (232) <notify-threshold>?
 - (233) <restart-threshold>?
 - (234) <notifier-id>?
 - (235) <notify-subject>?

- (236) <restart-subject>?
- (237) <postdata-read-timeout>? P
- (238) <reconnect-timeout>? P
- (239) <tmax-address>? P
- (240) <server-group-name>
- (241) <server-name>
- (242) <tmax-backup-address>? P
- (243) <tmax-backup-port>?
- (244) <dispatcher-config-class>?
- (245) <session-config>?
 - (246) <timeout>? P
 - (247) <shared>? P
 - (248) <persistent>? P
- (249) <print-error-to-browser>? P
- (250) <encoding>?
 - (251) <request-encoding>?
 - (252) <default>?
 - (253) <forced>?
 - (254) <response-encoding>?
 - (255) <default>?
 - (256) <forced>?
- (257) <postdata-encoding>?

(258) <default>?

(259) <forced>?

(260) <jsp-engine>?

(261) <keep-generated>? P

(262) <java-compiler>? P

(263) <jsp-work-dir>?

(264) <compile-output-dir>?

(265) <compile-option>?

(266) <compile-encoding>?

(267) <check-included-jspfile>? P

(268) <logging>?

(269) <user-log>?

(270) <level>? P

(271) <use-parent-handlers>? P

(272) <filter-class>?

(273) <handler>?

(274) <console-handler>

(275) <name>

(276) <level>? P

(277) <encoding>?

(278) <filter-class>?

(279) <file-handler>

(280) <name>

(281) <level>? P

(282) <encoding>?

(283) <filter-class>?

(284) <file-name>?

(285) <valid-day>

(286) <valid-hour>

(287) <buffer-size>? P

(288) <append>? P

(289) <smtp-handler>

(290) <name>

(291) <level>? P

(292) <encoding>?

(293) <filter-class>?

(294) <smtp-host-address>

(295) <from-address>

(296) <to-address>

(297) <cc-address>?

(298) <bcc-address>?

(299) <send-for-all-messages>? P

(300) <socket-handler>

(301) <name>

- (302) <level>? P
- (303) <encoding>?
- (304) <filter-class>?
- (305) <address>
- (306) <port>
- (307) <user-handler>
- (308) <handler-class>
- (309) <name>
- (310) <level>? P
- (311) <encoding>?
- (312) <filter-class>?
- (313) <handler-property>?
- (314) <property>*
- (315) <key>
- (316) <value>
- (317) <formatter-class>? P
- (318) <formatter-property>?
- (319) <property>*
- (320) <key>
- (321) <value>
- (322) <access-log>?
- (323) <enable>? P

- (324) <enable-analysis>? P
- (325) <level>? P
- (326) <use-parent-handlers>? P
- (327) <filter-class>?
- (328) <handler>?
- (329) <console-handler>
 - (330) <name>
 - (331) <level>? P
 - (332) <encoding>?
 - (333) <filter-class>?
- (334) <file-handler>
 - (335) <name>
 - (336) <level>? P
 - (337) <encoding>?
 - (338) <filter-class>?
 - (339) <file-name>?
 - (340) <valid-day>
 - (341) <valid-hour>
 - (342) <buffer-size>? P
 - (343) <append>? P
- (344) <smtp-handler>
 - (345) <name>

(346) <level>? P

(347) <encoding>?

(348) <filter-class>?

(349) <smtp-host-address>

(350) <from-address>

(351) <to-address>

(352) <cc-address>?

(353) <bcc-address>?

(354) <send-for-all-messages>? P

(355) <socket-handler>

(356) <name>

(357) <level>? P

(358) <encoding>?

(359) <filter-class>?

(360) <address>

(361) <port>

(362) <user-handler>

(363) <handler-class>

(364) <name>

(365) <level>? P

(366) <encoding>?

(367) <filter-class>?

(368) <handler-property>?

(369) <property>*

(370) <key>

(371) <value>

(372) <formatter-class>? P

(373) <formatter-property>?

(374) <property>*

(375) <key>

(376) <value>

(377) <format>?

(378) <session-cluster>?

(379) <session-routing>? P

(380) <shared>? P

(381) <session-server>?

(382) <server-name>

(383) <max>

(384) <min>?

(385) <step>? P

(386) <backup-server-name>?

(387) <connect-timeout>? P

(388) <read-timeout>? P

(389) <response-header>?

- (390) <session-id-cookie>?
- (391) <jsessionid-name>? P
- (392) <version>? P
- (393) <domain>?
- (394) <path>? P
- (395) <max-age>? P
- (396) <secure>? P
- (397) <custom-header>?
- (398) <header-field>*
- (399) <field-name>
- (400) <field-value>
- (401) <db-connection-pool>*
- (402) <pool-name>
- (403) <pool-type>? P
- (404) <pooling-rule>? P
- (405) <connection-url>
- (406) <driver-class-name>
- (407) <connection-argument>
- (408) <listener-id>*
- (409) <check-query>?
- (410) <dynamic-increment>? P
- (411) <get-connection-timeout>? P

- (412) <read-timeout>? P
- (413) <close-long-active-connection>? P
- (414) <max-active-time>? P
- (415) <max-use-count>? P
- (416) <login-delay>? P
- (417) <close-delay>? P
- (418) <fetch-size>? P
- (419) <db-pool-control>?
 - (420) <min>? P
 - (421) <max>
 - (422) <step>? P
 - (423) <max-idle-time>? P
 - (424) <max-alive-time>? P
- (425) <dba-connection>?
 - (426) <forced-close>? P
 - (427) <connection-url>
 - (428) <connection-argument>
- (429) <encryption>? P
- (430) <session-cluster>?
 - (431) <session-routing>? P
 - (432) <shared>? P
 - (433) <session-server>?

- (434) <server-name>
- (435) <max>
- (436) <min>?
- (437) <step>? P
- (438) <backup-server-name>?
- (439) <connect-timeout>? P
- (440) <read-timeout>? P
- (441) <security-switch>? P
- (442) <logging>?
- (443) <user-log>?
- (444) <level>? P
- (445) <use-parent-handlers>? P
- (446) <filter-class>?
- (447) <handler>?
- (448) <console-handler>
 - (449) <name>
 - (450) <level>? P
 - (451) <encoding>?
 - (452) <filter-class>?
- (453) <file-handler>
 - (454) <name>
 - (455) <level>? P

(456) <encoding>?

(457) <filter-class>?

(458) <file-name>?

(459) <valid-day>

(460) <valid-hour>

(461) <buffer-size>? P

(462) <append>? P

(463) <smtp-handler>

(464) <name>

(465) <level>? P

(466) <encoding>?

(467) <filter-class>?

(468) <smtp-host-address>

(469) <from-address>

(470) <to-address>

(471) <cc-address>?

(472) <bcc-address>?

(473) <send-for-all-messages>? P

(474) <socket-handler>

(475) <name>

(476) <level>? P

(477) <encoding>?

(478) <filter-class>?

(479) <address>

(480) <port>

(481) <user-handler>

(482) <handler-class>

(483) <name>

(484) <level>? P

(485) <encoding>?

(486) <filter-class>?

(487) <handler-property>?

(488) <property>*

(489) <key>

(490) <value>

(491) <formatter-class>? P

(492) <formatter-property>?

(493) <property>*

(494) <key>

(495) <value>

(496) <access-log>?

(497) <enable>? P

(498) <enable-analysis>? P

(499) <level>? P

(500) <use-parent-handlers>? P

(501) <filter-class>?

(502) <handler>?

(503) <console-handler>

(504) <name>

(505) <level>? P

(506) <encoding>?

(507) <filter-class>?

(508) <file-handler>

(509) <name>

(510) <level>? P

(511) <encoding>?

(512) <filter-class>?

(513) <file-name>?

(514) <valid-day>

(515) <valid-hour>

(516) <buffer-size>? P

(517) <append>? P

(518) <smtp-handler>

(519) <name>

(520) <level>? P

(521) <encoding>?

- (522) <filter-class>?
- (523) <smtp-host-address>
- (524) <from-address>
- (525) <to-address>
- (526) <cc-address>?
- (527) <bcc-address>?
- (528) <send-for-all-messages>? P
- (529) <socket-handler>
 - (530) <name>
 - (531) <level>? P
 - (532) <encoding>?
 - (533) <filter-class>?
 - (534) <address>
 - (535) <port>
- (536) <user-handler>
 - (537) <handler-class>
 - (538) <name>
 - (539) <level>? P
 - (540) <encoding>?
 - (541) <filter-class>?
 - (542) <handler-property>?
 - (543) <property>*

(544) <key>

(545) <value>

(546) <formatter-class>? P

(547) <formatter-property>?

(548) <property>*

(549) <key>

(550) <value>

(551) <format>?

(552) <properties>?

(553) <property>*

(554) <key>

(555) <value>

D.3 Element Reference

(1) <web-container>

Description

The top level element inside WEBMain.xml, which is used for configuring all aspects of the JEUS Web module.

Child Elements

(2) monitoring?
 (7) redirect-stdout?
 (8) redirect-stderr?
 (9) shutdown-timeout?
 (10) context-group+
 (401) db-connection-pool*
 (430) session-cluster?
 (441) security-switch?

(442)logging?
(552)properties?

(2) <web-container> **<monitoring>**

Description This is the element for configuring environment of monitoring thread of Web Container. Monitoring thread is thread of inspecting on satus of several resources or pools for certain interval. When checking each of them, connection in connection pool is removed or other action is done. Child element of this element sets essentially the time for status inspection of other pools.

Child Elements (3)check-thread-pool?
(4)check-class-reload?
(5)check-session?
(6)check-db-connection-pool?

(3) <web-container> <monitoring> **<check-thread-pool>**

Description This element sets the time interval used for checking the status of worker threads in the worker thread pools. Worker threads are responsible for handling client requests to the container. The worker threads exist in worker thread pools and there exist one such pool for each Web server connection. The check interval set here will be applied to all worker thread pools in the container.<p>When this time period expires, a check will be performed on each thread pool. If some condition is met (such as an max idle time has expired for a particular thread) an action will be taken (such as removing the thread).

Value Description A time interval with the unit of milliseconds (the time period will be rounded to the nearest second). The shortest time period possible is 1000 milliseconds (1 second).

Value Type long

Value Type Description Integer

<i>Default Value</i>	300000
<i>Defined Value</i>	-1 Never use this value except for benchmark test purposes. A greater time period will increase the performance slightly.

(4) <web-container> <monitoring> **<check-class-reload>**

<i>Description</i>	This element sets the time interval used for checking the status of class reloading as defined in the <auto-reload><enable-reload> element. The monitor thread for check-class-reload does not check contexts for which the check-on-demand setting is true (this setting is found in the jeus-web-dd file).
<i>Value Description</i>	A time interval with the unit of milliseconds (the time period will be rounded to the nearest second). The shortest time period possible is 1000 milliseconds (1 second).
<i>Value Type</i>	long
<i>Value Type Description</i>	Integer
<i>Default Value</i>	300000
<i>Defined Value</i>	-1 finalized production environment for inactivating class reloading.

(5) <web-container> <monitoring> **<check-session>**

<i>Description</i>	This element sets how often session time out conditions should be checked. Session time out conditions are defined either in <context-group><session-timeout>element or in web.xml for each context. [value Description]: A time interval with the unit of milliseconds (the time period will be rounded to the nearest second). The shortest time period possible is 1000 milliseconds (1 second).
--------------------	---

<i>Value Type</i>	long
<i>Value Type Description</i>	Integer
<i>Default Value</i>	300000
<i>Defined Value</i>	-1 Disables all checking of session time out meaning that all sessions will be preserved until an explicit <code>HTTPSession.invalidate()</code> is issued.

(6) <web-container> <monitoring> **<check-db-connection-pool>**

<i>Description</i>	This element sets the time interval used when checking the status of connections in the DB connection pool. When this time period expires, all DB connections are checked for blocking and time out (active and idle) conditions. If any condition is true, an appropriate action will be taken such as returning the DB connection to the pool or removing it from the pool.
<i>Value Description</i>	A time interval with the unit of milliseconds (the time period will be rounded to the nearest second). The shortest time period possible is 1000 milliseconds (1 second).
<i>Value Type</i>	long
<i>Value Type Description</i>	Integer
<i>Default Value</i>	300000
<i>Defined Value</i>	-1 Inactivates the inspection on DB connection.

(7) <web-container> **<redirect-stdout>**

<i>Description</i>	It determines whether to retransmit as JEUS_HOME\logs\JeusSystem\[nodename]_[container name]\stdout_[date].log file which is displacement of Web Container
--------------------	---

Value Type boolean

Default Value false

(8) <web-container> **<redirect-stderr>**

Description It determines whether to retransmit as
JEUS_HOME\logs\JeusSystem\[nodename]_[container
name]\stdout_[date].log file

Value Type boolean

Default Value false

(9) <web-container> **<shutdown-timeout>**

Description Value of one second for worker thread to wait while executing
prior to terminate Web Container. This option waits forcibly the
completion of operation on worker threads before downing the
Container. This will allow to down container after ignoring the
related cycle if worker thread to execute the operation does not
exist.

Value Type int

Default Value -1

(10) <web-container> **<context-group>**

Description This element configures a JEUS specific construct called a
"context group". Context groups may be viewed as virtual servers
hosting any number of contexts (Web applications). The settings
inside this element will be used by every context it includes.

Child Elements (11)group-name
(12)group-docbase?
(13)servlet-home?
(14)virtual-host*
(17)webserver-connection
(245)session-config?

(249)print-error-to-browser?
 (250)encoding?
 (260)jsp-engine?
 (268)logging?
 (378)session-cluster?
 (389)response-header?

(11) <web-container> <context-group> **<group-name>**

Description This is an internally used name for the group.

Value Description context group name

Value Type token

(12) <web-container> <context-group> **<group-docbase>**

Description Defines the document base directory for the context group. This directory will contain all the context directories belonging to this context group.

Value Description Absolute path or relative path attached in defined SERBLET-HOME directory (ex) "webapps" (For example, "c:\jeus50\webhome\servlet_home\webapps").

Value Type token

(13) <web-container> <context-group> **<servlet-home>**

Description Resetting SERVLET_HOME which system variable. This configuration defines the default directory to be located in context group document base.

Value Type token

(14) <web-container> <context-group> **<virtual-host>**

Child Elements (15)virtual-host-name
 (16)host-list+

(15) <web-container> <context-group> <virtual-host> **<virtual-**

host-name>

Description Internal distinct name of Virtual Host.

Value Description Temporary unique name.

Value Type token

(16) <web-container> <context-group> <virtual-host> <host-list>

Description Each host-list element contains a DNS name (e.g. "www.foo.com") or an IP address. The virtual host will be known under all of the DNS names specified within these elements. The DNS names or IP addresses must be unique within the same context group (two virtual hosts within the same context group cannot share the same DNS name(s)).

Value Description A DNS name or IP address.

Value Type token

(17) <web-container> <context-group> <webserver-connection>

Description Each <webserver-connection> element configures a set of Web server front end connections. This means that normally there will be a Web server sitting in front of the Web container and receiving client HTTP requests. The Web server analyzes the request and determines whether it requires processing by the Web container (i.e. a request for a Servlet). Any Web container requests are then forwarded from the Web server to the appropriate Web container. The component in the Web container that is responsible for receiving these requests is called a "listener". This element is used to configure and connect such a listener to a Web server. Note that several Web server listeners may be configured on the same machine and the same Web container as long as each <port> assignment is unique and unused. Using several Web server listeners is accomplished simply by using several <webserver-connection><xxxxx-listener> sub-elements.

<i>Child Elements</i>	(18)webtob-listener
	(72)apache-listener
	(96)http-listener
	(121)tcp-listener
	(145)udp-listener
	(170)secure-listener
	(195)ajp13-listener
	(219)tmax-listener

```
(18) <web-container> <context-group> <webserver-connection>
<webtob-listener>
```

<i>Description</i>	This element is web server listener used for setting WebtoB web serser. The setting of relative listener is differentiated from other listener types.
--------------------	---

<i>Child Elements</i>	(19)listener-id
	(20)port
	(21)output-buffer-size?
	(22)thread-pool
	(36)postdata-read-timeout?
	(37)scheme?
	(38)hth-count?
	(39)request-prefetch?
	(40)disable-pipe?
	(41)webtob-address?
	(42)registration-id?
	(43)webtob-home?
	(44)read-timeout?
	(45)reconnect-timeout?
	(46)webtob-backup*

```
(19) <web-container> <context-group> <webserver-connection>
<webtob-listener> <listener-id>
```

<i>Description</i>	A unique name that identifies this listener. This name is used to refer to the listener from other parts of the configuration file.
--------------------	---

Value Description An arbitrary but unique name.

Value Type token

```
(20) <web-container> <context-group> <webserver-connection>
<webtob-listener> <port>
```

Description The port used to receive requests from the Web server. This port must match some port setting in the Web server's configuration files.

Value Description A port number greater than 0 and configuration file of WebtoB is identical with the value of JSVPORT

Value Type nonNegativeLongType

Value Type Description This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.

```
(21) <web-container> <context-group> <webserver-connection>
<webtob-listener> <output-buffer-size>
```

Description Setting the size of temporarily displaying buffer for storing JSP/Servlet result from out.write() method. temporarily displaying the size of buffers out.write() methods. Web Browser will flush if this buffer is full.

Value Description Buffer size in Byte unit

Value Type nonNegativeLongType

Value Type Description This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.

Default Value 8192

Example <output-buffer-size>16384</output-buffer-size>

```
(22) <web-container> <context-group> <webserver-connection>
<webtob-listener> <thread-pool>
```

Child Elements

- (23) min
- (24) max?
- (25) step?
- (26) max-idle-time?
- (27) max-wait-queue?
- (28) max-queue?
- (29) thread-state-notify?

```
( 23 ) <web-container> <context-group> <webserver-connection>
<webtob-listener> <thread-pool> <min>
```

Description Initial number or maximum number for maintaining worker thread in the Pool.

Value Description The number of threads.

Value Type nonNegativeLongType

Value Type Description This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.

```
( 24 ) <web-container> <context-group> <webserver-connection>
<webtob-listener> <thread-pool> <max>
```

Description Maximum number for maintaining worker thread in Pool.

Value Description The number of threads.

Value Type nonNegativeLongType

Value Type Description This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.

```
( 25 ) <web-container> <context-group> <webserver-connection>
<webtob-listener> <thread-pool> <step>
```

Description This represents the number of worker threads when adding in the pool as pool size is increased.

Value Description The number of threads to be added.

Value Type nonNegativeLongType

Value Type Description This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.

Default Value 1

Example <step>4</step>

```
(26) <web-container> <context-group> <webserver-connection>
<webtob-listener> <thread-pool> <max-idle-time>
```

Description The maximum amount of time that a worker thread may stay in the idle waiting state before being removed from the pool.

Value Description A number of milliseconds.

Value Type int

Value Type Description Long

Default Value 300000

```
(27) <web-container> <context-group> <webserver-connection>
<webtob-listener> <thread-pool> <max-wait-queue>
```

Description This element sets the maximum number of requests to be placed in request wait queue prior to generate new worker thread for handling the requests.

Value Description Maximum number of request to be placed in Queue

Value Type int

Value Type Description Integer

Default Value 4

```
(28) <web-container> <context-group> <webserver-connection>
<webtob-listener> <thread-pool> <max-queue>
```

Description Page informing "busy" status will be returned to the client when

more additional requests are arrived. Setting maximum request value to be waited in queue. This element is not used in Tmax/WebtoB listener. The queue size of WebtoB listener configures MaxQCount value in WebtoB configuration file. After this queue is full, busy page will be returned to client when additional requests are arrived. This element is not used in Tmax/WebtoB listener. Queue size of WebtoB listener is composed with MaxQCount value in WebtoB configuration file.

Value Type int

Default Value -1

```
(29) <web-container> <context-group> <webserver-connection>
<webtob-listener> <thread-pool> <thread-state-notify>
```

Description This element defines worker thread which is blocked and determine whether to send notification on the number of blocked threads.

Child Elements (30)max-thread-active-time?
 (31)notify-threshold?
 (32)restart-threshold?
 (33)notifier-id?
 (34)notify-subject?
 (35)restart-subject?

```
(30) <web-container> <context-group> <webserver-connection>
<webtob-listener> <thread-pool> <thread-state-notify> <max-
thread-active-time>
```

Description Sets the time for defining when a thread should be regarded as blocked. When a thread has executed for this amount of time, it is regarded as blocked.

Value Description A number of milliseconds. Set this value to roughly 1.5 times the execution time of the thread that takes the longest to execute.

Value Type int

Value Type Description Integer

```
(31) <web-container> <context-group> <webserver-connection>
<webtob-listener> <thread-pool> <thread-state-notify> <notify-
threshold>
```

Description Sets the maximum number of blocked threads that may exist at any given time before a warning notification is sent out using an e-mail notifier.

Value Description A number of Worker thread.

Value Type int

Value Type Description Integer

```
(32) <web-container> <context-group> <webserver-connection>
<webtob-listener> <thread-pool> <thread-state-notify> <restart-
threshold>
```

Description Sets the maximum number of blocked worker threads that may exist at any given time before a restart of the Web container is performed.

Value Description A number of Servlet threads.

Value Type int

Value Type Description Integer

```
(33) <web-container> <context-group> <webserver-connection>
<webtob-listener> <thread-pool> <thread-state-notify> <notifier-
id>
```

Description The ID of the email notifier to use.

Value Type token

```
(34) <web-container> <context-group> <webserver-connection>
<webtob-listener> <thread-pool> <thread-state-notify> <notify-
subject>
```

Description Sets the subject line of the e-mail that is sent out when the notify-treshold is exceeded.

Value Description A message allowing the receiver (the administrator) to quickly determine what this e-mail is about.

Value Type token

```
(35) <web-container> <context-group> <webserver-connection>
<webtob-listener> <thread-pool> <thread-state-notify> <restart-
subject>
```

Description Sets the subject line of the e-mail that is sent out when the notify-treshold is exceeded.

Value Description A message allowing the receiver (the administrator) to quickly determine what this e-mail is about.

Value Type token

```
(36) <web-container> <context-group> <webserver-connection>
<webtob-listener> <postdata-read-timeout>
```

Description Setting Maximum waiting time to read post-data from web server/web client. Execute it through request.getInputStream().read() method.

Value Description Time Interval in Millisecond

Value Type nonNegativeLongType

Value Type Description This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.

Default Value 30000

Example <postdata-read-timeout>40000</postdata-read-timeout>

```
(37) <web-container> <context-group> <webserver-connection>
<webtob-listener> <scheme>
```

<i>Description</i>	The protocol d returned value on Javax.servlet.http.HttpServletRequest.getScheme() method can be set for this field
<i>Value Description</i>	http or https
<i>Value Type</i>	schemeType
<i>Default Value</i>	http
<i>Defined Value</i>	http The value will be used if SSL is not used. https The related value will be sued when using SSL from Web Server with secure listener.

Example `<scheme>https</scheme>`

```
(38) <web-container> <context-group> <webserver-connection>
<webtob-listener> <hth-count>
```

<i>Description</i>	Specifies the number of HTH processes in the WebtoB server.
<i>Value Description</i>	Set equal to the number of HTH processes in the WebtoB server.
<i>Value Type</i>	nonNegativeLongType
<i>Value Type Description</i>	This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.
<i>Default Value</i>	1

```
(39) <web-container> <context-group> <webserver-connection>
<webtob-listener> <request-prefetch>
```

<i>Description</i>	The function of prefetching webtob request. If this function is set as true, performance on process will be enhanced through storing it in request queue In case where web container is downing abnormally,webtob request to be prefetched can be lost
--------------------	---

Value Description true or false

Value Type boolean

Value Type Description Boolean

Default Value true

```
(40) <web-container> <context-group> <webserver-connection>
<webtob-listener> <disable-pipe>
```

Description When WebtoB server is executed in the same machine with Listener, more efficient network communication than pipe communication can be selected. At this case, related element is set as "false" If listener and WebtoB server operate in different machine, it will inactivate pipe network communication and activates socket communication.

Value Description Activating or inactivating this configuration as Boolean value.

Value Type boolean

Value Type Description Boolean

Default Value false

Defined Value true
Enables Socket Communication

false
Enables Pipe Communicaiton

Example <disable-pipe>true</disable-pipe>

```
(41) <web-container> <context-group> <webserver-connection>
<webtob-listener> <webtob-address>
```

Description The IP address of the WebtoB server that should be connected to this listener. This has to be set since it is up to the listener to

actively locate and connect to the WebtoB server. This is different from the other types of Web server connections where the Web container is passive and waits for a connection from the Web server.

Value Description IP address of WebtoB Server.

Value Type token

Default Value 127.0.0.1

```
(42) <web-container> <context-group> <webserver-connection>
<webtob-listener> <registration-id>
```

Description Registration ID name to be used when connecting with WebtoB server.

Value Description Registration ID name should be identical with value below *SERVER statement in WebtoB configuration file.

Value Type token

Default Value default

Example <registration-id>MyGroup</registration-id>

```
(43) <web-container> <context-group> <webserver-connection>
<webtob-listener> <webtob-home>
```

Description Configuring Home directory of WebtoB server enabling efficient pipe communication between Web Container and Web Server

Value Description Absolute Directory Path in which Path classifier is placed at the end of String. (“\” : Windows, “/”: Unix)

Value Type token

Example <webtob-home>c:\WebtoB\</webtob-home>

```
(44) <web-container> <context-group> <webserver-connection>
<webtob-listener> <read-timeout>
```

Description WebtoB Web Server sends "ping" continuously to Web Container for set interval on "svrchktime" variable of WebtoB configuration file. Web Container sets time interval in this element for checking on "ping" in which WebtoB has sent. If WebtoB ping is not inspected in the set time interval, it will reset by considering network link as broken.

Value Description millisecond value. The value is set more than "svrchktime" in WebtoB configuration file.

Value Type nonNegativeLongType

Value Type Description This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.

Default Value 0

Defined Value 0
Infinity

Example <read-timeout>120000</read-timeout>

```
(45) <web-container> <context-group> <webserver-connection>
<webtob-listener> <reconnect-timeout>
```

Description Some connections between the WebtoB server and the webtob-listener may be disconnected during the operation. Reconnect-timeout (this element) is applied as a timeout value of the reconnect attempt that follows in such cases. If the timeout is expired with all of the current WebtoB connections disconnected and if some WebtoB backups are defined, then the Web container tries to fail-over to the next WebtoB backup. If that WebtoB backup is also failed, it tries the next and so on. If the last WebtoB backup is failed, it tries the (main) webtob-listener again.

Value Description The timeout value unit is milliseconds

Value Type int

<i>Value Type Description</i>	Long
<i>Default Value</i>	-1
<i>Defined Value</i>	-1 Infinite timeout period (the listener will attempt to reconnect forever). 0 No reconnection will ever be attempted.

```
(46) <web-container> <context-group> <webserver-connection>
<webtob-listener> <webtob-backup>
```

<i>Child Elements</i>	(47)port (48)output-buffer-size? (49)thread-pool? (63)postdata-read-timeout? (64)scheme? (65)hth-count? (66)disable-pipe? (67)webtob-address? (68)registration-id? (69)webtob-home? (70)read-timeout? (71)reconnect-timeout?
-----------------------	---

```
(47) <web-container> <context-group> <webserver-connection>
<webtob-listener> <webtob-backup> <port>
```

<i>Description</i>	Port to be connected to WebtoB Server. This port number is identical with "JSVPORT" value in WebtoB server configuration file.
<i>Value Description</i>	Port number greater than 0 and is identical with JSVPORT value of WebtoB configuration file.
<i>Value Type</i>	nonNegativeLongType

Value Type Description This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.

```
(48) <web-container> <context-group> <webserver-connection>
<webtob-listener> <webtob-backup> <output-buffer-size>
```

Description Setting the size of temporarily displaying buffer for storing JSP/Servlet result from out.write() method. temporarily displaying the size of buffers out.write() methods. Web Browser will flush if this buffer is full.

Value Description Buffer size in Byte unit

Value Type nonNegativeLongType

Value Type Description This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.

Default Value 8192

Example <output-buffer-size>16384</output-buffer-size>

```
(49) <web-container> <context-group> <webserver-connection>
<webtob-listener> <webtob-backup> <thread-pool>
```

Description Thread pool is used for maintaining worker threads which handle requests entering into Web Container. Through set port, the operation will be processed according to request by receiving new worker thread from related pool. In case of WebtoB Listener, configuration on thread pool is related to HTH process in WebtoB server. Therefore, the maximum number will be set as 10 if the number of HTH processes are 3 and actual number of thread pool to be generated are 30. Each worker thread gets the continuous connection with WebtoB server.

Child Elements (50)min
 (51)max?
 (52)step?
 (53)max-idle-time?

(54)max-wait-queue?

(55)max-queue?

(56)thread-state-notify?

```
(50) <web-container> <context-group> <webserver-connection>
<webtob-listener> <webtob-backup> <thread-pool> <min>
```

Description Initial number or maximum number for maintaining worker thread in the Pool.

Value Description The number of threads.

Value Type nonNegativeLongType

Value Type Description This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.

```
(51) <web-container> <context-group> <webserver-connection>
<webtob-listener> <webtob-backup> <thread-pool> <max>
```

Description Maximum number for maintaining worker thread in Pool.

Value Description The number of threads.

Value Type nonNegativeLongType

Value Type Description This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.

```
(52) <web-container> <context-group> <webserver-connection>
<webtob-listener> <webtob-backup> <thread-pool> <step>
```

Description This represents the number of worker threads when adding in the pool as pool size is increased.

Value Description The number of threads to be added.

Value Type nonNegativeLongType

Value Type Description This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.

Default Value 1

Example <step>4</step>

```
(53) <web-container> <context-group> <webserver-connection>
<webtob-listener> <webtob-backup> <thread-pool> <max-idle-time>
```

Description The maximum amount of time that a worker thread may stay in the idle waiting state before being removed from the pool.

Value Description A number of milliseconds.

Value Type int

Value Type Description Long

Default Value 300000

```
(54) <web-container> <context-group> <webserver-connection>
<webtob-listener> <webtob-backup> <thread-pool> <max-wait-queue>
```

Description This element sets the maximum number of requests to be placed in request wait queue prior to generate new worker thread for handling the requests.

Value Description Maximum number of request to be placed in Queue

Value Type int

Value Type Description Integer

Default Value 4

```
(55) <web-container> <context-group> <webserver-connection>
<webtob-listener> <webtob-backup> <thread-pool> <max-queue>
```

Description Page informing "busy" status will be returned to the client when more additional requests are arrived. Setting maximum request value to be waited in queue. This element is not used in Tmax/WebtoB listener. The queue size of WebtoB listener configures MaxQCount value in WebtoB configuration file. After

this queue is full, busy page will be returned to client when additional requests are arrived. This element is not used in Tmax/WebtoB listener. Queue size of WebtoB listener is composed with MaxQCount value in WebtoB configuration file.

Value Type int

Default Value -1

```
(56) <web-container> <context-group> <webserver-connection>
<webtob-listener> <webtob-backup> <thread-pool> <thread-state-notify>
```

Description This element defines worker thread which is blocked and determine whether to send notification on the number of blocked threads.

Child Elements

- (57)max-thread-active-time?
- (58)notify-threshold?
- (59)restart-threshold?
- (60)notifier-id?
- (61)notify-subject?
- (62)restart-subject?

```
(57) <web-container> <context-group> <webserver-connection>
<webtob-listener> <webtob-backup> <thread-pool> <thread-state-notify> <max-thread-active-time>
```

Description Sets the time for defining when a thread should be regarded as blocked. When a thread has executed for this amount of time, it is regarded as blocked.

Value Description A number of milliseconds. Set this value to roughly 1.5 times the execution time of the thread that takes the longest to execute.

Value Type int

Value Type Description Integer

```
(58) <web-container> <context-group> <webserver-connection>
```

```
<webtob-listener> <webtob-backup> <thread-pool> <thread-state-notify> <notify-threshold>
```

Description Sets the maximum number of blocked threads that may exist at any given time before a warning notification is sent out using an e-mail notifier.

Value Description A number of Worker thread.

Value Type int

Value Type Description Integer

```
(59) <web-container> <context-group> <webserver-connection>
<webtob-listener> <webtob-backup> <thread-pool> <thread-state-notify> <restart-threshold>
```

Description Sets the maximum number of blocked worker threads that may exist at any given time before a restart of the Web container is performed.

Value Description A number of Servlet threads.

Value Type int

Value Type Description Integer

```
(60) <web-container> <context-group> <webserver-connection>
<webtob-listener> <webtob-backup> <thread-pool> <thread-state-notify> <notifier-id>
```

Description The ID of the email notifier to use.

Value Type token

```
(61) <web-container> <context-group> <webserver-connection>
<webtob-listener> <webtob-backup> <thread-pool> <thread-state-notify> <notify-subject>
```

Description Sets the subject line of the e-mail that is sent out when the notify-threshold is exceeded.

Value Description A message allowing the receiver (the administrator) to quickly determine what this e-mail is about.

Value Type token

```
(62) <web-container> <context-group> <webserver-connection>
<webtob-listener> <webtob-backup> <thread-pool> <thread-state-
notify> <restart-subject>
```

Description Sets the subject line of the e-mail that is sent out when the notify-threshold is exceeded.

Value Description A message allowing the receiver (the administrator) to quickly determine what this e-mail is about.

Value Type token

```
(63) <web-container> <context-group> <webserver-connection>
<webtob-listener> <webtob-backup> <postdata-read-timeout>
```

Description Setting Maximum waiting time to read post-data from web server/web client. Execute it through request.getInputStream().read() method.

Value Description Time Interval in Millisecond

Value Type nonNegativeLongType

Value Type Description This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.

Default Value 30000

Example <postdata-read-timeout>40000</postdata-read-timeout>

```
(64) <web-container> <context-group> <webserver-connection>
<webtob-listener> <webtob-backup> <scheme>
```

Description The protocol d returned value on Javax.servlet.http.HttpServletRequest.getScheme() method can be

	set for this field
<i>Value Description</i>	http or https
<i>Value Type</i>	schemeType
<i>Default Value</i>	http
<i>Defined Value</i>	http The value will be used if SSL is not used.
	https The related value will be used when using SSL from Web Server with secure listener.
<i>Example</i>	<scheme>https</scheme>

```
(65) <web-container> <context-group> <webserver-connection>
<webtob-listener> <webtob-backup> <hth-count>
```

<i>Description</i>	Specifies the number of HTH processes in the WebtoB server.
<i>Value Description</i>	Set equal to the number of HTH processes in the WebtoB server.
<i>Value Type</i>	nonNegativeLongType
<i>Value Type Description</i>	This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.
<i>Default Value</i>	1

```
(66) <web-container> <context-group> <webserver-connection>
<webtob-listener> <webtob-backup> <disable-pipe>
```

<i>Description</i>	When WebtoB server is executed in the same machine with Listener, more efficient network communication than pipe communication can be selected. At this case, related element is set as "false" If listener and WebtoB server operate in different machine, it will inactivate pipe network communication and activates socket communication.
--------------------	---

Value Description Activating or inactivating this configuration as Boolean value.

Value Type boolean

Value Type Description Boolean

Default Value false

Defined Value true
Enables Socket Communication

false

Enables Pipe Communicaiton

Example <disable-pipe>true</disable-pipe>

```
(67) <web-container> <context-group> <webserver-connection>
<webtob-listener> <webtob-backup> <webtob-address>
```

Description The IP address of the WebtoB server that should be connected to this listener. This has to be set since it is up to the listener to actively locate and connect to the WebtoB server. This is different from the other types of Web server connections where the Web container is passive and waits for a connection from the Web server.

Value Description An IP address of a WebtoB server.

Value Type token

Default Value 127.0.0.1

```
(68) <web-container> <context-group> <webserver-connection>
<webtob-listener> <webtob-backup> <registration-id>
```

Description Registration ID name used for connecting with WebtoB server.

Value Description Registration ID name should be identical with value located in *SERVER statement in WebtoB configuration file

<i>Value Type</i>	token
<i>Default Value</i>	default
<i>Example</i>	<registration-id>MyGroup</registration-id>

```
(69) <web-container> <context-group> <webserver-connection>
<webtob-listener> <webtob-backup> <webtob-home>
```

<i>Description</i>	Configuring Home directory of WebtoB server enabling efficient pipe communication between Web Container and Web Server
<i>Value Description</i>	Absolute Directory Path in which Path classifier is placed at the end of String("\ " Windows, "/" Unix).
<i>Value Type</i>	token
<i>Example</i>	<webtob-home>c:\WebtoB\</webtob-home>

```
(70) <web-container> <context-group> <webserver-connection>
<webtob-listener> <webtob-backup> <read-timeout>
```

<i>Description</i>	WebtoB web server is continuously sending "ping" to Web Container for the interval "svrchktime" variable of WebtoB configuration file is set. Time Interval is set for checking "ping" in which WebtoB sends. If WebtoB ping is not inspected in set time interval, it will reset as being considered as broken. Therefore, the value of this elemnt should be set greater than "svrchktime" value.
<i>Value Description</i>	millisecond value. For WebtoB configurtion file, sets the value greater than "svrchktime" value.
<i>Value Type</i>	nonNegativeLongType
<i>Value Type Description</i>	This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.
<i>Default Value</i>	0
<i>Defined Value</i>	0

Infinity

Example`<read-timeout>120000</read-timeout>`

```
(71) <web-container> <context-group> <webserver-connection>
<webtob-listener> <webtob-backup> <reconnect-timeout>
```

Description

Some connections between the WebtoB server and the webtob-listener may be disconnected during the operation. Reconnect-timeout (this element) is applied as a timeout value of the reconnect attempt that follows in such cases. If the timeout is expired with all of the current WebtoB connections disconnected and if some WebtoB backups are defined, then the Web container tries to fail-over to the next WebtoB backup. If that WebtoB backup is also failed, it tries the next and so on. If the last WebtoB backup is failed, it tries the (main) webtob-listener again.

Value Description

Timeout value in Millisecond unit.

Value Type

int

Value Type Description

Long

Default Value

-1

Defined Value

-1

Infinite number of Timeout Period(Listener will attempt to establish on reconnection.)

0

Will not attempt to reconnect.

```
(72) <web-container> <context-group> <webserver-connection>
<apache-listener>
```

Description

This element is server listener construction for Apache Web server.

<i>Child Elements</i>	(73)listener-id
	(74)port
	(75)output-buffer-size?
	(76)thread-pool
	(90)postdata-read-timeout?
	(91)scheme?
	(92)back-log?
	(93)busy-notify-page?
	(94)server-access-control?
	(95)allowed-server*

```
(73) <web-container> <context-group> <webserver-connection>
<apache-listener> <listener-id>
```

Description A unique name that identifies this listener. This name is used to refer to the listener from other parts of the configuration file.

Value Description An arbitrary but unique name..

Value Type token

```
(74) <web-container> <context-group> <webserver-connection>
<apache-listener> <port>
```

Description Port is used to receive HTTP requests from the client. The port should be identical with the value of ApJServDefaultPort of jeus-apache.conf file

Value Description Port value greater than 0

Value Type nonNegativeLongType

Value Type Description This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.

```
(75) <web-container> <context-group> <webserver-connection>
<apache-listener> <output-buffer-size>
```

Description Setting the size of temporarily displaying buffer for storing JSP/Servlet result from out.write() method. temporarily displaying

the size of buffers out.write() methods. Web Browser will flush if this buffer is full.

Value Description Buffer size in Byte unit

Value Type nonNegativeLongType

Value Type Description This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.

Default Value 8192

Example `<output-buffer-size>16384</output-buffer-size>`

```
(76) <web-container> <context-group> <webserver-connection>
<apache-listener> <thread-pool>
```

Description The thread pool is used to maintain a set of worker threads that will handle incoming requests to the Web container. Each time a new request is received on the port specified, a new worker thread is drawn from this pool and assigned to process the request.

Child Elements

- (77)min
- (78)max?
- (79)step?
- (80)max-idle-time?
- (81)max-wait-queue?
- (82)max-queue?
- (83)thread-state-notify?

```
(77) <web-container> <context-group> <webserver-connection>
<apache-listener> <thread-pool> <min>
```

Description Initial number or maximum number for maintaining worker thread in the Pool.

Value Description The number of threads.

Value Type nonNegativeLongType

Value Type Description This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.

```
(78) <web-container> <context-group> <webserver-connection>
<apache-listener> <thread-pool> <max>
```

Description Maximum number for maintaining worker thread in Pool.

Value Description The number of threads.

Value Type nonNegativeLongType

Value Type Description This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.

```
(79) <web-container> <context-group> <webserver-connection>
<apache-listener> <thread-pool> <step>
```

Description This represents the number of worker threads when adding in the pool as pool size is increased.

Value Description The number of threads to be added.

Value Type nonNegativeLongType

Value Type Description This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.

Default Value 1

Example <step>4</step>

```
(80) <web-container> <context-group> <webserver-connection>
<apache-listener> <thread-pool> <max-idle-time>
```

Description The maximum amount of time that a worker thread may stay in the idle waiting state before being removed from the pool.

Value Description A number of milliseconds.

Value Type int

Value Type Description Long

Default Value 300000

```
(81) <web-container> <context-group> <webserver-connection>
<apache-listener> <thread-pool> <max-wait-queue>
```

Description This element sets the maximum number of requests to be placed in request wait queue prior to generate new worker thread for handling the requests.

Value Description Maximum number of request to be placed in Queue

Value Type int

Value Type Description Integer

Default Value 4

```
(82) <web-container> <context-group> <webserver-connection>
<apache-listener> <thread-pool> <max-queue>
```

Description Page informing "busy" status will be returned to the client when more additional requests are arrived. Setting maximum request value to be waited in queue. This element is not used in Tmax/WebtoB listener. The queue size of WebtoB listener configures MaxQCount value in WebtoB configuration file. After this queue is full, busy page will be returned to client when additional requests are arrived. This element is not used in Tmax/WebtoB listener. Queue size of WebtoB listener is composed with MaxQCount value in WebtoB configuration file.

Value Type int

Default Value -1

```
(83) <web-container> <context-group> <webserver-connection>
<apache-listener> <thread-pool> <thread-state-notify>
```

Description This element defines worker thread which is blocked and

determine whether to send notification on the number of blocked threads.

Child Elements

(84)max-thread-active-time?
 (85)notify-threshold?
 (86)restart-threshold?
 (87)notifier-id?
 (88)notify-subject?
 (89)restart-subject?

```
(84) <web-container> <context-group> <webserver-connection>
<apache-listener> <thread-pool> <thread-state-notify> <max-
thread-active-time>
```

Description

Sets the time for defining when a thread should be regarded as blocked. When a thread has executed for this amount of time, it is regarded as blocked.

Value Description

A number of milliseconds. Set this value to roughly 1.5 times the execution time of the thread that takes the longest to execute.

Value Type

int

Value Type Description Integer

```
(85) <web-container> <context-group> <webserver-connection>
<apache-listener> <thread-pool> <thread-state-notify> <notify-
threshold>
```

Description

Sets the maximum number of blocked threads that may exist at any given time before a warning notification is sent out using an e-mail notifier.

Value Description

A number of Worker thread.

Value Type

int

Value Type Description Integer

```
(86) <web-container> <context-group> <webserver-connection>
```

```
<apache-listener> <thread-pool> <thread-state-notify> <restart-  
threshold>
```

Description Sets the maximum number of blocked worker threads that may exist at any given time before a restart of the Web container is performed.

Value Description A number of Servlet threads.

Value Type int

Value Type Description Integer

```
(87) <web-container> <context-group> <webserver-connection>  
<apache-listener> <thread-pool> <thread-state-notify> <notifier-  
id>
```

Description The ID of the email notifier to use.

Value Type token

```
(88) <web-container> <context-group> <webserver-connection>  
<apache-listener> <thread-pool> <thread-state-notify> <notify-  
subject>
```

Description Sets the subject line of the e-mail that is sent out when the notify-treshold is exceeded.

Value Description A message allowing the receiver (the administrator) to quickly determine what this e-mail is about.

Value Type token

```
(89) <web-container> <context-group> <webserver-connection>  
<apache-listener> <thread-pool> <thread-state-notify> <restart-  
subject>
```

Description Sets the subject line of the e-mail that is sent out when the notify-treshold is exceeded.

Value Description A message allowing the receiver (the administrator) to quickly

determine what this e-mail is about.

Value Type token

```
(90) <web-container> <context-group> <webserver-connection>
<apache-listener> <postdata-read-timeout>
```

Description Setting Maximum waiting time to read post-data from web server/web client. Execute it through request.getInputStream().read() method.

Value Description Time Interval in Millisecond

Value Type nonNegativeLongType

Value Type Description This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.

Default Value 30000

Example <postdata-read-timeout>40000</postdata-read-timeout>

```
(91) <web-container> <context-group> <webserver-connection>
<apache-listener> <scheme>
```

Description Setting the returned value classifier on Javax.servlet.http.HttpServletRequest.getScheme() method protocol. Setting value of "https" when using SSL function or using secure listener in WebtoB or Apache

Value Description http or https

Value Type schemeType

Default Value http

Defined Value http
 Certain value will be used if SSL is used.

https

Related value will be used when using SSL in Web Server with secure listener.

Example `<scheme>https</scheme>`

```
(92) <web-container> <context-group> <webserver-connection>
<apache-listener> <back-log>
```

Description Sets the socket's backlog value (the maximum size of the socket request wait queue as specified in the "java.net.ServerSocket(int port, int backlog)" constructor).

Value Description Value greater than 0

Value Type nonNegativeLongType

Value Type Description This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.

Default Value 50

```
(93) <web-container> <context-group> <webserver-connection>
<apache-listener> <busy-notify-page>
```

Description When queue is full, sends the page set in <busy-notify-page> element

Value Description value greater than 0

Value Type token

Value Type Description Setting static content file included in absolute path.

Example `<busy-notify-
page>c:\web\errorpages\busypage.html</busy-
notify-page>`

```
(94) <web-container> <context-group> <webserver-connection>
<apache-listener> <server-access-control>
```

Description Activates or inactivates access control on server. If this option is

activated and the server has permission in <allowed-server> element, this listener can be connected.

Value Description Time interval in Millisecond. Activating or inactivating this configuration as Boolean value.

Value Type boolean

Value Type Description Boolean

Default Value false

Defined Value true
Activates Server Access Control.

false
Inactivates Server Access Control.

Example <server-access-control>true</server-access-control>

```
(95) <web-container> <context-group> <webserver-connection>
<apache-listener> <b>allowed-server</b>
```

Description Each <allowed-server> element sets a Server IP address to connect to a certain listener. These elements are used only when the value of <server-access-control> element is true. It will reject entire requests from server which is not included in this element.

Value Description IP address allowing this listener to be connected.

Value Type token

Example <allowed-server>127.0.0.1</allowed-server>

```
(96) <web-container> <context-group> <webserver-connection>
<b>http-listener</b>
```

Description The HTTP listener represents a default internal Web server of the JEUS Web container. This internal Web server is rather basic and

should only be used in a development environment or a small size production environment. For a large production environment, a "real" Web server such as WebtoB or Apache should be used. When this listener element is configured, the Web container will thus be capable of receiving and processing simple HTTP requests on its own.

Child Elements

(97) listener-id
 (98) port
 (99) output-buffer-size?
 (100) thread-pool
 (114) postdata-read-timeout?
 (115) scheme?
 (116) back-log?
 (117) server-access-control?
 (118) allowed-server*
 (119) http-version?
 (120) read-timeout?

```
(97) <web-container> <context-group> <webserver-connection>
<http-listener> <listener-id>
```

Description

A unique name that identifies this listener. This name is used to refer to the listener from other parts of the configuration file.

Value Description

An arbitrary but unique name.

Value Type

token

```
(98) <web-container> <context-group> <webserver-connection>
<http-listener> <port>
```

Description

The port used to receive HTTP requests from the client. According to IANA (www.iana.org), the port assigned to HTTP is 80.

Value Description

A port number greater than 0.

Value Type

nonNegativeLongType

Value Type Description This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.

```
(99) <web-container> <context-group> <webserver-connection>
<http-listener> <output-buffer-size>
```

Description Setting the size of temporarily displaying buffer for storing JSP/Servlet result from out.write() method. temporarily displaying the size of buffers out.write() methods. Web Browser will flush if this buffer is full.

Value Description Buffer size in Byte unit

Value Type nonNegativeLongType

Value Type Description This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.

Default Value 8192

Example <output-buffer-size>16384</output-buffer-size>

```
(100) <web-container> <context-group> <webserver-connection>
<http-listener> <thread-pool>
```

Description The thread pool is used to maintain a set of worker threads that will handle incoming requests to the Web container. Each time a new request is received on the port specified, a new worker thread is drawn from this pool and assigned to process the request.

Child Elements (101)min
 (102)max?
 (103)step?
 (104)max-idle-time?
 (105)max-wait-queue?
 (106)max-queue?
 (107)thread-state-notify?

```
(101) <web-container> <context-group> <webserver-connection>
```

```
<http-listener> <thread-pool> <min>
```

Description Initial number or maximum number for maintaining worker thread in the Pool.

Value Description The number of threads.

Value Type nonNegativeLongType

Value Type Description This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.

```
(102) <web-container> <context-group> <webserver-connection>  
<http-listener> <thread-pool> <max>
```

Description Maximum number for maintaining worker thread in Pool.

Value Description The number of threads.

Value Type nonNegativeLongType

Value Type Description This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.

```
(103) <web-container> <context-group> <webserver-connection>  
<http-listener> <thread-pool> <step>
```

Description This represents the number of worker threads when adding in the pool as pool size is increased.

Value Description The number of threads to be added.

Value Type nonNegativeLongType

Value Type Description This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.

Default Value 1

Example <step>4</step>

```
(104) <web-container> <context-group> <webserver-connection>
```

```
<http-listener> <thread-pool> <max-idle-time>
```

<i>Description</i>	The maximum amount of time that a worker thread may stay in the idle waiting state before being removed from the pool.
<i>Value Description</i>	A number of milliseconds.
<i>Value Type</i>	int
<i>Value Type Description</i>	Long
<i>Default Value</i>	300000

```
(105) <web-container> <context-group> <webserver-connection>  
<http-listener> <thread-pool> <max-wait-queue>
```

<i>Description</i>	This element sets the maximum number of requests to be placed in request wait queue prior to generate new worker thread for handling the requests.
<i>Value Description</i>	Maximum number of request to be placed in Queue
<i>Value Type</i>	int
<i>Value Type Description</i>	Integer
<i>Default Value</i>	4

```
(106) <web-container> <context-group> <webserver-connection>  
<http-listener> <thread-pool> <max-queue>
```

<i>Description</i>	Page informing "busy" status will be returned to the client when more additional requests are arrived. Setting maximum request value to be waited in queue. This element is not used in Tmax/WebtoB listener. The queue size of WebtoB listener configures MaxQCount value in WebtoB configuration file. After this queue is full, busy page will be returned to client when additional requests are arrived. This element is not used in Tmax/WebtoB listener. Queue size of WebtoB listener is composed with MaxQCount value in WebtoB configuration file.
--------------------	--

Value Type int

Default Value -1

```
(107) <web-container> <context-group> <webserver-connection>
<http-listener> <thread-pool> <thread-state-notify>
```

Description This element defines worker thread which is blocked and determine whether to send notification on the number of blocked threads.

Child Elements

- (108)max-thread-active-time?
- (109)notify-threshold?
- (110)restart-threshold?
- (111)notifier-id?
- (112)notify-subject?
- (113)restart-subject?

```
(108) <web-container> <context-group> <webserver-connection>
<http-listener> <thread-pool> <thread-state-notify> <max-thread-
active-time>
```

Description Sets the time for defining when a thread should be regarded as blocked. When a thread has executed for this amount of time, it is regarded as blocked.

Value Description A number of milliseconds. Set this value to roughly 1.5 times the execution time of the thread that takes the longest to execute.

Value Type int

Value Type Description Integer

```
(109) <web-container> <context-group> <webserver-connection>
<http-listener> <thread-pool> <thread-state-notify> <notify-
threshold>
```

Description Sets the maximum number of blocked threads that may exist at any given time before a warning notification is sent out using an e-mail notifier.

Value Description A number of Worker thread.

Value Type int

Value Type Description Integer

```
(110) <web-container> <context-group> <webserver-connection>
<http-listener> <thread-pool> <thread-state-notify> <restart-
threshold>
```

Description Sets the maximum number of blocked worker threads that may exist at any given time before a restart of the Web container is performed.

Value Description A number of Servlet threads.

Value Type int

Value Type Description Integer

```
(111) <web-container> <context-group> <webserver-connection>
<http-listener> <thread-pool> <thread-state-notify> <notifier-
id>
```

Description The ID of the email notifier to use.

Value Type token

```
(112) <web-container> <context-group> <webserver-connection>
<http-listener> <thread-pool> <thread-state-notify> <notify-
subject>
```

Description Sets the subject line of the e-mail that is sent out when the notify-treshold is exceeded.

Value Description A message allowing the receiver (the administrator) to quickly determine what this e-mail is about.

Value Type token

```
(113) <web-container> <context-group> <webserver-connection>
```



```
<http-listener> <thread-pool> <thread-state-notify> <restart-  
subject>
```

Description Sets the subject line of the e-mail that is sent out when the notify-threshold is exceeded.

Value Description A message allowing the receiver (the administrator) to quickly determine what this e-mail is about.

Value Type token

```
(114) <web-container> <context-group> <webserver-connection>  
<http-listener> <postdata-read-timeout>
```

Description Setting Maximum waiting time to read post-data from web server/web client. Execute it through request.getInputStream().read() method.

Value Description Time Interval in Millisecond

Value Type nonNegativeLongType

Value Type Description This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.

Default Value 30000

Example <postdata-read-timeout>40000</postdata-read-timeout>

```
(115) <web-container> <context-group> <webserver-connection>  
<http-listener> <scheme>
```

Description The protocol d returned value on Javax.servlet.http.HttpServletRequest.getScheme() method can be set for this field

Value Description http or https

Value Type schemeType

<i>Default Value</i>	http
<i>Defined Value</i>	http The value will be used if SSL is not used.
	https The related value will be used when using SSL from Web Server with secure listener.

Example <scheme>https</scheme>

```
(116) <web-container> <context-group> <webserver-connection>
<http-listener> <b>back-log</b>
```

<i>Description</i>	Sets the socket's backlog value (the maximum size of the socket request wait queue as specified in the "java.net.ServerSocket(int port, int backlog)" constructor).
<i>Value Description</i>	value greater than 0.
<i>Value Type</i>	nonNegativeLongType
<i>Value Type Description</i>	This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.
<i>Default Value</i>	50

```
(117) <web-container> <context-group> <webserver-connection>
<http-listener> <b>server-access-control</b>
```

<i>Description</i>	Activates or inactivates access control on server. If this option is activated and the server has permission in <allowed-server> element, this listener can be connected.
<i>Value Description</i>	Activating or inactivating this configuration as Boolean value.
<i>Value Type</i>	boolean
<i>Value Type Description</i>	Boolean

<i>Default Value</i>	false
<i>Defined Value</i>	true Activates Server Access Control.
	false Inactivates Server Access Control.
<i>Example</i>	<code><server-access-control>true</server-access-control></code>

```
(118) <web-container> <context-group> <webserver-connection>
<http-listener> <allowed-server>
```

<i>Description</i>	<allowed-server> element sets IP address of one server to be connected to the certain listener. These elements are used only when the value of <server-access-control> element is true. It rejects all the requests from server excluded in this element.
<i>Value Description</i>	IP address of server allowing to connect this listener
<i>Value Type</i>	token
<i>Example</i>	<code><allowed-server>127.0.0.1</allowed-server></code>

```
(119) <web-container> <context-group> <webserver-connection>
<http-listener> <http-version>
```

<i>Description</i>	<http-version> element means HTTP version in related listener.
<i>Value Description</i>	HTTP version used in this listener.
<i>Value Type</i>	token
<i>Example</i>	<code><http-version>1.1</http-version></code>

```
(120) <web-container> <context-group> <webserver-connection>
<http-listener> <read-timeout>
```

<i>Description</i>	Read time out for maintaining Keep Alive.
--------------------	---

<i>Value Type</i>	nonNegativeLongType
<i>Value Type Description</i>	This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.
<i>Default Value</i>	5000

```
(121) <web-container> <context-group> <webserver-connection>
<tcp-listener>
```

Description The TCP listener works like the HTTP listener but supports custom network protocols other than HTTP.

Child Elements

```
(122)listener-id
(123)port
(124)output-buffer-size?
(125)thread-pool
(139)postdata-read-timeout?
(140)scheme?
(141)back-log?
(142)server-access-control?
(143)allowed-server*
(144)dispatcher-config-class?
```

```
(122) <web-container> <context-group> <webserver-connection>
<tcp-listener> <listener-id>
```

Description A unique name that identifies this listener. This name is used to refer to the listener from other parts of the configuration file.

Value Description An arbitrary but unique name.

Value Type token

```
(123) <web-container> <context-group> <webserver-connection>
<tcp-listener> <port>
```

Description The port used to receive requests from the Web browser.

Value Description A port number greater than 0.

Value Type nonNegativeLongType

Value Type Description This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.

```
(124) <web-container> <context-group> <webserver-connection>
<tcp-listener> <output-buffer-size>
```

Description Setting the size of temporarily displaying buffer for storing JSP/Servlet result from out.write() method. temporarily displaying the size of buffers out.write() methods. Web Browser will flush if this buffer is full.

Value Description Buffer size in Byte unit

Value Type nonNegativeLongType

Value Type Description This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.

Default Value 8192

Example <output-buffer-size>16384</output-buffer-size>

```
(125) <web-container> <context-group> <webserver-connection>
<tcp-listener> <thread-pool>
```

Description The thread pool is used to maintain a set of worker threads that will handle incoming requests to the Web container. Each time a new request is received on the port specified, a new worker thread is drawn from this pool and assigned to process the request.

Child Elements

- (126)min
- (127)max?
- (128)step?
- (129)max-idle-time?
- (130)max-wait-queue?
- (131)max-queue?
- (132)thread-state-notify?

```
(126) <web-container> <context-group> <webserver-connection>
<tcp-listener> <thread-pool> <min>
```

Description Initial number or maximum number for maintaining worker thread in the Pool.

Value Description The number of threads.

Value Type nonNegativeLongType

Value Type Description This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.

```
(127) <web-container> <context-group> <webserver-connection>
<tcp-listener> <thread-pool> <max>
```

Description Maximum number for maintaining worker thread in Pool.

Value Description The number of threads.

Value Type nonNegativeLongType

Value Type Description This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.

```
(128) <web-container> <context-group> <webserver-connection>
<tcp-listener> <thread-pool> <step>
```

Description This represents the number of worker threads when adding in the pool as pool size is increased.

Value Description The number of threads to be added.

Value Type nonNegativeLongType

Value Type Description This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.

Default Value 1

Example <step>4</step>

```
(129) <web-container> <context-group> <webserver-connection>
<tcp-listener> <thread-pool> <max-idle-time>
```

Description The maximum amount of time that a worker thread may stay in the idle waiting state before being removed from the pool.

Value Description A number of milliseconds.

Value Type int

Value Type Description Long

Default Value 300000

```
(130) <web-container> <context-group> <webserver-connection>
<tcp-listener> <thread-pool> <max-wait-queue>
```

Description This element sets the maximum number of requests to be placed in request wait queue prior to generate new worker thread for handling the requests.

Value Description Maximum number of request to be placed in Queue

Value Type int

Value Type Description Integer

Default Value 4

```
(131) <web-container> <context-group> <webserver-connection>
<tcp-listener> <thread-pool> <max-queue>
```

Description Page informing "busy" status will be returned to the client when more additional requests are arrived. Setting maximum request value to be waited in queue. This element is not used in Tmax/WebtoB listener. The queue size of WebtoB listener configures MaxQCount value in WebtoB configuration file. After this queue is full, busy page will be returned to client when additional requests are arrived. This element is not used in Tmax/WebtoB listener. Queue size of WebtoB listener is

composed with MaxQCount value in WebtoB configuration file.

Value Type int

Default Value -1

```
(132) <web-container> <context-group> <webserver-connection>
<tcp-listener> <thread-pool> <thread-state-notify>
```

Description This element defines worker thread which is blocked and determine whether to send notification on the number of blocked threads.

Child Elements (133)max-thread-active-time?
 (134)notify-threshold?
 (135)restart-threshold?
 (136)notifier-id?
 (137)notify-subject?
 (138)restart-subject?

```
(133) <web-container> <context-group> <webserver-connection>
<tcp-listener> <thread-pool> <thread-state-notify> <max-thread-
active-time>
```

Description Sets the time for defining when a thread should be regarded as blocked. When a thread has executed for this amount of time, it is regarded as blocked.

Value Description A number of milliseconds. Set this value to roughly 1.5 times the execution time of the thread that takes the longest to execute.

Value Type int

Value Type Description Integer

```
(134) <web-container> <context-group> <webserver-connection>
<tcp-listener> <thread-pool> <thread-state-notify> <notify-
threshold>
```

Description Sets the maximum number of blocked threads that may exist at any given time before a warning notification is sent out using an e-

mail notifier.

Value Description A number of Worker thread.

Value Type int

Value Type Description Integer

```
(135) <web-container> <context-group> <webserver-connection>
<tcp-listener> <thread-pool> <thread-state-notify> <restart-
threshold>
```

Description Sets the maximum number of blocked worker threads that may exist at any given time before a restart of the Web container is performed.

Value Description A number of Servlet threads.

Value Type int

Value Type Description Integer

```
(136) <web-container> <context-group> <webserver-connection>
<tcp-listener> <thread-pool> <thread-state-notify> <notifier-id>
```

Description The ID of the email notifier to use.

Value Type token

```
(137) <web-container> <context-group> <webserver-connection>
<tcp-listener> <thread-pool> <thread-state-notify> <notify-
subject>
```

Description Sets the subject line of the e-mail that is sent out when the notify-treshold is exceeded.

Value Description A message allowing the receiver (the administrator) to quickly determine what this e-mail is about.

Value Type token

```
(138) <web-container> <context-group> <webserver-connection>
```

```
<tcp-listener> <thread-pool> <thread-state-notify> <restart-  
subject>
```

Description Sets the subject line of the e-mail that is sent out when the notify-treshold is exceeded.

Value Description A message allowing the receiver (the administrator) to quickly determine what this e-mail is about.

Value Type token

```
(139) <web-container> <context-group> <webserver-connection>  
<tcp-listener> <postdata-read-timeout>
```

Description Setting Maximum waiting time to read post-data from web server/web client. Execute it through request.getInputStream().read() method.

Value Description Time Interval in Millisecond

Value Type nonNegativeLongType

Value Type Description This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.

Default Value 30000

Example <postdata-read-timeout>40000</postdata-read-timeout>

```
(140) <web-container> <context-group> <webserver-connection>  
<tcp-listener> <scheme>
```

Description The protocol d returned value on Javax.servlet.http.HttpServletRequest.getScheme() method can be set for this field

Value Description http or https

Value Type schemeType

<i>Default Value</i>	http
<i>Defined Value</i>	http The value will be used if SSL is not used.
	https The related value will be used when using SSL from Web Server with secure listener.

Example `<scheme>https</scheme>`

```
(141) <web-container> <context-group> <webserver-connection>
<tcp-listener> <back-log>
```

<i>Description</i>	Sets the socket's backlog value (the maximum size of the socket request wait queue as specified in the "java.net.ServerSocket(int port, int backlog)" constructor).
<i>Value Description</i>	A number greater than 0.
<i>Value Type</i>	nonNegativeLongType
<i>Value Type Description</i>	This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.
<i>Default Value</i>	50

```
(142) <web-container> <context-group> <webserver-connection>
<tcp-listener> <server-access-control>
```

<i>Description</i>	Activates or inactivates access control on server. If this option is activated and the server has permission in <allowed-server> element, this listener can be connected.
<i>Value Description</i>	Activating or inactivating this configuration as Boolean value.
<i>Value Type</i>	boolean
<i>Value Type Description</i>	Boolean

<i>Default Value</i>	false
<i>Defined Value</i>	true Activates Server Access Control.
	false Inactivates Server Access Control.
<i>Example</i>	<code><server-access-control>true</server-access-control></code>

```
(143) <web-container> <context-group> <webserver-connection>
<tcp-listener> <allowed-server>
```

<i>Description</i>	<allowed-server> element sets IP address of a server to be connected to the certain listener. These elements are used only when the value of <server-access-control> element is true. It rejects all the requests from server excluded in this element.
<i>Value Description</i>	IP address of server allowing to connect this listener
<i>Value Type</i>	token
<i>Example</i>	<code><allowed-server>127.0.0.1</allowed-server></code>

```
(144) <web-container> <context-group> <webserver-connection>
<tcp-listener> <dispatcher-config-class>
```

<i>Description</i>	Sets the class setting dispatcher. At this time, class should locate in the classpath which has been set and the name of class is full qualified class name.
<i>Value Type</i>	token

```
(145) <web-container> <context-group> <webserver-connection>
<udp-listener>
```

<i>Description</i>	Use UDP listener working just like the TCP listener.
<i>Child Elements</i>	(146)listener-id

```
(147)port
(148)input-buffer-size?
(149)output-buffer-size?
(150)thread-pool
(164)postdata-read-timeout?
(165)scheme?
(166)back-log?
(167)server-access-control?
(168)allowed-server*
(169)dispatcher-config-class?
```

```
(146) <web-container> <context-group> <webserver-connection>
<udp-listener> <listener-id>
```

Description Unique name distinguishing related listener. This name is used for mentioning on listner from other parts of the configuration file

Value Description Temporary unique name.

Value Type token

```
(147) <web-container> <context-group> <webserver-connection>
<udp-listener> <port>
```

Description Port used for receiving request from client.

Value Description Port value greater than 0.

Value Type nonNegativeLongType

Value Type Description This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.

```
(148) <web-container> <context-group> <webserver-connection>
<udp-listener> <input-buffer-size>
```

Description Maximum value of DatagramPacket when reading input.

Value Description Port value greater than 0.

<i>Value Type</i>	nonNegativeLongType
<i>Value Type Description</i>	This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.
<i>Default Value</i>	8192

```
(149) <web-container> <context-group> <webserver-connection>
<udp-listener> <output-buffer-size>
```

<i>Description</i>	Setting the size of temporarily displaying buffer for storing JSP/Servlet result from out.write() method. temporarily displaying the size of buffers out.write() methods. Web Browser will flush if this buffer is full.
--------------------	--

<i>Value Description</i>	Buffer size in Byte unit
--------------------------	--------------------------

<i>Value Type</i>	nonNegativeLongType
-------------------	---------------------

<i>Value Type Description</i>	This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.
-------------------------------	---

<i>Default Value</i>	8192
----------------------	------

<i>Example</i>	<output-buffer-size>16384</output-buffer-size>
----------------	--

```
(150) <web-container> <context-group> <webserver-connection>
<udp-listener> <thread-pool>
```

<i>Description</i>	The thread pool is used to maintain a set of worker threads that will handle incoming requests to the Web container. Each time a new request is received on the port specified, a new worker thread is drawn from this pool and assigned to process the request.
--------------------	--

<i>Child Elements</i>	(151)min (152)max? (153)step? (154)max-idle-time? (155)max-wait-queue?
-----------------------	--

(156)max-queue?

(157)thread-state-notify?

```
(151) <web-container> <context-group> <webserver-connection>
<udp-listener> <thread-pool> <min>
```

Description Initial number or maximum number for maintaining worker thread in the Pool.

Value Description The number of threads.

Value Type nonNegativeLongType

Value Type Description This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.

```
(152) <web-container> <context-group> <webserver-connection>
<udp-listener> <thread-pool> <max>
```

Description Maximum number for maintaining worker thread in Pool.

Value Description The number of threads.

Value Type nonNegativeLongType

Value Type Description This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.

```
(153) <web-container> <context-group> <webserver-connection>
<udp-listener> <thread-pool> <step>
```

Description This represents the number of worker threads when adding in the pool as pool size is increased.

Value Description The number of threads to be added.

Value Type nonNegativeLongType

Value Type Description This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.

Default Value 1

Example <step>4</step>

```
(154) <web-container> <context-group> <webserver-connection>
<udp-listener> <thread-pool> <max-idle-time>
```

Description The maximum amount of time that a worker thread may stay in the idle waiting state before being removed from the pool.

Value Description A number of milliseconds.

Value Type int

Value Type Description Long

Default Value 300000

```
(155) <web-container> <context-group> <webserver-connection>
<udp-listener> <thread-pool> <max-wait-queue>
```

Description This element sets the maximum number of requests to be placed in request wait queue prior to generate new worker thread for handling the requests.

Value Description Maximum number of request to be placed in Queue

Value Type int

Value Type Description Integer

Default Value 4

```
(156) <web-container> <context-group> <webserver-connection>
<udp-listener> <thread-pool> <max-queue>
```

Description Page informing "busy" status will be returned to the client when more additional requests are arrived. Setting maximum request value to be waited in queue. This element is not used in Tmax/WebtoB listener. The queue size of WebtoB listener configures MaxQCount value in WebtoB configuration file. After this queue is full, busy page will be returned to client when additional requests are arrived. This element is not used in

Tmax/WebtoB listener.Queue size of WebtoB listener is composed with MaxQCount value in WebtoB configuration file.

Value Type int

Default Value -1

```
(157) <web-container> <context-group> <webserver-connection>
<udp-listener> <thread-pool> <thread-state-notify>
```

Description This element defines worker thread which is blocked and determine whether to send notification on the number of blocked threads.

Child Elements

- (158)max-thread-active-time?
- (159)notify-threshold?
- (160)restart-threshold?
- (161)notifier-id?
- (162)notify-subject?
- (163)restart-subject?

```
(158) <web-container> <context-group> <webserver-connection>
<udp-listener> <thread-pool> <thread-state-notify> <max-thread-
active-time>
```

Description Sets the time for defining when a thread should be regarded as blocked. When a thread has executed for this amount of time, it is regarded as blocked.

Value Description A number of milliseconds. Set this value to roughly 1.5 times the execution time of the thread that takes the longest to execute.

Value Type int

Value Type Description Integer

```
(159) <web-container> <context-group> <webserver-connection>
<udp-listener> <thread-pool> <thread-state-notify> <notify-
threshold>
```

Description Sets the maximum number of blocked threads that may exist at any given time before a warning notification is sent out using an e-mail notifier.

Value Description A number of Worker thread.

Value Type int

Value Type Description Integer

```
(160) <web-container> <context-group> <webserver-connection>
<udp-listener> <thread-pool> <thread-state-notify> <restart-
threshold>
```

Description Sets the maximum number of blocked worker threads that may exist at any given time before a restart of the Web container is performed.

Value Description A number of Servlet threads.

Value Type int

Value Type Description Integer

```
(161) <web-container> <context-group> <webserver-connection>
<udp-listener> <thread-pool> <thread-state-notify> <notifier-id>
```

Description The ID of the email notifier to use.

Value Type token

```
(162) <web-container> <context-group> <webserver-connection>
<udp-listener> <thread-pool> <thread-state-notify> <notify-
subject>
```

Description Sets the subject line of the e-mail that is sent out when the notify-treshold is exceeded.

Value Description A message allowing the receiver (the administrator) to quickly determine what this e-mail is about.

Value Type token

```
(163) <web-container> <context-group> <webserver-connection>
<udp-listener> <thread-pool> <thread-state-notify> <restart-
subject>
```

Description Sets the subject line of the e-mail that is sent out when the notify-treshold is exceeded.

Value Description A message allowing the receiver (the administrator) to quickly determine what this e-mail is about.

Value Type token

```
(164) <web-container> <context-group> <webserver-connection>
<udp-listener> <postdata-read-timeout>
```

Description Setting Maximum waiting time to read post-data from web server/web client. Execute it through request.getInputStream().read() method.

Value Description Time Interval in Millisecond

Value Type nonNegativeLongType

Value Type Description This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.

Default Value 30000

Example <postdata-read-timeout>40000</postdata-read-timeout>

```
(165) <web-container> <context-group> <webserver-connection>
<udp-listener> <scheme>
```

Description The protocol d returned value on Javax.servlet.http.HttpServletRequest.getScheme() method can be set for this field

Value Description http or https

<i>Value Type</i>	schemeType
<i>Default Value</i>	http
<i>Defined Value</i>	http The value will be used if SSL is not used.
	https The related value will be used when using SSL from Web Server with secure listener.

Example <scheme>https</scheme>

```
(166) <web-container> <context-group> <webserver-connection>
<udp-listener> <b>back-log</b>
```

Description Sets the socket's backlog value (the maximum size of the socket request wait queue as specified in the "java.net.ServerSocket(int port, int backlog)" constructor).

Value Description Value greater than 0

Value Type nonNegativeLongType

Value Type Description This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.

Default Value 50

```
(167) <web-container> <context-group> <webserver-connection>
<udp-listener> <b>server-access-control</b>
```

Description Activates or inactivates access control on server. If this option is activated and the server has permission in <allowed-server> element, this listener can be connected.

Value Description Activating or inactivating this configuration as Boolean value.

Value Type boolean

Value Type Description Boolean

Default Value false

Defined Value true
Activates Server Access Control.

false
Inactivates Server Access Control.

Example `<server-access-control>true</server-access-control>`

```
(168) <web-container> <context-group> <webserver-connection>
<udp-listener> <allowed-server>
```

Description `<allowed-server>` element sets IP address of one server to be connected to the certain listener. These elements are used only when the value of `<server-access-control>` element is true. It rejects all the requests from server excluded in this element.

Value Description IP address of server allowing to connect this listener

Value Type token

Example `<allowed-server>127.0.0.1</allowed-server>`

```
(169) <web-container> <context-group> <webserver-connection>
<udp-listener> <dispatcher-config-class>
```

Description Sets the class setting dispatcher. At this time, class should locate in classpath which has been set and class name should be qualified class name.

Value Type token

```
(170) <web-container> <context-group> <webserver-connection>
<secure-listener>
```

Description The secure listener works just like the HTTP listener but also

supports HTTPS/SSL requests.

Child Elements

(171)listener-id
 (172)port
 (173)output-buffer-size?
 (174)thread-pool
 (188)postdata-read-timeout?
 (189)scheme?
 (190)back-log?
 (191)server-access-control?
 (192)allowed-server*
 (193)http-version?
 (194)read-timeout?

```
(171) <web-container> <context-group> <webserver-connection>
<secure-listener> <listener-id>
```

Description A unique name that identifies this listener. This name is used to refer to the listener from other parts of the configuration file.

Value Description An arbitrary but unique name.

Value Type token

```
(172) <web-container> <context-group> <webserver-connection>
<secure-listener> <port>
```

Description The port used to receive HTTP requests from the client. According to IANA (www.iana.org), the port assigned to HTTP is 80.

Value Description A port number greater than 0.

Value Type nonNegativeLongType

Value Type Description This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.

```
(173) <web-container> <context-group> <webserver-connection>
<secure-listener> <output-buffer-size>
```

<i>Description</i>	Setting the size of temporarily displaying buffer for storing JSP/Servlet result from out.write() method. temporarily displaying the size of buffers out.write() methods. Web Browser will flush if this buffer is full.
<i>Value Description</i>	Buffer size in Byte unit
<i>Value Type</i>	nonNegativeLongType
<i>Value Type Description</i>	This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.
<i>Default Value</i>	8192
<i>Example</i>	<code><output-buffer-size>16384</output-buffer-size></code>

```
(174) <web-container> <context-group> <webserver-connection>
<secure-listener> <thread-pool>
```

<i>Description</i>	The thread pool is used to maintain a set of worker threads that will handle incoming requests to the Web container. Each time a new request is received on the port specified, a new worker thread is drawn from this pool and assigned to process the request.
<i>Child Elements</i>	(175)min (176)max? (177)step? (178)max-idle-time? (179)max-wait-queue? (180)max-queue? (181)thread-state-notify?

```
(175) <web-container> <context-group> <webserver-connection>
<secure-listener> <thread-pool> <min>
```

<i>Description</i>	Initial number or maximum number for maintaining worker thread in the Pool.
<i>Value Description</i>	The number of threads.

Value Type nonNegativeLongType

Value Type Description This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.

```
(176) <web-container> <context-group> <webserver-connection>
<secure-listener> <thread-pool> <max>
```

Description Maximum number for maintaining worker thread in Pool.

Value Description The number of threads.

Value Type nonNegativeLongType

Value Type Description This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.

```
(177) <web-container> <context-group> <webserver-connection>
<secure-listener> <thread-pool> <step>
```

Description This represents the number of worker threads when adding in the pool as pool size is increased.

Value Description The number of threads to be added.

Value Type nonNegativeLongType

Value Type Description This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.

Default Value 1

Example <step>4</step>

```
(178) <web-container> <context-group> <webserver-connection>
<secure-listener> <thread-pool> <max-idle-time>
```

Description The maximum amount of time that a worker thread may stay in the idle waiting state before being removed from the pool.

Value Description A number of milliseconds.

Value Type int

Value Type Description Long

Default Value 300000

```
(179) <web-container> <context-group> <webserver-connection>
<secure-listener> <thread-pool> <max-wait-queue>
```

Description This element sets the maximum number of requests to be placed in request wait queue prior to generate new worker thread for handling the requests.

Value Description Maximum number of request to be placed in Queue

Value Type int

Value Type Description Integer

Default Value 4

```
(180) <web-container> <context-group> <webserver-connection>
<secure-listener> <thread-pool> <max-queue>
```

Description Page informing "busy" status will be returned to the client when more additional requests are arrived. Setting maximum request value to be waited in queue. This element is not used in Tmax/WebtoB listener. The queue size of WebtoB listener configures MaxQCount value in WebtoB configuration file. After this queue is full, busy page will be returned to client when additional requests are arrived. This element is not used in Tmax/WebtoB listener. Queue size of WebtoB listener is composed with MaxQCount value in WebtoB configuration file.

Value Type int

Default Value -1

```
(181) <web-container> <context-group> <webserver-connection>
<secure-listener> <thread-pool> <thread-state-notify>
```

Description This element defines worker thread which is blocked and determine whether to send notification on the number of blocked threads.

Child Elements (182)max-thread-active-time?
 (183)notify-threshold?
 (184)restart-threshold?
 (185)notifier-id?
 (186)notify-subject?
 (187)restart-subject?

```
(182) <web-container> <context-group> <webserver-connection>
<secure-listener> <thread-pool> <thread-state-notify> <max-
thread-active-time>
```

Description Sets the time for defining when a thread should be regarded as blocked. When a thread has executed for this amount of time, it is regarded as blocked.

Value Description A number of milliseconds. Set this value to roughly 1.5 times the execution time of the thread that takes the longest to execute.

Value Type int

Value Type Description Integer

```
(183) <web-container> <context-group> <webserver-connection>
<secure-listener> <thread-pool> <thread-state-notify> <notify-
threshold>
```

Description Sets the maximum number of blocked threads that may exist at any given time before a warning notification is sent out using an e-mail notifier.

Value Description A number of Worker thread.

Value Type int

Value Type Description Integer

```
(184) <web-container> <context-group> <webserver-connection>
<secure-listener> <thread-pool> <thread-state-notify> <restart-
threshold>
```

Description Sets the maximum number of blocked worker threads that may exist at any given time before a restart of the Web container is performed.

Value Description A number of Servlet threads.

Value Type int

Value Type Description Integer

```
(185) <web-container> <context-group> <webserver-connection>
<secure-listener> <thread-pool> <thread-state-notify> <notifier-
id>
```

Description The ID of the email notifier to use.

Value Type token

```
(186) <web-container> <context-group> <webserver-connection>
<secure-listener> <thread-pool> <thread-state-notify> <notify-
subject>
```

Description Sets the subject line of the e-mail that is sent out when the notify-treshold is exceeded.

Value Description A message allowing the receiver (the administrator) to quickly determine what this e-mail is about.

Value Type token

```
(187) <web-container> <context-group> <webserver-connection>
<secure-listener> <thread-pool> <thread-state-notify> <restart-
subject>
```

Description Sets the subject line of the e-mail that is sent out when the notify-treshold is exceeded.

Value Description A message allowing the receiver (the administrator) to quickly determine what this e-mail is about.

Value Type token

```
(188) <web-container> <context-group> <webserver-connection>
<secure-listener> <postdata-read-timeout>
```

Description Setting Maximum waiting time to read post-data from web server/web client. Execute it through request.getInputStream().read() method.

Value Description Time Interval in Millisecond

Value Type nonNegativeLongType

Value Type Description This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.

Default Value 30000

Example <postdata-read-timeout>40000</postdata-read-timeout>

```
(189) <web-container> <context-group> <webserver-connection>
<secure-listener> <scheme>
```

Description The protocol d returned value on Javax.servlet.http.HttpServletRequest.getScheme() method can be set for this field

Value Description http or https

Value Type schemeType

Default Value http

Defined Value http
The value will be used if SSL is not used.

https

The related value will be used when using SSL from Web Server with secure listener.

Example

```
<scheme>https</scheme>
```

```
(190) <web-container> <context-group> <webserver-connection>
<secure-listener> <back-log>
```

Description

Sets the socket's backlog value (the maximum size of the socket request wait queue as specified in the "java.net.ServerSocket(int port, int backlog)" constructor).

Value Description

value greater than 0.

Value Type

nonNegativeLongType

Value Type Description

This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.

Default Value

50

```
(191) <web-container> <context-group> <webserver-connection>
<secure-listener> <server-access-control>
```

Description

Activates or inactivates access control on server. If this option is activated and the server has permission in <allowed-server> element, this listener can be connected.

Value Description

Activating or inactivating this configuration as Boolean value.

Value Type

boolean

Value Type Description

Boolean

Default Value

false

Defined Value

true

Activates Server Access Control.

false

Inactivates Server Access Control.

Example `<server-access-control>true</server-access-control>`

```
(192) <web-container> <context-group> <webserver-connection>
<secure-listener> <b>allowed-server</b>
```

Description `<allowed-server>` element sets IP address of one server to be connected to the certain listener. These elements are used only when the value of `<server-access-control>` element is true. It rejects all the requests from server excluded in this element.

Value Description IP address of server allowing to connect this listener

Value Type token

Example `<allowed-server>127.0.0.1</allowed-server>`

```
(193) <web-container> <context-group> <webserver-connection>
<secure-listener> <b>http-version</b>
```

Description `<http-version>` element means HTTP version in related listener.

Value Description HTTP version used in this listener.

Value Type token

Example `<http-version>1.1</http-version>`

```
(194) <web-container> <context-group> <webserver-connection>
<secure-listener> <b>read-timeout</b>
```

Description Read time out for maintaining Keep Alive.

Value Type nonNegativeLongType

Value Type Description This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.

Default Value 5000

```
(195) <web-container> <context-group> <webserver-connection>
<ajp13-listener>
```

Description The AJP1.3 listener works exactly like the <apache-listener> except that this listener supports the AJP13 protocol, which is used by the Apache mod_jk module (apache-listener supports the AJP12 protocol).

Child Elements

- (196) listener-id
- (197) port
- (198) output-buffer-size?
- (199) thread-pool
- (213) postdata-read-timeout?
- (214) scheme?
- (215) back-log?
- (216) busy-notify-page?
- (217) server-access-control?
- (218) allowed-server*

```
(196) <web-container> <context-group> <webserver-connection>
<ajp13-listener> <listener-id>
```

Description A unique name that identifies this listener. This name is used to refer to the listener from other parts of the configuration file.

Value Description An arbitrary but unique name..

Value Type token

```
(197) <web-container> <context-group> <webserver-connection>
<ajp13-listener> <port>
```

Description Port is used to receive HTTP requests from the client. The port should be identical with the value of ApJServDefaultPort of jeus-apache.conf file

Value Description Port value greater than 0

Value Type nonNegativeLongType

Value Type Description This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.

```
(198) <web-container> <context-group> <webserver-connection>
<ajp13-listener> <output-buffer-size>
```

Description Setting the size of temporarily displaying buffer for storing JSP/Servlet result from out.write() method. temporarily displaying the size of buffers out.write() methods. Web Browser will flush if this buffer is full.

Value Description Buffer size in Byte unit

Value Type nonNegativeLongType

Value Type Description This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.

Default Value 8192

Example <output-buffer-size>16384</output-buffer-size>

```
(199) <web-container> <context-group> <webserver-connection>
<ajp13-listener> <thread-pool>
```

Description The thread pool is used to maintain a set of worker threads that will handle incoming requests to the Web container. Each time a new request is received on the port specified, a new worker thread is drawn from this pool and assigned to process the request.

Child Elements (200)min
 (201)max?
 (202)step?
 (203)max-idle-time?
 (204)max-wait-queue?
 (205)max-queue?
 (206)thread-state-notify?

```
(200) <web-container> <context-group> <webserver-connection>
```



```
<ajp13-listener> <thread-pool> <min>
```

Description Initial number or maximum number for maintaining worker thread in the Pool.

Value Description The number of threads.

Value Type nonNegativeLongType

Value Type Description This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.

```
(201) <web-container> <context-group> <webserver-connection>  
<ajp13-listener> <thread-pool> <max>
```

Description Maximum number for maintaining worker thread in Pool.

Value Description The number of threads.

Value Type nonNegativeLongType

Value Type Description This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.

```
(202) <web-container> <context-group> <webserver-connection>  
<ajp13-listener> <thread-pool> <step>
```

Description This represents the number of worker threads when adding in the pool as pool size is increased.

Value Description The number of threads to be added.

Value Type nonNegativeLongType

Value Type Description This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.

Default Value 1

Example <step>4</step>

```
(203) <web-container> <context-group> <webserver-connection>
```

```
<ajp13-listener> <thread-pool> <max-idle-time>
```

<i>Description</i>	The maximum amount of time that a worker thread may stay in the idle waiting state before being removed from the pool.
<i>Value Description</i>	A number of milliseconds.
<i>Value Type</i>	int
<i>Value Type Description</i>	Long
<i>Default Value</i>	300000

```
(204) <web-container> <context-group> <webserver-connection>
<ajp13-listener> <thread-pool> <max-wait-queue>
```

<i>Description</i>	This element sets the maximum number of requests to be placed in request wait queue prior to generate new worker thread for handling the requests.
<i>Value Description</i>	Maximum number of request to be placed in Queue
<i>Value Type</i>	int
<i>Value Type Description</i>	Integer
<i>Default Value</i>	4

```
(205) <web-container> <context-group> <webserver-connection>
<ajp13-listener> <thread-pool> <max-queue>
```

<i>Description</i>	Page informing "busy" status will be returned to the client when more additional requests are arrived. Setting maximum request value to be waited in queue. This element is not used in Tmax/WebtoB listener. The queue size of WebtoB listener configures MaxQCount value in WebtoB configuration file. After this queue is full, busy page will be returned to client when additional requests are arrived. This element is not used in Tmax/WebtoB listener. Queue size of WebtoB listener is composed with MaxQCount value in WebtoB configuration file.
--------------------	--

Value Type int

Default Value -1

```
(206) <web-container> <context-group> <webserver-connection>
<ajp13-listener> <thread-pool> <thread-state-notify>
```

Description This element defines worker thread which is blocked and determine whether to send notification on the number of blocked threads.

Child Elements

- (207)max-thread-active-time?
- (208)notify-threshold?
- (209)restart-threshold?
- (210)notifier-id?
- (211)notify-subject?
- (212)restart-subject?

```
(207) <web-container> <context-group> <webserver-connection>
<ajp13-listener> <thread-pool> <thread-state-notify> <max-
thread-active-time>
```

Description Sets the time for defining when a thread should be regarded as blocked. When a thread has executed for this amount of time, it is regarded as blocked.

Value Description A number of milliseconds. Set this value to roughly 1.5 times the execution time of the thread that takes the longest to execute.

Value Type int

Value Type Description Integer

```
(208) <web-container> <context-group> <webserver-connection>
<ajp13-listener> <thread-pool> <thread-state-notify> <notify-
threshold>
```

Description Sets the maximum number of blocked threads that may exist at any given time before a warning notification is sent out using an e-mail notifier.

Value Description A number of Worker thread.

Value Type int

Value Type Description Integer

```
(209) <web-container> <context-group> <webserver-connection>
<ajp13-listener> <thread-pool> <thread-state-notify> <restart-
threshold>
```

Description Sets the maximum number of blocked worker threads that may exist at any given time before a restart of the Web container is performed.

Value Description A number of Servlet threads.

Value Type int

Value Type Description Integer

```
(210) <web-container> <context-group> <webserver-connection>
<ajp13-listener> <thread-pool> <thread-state-notify> <notifier-
id>
```

Description The ID of the email notifier to use.

Value Type token

```
(211) <web-container> <context-group> <webserver-connection>
<ajp13-listener> <thread-pool> <thread-state-notify> <notify-
subject>
```

Description Sets the subject line of the e-mail that is sent out when the notify-treshold is exceeded.

Value Description A message allowing the receiver (the administrator) to quickly determine what this e-mail is about.

Value Type token

```
(212) <web-container> <context-group> <webserver-connection>
```

```
<ajp13-listener> <thread-pool> <thread-state-notify> <restart-  
subject>
```

Description Sets the subject line of the e-mail that is sent out when the notify-threshold is exceeded.

Value Description A message allowing the receiver (the administrator) to quickly determine what this e-mail is about.

Value Type token

```
(213) <web-container> <context-group> <webserver-connection>  
<ajp13-listener> <postdata-read-timeout>
```

Description Setting Maximum waiting time to read post-data from web server/web client. Execute it through request.getInputStream().read() method.

Value Description Time Interval in Millisecond

Value Type nonNegativeLongType

Value Type Description This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.

Default Value 30000

Example <postdata-read-timeout>40000</postdata-read-timeout>

```
(214) <web-container> <context-group> <webserver-connection>  
<ajp13-listener> <scheme>
```

Description Setting the returned value classifier on Javax.servlet.http.HttpServletRequest.getScheme() method protocol. Setting value of "https" when using SSL function or using secure listener in WebtoB or Apache

Value Description http or https

Value Type schemeType

<i>Default Value</i>	http
<i>Defined Value</i>	http Certain value will be used if SSL is used.
	https Related value will be used when using SSL in Web Server with secure listener.
<i>Example</i>	<scheme>https</scheme>

```
(215) <web-container> <context-group> <webserver-connection>
<ajp13-listener> <b>back-log</b>
```

<i>Description</i>	Sets the socket's backlog value (the maximum size of the socket request wait queue as specified in the "java.net.ServerSocket(int port, int backlog)" constructor).
<i>Value Description</i>	Value greater than 0
<i>Value Type</i>	nonNegativeLongType
<i>Value Type Description</i>	This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.
<i>Default Value</i>	50

```
(216) <web-container> <context-group> <webserver-connection>
<ajp13-listener> <b>busy-notify-page</b>
```

<i>Description</i>	When queue is full, sends the page set in <busy-notify-page> element
<i>Value Description</i>	value greater than 0
<i>Value Type</i>	token
<i>Value Type Description</i>	Setting static content file included in absolute path.
<i>Example</i>	<busy-notify-

```
page>c:\web\errorpages\busypage.html</busy-
notify-page>
```

```
(217) <web-container> <context-group> <webserver-connection>
<ajp13-listener> <server-access-control>
```

<i>Description</i>	Activates or inactivates access control on server. If this option is activated and the server has permission in <allowed-server> element, this listener can be connected.
<i>Value Description</i>	Time interval in Millisecond. Activating or inactivating this configuration as Boolean value.
<i>Value Type</i>	boolean
<i>Value Type Description</i>	Boolean
<i>Default Value</i>	false
<i>Defined Value</i>	true Activates Server Access Control.
	false Inactivates Server Access Control.
<i>Example</i>	<server-access-control>true</server-access-control>

```
(218) <web-container> <context-group> <webserver-connection>
<ajp13-listener> <allowed-server>
```

<i>Description</i>	Each <allowed-server> element sets a Server IP address to connect to a certain listener. These elements are used only when the value of <server-access-control> element is true. It will reject entire requests from server which is not included in this element.
<i>Value Description</i>	IP address allowing this listener to be connected.
<i>Value Type</i>	token

Example <allowed-server>127.0.0.1</allowed-server>

```
(219) <web-container> <context-group> <webserver-connection>
<tmax-listener>
```

Description This element used for the configurations when connecting with Tmax. This element is identical with Webtob listener.

Child Elements (220)listener-id
 (221)port
 (222)output-buffer-size?
 (223)thread-pool
 (237)postdata-read-timeout?
 (238)reconnect-timeout?
 (239)tmax-address?
 (240)server-group-name
 (241)server-name
 (242)tmax-backup-address?
 (243)tmax-backup-port?
 (244)dispatcher-config-class?

```
(220) <web-container> <context-group> <webserver-connection>
<tmax-listener> <listener-id>
```

Description A unique name that identifies this listener. This name is used to refer to the listener from other parts of the configuration file.

Value Description An arbitrary but unique name

Value Type token

```
(221) <web-container> <context-group> <webserver-connection>
<tmax-listener> <port>
```

Description Port used for connecting tmax server.

Value Description Port number greater than 0.

Value Type nonNegativeLongType

Value Type Description This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.

```
(222) <web-container> <context-group> <webserver-connection>
<txmax-listener> <output-buffer-size>
```

Description Setting the size of buffer to be displayed from JSP/Servlet from out.write() method. It will flush web browser after filling this buffer.

Value Description Buffer size in Byte unit

Value Type nonNegativeLongType

Value Type Description This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.

Default Value 8192

Example <output-buffer-size>16384</output-buffer-size>

```
(223) <web-container> <context-group> <webserver-connection>
<txmax-listener> <thread-pool>
```

Child Elements

- (224)min
- (225)max?
- (226)step?
- (227)max-idle-time?
- (228)max-wait-queue?
- (229)max-queue?
- (230)thread-state-notify?

```
(224) <web-container> <context-group> <webserver-connection>
<txmax-listener> <thread-pool> <min>
```

Description Initial number or maximum number for maintaining worker thread in the Pool.

Value Description The number of threads.

Value Type nonNegativeLongType

Value Type Description This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.

```
(225) <web-container> <context-group> <webserver-connection>
<tmax-listener> <thread-pool> <max>
```

Description Maximum number for maintaining worker thread in Pool.

Value Description The number of threads.

Value Type nonNegativeLongType

Value Type Description This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.

```
(226) <web-container> <context-group> <webserver-connection>
<tmax-listener> <thread-pool> <step>
```

Description This represents the number of worker threads when adding in the pool as pool size is increased.

Value Description The number of threads to be added.

Value Type nonNegativeLongType

Value Type Description This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.

Default Value 1

Example <step>4</step>

```
(227) <web-container> <context-group> <webserver-connection>
<tmax-listener> <thread-pool> <max-idle-time>
```

Description The maximum amount of time that a worker thread may stay in the idle waiting state before being removed from the pool.

Value Description A number of milliseconds.

Value Type int

Value Type Description Long

Default Value 300000

```
(228) <web-container> <context-group> <webserver-connection>
<tmax-listener> <thread-pool> <max-wait-queue>
```

Description This element sets the maximum number of requests to be placed in request wait queue prior to generate new worker thread for handling the requests.

Value Description Maximum number of request to be placed in Queue

Value Type int

Value Type Description Integer

Default Value 4

```
(229) <web-container> <context-group> <webserver-connection>
<tmax-listener> <thread-pool> <max-queue>
```

Description Page informing "busy" status will be returned to the client when more additional requests are arrived. Setting maximum request value to be waited in queue. This element is not used in Tmax/WebtoB listener. The queue size of WebtoB listener configures MaxQCount value in WebtoB configuration file. After this queue is full, busy page will be returned to client when additional requests are arrived. This element is not used in Tmax/WebtoB listener. Queue size of WebtoB listener is composed with MaxQCount value in WebtoB configuration file.

Value Type int

Default Value -1

```
(230) <web-container> <context-group> <webserver-connection>
<tmax-listener> <thread-pool> <thread-state-notify>
```

Description This element defines worker thread which is blocked and determine whether to send notification on the number of blocked threads.

Child Elements (231)max-thread-active-time?
 (232)notify-threshold?
 (233)restart-threshold?
 (234)notifier-id?
 (235)notify-subject?
 (236)restart-subject?

```
(231) <web-container> <context-group> <webserver-connection>
<tmax-listener> <thread-pool> <thread-state-notify> <max-thread-
active-time>
```

Description Sets the time for defining when a thread should be regarded as blocked. When a thread has executed for this amount of time, it is regarded as blocked.

Value Description A number of milliseconds. Set this value to roughly 1.5 times the execution time of the thread that takes the longest to execute.

Value Type int

Value Type Description Integer

```
(232) <web-container> <context-group> <webserver-connection>
<tmax-listener> <thread-pool> <thread-state-notify> <notify-
threshold>
```

Description Sets the maximum number of blocked threads that may exist at any given time before a warning notification is sent out using an e-mail notifier.

Value Description A number of Worker thread.

Value Type int

Value Type Description Integer

```
(233) <web-container> <context-group> <webserver-connection>
<txmax-listener> <thread-pool> <thread-state-notify> <restart-
threshold>
```

Description Sets the maximum number of blocked worker threads that may exist at any given time before a restart of the Web container is performed.

Value Description A number of Servlet threads.

Value Type int

Value Type Description Integer

```
(234) <web-container> <context-group> <webserver-connection>
<txmax-listener> <thread-pool> <thread-state-notify> <notifier-
id>
```

Description The ID of the email notifier to use.

Value Type token

```
(235) <web-container> <context-group> <webserver-connection>
<txmax-listener> <thread-pool> <thread-state-notify> <notify-
subject>
```

Description Sets the subject line of the e-mail that is sent out when the notify-treshold is exceeded.

Value Description A message allowing the receiver (the administrator) to quickly determine what this e-mail is about.

Value Type token

```
(236) <web-container> <context-group> <webserver-connection>
<txmax-listener> <thread-pool> <thread-state-notify> <restart-
subject>
```

Description Sets the subject line of the e-mail that is sent out when the notify-treshold is exceeded.

Value Description A message allowing the receiver (the administrator) to quickly determine what this e-mail is about.

Value Type token

```
( 237 ) <web-container> <context-group> <webserver-connection>
<tmax-listener> <postdata-read-timeout>
```

Description Setting Maximum waiting time to read post-data from web server/web client. Execute it through request.getInputStream().read() method.

Value Description Time Interval in Millisecond

Value Type nonNegativeLongType

Value Type Description This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.

Default Value 30000

Example <postdata-read-timeout>40000</postdata-read-timeout>

```
( 238 ) <web-container> <context-group> <webserver-connection>
<tmax-listener> <reconnect-timeout>
```

Description Some connections between the WebtoB server and the webtoB-listener may be disconnected during the operation. Reconnect-timeout (this element) is applied as a timeout value of the reconnect attempt that follows in such cases. If the timeout is expired with all of the current WebtoB connections disconnected and if some WebtoB backups are defined, then the Web container tries to fail-over to the next WebtoB backup. If that WebtoB backup is also failed, it tries the next and so on. If the last WebtoB backup is failed, it tries the (main) webtoB-listener again

Value Description The timeout value unit is milliseconds.

Value Type int

Value Type Description Long

Default Value -1

Defined Value -1

Infinite timeout period (the listener will attempt to reconnect forever).

0

No reconnection will ever be attempted

```
(239) <web-container> <context-group> <webserver-connection>
<tmax-listener> <tmax-address>
```

Description Set IP address of tmax server.

Value Description IP address of tmax server

Value Type token

Default Value 127.0.0.1

```
(240) <web-container> <context-group> <webserver-connection>
<tmax-listener> <server-group-name>
```

Description Group including tmax server to be connected.

Value Description tmax server group name

Value Type token

```
(241) <web-container> <context-group> <webserver-connection>
<tmax-listener> <server-name>
```

Description Name of tmax server to be connected.

Value Description Name of tmax server

Value Type token

```
(242) <web-container> <context-group> <webserver-connection>
<tmax-listener> <tmax-backup-address>
```

Description IP address of Tmax server used as backup.

Value Description IP address of tmax server

Value Type token

Default Value 127.0.0.1

```
(243) <web-container> <context-group> <webserver-connection>
<tmax-listener> <tmax-backup-port>
```

Description Port of Tmax server used as backup.

Value Description Port number which is greater than 0 .

Value Type nonNegativeLongType

Value Type Description This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.

```
(244) <web-container> <context-group> <webserver-connection>
<tmax-listener> <dispatcher-config-class>
```

Description This element configures dispatcher configuration class. At this time, it should be located in place set as classpath and class name should be full qualified class name.

Value Type token

```
(245) <web-container> <context-group> <session-config>
```

Description The settings below this element pertains to the internal management of session objects (HTTPSession objects) of the context group.

Child Elements (246) timeout?
 (247) shared?
 (248) persistent?

(246) <web-container> <context-group> <session-config> <timeout>

<i>Description</i>	This element determines the time after which a session object is considered stale and should be removed. The session timeout settings in web.xml takes precedence over any settings defined here,
<i>Value Description</i>	A time period expressed as a number of minutes after which a shared session should be removed.
<i>Value Type</i>	int
<i>Value Type Description</i>	Integer
<i>Default Value</i>	30
<i>Defined Value</i>	-1 When this value has been set, the inspection for session timeout would not be executed. Session will be preserved when other elements have not been set in web.xml.

(247) <web-container> <context-group> <session-config> <shared>

<i>Description</i>	This element activates or inactivates shared session of entire Contexts below context group. Session sharing means to be used in other Context when generating HttpSession object to a context
<i>Value Description</i>	Activating or inactivating this setting as Boolean value.
<i>Value Type</i>	boolean
<i>Value Type Description</i>	Boolean
<i>Default Value</i>	false
<i>Defined Value</i>	true Sharing session. false

Non-sharing session.

(248) <web-container> <context-group> <session-config>
<persistent>

Description If this element is set as "false" and Session Server is configured, Session object (<session-cluster>element of JEUSMain.xml) and property value on Servlet Context will be deleted when it is reloaded after being updated. However, if this element is set as "true" and Session Server is not defined, Session object and property value on Servlet Context will be maintained even when it is reloaded after being updated. If Session Server is defined, this element will be usesee(Doesn't affect Servlet engine). If Session Server is defined, Session Object and property of Servlet Context will be maintained despite of the value of this element.

Value Description Activating or inactivating this configuration as Boolean value.

Value Type boolean

Value Type Description Boolean

Default Value false

Defined Value true
 Activating the related configuration.

 false
 Inactivating the related configuration.

(249) <web-container> <context-group> **<print-error-to-browser>**

Description Determines whether detailed error descriptions should be sent to the browser in case of a server-side failure. These messages are useful during development time but should be turned of in a production environment.

Value Type boolean

Default Value false

```
(250) <web-container> <context-group> <encoding>
```

Child Elements

- (251)request-encoding?
- (254)response-encoding?
- (257)postdata-encoding?

```
(251) <web-container> <context-group> <encoding> <request-encoding>
```

Description This element governs the encoding to be applied to received requests from a Web browser. The encoding is applied when converting the HTTP byte array (including HTTP URI, query strings, cookies and any post data) to a java.lang.String object inside the Web container.

Child Elements

- (252)default?
- (253)forced?

```
(252) <web-container> <context-group> <encoding> <request-encoding> <default>
```

Description The default encoding to use if the user has not set the encoding explicitly through standard Servlet API, the forced tag is not used and if no encoding was specified in the HTTP request

Value Description Standard character encoding.

Value Type token

```
(253) <web-container> <context-group> <encoding> <request-encoding> <forced>
```

Description If specified, the encoding set here will always be used when converting HTTP data to java.lang.String objects. It thus has the highest priority.

Value Description A standard character encoding

Value Type token

```
(254) <web-container> <context-group> <encoding> <response-encoding>
```

Description This element determines which encoding to use for the Web container's responses when converting from `PrintWriter.println()` to a byte array and also sets the "XXX" part of the HTTP header "Content-Type: text/html; charset=XXX".

Child Elements (255) default?
(256) forced?

```
(255) <web-container> <context-group> <encoding> <response-encoding> <default>
```

Description The default encoding to use if the user has not set the encoding explicitly through standard Servlet API, the forced tag is not used and if no encoding was specified in the HTTP request

Value Description Standard character encoding.

Value Type token

```
(256) <web-container> <context-group> <encoding> <response-encoding> <forced>
```

Description If specified, the encoding set here will always be used when converting HTTP data to `java.lang.String` objects. It thus has the highest priority.

Value Description A standard character encoding

Value Type token

```
(257) <web-container> <context-group> <encoding> <postdata-encoding>
```

Description Setting encoded data sent using HTTP POST method.

Child Elements (258) default?
(259) forced?

```
(258) <web-container> <context-group> <encoding> <postdata-encoding> <default>
```

Description The default encoding to use if the user has not set the encoding explicitly through standard Servlet API, the forced tag is not used and if no encoding was specified in the HTTP request

Value Description Standard character encoding.

Value Type token

```
(259) <web-container> <context-group> <encoding> <postdata-encoding> <forced>
```

Description If specified, the encoding set here will always be used when converting HTTP data to java.lang.String objects. It thus has the highest priority.

Value Description A standard character encoding

Value Type token

```
(260) <web-container> <context-group> <jsp-engine>
```

Child Elements

- (261) keep-generated?
- (262) java-compiler?
- (263) jsp-work-dir?
- (264) compile-output-dir?
- (265) compile-option?
- (266) compile-encoding?
- (267) check-included-jspfile?

```
(261) <web-container> <context-group> <jsp-engine> <keep-generated>
```

Description This element determines availability of storing java source file after generating Servlet Class File through compiling source file generated from JSP page. These files are useful for debugging purpose.

Value Description This configuration activates or inactivates as Boolean value.

Value Type boolean

Value Type Description Boolean

Default Value true

```
(262) <web-container> <context-group> <jsp-engine> <java-  
compiler>
```

Description This element configures Java Compiler for compiling Java source code of eJSP. It is not recommended since default configuration is the most efficient way.

Value Description Character set and compiler configuring java compiler should be included within system path in order to execute.

Value Type token

Default Value com.sun.tools.javac

Defined Value sun.tools.javac
SUN Microsystems Java Compiler in which "tools.jar" is included is used (This configuration is recommended).

javac

This element configures external "javac" compiler including in system path.

com.sun.tools.javac

This is operated in serveral systems. If this operates, this is priotized option since it is more efficient and stable than sun.tools.javac.

```
(263) <web-container> <context-group> <jsp-engine> <jsp-work-  
dir>
```

Description JSP workdir is location in which Servlet source file and class files

are stored. Developer or administrator can use related directory for debugging purpose. This element is not generally needed to be used.

Value Description Absolute path used for compiling JSP.

Value Type token

(264) <web-container> <context-group> <jsp-engine> **<compile-output-dir>**

Description Through configuring this property, generated class files can be placed in other directory besides <jsp-work-dir> directory. If this property is not set, class files will be located in jspwork directory. This element is not commonly used.

Value Description Absolute path used for storing JSP class files.

Value Type token

(265) <web-container> <context-group> <jsp-engine> **<compile-option>**

Description This is option used as Servlet compiler and is not commonly used.

Value Description Compile option property used for compiling (Referencing <java-compiler>element)

Value Type token

(266) <web-container> <context-group> <jsp-engine> **<compile-encoding>**

Description This property is value of configuring in ?encoding option when compiling generated Servlet source file through parsing JSP files. This configuration is not commonly used. Web Container automatically selects proper configuration.

Value Description Character set checking encoding scheme to use. For additional information, please refer to javac compiler documentation.

<i>Value Type</i>	token
<i>Defined Value</i>	default This is automatically selected in Web Container.

```
(267) <web-container> <context-group> <jsp-engine> <check-  
included-jspfile>
```

<i>Description</i>	If this property is set as "true", it enables to recompile related JSP file when executing included JSP file. Included JSP files are not checked for being updated.
<i>Value Description</i>	Activating or inactivating this configuration as Boolean value.
<i>Value Type</i>	boolean
<i>Value Type Description</i>	Boolean
<i>Default Value</i>	false
<i>Defined Value</i>	true Activating related configuration. false Inactivating related configuration.

```
(268) <web-container> <context-group> <logging>
```

<i>Child Elements</i>	(269)user-log? (322)access-log?
-----------------------	------------------------------------

```
(269) <web-container> <context-group> <logging> <user-log>
```

<i>Description</i>	This element sets the defined message handling through ServletContext.log() method.
<i>Child Elements</i>	(270)level? (271)use-parent-handlers? (272)filter-class?

(273)handler?

```
( 270 ) <web-container> <context-group> <logging> <user-log>
<level>
```

Description Level of logging is set for this field. Please refer to Level class of logging API of J2SE.

Value Type loggingLevelType

Default Value INFO

Defined Value FATAL

NOTICE

INFORMATION

DEBUG

fatal

notice

information

debug

SEVERE

WARNING

INFO

CONFIG

FINE

FINER

FINEST

ALL

severe

warning

info

config

fine

finer

finest

all

FATAL

NOTICE

INFORMATION

DEBUG

fatal

notice

information

debug

SEVERE

WARNING

INFO

CONFIG

FINE

FINER

FINEST

ALL

severe

warning

info

config

fine

finer

finest

all

```
(271) <web-container> <context-group> <logging> <user-log> <use-
parent-handlers>
```

Description This determines whether handlers of parent logger will be used for this logger.

Value Type boolean

Default Value false

```
(272) <web-container> <context-group> <logging> <user-log>
<filter-class>
```

Description Fully qualified class name of filter class set in this logger is set for this field.

Value Type token

Example <filter-
class>com.tmax.logging.filter.MyFilter</filter-
class>

```
(273) <web-container> <context-group> <logging> <user-log>
<handler>
```

Description Handler to use in logger is set for this field.

Child Elements (274)console-handler
 (279)file-handler
 (289)smtp-handler
 (300)socket-handler
 (307)user-handler

```
(274) <web-container> <context-group> <logging> <user-log>
<handler> <console-handler>
```

Description This handler will be used when logging is left in the screen.

Child Elements (275)name
 (276)level?
 (277)encoding?

(278)filter-class?

```
(275) <web-container> <context-group> <logging> <user-log>
<handler> <console-handler> <name>
```

Description The name of handler will be set. At this moment, this name should be set as unique in a logger. This name is used for refering handler including tool.

Value Type token

Example <name>handler1</name>

```
(276) <web-container> <context-group> <logging> <user-log>
<handler> <console-handler> <level>
```

Description The level of this handler will be set for this field. In case where level of message going through logger is included in this handler level, the message will be displayed through this handler.

Value Type loggingLevelType

Default Value FINEST

Defined Value FATAL

NOTICE

INFORMATION

DEBUG

fatal

notice

information

debug

SEVERE

WARNING

INFO

CONFIG

FINE

FINER

FINEST

ALL

severe

warning

info

config

fine

finer

finest

all

FATAL

NOTICE

INFORMATION

DEBUG

fatal

notice

information

debug

SEVERE

WARNING

INFO

CONFIG

FINE

FINER

FINEST

ALL

severe

warning

info

config

fine

finer

finest

all

```
(277) <web-container> <context-group> <logging> <user-log>
<handler> <console-handler> <encoding>
```

Description Encoding in which this handler leaves message is set for this field.

Value Type token

```
(278) <web-container> <context-group> <logging> <user-log>
<handler> <console-handler> <filter-class>
```

Description Fully qualified class name of filter class set in this handler is set for this field.

Value Type token

Example <filter-
class>com.tmax.logging.filter.MyFilter</filter-
class>

```
(279) <web-container> <context-group> <logging> <user-log>
<handler> <file-handler>
```

Description This is the handler to be used in case of displaying logging as file.

Child Elements (280)name
 (281)level?
 (282)encoding?
 (283)filter-class?

(284)file-name?
 (285)valid-day
 (286)valid-hour
 (287)buffer-size?
 (288)append?

```
(280) <web-container> <context-group> <logging> <user-log>
<handler> <file-handler> <name>
```

Description The name of handler will be set. At this moment, this name should be set as unique in a logger. This name is used for refering handler including tool.

Value Type token

Example <name>handler1</name>

```
(281) <web-container> <context-group> <logging> <user-log>
<handler> <file-handler> <level>
```

Description The level of this handler will be set for this field. In case where level of message going through logger is included in this handler level, the message will be displayed through this handler.

Value Type loggingLevelType

Default Value FINEST

Defined Value FATAL

NOTICE

INFORMATION

DEBUG

fatal

notice

information

debug

SEVERE

WARNING

INFO

CONFIG

FINE

FINER

FINEST

ALL

severe

warning

info

config

fine

finer

finest

all

FATAL

NOTICE

INFORMATION

DEBUG

fatal

notice

information

debug

SEVERE

WARNING

INFO

CONFIG

FINE

FINER

FINEST

ALL

severe

warning

info

config

fine

finer

finest

all

```
(282) <web-container> <context-group> <logging> <user-log>
<handler> <file-handler> <encoding>
```

Description Encoding in which this handler leaves message is set for this field.

Value Type token

```
(283) <web-container> <context-group> <logging> <user-log>
<handler> <file-handler> <filter-class>
```

Description Fully qualified class name of filter class set in this handler is set for this field.

Value Type token

Example <filter-
class>com.tmax.logging.filter.MyFilter</filter-
class>

```
(284) <web-container> <context-group> <logging> <user-log>
<handler> <file-handler> <file-name>
```

Description File name in which this handler uses will be set for this field. If

user has not set for this field, default file name of each logger will be used. Please refer to JEUS Server Manual for each default file name.

Value Type token

Example <file-name>C:\logs\mylog.log</file-name>

```
(285) <web-container> <context-group> <logging> <user-log>
<handler> <file-handler> <valid-day>
```

Description File in which this handler uses will be used and updated frequently for certain set amount of time . This setting is used for changing file as unit of day. In this case, used date will be followed right after file name.

Value Description day

Value Type off-intType

Value Type Description The default value is in non-negative int type. In case where the value is -1, it is considered as off.

Example <valid-day>1</valid-day>

```
(286) <web-container> <context-group> <logging> <user-log>
<handler> <file-handler> <valid-hour>
```

Description In case of using and updating frequently on the file in which this handler uses for a certain amount of time. This setting is used for updating time unit. In this case, used date and time will be automatically followed right after file name in which this handler uses.

Value Description This represents in unit of hour and it should get divisor of 24 + N*24(n is integer greater than 0).

Value Type off-intType

Value Type Description The default value is in non-negative int type. In case where the value is -1, it is considered as off.

Example <valid-hour>3</valid-hour>

```
(287) <web-container> <context-group> <logging> <user-log>
<handler> <file-handler> <b>buffer-size</b>
```

Description The size of buffer is set when displaying of this handler as file.

Value Description The unit in byte. [Performance Recommendation]: The logging performance will be improved as this value is bigger even though message will be displayed with delay.

Value Type nonNegativeIntType

Value Type Description This element is int type which is over 0. Thus, it includes values which are over 0 in the range of int type.

Default Value 1024

```
(288) <web-container> <context-group> <logging> <user-log>
<handler> <file-handler> <b>append</b>
```

Description In case where file in which this handler uses already exists, this element will be used for determining whether to be written after the file name. If it has been set as false, existing file will be deleted.

Value Type boolean

Default Value true

```
(289) <web-container> <context-group> <logging> <user-log>
<handler> <b>smtp-handler</b>
```

Description This is the handler to use when sending logging to email.
[Performance Recommendation]: There is a warning when a huge amount of email exist without proper filter.

Child Elements (290)name
 (291)level?
 (292)encoding?
 (293)filter-class?

```
(294)smtp-host-address
(295)from-address
(296)to-address
(297)cc-address?
(298)bcc-address?
(299)send-for-all-messages?
```

```
(290) <web-container> <context-group> <logging> <user-log>
<handler> <smtp-handler> <name>
```

Description The name of handler will be set. At this moment, this name should be set as unique in a logger. This name is used for refering handler including tool.

Value Type token

Example <name>handler1</name>

```
(291) <web-container> <context-group> <logging> <user-log>
<handler> <smtp-handler> <level>
```

Description The level of this handler will be set for this field. In case where level of message going through logger is included in this handler level, the message will be displayed through this handler.

Value Type loggingLevelType

Default Value FINEST

Defined Value FATAL

NOTICE

INFORMATION

DEBUG

fatal

notice

information

debug

SEVERE

WARNING

INFO

CONFIG

FINE

FINER

FINEST

ALL

severe

warning

info

config

fine

finer

finest

all

FATAL

NOTICE

INFORMATION

DEBUG

fatal

notice

information

debug

SEVERE

WARNING

INFO

CONFIG

FINE

FINER

FINEST

ALL

severe

warning

info

config

fine

finer

finest

all

```
(292) <web-container> <context-group> <logging> <user-log>
<handler> <smtp-handler> <encoding>
```

Description Encoding in which this handler leaves message is set for this field.

Value Type token

```
(293) <web-container> <context-group> <logging> <user-log>
<handler> <smtp-handler> <filter-class>
```

Description Fully qualified class name of filter class set in this handler is set for this field.

Value Type token

Example <filter-
class>com.tmax.logging.filter.MyFilter</filter-
class>

```
(294) <web-container> <context-group> <logging> <user-log>
<handler> <smtp-handler> <smtp-host-address>
```

Description The address of Smtip server to send email is set for this field.

Value Type token

```
(295) <web-container> <context-group> <logging> <user-log>
<handler> <smtp-handler> <from-address>
```

Description The address of sender for email is set for this field.

Value Type token

```
(296) <web-container> <context-group> <logging> <user-log>
<handler> <smtp-handler> <to-address>
```

Description The address of receiver of email is set for this field.

Value Type token

```
(297) <web-container> <context-group> <logging> <user-log>
<handler> <smtp-handler> <cc-address>
```

Description The cc-address of receiver is set for this field.

Value Type token

```
(298) <web-container> <context-group> <logging> <user-log>
<handler> <smtp-handler> <bcc-address>
```

Description The bcc-address of receiver is set for this field.

Value Type token

```
(299) <web-container> <context-group> <logging> <user-log>
<handler> <smtp-handler> <send-for-all-messages>
```

Description When messages enter into this handler through using log() method of logger registered in this handler, it determines whether to consider it as the sender object of email. If it has been set as false, messages which are only called through using specific send() method are transmitted.

Value Type boolean

Default Value false

```
(300) <web-container> <context-group> <logging> <user-log>
<handler> <socket-handler>
```

Description This element is handler used for sending to socket in which logging is set. [Performance Recommendation]: The performance will be decreased without using proper filter since logging message is transmitted as one Socket.

Child Elements (301)name
 (302)level?
 (303)encoding?
 (304)filter-class?
 (305)address
 (306)port

```
(301) <web-container> <context-group> <logging> <user-log>
<handler> <socket-handler> <name>
```

Description The name of handler will be set. At this moment, this name should be set as unique in a logger. This name is used for refering handler including tool.

Value Type token

Example <name>handler1</name>

```
(302) <web-container> <context-group> <logging> <user-log>
<handler> <socket-handler> <level>
```

Description The level of this handler will be set for this field. In case where level of message going through logger is included in this handler level, the message will be displayed through this handler.

Value Type loggingLevelType

Default Value FINEST

Defined Value

FATAL

NOTICE

INFORMATION

DEBUG

fatal

notice

information

debug

SEVERE

WARNING

INFO

CONFIG

FINE

FINER

FINEST

ALL

severe

warning

info

config

fine

finer

finest

all

FATAL

NOTICE

INFORMATION

DEBUG

fatal

notice

information

debug

SEVERE

WARNING

INFO

CONFIG

FINE

FINER

FINEST

ALL

severe

warning

info

config

fine

finer

finest

all

```
(303) <web-container> <context-group> <logging> <user-log>
<handler> <socket-handler> <encoding>
```

Description Encoding in which this handler leaves message is set for this field.

Value Type token

```
(304) <web-container> <context-group> <logging> <user-log>
<handler> <socket-handler> <filter-class>
```

Description Fully qualified class name of filter class set in this handler is set for

this field.

Value Type token

Example <filter-
 class>com.tmax.logging.filter.MyFilter</filter-
 class>

```
(305) <web-container> <context-group> <logging> <user-log>
<handler> <socket-handler> <address>
```

Description IP address of sending messages when generating this handler.

Value Type token

```
(306) <web-container> <context-group> <logging> <user-log>
<handler> <socket-handler> <port>
```

Description Port of sending messages when generating this handler.

Value Type nonNegativeIntType

Value Type Description This element is int type which is over 0. Thus, it includes values which are over 0 in the range of int type.

```
(307) <web-container> <context-group> <logging> <user-log>
<handler> <user-handler>
```

Description In case of using handler in which user has created according to J2SE logging API, this setting

Child Elements (308)handler-class
 (309)name
 (310)level?
 (311)encoding?
 (312)filter-class?
 (313)handler-property?
 (317)formatter-class?
 (318)formatter-property?

```
(308) <web-container> <context-group> <logging> <user-log>
```



```
<handler> <user-handler> <handler-class>
```

Description Fully qualified class name in which user has created is set. This class inherits java.util.logging.Handler and jeus.util.logging.JeusHandler will be constructed.

Value Type token

Example `<handler-class>com.tmax.logging.handler.MyHandler</handler-class>`

```
(309) <web-container> <context-group> <logging> <user-log>
<handler> <user-handler> <name>
```

Description The name of handler will be set. At this moment, this name should be set as unique in a logger. This name is used for refering handler including tool.

Value Type token

Example `<name>handler1</name>`

```
(310) <web-container> <context-group> <logging> <user-log>
<handler> <user-handler> <level>
```

Description The level of this handler will be set for this field. In case where level of message going through logger is included in this handler level, the message will be displayed through this handler.

Value Type loggingLevelType

Default Value FINEST

Defined Value FATAL

NOTICE

INFORMATION

DEBUG

fatal

notice

information

debug

SEVERE

WARNING

INFO

CONFIG

FINE

FINER

FINEST

ALL

severe

warning

info

config

fine

finer

finest

all

FATAL

NOTICE

INFORMATION

DEBUG

fatal

notice

information

debug

SEVERE

WARNING

INFO

CONFIG

FINE

FINER

FINEST

ALL

severe

warning

info

config

fine

finer

finest

all

```
(311) <web-container> <context-group> <logging> <user-log>
<handler> <user-handler> <encoding>
```

Description Encoding in which this handler leaves message is set for this field.

Value Type token

```
(312) <web-container> <context-group> <logging> <user-log>
<handler> <user-handler> <filter-class>
```

Description Fully qualified class name of filter class set in this handler is set for this field.

Value Type token

Example <filter-
class>com.tmax.logging.filter.MyFilter</filter-
class>

```
(313) <web-container> <context-group> <logging> <user-log>
<handler> <user-handler> <handler-property>
```

Description When handler is generated, property to parse is set. These properties are delivered to handler as being stored in Map object as key-value through JeusHandler.setProperty() method.

Child Elements (314)property*

```
(314) <web-container> <context-group> <logging> <user-log>
<handler> <user-handler> <handler-property> <property>
```

Description Property to be delivered to handler is set.

Child Elements (315)key
(316)value

```
(315) <web-container> <context-group> <logging> <user-log>
<handler> <user-handler> <handler-property> <property> <key>
```

Description This element is the key value of property.

Value Type token

```
(316) <web-container> <context-group> <logging> <user-log>
<handler> <user-handler> <handler-property> <property> <value>
```

Description The value of property.

Value Type token

```
(317) <web-container> <context-group> <logging> <user-log>
<handler> <user-handler> <formatter-class>
```

Description Fully qualified class name of formatter in which this handler is used. This class inherits java.util.logging.Formatter and jeus.util.logging.JeusFormatter will be constructed.

Value Type token

Default Value jeus.util.logging.SimpleFormatter

Example `<formatter-
class>com.tmax.logging.handler.MyHandler</formatter-
class>`

```
(318) <web-container> <context-group> <logging> <user-log>
<handler> <user-handler> <formatter-property>
```

Description Property to parse to formatter when handler is generated. This properties are delivered to formatter through using JeusFormatter.setProperty() method stored in Map object.

Child Elements (319)property*

```
(319) <web-container> <context-group> <logging> <user-log>
<handler> <user-handler> <formatter-property> <property>
```

Description Property to be delivered to handler is set.

Child Elements (320)key
 (321)value

```
(320) <web-container> <context-group> <logging> <user-log>
<handler> <user-handler> <formatter-property> <property> <key>
```

Description This element is the key value of property.

Value Type token

```
(321) <web-container> <context-group> <logging> <user-log>
<handler> <user-handler> <formatter-property> <property> <value>
```

Description The value of property.

Value Type token

```
(322) <web-container> <context-group> <logging> <access-log>
```

Description This element configures on handling log of client of Container.

Child Elements (323)enable?
 (324)enable-analysis?

(325)level?
 (326)use-parent-handlers?
 (327)filter-class?
 (328)handler?
 (377)format?

(323) <web-container> <context-group> <logging> <access-log>
<enable>

Description Setting for the availabilitiy of analyzing an access logging.

Value Type boolean

Default Value true

(324) <web-container> <context-group> <logging> <access-log>
<enable-analysis>

Description Setting for the availability of analyzing an access logging.

Value Type boolean

Default Value false

(325) <web-container> <context-group> <logging> <access-log>
<level>

Description Level of logging is set for this field. Please refer to Level class of logging API of J2SE.

Value Type loggingLevelType

Default Value INFO

Defined Value FATAL

NOTICE

INFORMATION

DEBUG

fatal

notice

information

debug

SEVERE

WARNING

INFO

CONFIG

FINE

FINER

FINEST

ALL

severe

warning

info

config

fine

finer

finest

all

FATAL

NOTICE

INFORMATION

DEBUG

fatal

notice

information

debug

SEVERE

WARNING

INFO

CONFIG

FINE

FINER

FINEST

ALL

severe

warning

info

config

fine

finer

finest

all

```
( 326 ) <web-container> <context-group> <logging> <access-log>  
<use-parent-handlers>
```

<i>Description</i>	This determines whether handlers of parent logger will be used for this logger.
--------------------	---

<i>Value Type</i>	boolean
-------------------	---------

<i>Default Value</i>	false
----------------------	-------

```
( 327 ) <web-container> <context-group> <logging> <access-log>  
<filter-class>
```

<i>Description</i>	Fully qualified class name of filter class set in this logger is set for this field.
--------------------	--

<i>Value Type</i>	token
-------------------	-------

Example <filter-
 class>com.tmax.logging.filter.MyFilter</filter-
 class>

```
(328) <web-container> <context-group> <logging> <access-log>
<handler>
```

Description Handler to use in logger is set for this field.

Child Elements (329)console-handler
 (334)file-handler
 (344)smtp-handler
 (355)socket-handler
 (362)user-handler

```
(329) <web-container> <context-group> <logging> <access-log>
<handler> <console-handler>
```

Description This handler will be used when logging is left in the screen.

Child Elements (330)name
 (331)level?
 (332)encoding?
 (333)filter-class?

```
(330) <web-container> <context-group> <logging> <access-log>
<handler> <console-handler> <name>
```

Description The name of handler will be set. At this moment, this name should be set as unique in a logger. This name is used for refering handler including tool.

Value Type token

Example <name>handler1</name>

```
(331) <web-container> <context-group> <logging> <access-log>
<handler> <console-handler> <level>
```

Description The level of this handler will be set for this field. In case where level of message going through logger is included in this handler

level, the message will be displayed through this handler.

<i>Value Type</i>	loggingLevelType
-------------------	------------------

<i>Default Value</i>	FINEST
----------------------	--------

<i>Defined Value</i>	FATAL
----------------------	-------

NOTICE

INFORMATION

DEBUG

fatal

notice

information

debug

SEVERE

WARNING

INFO

CONFIG

FINE

FINER

FINEST

ALL

severe

warning

info

config

fine

finer

finest

all

FATAL

NOTICE

INFORMATION

DEBUG

fatal

notice

information

debug

SEVERE

WARNING

INFO

CONFIG

FINE

FINER

FINEST

ALL

severe

warning

info

config

fine

finer

finest

all

```
(332) <web-container> <context-group> <logging> <access-log>
<handler> <console-handler> <encoding>
```

Description Encoding in which this handler leaves message is set for this field.

Value Type token

```
(333) <web-container> <context-group> <logging> <access-log>
<handler> <console-handler> <filter-class>
```

<i>Description</i>	Fully qualified class name of filter class set in this handler is set for this field.
<i>Value Type</i>	token
<i>Example</i>	<pre><filter- class>com.tmax.logging.filter.MyFilter</filter- class></pre>

```
(334) <web-container> <context-group> <logging> <access-log>
<handler> <file-handler>
```

<i>Description</i>	This is the handler to be used in case of displaying logging as file.
<i>Child Elements</i>	<pre>(335)name (336)level? (337)encoding? (338)filter-class? (339)file-name? (340)valid-day (341)valid-hour (342)buffer-size? (343)append?</pre>

```
(335) <web-container> <context-group> <logging> <access-log>
<handler> <file-handler> <name>
```

<i>Description</i>	The name of handler will be set. At this moment, this name should be set as unique in a logger. This name is used for refering handler including tool.
<i>Value Type</i>	token
<i>Example</i>	<pre><name>handler1</name></pre>

```
(336) <web-container> <context-group> <logging> <access-log>
<handler> <file-handler> <level>
```

<i>Description</i>	The level of this handler will be set for this field. In case where level of message going through logger is included in this handler level, the message will be displayed through this handler.
<i>Value Type</i>	loggingLevelType
<i>Default Value</i>	FINEST
<i>Defined Value</i>	FATAL
	NOTICE
	INFORMATION
	DEBUG
	fatal
	notice
	information
	debug
	SEVERE
	WARNING
	INFO
	CONFIG
	FINE
	FINER
	FINEST

ALL

severe

warning

info

config

fine

finer

finest

all

FATAL

NOTICE

INFORMATION

DEBUG

fatal

notice

information

debug

SEVERE

WARNING

INFO

CONFIG

FINE

FINER

FINEST

ALL

severe

warning

info

config

fine

finer

finest

all

```
(337) <web-container> <context-group> <logging> <access-log>  
<handler> <file-handler> <encoding>
```

Description Encoding in which this handler leaves message is set for this field.

Value Type token

```
(338) <web-container> <context-group> <logging> <access-log>
<handler> <file-handler> <filter-class>
```

Description Fully qualified class name of filter class set in this handler is set for this field.

Value Type token

Example <filter-
class>com.tmax.logging.filter.MyFilter</filter-
class>

```
(339) <web-container> <context-group> <logging> <access-log>
<handler> <file-handler> <file-name>
```

Description File name in which this handler uses will be set for this field. If user has not set for this field, default file name of each logger will be used. Please refer to JEUS Server Manual for each default file name.

Value Type token

Example <file-name>C:\logs\mylog.log</file-name>

```
(340) <web-container> <context-group> <logging> <access-log>
<handler> <file-handler> <valid-day>
```

Description File in which this handler uses will be used and updated frequently for certain set amount of time . This setting is used for changing file as unit of day. In this case, used date will be followed right after file name.

Value Description day

Value Type off-intType

Value Type Description The default value is in non-negative int type. In case where the

value is -1, it is considered as off.

Example `<valid-day>1</valid-day>`

```
(341) <web-container> <context-group> <logging> <access-log>
<handler> <file-handler> <valid-hour>
```

Description In case of using and updating frequently on the file in which this handler uses for a certain amount of time. This setting is used for updating time unit. In this case, used date and time will be automaticallly followed right after file name in which this handler uses.

Value Description This represents in unit of hour and it should get divisor of 24 + N*24(n is integer greater than 0).

Value Type off-intType

Value Type Description The default value is in non-negative int type. In case where the value is -1, it is considered as off.

Example `<valid-hour>3</valid-hour>`

```
(342) <web-container> <context-group> <logging> <access-log>
<handler> <file-handler> <buffer-size>
```

Description The size of buffer is set when displaying of this handler as file.

Value Description The unit in byte. [Performance Recommendation]: The logging performance will be improved as this value is bigger even though message will be displayed with delay.

Value Type nonNegativeIntType

Value Type Description This element is int type which is over 0. Thus, it includes values which are over 0 in the range of int type.

Default Value 1024

```
(343) <web-container> <context-group> <logging> <access-log>
<handler> <file-handler> <append>
```

<i>Description</i>	In case where file in which this handler uses already exists, this element will be used for determining whether to be written after the file name. If it has been set as false, existing file will be deleted.
<i>Value Type</i>	boolean
<i>Default Value</i>	true

```
(344) <web-container> <context-group> <logging> <access-log>
<handler> <smtp-handler>
```

<i>Description</i>	This is the handler to use when sending logging to email. [Performance Recommendation]: There is a warning when a huge amount of email exist without proper filter.
--------------------	--

<i>Child Elements</i>	(345)name (346)level? (347)encoding? (348)filter-class? (349)smtp-host-address (350)from-address (351)to-address (352)cc-address? (353)bcc-address? (354)send-for-all-messages?
-----------------------	--

```
(345) <web-container> <context-group> <logging> <access-log>
<handler> <smtp-handler> <name>
```

<i>Description</i>	The name of handler will be set. At this moment, this name should be set as unique in a logger. This name is used for referring handler including tool.
<i>Value Type</i>	token
<i>Example</i>	<name>handler1</name>

```
(346) <web-container> <context-group> <logging> <access-log>
<handler> <smtp-handler> <level>
```

<i>Description</i>	The level of this handler will be set for this field. In case where level of message going through logger is included in this handler level, the message will be displayed through this handler.
<i>Value Type</i>	loggingLevelType
<i>Default Value</i>	FINEST
<i>Defined Value</i>	FATAL
	NOTICE
	INFORMATION
	DEBUG
	fatal
	notice
	information
	debug
	SEVERE
	WARNING
	INFO
	CONFIG
	FINE
	FINER
	FINEST

ALL

severe

warning

info

config

fine

finer

finest

all

FATAL

NOTICE

INFORMATION

DEBUG

fatal

notice

information

debug

SEVERE

WARNING

INFO

CONFIG

FINE

FINER

FINEST

ALL

severe

warning

info

config

fine

finer

finest

all

```
(347) <web-container> <context-group> <logging> <access-log>  
<handler> <smtp-handler> <encoding>
```


Description Encoding in which this handler leaves message is set for this field.

Value Type token

```
(348) <web-container> <context-group> <logging> <access-log>
<handler> <smtp-handler> <filter-class>
```

Description Fully qualified class name of filter class set in this handler is set for this field.

Value Type token

Example <filter-
class>com.tmax.logging.filter.MyFilter</filter-
class>

```
(349) <web-container> <context-group> <logging> <access-log>
<handler> <smtp-handler> <smtp-host-address>
```

Description The address of Smtip server to send email is set for this field.

Value Type token

```
(350) <web-container> <context-group> <logging> <access-log>
<handler> <smtp-handler> <from-address>
```

Description The address of sender for email is set for this field.

Value Type token

```
(351) <web-container> <context-group> <logging> <access-log>
<handler> <smtp-handler> <to-address>
```

Description The address of receiver of email is set for this field.

Value Type token

```
(352) <web-container> <context-group> <logging> <access-log>
<handler> <smtp-handler> <cc-address>
```

Description The cc-address of receiver is set for this field.

Value Type token

```
( 353 ) <web-container> <context-group> <logging> <access-log>
<handler> <smtp-handler> <bcc-address>
```

Description The bcc-address of receiver is set for this field.

Value Type token

```
( 354 ) <web-container> <context-group> <logging> <access-log>
<handler> <smtp-handler> <send-for-all-messages>
```

Description When messages enter into this handler through using log() method of logger registered in this handler, it determines whether to consider it as the sender object of email. If it has been set as false, messages which are only called through using specific send() method are transmitted.

Value Type boolean

Default Value false

```
( 355 ) <web-container> <context-group> <logging> <access-log>
<handler> <socket-handler>
```

Description This element is handler used for sending to socket in which logging is set. [Performance Recommendation]: The performance will be decreased without using proper filter since logging message is transmitted as one Socket.

Child Elements (356)name
 (357)level?
 (358)encoding?
 (359)filter-class?
 (360)address
 (361)port

```
( 356 ) <web-container> <context-group> <logging> <access-log>
<handler> <socket-handler> <name>
```

Description The name of handler will be set. At this moment, this name should be set as unique in a logger. This name is used for refering handler including tool.

Value Type token

Example <name>handler1</name>

```
(357) <web-container> <context-group> <logging> <access-log>
<handler> <socket-handler> <level>
```

Description The level of this handler will be set for this field. In case where level of message going through logger is included in this handler level, the message will be displayed through this handler.

Value Type loggingLevelType

Default Value FINEST

Defined Value FATAL

NOTICE

INFORMATION

DEBUG

fatal

notice

information

debug

SEVERE

WARNING

INFO

CONFIG

FINE

FINER

FINEST

ALL

severe

warning

info

config

fine

finer

finest

all

FATAL

NOTICE

INFORMATION

DEBUG

fatal

notice

information

debug

SEVERE

WARNING

INFO

CONFIG

FINE

FINER

FINEST

ALL

severe

warning

info

config

fine

finer

finest

all

```
(358) <web-container> <context-group> <logging> <access-log>
<handler> <socket-handler> <b>encoding</b>
```

Description Encoding in which this handler leaves message is set for this field.

Value Type token

```
(359) <web-container> <context-group> <logging> <access-log>
<handler> <socket-handler> <b>filter-class</b>
```

Description Fully qualified class name of filter class set in this handler is set for this field.

Value Type token

Example <filter-
class>com.tmax.logging.filter.MyFilter</filter-
class>

```
(360) <web-container> <context-group> <logging> <access-log>
<handler> <socket-handler> <b>address</b>
```

Description IP address of sending messages when generating this handler.

Value Type token

```
(361) <web-container> <context-group> <logging> <access-log>
<handler> <socket-handler> <b>port</b>
```

Description Port of sending messages when generating this handler.

Value Type nonNegativeIntType

Value Type Description This element is int type which is over 0. Thus, it includes values which are over 0 in the range of int type.

```
(362) <web-container> <context-group> <logging> <access-log>
<handler> <user-handler>
```

Description In case of using handler in which user has created according to J2SE logging API, this setting

Child Elements

```
(363) handler-class
(364) name
(365) level?
(366) encoding?
(367) filter-class?
(368) handler-property?
(372) formatter-class?
(373) formatter-property?
```

```
(363) <web-container> <context-group> <logging> <access-log>
<handler> <user-handler> <handler-class>
```

Description Fully qualified class name in which user has created is set. This class inherits java.util.logging.Handler and jeus.util.logging.JeusHandler will be constructed.

Value Type token

Example

```
<handler-
class>com.tmax.logging.handler.MyHandler</handler-
class>
```

```
(364) <web-container> <context-group> <logging> <access-log>
<handler> <user-handler> <name>
```

Description The name of handler will be set. At this moment, this name should be set as unique in a logger. This name is used for refering handler including tool.

Value Type token

Example <name>handler1</name>

```
(365) <web-container> <context-group> <logging> <access-log>
<handler> <user-handler> <level>
```

Description The level of this handler will be set for this field. In case where level of message going through logger is included in this handler level, the message will be displayed through this handler.

Value Type loggingLevelType

Default Value FINEST

Defined Value FATAL

NOTICE

INFORMATION

DEBUG

fatal

notice

information

debug

SEVERE

WARNING

INFO

CONFIG

FINE

FINER

FINEST

ALL

severe

warning

info

config

fine

finer

finest

all

FATAL

NOTICE

INFORMATION

DEBUG

fatal

notice

information

debug

SEVERE

WARNING

INFO

CONFIG

FINE

FINER

FINEST

ALL

severe

warning

info

config

fine

finer

finest

all

```
(366) <web-container> <context-group> <logging> <access-log>
<handler> <user-handler> <encoding>
```

Description Encoding in which this handler leaves message is set for this field.

Value Type token

```
(367) <web-container> <context-group> <logging> <access-log>
<handler> <user-handler> <filter-class>
```

Description Fully qualified class name of filter class set in this handler is set for this field.

Value Type token

Example

```
<filter-
class>com.tmax.logging.filter.MyFilter</filter-
class>
```

```
(368) <web-container> <context-group> <logging> <access-log>
<handler> <user-handler> <handler-property>
```

Description When handler is generated, property to parse is set. These properties are delivered to handler as being stored in Map object as key-value through JeusHandler.setProperty() method.

Child Elements (369)property*

```
(369) <web-container> <context-group> <logging> <access-log>
<handler> <user-handler> <handler-property> <property>
```

Description Property to be delivered to handler is set.

Child Elements

```
(370)key
(371)value
```

```
(370) <web-container> <context-group> <logging> <access-log>
<handler> <user-handler> <handler-property> <property> <key>
```

Description This element is the key value of property.

Value Type token

```
( 371 ) <web-container> <context-group> <logging> <access-log>
<handler> <user-handler> <handler-property> <property> <value>
```

Description The value of property.

Value Type token

```
( 372 ) <web-container> <context-group> <logging> <access-log>
<handler> <user-handler> <formatter-class>
```

Description Fully qualified class name of formatter in which this handler is used. This class inherits java.util.logging.Formatter and jeus.util.logging.JeusFormatter will be constructed.

Value Type token

Default Value jeus.util.logging.SimpleFormatter

Example <formatter-
 class>com.tmax.logging.handler.MyHandler</formatter-
 class>

```
( 373 ) <web-container> <context-group> <logging> <access-log>
<handler> <user-handler> <formatter-property>
```

Description Property to parse to formatter when handler is generated. This properties are delivered to formatter through using JeusFormatter.setProperty() method stored in Map object.

Child Elements (374)property*

```
( 374 ) <web-container> <context-group> <logging> <access-log>
<handler> <user-handler> <formatter-property> <property>
```

Description Property to be delivered to handler is set.

Child Elements (375)key

(376)value

```
( 375 ) <web-container> <context-group> <logging> <access-log>
<handler> <user-handler> <formatter-property> <property> <b>key</b>
```

Description This element is the key value of property.

Value Type token

```
( 376 ) <web-container> <context-group> <logging> <access-log>
<handler> <user-handler> <formatter-property> <property> <b>value</b>
```

Description The value of property.

Value Type token

```
( 377 ) <web-container> <context-group> <logging> <access-log>
<b>format</b>
```

Description Setting the format for time-stamp. This configuration is only used for access log.

Value Type token

```
( 378 ) <web-container> <context-group> <b>session-cluster</b>
```

Child Elements

- (379)session-routing?
- (380)shared?
- (381)session-server?

```
( 379 ) <web-container> <context-group> <session-cluster>
<b>session-routing</b>
```

Description Session routing is a technique to manage sessions across clustered Web containers. This essentially means that special information, a Web container "ID number", will be attached to each outgoing cookie (session object). Using this ID, Web servers may later redirect (or "route") a client request back to the same Web container that issued the ID (that Web container also holds the actual HttpSession object associated with the ID). <p>This technique is rather efficient in terms of performance but is not

completely reliable since a failure in the Web container while processing a client request might cause the session object data to be lost. For better reliability it should be combined with the use of a session server as described in the `<session-server>` element. `<p>`Session routing is used when either WebtoB or Apache Web servers are connected to multiple Web containers. WebtoB handles session routing without any special configurations while Apache will require some setup before it can be used with this feature.

<i>Value Description</i>	Activating or inactivating this configuration as Boolean value.
<i>Value Type</i>	boolean
<i>Value Type Description</i>	Boolean
<i>Default Value</i>	true
<i>Defined Value</i>	true Activating session sharing between Cluster Web Containers. false Inactivating session sharing between Cluster Web Containers.

```
(380) <web-container> <context-group> <session-cluster> <shared>
```

<i>Description</i>	When session routing is set as false and session server is used, all the context group and contexts in it can share the session. However, session can be shared among groups if session server is not used or session routing is true. For such environment, this option should be set as true for maintaining session among context groups.
<i>Value Type</i>	boolean
<i>Default Value</i>	false

```
(381) <web-container> <context-group> <session-cluster>
```

<session-server>*Description*

This element enables and configures the use of a session server for session sharing and backup functionality. The session server is configured for use with this context group but it will not run in the Web container but rather in a JEUS server. Session servers are therefore configured in the file JEUSMain.xml. This element will merely configure the connection to such a session server.<p> The use of a session server essentially means that all session data will be "backed up" in the session server so that it can be recreated at a later time if the Web container experiences a failure and the data is lost. It also enables other Web containers than the one that originally created the session to fetch the session object, which is good if one Web container is downed and another container has to take over its session information.<p>For even better reliability, each Web container may also be configured to use a backup session server in case the main server is downed.

Child Elements

(382) server-name
 (383) max
 (384) min?
 (385) step?
 (386) backup-server-name?
 (387) connect-timeout?
 (388) read-timeout?

```
(382) <web-container> <context-group> <session-cluster>
<session-server> <server-name>
```

Description

This element configures Session Server name(JNDI export name) for using Session Server in this Context group This export name sets the value in <session-manager><name> in JEUSMain.xml

Value Description

JNDI export name distinguishes Primary Session Server to be used in this Context group.

Value Type

token

```
(383) <web-container> <context-group> <session-cluster>
```

```
<session-server> <max>
```

<i>Description</i>	This element is used for configuring maximum number of connection pools with Session Server for maintaining them
<i>Value Description</i>	Maximum number used for maintaining connection pool.
<i>Value Type</i>	nonNegativeLongType
<i>Value Type Description</i>	This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.

```
(384) <web-container> <context-group> <session-cluster>
<session-server> <min>
```

<i>Value Description</i>	A minimum number of connections to maintain.
<i>Value Type</i>	nonNegativeLongType
<i>Value Type Description</i>	This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.

```
(385) <web-container> <context-group> <session-cluster>
<session-server> <step>
```

<i>Description</i>	This element sets the number of connections to add connection pool with Session Server when it is needed to generate a new connection.
<i>Value Type</i>	nonNegativeLongType
<i>Value Type Description</i>	This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.
<i>Default Value</i>	1

```
(386) <web-container> <context-group> <session-cluster>
<session-server> <backup-server-name>
```

<i>Description</i>	The name of the backup session server to use if the main session server is unreachable.
--------------------	---

Value Description JNDI export name of Session Server to be used as Backup Session Server.

Value Type token

```
(387) <web-container> <context-group> <session-cluster>
<session-server> <connect-timeout>
```

Description Maximum time taken for waiting for establishing a new connection with Session Server when connection is initially generated.

Value Type nonNegativeLongType

Value Type Description This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.

Default Value 5000

```
(388) <web-container> <context-group> <session-cluster>
<session-server> <read-timeout>
```

Description Maximum number of time taken for reading the data from Session Server.

Value Type nonNegativeLongType

Value Type Description This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.

Default Value 20000

```
(389) <web-container> <context-group> <response-header>
```

Child Elements (390)session-id-cookie?
(397)custom-header?

```
(390) <web-container> <context-group> <response-header>
<session-id-cookie>
```

Description This element constructs the property of session id cookie sending to client by Web Container for maintaining client session.

Child Elements

(391) `jsessionid-name?`
(392) `version?`
(393) `domain?`
(394) `path?`
(395) `max-age?`
(396) `secure?`

```
(391) <web-container> <context-group> <response-header>
<session-id-cookie> <jsessionid-name>
```

Description This configuration will be used when using other names instead of standard name "JSESSIONID" as the name of session ID.

Value Type token

Default Value JSESSIONID

```
(392) <web-container> <context-group> <response-header>
<session-id-cookie> <version>
```

Description Configuring cookie ID version.

Value Description Either 0 or 1

Value Type token

Value Type Description Integer

Default Value 0

Defined Value 0
Default NS cookie type.

1
RFC spec.default cookie type

```
(393) <web-container> <context-group> <response-header>
<session-id-cookie> <domain>
```

Description Sets the domain name of the server that sent the session id cookie.

The cookie will only be returned with requests to this domain.

Value Description A valid domain name. It must start with a dot and must not specify the hostname (by RFC spec: RFC-2019).

Value Type token

```
(394) <web-container> <context-group> <response-header>
<session-id-cookie> <path>
```

Description Sets the URL path within a domain for which a session id cookie will be sent. The cookie is sent along with any request to this URL as long as the domain also matches (see the `<domain>` element). For instance, if the path is set to `/examples`, the domain to `.foo.com` and the client requests `www.foo.com/examples`, the client's cookie will be sent along with the request.

Value Description Proper URL path below domain/

Value Type token

Default Value /

Defined Value /

Cookie is sent despite of the request path.

```
(395) <web-container> <context-group> <response-header>
<session-id-cookie> <max-age>
```

Description Sets the "expires" attribute of the session id cookie. When this time period expires, the cookie is considered stale and is removed from the client and not sent out anymore.

Value Description Time period in second unit

Value Type int

Default Value -1

<i>Defined Value</i>	-1
	This value represents that it does not use "expires" property. Therefore, cookie will be terminated after terminating the session of user.

```
(396) <web-container> <context-group> <response-header>
<session-id-cookie> <secure>
```

<i>Description</i>	"secure" property of Session id cookie is set. If this element has been set as "true", session id cookie is sent to secure HTTPS connection.
--------------------	--

<i>Value Description</i>	Boolean value to determine wheter cookie is shown above only HTTPS.
--------------------------	---

<i>Value Type</i>	boolean
-------------------	---------

<i>Default Value</i>	false
----------------------	-------

<i>Defined Value</i>	true
	session id cookie is sent to only secure HTTPS connection.
	false
	Session id cookie will be sent despite of connection type .

```
(397) <web-container> <context-group> <response-header> <custom-
header>
```

<i>Description</i>	This element defines custom field for containing HTTP response message.
--------------------	---

<i>Child Elements</i>	(398)header-field*
-----------------------	--------------------

```
(398) <web-container> <context-group> <response-header> <custom-
header> <header-field>
```

<i>Description</i>	A field name and a value to include in all HTTP responses of this context group.
--------------------	--

Child Elements

- (399)field-name
- (400)field-value

```
(399) <web-container> <context-group> <response-header> <custom-
header> <header-field> <b>field-name</b>
```

Description HTTP Responding Field Name.

Value Description Field name.

Value Type token

```
(400) <web-container> <context-group> <response-header> <custom-
header> <header-field> <b>field-value</b>
```

Description Value used for the field.

Value Description Value.

Value Type token

```
(401) <web-container> <b>db-connection-pool</b>
```

Description For each instance of this element that is added, a so called "database connection pool" will be created and configured to run in the Web container. These database connection pools increase the performance of database access by keeping a pool of open database connections that the clients (in this case Servlets) can request directly. Each configured database connection pool will connect to one database only, so to use several databases, multiple connection pool XML elements must be created. Also note that all connection pools configured for a specific Web container are shared between all contexts and context groups within that container.

Child Elements

- (402)pool-name
- (403)pool-type?
- (404)pooling-rule?
- (405)connection-url
- (406)driver-class-name

```
(407)connection-argument
(408)listener-id*
(409)check-query?
(410)dynamic-increment?
(411)get-connection-timeout?
(412)read-timeout?
(413)close-long-active-connection?
(414)max-active-time?
(415)max-use-count?
(416)login-delay?
(417)close-delay?
(418)fetch-size?
(419)db-pool-control?
(425)dba-connection?
(429)encryption?
```

```
(402) <web-container> <db-connection-pool> <pool-name>
```

Description The name/ID of the DB connection pool. This name is used in Servlet code to access the pool.

Value Description Name

Value Type token

```
(403) <web-container> <db-connection-pool> <pool-type>
```

Description Type of DB Connection Pool.

Value Description One of four values mentioned below

Value Type pool-typeType

Default Value shared

Defined Value shared

Each client shares the entire connections in the Pool. It is required to define <db-pool-control>element and its child element

non-shared

Distributing each client to DB connection. These connections will not be returned from Pool after being used. This means an actual "pooling" function is not used and <listener-id> element should be set.

shared:non-jeus

This is used for migration. Value for child element should be set with <db-pool-control> element

non-shared:non-jeus

This is used for migration. <listener-id> element should be set.

(404) <web-container> <db-connection-pool> **<pooling-rule>**

<i>Description</i>	Determines the algorithm for selecting and assigning a new connection from the pool.
<i>Value Description</i>	Value which is either of queue or stack.
<i>Value Type</i>	pooling-ruleType
<i>Default Value</i>	stack
<i>Defined Value</i>	queue Queue stack Stack

(405) <web-container> <db-connection-pool> **<connection-url>**

<i>Description</i>	The connection URL to the database to use for this connection pool.
<i>Value</i>	A database connection URL
<i>Description</i>	
<i>Value Type</i>	token

Example `<connection-
url>jdbc:oracle:thin:@111.111.111.111:1521:ORA92</connection-
url>`

(406) `<web-container> <db-connection-pool> <driver-class-name>`

Description This attribute indicates the class name of the actual driver for pooling.

Value Description A fully qualified class name.

Value Type token

Example `<driver-class-
name>oracle.jdbc.driver.OracleDriver</driver-
class-name>`

(407) `<web-container> <db-connection-pool> <connection-argument>`

Description This attribute specifies the properties that will be used for establishing a DB connection.

Value Description : A string with connection arguments using the format "name1=value1;name2=value2" etc.

Value Type token

Example `<connection-
argument>user=scott;password=tiger</connection-
argument>`

(408) `<web-container> <db-connection-pool> <listener-id>`

Description When non-sharing mode is selected, Web Server listener should be set for obtaining a certain connection. This access control list is expressed as one or more elements. The value should be identical with WebServer listener ID located below webserver-connection element.

Value Description The value should be identical with listener ID(listener-id element) located below webserver-connection element

Value Type token

Example <listener-id>WebListener1</listener-id>

```
(409) <web-container> <db-connection-pool> <check-query>
```

Description This element configures SQL query checking DB Connection to be used by Web Container. It may not operate in normal in DBMS server even if internally defined SQL statement is used as default. For such case, basic query statment is defined in this element. This element should not be used when it is not necessary.

Value Description Simple SQL query used for checking alive of DB Connection to be used by Web Container.

Value Type token

Example <check-query>select date from dual</check-query>

```
(410) <web-container> <db-connection-pool> <dynamic-increment>
```

Description This configuration dynamically increases the number of connection pools despite of the maximum number of them to be used. Ex) A client sends the request to get DB connection of this Pool and the maximum number of pools are generated. It will generate the new connection when setting it as "true" in case of being used. It will return the connection to following client. This is used only it is set as "shared".

Value Description Activating or inactivating this configuration as Boolean value.

Value Type boolean

Default Value false

Example <dynamic-increment>true</dynamic-increment>

```
(411) <web-container> <db-connection-pool> <get-connection-  
timeout>
```

<i>Description</i>	If <dynamic-increment> element is inactivated, this element enables to set maximum time to wait the connection enabled after requesting new connection It will return "null" value if connection to be used does not exist, it will return the value of "null"
<i>Value Description</i>	A time interval with the unit of milliseconds (the time period will be rounded to the nearest second).
<i>Value Type</i>	int
<i>Default Value</i>	10000
<i>Example</i>	<get-connection-timeout>30000</get-connection-timeout>

```
(412) <web-container> <db-connection-pool> <read-timeout>
```

<i>Description</i>	Maximum waiting time for reading the data.
<i>Value Type</i>	nonNegativeLongType
<i>Value Type Description</i>	This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.
<i>Default Value</i>	0

```
(413) <web-container> <db-connection-pool> <close-long-active-connection>
```

<i>Description</i>	If the value of this element is set as "true" and DB connection extracted from this Pool is not returned during the time period set in <max-active-time>element, this DB connection will be returned as Pool after being terminated through being considered as "time out" This sometimes occurs when monitoring thread is set in <monitoring><check-db-connection-pool> and activated.
<i>Value Description</i>	Activating or inactivating this configuration as boolean value.
<i>Value Type</i>	boolean
<i>Value Type Description</i>	Boolean

<i>Default Value</i>	false
<i>Defined Value</i>	true Activating this configuration
	false Inactivating this configuration

(414) <web-container> <db-connection-pool> **<max-active-time>**

<i>Description</i>	Time interval returning DB connection Pool when DB connection is in time-out . This element is used only when <close-long-active-connection>element is activated.
<i>Value Description</i>	A time interval with the unit of milliseconds (the time period will be rounded to the nearest second).
<i>Value Type</i>	int
<i>Value Type Description</i>	Long
<i>Default Value</i>	-1
<i>Defined Value</i>	-1 This means that connections are not returned as pool.

(415) <web-container> <db-connection-pool> **<max-use-count>**

<i>Description</i>	Determines the maximum number of usage in a physical DB connection used from Client. It is recommended to set it as default since time is wasted for regenerating it.
<i>Value Description</i>	Maximum number of using one DB connection in the Pool.
<i>Value Type</i>	int
<i>Value Type Description</i>	Integer
<i>Default Value</i>	-1

<i>Defined Value</i>	-1
	Doesn't regenerate the DB connection in the Pool.

(416) <web-container> <db-connection-pool> **<login-delay>**

<i>Description</i>	This element sets the waiting time for enabling each initial connection to be waited for the interval when DB connection pool is initialized. This is not commonly used.
--------------------	--

<i>Value Description</i>	Time period expressed with Millisecond
--------------------------	--

<i>Value Type</i>	int
-------------------	-----

<i>Value Type Description</i>	Long
-------------------------------	------

<i>Default Value</i>	-1
----------------------	----

<i>Defined Value</i>	-1
	Does not maintain connection.

(417) <web-container> <db-connection-pool> **<close-delay>**

<i>Description</i>	If DB connection pool is being terminated, the termination time for connection will be delayed for the set amount of time. This value provides the chance for clients to "wait period" to efficiently close DB connection in which they use. When connection still exists even for the cycle, it will forcibly be closed and terminated. This element is not commonly used.
--------------------	---

<i>Value Description</i>	Time Interval in Millisecond.
--------------------------	-------------------------------

<i>Value Type</i>	int
-------------------	-----

<i>Value Type Description</i>	Long
-------------------------------	------

<i>Default Value</i>	-1
----------------------	----

<i>Defined Value</i>	-1
	Terminating enter connections to be used.

(418) <web-container> <db-connection-pool> <fetch-size>

<i>Description</i>	The number obtaining row in one ResultSet. This value ignores the value of Statement.setFetchSize(int row) which is the method of java.sql.package.
<i>Value Description</i>	The number of Row
<i>Value Type</i>	int
<i>Value Type Description</i>	Integer
<i>Default Value</i>	-1
<i>Defined Value</i>	-1 Doesn't provide the fetch size.

(419) <web-container> <db-connection-pool> <db-pool-control>

<i>Description</i>	This element configures the DB connection pool size and either of "shared" or "shared:non-jeus" value will be used for <pool-type> element value.
<i>Child Elements</i>	(420)min? (421)max (422)step? (423)max-idle-time? (424)max-alive-time?

(420) <web-container> <db-connection-pool> <db-pool-control>
<min>

<i>Value Description</i>	A number of connections
<i>Value Type</i>	nonNegativeLongType
<i>Value Type Description</i>	This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.
<i>Default Value</i>	1

```
(421) <web-container> <db-connection-pool> <db-pool-control>
<max>
```

<i>Description</i>	Maximum number of pools used for maintaining DB connection pool.
<i>Value Description</i>	The number of connection.
<i>Value Type</i>	nonNegativeLongType
<i>Value Type Description</i>	This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.

```
(422) <web-container> <db-connection-pool> <db-pool-control>
<step>
```

<i>Description</i>	The Number of simultaneous connection whenever there is a need for new conenction to be added in Pool
<i>Value Description</i>	The number of connection to be added in the Pool
<i>Value Type</i>	nonNegativeLongType
<i>Value Type Description</i>	This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.
<i>Default Value</i>	1

```
(423) <web-container> <db-connection-pool> <db-pool-control>
<max-idle-time>
```

<i>Description</i>	Sets the "time out" period of unused connections in the pool. When a connection has been idle for this amount of time and if the number of connections in the pool is more than min it is disconnected from the database and removed from the pool. The idle time checking is performed at a specified time interval by the DB connection pool monitoring thread defined below the <monitoring> element.
<i>Value Description</i>	A number of milliseconds.

<i>Value Type</i>	int
<i>Value Type Description</i>	Integer
<i>Default Value</i>	-1
<i>Defined Value</i>	-1 No "time out" check is performed and all idle connections stay in the pool indefinitely.

```
(424) <web-container> <db-connection-pool> <db-pool-control>
<max-alive-time>
```

<i>Description</i>	Sets the expire time on connection in the Pool. When a connection is given for the client thread, it is determined whether the connection is initially generated and last for set amount of time by having alive status.
<i>Value Description</i>	Value in Millisecond.
<i>Value Type</i>	int
<i>Value Type Description</i>	Integer
<i>Default Value</i>	-1
<i>Defined Value</i>	-1 All the connections used are maintained without inspecting on being removed.

```
(425) <web-container> <db-connection-pool> <dba-connection>
```

<i>Description</i>	DBA connection is used for obtaining administrator level connection from DB. If this certain DBAconnection is set, it will be used for solving various problems through accessing to DB connection pool of Web Container. (Closing Authority on DBA when there is connection having problem). This element implies that <close-long-active-connection>element should be activated.
--------------------	--

Child Elements

- (426) forced-close?
- (427) connection-url
- (428) connection-argument

```
( 426 ) <web-container> <db-connection-pool> <dba-connection>
<forced-close>
```

Description If this element is set as "true", it will exeucte "kill" operation with performing identical command of DBA low-level without calling "close()" method.

Value Description Activating or inactivating this setting as Boolean value.

Value Type boolean

Value Type Description Boolean

Default Value false

```
( 427 ) <web-container> <db-connection-pool> <dba-connection>
<connection-url>
```

Description Connection URL for connecting to certain DB

Value Connection URL of one DB

Description

Value Type token

Example <connection-
url>jdbc:oracle:thin@111.111.111.111:1521:ORA805</connection-
url>

```
( 428 ) <web-container> <db-connection-pool> <dba-connection>
<connection-argument>
```

Description Connection Parameter including user name and password of certain user.

Value String on Connection Parameter is seperated with using ";" having

Description name=vlaue

Value Type token

Example <connection-
argument>user=system/password=manager</connection-
argument>

(429) <web-container> <db-connection-pool> **<encryption>**

Description This is the element determines whether to encode the given password in <connection-argument>element.(Encoding the password using encode utility located in JEUS_HOME\bin.

Value Description Activating or inactivating this configuration as Boolean value.

Value Type boolean

Value Type Description Boolean

Default Value false

Defined Value true
Encoding password.

false
Not to encode password.

(430) <web-container> **<session-cluster>**

Description This element is used to configure the management and routing of client session objects across several clustered Web containers. Three techniques are possible: 1) routing, 2) use of a session server and 3) both methods 1 and 2 combined. This element will configure session management for all context groups and their sub-contexts (i.e. a container wide configuration). Any session cluster settings inside the context groups will be overridden by the settings in this element.

Child Elements (431)session-routing?
(432)shared?

(433)session-server?

```
(431) <web-container> <session-cluster> <session-routing>
```

Description

Session routing is a technique to manage sessions across clustered Web containers. This essentially means that special information, a Web container "ID number", will be attached to each outgoing cookie (session object). Using this ID, Web servers may later redirect (or "route") a client request back to the same Web container that issued the ID (that Web container also holds the actual HttpSession object associated with the ID). This technique is rather efficient in terms of performance but is not completely reliable since a failure in the Web container while processing a client request might cause the session object data to be lost. For better reliability it should be combined with the use of a session server as described in the `<session-server>` element. Session routing is used when either WebtoB or Apache Web servers are connected to multiple Web containers. WebtoB handles session routing without any special configurations while Apache will require some setup before it can be used with this feature.

<i>Value Description</i>	Activating or inactivating this configuration as Boolean value.
--------------------------	---

<i>Value Type</i>	boolean
-------------------	---------

<i>Value Type Description</i>	Boolean
-------------------------------	---------

<i>Default Value</i>	true
----------------------	------

<i>Defined Value</i>	true	Activating session sharing between Cluster Web Containers.
	false	Inactivating session sharing between Cluster Web Containers.

```
(432) <web-container> <session-cluster> <shared>
```

Description When session routing is set as false and session server is used, all the context group and contexts in it can share the session. However, session can be shared among groups if session server is not used or session routing is true. For such environment, this option should be set as true for maintaining session among context groups.

Value Type boolean

Default Value false

```
(433) <web-container> <session-cluster> <session-server>
```

Description This element enables and configures the use of a session server for session sharing and backup functionality. The session server is configured for use with this context group but it will not run in the Web container but rather in a JEUS server. Session servers are therefore configured in the file JEUSMain.xml. This element will merely configure the connection to such a session server.<p> The use of a session server essentially means that all session data will be "backed up" in the session server so that it can be recreated at a later time if the Web container experiences a failure and the data is lost. It also enables other Web containers than the one that originally created the session to fetch the session object, which is good if one Web container is downed and another container has to take over its session information.<p>For even better reliability, each Web container may also be configured to use a backup session server in case the main server is downed.

Child Elements

- (434) server-name
- (435) max
- (436) min?
- (437) step?
- (438) backup-server-name?
- (439) connect-timeout?
- (440) read-timeout?

```
(434) <web-container> <session-cluster> <session-server>
```

<server-name>

<i>Description</i>	This element configures Session Server name(JNDI export name) for using Session Server in this Context group This export name sets the value in <session-manager><name> in JEUSMain.xml
<i>Value Description</i>	JNDI export name distinguishes Primary Session Server to be used in this Context group.
<i>Value Type</i>	token

(435) <web-container> <session-cluster> <session-server> <max>

<i>Description</i>	This element is used for configuring maximum number of connection pools with Session Server for maintaining them
<i>Value Description</i>	Maximum number used for maintaining connection pool.
<i>Value Type</i>	nonNegativeLongType
<i>Value Type Description</i>	This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.

(436) <web-container> <session-cluster> <session-server> <min>

<i>Value Description</i>	A minimum number of connections to maintain.
<i>Value Type</i>	nonNegativeLongType
<i>Value Type Description</i>	This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.

(437) <web-container> <session-cluster> <session-server> <step>

<i>Description</i>	This element sets the number of connections to add connection pool with Session Server when it is needed to generate a new connection.
<i>Value Type</i>	nonNegativeLongType
<i>Value Type Description</i>	This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.

Default Value 1

```
(438) <web-container> <session-cluster> <session-server>
<backup-server-name>
```

Description The name of the backup session server to use if the main session server is unreachable.

Value Description JNDI export name of Session Server to be used as Backup Session Server.

Value Type token

```
(439) <web-container> <session-cluster> <session-server>
<connect-timeout>
```

Description Maximum time taken for waiting for establishing a new connection with Session Server when connection is initially generated.

Value Type nonNegativeLongType

Value Type Description This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.

Default Value 5000

```
(440) <web-container> <session-cluster> <session-server> <read-
timeout>
```

Description Maximum number of time taken for reading the data from Session Server.

Value Type nonNegativeLongType

Value Type Description This element is long type which is over 0. Thus, it includes values which are over 0 in the range of long type.

Default Value 20000

```
(441) <web-container> <security-switch>
```

<i>Description</i>	Activating or inactivating entire Security (authentication) check If this option is set as "false", structure of security will be ignored. This element is used for test and development.
--------------------	---

<i>Value Type</i>	boolean
-------------------	---------

<i>Default Value</i>	true
----------------------	------

```
(442) <web-container> <logging>
```

<i>Child Elements</i>	(443)user-log? (496)access-log?
-----------------------	------------------------------------

```
(443) <web-container> <logging> <user-log>
```

<i>Description</i>	This element sets the defined message handling through ServletContext.log() method.
--------------------	---

<i>Child Elements</i>	(444)level? (445)use-parent-handlers? (446)filter-class? (447)handler?
-----------------------	---

```
(444) <web-container> <logging> <user-log> <level>
```

<i>Description</i>	Level of logging is set for this field. Please refer to Level class of logging API of J2SE.
--------------------	---

<i>Value Type</i>	loggingLevelType
-------------------	------------------

<i>Default Value</i>	INFO
----------------------	------

<i>Defined Value</i>	FATAL
----------------------	-------

NOTICE

INFORMATION

DEBUG

fatal

notice

information

debug

SEVERE

WARNING

INFO

CONFIG

FINE

FINER

FINEST

ALL

severe

warning

info

config

fine

finer

finest

all

FATAL

NOTICE

INFORMATION

DEBUG

fatal

notice

information

debug

SEVERE

WARNING

INFO

CONFIG

FINE

FINER

FINEST

ALL

severe

warning

info

config

fine

finer

finest

all

```
(445) <web-container> <logging> <user-log> <use-parent-handlers>
```

Description This determines whether handlers of parent logger will be used for this logger.

Value Type boolean

Default Value false

```
(446) <web-container> <logging> <user-log> <filter-class>
```

Description Fully qualified class name of filter class set in this logger is set for this field.

Value Type token

Example `<filter-class>com.tmax.logging.filter.MyFilter</filter-class>`

```
(447) <web-container> <logging> <user-log> <handler>
```

Description Handler to use in logger is set for this field.

Child Elements (448)console-handler
 (453)file-handler
 (463)smtp-handler
 (474)socket-handler
 (481)user-handler

```
(448) <web-container> <logging> <user-log> <handler> <console-  
handler>
```

Description This handler will be used when logging is left in the screen.

Child Elements (449)name
 (450)level?
 (451)encoding?
 (452)filter-class?

```
(449) <web-container> <logging> <user-log> <handler> <console-  
handler> <name>
```

Description The name of handler will be set. At this moment, this name should be set as unique in a logger. This name is used for refering handler including tool.

Value Type token

Example <name>handler1</name>

```
(450) <web-container> <logging> <user-log> <handler> <console-  
handler> <level>
```

Description The level of this handler will be set for this field. In case where level of message going through logger is included in this handler level, the message will be displayed through this handler.

Value Type loggingLevelType

Default Value FINEST

Defined Value

FATAL

NOTICE

INFORMATION

DEBUG

fatal

notice

information

debug

SEVERE

WARNING

INFO

CONFIG

FINE

FINER

FINEST

ALL

severe

warning

info

config

fine

finer

finest

all

FATAL

NOTICE

INFORMATION

DEBUG

fatal

notice

information

debug

SEVERE

WARNING

INFO

CONFIG

FINE

FINER

FINEST

ALL

severe

warning

info

config

fine

finer

finest

all

```
(451) <web-container> <logging> <user-log> <handler> <console-  
handler> <encoding>
```

Description Encoding in which this handler leaves message is set for this field.

Value Type token

```
(452) <web-container> <logging> <user-log> <handler> <console-  
handler> <filter-class>
```

Description Fully qualified class name of filter class set in this handler is set for

this field.

Value Type token

Example <filter-
 class>com.tmax.logging.filter.MyFilter</filter-
 class>

```
(453) <web-container> <logging> <user-log> <handler> <file-  

handler>
```

Description This is the handler to be used in case of displaying logging as file.

Child Elements (454)name
 (455)level?
 (456)encoding?
 (457)filter-class?
 (458)file-name?
 (459)valid-day
 (460)valid-hour
 (461)buffer-size?
 (462)append?

```
(454) <web-container> <logging> <user-log> <handler> <file-  

handler> <name>
```

Description The name of handler will be set. At this moment, this name should be set as unique in a logger. This name is used for refering handler including tool.

Value Type token

Example <name>handler1</name>

```
(455) <web-container> <logging> <user-log> <handler> <file-  

handler> <level>
```

Description The level of this handler will be set for this field. In case where level of message going through logger is included in this handler level, the message will be displayed through this handler.

<i>Value Type</i>	loggingLevelType
-------------------	------------------

<i>Default Value</i>	FINEST
----------------------	--------

<i>Defined Value</i>	FATAL
----------------------	-------

	NOTICE
--	--------

	INFORMATION
--	-------------

	DEBUG
--	-------

	fatal
--	-------

	notice
--	--------

	information
--	-------------

	debug
--	-------

	SEVERE
--	--------

	WARNING
--	---------

	INFO
--	------

	CONFIG
--	--------

	FINE
--	------

	FINER
--	-------

	FINEST
--	--------

	ALL
--	-----

severe

warning

info

config

fine

finer

finest

all

FATAL

NOTICE

INFORMATION

DEBUG

fatal

notice

information

debug

SEVERE

WARNING

INFO

CONFIG

FINE

FINER

FINEST

ALL

severe

warning

info

config

fine

finer

finest

all

```
(456) <web-container> <logging> <user-log> <handler> <file-  
handler> <encoding>
```

Description Encoding in which this handler leaves message is set for this field.

Value Type token

```
(457) <web-container> <logging> <user-log> <handler> <file-  
handler> <filter-class>
```

<i>Description</i>	Fully qualified class name of filter class set in this handler is set for this field.
<i>Value Type</i>	token
<i>Example</i>	<filter- class>com.tmax.logging.filter.MyFilter</filter- class>

```
(458) <web-container> <logging> <user-log> <handler> <file-  
handler> <file-name>
```

<i>Description</i>	File name in which this handler uses will be set for this field. If user has not set for this field, default file name of each logger will be used. Please refer to JEUS Server Manual for each default file name.
<i>Value Type</i>	token
<i>Example</i>	<file-name>C:\logs\mylog.log</file-name>

```
(459) <web-container> <logging> <user-log> <handler> <file-  
handler> <valid-day>
```

<i>Description</i>	File in which this handler uses will be used and updated frequently for certain set amount of time . This setting is used for changing file as unit of day. In this case, used date will be followed right after file name.
<i>Value Description</i>	day
<i>Value Type</i>	off-intType
<i>Value Type Description</i>	The default value is in non-negative int type. In case where the value is -1, it is considered as off.
<i>Example</i>	<valid-day>1</valid-day>

```
(460) <web-container> <logging> <user-log> <handler> <file-  
handler> <valid-hour>
```

Description In case of using and updating frequently on the file in which this handler uses for a certain amount of time. This setting is used for updating time unit. In this case, used date and time will be automatically followed right after file name in which this handler uses.

Value Description This represents in unit of hour and it should get divisor of 24 + N*24(n is integer greater than 0).

Value Type off-intType

Value Type Description The default value is in non-negative int type. In case where the value is -1, it is considered as off.

Example <valid-hour>3</valid-hour>

```
(461) <web-container> <logging> <user-log> <handler> <file-  
handler> <buffer-size>
```

Description The size of buffer is set when displaying of this handler as file.

Value Description The unit in byte. [Performance Recommendation]: The logging performance will be improved as this value is bigger even though message will be displayed with delay.

Value Type nonNegativeIntType

Value Type Description This element is int type which is over 0. Thus, it includes values which are over 0 in the range of int type.

Default Value 1024

```
(462) <web-container> <logging> <user-log> <handler> <file-  
handler> <append>
```

Description In case where file in which this handler uses already exists, this element will be used for determining whether to be written after the file name. If it has been set as false, existing file will be

deleted.

Value Type boolean

Default Value true

```
(463) <web-container> <logging> <user-log> <handler> <smtp-  
handler>
```

Description This is the handler to use when sending logging to email.
[Performance Recommendation]: There is a warning when a huge amount of email exist without proper filter.

Child Elements (464)name
 (465)level?
 (466)encoding?
 (467)filter-class?
 (468)smtp-host-address
 (469)from-address
 (470)to-address
 (471)cc-address?
 (472)bcc-address?
 (473)send-for-all-messages?

```
(464) <web-container> <logging> <user-log> <handler> <smtp-  
handler> <name>
```

Description The name of handler will be set. At this moment, this name should be set as unique in a logger. This name is used for refering handler including tool.

Value Type token

Example <name>handler1</name>

```
(465) <web-container> <logging> <user-log> <handler> <smtp-  
handler> <level>
```

Description The level of this handler will be set for this field. In case where level of message going through logger is included in this handler

level, the message will be displayed through this handler.

Value Type loggingLevelType

Default Value FINEST

Defined Value FATAL

NOTICE

INFORMATION

DEBUG

fatal

notice

information

debug

SEVERE

WARNING

INFO

CONFIG

FINE

FINER

FINEST

ALL

severe

warning

info

config

fine

finer

finest

all

FATAL

NOTICE

INFORMATION

DEBUG

fatal

notice

information

debug

SEVERE

WARNING

INFO

CONFIG

FINE

FINER

FINEST

ALL

severe

warning

info

config

fine

finer

finest

all

```
(466) <web-container> <logging> <user-log> <handler> <smtp-  
handler> <encoding>
```

Description Encoding in which this handler leaves message is set for this field.

Value Type token

```
(467) <web-container> <logging> <user-log> <handler> <smtp-  
handler> <filter-class>
```

Description Fully qualified class name of filter class set in this handler is set for this field.

Value Type token

Example <filter-
class>com.tmax.logging.filter.MyFilter</filter-
class>

```
(468) <web-container> <logging> <user-log> <handler> <smtp-  
handler> <smtp-host-address>
```

Description The address of Smtip server to send email is set for this field.

Value Type token

```
(469) <web-container> <logging> <user-log> <handler> <smtp-  
handler> <from-address>
```

Description The address of sender for email is set for this field.

Value Type token

```
(470) <web-container> <logging> <user-log> <handler> <smtp-  
handler> <to-address>
```

Description The address of receiver of email is set for this field.

Value Type token

```
(471) <web-container> <logging> <user-log> <handler> <smtp-  
handler> <cc-address>
```

Description The cc-address of receiver is set for this field.

Value Type token

```
(472) <web-container> <logging> <user-log> <handler> <smtp-
```



```
handler> <bcc-address>
```

Description The bcc-address of receiver is set for this field.

Value Type token

```
(473) <web-container> <logging> <user-log> <handler> <smtp-  
handler> <send-for-all-messages>
```

Description When messages enter into this handler through using log() method of logger registered in this handler, it determines whether to consider it as the sender object of email. If it has been set as false, messages which are only called through using specific send() method are transmitted.

Value Type boolean

Default Value false

```
(474) <web-container> <logging> <user-log> <handler> <socket-  
handler>
```

Description This element is handler used for sending to socket in which logging is set. [Performance Recommendation]: The performance will be decreased without using proper filter since logging message is transmitted as one Socket.

Child Elements

- (475)name
- (476)level?
- (477)encoding?
- (478)filter-class?
- (479)address
- (480)port

```
(475) <web-container> <logging> <user-log> <handler> <socket-  
handler> <name>
```

Description The name of handler will be set. At this moment, this name should be set as unique in a logger. This name is used for refering handler including tool.

Value Type token

Example <name>handler1</name>

```
(476) <web-container> <logging> <user-log> <handler> <socket-  
handler> <level>
```

Description The level of this handler will be set for this field. In case where level of message going through logger is included in this handler level, the message will be displayed through this handler.

Value Type loggingLevelType

Default Value FINEST

Defined Value FATAL

NOTICE

INFORMATION

DEBUG

fatal

notice

information

debug

SEVERE

WARNING

INFO

CONFIG

FINE

FINER

FINEST

ALL

severe

warning

info

config

fine

finer

finest

all

FATAL

NOTICE

INFORMATION

DEBUG

fatal

notice

information

debug

SEVERE

WARNING

INFO

CONFIG

FINE

FINER

FINEST

ALL

severe

warning

info

config

fine

finer

finest

all

```
(477) <web-container> <logging> <user-log> <handler> <socket-  
handler> <encoding>
```

Description Encoding in which this handler leaves message is set for this field.

Value Type token

```
(478) <web-container> <logging> <user-log> <handler> <socket-  
handler> <filter-class>
```

Description Fully qualified class name of filter class set in this handler is set for this field.

Value Type token

Example <filter-
class>com.tmax.logging.filter.MyFilter</filter-
class>

```
(479) <web-container> <logging> <user-log> <handler> <socket-  
handler> <address>
```

Description IP address of sending messages when generating this handler.

Value Type token

```
(480) <web-container> <logging> <user-log> <handler> <socket-  
handler> <port>
```

Description Port of sending messages when generating this handler.

Value Type nonNegativeIntType

Value Type Description This element is int type which is over 0. Thus, it includes values which are over 0 in the range of int type.

```
(481) <web-container> <logging> <user-log> <handler> <user-  
handler>
```

Description In case of using handler in which user has created according to J2SE logging API, this setting

Child Elements (482)handler-class
 (483)name
 (484)level?
 (485)encoding?
 (486)filter-class?
 (487)handler-property?
 (491)formatter-class?
 (492)formatter-property?

```
(482) <web-container> <logging> <user-log> <handler> <user-  
handler> <handler-class>
```

Description Fully qualified class name in which user has created is set. This class inherits java.util.logging.Handler and jeus.util.logging.JeusHandler will be constructed.

Value Type token

Example <handler-
 class>com.tmax.logging.handler.MyHandler</handler-
 class>

```
(483) <web-container> <logging> <user-log> <handler> <user-  
handler> <name>
```

Description The name of handler will be set. At this moment, this name should be set as unique in a logger. This name is used for refering handler including tool.

Value Type token

Example <name>handler1</name>

```
(484) <web-container> <logging> <user-log> <handler> <user-  
handler> <level>
```

Description The level of this handler will be set for this field. In case where

level of message going through logger is included in this handler level, the message will be displayed through this handler.

Value Type loggingLevelType

Default Value FINEST

Defined Value FATAL

NOTICE

INFORMATION

DEBUG

fatal

notice

information

debug

SEVERE

WARNING

INFO

CONFIG

FINE

FINER

FINEST

ALL

severe

warning

info

config

fine

finer

finest

all

FATAL

NOTICE

INFORMATION

DEBUG

fatal

notice

information

debug

SEVERE

WARNING

INFO

CONFIG

FINE

FINER

FINEST

ALL

severe

warning

info

config

fine

finer

finest

all

```
(485) <web-container> <logging> <user-log> <handler> <user-  
handler> <encoding>
```

Description

Encoding in which this handler leaves message is set for this field.

Value Type token

```
(486) <web-container> <logging> <user-log> <handler> <user-
handler> <filter-class>
```

Description Fully qualified class name of filter class set in this handler is set for this field.

Value Type token

Example <filter-
class>com.tmax.logging.filter.MyFilter</filter-
class>

```
(487) <web-container> <logging> <user-log> <handler> <user-
handler> <handler-property>
```

Description When handler is generated, property to parse is set. These properties are delivered to handler as being stored in Map object as key-value through JeusHandler.setProperty() method.

Child Elements (488)property*

```
(488) <web-container> <logging> <user-log> <handler> <user-
handler> <handler-property> <property>
```

Description Property to be delivered to handler is set.

Child Elements (489)key
 (490)value

```
(489) <web-container> <logging> <user-log> <handler> <user-
handler> <handler-property> <property> <key>
```

Description This element is the key value of property.

Value Type token

```
(490) <web-container> <logging> <user-log> <handler> <user-
handler> <handler-property> <property> <value>
```

Description The value of property.

Value Type token

```
(491) <web-container> <logging> <user-log> <handler> <user-  
handler> <formatter-class>
```

Description Fully qualified class name of formatter in which this handler is used. This class inherits java.util.logging.Formatter and jeus.util.logging.JeusFormatter will be constructed.

Value Type token

Default Value jeus.util.logging.SimpleFormatter

Example <formatter-
class>com.tmax.logging.handler.MyHandler</formatter-
class>

```
(492) <web-container> <logging> <user-log> <handler> <user-  
handler> <formatter-property>
```

Description Property to parse to formatter when handler is generated. This properties are delivered to formatter through using JeusFormatter.setProperty() method stored in Map object.

Child Elements (493)property*

```
(493) <web-container> <logging> <user-log> <handler> <user-  
handler> <formatter-property> <property>
```

Description Property to be delivered to handler is set.

Child Elements (494)key
(495)value

```
(494) <web-container> <logging> <user-log> <handler> <user-  
handler> <formatter-property> <property> <key>
```

Description This element is the key value of property.

Value Type token

```
(495) <web-container> <logging> <user-log> <handler> <user-  
handler> <formatter-property> <property> <value>
```

Description The value of property.

Value Type token

```
(496) <web-container> <logging> <access-log>
```

Description This element configures on handling log of client of Container.

Child Elements (497)enable?
 (498)enable-analysis?
 (499)level?
 (500)use-parent-handlers?
 (501)filter-class?
 (502)handler?
 (551)format?

```
(497) <web-container> <logging> <access-log> <enable>
```

Description Setting for the availabilitiy of analyzing an access logging.

Value Type boolean

Default Value true

```
(498) <web-container> <logging> <access-log> <enable-analysis>
```

Description Setting for the availability of analyzing an access logging.

Value Type boolean

Default Value false

```
(499) <web-container> <logging> <access-log> <level>
```

Description Level of logging is set for this field. Please refer to Level class of logging API of J2SE.

<i>Value Type</i>	loggingLevelType
-------------------	------------------

<i>Default Value</i>	INFO
----------------------	------

<i>Defined Value</i>	FATAL
----------------------	-------

	NOTICE
--	--------

	INFORMATION
--	-------------

	DEBUG
--	-------

	fatal
--	-------

	notice
--	--------

	information
--	-------------

	debug
--	-------

	SEVERE
--	--------

	WARNING
--	---------

	INFO
--	------

	CONFIG
--	--------

	FINE
--	------

	FINER
--	-------

	FINEST
--	--------

	ALL
--	-----

severe

warning

info

config

fine

finer

finest

all

FATAL

NOTICE

INFORMATION

DEBUG

fatal

notice

information

debug

SEVERE

WARNING

INFO

CONFIG

FINE

FINER

FINEST

ALL

severe

warning

info

config

fine

finer

finest

all

```
(500) <web-container> <logging> <access-log> <use-parent-handlers>
```

Description This determines whether handlers of parent logger will be used for this logger.

Value Type boolean

Default Value false

```
(501) <web-container> <logging> <access-log> <filter-class>
```

Description Fully qualified class name of filter class set in this logger is set for this field.

Value Type token

Example <filter-
class>com.tmax.logging.filter.MyFilter</filter-
class>

```
(502) <web-container> <logging> <access-log> <handler>
```

Description Handler to use in logger is set for this field.

Child Elements (503)console-handler
(508)file-handler
(518)smtp-handler
(529)socket-handler
(536)user-handler

```
(503) <web-container> <logging> <access-log> <handler> <console-  
handler>
```

Description This handler will be used when logging is left in the screen.

Child Elements (504)name
(505)level?
(506)encoding?
(507)filter-class?

```
(504) <web-container> <logging> <access-log> <handler> <console-  
handler> <name>
```

Description The name of handler will be set. At this moment, this name should be set as unique in a logger. This name is used for refering handler including tool.

Value Type token

Example `<name>handler1</name>`

```
(505) <web-container> <logging> <access-log> <handler> <console-  
handler> <level>
```

Description The level of this handler will be set for this field. In case where level of message going through logger is included in this handler level, the message will be displayed through this handler.

Value Type loggingLevelType

Default Value FINEST

Defined Value FATAL

NOTICE

INFORMATION

DEBUG

fatal

notice

information

debug

SEVERE

WARNING

INFO

CONFIG

FINE

FINER

FINEST

ALL

severe

warning

info

config

fine

finer

finest

all

FATAL

NOTICE

INFORMATION

DEBUG

fatal

notice

information

debug

SEVERE

WARNING

INFO

CONFIG

FINE

FINER

FINEST

ALL

severe

warning

info

config

fine

finer

finest

all

```
(506) <web-container> <logging> <access-log> <handler> <console-  
handler> <encoding>
```

Description Encoding in which this handler leaves message is set for this field.

Value Type token

```
(507) <web-container> <logging> <access-log> <handler> <console-  
handler> <filter-class>
```

Description Fully qualified class name of filter class set in this handler is set for this field.

Value Type token

Example <filter-
class>com.tmax.logging.filter.MyFilter</filter-
class>

```
(508) <web-container> <logging> <access-log> <handler> <file-  
handler>
```

Description This is the handler to be used in case of displaying logging as file.

Child Elements (509)name
 (510)level?
 (511)encoding?
 (512)filter-class?
 (513)file-name?
 (514)valid-day
 (515)valid-hour
 (516)buffer-size?
 (517)append?

```
(509) <web-container> <logging> <access-log> <handler> <file-  
handler> <name>
```

Description The name of handler will be set. At this moment, this name should

be set as unique in a logger. This name is used for refering handler including tool.

Value Type token

Example <name>handler1</name>

```
(510) <web-container> <logging> <access-log> <handler> <file-  
handler> <level>
```

Description The level of this handler will be set for this field. In case where level of message going through logger is included in this handler level, the message will be displayed through this handler.

Value Type loggingLevelType

Default Value FINEST

Defined Value FATAL

NOTICE

INFORMATION

DEBUG

fatal

notice

information

debug

SEVERE

WARNING

INFO

CONFIG

FINE

FINER

FINEST

ALL

severe

warning

info

config

fine

finer

finest

all

FATAL

NOTICE

INFORMATION

DEBUG

fatal

notice

information

debug

SEVERE

WARNING

INFO

CONFIG

FINE

FINER

FINEST

ALL

severe

warning

info

config

fine

finer

finest

all

```
(511) <web-container> <logging> <access-log> <handler> <file-  
handler> <encoding>
```

Description Encoding in which this handler leaves message is set for this field.

Value Type token

```
(512) <web-container> <logging> <access-log> <handler> <file-  
handler> <filter-class>
```

Description Fully qualified class name of filter class set in this handler is set for this field.

Value Type token

Example <filter-
class>com.tmax.logging.filter.MyFilter</filter-
class>

```
(513) <web-container> <logging> <access-log> <handler> <file-  
handler> <file-name>
```

Description File name in which this handler uses will be set for this field. If user has not set for this field, default file name of each logger will be used. Please refer to JEUS Server Manual for each default file name.

Value Type token

Example <file-name>C:\logs\mylog.log</file-name>

```
(514) <web-container> <logging> <access-log> <handler> <file-  
handler> <valid-day>
```


<i>Description</i>	File in which this handler uses will be used and updated frequently for certain set amount of time . This setting is used for changing file as unit of day. In this case, used date will be followed right after file name.
<i>Value Description</i>	day
<i>Value Type</i>	off-intType
<i>Value Type Description</i>	The default value is in non-negative int type. In case where the value is -1, it is considered as off.
<i>Example</i>	<code><valid-day>1</valid-day></code>

```
(515) <web-container> <logging> <access-log> <handler> <file-  
handler> <valid-hour>
```

<i>Description</i>	In case of using and updating frequently on the file in which this handler uses for a certain amount of time. This setting is used for updating time unit. In this case, used date and time will be automatically followed right after file name in which this handler uses.
<i>Value Description</i>	This represents in unit of hour and it should get divisor of 24 + N*24(n is integer greater than 0).
<i>Value Type</i>	off-intType
<i>Value Type Description</i>	The default value is in non-negative int type. In case where the value is -1, it is considered as off.
<i>Example</i>	<code><valid-hour>3</valid-hour></code>

```
(516) <web-container> <logging> <access-log> <handler> <file-  
handler> <buffer-size>
```

<i>Description</i>	The size of buffer is set when displaying of this handler as file.
<i>Value Description</i>	The unit in byte. [Performance Recommendation]: The logging performance will be improved as this value is bigger even though message will be displayed with delay.

Value Type nonNegativeIntType

Value Type Description This element is int type which is over 0. Thus, it includes values which are over 0 in the range of int type.

Default Value 1024

```
(517) <web-container> <logging> <access-log> <handler> <file-
handler> <append>
```

Description In case where file in which this handler uses already exists, this element will be used for determining whether to be written after the file name. If it has been set as false, existing file will be deleted.

Value Type boolean

Default Value true

```
(518) <web-container> <logging> <access-log> <handler> <smtp-
handler>
```

Description This is the handler to use when sending logging to email.
[Performance Recommendation]: There is a warning when a huge amount of email exist without proper filter.

Child Elements (519)name
 (520)level?
 (521)encoding?
 (522)filter-class?
 (523)smtp-host-address
 (524)from-address
 (525)to-address
 (526)cc-address?
 (527)bcc-address?
 (528)send-for-all-messages?

```
(519) <web-container> <logging> <access-log> <handler> <smtp-
handler> <name>
```

Description The name of handler will be set. At this moment, this name should be set as unique in a logger. This name is used for refering handler including tool.

Value Type token

Example <name>handler1</name>

```
(520) <web-container> <logging> <access-log> <handler> <smtp-  
handler> <level>
```

Description The level of this handler will be set for this field. In case where level of message going through logger is included in this handler level, the message will be displayed through this handler.

Value Type loggingLevelType

Default Value FINEST

Defined Value

FATAL

NOTICE

INFORMATION

DEBUG

fatal

notice

information

debug

SEVERE

WARNING

INFO

CONFIG

FINE

FINER

FINEST

ALL

severe

warning

info

config

fine

finer

finest

all

FATAL

NOTICE

INFORMATION

DEBUG

fatal

notice

information

debug

SEVERE

WARNING

INFO

CONFIG

FINE

FINER

FINEST

ALL

severe

warning

info

config

fine

finer

finest

all

```
(521) <web-container> <logging> <access-log> <handler> <smtp-  
handler> <encoding>
```

Description Encoding in which this handler leaves message is set for this field.

Value Type token

```
(522) <web-container> <logging> <access-log> <handler> <smtp-  
handler> <filter-class>
```

Description Fully qualified class name of filter class set in this handler is set for this field.

Value Type token

Example <filter-
class>com.tmax.logging.filter.MyFilter</filter-
class>

```
(523) <web-container> <logging> <access-log> <handler> <smtp-  
handler> <smtp-host-address>
```

Description The address of Smtip server to send email is set for this field.

Value Type token

```
(524) <web-container> <logging> <access-log> <handler> <smtp-  
handler> <from-address>
```

Description The address of sender for email is set for this field.

Value Type token

```
(525) <web-container> <logging> <access-log> <handler> <smtp-  
handler> <to-address>
```

Description The address of receiver of email is set for this field.

Value Type token

```
(526) <web-container> <logging> <access-log> <handler> <smtp-  
handler> <cc-address>
```

Description The cc-address of receiver is set for this field.

Value Type token

```
(527) <web-container> <logging> <access-log> <handler> <smtp-  
handler> <bcc-address>
```

Description The bcc-address of receiver is set for this field.

Value Type token

```
(528) <web-container> <logging> <access-log> <handler> <smtp-  
handler> <send-for-all-messages>
```

Description When messages enter into this handler through using log() method of logger registered in this handler, it determines whether to consider it as the sender object of email. If it has been set as false, messages which are only called through using specific send() method are transmitted.

Value Type boolean

Default Value false

```
(529) <web-container> <logging> <access-log> <handler> <socket-  
handler>
```

Description This element is handler used for sending to socket in which logging is set. [Performance Recommendation]: The performance will be decreased without using proper filter since logging message is transmitted as one Socket.

Child Elements

- (530)name
- (531)level?
- (532)encoding?
- (533)filter-class?
- (534)address
- (535)port

```
(530) <web-container> <logging> <access-log> <handler> <socket-  
handler> <name>
```

Description The name of handler will be set. At this moment, this name should be set as unique in a logger. This name is used for refering handler including tool.

Value Type token

Example <name>handler1</name>

```
(531) <web-container> <logging> <access-log> <handler> <socket-  
handler> <level>
```

Description The level of this handler will be set for this field. In case where level of message going through logger is included in this handler level, the message will be displayed through this handler.

Value Type loggingLevelType

Default Value FINEST

Defined Value FATAL

NOTICE

INFORMATION

DEBUG

fatal

notice

information

debug

SEVERE

WARNING

INFO

CONFIG

FINE

FINER

FINEST

ALL

severe

warning

info

config

fine

finer

finest

all

FATAL

NOTICE

INFORMATION

DEBUG

fatal

notice

information

debug

SEVERE

WARNING

INFO

CONFIG

FINE

FINER

FINEST

ALL

severe

warning

info

config

fine

finer

finest

all

```
(532) <web-container> <logging> <access-log> <handler> <socket-  
handler> <encoding>
```

Description Encoding in which this handler leaves message is set for this field.

Value Type token

```
(533) <web-container> <logging> <access-log> <handler> <socket-  
handler> <filter-class>
```

Description Fully qualified class name of filter class set in this handler is set for this field.

Value Type token

Example

```
<filter-  
class>com.tmax.logging.filter.MyFilter</filter-  
class>
```

```
(534) <web-container> <logging> <access-log> <handler> <socket-  
handler> <address>
```

Description IP address of sending messages when generating this handler.

Value Type token

```
(535) <web-container> <logging> <access-log> <handler> <socket-  
handler> <port>
```

Description Port of sending messages when generating this handler.

Value Type nonNegativeIntType

Value Type Description This element is int type which is over 0. Thus, it includes values which are over 0 in the range of int type.

```
(536) <web-container> <logging> <access-log> <handler> <user-  
handler>
```

Description In case of using handler in which user has created according to J2SE logging API, this setting

Child Elements (537)handler-class
 (538)name
 (539)level?
 (540)encoding?
 (541)filter-class?
 (542)handler-property?
 (546)formatter-class?
 (547)formatter-property?

```
(537) <web-container> <logging> <access-log> <handler> <user-  
handler> <handler-class>
```

Description Fully qualified class name in which user has created is set. This class inherits java.util.logging.Handler and jeus.util.logging.JeusHandler will be constructed.

Value Type token

Example <handler-
 class>com.tmax.logging.handler.MyHandler</handler-

class>

```
(538) <web-container> <logging> <access-log> <handler> <user-  
handler> <name>
```

Description The name of handler will be set. At this moment, this name should be set as unique in a logger. This name is used for refering handler including tool.

Value Type token

Example <name>handler1</name>

```
(539) <web-container> <logging> <access-log> <handler> <user-  
handler> <level>
```

Description The level of this handler will be set for this field. In case where level of message going through logger is included in this handler level, the message will be displayed through this handler.

Value Type loggingLevelType

Default Value FINEST

Defined Value FATAL

NOTICE

INFORMATION

DEBUG

fatal

notice

information

debug

SEVERE

WARNING

INFO

CONFIG

FINE

FINER

FINEST

ALL

severe

warning

info

config

fine

finer

finest

all

FATAL

NOTICE

INFORMATION

DEBUG

fatal

notice

information

debug

SEVERE

WARNING

INFO

CONFIG

FINE

FINER

FINEST

ALL

severe

warning

info

config

fine

finer

finest

all

```
(540) <web-container> <logging> <access-log> <handler> <user-  
handler> <encoding>
```

Description Encoding in which this handler leaves message is set for this field.

Value Type token

```
(541) <web-container> <logging> <access-log> <handler> <user-  
handler> <filter-class>
```

Description Fully qualified class name of filter class set in this handler is set for this field.

Value Type token

Example <filter-
 class>com.tmax.logging.filter.MyFilter</filter-
 class>

```
(542) <web-container> <logging> <access-log> <handler> <user-  
handler> <handler-property>
```

Description When handler is generated, property to parse is set. These properties are delivered to handler as being stored in Map object as key-value through JeusHandler.setProperty() method.

Child Elements (543)property*


```
(543) <web-container> <logging> <access-log> <handler> <user-  
handler> <handler-property> <property>
```

Description Property to be delivered to handler is set.

Child Elements (544)key
(545)value

```
(544) <web-container> <logging> <access-log> <handler> <user-  
handler> <handler-property> <property> <key>
```

Description This element is the key value of property.

Value Type token

```
(545) <web-container> <logging> <access-log> <handler> <user-  
handler> <handler-property> <property> <value>
```

Description The value of property.

Value Type token

```
(546) <web-container> <logging> <access-log> <handler> <user-  
handler> <formatter-class>
```

Description Fully qualified class name of formatter in which this handler is used. This class inherits java.util.logging.Formatter and jeus.util.logging.JeusFormatter will be constructed.

Value Type token

Default Value jeus.util.logging.SimpleFormatter

Example <formatter-
class>com.tmax.logging.handler.MyHandler</formatter-
class>

```
(547) <web-container> <logging> <access-log> <handler> <user-  
handler> <formatter-property>
```

Description Property to parse to formatter when handler is generated. This

properties are delivered to formatter through using
JeusFormatter.setProperty() method stored in Map object.

Child Elements (548)property*

```
(548) <web-container> <logging> <access-log> <handler> <user-  
handler> <formatter-property> <property>
```

Description Property to be delivered to handler is set.

Child Elements (549)key
 (550)value

```
(549) <web-container> <logging> <access-log> <handler> <user-  
handler> <formatter-property> <property> <key>
```

Description This element is the key value of property.

Value Type token

```
(550) <web-container> <logging> <access-log> <handler> <user-  
handler> <formatter-property> <property> <value>
```

Description The value of property.

Value Type token

```
(551) <web-container> <logging> <access-log> <format>
```

Description Setting the format for time-stamp. This configuration is only used
for access log.

Value Type token

```
(552) <web-container> <properties>
```

Description Sets the property of container-wide.

Child Elements (553)property*

```
(553) <web-container> <properties> <property>
```

Child Elements (554)key
(555)value

```
(554) <web-container> <properties> <property> <key>
```

Description This element is the key value of property.

Value Type token

```
(555) <web-container> <properties> <property> <value>
```

Description The value of property.

Value Type token

D.4 WEBMain.xml 파일 예제

<<WEBMain.xml>>

```
<?xml version="1.0"?>
<web-container xmlns="http://www.tmaxsoft.com/xml/ns/jeus">
  <monitoring>
    <check-thread-pool>60000</check-thread-pool>
    <check-class-reload>60000</check-class-reload>
    <check-session>60000</check-session>
    <check-db-connection-pool>
      6000
    </check-db-connection-pool>
  </monitoring>
  <redirect-stdout>true</redirect-stdout>
  <redirect-stderr>true</redirect-stderr>
  <shutdown-timeout>10000</shutdown-timeout>
  <context-group>
    <group-name>MyGroup</group-name>
    <virtual-host>
      <virtual-host-name>Host1</virtual-host-name>
      <host-list>www1.foo.com</host-list>
    </virtual-host>
  </context-group>
</web-container>
```

```
<context-name>Examples</context-name>
<context-path>/examples</context-path>
</context>
</virtual-host>
<context>
  <context-name>Examples</context-name>
  <context-path>/examples</context-path>
</context>
<webserver-connection>
  <webtob-listener>
    <listener-id>WebListener1</listener-id>
    <port>9900</port>
    <output-buffer-size>16384</output-buffer-size>
    <thread-pool>
      <min>10</min>
      <max>10</max>
      <step>4</step>
      <max-idle-time>60000</max-idle-time>
      <max-wait-queue>2</max-wait-queue>
      <max-queue></max-queue>
      <thread-state-notify>
        <max-thread-active-time>
          150000
        </max-thread-active-time>
        <notify-threshold>
          100
        </notify-threshold>
        <restart-threshold>
          18
        </restart-threshold>
        <notifier-id>MyNotifier</notifier-id>
        <notify-subject>
          JEUS WEB CONTAINER THREAD STATE
          WARNING
        </notify-subject>
        <restart-subject>
          JEUS WEB CONTAINER RESTART WARNING
        </restart-subject>
      </thread-state-notify>
    </thread-pool>
  </webtob-listener>
</webserver-connection>
```

```
</thread-pool>
<postdata-read-timeout>
    40000
</postdata-read-timeout>
<scheme>http</scheme>
<hth-count>4</hth-count>
<request-queueing>true</request-queueing>
<disable-pipe>true</disable-pipe>
<webtob-address>
    111.111.111.111
</webtob-address>
<registration-id>MyGroup</registration-id>
<webtob-home>c:\WebtoB\</webtob-home>
<read-timeout>120000</read-timeout>
<reconnect-timeout>60000</reconnect-timeout>
<webtob-backup>
    <port>9901</port>
    <output-buffer-size>
        16384
    </output-buffer-size>
    <thread-pool>
        <min>10</min>
        <max>10</max>
        <step>4</step>
        <max-idle-time>60000</max-idle-time>
        <max-wait-queue>2</max-wait-queue>
        <max-queue></max-queue>
        <thread-state-notify>
            <max-thread-active-time>
                150000
            </max-thread-active-time>
            <notify-threshold>
                100
            </notify-threshold>
            <restart-threshold>
                100
            </restart-threshold>
            <notifier-id>
                MyNotifier
            </notifier-id>
        </thread-state-notify>
    </thread-pool>
</webtob-backup>
</webtob>
```

```

        </notifier-id>
        <notify-subject>
            JEUS WEB CONTAINER THREAD
            STATE WARNING
        </notify-subject>
        <restart-subject>
            JEUS WEB CONTAINER RESTART
            WARNING
        </restart-subject>
    </thread-state-notify>
</thread-pool>
<postdata-read-timeout>
    40000
</postdata-read-timeout>
<scheme>http</scheme>
<hth-count>4</hth-count>
<request-queueing>true</request-queueing>
<disable-pipe>true</disable-pipe>
<webtob-address>
    111.111.111.112
</webtob-address>
<registration-id>MyGroup</registration-id>
<webtob-home>c:\WebtoB\</webtob-home>
<read-timeout>120000</read-timeout>
<reconnect-timeout>
    60000
</reconnect-timeout>
</webtob-backup>
</webtob-listener>
<apache-listener>
    <listener-id>WebListener2</listener-id>
    <port>8007</port>
    <output-buffer-size>16384</output-buffer-size>
    <thread-pool>
        <min>10</min>
        <max>20</max>
        <step>4</step>
        <max-idle-time>60000</max-idle-time>
        <max-wait-queue>2</max-wait-queue>
    </thread-pool>
</apache-listener>

```

```
<max-queue>50</max-queue>
<thread-state-notify>
  <max-thread-active-time>
    150000
  </max-thread-active-time>
  <notify-threshold>
    100
  </notify-threshold>
  <restart-threshold>
    18
  </restart-threshold>
  <notifier-id>MyNotifier</notifier-id>
  <notify-subject>
    JEUS WEB CONTAINER THREAD
    STATE WARNING
  </notify-subject>
  <restart-subject>
    JEUS WEB CONTAINER
    RESTART WARNING
  </restart-subject>
</thread-state-notify>
</thread-pool>
<postdata-read-timeout>
  40000
</postdata-read-timeout>
<scheme>http</scheme>
<back-log>100</back-log>
<busy-notify-page>
  c:\web\errorpages\busypage.html
</busy-notify-page>
<server-access-control>
  true
</server-access-control>
<allowed-server>127.0.0.1</allowed-server>
</apache-listener>
<http-listener>
  <listener-id>WebListener3</listener-id>
  <port>8088</port>
  <output-buffer-size>16384</output-buffer-size>
```

```
<thread-pool>
  <min>10</min>
  <max>20</max>
  <step>4</step>
  <max-idle-time>60000</max-idle-time>
  <max-wait-queue>2</max-wait-queue>
  <max-queue>50</max-queue>
  <thread-state-notify>
    <max-thread-active-time>
      150000
    </max-thread-active-time>
    <notify-threshold>
      100
    </notify-threshold>
    <restart-threshold>
      18
    </restart-threshold>
    <notifier-id>MyNotifier</notifier-id>
    <notify-subject>
      JEUS WEB CONTAINER THREAD
      STATE WARNING
    </notify-subject>
    <restart-subject>
      JEUS WEB CONTAINER
      RESTART WARNING
    </restart-subject>
  </thread-state-notify>
</thread-pool>
<postdata-read-timeout>
  40000
</postdata-read-timeout>
<scheme>http</scheme>
<back-log>100</back-log>
<server-access-control>
  true
</server-access-control>
<allowed-server>111.111.111.1</allowed-server>
</http-listener>
<tcp-listener>
```



```
<listener-id>WebListener4</listener-id>
<port>8088</port>
<output-buffer-size>16384</output-buffer-size>
<thread-pool>
  <min>10</min>
  <max>20</max>
  <step>4</step>
  <max-idle-time>60000</max-idle-time>
  <max-wait-queue>2</max-wait-queue>
  <max-queue>50</max-queue>
  <thread-state-notify>
    <max-thread-active-time>
      150000
    </max-thread-active-time>
    <notify-threshold>
      100
    </notify-threshold>
    <restart-threshold>
      18
    </restart-threshold>
    <notifier-id>MyNotifier</notifier-id>
    <notify-subject>
      JEUS WEB CONTAINER THREAD
      STATE WARNING
    </notify-subject>
    <restart-subject>
      JEUS WEB CONTAINER
      RESTART WARNING
    </restart-subject>
  </thread-state-notify>
</thread-pool>
<postdata-read-timeout>
  40000
</postdata-read-timeout>
<scheme>http</scheme>
<back-log>100</back-log>
<server-access-control>
  true
</server-access-control>
```

```
<allowed-server>111.111.111.1</allowed-server>
<dispatcher-config-class>
    mypackage.mydispatcher
</dispatcher-config-class>
</tcp-listener>
<secure-listener>
    <listener-id>WebListener5</listener-id>
    <port>8088</port>
    <output-buffer-size>16384</output-buffer-size>
    <thread-pool>
        <min>10</min>
        <max>20</max>
        <step>4</step>
        <max-idle-time>60000</max-idle-time>
        <max-wait-queue>2</max-wait-queue>
        <max-queue>50</max-queue>
        <thread-state-notify>
            <max-thread-active-time>
                150000
            </max-thread-active-time>
            <notify-threshold>
                100
            </notify-threshold>
            <restart-threshold>
                18
            </restart-threshold>
            <notifier-id>MyNotifier</notifier-id>
            <notify-subject>
                JEUS WEB CONTAINER THREAD
                STATE WARNING
            </notify-subject>
            <restart-subject>
                JEUS WEB CONTAINER
                RESTART WARNING
            </restart-subject>
        </thread-state-notify>
    </thread-pool>
    <postdata-read-timeout>
        40000
```

```
</postdata-read-timeout>
<scheme>https</scheme>
<back-log>100</back-log>
<server-access-control>
    true
</server-access-control>
<allowed-server>111.111.111.1</allowed-server>
</secure-listener>
<ajp13-listener>
    <listener-id>WebListener6</listener-id>
    <port>8007</port>
    <output-buffer-size>16384</output-buffer-size>
    <thread-pool>
        <min>10</min>
        <max>20</max>
        <step>4</step>
        <max-idle-time>60000</max-idle-time>
        <max-wait-queue>2</max-wait-queue>
        <max-queue>50</max-queue>
        <thread-state-notify>
            <max-thread-active-time>
                150000
            </max-thread-active-time>
            <notify-threshold>
                100
            </notify-threshold>
            <restart-threshold>
                18
            </restart-threshold>
            <notifier-id>MyNotifier</notifier-id>
            <notify-subject>
                JEUS WEB CONTAINER THREAD
                STATE WARNING
            </notify-subject>
            <restart-subject>
                JEUS WEB CONTAINER
                RESTART WARNING
            </restart-subject>
        </thread-state-notify>
    </thread-pool>
</ajp13-listener>
```

```
</thread-pool>
<postdata-read-timeout>
    40000
</postdata-read-timeout>
<scheme>http</scheme>
<back-log>100</back-log>
<busy-notify-page>
    c:\web\errorpages\busypage.html
</busy-notify-page>
<server-access-control>
    true
</server-access-control>
<allowed-server>127.0.0.1</allowed-server>
</ajp13-listener>
</webserver-connection>
<group-docbase>webapps</group-docbase>
<session-config>
    <timeout>20</timeout>
    <shared>true</shared>
    <persistent>true</persistent>
</session-config>
<print-error-to-browser>true</print-error-to-browser>
<encoding>
    <request-encoding>
        <default>UTF-8</default>
        <forced>UTF-8</forced>
    </request-encoding>
    <response-encoding>
        <default>UTF-8</default>
        <forced>UTF-8</forced>
    </response-encoding>
    <postdata-encoding>
        <default>UTF-8</default>
        <forced>UTF-8</forced>
    </postdata-encoding>
</encoding>
<jsp-engine>
    <keep-generated>true</keep-generated>
    <java-compiler>javac</java-compiler>
```

```
<jsp-work-dir>c:\MyJSPWorkdir\</jsp-work-dir>
<compile-output-dir>
  c:\MyJSPWorkdir\
</compile-output-dir>
<compile-option>-g:none -verbose</compile-option>
<compile-encoding>8859_1</compile-encoding>
<check-included-jspfile>
  true
</check-included-jspfile>
</jsp-engine>
<logging>
  <user-log>
    <level>FINE</level>
    <use-parent-handlers>
      true
    </use-parent-handlers>
    <handler>
      <smtp-handler>
        <name>smtpHandler</name>
        <level>SEVERE</level>
        <smtp-host-address>
          mail.com
        </smtp-host-address>
        <from-address>
          jeus@mail.com
        </from-address>
        <to-address>
          admin@mail.com
        </to-address>
        <send-for-all-messages>
          false
        </send-for-all-messages>
      </smtp-handler>
    </handler>
  </user-log>
  <access-log>
    <enable>true</enable>
    <format>
      [%{yyyy.MM.dd HH:mm:ss}t] %a
    </format>
  </access-log>
</logging>
```

```
        %m %U%q" %s %Dms
    </format>
    <handler>
        <file-handler>
            <name>fileHandler</name>
            <valid-hour>1</valid-hour>
        </file-handler>
    </handler>
</access-log>
</logging>
<session-cluster>
    <session-routing>true</session-routing>
    <session-server>
        <server-name>session1</server-name>
        <max>10</max>
        <min>2</min>
        <step>3</step>
        <get-connection-timeout>
            120000
        </get-connection-timeout>
        <backup-server-name>
            session2
        </backup-server-name>
        <connect-timeout>120000</connect-timeout>
        <read-timeout>60000</read-timeout>
    </session-server>
</session-cluster>
<response-header>
    <session-id-cookie>
        <version>1</version>
        <domain>.foo.com</domain>
        <path>/examples</path>
        <max-age>3600</max-age>
        <secure>true</secure>
    </session-id-cookie>
    <custom-header>
        <header-field>
            <field-name>Test</field-name>
            <field-value>TestValue</field-value>
```

```
        </header-field>
    </custom-header>
</response-header>
</context-group>
<db-connection-pool>
    <pool-name>oraclePool</pool-name>
    <pool-type>non-shared</pool-type>
    <pooling-rule>stack</pooling-rule>
    <connection-url>
        jdbc:oracle:thin:
        @111.111.111.111:1521:ORA805
    </connection-url>
    <driver-class-name>
        oracle.jdbc.driver.OracleDriver
    </driver-class-name>
    <connection-argument>
        user=scott;password=tiger
    </connection-argument>
    <listener-id>WebListener1</listener-id>
    <check-query>select date from dual</check-query>
    <dynamic-increment>true</dynamic-increment>
    <get-connection-timeout>30000</get-connection-timeout>
    <read-timeout>60000</read-timeout>
    <close-long-active-connection>
        true
    </close-long-active-connection>
    <max-active-time>300000</max-active-time>
    <max-use-count>30</max-use-count>
    <login-delay>100</login-delay>
    <close-delay>100</close-delay>
    <fetch-size>100</fetch-size>
    <db-pool-control>
        <min>10</min>
        <max>100</max>
        <step>10</step>
        <max-idle-time>300000</max-idle-time>
        <max-alive-time>600000</max-alive-time>
    </db-pool-control>
</dba-connection>
```

```
<forced-close>true</forced-close>
<connection-url>
    jdbc:oracle:thin
    @111.111.111.111:1521:ORA805
</connection-url>
<connection-argument>
    user=system;password=manager
</connection-argument>
</dba-connection>
<encryption>true</encryption>
</db-connection-pool>
<session-cluster>
    <session-routing>true</session-routing>
    <shared>true</shared>
    <session-server>
        <server-name>session1</server-name>
        <max>10</max>
        <min>2</min>
        <step>3</step>
        <get-connection-timeout>
            120000
        </get-connection-timeout>
        <backup-server-name>session2</backup-server-name>
        <connect-timeout>120000</connect-timeout>
        <read-timeout>60000</read-timeout>
    </session-server>
</session-cluster>
<security-switch>true</security-switch>
</web-container>
```


E jeus-web-dd.xml XML 설정 레퍼런스

E.1 소개

이 부록의 레퍼런스는 Web Context 의 deployment 디스크립터 “jeus-web-dd_[context name].xml”의 모든 XML 태그에 대해서 설명하고 있다. 이 파일의 XSD 파일은 “JEUS_HOME/config/xsds”디렉토리의 ”jeus-web-dd.xsd”파일이다.

이 레퍼런스는 3 개의 절로 나뉘져 있다.

1. **XSD/XML 트리**: XML 설정 파일의 모든 태그 리스트를 정리했다. 각 노드의 형식은 다음과 같다.

- a. 태그 레퍼런스로 빨리 찾아보기 위해서 각 태그마다 인덱스 번호(예 (11))를 붙여놓았다. 태그 레퍼런스에서는 이 번호 순서로 설명한다.
- b. XML Schema 에서 정의한 XML 태그명을 <tag name> 형식으로 표시한다.
- c. XML Schema 에서 정의한 Cardinality 를 표시한다. “?” = 0 개나 1 개의 element, “+” = 1 개 이상의 element, “*” = 0 개 이상의 element, (기호가 없음) = 정확히 1 개의 element
- d. 몇몇 태그에는 “P” 문자를 붙여놓았는데, 해당 태그는 성능에 관계되는 태그라는 것을 뜻한다. 이 태그는 설정을 튜닝할 때 사용된다.

2. **태그 레퍼런스**: 트리에 있는 각 XML 태그를 설명한다.

- a. **Description**: 태그에 대한 간단한 설명.
- b. **Value Description**: 입력하는 값과 타입.
- c. **Value Type**: 값의 데이터 타입 (예: String).

- d. **Default Value:** 해당 XML 을 사용하지 않았을 때 기본적으로 사용되는 값.
 - e. **Defined Values:** 이미 정해져 있는 값.
 - f. **Example:** 해당 XML 태그에 대한 Example.
 - g. **Performance Recommendation:** 성능 향상을 위해서 추천하는 값.
 - h. **Child Elements:** 자신의 태그 안에 사용되는 태그.
3. **Example XML 파일:** “jeus-web-dd_[context name].xml”에 대한 완전한 예제.

E.2 XML Schema/XML Tree

- (1) <jeus-web-dd>?
 - (2) <context-path>?
 - (3) <docbase>?
 - (4) <user-log>?
 - (5) <level>? P
 - (6) <use-parent-handlers>? P
 - (7) <filter-class>?
 - (8) <handler>?
 - (9) <console-handler>
 - (10) <name>
 - (11) <level>? P
 - (12) <encoding>?
 - (13) <filter-class>?
 - (14) <file-handler>
 - (15) <name>
 - (16) <level>? P
 - (17) <encoding>?
 - (18) <filter-class>?
 - (19) <file-name>?
 - (20) <valid-day>
 - (21) <valid-hour>
 - (22) <buffer-size>? P

```
(23) <append>? P
(24) <smtp-handler>
    (25) <name>
    (26) <level>? P
    (27) <encoding>?
    (28) <filter-class>?
    (29) <smtp-host-address>
    (30) <from-address>
    (31) <to-address>
    (32) <cc-address>?
    (33) <bcc-address>?
    (34) <send-for-all-messages>? P
(35) <socket-handler>
    (36) <name>
    (37) <level>? P
    (38) <encoding>?
    (39) <filter-class>?
    (40) <address>
    (41) <port>
(42) <user-handler>
    (43) <handler-class>
    (44) <name>
    (45) <level>? P
    (46) <encoding>?
    (47) <filter-class>?
    (48) <handler-property>?
        (49) <property>*
            (50) <key>
            (51) <value>
    (52) <formatter-class>? P
    (53) <formatter-property>?
        (54) <property>*
            (55) <key>
            (56) <value>
(57) <enable-jsp>? P
(58) <auto-reload>?
    (59) <enable-reload>? P
    (60) <check-on-demand>? P
(61) <max-instance-pool-size>? P
```

```
(62) <added-classpath>?
      (63) <class-path>*
(64) <allow-indexing>?
      (65) <directory>*
(66) <deny-download>?
      (67) <file>*
      (68) <extension>*
      (69) <directory>*
(70) <aliasing>?
      (71) <alias>*
          (72) <alias-name>
          (73) <real-path>
(74) <file-caching>?
      (75) <max-idle-time>? P
      (76) <max-cache-memory>? P
      (77) <directory>*
(78) <role-mapping>?
      (79) <role-permission>*
          (80) <principal>+
          (81) <role>
          (82) <actions>?
          (83) <classname>?
          (84) <excluded>?
          (85) <unchecked>?
(86) <ejb-ref>?
      (87) <jndi-info>*
          (88) <ref-name>
          (89) <export-name>
(90) <res-ref>?
      (91) <jndi-info>+
          (92) <ref-name>
          (93) <export-name>
(94) <res-env-ref>?
      (95) <jndi-info>*
          (96) <ref-name>
          (97) <export-name>
(98) <service-ref>?
      (99) <service-client>*
          (100) <service-ref-name>
```

```

(101) <port-info>*
    (102) <service-endpoint-interface>?
    (103) <wsdl-port>?
    (104) <stub-property>*
        (105) <name>
        (106) <value>
    (107) <call-property>*
        (108) <name>
        (109) <value>
    (110) <security>?
        (111) <request-sender>?
            (112) <action-list>
            (113) <password-callback-class>?
            (114) <user>
            (115) <timeToLive>?
            (116) <userNameToken>?
                (117) <passwordType>?
                (118) <userTokenElements>?
            (119) <signature-infos>?
                (120) <signature-info>+
                    (121) <signatureParts>?
                    (122) <keyIdentifier>
                    (123) <keystore>
                        (124) <key-type>
                        (125) <keystore-
password>
                                (126) <keystore-
filename>
                                (127) <encryption-infos>?
                                    (128) <encryption-info>+
                                        (129) <encryptionParts>?
                                        (130)
<encryptionSymAlgorithm>?
                                (131) <encryptionUser>?
                                (132) <keyIdentifier>
                                (133) <keystore>?
                                    (134) <key-type>
                                    (135) <keystore-
password>

```

```

(136) <keystore-
filename>
(137) <embeddedKey>?
(138)
<embeddedKeyCallbackClass>
(139) <key-name>
(140) <response-receiver>?
(141) <action-list>
(142) <password-callback-class>?
(143) <timeToLive>?
(144) <decryption>?
(145) <keystore>
(146) <key-type>
(147) <keystore-password>
(148) <keystore-filename>
(149) <signature-verification>?
(150) <keystore>
(151) <key-type>
(152) <keystore-password>
(153) <keystore-filename>
(154) <service-impl-class>?
(155) <wsdl-override>?
(156) <service-qname>?
(157) <call-property>*
(158) <name>
(159) <value>
(160) <url-rewriting>? P
(161) <enable-default-login>? P
(162) <jsp-engine>?
(163) <keep-generated>? P
(164) <java-compiler>? P
(165) <jsp-work-dir>?
(166) <compile-output-dir>?
(167) <compile-option>?
(168) <compile-encoding>?
(169) <check-included-jspfile>? P
(170) <keep-generated>?
(171) <fast-deploy>?
(172) <servlet>*

```

```

(173) <servlet-name>
(174) <run-as-identity>?
      (175) <principal-name>
(176) <webinf-first>? P

```

E.3 Element 레퍼런스

(1) <jeus-web-dd>

Description JEUS applciation(Servlet Application) Deployment Descriptor contains exactly one of these elements at the top level. The sub-element of this element (<context;>) contains information pertaining to the deployment and registration of on context (Web application).

Child Elements

```

(2)context-path?
(3)docbase?
(4)user-log?
(57)enable-jsp?
(58)auto-reload?
(61)max-instance-pool-size?
(62)added-classpath?
(64)allow-indexing?
(66)deny-download?
(70)aliasing?
(74)file-caching?
(78)role-mapping?
(86)ejb-ref?
(90)res-ref?
(94)res-env-ref?
(98)service-ref?
(160)url-rewriting?
(161)enable-default-login?
(162)jsp-engine?
(170)keep-generated?
(171)fast-deploy?

```

```
(172)servlet*
(176)webinf-first?
```

(2) <jeus-web-dd> **<context-path>**

<i>Description</i>	The URL path of the context. This path is entered by the user directly after the domain name (and port number). Example: if the context path of a context A is "/examples", the URL "http://www.foo.com/examples/index.html" would result in the index.html file of the context A to be fetched. Note: this element must be set if it is not set in WEBMain.xml.
<i>Value Description</i>	An URL path starting with a "/" character. The path must be unique within this virtual host.
<i>Value Type</i>	token

(3) <jeus-web-dd> **<docbase>**

<i>Description</i>	This element sets the context's document base directory where all the files of the Web application will be stored. This directory is relative to and will be appended to the containing context group's document base directory (as set in the <context-group><group-docbase> element). It is also possible to specify an absolute path. If this element is not set, the context will be stored in the default context directory (which means directly below the context group directory).
<i>Value Description</i>	An absolute or relative directory path for this context. If relative directory path (NOT starting with a directory separator '\ ' or '/') is specified, that path will be appended to the context group's document base.
<i>Value Type</i>	token

(4) <jeus-web-dd> **<user-log>**

<i>Description</i>	This element sets the handling of messages that were defined by the programmer using the ServletContext.log() method (from within a Servlet). This element is identical to the <context-
--------------------	--

group><logging><user-log> element with the exception that the default log file used will be named after the name of the context (e.g. "Examples.log"). By using this element, log files for individual contexts can be obtained.

Child Elements

(5) level?
 (6) use-parent-handlers?
 (7) filter-class?
 (8) handler?

(5) <jeus-web-dd> <user-log> <level>

Description Level of logging is set for this field. Please refer to Level class of logging API of J2SE.

Value Type loggingLevelType

Default Value INFO

Defined Value FATAL

NOTICE

INFORMATION

DEBUG

fatal

notice

information

debug

SEVERE

WARNING

INFO

CONFIG

FINE

FINER

FINEST

ALL

severe

warning

info

config

fine

finer

finest

all

FATAL

NOTICE

INFORMATION

DEBUG

fatal

notice

information

debug

SEVERE

WARNING

INFO

CONFIG

FINE

FINER

FINEST

ALL

severe

warning

info

config

fine

finer

finest

all

(6) <jeus-web-dd> <user-log> **<use-parent-handlers>**

Description This determines whether handlers of parent logger will be used for this logger.

Value Type boolean

Default Value false

(7) <jeus-web-dd> <user-log> **<filter-class>**

Description Fully qualified class name of filter class set in this logger is set for this field.

Value Type token

Example <filter-class>com.tmax.logging.filter.MyFilter</filter-class>

(8) <jeus-web-dd> <user-log> **<handler>**

Description Handler to use in logger is set for this field.

Child Elements (9)console-handler
 (14)file-handler
 (24)smtp-handler
 (35)socket-handler
 (42)user-handler

(9) <jeus-web-dd> <user-log> <handler> **<console-handler>**

Description This handler will be used when logging is left in the screen.

Child Elements

- (10) name
- (11) level?
- (12) encoding?
- (13) filter-class?

```
(10) <jeus-web-dd> <user-log> <handler> <console-handler> <name>
```

Description The name of handler will be set. At this moment, this name should be set as unique in a logger. This name is used for refering handler including tool.

Value Type token

Example <name>handler1</name>

```
(11) <jeus-web-dd> <user-log> <handler> <console-handler>
<level>
```

Description The level of this handler will be set for this field. In case where level of message going through logger is included in this handler level, the message will be displayed through this handler.

Value Type loggingLevelType

Default Value FINEST

Defined Value FATAL

NOTICE

INFORMATION

DEBUG

fatal

notice

information

debug

SEVERE

WARNING

INFO

CONFIG

FINE

FINER

FINEST

ALL

severe

warning

info

config

fine

finer

finest

all

FATAL

NOTICE

INFORMATION

DEBUG

fatal

notice

information

debug

SEVERE

WARNING

INFO

CONFIG

FINE

FINER

FINEST

ALL

severe

warning

info

config

fine

finer

finest

all

```
(12) <jeus-web-dd> <user-log> <handler> <console-handler>
<encoding>
```

Description Encoding in which this handler leaves message is set for this field.

Value Type token

```
(13) <jeus-web-dd> <user-log> <handler> <console-handler>
<filter-class>
```

Description Fully qualified class name of filter class set in this handler is set for this field.

Value Type token

Example <filter-
class>com.tmax.logging.filter.MyFilter</filter-
class>

```
(14) <jeus-web-dd> <user-log> <handler> <b>file-handler</b>
```

Description This is the handler to be used in case of displaying logging as file.

Child Elements (15)name
 (16)level?
 (17)encoding?

(18)filter-class?
 (19)file-name?
 (20)valid-day
 (21)valid-hour
 (22)buffer-size?
 (23)append?

```
(15) <jeus-web-dd> <user-log> <handler> <file-handler> <name>
```

Description The name of handler will be set. At this moment, this name should be set as unique in a logger. This name is used for refering handler including tool.

Value Type token

Example <name>handler1</name>

```
(16) <jeus-web-dd> <user-log> <handler> <file-handler> <level>
```

Description The level of this handler will be set for this field. In case where level of message going through logger is included in this handler level, the message will be displayed through this handler.

Value Type loggingLevelType

Default Value FINEST

Defined Value FATAL

NOTICE

INFORMATION

DEBUG

fatal

notice

information

debug

SEVERE

WARNING

INFO

CONFIG

FINE

FINER

FINEST

ALL

severe

warning

info

config

fine

finer

finest

all

FATAL

NOTICE

INFORMATION

DEBUG

fatal

notice

information

debug

SEVERE

WARNING

INFO

CONFIG

FINE

FINER

FINEST

ALL

severe

warning

info

config

fine

finer

finest

all

```
(17) <jeus-web-dd> <user-log> <handler> <file-handler>
<encoding>
```

Description Encoding in which this handler leaves message is set for this field.

Value Type token

```
(18) <jeus-web-dd> <user-log> <handler> <file-handler> <filter-
class>
```

Description Fully qualified class name of filer class set in this handler is set for this field.

Value Type token

Example <filter-
class>com.tmax.logging.filter.MyFilter</filter-
class>

```
(19) <jeus-web-dd> <user-log> <handler> <file-handler> <file-
name>
```

Description File name in which this handler uses will be set for this field. If user has not set for this field, default file name of each logger will

be used. Please refer to JEUS Server Manual for each default file name.

Value Type token

Example <file-name>C:\logs\mylog.log</file-name>

```
(20) <jeus-web-dd> <user-log> <handler> <file-handler> <valid-day>
```

Description File in which this handler uses will be used and updated frequently for certain set amount of time . This setting is used for changing file as unit of day. In this case, used date will be followed right after file name.

Value Description day

Value Type off-intType

Value Type Description The default value is in non-negative int type. In case where the value is -1, it is considered as off.

Example <valid-day>1</valid-day>

```
(21) <jeus-web-dd> <user-log> <handler> <file-handler> <valid-hour>
```

Description In case of using and updating frequently on the file in which this handler uses for a certain amount of time. This setting is used for updating time unit. In this case, used date and time will be automatically followed right after file name in which this handler uses.

Value Description This represents in unit of hour and it should get divisor of 24 + N*24(n is integer greater than 0).

Value Type off-intType

Value Type Description The default value is in non-negative int type. In case where the value is -1, it is considered as off.

Example <valid-hour>3</valid-hour>

(22) <jeus-web-dd> <user-log> <handler> <file-handler> **<buffer-size>**

Description The size of buffer is set when displaying of this handler as file.

Value Description The unit in byte. [Performance Recommendation]: The logging performance will be improved as this value is bigger even though message will be displayed with delay.

Value Type nonNegativeIntType

Value Type Description This element is int type which is over 0. Thus, it includes values which are over 0 in the range of int type.

Default Value 1024

(23) <jeus-web-dd> <user-log> <handler> <file-handler> **<append>**

Description In case where file in which this handler uses already exists, this element will be used for determining whether to be written after the file name. If it has been set as false, existing file will be deleted.

Value Type boolean

Default Value true

(24) <jeus-web-dd> <user-log> <handler> **<smtp-handler>**

Description This is the handler to use when sending logging to email.
[Performance Recommendation]: There is a warning when a huge amount of email exist without proper filter.

Child Elements (25)name
 (26)level?
 (27)encoding?
 (28)filter-class?
 (29)smtp-host-address
 (30)from-address

- (31) to-address
- (32) cc-address?
- (33) bcc-address?
- (34) send-for-all-messages?

```
(25) <jeus-web-dd> <user-log> <handler> <smtp-handler> <name>
```

<i>Description</i>	The name of handler will be set. At this moment, this name should be set as unique in a logger. This name is used for refering handler including tool.
<i>Value Type</i>	token
<i>Example</i>	<name>handler1</name>

```
(26) <jeus-web-dd> <user-log> <handler> <smtp-handler> <level>
```

<i>Description</i>	The level of this handler will be set for this field. In case where level of message going through logger is included in this handler level, the message will be displayed through this handler.
<i>Value Type</i>	loggingLevelType
<i>Default Value</i>	FINEST
<i>Defined Value</i>	FATAL
	NOTICE
	INFORMATION
	DEBUG
	fatal
	notice
	information

debug

SEVERE

WARNING

INFO

CONFIG

FINE

FINER

FINEST

ALL

severe

warning

info

config

fine

finer

finest

all

FATAL

NOTICE

INFORMATION

DEBUG

fatal

notice

information

debug

SEVERE

WARNING

INFO

CONFIG

FINE

FINER

FINEST

ALL

severe

warning

info

config

fine

finer

finest

all

```
(27) <jeus-web-dd> <user-log> <handler> <smtp-handler>
<encoding>
```

Description Encoding in which this handler leaves message is set for this field.

Value Type token

```
(28) <jeus-web-dd> <user-log> <handler> <smtp-handler> <filter-
class>
```

Description Fully qualified class name of filter class set in this handler is set for this field.

Value Type token

Example <filter-
class>com.tmax.logging.filter.MyFilter</filter-
class>

```
(29) <jeus-web-dd> <user-log> <handler> <smtp-handler> <smtp-
host-address>
```

Description The address of Smtip server to send email is set for this field.

Value Type token

```
(30) <jeus-web-dd> <user-log> <handler> <smtp-handler> <from-
```

address>

Description The address of sender for email is set for this field.

Value Type token

(31) <jeus-web-dd> <user-log> <handler> <smtp-handler> **<to-address>**

Description The address of receiver of email is set for this field.

Value Type token

(32) <jeus-web-dd> <user-log> <handler> <smtp-handler> **<cc-address>**

Description The cc-address of receiver is set for this field.

Value Type token

(33) <jeus-web-dd> <user-log> <handler> <smtp-handler> **<bcc-address>**

Description The bcc-address of receiver is set for this field.

Value Type token

(34) <jeus-web-dd> <user-log> <handler> <smtp-handler> **<send-for-all-messages>**

Description When messages enter into this handler through using log() method of logger registered in this handler, it determines whether to consider it as the sender object of email. If it has been set as false, messages which are only called through using specific send() method are transmitted.

Value Type boolean

Default Value false

(35) <jeus-web-dd> <user-log> <handler> **<socket-handler>**

<i>Description</i>	This element is handler used for sending to socket in which logging is set. [Performance Recommendation]: The performance will be decreased without using proper filter since logging message is transmitted as one Socket.
--------------------	---

<i>Child Elements</i>	(36)name (37)level? (38)encoding? (39)filter-class? (40)address (41)port
-----------------------	---

```
(36) <jeus-web-dd> <user-log> <handler> <socket-handler> <name>
```

<i>Description</i>	The name of handler will be set. At this moment, this name should be set as unique in a logger. This name is used for refering handler including tool.
--------------------	--

<i>Value Type</i>	token
-------------------	-------

<i>Example</i>	<name>handler1</name>
----------------	-----------------------

```
(37) <jeus-web-dd> <user-log> <handler> <socket-handler> <level>
```

<i>Description</i>	The level of this handler will be set for this field. In case where level of message going through logger is included in this handler level, the message will be displayed through this handler.
--------------------	--

<i>Value Type</i>	loggingLevelType
-------------------	------------------

<i>Default Value</i>	FINEST
----------------------	--------

<i>Defined Value</i>	FATAL
----------------------	-------

NOTICE

INFORMATION

DEBUG

fatal

notice

information

debug

SEVERE

WARNING

INFO

CONFIG

FINE

FINER

FINEST

ALL

severe

warning

info

config

fine

finer

finest

all

FATAL

NOTICE

INFORMATION

DEBUG

fatal

notice

information

debug

SEVERE

WARNING

INFO

CONFIG

FINE

FINER

FINEST

ALL

severe

warning

info

config

fine

finer

finest

all

```
(38) <jeus-web-dd> <user-log> <handler> <socket-handler>
<encoding>
```

Description Encoding in which this handler leaves message is set for this field.

Value Type token

```
(39) <jeus-web-dd> <user-log> <handler> <socket-handler>
<filter-class>
```

Description Fully qualified class name of filter class set in this handler is set for this field.

Value Type token

Example

```
<filter-
class>com.tmax.logging.filter.MyFilter</filter-
class>
```

```
(40) <jeus-web-dd> <user-log> <handler> <socket-handler>
```

<address>

Description IP address of sending messages when generating this handler.

Value Type token

(41) <jeus-web-dd> <user-log> <handler> <socket-handler> **<port>**

Description Port of sending messages when generating this handler.

Value Type nonNegativeIntType

Value Type Description This element is int type which is over 0. Thus, it includes values which are over 0 in the range of int type.

(42) <jeus-web-dd> <user-log> <handler> **<user-handler>**

Description In case of using handler in which user has created according to J2SE logging API, this setting

Child Elements (43)handler-class
 (44)name
 (45)level?
 (46)encoding?
 (47)filter-class?
 (48)handler-property?
 (52)formatter-class?
 (53)formatter-property?

(43) <jeus-web-dd> <user-log> <handler> <user-handler> **<handler-class>**

Description Fully qualified class name in which user has created is set. This class inherits java.util.logging.Handler and jeus.util.logging.JeusHandler will be constructed.

Value Type token

Example <handler-
 class>com.tmax.logging.handler.MyHandler</handler-
 class>


```
(44) <jeus-web-dd> <user-log> <handler> <user-handler> <name>
```

Description The name of handler will be set. At this moment, this name should be set as unique in a logger. This name is used for refering handler including tool.

Value Type token

Example <name>handler1</name>

```
(45) <jeus-web-dd> <user-log> <handler> <user-handler> <level>
```

Description The level of this handler will be set for this field. In case where level of message going through logger is included in this handler level, the message will be displayed through this handler.

Value Type loggingLevelType

Default Value FINEST

Defined Value

FATAL

NOTICE

INFORMATION

DEBUG

fatal

notice

information

debug

SEVERE

WARNING

INFO

CONFIG

FINE

FINER

FINEST

ALL

severe

warning

info

config

fine

finer

finest

all

FATAL

NOTICE

INFORMATION

DEBUG

fatal

notice

information

debug

SEVERE

WARNING

INFO

CONFIG

FINE

FINER

FINEST

ALL

severe

warning

info

config

fine

finer

finest

all

```
(46) <jeus-web-dd> <user-log> <handler> <user-handler>
<encoding>
```

Description Encoding in which this handler leaves message is set for this field.

Value Type token

```
(47) <jeus-web-dd> <user-log> <handler> <user-handler> <filter-
class>
```

Description Fully qualified class name of filter class set in this handler is set for this field.

Value Type token

Example <filter-
class>com.tmax.logging.filter.MyFilter</filter-
class>

```
(48) <jeus-web-dd> <user-log> <handler> <user-handler> <handler-
property>
```

Description When handler is generated, property to parse is set. These properties are delivered to handler as being stored in Map object as key-value through JeusHandler.setProperty() method.

Child Elements (49)property*

```
(49) <jeus-web-dd> <user-log> <handler> <user-handler> <handler-
property> <property>
```

Description Property to be delivered to handler is set.

Child Elements (50)key
(51)value

```
(50) <jeus-web-dd> <user-log> <handler> <user-handler> <handler-  
property> <property> <key>
```

Description This element is the key value of property.

Value Type token

```
(51) <jeus-web-dd> <user-log> <handler> <user-handler> <handler-  
property> <property> <value>
```

Description The value of property.

Value Type token

```
(52) <jeus-web-dd> <user-log> <handler> <user-handler>  
<formatter-class>
```

Description Fully qualified class name of formatter in which this handler is used. This class inherits java.util.logging.Formatter and jeus.util.logging.JeusFormatter will be constructed.

Value Type token

Default Value jeus.util.logging.SimpleFormatter

Example `<formatter-
class>com.tmax.logging.handler.MyHandler</formatter-
class>`

```
(53) <jeus-web-dd> <user-log> <handler> <user-handler>  
<formatter-property>
```

Description Property to parse to formatter when handler is generated. This properties are delivered to formatter through using JeusFormatter.setProperty() method stored in Map object.

Child Elements (54)property*

```
(54) <jeus-web-dd> <user-log> <handler> <user-handler>
<formatter-property> <property>
```

Description Property to be delivered to handler is set.

Child Elements (55)key
 (56)value

```
(55) <jeus-web-dd> <user-log> <handler> <user-handler>
<formatter-property> <property> <key>
```

Description This element is the key value of property.

Value Type token

```
(56) <jeus-web-dd> <user-log> <handler> <user-handler>
<formatter-property> <property> <value>
```

Description The value of property.

Value Type token

```
(57) <jeus-web-dd> <enable-jsp>
```

Description Determines the usage availabiltiy of JSP engine.

Value Description Activating or inactivating this configuration as Boolean value.

Value Type boolean

Value Type Description Boolean

Default Value true

Defined Value true
 Activating JSP in Web Container.

false

Inactivating JSP in Web Container. When it is set, this value will prevent from having an access to entire JSP files(Returning

Service Unavailable message as browser.)

(58) <jeus-web-dd> **<auto-reload>**

Description Determines whether to reload Servlets classes and custom tag classes automatically if the classes are changed. This feature should only be enabled in a development environment because of the extra processing power required to do the checking and reloading.

Child Elements (59)enable-reload?
(60)check-on-demand?

(59) <jeus-web-dd> <auto-reload> **<enable-reload>**

Description Determine whether to activate or inactivate Servlet auto-reload.

Value Description Activating or inactivating this configuration as Boolean value.

Value Type boolean

Value Type Description Boolean

Default Value false

Defined Value true
Activating the related configuration.

false
Inactivating the related configuration .

(60) <jeus-web-dd> <auto-reload> **<check-on-demand>**

Description Set for inspecting or not when each request reached for reloading Servlet class.

Value Description Activating or inactivating this configuration as Boolean value.

Value Type boolean

Value Type Description Boolean

<i>Default Value</i>	false
<i>Defined Value</i>	true
	Activating the related configuration.
	false
	Inactivating the related configuration .

(61) <jeus-web-dd> **<max-instance-pool-size>**

<i>Description</i>	Sets the maximum number of Servlet instances to create if the single threaded model is used (thus only for Servlets implementing the "SingleThreadedModel" interface). If more requests for a single threaded Servlet arrive than the maximum allowed instance count set here, those requests will be queued until a single threaded Servlet instance becomes available.
<i>Value Description</i>	Maximum number of single threaded Servlet instance to be generated for this Context group.
<i>Value Type</i>	int
<i>Value Type Description</i>	Integer
<i>Default Value</i>	-1
<i>Defined Value</i>	-1
	If this value is used, there will be no limitation for the number of instances on single threaded Servlets

(62) <jeus-web-dd> **<added-classpath>**

<i>Description</i>	This element specifies additional classes to add to the class path when Servlets are compiled or executed. By default, the classes and packages below the "\\WEB-INF\\classes\\" directory as well as any JAR files below the "\\WEB-INF\\lib\\" directory are included in the class path. The "\\WEB-INF\\" directory is located just beneath the context's document base directory.
--------------------	---

Child Elements (63)class-path*

(63) <jeus-web-dd> <added-classpath> **<class-path>**

Description Declares user defined classpath for adding when executing and compiling Servlet.

Value Description Directory or JAR file can be configured. When configuring the directory, it should end with directory classifier ("\" Windows ,"/" Unix).

Value Type token

Example <class-path>c:\mylib\subdir\</class-path>

(64) <jeus-web-dd> **<allow-indexing>**

Description The sub-elements of this element specifies the URL paths that may be indexed. Indexing means that the contents of the underlying directory of an URL path is listed in the Web browsers window. Indexing occurs if three conditions are fulfilled: 1. the user requested an URL path that did not contain an explicit file or resource name (e.g. no "index.html" was specified), 2. no "default welcome file" was specified in the Web applications's web.xml descriptor file and 3. indexing is allowed for the requested directory (allowed directories are listed in the sub-elements of this element). If all of these conditions are fulfilled, a directory list will be sent back to the client.

Child Elements (65)directory*

(65) <jeus-web-dd> <allow-indexing> **<directory>**

Description This element activates directory indexing on set URL path.

Value Description URL path starting and ending with "/".

Value Type token

(66) <jeus-web-dd> **<deny-download>**

<i>Description</i>	This element specifies any files that should not be downloadable under any circumstances. For example, if a file named "secret.dat" exists in a directory called "data" of some context with the context path "/customers" in a domain called "www.foo.com" and a client requests the URL "www.foo.com/customers/data/secret.dat", the "secret.dat" file would normally be downloaded. To prevent this from happening you will need to make use of this XML element in order to create a download denial condition that matches the file "secret.
--------------------	---

<i>Child Elements</i>	(67)file* (68)extension* (69)directory*
-----------------------	---

```
(67) <jeus-web-dd> <deny-download> <file>
```

<i>Description</i>	This element makes sure that set file will not be downloaded.
--------------------	---

<i>Value Description</i>	The path followed by file name according to URL context.
--------------------------	--

<i>Value Type</i>	token
-------------------	-------

<i>Example</i>	<file>/data/secret.dat</file>
----------------	-------------------------------

```
(68) <jeus-web-dd> <deny-download> <extension>
```

<i>Description</i>	It rejects downloading entire file in Context for given extension.
--------------------	--

<i>Value Description</i>	File Extension
--------------------------	----------------

<i>Value Type</i>	token
-------------------	-------

<i>Example</i>	<extension>dat</extension>
----------------	----------------------------

```
(69) <jeus-web-dd> <deny-download> <directory>
```

<i>Description</i>	For preventing on downloading an entire file in certain directory, this element is used (Related to URL path of context).
--------------------	---

<i>Value Description</i>	Directory starting and ending with "/". Directory related to URL path of Context.
--------------------------	---

Value Type token

Example <directory>/data/</directory>

(70) <jeus-web-dd> **<aliasing>**

Description Aliasing is used to create symbolic links from requested URL paths to actual file system directories. By default, each URL path below a context path will be assumed to reflect a sub-directory of the context's document base directory. However, if this is not the case and the URL path should be directed to a different directory, this element should be used to override the default behaviour.

Child Elements (71)alias*

(71) <jeus-web-dd> <aliasing> **<alias>**

Description Each <alias> element defines one mapping between a request URL and an arbitrary directory. Example: the user requests the URL "www.foo.com/examples/images/hello.gif". By default the container would look for the "hello.gif" file in the directory "/images/" below the "Example" context's base directory. However, if the "hello.gif" file is located in the "c:\web\images\" directory and not in the context, an alias element would be needed. If several mappings are necessary, use several instances of this element.

Child Elements (72)alias-name
(73)real-path

(72) <jeus-web-dd> <aliasing> <alias> **<alias-name>**

Description This element the URL path that should be mapped to a specific directory.

Value Description Directory starting and ending with "/". Directory related with URL path of Context.

Value Type token

Example <alias-name>/images/</alias-name>

(73) <jeus-web-dd> <aliasing> <alias> **<real-path>**

Description This element the URL path that should be mapped to <alias-name> element.

Value Description An URL path starting and ending with a "/". The path is relative to the context's path ("\" Window,"/" Unix).

Value Type token

Example <real-path>c:\web\images\</real-path>

(74) <jeus-web-dd> **<file-caching>**

Description This element configures the caching of static resources such as images and HTML files. Caching means that often-requested files are temporarily stored and fetched from main memory instead of from the secondary storage (e.g. the hard disc drive). This will increase the performance of fetching such static resources.

Child Elements (75)max-idle-time?
(76)max-cache-memory?
(77)directory*

(75) <jeus-web-dd> <file-caching> **<max-idle-time>**

Description Time period in where Cached static resource is not used for the time except for caching the related resource.

Value Description Unit of Millisecond.

Value Type int

Default Value -1

Example <max-idle-time>1800</max-idle-time>

(76) <jeus-web-dd> <file-caching> **<max-cache-memory>**

Description Allocated maximum memory for storing static resource.

Value Description The number of Megabytes unit

Value Type int

Value Type Description Integer

Default Value -1

Defined Value -1
It is not limited for cache size.

(77) <jeus-web-dd> <file-caching> **<directory>**

Description Static contents are set for one directory path to be cached. For configuring several numbers of cache directory, the number of elements are added.

Value Description URL Directory path related with Context docbase to be cached by Content.

Value Type token

(78) <jeus-web-dd> **<role-mapping>**

Description This element maps any J2EE role names used in the <security-role> element of the web.xml file of this context to actual system user names. The web.xml file will be located under the "\WEB-INF\" directory of this context's document base directory.

Child Elements (79)role-permission*

(79) <jeus-web-dd> <role-mapping> **<role-permission>**

Child Elements (80)principal+
(81)role
(82)actions?
(83)classname?
(84)excluded?
(85)unchecked?

(80) <jeus-web-dd> <role-mapping> <role-permission> **<principal>**

Description Role related to user principal is set.

Value Description Principal name set in subjects.xml of security.

Value Type token

(81) <jeus-web-dd> <role-mapping> <role-permission> **<role>**

Description Role name contributed to principals is set.

Value Type token

(82) <jeus-web-dd> <role-mapping> <role-permission> **<actions>**

Description Action for this role permission object is defined. There is no action for role permission

Value Type token

(83) <jeus-web-dd> <role-mapping> <role-permission> **<classname>**

Description Role permission class name to use is set. If it has not been set, it uses class in which JEUS provides as default.

Value Type token

(84) <jeus-web-dd> <role-mapping> <role-permission> **<excluded>**

Description This disables for users to use role.

Example <excluded/>

(85) <jeus-web-dd> <role-mapping> <role-permission> **<unchecked>**

Description This enables for users to use role without checking anything.

Example <unchecked/>

(86) <jeus-web-dd> **<ejb-ref>**

Description This element is connected to JNDI name used in actual system for

EJB reference set in web.xml

Child Elements (87) jndi-info*

```
(87) <jeus-web-dd> <ejb-ref> <jndi-info>
```

Description Each <jndi-info> element connects with JNDI name in one reference.

Child Elements (88) ref-name
(89) export-name

```
(88) <jeus-web-dd> <ejb-ref> <jndi-info> <ref-name>
```

Description This element can declare reference name to be used in source code.

Value Description Reference name to be binded in actual JNDI name. This element corresponds with ref-name of J2EE Standard descriptor element.

Value Type token

Example <ref-name>ejb/AccountEJB</ref-name>

```
(89) <jeus-web-dd> <ejb-ref> <jndi-info> <export-name>
```

Description Actual JNDI name defined in JEUS DD.

Value Type token

Example <export-name>ACCEJB</export-name>

```
(90) <jeus-web-dd> <res-ref>
```

Description This element links symbolic resource references (for example datasource references) in web.xml to the actual JNDI names used in the system.

Child Elements (91) jndi-info+

```
(91) <jeus-web-dd> <res-ref> <jndi-info>
```

Description This element configures export name and reference name of each

resource reference to be registered in JNDI of this Engine Container.

Child Elements (92)ref-name
 (93)export-name

(92) <jeus-web-dd> <res-ref> <jndi-info> **<ref-name>**

Description This element can declare reference name to be used in source code.

Value Description Reference name to be binded in actual JNDI name. This element corresponds with ref-name of J2EE Standard descriptor element.

Value Type token

Example <ref-name>ejb/AccountEJB</ref-name>

(93) <jeus-web-dd> <res-ref> <jndi-info> **<export-name>**

Description Actual JNDI name defined in JEUS DD.

Value Type token

Example <export-name>ACCEJB</export-name>

(94) <jeus-web-dd> **<res-env-ref>**

Description This element links symbolic resource environment references (source code references to administered objects) in web.xml to the actual JNDI names used in the system.

Child Elements (95)jndi-info*

(95) <jeus-web-dd> <res-env-ref> **<jndi-info>**

Description Each <jndi-info> element connects with JNDI name in one reference.

Child Elements (96)ref-name
 (97)export-name


```
(96) <jeus-web-dd> <res-env-ref> <jndi-info> <ref-name>
```

<i>Description</i>	This element can declare reference name to be used in source code.
<i>Value Description</i>	Reference name to be binded in actual JNDI name. This element corresponds with ref-name of J2EE Standard descriptor element.
<i>Value Type</i>	token
<i>Example</i>	<ref-name>ejb/AccountEJB</ref-name>

```
(97) <jeus-web-dd> <res-env-ref> <jndi-info> <export-name>
```

<i>Description</i>	Actual JNDI name defined in JEUS DD.
<i>Value Type</i>	token
<i>Example</i>	<export-name>ACCEJB</export-name>

```
(98) <jeus-web-dd> <service-ref>
```

<i>Description</i>	Root element on configuration documentation of JEUS Web Service Client (root element).
<i>Child Elements</i>	(99)service-client*

```
(99) <jeus-web-dd> <service-ref> <service-client>
```

<i>Description</i>	Settings for Web Service Client to be batched are specified for this field.
<i>Child Elements</i>	(100)service-ref-name (101)port-info* (154)service-impl-class? (155)wsdl-override? (156)service-qname? (157)call-property*

```
(100) <jeus-web-dd> <service-ref> <service-client> <service-ref-name>
```

Description This element is an endpoint name of Web Service related to WSDL file. <webservice-description-name> element corresponds to <webservice-description-name> of webservice.xml which is standard batch descriptor.

Value Type token

```
(101) <jeus-web-dd> <service-ref> <service-client> <port-info>
```

Description Web Service port information to be batched in which Web Service client calls is specified for this field.

Child Elements (102)service-endpoint-interface?
 (103)wsdl-port?
 (104)stub-property*
 (107)call-property*
 (110)security?

```
(102) <jeus-web-dd> <service-ref> <service-client> <port-info>  
<service-endpoint-interface>
```

Description The class presenting service endpoint interface of WSDL port is specified. This element corresponds to <service-endpoint-interface> of <port-component-ref> in <service-ref>.

Value Type token

```
(103) <jeus-web-dd> <service-ref> <service-client> <port-info>  
<wsdl-port>
```

Description This element describes WSDL port definition connecting to <port-info>.

Value Type QName

```
(104) <jeus-web-dd> <service-ref> <service-client> <port-info>  
<stub-property>
```

Description The properties which are set in jacax.xml.rpc.Stub using specific port are specified for this field.

Child Elements (105)name
(106)value

```
(105) <jeus-web-dd> <service-ref> <service-client> <port-info>
<stub-property> <name>
```

Description The key name for setting property on javax.xml.rpc.Call or javax.xml.rpc.Stub is specified for this field.

Value Type string

```
(106) <jeus-web-dd> <service-ref> <service-client> <port-info>
<stub-property> <value>
```

Description The value corresponds to the key used for setting property on javax.xml.rpc.Stub or javax.xml.rpc.Call.

Value Type string

```
(107) <jeus-web-dd> <service-ref> <service-client> <port-info>
<call-property>
```

Description The properties to set in javax.xml.rpc.Call object being used in specific prot are specified for this field.

Child Elements (108)name
(109)value

```
(108) <jeus-web-dd> <service-ref> <service-client> <port-info>
<call-property> <name>
```

Description The key name for setting property on javax.xml.rpc.Call or javax.xml.rpc.Stub is specified for this field.

Value Type string

```
(109) <jeus-web-dd> <service-ref> <service-client> <port-info>
<call-property> <value>
```

Description The value corresponds to the key used for setting property on javax.xml.rpc.Stub or javax.xml.rpc.Call.

Value Type string

```
(110) <jeus-web-dd> <service-ref> <service-client> <port-info>
<security>
```

Description This is configuration for Web Service Client configuration of Web Service Security (WS-Security).

Child Elements (111)request-sender?
(140)response-receiver?

```
(111) <jeus-web-dd> <service-ref> <service-client> <port-info>
<security> <request-sender>
```

Description This is configuration for applying security in message to call Web Service.

Child Elements (112)action-list
(113)password-callback-class?
(114)user
(115)timeToLive?
(116)userNameToken?
(119)signature-infos?
(127)encryption-infos?

```
(112) <jeus-web-dd> <service-ref> <service-client> <port-info>
<security> <request-sender> <action-list>
```

Description This enumerates security type to be applied in String type. One of Timestamp, Encrypt, Signature or UsernameToken can be entered. Each field is classified as blank.(Ex) UsernameToken Signature Encrypt)

Value Type string

```
(113) <jeus-web-dd> <service-ref> <service-client> <port-info>
<security> <request-sender> <password-callback-class>
```

Description This element is full package name of callback class for configuring password.

Value Type string

```
(114) <jeus-web-dd> <service-ref> <service-client> <port-info>
<security> <request-sender> <user>
```

Description This element configures name to be entered into UsernameToken.

Value Type string

```
(115) <jeus-web-dd> <service-ref> <service-client> <port-info>
<security> <request-sender> <timeToLive>
```

Description Available time period of message to be sent is configured as second unit. The default value is 300 seconds.

Value Type string

```
(116) <jeus-web-dd> <service-ref> <service-client> <port-info>
<security> <request-sender> <userNameToken>
```

Description This element configures UsernameToken.

Child Elements (117)passwordType?
(118)userTokenElements?

```
(117) <jeus-web-dd> <service-ref> <service-client> <port-info>
<security> <request-sender> <userNameToken> <passwordType>
```

Description This element configures password type to be used in UsernameToken. Either of "PasswordDigest" and "PasswordText" can be used.

Value Type passwordTypeType

Defined Value PasswordDigest

PasswordText

PasswordDigest

PasswordText

```
(118) <jeus-web-dd> <service-ref> <service-client> <port-info>
<security> <request-sender> <userNameToken> <userTokenElements>
```

Description This element is the list of elements to be added in UsernameToken. Each field is classified with blank. Either of "nonce" or "created" can be used. passwordType can be used when it is "PasswordText".

Value Type string

```
(119) <jeus-web-dd> <service-ref> <service-client> <port-info>
<security> <request-sender> <signature-infos>
```

Description This element is configuration for making signature on message.

Child Elements (120)signature-info+

```
(120) <jeus-web-dd> <service-ref> <service-client> <port-info>
<security> <request-sender> <signature-infos> <signature-info>
```

Description This is configuration for message signature. It enables to have multiple configurations.

Child Elements (121)signatureParts?
 (122)keyIdentifier
 (123)keystore

```
(121) <jeus-web-dd> <service-ref> <service-client> <port-info>
<security> <request-sender> <signature-infos> <signature-info>
<signatureParts>
```

Description This is used for signaturing certain part of message.
 "{ }{http://schemas.xmlsoap.org/soap/envelope/}Body; It can be arranged as mode such as "Token" If it has not been configured, it will write signaure for entire SOAP body.

Value Type string

```
(122) <jeus-web-dd> <service-ref> <service-client> <port-info>
<security> <request-sender> <signature-infos> <signature-info>
<keyIdentifier>
```

Description This element is mode expressing key information to be used in signature. One of IssuerSerial, DirectReference, SKIKeyIdentifier or X509KeyIdentifier is used.

Value Type sigKeyIdentifierType

Defined Value

IssuerSerial

DirectReference

SKIKeyIdentifier

X509KeyIdentifier

IssuerSerial

DirectReference

SKIKeyIdentifier

X509KeyIdentifier

```
(123) <jeus-web-dd> <service-ref> <service-client> <port-info>
<security> <request-sender> <signature-infos> <signature-info>
<keystore>
```

Description This element is configuration of keysotre which stores individual key for signature of message.

Child Elements

(124)key-type

(125)keystore-password

(126)keystore-filename

```
(124) <jeus-web-dd> <service-ref> <service-client> <port-info>
```

```
<security> <request-sender> <signature-infos> <signature-info>
<keystore> <key-type>
```

Description This element is key type storing in key store . (JKS or pkcs12)

Value Type string

```
(125) <jeus-web-dd> <service-ref> <service-client> <port-info>
<security> <request-sender> <signature-infos> <signature-info>
<keystore> <keystore-password>
```

Description This element is configuration for password for getting access to key store.

Value Type string

```
(126) <jeus-web-dd> <service-ref> <service-client> <port-info>
<security> <request-sender> <signature-infos> <signature-info>
<keystore> <keystore-filename>
```

Description This element is file name of key store. It writes file name or file name including absolute path including In case where only file name is written, it will be found in class path.

Value Type string

```
(127) <jeus-web-dd> <service-ref> <service-client> <port-info>
<security> <request-sender> <encryption-infos>
```

Description This element is configuration for encoding message.

Child Elements (128)encryption-info+

```
(128) <jeus-web-dd> <service-ref> <service-client> <port-info>
<security> <request-sender> <encryption-infos> <encryption-info>
```

Description This element checks configurations for Web Service Client to be arranged.

Child Elements (129)encryptionParts?
 (130)encryptionSymAlgorithm?

(131)encryptionUser?

(132)keyIdentifier

(133)keystore?

(137)embeddedKey?

```
(129) <jeus-web-dd> <service-ref> <service-client> <port-info>
<security> <request-sender> <encryption-infos> <encryption-info>
<encryptionParts>
```

Description This is configuration for encoding certain part. This is the format such as
 "{mode}{ns}{localname};{mode}{ns}{localname};...". Default mode
 value is content
 예: {Content}{http://example.org/payment}CreCard;{Element}{ }UserName

Value Type string

```
(130) <jeus-web-dd> <service-ref> <service-client> <port-info>
<security> <request-sender> <encryption-infos> <encryption-info>
<encryptionSymAlgorithm>
```

Description This element is algorithm used for encoding. It supports AES_128,
 AES_256, TRIPLE_DES and AES_192.

Value Type string

```
(131) <jeus-web-dd> <service-ref> <service-client> <port-info>
<security> <request-sender> <encryption-infos> <encryption-info>
<encryptionUser>
```

Description Name of key used for encoding.

Value Type string

```
(132) <jeus-web-dd> <service-ref> <service-client> <port-info>
<security> <request-sender> <encryption-infos> <encryption-info>
<keyIdentifier>
```

Description This element is mode of expressing information on key used for
 encoding. One of IssuerSerial, DirectReference, SKIKeyIdentifier,
 X509KeyIdentifier and EmbeddedKeyName is used.

<i>Value Type</i>	encKeyIdentifierType
<i>Defined Value</i>	IssuerSerial
	DirectReference
	SKIKeyIdentifier
	X509KeyIdentifier
	EmbeddedKeyName
	IssuerSerial
	DirectReference
	SKIKeyIdentifier
	X509KeyIdentifier
	EmbeddedKeyName

```
(133) <jeus-web-dd> <service-ref> <service-client> <port-info>
<security> <request-sender> <encryption-infos> <encryption-info>
<keystore>
```

Description This element is configuration for storage of key used for encoding.

Child Elements (134)key-type
 (135)keystore-password
 (136)keystore-filename

```
(134) <jeus-web-dd> <service-ref> <service-client> <port-info>
<security> <request-sender> <encryption-infos> <encryption-info>
<keystore> <key-type>
```

Description This element is key type storing in key store . (JKS or pkcs12)

Value Type string

```
(135) <jeus-web-dd> <service-ref> <service-client> <port-info>
<security> <request-sender> <encryption-infos> <encryption-info>
<keystore> <keystore-password>
```

Description This element is configuration for password for getting access to key store.

Value Type string

```
(136) <jeus-web-dd> <service-ref> <service-client> <port-info>
<security> <request-sender> <encryption-infos> <encryption-info>
<keystore> <keystore-filename>
```

Description This element is file name of key store. It writes file name or file name including absolute path including In case where only file name is written, it will be found in class path.

Value Type string

```
(137) <jeus-web-dd> <service-ref> <service-client> <port-info>
<security> <request-sender> <encryption-infos> <encryption-info>
<embeddedKey>
```

Description This element configures key shared by WebService and WebService Client. keyIdentifier can be used after being set as "EmbeddedKeyName".

Child Elements (138) embeddedKeyCallbackClass
(139) key-name

```
(138) <jeus-web-dd> <service-ref> <service-client> <port-info>
<security> <request-sender> <encryption-infos> <encryption-info>
<embeddedKey> <embeddedKeyCallbackClass>
```

Description In case of using session key, it configures callback class containing byte information.

Value Type string

```
(139) <jeus-web-dd> <service-ref> <service-client> <port-info>
<security> <request-sender> <encryption-infos> <encryption-info>
<embeddedKey> <key-name>
```

Description This element configures name of session key.

Value Type string

```
(140) <jeus-web-dd> <service-ref> <service-client> <port-info>
<security> <response-receiver>
```

Description In case where security is applied to Web Service response message, this is configuration for processing it.

Child Elements (141)action-list
 (142)password-callback-class?
 (143)timeToLive?
 (144)decryption?
 (149)signature-verification?

```
(141) <jeus-web-dd> <service-ref> <service-client> <port-info>
<security> <response-receiver> <action-list>
```

Description This element configures on any of security types applied to message Any of Timestamp, Encrypt, Signature and UsernameToken can be entered for this feild. Each field is classified by blank.(Ex)UsernameToken Signature Encrypt)

Value Type string

```
(142) <jeus-web-dd> <service-ref> <service-client> <port-info>
<security> <response-receiver> <password-callback-class>
```

Description Password Callback class name is entered as full package name.

Value Type string

```
(143) <jeus-web-dd> <service-ref> <service-client> <port-info>
<security> <response-receiver> <timeToLive>
```

Description This element configures available time period of recieved

message(Unit in second). The default value is 300 seconds from generated time.

Value Type string

```
(144) <jeus-web-dd> <service-ref> <service-client> <port-info>
<security> <response-receiver> <decryption>
```

Description This element is configuration for decoding encoded part of received message.

Child Elements (145)keystore

```
(145) <jeus-web-dd> <service-ref> <service-client> <port-info>
<security> <response-receiver> <decryption> <keystore>
```

Description This element is configuration on key store

Child Elements (146)key-type
(147)keystore-password
(148)keystore-filename

```
(146) <jeus-web-dd> <service-ref> <service-client> <port-info>
<security> <response-receiver> <decryption> <keystore> <key-
type>
```

Description This element is key type storing in key store . (JKS or pkcs12)

Value Type string

```
(147) <jeus-web-dd> <service-ref> <service-client> <port-info>
<security> <response-receiver> <decryption> <keystore>
<keystore-password>
```

Description This element is configuration for password for getting access to key store.

Value Type string

```
(148) <jeus-web-dd> <service-ref> <service-client> <port-info>
<security> <response-receiver> <decryption> <keystore>
```

<keystore-filename>

Description This element is file name of key store. It writes file name or file name including absolute path including In case where only file name is written, it will be found in class path.

Value Type string

```
(149) <jeus-web-dd> <service-ref> <service-client> <port-info>
<security> <response-receiver> <signature-verification>
```

Description This element is configuration for inspecting signature of recieved message.

Child Elements (150)keystore

```
(150) <jeus-web-dd> <service-ref> <service-client> <port-info>
<security> <response-receiver> <signature-verification>
<keystore>
```

Description This element is key store configuration for examining signature.

Child Elements (151)key-type
 (152)keystore-password
 (153)keystore-filename

```
(151) <jeus-web-dd> <service-ref> <service-client> <port-info>
<security> <response-receiver> <signature-verification>
<keystore> <key-type>
```

Description This element is key type storing in key store . (JKS or pkcs12)

Value Type string

```
(152) <jeus-web-dd> <service-ref> <service-client> <port-info>
<security> <response-receiver> <signature-verification>
<keystore> <keystore-password>
```

Description This element is configuration for password for getting access to key store.

Value Type string

```
(153) <jeus-web-dd> <service-ref> <service-client> <port-info>
<security> <response-receiver> <signature-verification>
<keystore> <keystore-filename>
```

Description This element is file name of key store. It writes file name or file name including absolute path including In case where only file name is written, it will be found in class path.

Value Type string

```
(154) <jeus-web-dd> <service-ref> <service-client> <service-impl-class>
```

Description Construction of service for Web Service Client is specified for this field. There is no need to set Web Service batcher manually since it is generated automatically.

Value Type token

```
(155) <jeus-web-dd> <service-ref> <service-client> <wsdl-override>
```

Description Location fo WSDL file which replaces <wsdl-file> of <service-ref> is for this field. Specified location should be available URL.

Value Type string

```
(156) <jeus-web-dd> <service-ref> <service-client> <service-qname>
```

Description WSDL service of WSDL is specified for this field.

Value Type QName

```
(157) <jeus-web-dd> <service-ref> <service-client> <call-property>
```

Description Properties set for entire javax.xml.rpc.Call object used in WSDL service are specified for this field

<i>Child Elements</i>	(158)name
	(159)value

```
(158) <jeus-web-dd> <service-ref> <service-client> <call-  
property> <name>
```

<i>Description</i>	The key name for setting property on javax.xml.rpc.Call or javax.xml.rpc.Stub is specified for this field.
--------------------	--

<i>Value Type</i>	string
-------------------	--------

```
(159) <jeus-web-dd> <service-ref> <service-client> <call-  
property> <value>
```

<i>Description</i>	The value corresponds to the key used for setting property on <code>javax.xml.rpc.Stub</code> or <code>javax.xml.rpc.Call</code> .
--------------------	--

<i>Value Type</i>	string
-------------------	--------

```
(160) <jeus-web-dd> <url-rewriting>
```

Description

By default, the Web container will use cookies to maintain the client's session id over several requests. One problem is that most Web browsers do not send any session ID cookie information with a new request if that request has a different domain name than the domain name for the request where the cookie was originally created. This option (url-rewriting), if set to "true", will force the use of URL rewriting instead of relying on cookies (even if the Web browser supports cookies). This way, session tracking will work even if different domain names are used across several calls. This will cause all the URL's returned by the context to have a unique JSESSIONID URL parameter attached. If set to "false", this feature will not be used and the default cookie-based behavior will be used.

<i>Value Description</i>	Activating or inactivating this configuration as Boolean value.
--------------------------	---

<i>Value Type</i>	boolean
-------------------	---------

<i>Value Type Description</i>	Boolean
-------------------------------	---------

<i>Default Value</i>	false
<i>Defined Value</i>	true Activating the related configuration.
	false Inactivating the related configuration.

(161) <jeus-web-dd> **<enable-default-login>**

<i>Description</i>	Defines the availability of using LoginFilter provided by JEUS WebContainer.
<i>Value Type</i>	boolean
<i>Default Value</i>	false

(162) <jeus-web-dd> **<jsp-engine>**

<i>Description</i>	This element makes JSP-related settings for this context. The settings will apply to all JSP pages contained in this context.
<i>Child Elements</i>	(163)keep-generated? (164)java-compiler? (165)jsp-work-dir? (166)compile-output-dir? (167)compile-option? (168)compile-encoding? (169)check-included-jspfile?

(163) <jeus-web-dd> <jsp-engine> **<keep-generated>**

<i>Description</i>	This element determines availability of storing java source file after generating Servlet Class File through compiling source file generated from JSP page. These files are useful for debugging purpose.
<i>Value Description</i>	This configuration activates or inactivates as Boolean value.
<i>Value Type</i>	boolean

Value Type Description Boolean

Default Value true

(164) <jeus-web-dd> <jsp-engine> **<java-compiler>**

Description This element configures Java Compiler for compiling Java source code of eJSP. It is not recommended since default configuration is the most efficient way.

Value Description Character set and compiler configuring java compiler should be included within system path in order to execute.

Value Type token

Default Value com.sun.tools.javac

Defined Value sun.tools.javac
SUN Microsystems Java Compiler in which "tools.jar" is included is used (This configuration is recommended).

javac

This element configures external "javac" compiler including in system path.

com.sun.tools.javac

This is operated in serveral systems. If this operates, this is priotized option since it is more efficient and stable than sun.tools.javac.

(165) <jeus-web-dd> <jsp-engine> **<jsp-work-dir>**

Description JSP workdir is location in which Servlet source file and class files are stored. Developer or administrator can use related directory for debugging purpose. This element is not generally needed to be used.

Value Description Absolute path used for compiling JSP.

Value Type token

(166) <jeus-web-dd> <jsp-engine> **<compile-output-dir>**

Description Through configuring this property, generated class files can be placed in other directory besides <jsp-work-dir> directory. If this property is not set, class files will be located in jspwork directory. This element is not commonly used.

Value Description Absolute path used for storing JSP class files.

Value Type token

(167) <jeus-web-dd> <jsp-engine> **<compile-option>**

Description This is option used as Servlet compiler and is not commonly used.

Value Description Compile option property used for compiling (Referencing <java-compiler>element)

Value Type token

(168) <jeus-web-dd> <jsp-engine> **<compile-encoding>**

Description This property is value of configuring in ?encoding option when compiling generated Servlet source file through parsing JSP files. This configuration is not commonly used. Web Container automatically selects proper configuration.

Value Description Character set checking encoding scheme to use. For additional information, please refer to javac compiler documentation.

Value Type token

Defined Value default
This is automatically selected in Web Container.

(169) <jeus-web-dd> <jsp-engine> **<check-included-jspfile>**

Description If this property is set as "true", it enables to recompile related JSP file when executing included JSP file. Included JSP files are not

checked for being updated.

Value Description Activating or inactivating this configuration as Boolean value.

Value Type boolean

Value Type Description Boolean

Default Value false

Defined Value true
 Activating related configuration.

false
 Inactivating related configuration.

(170) <jeus-web-dd> **<keep-generated>**

Description This element determines whether the .java source files that are generated from the JSP pages should be preserved or removed after compiling them into Servlet class files. Keeping these files is useful for debugging purposes.

Value Type boolean

(171) <jeus-web-dd> **<fast-deploy>**

Description When this web component includes Web Service or Web Service Client, This will set the availabiltiy of using class file installed with web component when it is not compiled by generating Stub of Tie and Web Service Client of Web Service.

Value Type boolean

(172) <jeus-web-dd> **<servlet>**

Description This element contains information on servlet of jeus within context.

Child Elements (173)servlet-name

(174)run-as-identity?

(173) <jeus-web-dd> <servlet> **<servlet-name>**

Description This element refers servlet-name configured in web.xml.

Value Type token

(174) <jeus-web-dd> <servlet> **<run-as-identity>**

Description This element defines mapping of run-as-specified-identity role name defined in web.xml as actual user name(principal)

Child Elements (175)principal-name

(175) <jeus-web-dd> <servlet> <run-as-identity> **<principal-name>**

Description Principal name used as role of run-as-specified-identity within ejb-jar.xml or web.xml.

Value Description Principal name configured in subjects.xml of security.

Value Type token

(176) <jeus-web-dd> **<webinf-first>**

Description class 를 로딩할 때 web-inf 디렉토리 아래에서 먼저 찾을 것인지의 여부를 설정한다. true 로 설정하면 web-inf 아래에서 먼저 찾고 false 로 설정되면 상위 classloader 에서 먼저 찾는다. true 로 설정시 중복된 class 로 인한 ClassCastException 을 주의하여야 한다.

Value Type boolean

Default Value false

E.4 jeus-web-dd.xml 파일 예제

```
<<jeus-web-dd.xml>>

<?xml version="1.0"?>
<jeus-web-dd xmlns="http://www.tmaxsoft.com/xml/ns/jeus">
  <context-path>/examples</context-path>
  <docbase>examples</docbase>
  <logging>
    <user-log>
      <handler>
        <file-handler>
          <name>userHandler</name>
          <valid-day>7</valid-day>
          <file-name>c:\MyContextLog.log</file-name>
        </file-handler>
      <handler>
    </user-log>
  </logging>
  <enable-jsp>true</enable-jsp>
  <auto-reload>
    <enable-reload>true</enable-reload>
    <check-on-demand>true</check-on-demand>
  </auto-reload>
  <max-instance-pool-size>20</max-instance-pool-size>
  <added-classpath>
    <class-path>c:\mylib\subdir\</class-path>
  </added-classpath>
  <allow-indexing>
    <directory>/images/</directory>
  </allow-indexing>
  <deny-download>
    <file>/data/secret.dat</file>
    <extension>dat</extension>
    <directory>/data/</directory>
  </deny-download>
  <aliasing>
    <alias>
      <alias-name>/images/</alias-name>
```

```
<real-path>c:\web\images\</real-path>
</alias>
</aliasing>
<file-caching>
  <max-idle-time>1800000</max-idle-time>
  <max-cache-memory>10</max-cache-memory>
  <directory>/images/</directory>
</file-caching>
<role-mapping>
  <role-permission>
    <role>administrator</role>
    <principal>peter</principal>
  </role-permission>
</role-mapping>
<ejb-ref>
  <jndi-info>
    <ref-name>ejb/account</ref-name>
    <export-name>AccountEJB</export-name>
  </jndi-info>
</ejb-ref>
<res-ref>
  <jndi-info>
    <ref-name>jdbc/EmployeeAppDB</ref-name>
    <export-name>EmployeeDB</export-name>
  </jndi-info>
</res-ref>
<res-env-ref>
  <jndi-info>
    <ref-name>jms/StockQueue</ref-name>
    <export-name>StockQueue</export-name>
  </jndi-info>
</res-env-ref>
<url-rewriting>true</url-rewriting>
<jsp-engine>
  <keep-generated>true</keep-generated>
  <java-compiler>javac</java-compiler>
  <jsp-work-dir>c:\MyJSPWorkdir\</jsp-work-dir>
  <compile-output-dir>c:\MyJSPWorkdir\</compile-
output-dir>
```

```
        <compile-option>-g:none -verbose</compile-option>
        <compile-encoding>8859_1</compile-encoding>
        <check-included-jspfile>true</check-included-
jspfile>
        </jsp-engine>
</jeus-web-dd>
```


F TCP Listener API

F.1 소개

이 부록은 TCP listener 를 위해 커스텀 TCP 핸들러를 실현하는데 사용되어지는 TCPDispatcherConfig 와 TCPServlet 클래스의 내용을 나열한다.

해당 API 와 TCP listener 를 어떻게 사용할 지에 대한 자세한 정보는 7 장을 참조한다.

F.2 jeus.servlet.tcp 패키지

abstract class TCPDispatcherConfig

```
public abstract class TCPDispatcherConfig extends  
java.lang.Object
```

이 추상클래스는 TCP 클라이언트로부터 들어오는 요청 패킷에 대한 몇 가지 정보를 제공한다. JEUS Web Container 의 TCP 클라이언트와 TCP listener 사이의 프로토콜은 해당 클래스를 상속하여 커스터마이즈 할 수 있고 WEBMain.xml 에 <dispatcher-config-class>에 해당 클래스를 등록한다.

Constructors

```
public TCPDispatcherConfig()
```

Description

- 디폴트 생성자.

Methods

```
public abstract int getBodyLength(byte[] header)
```

Description

- 이 메소드는 TCP listener 에 포함되며 들어오는 요청 패킷에서 요청바디의 길이를 돌려준다(바이트).

Arguments

- `byte[] header`: 헤더를 구성하는 바이트 (TCP listener 에 의해 제공된다).

Return value

- `int`: 요청 바디의 바이트수.

```
public abstract java.lang.String getContextPath(byte[] header)
```

Description

- 이 메소드는 TCP listener 에 포함되며 TCP 기본 메시지를 다루는데 사용되는 Servlet Context 의 Context 의 경로를 돌려준다. 반드시 “/”로 시작한다.

Arguments

- `byte[] header`: 헤더를 구성하는 바이트 (TCP listener 에 의해 제공된다).

Return value

- `java.lang.String`: “/”로 시작하는 context.

```
public abstract int getHeaderLength()
```

Description

- 해당 메소드는 TCP listener 에 포함되며 이행된 프로토콜 헤더의 고정길이를 돌려준다. 그 값은 공간을 할당하고 새로운 메시지가 TCP listener 에 받을 때 헤더의 길이를 결정하는데 사용된다.

Return value

- `int`: 프로토콜 헤더의 고정길이.

```
public abstract java.lang.String getPathInfo(byte[] header)
```

Description

- 이 메소드는 TCP listener 에 포함되며 제공한 헤더로부터 어떤 extra 경로 정보를 돌려준다.

Arguments

- byte[] header: 헤더를 구성하는 바이트 (TCP listener 에 의해 제공된다).

Return value

- java.lang.String: 헤더로부터 추출된 extra 경로정보. null 로 반환될 수 있다.

```
public abstract java.lang.String getServletPath(byte[] header)
```

Description

- 이 메소드는 TCP listener 에 포함되며 들어오는 요청을 다루는 Servlet 의 이름 또는 경로를 돌려준다. 해당 Servlet 은 jeus.servlet.tcp.TCPServlet 의 하위 클래스이고 getContextPath()메소드에서 리턴하는 Context 경로를 web.xml 에 등록해야 한다.

Arguments

- byte[] header: 헤더를 구성하는 바이트 (TCP listener 에 의해 제공된다).

Return value

- java.lang.String: TCP Servlet 의 경로.

```
public abstract java.lang.String getSessionId(byte[] header)
```

Description

- 이 메소드는 TCP listener 에 포함되며 헤더에 임베디드된 세션 ID 를 돌려준다.

Arguments

- byte[] header: 헤더를 구성하는 바이트 (TCP listener 에 의해 제공된다).

Return value

- java.lang.String: 만일 세션 ID 가 헤더에 있을 때 세션 ID 또는 null 값.

```
public abstract void init()
```

Description

- 이 메소드는 코드를 초기화한다. Web Container 가 부팅할 때 한번만 호출될 것이다.

```
public boolean isPersistentConnection()
```

Description

- 이 메소드는 TCP listener 에 포함되며 TCP listener 에 소켓 커넥션이 drop 하거나 TCP 핸들러가 TCP 클라이언트의 요청 처리가 완료될 때의 논리값을 돌려준다.

Return value

- boolean: 만일 커넥션이 유지되려면 “true”(Telnet 프로토콜과 같다)로 설정하고 만일 커넥션이 각 요청/응답 후에 close 하려면 “false”로 설정한다(HTTP 프로토콜과 같다).

abstract class TCPServlet

```
public abstract class TCPServlet extends  
javax.servlet.GenericServlet
```

이 추상클래스는 TCP 핸들러의 슈퍼클래스로 개발자는 반드시 상속해야 한다. TCP 핸들러는 커스텀 프로토콜과 메시지를 다루는 TCP listener 로부터 요청을 받는다.

이 상속 클래스/핸들러는 web.xml과 JEUS Web Context에 일반 Servlet으로 등록해야 한다. 보다 많은 정보는 9 장을 참조한다.

Constructors

```
public TCPServlet()
```

Description

- 디폴트 생성자.

Methods

```
public abstract void service(TCPServletRequest req,
TCPServletResponse res)
```

Description

- TCP listener 는 TCP 클라이언트로부터 그것의 포트에 새로운 메시지가 올 때 해당 메소드를 호출한다. 메소드는 통신에 사용되는 커스텀 프로토콜에 따라 implementation-specific 으로 메시지를 다뤄야 한다. 요청 메시지는 req 레퍼런스에 있고 res 레퍼런스는 응답을 쓸 때 사용된다.

Arguments

- TCPServletRequest req: 요청메시지는 헤더와 바디를 포함한다. 이 요청은 TCP 클라이언트로부터 TCP listener 를 통해 받아지고 service() 메소드로 포워드된다. TCPServletRequest 클래스는 다음과 같이 선언한다: public abstract class **TCPServletRequest** extends javax.servlet.http.HttpServletRequest (아래 참조).
- TCPServletResponse res: service()메소드는 이 객체의 요청에 그것의 응답을 기록해야 한다(output-stream 을 사용한다). TCPServletResponse 클래스는 다음과 같이 선언한다: public class **TCPServletResponse** extends javax.servlet.http.HttpServletResponse (아래 참조).

abstract class TCPServletRequest

```
public abstract class TCPServletRequest extends
javax.servlet.http.HttpServletRequest
```

Methods

```
public byte[] getHeader()
```

Description

- 들어오는 요청패킷의 헤더 부분을 돌려준다. 헤더의 길이는 TCPDispatcherConfig.getHeaderLength()의 결과이다.

Return value

- `byte[]`: 들어오는 요청의 헤더.

```
public byte[] getBody()
```

Description

- 들어오는 요청패킷의 바디부분을 돌려준다. 바디의 길이는 `TCPDispatcherConfig.getBodyLength()`의 결과이다.

Return value

- `byte[]`: 들어오는 요청의 바디.

abstract class TCPServletResponse

```
public class TCPServletResponse extends  
    javax.servlet.http.HttpServletResponse
```

Methods

```
public void closeOnExit()
```

Description

- 요청이 완료된 후 TCP 클라이언트와 TCP listener 사이에 물리적인 커넥션을 종료한다. `TCPDispatcherConfig.isPersistentConnection()`의 값이 `true` 일 때 해당 메소드는 사용된다. 그러나 개발자는 최근 클라이언트 커넥션을 끊어버리는 것을 원한다.

G JEUS WebContainer Properties

G.1 소개

JEUS WebContainer 에는 기타 필요한 세팅 사항을 프로퍼티로 받을 수 있도록 하고 있다.

이 Java 환경 변수는 “-D” 로 전환한 후 덧붙여지고 사용 값은 “=” 문자 뒤에 덧붙여진다. 다음은 예이다:

```
-Djeus.servlet.jsp.reload=false
```

상기 샘플이 JEUSMain.xml 에 container 에 대한 command-option element 에 정의되어 jsp 에 대한 reload 기능을 disable 할 수 있다.

참고: WebContainer 에 정의하고자 하는 속성들을 JEUS_HOME/config/<nodename>/nodename_servlet_engine 폴더 하위에 engine.properties 생성하여 정의하여도 일괄적으로 WebContainer 에 해당 속성이 적용될 수 있다.

G.2 환경 변수 참조

표 4. JEUS WebContainer Properties.

WebContainer Properties	의미	Default 값 (또는 예)
jeus.servlet.jsp.reload	WebContainer 에서 jsp resource 에 대한 reload 기능 적용 여부를 지정한다.	True
jeus.servlet.jsp.noPackagePrefix	jsp 컴파일시에 디폴트 패키지로 구성된 bean 을 사용할 경우 해당 옵션을 true 로 설정해야 한다.	false
jeus.sevlet.jsp.jspParam.urlEncode	jsp 에서 forward 나 include 를 사용할 때 url 인코딩 여부를 적용한다.	false
jeus.servlet.classload.spec23	서블릿 스펙 2.3 을 지원하는 클래스 로더의 적용 여부를 지정한다.	false
jeus.servlet.webadmin.suspendCmd.timeout	Webtob listener 사용시 webadmin 에서 suspend 명령을 내렸을 때 thread 가 suspend 상태로 변하는데 기다리는 시간의 timeout	120 * 1000 (ms)
jeus.servlet.request.url.allowDotDot	“/abc/./../”와 같은 형태의 url 로 요청이 오는 경우 보안 문제 관련하여 상대 경로 표시를 무시 여부를 지정한다.	false

WebContainer Properties	의미	Default 값 (또는 예)
jeus.servlet.response.header. encoding	Buffer 로부터 응답헤더의 내용을 읽을 때 Content-Type 의 charset 값 과 다른 사용할 encoding 값 설정 한다.	N/A
jeus.servlet.request.cookie. encoding	기본적으로 request encoding 값을 따르나 request encoding 값과 다른 encoding 값을 적용하고 싶을 때 설정한다.	N/A
jeus.sevlet.pathinfo. ignoreDoublePath	http://localhost/examples/////context 라는 request url을 /examples/context처럼 수행하기 위 한 여부 값을 설정한다.	false
jeus.servlet.jsp.compile. retrycount	jsp 를 compile 하고 로딩할 때 java.lang.ClasFormatError, java.lang.VerifyError 가 발생하는 경우에 재컴파일 시도를 하는 횟 수값을 설정한다.	2

JEUS Properties	WebContainer 의미	Default 값 (또는 예)
jeus.servlet.jsp.jspwriter.null	jsp writer 에 string 내용이 아무것도 없는 경우 default string 값을 설정한다.(jeus.webcontainer.j2ee.compatible=true 이면 string 값은 무조건 null 임)	""
jeus.servlet.urldecode.cookie	cookie 값의 내용이 encoding 이 되었을 때 자동적으로 decoding 해주는 옵션값 설정.	false
jeus.servlet.request.enableDNS	Jeus 가 운용하는 시스템의 DNS Server 가 응답 속도가 느릴 경우 javax.servlet.http.HttpServletRequest.getRemoteHost() 수행에 대한 응답이 느려질 수 있으므로 해당 옵션을 false 로 하면 getRemoteAddr()를 대신 수행하여 DNS Server 의 응답속도를 줄일 수 있음.	true
jeus.servlet.response.header.serverInfo	Response header 의 기본으로 포함되어 있는 "Server" 정보 속성의 포함 여부를 설정한다.	true
jeus.servlet.welcomefile.sendRedirect	Welcome file 로 등록된 resource 를 수행할 때 sendRedirect 로 수행할지의 여부를 설정한다. false 값으로 설정하면 forward 로 수행되어 request header 값을 그대로 유지할 수 있다.	true
jeus.servlet.context.attribute.serialize	Context attribute 에 대해서 serialize 해서 저장할 것인지의 여부를 설정한다.	true

색인

__DEFAULT_CONTEXT__ 210, 232

가상 호스트 ID 이름 228

기본 가상 호스트 230, 235

네트워크 이름 228, 235

로깅 95

리스너 113

리스너 튜닝 173

보안 리스너 118

보안 스위치 51

웹 어플리케이션 21, 200

인코딩 92

자동 모니터링 50, 56

A

AJP13 Listener 119

Apache 리스너 118

ApJServBalance 198

ApJServHost 198

ApJServMount 198

ApJServRoute 198

ApJServShmFile 198

auto reload 210

C

Context 21, 200

Context Group 88

Context Group 디렉토리 구조 97

Context Group 모니터링 110

Context Group 설정 99

Context Group 제어 108

Context Group 튜닝 111

Context Group Architecture 89

context name 210, 217

context path 210, 217

D

Database Connection Pool 74

DBA Override 77

Default virtual host 230

Deployment Descriptor 209

DNS 232

document base directory 210

H

hosts	232
HTTP 리스너	117

J

JEUS 시스템	38
JEUS_HOME	44
jeusadmin	42
jeus-web-dd.xml	210
JServ	135
JSP Engine	94
JSP switch	211

M

maximum instance pool size	211
Mixed Mode	188
Multi Session Mode	189

N

Non-shared Mode	76
-----------------------	----

R

Reload	223
Request prefetch	128
Request queuing	129
Response Header	96
Response Header 설정	106
Restart	223

S

Session Tracking	176
Session Tracking 설정	191

Session Tracking 튜닝	198
Shared Mode	76
Shared Session Data	190
Suspend	223

T

TCP 리스너	118
TCP 클라이언트	147
TCP 핸들러	147
Terminate	223
Tmax Listener	119

U

UDP 리스너	118
Undeploy	223
URL Rewriting	190

W

WAS	27
Web 디렉토리 구조	42
Web Container	37
Web Container 구조	48
Web Container 모니터링	71
Web Container Tuning	72
Web context 등록	208
Web Context 모니터링	224
Web Context 요청	220
Web Context 제어	222
Web Context 튜닝	225
WEBMain.xml	56
WebtoB 리스너	118
Worker 쓰레드 풀	114