



Sun Java™ System

适用于 UNIX 的 Sun Java Enterprise System 5 升级指南

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

文件号码 820-0890
2007 年 6 月

版权所有 © 2007 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. 保留所有权利。

对于本文中介绍的产品，Sun Microsystems, Inc. 对其所涉及的技术拥有相关的知识产权。需特别指出的是（但不局限于此），这些知识产权可能包含在 <http://www.sun.com/patents> 中列出的一项或多项美国专利，以及在美国和其他国家 / 地区申请的一项或多项其他专利或待批专利。

本产品包含 SUN MICROSYSTEMS, INC. 的机密信息和商业秘密。未经 SUN MICROSYSTEMS, INC. 的事先明确书面许可，不得使用、泄露或复制。美国政府权利 - 商业软件。政府用户应遵循 Sun Microsystems, Inc. 的标准许可协议，以及 FAR（Federal Acquisition Regulations，即“联邦政府采购法规”）的适用条款及其补充条款。

本发行版可能包含由第三方开发的内容。

本产品的某些部分可能是从 Berkeley BSD 系统衍生出来的，并获得了加利福尼亚大学的许可。UNIX 是 X/Open Company, Ltd. 在美国和其他国家 / 地区独家许可的注册商标。

Sun、Sun Microsystems、Sun 徽标、Java、Solaris、JDK、Java Naming and Directory Interface、JavaMail、JavaHelp、J2SE、iPlanet、Duke 徽标、Java 咖啡杯徽标、Solaris 徽标、SunTone Certified 徽标和 Sun ONE 徽标是 Sun Microsystems, Inc. 在美国和其他国家 / 地区的商标或注册商标。

所有的 SPARC 商标的使用均已获得许可，它们是 SPARC International, Inc. 在美国和其他国家 / 地区的商标或注册商标。标有 SPARC 商标的产品均基于由 Sun Microsystems, Inc. 开发的体系结构。

Legato 和 Legato 徽标是注册商标，它们和 Legato NetWorker 都是 Legato Systems, Inc. 的商标或注册商标。Netscape Communications Corp 徽标是 Netscape Communications Corporation 的商标或注册商标。

OPEN LOOK 和 Sun(TM) 图形用户界面是 Sun Microsystems, Inc. 为其用户和许可证持有者开发的。Sun 感谢 Xerox 在研究和开发可视或图形用户界面的概念方面为计算机行业所做的开拓性贡献。Sun 已从 Xerox 获得了对 Xerox 图形用户界面的非独占性许可证，该许可证还适用于实现 OPEN LOOK GUI 和在其他方面遵守 Sun 书面许可协议的 Sun 许可证持有者。

本服务手册所介绍的产品以及所包含的信息受美国出口控制法制约，并应遵守其他国家 / 地区的进出口法律。严禁将本产品直接或间接地用于核设施、导弹、生化武器或海上核设施，也不能直接或间接地出口给核设施、导弹、生化武器或海上核设施的最终用户。严禁出口或转口到美国禁运的国家 / 地区以及美国禁止出口清单中所包含的实体，包括但不限于被禁止的个人以及特别指定的国家 / 地区的公民。

本文档按“原样”提供，对于所有明示或默示的条件、陈述和担保，包括对适销性、适用性或非侵权性的默示保证，均不承担任何责任，除非此免责声明的适用范围在法律上无效。

目录

- 表格列表 15**
- 前言 19**
 - 目标读者 20
 - 本书所采用的约定 20
 - 管理界面 20
 - 印刷约定 21
 - 符号 21
 - Shell 提示符 22
 - 相关文档 22
 - Java ES 文档集中的书籍 22
 - 搜索 Sun 产品文档 24
 - 联机访问 Sun 资源 24
 - 联系 Sun 技术支持 24
 - 第三方 Web 站点引用 25
 - Sun 欢迎您提出意见 25
- 第 1 章 升级规划 27**
 - Java ES 5 组件 28
 - 发行版本 5 产品组件 28
 - 发行版本 5 共享组件 29
 - Java ES 升级技术 31
 - 产品组件升级 31
 - 产品组件升级方法 31
 - 各产品组件所用的升级方法 33
 - 共享组件升级 34
 - 升级过程 35

升级计划注意事项	36
升级依赖性	36
所支持的升级方式和策略	37
选择性升级或全部升级	39
多实例升级	40
操作系统注意事项	40
所需的操作系统修补程序	40
双重升级: Java ES 和操作系统软件	41
操作系统升级	42
Solaris 10 多区域环境	42
Java ES 组件依赖性	43
对共享组件的依赖性	43
对产品组件的依赖性	46
升级先后顺序指导原则	50
特殊情况	53
选择性升级: Application Server 未升级	53
将 Portal Server 过渡功能版本 (IFR) 7.0 升级到 Java ES 5	53
Java ES 5 升级与 Solaris 10 区域	54
Java ES 安装程序中的区域支持	54
产品组件的升级	54
同步所有共享组件	55
建议的升级做法	55
特殊情况或例外情况	56
产品组件特殊情况	56
共享组件特殊情况	57
 第 2 章 升级 Java ES 共享组件	 59
共享组件升级概述	60
一般注意事项	60
同步共享组件	60
同步所有共享组件	61
Solaris 10 区域注意事项	62
升级共享组件的方法	62
共享组件升级过程	63
特殊情况升级过程	64
Java SE 升级过程	64
检查 Java SE 符号链接	65
检验当前 Java SE 版本	66
Common Agent Container 升级过程	66

第 3 章 Sun Cluster 软件	69
Sun Cluster 软件升级概述	70
关于发行版本 5 Sun Cluster 软件	70
Sun Cluster 软件升级指导说明	71
Sun Cluster 数据	71
Sun Cluster 升级策略	72
兼容性问题	72
Sun Cluster 组件依赖性	72
双重升级	72
将 Sun Cluster 软件升级到 Java ES 发行版本 5	73
简介	73
Sun Cluster 升级	74
升级前的任务	74
升级 Sun Cluster 软件	77
检验升级	78
升级后的任务	78
升级回滚	78
 第 4 章 Sun Cluster Geographic Edition	 79
Sun Cluster Geographic Edition 升级概述	80
关于 Java ES 发行版本 5	80
Java ES 发行版本 5 升级指导说明	80
Sun Cluster Geographic Edition 数据	80
Sun Cluster Geographic Edition 升级策略	81
兼容性问题	81
组件依赖性	81
双重升级	81
从版本 3.1 8/05 升级 Sun Cluster Geographic Edition	82
简介	82
版本 3.1 8/05 Sun Cluster Geographic Edition 升级	83
升级前的任务	83
升级 3.1 8/05 Sun Cluster Geographic Edition	85
检验升级	87
升级后的任务	87
升级回滚	87

第 5 章 Directory Server	89
Directory Server 升级概述	90
关于 Java ES 发行版本 5	90
Java ES 发行版本 5 升级指导说明	90
Directory Server 数据	91
Directory Server 升级策略	92
兼容性问题	92
组件依赖性	92
双重升级	93
从 Java ES 发行版本 4 升级 Directory Server	94
简介	94
发行版本 4 Directory Server 的升级	95
升级前的任务	95
升级发行版本 4 Directory Server	96
检验升级	100
升级后的任务	101
升级回滚	101
多实例升级	101
Directory Server 复制的滚动升级	101
作为数据服务升级 Directory Server	101
从 Java ES 发行版本 3 升级 Directory Server	102
从 Java ES 发行版本 2 升级 Directory Server	103
 第 6 章 Directory Proxy Server	 105
Directory Proxy Server 升级概述	106
关于 Java ES 发行版本 5	106
Java ES 发行版本 5 升级指导说明	106
Directory Proxy Server 数据	107
Directory Proxy Server 升级策略	108
兼容性问题	108
组件依赖性	108
双重升级	109
从 Java ES 发行版本 4 升级 Directory Proxy Server	110
简介	110
发行版本 4 Directory Proxy Server 的升级	111
升级前的任务	111
升级发行版本 4 Directory Proxy Server	112
检验升级	114
升级后的任务	114
升级回滚	115
多实例升级	115
从 Java ES 发行版本 3 升级 Directory Proxy Server	116
从 Java ES 发行版本 2 升级 Directory Proxy Server	117

第 7 章 Web Server	119
Web Server 升级概述	120
关于 Java ES 发行版本 5 Web Server	120
Web Server 升级指导说明	120
Web Server 数据	121
Web Server 升级策略	121
兼容性问题	122
Web Server 组件依赖性	122
双重升级	122
从 Java ES 发行版本 4 升级 Web Server	123
简介	123
发行版本 4 Web Server 的升级	124
升级前的任务	124
升级发行版本 4 Web Server	125
检验升级	134
升级后的任务	134
升级回滚	138
从 Java ES 发行版本 3 升级 Web Server	139
从 Java ES 发行版本 2 升级 Web Server	140
 第 8 章 Java DB	 141
Java DB 升级概述	142
关于 Java ES 发行版本 5	142
Java ES 发行版本 5 升级指导说明	142
Java DB 数据	143
Java DB 升级策略	143
兼容性问题	143
组件依赖性	143
双重升级	144
从 Java ES 发行版本 4 升级 Java DB	145
简介	145
发行版本 4 Java DB 的升级	146
升级前的任务	146
升级发行版本 4 Java DB	147
检验升级	148
升级后的任务	149
升级回滚	149
多实例升级	149

第 9 章 High Availability Session Store	151
HADB 升级概述	152
关于 Java ES 发行版本 5 HADB	152
HADB 升级指导说明	152
HADB 数据	153
HADB 升级策略	153
兼容性问题	153
HADB 组件依赖性	153
双重升级	153
从 Java ES 发行版本 4 升级 HADB	154
简介	154
发行版本 4 HADB 的升级	155
升级前的任务	155
升级发行版本 4 HADB	156
检验升级	158
升级后的任务	159
升级回滚	159
从 Java ES 发行版本 3 升级 HADB	160
 第 10 章 Message Queue	 161
Message Queue 升级概述	162
关于 Java ES 发行版本 5 Message Queue	162
Message Queue 升级指导说明	163
Message Queue 数据	164
Message Queue 升级策略	165
兼容性问题	165
Message Queue 组件依赖性	165
双重升级	166
从 Java ES 发行版本 4 升级 Message Queue	167
简介	167
发行版本 4 Message Queue 的升级	168
升级前的任务	168
升级发行版本 4 Message Queue	169
检验 Message Queue 升级	170
升级后的任务	170
升级回滚	171
多实例升级	172
从 Java ES 发行版本 3 升级 Message Queue	173
从 Java ES 发行版本 2 升级 Message Queue	174
简介	174

发行版本 2 兼容性问题	175
协议兼容性	175
代理兼容性	175
受管对象兼容性	176
管理工具兼容性	176
客户机兼容性	176
发行版本 2 Message Queue 的升级	177
升级前的任务	177
升级发行版本 2 Message Queue (Solaris)	178
升级发行版本 2 Message Queue (Linux)	179
安装兼容性软件包 (Linux)	181
检验 Message Queue 升级	181
升级后的任务	181
升级回滚	182
多实例升级	182
 第 11 章 Application Server	183
Application Server 升级概述	185
关于 Java ES 发行版本 5 Application Server	185
Application Server 升级指导说明	185
Application Server 数据	188
Application Server 升级策略	188
兼容性问题	189
Application Server 组件依赖性	189
双重升级	190
从 Java ES 发行版本 4 升级 Application Server	191
简介	191
发行版本 4 Application Server 的升级	192
升级前的任务	192
升级发行版本 4 Application Server	194
检验升级	196
升级后的任务	196
升级回滚	196
从 Java ES 发行版本 3 升级 Application Server	197
从 Java ES 发行版本 2 升级 Application Server	198
简介	198
发行版本 2 Application Server 的升级	199
升级前的任务	199
升级发行版本 2 Application Server	202
检验升级	206
升级后的任务	206
升级回滚	206
在 Solaris 10 多区域环境中升级 Solaris 捆绑的 Application Server	207

第 12 章 Service Registry	209
Service Registry 升级概述	210
关于 Java ES 发行版本 5	210
Java ES 发行版本 5 升级指导说明	210
Service Registry 数据	211
Service Registry 升级策略	211
兼容性问题	211
组件依赖性	211
双重升级	212
从 Java ES 发行版本 4 升级 Service Registry	213
简介	213
发行版本 4 Service Registry 的升级	214
升级前的任务	214
升级发行版本 4 Service Registry	216
检验升级	218
升级后的任务	218
升级回滚	219
多实例升级	220
 第 13 章 Web Proxy Server	 221
Web Proxy Server 升级概述	222
关于 Java ES 发行版本 5 Web Proxy Server	222
Web Proxy Server 升级指导说明	222
Web Proxy Server 数据	223
Web Proxy Server 升级策略	223
兼容性问题	223
Web Proxy Server 组件依赖性	224
双重升级	224
从 Java ES 发行版本 4 升级 Web Proxy Server	225
简介	225
发行版本 4 Web Proxy Server 的升级	226
升级前的任务	226
升级发行版本 4 Web Proxy Server (Solaris)	227
升级发行版本 4 Web Proxy Server (Linux)	228
检验升级	230
升级后的任务	230
升级回滚 (Solaris)	230
从版本 3.6 升级 Web Proxy Server	231
简介	231
版本 3.6 Web Proxy Server 升级	232
升级前的任务	232
升级版本 3.6 Web Proxy Server	233
检验升级	234
升级后的任务	234
版本 3.6 的升级回滚	234

第 14 章 Access Manager	235
Access Manager 升级概述	236
关于 Java ES 发行版本 5 Access Manager	236
Access Manager 升级指导说明	237
Access Manager 数据	238
Access Manager 升级策略	239
兼容性问题	239
Access Manager 组件依赖性	240
Web 容器升级方案	241
双重升级	241
从 Java ES 发行版本 4 升级 Access Manager	242
简介	242
发行版本 4 Access Manager 的全升级	243
升级前的任务	243
升级发行版本 4 Access Manager	246
检验 Access Manager 升级	255
升级后的任务	256
升级回滚	257
多实例升级	257
仅升级发行版本 4 Access Manager SDK	259
升级前的任务	260
升级发行版本 4 Access Manager SDK	260
检验 Access Manager SDK 升级	261
升级回滚	261
从 Java ES 发行版本 3 升级 Access Manager	262
发行版本 3 Access Manager 的升级	262
多实例升级	262
从 Java ES 发行版本 2 升级 Access Manager	265
升级前的任务	265
升级 Access Manager 依赖项	266
升级目录模式	266
为目录重新建立索引	266
发行版本 2 Access Manager 的升级	267
升级发行版本 2 Access Manager: Web Server Web 容器	267
升级发行版本 2 Access Manager: Application Server Web 容器	267
检验 Access Manager 升级	277
升级后的任务	277
升级回滚	277
多实例升级	277

第 15 章 Portal Server	279
Portal Server 升级概述	281
关于 Java ES 发行版本 5 Portal Server	281
Portal Server 升级指导说明	281
Portal Server 数据	282
Portal Server 升级策略	283
兼容性问题	283
Portal Server 组件依赖性	284
选择性升级问题	285
Web 容器升级方案	286
双重升级	286
从 Java ES 发行版本 4 升级 Portal Server	287
简介	287
发行版本 4 Portal Server 的升级	288
发行版本 4 升级前的任务	288
升级发行版本 4 Portal Server (Solaris)	293
升级发行版本 4 Portal Server (Linux)	297
检验升级	300
发行版本 4 升级后的任务	300
升级回滚 (Solaris)	308
升级回滚 (Linux)	309
多实例升级	310
从 Java ES 发行版本 3 升级 Portal Server	313
发行版本 3 升级前的任务：升级 Portal Server 依赖项	313
升级发行版本 3 Portal Server	314
发行版本 3 升级后的任务	314
多实例升级	315
从 Java ES 发行版本 2 升级 Portal Server	316
发行版本 2 升级前的任务	316
升级发行版本 2 Portal Server	317
升级发行版本 2 Portal Server: Web Server Web 容器	317
升级发行版本 2 Portal Server: Application Server Web 容器	318
发行版本 2 升级后的任务	321
单点登录配置	321
启用 URLScrapper 频道	325
删除网关服务条目	325
多实例升级	325

从过渡功能版本 7.0 升级 Portal Server	326
Portal Server IFR 升级简介	326
Portal Server IFR 7.0 升级	327
IFR 7 升级前的任务	327
升级 Portal Server IFR 7.0 (Solaris)	329
升级 IFR 7 Portal Server (Linux)	331
检验升级	334
IFR 7 升级后的任务	335
升级回滚 (Solaris)	342
升级回滚 (Linux)	343
多实例升级	343
第 16 章 Portal Server Secure Remote Access	345
Portal Server Secure Remote Access 升级概述	347
关于 Java ES 发行版本 5 Portal Server Secure Remote Access	347
Portal Server Secure Remote Access 升级指导说明	348
Portal Server Secure Remote Access 数据	349
Portal Server Secure Remote Access 升级策略	349
兼容性问题	349
Portal Server Secure Remote Access 组件依赖性	350
选择性升级问题	350
双重升级	351
从 Java ES 发行版本 4 升级 Portal Server Secure Remote Access	352
简介	352
发行版本 4 Portal Server Secure Remote Access 的升级	353
发行版本 4 升级前的任务	354
升级发行版本 4 Portal Server Secure Remote Access (Solaris)	357
升级发行版本 4 Portal Server Secure Remote Access (Linux)	359
检验升级	361
发行版本 4 升级后的任务	361
升级回滚 (Solaris)	363
升级回滚 (Linux)	364
多实例升级	364
从 Java ES 发行版本 3 升级 Portal Server Secure Remote Access	369
升级 Portal Server Secure Remote Access 依赖项	369
升级发行版本 3 Portal Server Secure Remote Access	370
多实例升级	370

从 Java ES 发行版本 2 升级 Portal Server Secure Remote Access	371
发行版本 2 升级前的任务	371
升级 Portal Server Secure Remote Access 依赖项	371
删除网关服务条目	372
升级发行版本 2 Portal Server Secure Remote Access	372
发行版本 2 升级后的任务	373
为 Proxylet 服务设置 Portal Server 域	373
多实例升级	373
从过渡功能版本 7.0 升级 Portal Server Secure Remote Access	374
简介	374
Portal Server Secure Remote Access IFR 7.0 升级	375
升级前的任务	375
升级 Portal Server Secure Remote Access IFR 7.0 (Solaris)	375
升级 Portal Server Secure Remote Access IFR 7.0 (Linux)	377
检验升级	379
升级后的任务	379
升级回滚 (Solaris)	379
升级回滚 (Linux)	380
多实例升级	380
 附录 A Java Enterprise System 发行版本内容	381
Java ES 2003Q4 (发行版本 1)	382
发行版本 1 安装程序可选组件	382
发行版本 1 共享组件	383
Java ES 2004Q2 (发行版本 2)	384
发行版本 2 安装程序可选组件	384
发行版本 2 共享组件	385
Java ES 2005Q1 (发行版本 3)	387
发行版本 3 安装程序可选组件	387
发行版本 3 共享组件	390
Java ES 2005Q4 (发行版本 4)	392
发行版本 4 安装程序可选组件	392
发行版本 4 共享组件	395
Java ES 5 (发行版本 5)	397
发行版本 5 安装程序可选组件	397
发行版本 5 共享组件	400
 索引	403

表格列表

表 1	印刷约定	21
表 2	符号约定	21
表 3	Shell 提示符	22
表 4	Java Enterprise System 文档	23
表 1-1	Java ES 5 产品组件	28
表 1-2	Java ES 5 共享组件	29
表 1-3	Java ES 产品组件升级方法	33
表 1-4	升级过程中的各个阶段	35
表 1-5	升级到 Java ES 5（发行版本 5）的方式	37
表 1-6	选择性升级与全部升级之比较	39
表 1-7	Java ES/ 操作系统支持矩阵	41
表 1-8	Java ES 5 产品组件的双重升级支持	42
表 1-9	Java ES 5（发行版本 5）产品组件的共享组件依赖性	43
表 1-10	Java ES 产品组件依赖性	47
表 1-11	Application Server 未升级到发行版本 5 时所需的修补程序	53
表 2-1	按平台而定的指向 Java SE 的符号链接	65
表 2-2	Java SE 版本检验输出	66
表 2-3	Common Agent Container 目录路径	66
表 3-1	升级到 Java ES 5（发行版本 5）：Sun Cluster 3.1 8/05 软件的方式	71
表 3-2	Sun Cluster 数据用途	71
表 3-3	Sun Cluster 版本检验输出	75
表 4-1	Sun Cluster Geographic Edition 目录路径	79
表 4-2	升级到 Java ES 5（发行版本 5）：Sun Cluster Geographic Edition 3.1 2006Q4 的方式	80
表 4-3	Sun Cluster Geographic Edition 数据用途	80

表 4-4	Sun Cluster Geographic Edition 版本检验输出	83
表 5-1	Directory Server 目录路径	89
表 5-2	升级到 Java ES 5（发行版本 5）：Directory Server 6.0 的方式	90
表 5-3	Directory Server 数据用途	92
表 5-4	Directory Server 版本检验输出	95
表 6-1	Directory Proxy Server 目录路径	105
表 6-2	升级到 Java ES 5（发行版本 5）：Directory Proxy Server 6.0 的方式	106
表 6-3	Directory Proxy Server 数据用途	107
表 6-4	Directory Proxy Server 版本检验输出	111
表 7-1	Web Server 目录路径	119
表 7-2	升级到 Java ES 5（发行版本 5）：Web Server 7.0 的方式	120
表 7-3	Web Server 数据用途	121
表 7-4	Web Server 默认名称	122
表 7-5	Web Server 版本检验输出	124
表 7-6	WSInstall.properties 值	129
表 7-7	wadm 公共命令选项	131
表 7-8	wadm migrate-server 命令选项和操作对象	132
表 7-9	wadm create-instance 命令选项和操作对象	133
表 7-10	wadm set-virtual-server-prop 命令选项和操作对象	135
表 7-11	wadm deploy-cofig 命令选项和操作对象	135
表 7-12	wadm set-search-collection-prop 命令选项和操作对象	137
表 8-1	Java DB 目录路径	141
表 8-2	升级到 Java ES 5（发行版本 5）：Java DB 10.1.3 的方式	142
表 8-3	Java DB 数据用途	143
表 8-4	Java DB 版本检验输出	146
表 9-1	HADB 目录路径	151
表 9-2	升级到 Java ES 5（发行版本 5）：HADB 4.4.3 的方式	152
表 9-3	HADB 数据用途	153
表 9-4	HADB 版本检验输出	155
表 9-5	用于在 Solaris 平台上升级 HADB 的软件包版本	156
表 10-1	Message Queue 目录路径	161
表 10-2	升级到 Java ES 5（发行版本 5）：Message Queue 3.7 UR1 的方式	163
表 10-3	Message Queue 数据用途 (Solaris OS)	164
表 10-4	Message Queue 版本检验输出	168

表 10-5	Message Queue 版本检验输出	177
表 11-1	Application Server 目录路径	184
表 11-2	升级到 Java ES 5（发行版本 5）：Application Server Enterprise Edition 8.2 的方式	186
表 11-3	Solaris OS 捆绑的 Application Server 版本的升级方式	187
表 11-4	Application Server 数据用途	188
表 11-5	Application Server 版本检验输出	192
表 12-1	Service Registry 目录路径	209
表 12-2	升级到 Java ES 5（发行版本 5）：Service Registry 3.1 的方式	210
表 12-3	Service Registry 数据用途	211
表 12-4	Service Registry 版本检验输出	215
表 13-1	Web Proxy Server 目录路径	221
表 13-2	升级到 Java ES 5（发行版本 5）：Web Proxy Server 4.0.4 的方式	222
表 13-3	Web Proxy Server 数据用途	223
表 13-4	Web Proxy Server 版本检验输出	226
表 13-5	用于在 Solaris 上升级 Web Proxy Server 的修补程序	227
表 13-6	用于在 Linux 上升级 Web Proxy Server 的修补程序	229
表 13-7	Web Proxy Server 版本检验输出	232
表 14-1	Access Manager 目录路径	235
表 14-2	升级到 Java ES 5（发行版本 5）：Access Manager 7.1 的方式	237
表 14-3	Access Manager 数据用途	238
表 14-4	用于 Access Manager 升级的 Web 容器升级方案	241
表 14-5	Access Manager 版本检验输出	243
表 14-6	用于升级 Access Manager Mobile Access 软件的修补程序	247
表 14-7	Mobile Access 修补程序参数	247
表 14-8	Access Manager 配置参数：ampre71upgrade	248
表 14-9	amconfig 支持的管理服务器模式	251
表 14-10	Access Manager 配置参数：amconfig	252
表 14-11	amconfig 参数：发行版本 5 Web Server	253
表 14-12	Access Manager 配置参数：amupgrade	254
表 14-13	用于升级 Access Manager Mobile Access 软件的修补程序	268
表 14-14	Mobile Access 修补程序参数	269
表 14-15	Access Manager 配置参数：ampre71upgrade	270
表 14-16	Access Manager 配置参数：amconfig	273

表 14-17	Access Manager 配置参数: amupgrade	275
表 15-1	Portal Server 目录路径	280
表 15-2	升级到 Java ES 5（发行版本 5）: Portal Server 7.1 的方式	281
表 15-3	Portal Server 数据用途	282
表 15-4	用于 Portal Server 升级的 Web 容器升级方案	286
表 15-5	Portal Server 版本检验输出	289
表 15-6	psupgrade 脚本根据 Web 容器升级方案所需的信息	290
表 15-7	JVM 设置的位置	292
表 15-8	用于将 Portal Server IFR 升级到发行版本 5 的修补程序	326
表 16-1	Portal Server Secure Remote Access 目录路径	346
表 16-2	升级到 Java ES 5（发行版本 5）: Portal Server Secure Remote Access 7.1 的方式	348
表 16-3	Portal Server Secure Remote Access 数据用途	349
表 16-4	Portal Server Secure Remote Access 版本检验输出	354
表 16-5	用于将 Portal Server Secure Remote Access IFR 升级到发行版本 5 的修补程序	374

前言

Java Enterprise System 适用于 UNIX 的升级指南包含在 Sun Solaris™ 操作系统 (Solaris Operating System, Solaris OS) 或 Red Hat Enterprise Linux (RHEL) 操作系统环境中升级 Sun Java™ Enterprise System (Java ES) 软件所需的信息。其中未述及 HP-UX 环境中的升级。

本指南记述了从 Java ES 2004Q2（发行版本 2）、Java ES 2005Q1（发行版本 3）和 Java ES 2005Q4（发行版本 4）到 Java ES 5（发行版本 5）的升级。

本前言包含以下各节：

- 第 20 页中的“目标读者”
- 第 20 页中的“本书所采用的约定”
- 第 22 页中的“相关文档”
- 第 24 页中的“联机访问 Sun 资源”
- 第 24 页中的“联系 Sun 技术支持”
- 第 25 页中的“第三方 Web 站点引用”
- 第 25 页中的“Sun 欢迎您提出意见”

目标读者

本书适用于想要升级 Java ES 软件的系统管理员或软件技术人员。

本书假定您熟悉以下内容：

- 企业级软件产品的安装
- 所处环境中当前部署的 Java ES 组件
- 所支持 Java ES 平台上的系统管理和联网
- 群集模型（如果安装的是群集软件）

本书所采用的约定

本节各表介绍了本书所采用的约定。

管理界面

对于本升级指南中记述的大多数升级过程，可以使用两种管理界面：图形用户界面 (graphical user interface, GUI) 和命令行界面。

大多数情况下，在记述 Java ES 组件升级过程时，本升级指南使用的是命令行界面。命令行界面可用于编写 Java ES 部署升级脚本，这样便可在必要时重复执行过程。

但是，如果在过程中使用的是 Java ES 安装程序，则会介绍 GUI 界面而不是基于文本的交互式界面。可以将 Java ES 安装程序会话保存在一个状态文件中，必要时可使用该文件重复执行过程。

印刷约定

下表介绍了本书所采用的印刷约定。

表 1 印刷约定

字体	含义	示例
AaBbCc123 (等宽字体)	API 和语言元素、HTML 标记、Web 站点 URL、命令名、文件名、目录路径名、计算机屏幕输出、样例代码。	编辑 .login 文件。 使用 <code>ls -a</code> 列出所有文件。 <code>% You have mail.</code>
AaBbCc123 (等宽粗体)	用户键入的内容，与计算机屏幕输出的显示不同。	<code>% su</code> Password:
AaBbCc123 (斜体)	保留未译的新词或术语以及要强调的词。 命令或路径名中要使用实名或值替换的占位符。	这些称为 <i>class</i> 选项。 文件位于 <i>install-dir/bin</i> 目录。
新词术语强调	新词或术语以及要强调的词。	切勿保存文件。
《书名》	书名	阅读《用户指南》的第 6 章。

符号

下表介绍了本书所采用的符号约定。

表 2 符号约定

符号	说明	示例	含义
[]	包含可选命令选项。	<code>ls [-l]</code>	-l 选项不是必需的。
{ }	包含所需命令选项的一组选择。	<code>-d {y n}</code>	-d 选项要求您使用 y 参数或 n 参数。
-	结合同时发生的多个击键。	Control-A	按 A 键的同时按 Control 键。
+	结合相继发生的多个击键。	Ctrl+A+N	按 Control 键后松开，然后按后续各键。
>	表示图形用户界面中的菜单项选择。	“文件” > “新建” > “模板”	从“文件”菜单中选择“新建”。从“新建”子菜单中选择“模板”。

Shell 提示符

下表介绍了本书所采用的 shell 提示符。

表 3 Shell 提示符

Shell	提示符
UNIX 或 Linux 上的 C shell	<i>machine-name%</i>
UNIX 或 Linux 上的 C shell 超级用户	<i>machine-name#</i>
UNIX 或 Linux 上的 Bourne shell 和 Korn shell	\$
UNIX 或 Linux 上的 Bourne shell 和 Korn shell 超级用户	#
Windows 命令行	C:\

相关文档

通过 <http://docs.sun.com>SM Web 站点可以联机访问 Sun 技术文档。您可以浏览文档库或者搜索具体的书名或主题。

Java ES 文档集中的书籍

Java ES 手册以可移植文档格式 (Portable Document Format, PDF) 和超文本标记语言 (Hypertext Markup Language, HTML) 格式的联机文件方式提供。行动不便的用户借助辅助技术也可以阅读这两种格式的文件。可从以下网址访问 Sun™ 文档 Web 站点：

<http://docs.sun.com>

Java ES 文档包括系统的整体信息及其各组件的详细信息。可从以下网址访问此文档：

<http://docs.sun.com/coll/1286.2> 及 <http://docs.sun.com/coll/1382.2>

下表列出了 Java ES 文档集中的系统级手册。左列提供了各文档的名称和文件号码位置，右列概括介绍了相应文档的内容。

表 4 Java Enterprise System 文档

文档	目录
《适用于 UNIX 的 Java Enterprise System 5 发行说明》 http://docs.sun.com/doc/820-0451	含有有关 Java ES 的最新信息，包括已知问题。此外，各组件还有各自的发行说明。
《Java Enterprise System 5 技术概述》 http://docs.sun.com/doc/820-0883	介绍 Java ES 的基本技术和概念。描述组件、体系结构、过程和功能。
《Java Enterprise System 部署规划指南》 http://docs.sun.com/doc/819-3450	简介如何基于 Java ES 规划和设计企业部署解决方案。介绍部署规划和设计的基本概念及原理，论述解决方案生命周期，并提供基于 Java ES 规划解决方案时使用的高级示例和策略。
《Java Enterprise System 5 安装规划指南》 http://docs.sun.com/doc/820-0876	帮助您在硬件、操作系统和网络方面为 Java ES 的部署制定实现规范。描述要在安装和配置计划中解决的问题，如组件依赖性。
《适用于 UNIX 的 Java Enterprise System 5 安装指南》 http://docs.sun.com/doc/820-0103	指导您完成在 Solaris 操作系统或 Linux 操作系统上安装 Java ES 的整个过程。还说明了如何在安装后配置组件以及检验它们是否正常工作。
《适用于 UNIX 的 Java Enterprise System 5 安装参考》 http://docs.sun.com/doc/820-0589	给出有关配置参数的附加信息，提供要在配置规划时使用的工作表，并且列出了参考资料，如默认目录和端口号。
《适用于 UNIX 的 Java Enterprise System 5 升级指南》 http://docs.sun.com/doc/820-0890	说明如何在 Solaris 操作系统或 Linux 操作环境上升级 Java ES 。
《Sun Java Enterprise System 5 监视指南》 http://docs.sun.com/doc/820-1592	说明如何为每个产品组件设置 Monitoring Framework ，以及如何使用 Monitoring Console 查看实时数据和设置阈值报警。
《Java Enterprise System 术语表》 http://docs.sun.com/doc/819-4631	定义 Java ES 文档中使用的术语。

搜索 Sun 产品文档

除了从 docs.sun.com Web 站点搜索 Sun 产品文档外，还可以使用搜索引擎进行搜索，方法是在搜索字段中键入以下语法：

```
search-term site:docs.sun.com
```

例如，要搜索 “broker”，请键入以下内容：

```
broker site:docs.sun.com
```

要将其他 Sun Web 站点包括到搜索范围内（例如，java.sun.com、www.sun.com、developers.sun.com），请在搜索字段中将 docs.sun.com 替换为 sun.com。

联机访问 Sun 资源

有关产品下载、专业服务、修补程序和支持以及其他开发者信息，请访问以下网址：

- 下载中心
<http://www.sun.com/software/download/>
- 客户解决方案
<http://www.sun.com/service/sunjavasystem/sjsservicessuite.html>
- Sun 企业服务、Solaris 修补程序及支持
<http://sunsolve.sun.com/>
- 开发者信息
<http://developers.sun.com>

以下位置提供了有关 Java Enterprise System 及其组件的信息：

<http://www.sun.com/software/javaenterprisesystem/index.html>

联系 Sun 技术支持

如果您遇到通过本文档无法解决的技术问题，请访问
<http://www.sun.com/service/contacting>。

第三方 Web 站点引用

Sun 对本文档中提到的第三方 Web 站点的可用性不承担任何责任。对于此类站点或资源中的（或通过它们获得的）任何内容、广告、产品或其他资料，Sun 并不表示认可，也不承担任何责任。对于因使用或依靠此类站点或资源中的（或通过它们获得的）任何内容、产品或服务而造成的或连带产生的实际或名义损坏或损失，Sun 概不负责，也不承担任何责任。

Sun 欢迎您提出意见

Sun 致力于提高其文档的质量，并十分乐意收到您的意见和建议。

要共享您的意见，请访问 <http://docs.sun.com>，然后单击 "Send Comments"（发送意见）。请在联机表单中提供文档标题和文件号码。文件号码包含 7 位或 9 位数字，可在书的标题页或在文档顶部找到该号码。提出意见时您还需要在表格中输入文件的英文文件号码和标题。本文件的英文文件号码是 819-6553，文件标题为《Sun Java Enterprise System 5 Upgrade Guide for UNIX》。

Sun 欢迎您提出意见

升级规划

本章提供了进行升级规划时所用的信息，适用于在 Sun Solaris™ 操作系统或 Red Hat Enterprise Linux（简称为 Linux）操作系统环境中将 Sun Java™ Enterprise System (Java ES) 软件升级到 Java ES 5。

其中包括以下各节：

- 第 28 页中的“Java ES 5 组件”
- 第 31 页中的“Java ES 升级技术”
- 第 35 页中的“升级过程”
- 第 36 页中的“升级计划注意事项”
- 第 43 页中的“Java ES 组件依赖性”
- 第 50 页中的“升级先后顺序指导原则”
- 第 53 页中的“特殊情况”
- 第 54 页中的“Java ES 5 升级与 Solaris 10 区域”

Java ES 5 组件

在讲述 Java ES 软件升级规划之前，本节先介绍 Java ES 5（发行版本 5）中所包括的组件。视具体的升级方案，可能需要将其中的一个或多个组件升级到其在发行版本 5 中的版本。

Java ES 组件分成了多种不同的类型，如《Java Enterprise System 5 技术概述》(<http://docs.sun.com/doc/820-0883>) 中所述：

- **产品组件。** Java ES 产品组件包括：
 - 系统服务组件，这些组件提供主要的 Java ES 基础结构服务
 - 服务质量组件，这些组件用于增强系统服务产品组件可以在 Java ES 安装程序内进行选择。
- **共享组件。** Java ES 共享组件是 Java ES 产品组件所依赖的本地共享库。共享组件由 Java ES 安装程序自动进行安装。安装哪些共享组件取决于所安装的产品组件。

发行版本 5 产品组件

下表按字母顺序列出了发行版本 5 产品组件，以及在后续表格中所使用的缩写形式。对于其中的服务质量组件，该表包括它们所提供的服务增强类型。

表 1-1 Java ES 5 产品组件

产品组件	缩写	版本	类型
Access Manager	AM	7.1	系统服务组件
Application Server	AS	8.2	系统服务组件
Directory Proxy Server	DPS	6.0	服务质量：访问组件
Directory Server	DS	6.0	系统服务组件
High Availability Session Store	HADB	4.4.3	服务质量：可用性组件
Java DB	JavaDB	10.2	系统服务组件
Message Queue	MQ	3.7 UR1	系统服务组件
Monitoring Console	MC	1.0	服务质量：管理组件
Portal Server	PS	7.1	系统服务组件

表 1-1 Java ES 5 产品组件（续）

产品组件	缩写	版本	类型
Portal Server Secure Remote Access	PSRA	7.1	服务质量：访问组件
Service Registry	SR	3.1	系统服务组件
Sun Cluster	SC	3.1 8/05	服务质量：可用性组件
Sun Cluster Geographic Edition	SCG	2006Q4	服务质量：可用性组件
Web Proxy Server	WPS	4.0.4	服务质量：访问组件
Web Server	WS	7.0	系统服务组件

发行版本 5 共享组件

下表按字母顺序列出了发行版本 5 共享组件，以及在后续表格中所使用的缩写形式。

表 1-2 Java ES 5 共享组件

共享组件	版本	缩写
Apache Commons Logging	1.0.3	ACL
Jakarta ANT Java/XML-based build tool	1.6.5	ANT
Berkeley Database	4.2.52	BDB
Common Agent Container	1.1 和 2.0	CAC
FastInfoSet	1.0.2	FIS
International Components for Unicode	3.2	ICU
Instant Messenger SDK	6.2.8	IM-SDK
Java Platform, Standard Edition	5.0 Update 7	Java SE
JavaBeans™ Activation Framework	1.0.3	JAF
Java Studio Web Application Framework	2.1.5	JATO
JavaHelp™ runtime	2.0	JavaHelp
JavaMail™ runtime	1.3.2	JavaMail
Java Architecture for XML Binding runtime	2.0.3	JAXB
Java API for XML Processing	1.3.1	JAXP
Java API for XML Registries runtime	1.0.8	JAXR

表 1-2 Java ES 5 共享组件（续）

共享组件	版本	缩写
Java API for XML-based Remote Procedure Call runtime	1.1.3_01	JAX-RPC
Java API for Web Services runtime	2.0	JAXWS
Java Calendar API	1.2	JCAPI
Java Dynamic Management™ Kit runtime	5.1.2	JDMK
Java Security Services (Network Security Services for Java)	4.2.4 和 3.1.11	JSS 和 JSS3
JavaServer Pages™ Standard Tag Library	1.0.6	JSTL
KT Search Engine	1.3.4	KTSE
LDAP C SDK	6.0	LDAP C SDK
LDAP Java SDK	4.19	LDAP J SDK
Mobile Access Core	6.2	MA Core
Netscape Portable Runtime	4.6.4	NSPR
Network Security Services	3.11.4	NSS
SOAP Runtime with Attachments API for Java	1.3	SAAJ
Simple Authentication and Security Layer	2.19	SASL
Sun Explorer Data Collector（仅限 Solaris）	4.3.1	SEDC
Sun Java Monitoring Framework	2.0	MFWK
Sun Java Web Console	3.0.2	SJWC
Web Services Common Library	2.0	WSCL
XML Web Services Security	2.0	XWSS

Java ES 升级技术

只用单个系统实用程序根本无法升级所有的 Java ES 组件。此外，如下面各节所述，产品组件和共享组件的升级特点和升级技术也各不相同。

产品组件升级

将 Java ES 产品组件升级到发行版本 5 的过程，是依照本升级指南中所述的组件特定升级过程，按组件、计算机逐个执行的。

产品组件升级可以是主要功能升级或错误修复升级，前者可能与组件的先前版本不兼容，后者与先前版本完全兼容。鉴于 Java ES 组件之间的依赖性，执行某一升级在本质上可能会影响到是否还需要升级其他组件。

Java ES 产品组件升级包括两个基本操作，即镜像 Java ES 产品组件的最初安装和配置：

- **软件升级的安装。** 升级软件用于增强或修复现有软件，或是替换现有软件。通过对现有软件包应用修补程序、有选择地替换现有软件包、安装新软件包或完全重新安装组件软件，便可实现软件安装。
- **重新配置。** 重新配置包括为支持升级软件而需要对配置数据、用户数据或动态应用程序数据所做的任何更改。数据更改可以是附加数据、数据格式更改（无论是在属性文件中还是在数据库模式中），也可以是将数据迁移到新的位置。重新配置有时会要求您执行某个过程，有时则会自动进行。在某些情况下，重新配置还需要将组件软件重新部署到某个 Web 容器。

此外，Java ES 产品组件升级通常需要先执行升级前任务，该次升级才能起作用，在某些情况下，还需要执行升级后过程。

产品组件升级方法

用于安装升级软件和执行组件重新配置的、特定于组件的升级过程包括以下升级方法：

- 使用 [Java ES 安装程序的升级功能](#)
- [执行产品组件的全新安装](#)
- [运行特定于组件的升级实用程序](#)
- [修补现有的组件软件包](#)

使用 Java ES 安装程序的升级功能

发行版本 5 安装程序包括一项升级功能，在以下几种特殊情况下可通过它来执行产品组件升级：Application Server、Message Queue、HADB 和 Java DB。当

Java ES 安装程序检测到这些产品组件先前安装的发行版本时，它会将这些组件标记为“可升级”。

在升级上述任意组件之前，安装程序会检查是否有当前版本和先前版本的共享组件。如果安装程序检测到选定组件所需的共享组件为先前版本或者不存在，则安装程序会升级当前安装的所有共享组件，并安装选定组件所需的任何缺少的共享组件。在某些情况下（特别是 Application Server），安装程序还会对要升级的组件所依赖的产品组件进行升级。

安装程序会删除先前版本的软件包，安装发行版本 5 产品组件软件包，然后根据需要重新配置所升级的产品组件。（但是，对于同 Solaris 9 操作系统捆绑在一起的 Application Server 而言，安装程序不会删除软件包；参见第 199 页中的“发行版本 2 Application Server 的升级”）。

如果使用 Solaris 10 操作系统区域 (zone) 功能，则需要特别注意。参见第 54 页中的“Java ES 安装程序中的区域支持”。

执行产品组件的全新安装

一些产品组件是通过使用 Java ES 安装程序执行组件的全新安装来进行升级的。首先删除先前版本的软件包，然后在同一路径中安装发行版本 5，或者在类似的路径中安装发行版本 5 并使先前版本保持原样。

两种情况下，都要通过以下方式重新配置产品组件：将先前版本的配置数据迁移到新安装、执行新配置，或是将这两种方法相结合。对于一些产品组件，提供了一个实用程序来重新配置或迁移该组件的配置数据。

运行特定于组件的升级实用程序

一些产品组件提供了升级实用程序或脚本，用于将组件自动升级到发行版本 5。该实用程序通常既可执行软件包的升级又可执行升级过程中所需的任何重新配置。为将这些组件部署到某个 Web 容器，该实用程序通常会将已升级的组件软件重新部署到该 Web 容器。

修补现有的组件软件包

对于一些产品组件，升级是通过手动修补现有软件包来执行的。虽然 Solaris 和 Linux 平台运用了类似的技术，通过软件包注册表来管理已安装的软件包并跟踪对这些软件包所做的更改，但是这两种平台之间在修补技术上的差异会对升级过程产生影响。

- **Solaris 平台。** 通过 Solaris 的 pkgadd 和 pkgrm 命令来安装和删除软件包。软件包内容一经安装，即可使用通过 patchadd 和 patchrm 命令应用或删除的修补程序来进行修改。Solaris 软件包的修补程序是通过 SunSolve Web 站点进行分发的，其网址为：

<http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access>

Solaris 修补程序可以修补一个或多个软件包。patchadd 命令会保存所修补软件包的备份，以方便使用 patchrm 命令删除修补程序。修补程序通过修补程序 ID 进行标识，该 ID 包括一个修补程序号，其后紧跟着修订版本号，该版本号会随着修补程序的不断修改而递增。

- **Linux 平台。** 可通过 rpm 命令来安装或更新 Red Hat Enterprise Linux 软件包 (RPM)。但是，软件包内容一旦安装，便无法使用修补程序进行修改。不过，可使用 rpm -U 命令选项更新 RPM 软件包，该选项会用较新的软件包替换当前软件包。

为方便起见，许多 RPM 软件包升级内容不但会在 Java ES 发行版本 5 分发时进行分发，而且还会通过 SunSolve Web 站点进行分发，其网址为：
<http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access>

在通过 SunSolve 分发时，会将 RPM 软件包封装为修补程序，并为其分配一个与 Solaris 修补程序类似的修补程序 ID 和修订版本号。这些 Linux 修补程序可包括一个或多个 RPM 软件包，每个软件包均通过唯一的 RPM 名称、RPM 编号和修订版本号加以标识，其中，修订版本号会随着 RPM 软件包的不断修改而递增。

各产品组件所用的升级方法

下表显示了用于将每个产品组件升级到发行版本 5 的升级方法：

表 1-3 Java ES 产品组件升级方法

产品组件	安装已升级的软件	重新配置
Access Manager	替换软件包：使用软件包删除脚本 + 全新安装	使用 amconfig 和 amupgrade 脚本重新配置并重新部署到 Web 容器
Application Server	替换软件包：使用 Java ES 安装程序的升级功能	无，但有一种情况例外：从发行版本 2 升级时，要使用 postinstall 和 asupgrade 脚本
Directory Proxy Server	执行全新安装而不替换先前的软件包。	手动重新配置
Directory Server	执行全新安装而不替换先前的软件包。	使用 dsmig 命令迁移目录数据
High Availability Session Store	替换软件包：使用 Java ES 安装程序的升级功能或进行同样的全新安装	无
Java DB	替换软件包：使用 Java ES 安装程序的升级功能	无
Message Queue	替换软件包：使用 Java ES 安装程序的升级功能或 mqupgrade 脚本（从发行版本 2 升级）	无，但有一种情况例外：在 Linux 上从发行版本 2 升级时要使用 mqmigrate 脚本

表 1-3 Java ES 产品组件升级方法（续）

产品组件	安装已升级的软件	重新配置
Portal Server	替换软件包：使用 psupgrade 脚本	使用 psupgrade 脚本重新配置并重新部署到 Web 容器
Portal Server Secure Remote Access	替换软件包：使用 psupgrade 脚本	使用 psupgrade 脚本重新配置
Service Registry	执行全新安装而不替换先前的软件包。	手动重新配置以及使用 ant upgrade 脚本重新部署到 Application Server 域
Sun Cluster	替换软件包：使用 scinstall 脚本替换二进制文件	使用 scinstall 脚本迁移配置
Sun Cluster Geo	替换软件包：使用 uninstall 脚本 + 全新安装	无
Web Proxy Server	修补二进制文件	无
Web Server	执行全新安装而不替换先前的软件包。	使用 wadm migrate-server 命令迁移服务器实例配置

共享组件升级

Java ES 共享组件升级是升级依赖于这些共享组件的产品组件不可或缺的组成部分。

升级共享组件不需要重新配置组件，也不需要执行升级前或升级后过程。此外，共享组件升级无法回滚到其先前的版本。

大量（大约 30 个）的 Java ES 共享组件以及共享组件与产品组件之间的复杂交互，要求单个操作系统实例内的所有共享组件均要同步到同一个 Java ES 发行版本。操作系统实例是指运行 Solaris 9、Solaris 10 或 Red Hat Enterprise Linux 操作系统的单台计算机，或是运行 Solaris 10 操作系统的计算机上的任何一个虚拟操作系统环境（区域）。

由于要求同步，因此不应逐个升级 Java ES 共享组件，而需将共享组件同时升级到其在发行版本 5 中的版本。

共享组件到发行版本 5 的同步是通过使用 Java ES 安装程序实现的。在执行产品组件的升级时（参见第 31 页中的“使用 Java ES 安装程序的升级功能”），或是在执行产品组件的全新安装时，安装程序会对共享组件进行同步。安装程序还包括一个同步功能，用于升级任何现有的共享组件以及安装任何缺少的共享组件。有关此功能更详尽的说明，参见第 61 页中的“同步所有共享组件”。

升级过程

Java ES 升级过程包括多个阶段，在生产环境中执行这些阶段之前，通常会首先在阶段试验环境中执行它们。使用阶段试验环境，不仅可以测试每个阶段，而且还可以编写供 IT 人员使用的脚本，以升级复杂的 Java ES 部署。

在阶段试验环境中对升级过程进行过测试后，如果您确信升级工作正常，便可在生产环境中再现该过程。

该过程包括下表中所示的各个阶段，在本升级指南中对它们进行了记述。这些阶段不仅适用于单个组件升级，而且在总体上也适用于 Java ES 部署。

表 1-4 升级过程中的各个阶段

升级阶段	说明
计划	制定一个升级计划。在其中指定要升级的 Java ES 组件，以及需要按什么顺序在部署中的不同计算机或操作系统实例上升级这些组件。
升级前的准备	备份配置和应用程序数据、执行对操作系统的任何修补、升级所需的任何依赖项，以及执行用于为升级任何单个组件做准备的其他任务。
升级	获取升级所需的所有必备软件包、修补程序和工具。安装升级软件并按规定重新配置每个组件，包括将数据迁移到已升级的系统。
检验	使用规定的检验测试（包括启动已升级的软件组件和测试各种使用方案），检验是否已成功进行了升级。
升级后的过程	执行任何附加配置、自定义或其他任务，例如，整合新的功能。这些操作可能是使已升级的组件能够正常运行所必需的。
回滚 / 恢复	回滚升级并检验该回滚是否成功。有时出于某种原因，必须要将生产环境恢复到其先前的状态，因此，测试升级回滚十分重要。

升级计划注意事项

在升级计划中，应指定将要升级到发行版本 5 的 Java ES 组件，以及将要按什么顺序在 Java ES 部署中的不同计算机或操作系统实例上升级这些组件。

您的计划不仅将取决于升级目的和优先级，还将取决于部署体系结构的范围和复杂性。

例如，Java ES 部署体系结构可能由单台计算机上运行的单个 Java ES 组件组成，而升级目的是为了修复上一软件发行版本中的某些错误。反之，Java ES 部署体系结构也可能由部署于多台不同计算机上的多个相互依赖的 Java ES 组件组成，而升级目的是为了获得一些新的功能并且要以最短的停机时间达到此目的，为此要将需要升级的组件数量降至最低。

通常，Java ES 组件数量以及部署体系结构中的计算机数量越多，升级计划就越复杂。

不过，除了部署体系结构的范围和复杂性以外，升级计划还取决于诸多注意事项。这些注意事项包括以下因素：

- [升级依赖性](#)
- [所支持的升级方式和策略](#)
- [选择性升级或全部升级](#)
- [多实例升级](#)
- [操作系统注意事项](#)

升级依赖性

在对 Java ES 产品组件的升级进行规划时，其中的一个主要问题就是要了解该组件对其他 Java ES 组件的依赖性，以及是否需要升级其他组件以支持依赖组件的升级。

有两种类型的升级依赖性：

- **硬性升级依赖性。** 升级产品组件要求您升级它所依赖的组件。这种要求可能归因于新功能、新接口或依赖组件所需的错误修复。对于硬性升级依赖性，若不先升级该组件所依赖的组件，就无法成功升级和使用依赖组件。
- **软性升级依赖性。** 升级产品组件不要求您升级它所依赖的组件。对于软性升级依赖性，不用升级该组件所依赖的组件，便可成功升级和使用依赖组件。

升级 Java ES 产品组件要求您升级该组件具有**硬性**升级依赖性的所有组件，但除了本书中指出的几种情况外，您可以不升级该组件具有**软性**升级依赖性的组件。当升

级涉及多个互相依赖的组件时，只要所升级的 Java ES 组件中有一个组件对某个特定组件具有硬性升级依赖性，就必须升级该组件。

在少数特殊情况下，由于所引入的不兼容性，组件的升级还要求您升级它所支持的组件。在本书中指出了这些特殊情况。

所支持的升级方式和策略

您的升级计划取决于您希望升级到发行版本 5 的 Java ES 发行版本。

虽然有可能将所有先前发行版本的 Java ES 软件都升级到 Java ES 5（发行版本 5），但仅支持从 Java ES 2005Q4（发行版本 4）、Java ES 2005Q1（发行版本 3）和 Java ES 2004Q2（发行版本 2）进行升级。尽管本升级指南提供了从 Java ES 2003Q4（发行版本 1）和先于 Java ES 的发行版本进行升级的策略，却未提供执行此类升级的过程。

下表介绍了升级到发行版本 5 的不同方式及其特征，以及在执行升级时所要使用的升级策略。

由于表中所述的各种升级方式之间存在差异，而且产品组件升级过程通常取决于所要升级的发行版本，因此在本升级指南中，将介绍每个产品组件升级的各章分成了多个章节：每一节均体现了一种不同的升级方式。

表 1-5 升级到 Java ES 5（发行版本 5）的方式

Java ES			
产品版本	发行版本	系统特征	升级策略
2005Q4	发行版本 4	Java ES 5（发行版本 5）支持在单台计算机上混合使用发行版本 4 和发行版本 5 产品组件，但是要求将共享组件同步到同一个发行版本。发行版本 4 和发行版本 5 产品组件之间的互操作性已经过测试，并且在《适用于 UNIX 的 Java Enterprise System 5 发行说明》(http://docs.sun.com/doc/820-0451) 中指出了已知的接口不兼容性。	发行版本 4 与发行版本 5 产品组件的共存使得您可能在单台计算机上或在由多台计算机组成的部署体系结构中，有选择地将发行版本 4 产品组件升级到发行版本 5。 如果有任何发行版本 5 产品组件要求支持发行版本 5 共享组件，则必须将计算机上的所有共享组件均同步到发行版本 5。

表 1-5 升级到 Java ES 5（发行版本 5）的方式（续）

产品版本	Java ES 发行版本	系统特征	升级策略
2005Q1	发行版本 3	类似于上述发行版本 4 的升级方式。 Java ES 5（发行版本 5） 支持在单台计算机上混合使用发行版本 3（发行版本 4 也可）和发行版本 5 产品组件，但是要求将共享组件同步到同一个发行版本。发行版本 3 与发行版本 5 组件之间的互操作性已经过测试，并且在《适用于 UNIX 的 Java Enterprise System 5 发行说明》(http://docs.sun.com/doc/820-0451) 中指出了已知的接口不兼容性。	类似于上述发行版本 4 的升级方式。发行版本 3 与发行版本 5 组件的共存使得您有可能在单台计算机上或在由多台计算机组成的部署体系结构中，有选择地将发行版本 3 组件升级到发行版本 5。 如果有任何发行版本 5 产品组件要求支持发行版本 5 共享组件，则必须将计算机上的所有共享组件均同步到发行版本 5。
2004Q2	发行版本 2	与上述发行版本 4 和发行版本 3 的升级方式大不相同。 Java ES 5（发行版本 5） 不支持在单台计算机上混合使用发行版本 2 和发行版本 5 组件，无论是产品组件还是共享组件。在这些发行版本之间存在已知的接口不兼容性，并且发行版本 2 与发行版本 5 组件之间的互操作性未经证实（未经过测试）。	在单台计算机上将组件从发行版本 2 升级到发行版本 5 时，必须将所有的发行版本 2 组件都升级到发行版本 5。不过，有时有可能能够在部署体系结构内混合使用驻留在不同计算机上的发行版本 2 和发行版本 5 组件。
2003Q4 及以前版本	发行版本 1 和先于 Java ES 的版本	类似于上述发行版本 2 的升级方式。 Java ES 5（发行版本 5） 不支持在单台计算机上混合使用发行版本 2 和发行版本 5 组件，无论是产品组件还是共享组件。在这些发行版本之间存在已知的接口不兼容性，并且发行版本 1 或以前发行版本与发行版本 5 组件之间的互操作性未经证实（未经过测试）。	Java ES 对于将发行版本 1 或以前发行版本直接升级到发行版本 5 一事并未证实。 不过，在某些情况下，可以先从发行版本 1 升级到 Java ES 发行版本 3，如发行版本 3 的 Java Enterprise System 2005Q1 升级与迁移指南 (http://docs.sun.com/doc/819-2237) 中所述，然后再从发行版本 3 升级到发行版本 5。在这些情况下，本升级指南中针对该组件的升级指导说明指出了这种可能性。 在其他情况下，从发行版本 1 到发行版本 5 的升级可以采用与从发行版本 2 或发行版本 3 升级到发行版本 5 相同的方式执行，而且在这些情况下，本升级指南中针对该组件的升级指导说明指出了这种可能性。

注	当产品组件发布介于正式 Java ES 发行版本之间的过渡功能发行版本 (Interim Feature Release, IFR) 时，通常会按照与前一 Java ES 发行版本相同的过程来执行 IFR 的升级。例如，如果 IFR 出现于发行版本 3 和发行版本 4 之间，则应按照从发行版本 3 升级到发行版本 5 的过程来升级组件。若情况并非如此（例如，对于 Portal Server 和 Portal Server Secure Remote Access），则会在本升级指南中记述该 IFR 特定的升级过程。
---	--

选择性升级或全部升级

硬性与软性升级依赖性之间的差别使得您有可能在升级计划中有选择地在已部署的系统内升级 Java ES 产品组件。选择性升级适于在单台计算机上从发行版本 3 和发行版本 4 升级到发行版本 5。不支持单台计算机上从发行版本 2 到发行版本 5 的选择性升级。

通常，可以选择是要执行选择性升级，还是要升级计算机上的所有 Java ES 产品组件：

- **选择性升级。** 在此方法中，首先要确定您希望升级到发行版本 5 的 Java ES 产品组件。然后确定对该组件具有硬性依赖性的组件也需要进行升级。对后续每个硬性升级依赖项重复此过程，直到再无其他组件需要升级为止。这样一来便指定了所有需要升级的 Java ES 产品组件。
- **全部升级。** 在此方法中，要将所有已部署的 Java ES 产品组件都升级到发行版本 5。在某些情况下，鉴于部署的复杂性，出于业务方面的原因，一次对整个系统进行升级是行不通的。

下表对这两种用于执行升级的方法进行了比较。

表 1-6 选择性升级与全部升级之比较

升级方法	优点	缺点
选择性升级	所要升级的组件数目最少。	导致已部署系统中所有组件的版本不一致
全部升级	可使已部署系统中所有组件的版本保持一致。	所要升级的组件数目最多

在 Java ES 发行版本 4 中也支持选择性升级。因而有可能使发行版本 3 产品组件和发行版本 4 产品组件在计算机上共存，两者均可有选择地升级到发行版本 5。

多实例升级

升级计划中升级过程的顺序取决于如何在部署体系结构中使用冗余。可以使用 Java ES 组件的多个实例来实现高可用性、可伸缩性、可维护性，或者以某种综合方式达到这些服务质量。有三种技术在 Java ES 部署体系结构中利用了冗余组件：负载均衡（Directory Proxy Server、Web Server、Web Proxy Server、Application Server、Access Manager 和 Portal Server）、高可用性技术（Sun Cluster 和 High Availability Session Store），以及 Directory Server 复制。

在大多数运用冗余的情况中，必须在不引起严重停机的情况下执行升级。进行滚动升级，目的是为了陆续升级组件的冗余实例而不累及它们所提供的服务。

冗余实例通常部署在多台计算机上。在进行升级规划时，可能需要将复制组件的升级与其他组件的升级隔离开来，才能最大程度地缩短停机时间。在执行滚动升级之前，要先在每台计算机上执行复制组件的所有升级前的任务。

每种复制技术均有配置或重新配置过程，它们可能会影响 Java ES 组件总体的升级顺序。例如，在 Sun Cluster 环境中运行的组件可能要求先对 Sun Cluster 进行升级，然后再升级运行于 Sun Cluster 环境中的组件。

在本升级指南中，描述每个产品组件升级的各章介绍了如何为其各自的组件执行多实例升级。

操作系统注意事项

如下所述，有多个操作系统注意事项会对 Java ES 升级计划产生影响。

所需的操作系统修补程序

要成功升级 Java ES 产品组件，必须先修补操作系统或是将操作系统更新到 Java ES 5 产品组件所要求的级别。然而，最好是在执行特定产品组件的升级之前，使操作系统整体达到 Java ES 5 所要求的级别，而不是针对每一种情况应用所需的特定修补程序或修复。

- **Solaris 平台。**操作系统修补程序是通过 SunSolve Web 站点以修补程序群组的形式提供的，修补程序群组是可以一同应用的操作系统修补程序集。可在以下网址获得 Solaris 9 和 10 支持 Java ES 发行版本 5 所需的修补程序群组：
<http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access>
- **Linux 平台。**在 <https://www.redhat.com/apps/download/> 上提供有更新发行版本。不过，在执行 Java ES 升级之前，不必更新 Linux 操作系统。

双重升级：Java ES 和操作系统软件

当您试图将 Java ES 软件或操作系统软件升级到不受支持的版本时，操作系统和 Java ES 软件会变得不相配。下表显示了相关的支持矩阵。

表 1-7 Java ES/ 操作系统支持矩阵

Java ES 发行版本	Solaris			RHEL		
	8	9	10	2.1	3.0	4.0
2003Q4（发行版本 1）	X	X		X		
2004Q2（发行版本 2）	X	X		X		
2005Q1（发行版本 3）	X	X	X	X	X	
2005Q4（发行版本 4）	X	X	X	X	X	
5（发行版本 5）		X	X		X	X

如果升级 Java ES 软件或操作系统软件会导致生成不受支持的配置，则必须执行双重升级，即 Java ES 和操作系统都要进行升级。以下两种情况下可能需要双重升级：

- 您要将操作系统升级到不受已安装 Java ES 软件支持的版本。

例如，在 Solaris 8 和 9 操作系统以及 Red Hat Enterprise Linux (RHEL) 2.1 上支持 Java ES 2004Q2（发行版本 2）。如果您希望将操作系统平台升级到 Solaris 10 或 RHEL 3.0（它们不受 Java ES 发行版本 2 支持），则还需要将 Java ES 发行版本 2 升级到支持已升级平台的 Java ES 发行版本。在这种情况下，最好是升级到 Java ES 5（发行版本 5）。

- 您要将 Java ES 升级到不受现有操作系统软件支持的版本。

例如，在 Solaris 8 和 RHEL 2.1 上支持 Java ES 2005Q1（发行版本 3）和 Java ES 2005Q4（发行版本 4）。然而，如果您要将 Java ES 升级到发行版本 5（它在 Solaris 8 和 RHEL 2.1 上不受支持），则必须将操作系统升级到 Java ES 5（发行版本 5）所支持的版本。在这种情况下，最好是升级到 Solaris 10 或 RHEL 4.0。

通常，可以采用以下两种方法执行双重升级：

- **操作系统全新安装。**安装新的操作系统，接着执行 Java ES 发行版本 5 的全新安装，其中包括迁移早期版本的产品组件数据（如配置数据、运行时数据、自定义等等）。可将操作系统安装在新系统（或 Solaris 10 区域）上，否则会擦除现有文件系统。对于后一种情况，必须先备份组件数据，然后在操作系统安装完毕后进行恢复。
- **操作系统就地升级。**执行操作系统升级，将现有文件系统留在原来的位置，接着将 Java ES 产品组件升级到发行版本 5。要使这种方法起作用，操作系统升级必须对已安装 Java ES 产品组件及其数据以及必需的共享组件的升级没有任何影响。

如果有任何 Java ES 产品组件不支持双重升级，即，以上两种方法都不起作用，则必须在执行操作系统安装或升级之后，重新安装并全新配置该组件。

下表显示了每个 Java ES 产品组件所支持的双重升级方法。

表 1-8 Java ES 5 产品组件的双重升级支持

产品组件	操作系统全新安装	操作系统就地升级
Access Manager	不支持	不支持
Application Server	仅在同一台计算机上支持	支持
Directory Proxy Server	支持	支持
Directory Server	支持	支持
High Availability Session Store	在 Application Server 双重升级上下文中执行	在 Application Server 双重升级上下文中执行
Java DB	支持	支持
Message Queue	支持	支持
Portal Server	不支持	支持
Portal Server Secure Remote Access	不支持	支持
Service Registry	支持	支持
Sun Cluster	不支持	支持
Sun Cluster Geographic Edition	不支持	在 Sun Cluster 双重升级上下文中执行
Web Proxy Server	不支持	支持
Web Server	不支持	支持

操作系统升级

在某些情况下，升级 Solaris 操作系统时会以较早版本覆盖现有的 Java ES 共享组件。在这种情况下，可将与 Solaris 操作系统捆绑在一起的 Message Queue 升级到发行版本 5，以此来恢复正确的 Java ES 版本。升级 Message Queue 时还会强制升级所有驻留的共享组件。

Solaris 10 多区域环境

在多区域环境中安装和升级 Java ES 组件，会涉及到许多问题。有关在 Solaris 10 区域中部署 Java ES 的益处和局限以及在多区域环境中升级 Java ES 组件的建议做法的说明，参见第 54 页中的“Java ES 5 升级与 Solaris 10 区域”。

Java ES 组件依赖性

升级计划中最重要的注意事项之一便是所部署系统中各种 Java ES 组件之间的依赖性。组件之间的依赖性在本质上会影响您执行组件升级的顺序。

本节就影响升级计划的 Java ES 组件依赖性进行了说明。

- 对共享组件的依赖性
- 对产品组件的依赖性

对共享组件的依赖性

第 43 页中的表 1-9 显示了 Java ES 5（发行版本 5）产品组件对 Java ES 共享组件的依赖性。表 1-9 列标题中的产品组件缩写取自第 28 页中的表 1-1。共享组件的缩写详见第 29 页中的表 1-2。

在表 1-9 的矩阵中，发行版本 3 和发行版本 4 升级到发行版本 5 时的硬性升级依赖性标记为 "H"，软性升级依赖性标记为 "S"。对于发行版本 2 升级到发行版本 5 的情况，按照定义，所有共享组件依赖性均为硬性升级依赖性；必须将所有共享组件都从发行版本 2 升级到发行版本 5。

表 1-9 Java ES 5（发行版本 5）产品组件的共享组件依赖性

共享组件	AM	AS	DPS	DS	DSConsole	HADB	JavaDB	MQ	MC	PS	PSRA	SC	SCG	SR	WPS	WS
ANT		S							S	H	H			H		
ACL	S													H		
BDB	S															
CAC	H	S	H	H	H				S	S		S ₁	S ₁			
FIS																
ICU		S	H	H						S					S	S
IM-SDK										S						
Java SE	S	S	H	H	H	S	H	S	S	S	S	S	S	H	S	S
JAF	S	S								S	S			H		
JATO	S	S							S	S		S	S			

表 1-9 Java ES 5（发行版本 5）产品组件的共享组件依赖性（续）

共享组件	AM	AS	DPS	DS	DS Console	HADB	JavaDB	MQ	MC	PS	PSRA	SC	SCG	SR	WPS	WS
JavaHelp™	S	S						S	S							S
JavaMail™	S	S								S	S			H		S
JAXB	S	S														S
JAXP	S	S								S	S			H		S
JAXR	S	S												H		S
JAX-RPC	S	S												H		S
JAXWS																S
JCAPI																
JDMK	H	S	H	H	H				S			S	S			S
JSS	S									S	S				S	S
JSTL																
KTSE										S					S	S
LDAP C SDK	H			H											S	S
LDAP J SDK	S															
MA Core	S									H	H					
MFWK	H			H					H							
NSPR	S	S	H	H				H	S	S	S	S	S		S	H
NSS	S	S		H				H	S	S	S	S	S		S	H
SAAJ	S	S								S	S			H		
SASL				H											S	S
SEDC												S	S			
SJWC	S	S			H				H			S	S			
WSCL	S	S												H		S
XWSS														H		

1. 此依赖性专门针对 Common Agent Container (CAC) 版本 1.1。

表 1-9 中所示的任何产品组件的依赖性，既表示直接共享组件依赖性又表示间接共享组件依赖性：产品组件可能依赖于特定的共享组件（直接依赖性），而该共享组件又依赖于一个或多个其他共享组件（间接依赖性）。第 45 页中的图 1-1 说明了共享组件之间的相互依赖性。

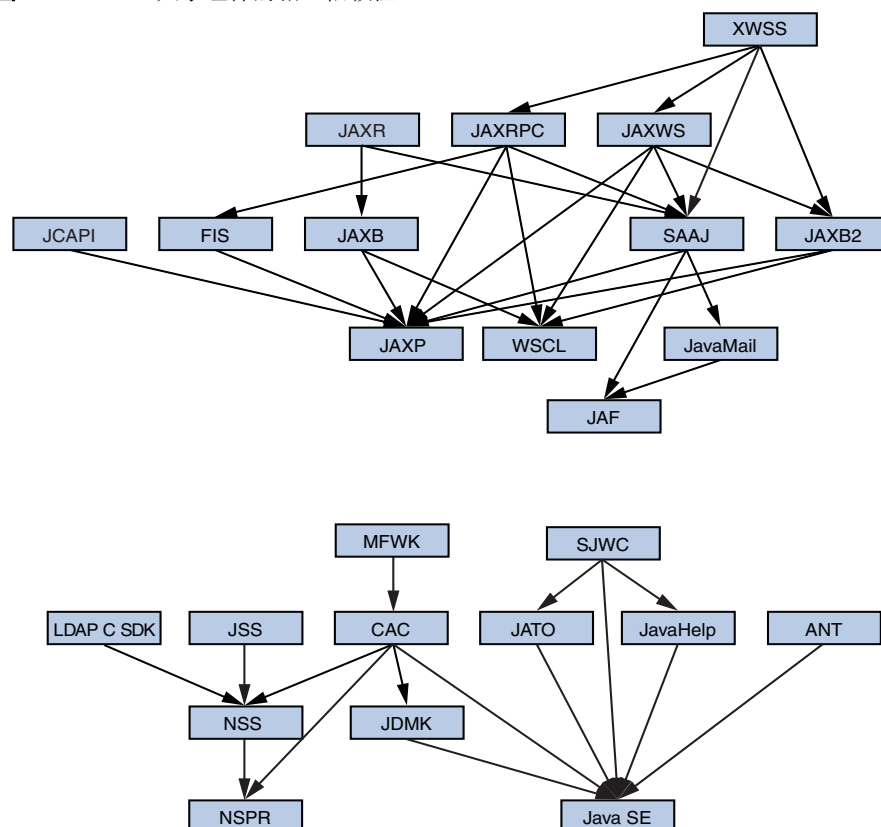
表 1-9 显示了在给定的计算机上升级一个或多个产品组件时必须升级的共享组件。

但是，由于共享组件必须同步（参见第 34 页中的“共享组件升级”），因此无法逐个升级 Java ES 共享组件，而必须将计算机上或操作系统实例中的所有共享组件都同时升级到其在发行版本 5 中的版本。

如果不涉及任何硬性升级依赖性，则不需要升级共享组件。但是，较好的做法是将底层的 Java ES 基础共享组件升级到最新版本。事实上，Java ES 安装程序在安装或升级产品组件时，会自动将驻留在主机上的所有共享组件均同步到发行版本 5。

有关如何手动升级共享组件的信息，参阅第 2 章“升级 Java ES 共享组件”。

图 1-1 共享组件的相互依赖性



对产品组件的依赖性

对产品组件的依赖性分为两大类：运行时依赖性和配置依赖性。

- **运行时依赖性。** 软件系统部署组件之间的交互是软件系统正常工作的基础。在《Java Enterprise System 5 技术概述》中对 Java ES 产品组件之间的基础结构依赖性进行了讨论。如果某个发行版本 5 产品组件对另一个产品组件具有硬性升级依赖性，则仅当它所依赖的组件也进行了升级时，才能按预期的那样成功升级和使用依赖组件。
- **配置依赖性。** 在某些情况下，必须安装、配置 Java ES 组件并使其处于运行状态，方可配置另一个组件。例如，Directory Server 用户/组目录必须正在运行，方可注册 Access Manager 服务。组件升级过程通常包括已升级组件的重新配置或配置数据的迁移。配置依赖性可能会对升级过程的顺序产生影响。

对于运行时依赖性，产品组件之间的关系可能为以下三种类型：

- **强制。** 组件不能在缺少支持组件的情况下进行操作。
- **可选。** 组件可以在缺少支持组件的情况下进行操作，但是其某个功能子集需要支持组件。
- **协作依赖性。** 两个组件均可在缺少对方支持的情况下进行操作，但一起使用这些组件会使某些功能或性能得到增强。

下表显示了第 28 页中的表 1-1 中所列 Java ES 产品组件之间的依赖性和依赖关系。可使用该信息来确定影响升级计划的硬性升级依赖性。

第一列按字母顺序列出了发行版本 5 产品组件，第二列显示了与发行版本 5 组件有依赖关系的其他 Java ES 组件，第三列提供了支持发行版本 5 依赖性的 Java ES 发行版本，第四列描述了依赖关系的特征，最后一列指出了特殊的依赖特征，如支持组件是否必须在本地（相对于远程而言）或其他第三方产品是否能够支持该依赖性。

如果要升级到发行版本 5 的产品组件对发行版本 5 的支持组件（相对于较早发行版本而言）具有依赖性，则支持组件就体现出一种硬性升级依赖性：必须将支持组件也升级到发行版本 5。

表 1-10 Java ES 产品组件依赖性

发行版本 5 产品组件	依赖项 ¹	Java ES 发行版本	依赖性质	特征
Access Manager	Directory Server	2-5	强制：存储配置数据和启用用户数据查找	
	J2EE Web 容器：		强制：提供 Web 容器运行时服务	仅限本地
	- Application Server	4-5		还受到另外的支持；
	- Web Server	4-5		- Weblogic ² - WebSphere ³
Access Manager SDK	Access Manager	3-5	强制：提供 Access Manager 服务	
Access Manager Distr. Authentication	Access Manager	4-5	强制：提供 Access Manager 服务	
	J2EE Web 容器：		强制：提供 Web 容器运行时服务	仅限本地
	- Application Server	4-5		还受到另外的支持；
	- Web Server	4-5		- Weblogic ² - WebSphere ³
Access Manager Session Failover	Access Manager	5	强制：提供 Access Manager 服务	
	Message Queue	4-5	强制：提供可靠的异步消息传送	
Application Server	Message Queue	3-5	强制：提供可靠的异步消息传送	仅限本地
	High Availability Session Store (HADB)	5	强制：存储为支持实例间故障转移所需的会话状态	仅限本地
	Java DB	5	强制：提供默认开发者数据库和其他持久存储。	仅限本地
	Web Server	3-5	可选：提供实例间的负载平衡	仅限本地
Directory Proxy Server	Directory Server	1-5	协作依赖性：其结果会提高目录请求的安全性和性能。向 Directory Proxy Server 提供数据	
Directory Server	Directory Proxy Server	1-5	协作依赖性：其结果会提高目录请求的安全性和性能。分配负载并将来自 Directory Server 的数据放入高速缓存	
High Availability Session Store (HADB)	无			
Java DB	无			

表 1-10 Java ES 产品组件依赖性（续）

发行版本 5 产品组件	依赖项¹	Java ES 发行版本	依赖性质	特征
Message Queue	Directory Server	2-5	可选：存储受管理对象和用户数据	
	J2EE Web 容器： - Application Server	2-5	可选：支持客户机与 Message Queue 代理之间的 HTTP 传输	
	- Web Server	2-5		
	Java DB	5	可选：存储持久消息	仅限本地
	Sun Cluster	2-5	可选：支持高可用性	
Monitoring Cosole	无			
Portal Server	Directory Server	4-5	强制：存储和启用用户配置文件查找	
	J2EE Web 容器： - Application Server	4-5	强制：提供 Web 容器运行时服务	仅限本地
	- Web Server	4-5		
	Access Manager 或 Access Manager SDK	4-5	强制：提供验证与授权服务、单点登录	仅限本地（如果 Access Manager 在远程，则 Access Manager SDK 必须在本地使用）
	Portal Server Secure Remote Access	5	可选：通过 Gateway、Rewriter Proxy 和 Netlet Proxy 组件提供安全远程访问	
	Service RegistryClient	5	强制：提供编译所需的库	
	Java DB	5	强制：提供对若干 portlet 应用程序的支持	
Portal Server Secure Remote Access Gateway	Portal Server	5	强制：支持 Gateway 功能	
	Access Manager 或 Access Manager SDK	4-5	强制：提供验证与授权服务、单点登录	仅限本地（如果 Access Manager 在远程，则 Access Manager SDK 必须在本地使用）
	Directory Server	4-5	强制：存储和启用用户数据查找	
Rewriter Proxy	Portal Server	5	强制：支持 Rewriter Proxy 功能	
Netlet Proxy	Portal Server	5	强制：支持 Netlet Proxy 功能	

表 1-10 Java ES 产品组件依赖性 (续)

发行版本 5 产品组件	依赖项 ¹	Java ES 发行版本	依赖性质	特征
Service Registry Deployment	Application Server	5	强制：提供容器运行时服务	仅限本地
	Java DB	5	强制：提供用于存储服务及相关元数据的默认数据库	仅限本地
	Service RegistryClient	5	强制：提供所需的客户机库	仅限本地
Client	无			
Sun Cluster	无			
Sun Cluster Agents	Sun Cluster	4-5	强制：提供对 Sun Cluster 服务的访问	仅限本地
Sun Cluster Geographic Edition	Sun Cluster	4-5	强制：支持 Sun Cluster Geographic Edition 功能	仅限本地
Web Proxy Server	Directory Server	2-5	可选：提供基于 LDAP 的验证	
	Web Server	2-5	协作依赖性：其结果会提高 HTTP 请求的安全性和性能。向 Web Proxy Server 提供数据	还受到另外的支持； - Weblogic ² - WebSphere ³
Web Server	Directory Server	1-5	可选：提供基于 LDAP 的验证	
	Web Proxy Server	1-5	协作依赖性：其结果会提高 HTTP 请求的安全性和性能。分配负载并将来自 Web Server 的数据放入高速缓存	

1. 对于每个产品组件，各依赖项是按其通常的升级顺序列出的。

2. BEA Weblogic Server

3. IBM WebSphere Application Server

升级先后顺序指导原则

前面几节讲述了如何在选择性升级或全部升级之间进行选择，硬性升级依赖性的影响以及其他一些因素，所有这些都会对您打算升级哪些 Java ES 组件以及需要按何顺序进行升级产生影响。不过，可以遵循几条一般的先后顺序指导原则，尽管它们并不适用于每一种情况。

下面列出了在单台计算机上或在已部署系统中成功升级 Java ES 组件，可以采用的顺序。在您进行升级规划时，可忽略部署体系结构中不包括的那些组件，或者，如果您执行的是选择性升级，则可忽略体现软性升级依赖性的那些组件。

本升级指南中的各章是按组件在以下列表中的出现顺序进行编排的。

注	在升级 Java ES 组件之前，务必要应用操作系统所需的任何更新（参见第 40 页中的“所需的操作系统修补程序”）。
	还要对照第 53 页中的“特殊情况”，看看是否有任何特殊情况适用于您的升级方案。

1. 共享组件（参见第 59 页中的第 2 章“升级 Java ES 共享组件”）

应先升级共享组件，然后再对依赖于它们的组件进行升级。在大多数情况下，共享组件升级是由 Java ES 安装程序处理的，可是对于 Web Proxy Server 和 Portal Server 而言，您必须显式升级共享组件。

2. Sun Cluster 软件（参见第 69 页中的第 3 章“Sun Cluster 软件”）

如果有任何组件运行于 Sun Cluster 环境，且 Sun Cluster 软件需要升级，则应先于使用 Sun Cluster 服务的组件对其进行升级。Sun Cluster 代理在升级时应在 Sun Cluster 升级过程中进行升级。

3. Sun Cluster Geographic Edition 软件（参见第 79 页中的第 4 章“Sun Cluster Geographic Edition”）

Sun Cluster Geographic Edition 应在它所依赖的 Sun Cluster 软件之后而在使用 Sun Cluster 服务的任何组件之前进行升级。

4. Directory Server（参见第 89 页中的第 5 章“Directory Server”）

许多组件都在 Directory Server 中存储用户数据或配置数据，因此通常应先对 Directory Server 执行升级，然后再升级对 Directory Server 具有运行时或配置依赖性的组件。

5. Directory Proxy Server（参见第 105 页中的第 6 章“Directory Proxy Server”）

Directory Proxy Server 对 Directory Server 具有软性升级依赖性，并且可以随时进行升级。但有些组件可能会通过 Directory Proxy Server 访问 Directory Server，因此，如果升级 Directory Proxy Server，则它应在 Directory Server 之后进行升级。

6. Web Server（参见第 119 页中的第 7 章“Web Server”）

许多 Java ES 组件都要求支持 Web 容器，如果进行升级，则应在升级需要 Web 容器服务的组件之前升级 Web 容器。Web 容器服务通常由 Web Server 或 Application Server 提供，但如果体系结构中包含这两者，则在升级 Application Server 之前，先要升级 Web Server。

7. Java DB（参见第 141 页中的第 8 章“Java DB”）

Java DB 必须在 Application Server 之前进行升级，因为后者要求将 Java DB 作为默认数据库。不过，在升级 Application Server 时，Java ES 安装程序会自动升级 Java DB。

8. High Availability Session Store（参见第 151 页中的第 9 章“High Availability Session Store”）

High Availability Session Store (HADB) 必须在 Application Server 之前进行升级，因为后者要求 High Availability Session Store 提供高可用性。不过，在升级 Application Server 时，Java ES 安装程序会自动升级 HADB。

9. Message Queue（参见第 161 页中的第 10 章“[Message Queue](#)”）

Message Queue 必须在 Application Server 之前升级，因为后者要求 Message Queue 与 Java Enterprise Edition (Java EE) 兼容。不过，在升级 Application Server 时，Java ES 安装程序会自动升级 Message Queue。

10. Application Server（参见第 183 页中的第 11 章“[Application Server](#)”）

Application Server 依赖于 Message Queue 和 High Availability Session Store，如果升级，它应在这两个组件之后进行升级。Application Server 的负载平衡插件还依赖于 Web Server，因此，如果使用该功能，则 Application Server 应在 Web Server 之后进行升级。

11. Service Registry（参见第 209 页中的第 12 章“[Service Registry](#)”）

Service Registry 可在升级完 Application Server 之后的任何时候进行升级，因为 Service Registry 的运行时容器服务依赖于 Application Server。

12. Web Proxy Server（参见第 221 页中的第 13 章“[Web Proxy Server](#)”）

Web Proxy Server 可在任何时候升级，然而它通常应在 Web Server 或 Application Server 组件之后进行升级，因为前者要为后两者提供代理服务。Web Proxy Server 是一个新的 Java ES 发行版本 5 组件，可从其以前的非 Java ES 发行版本进行升级。

13. Access Manager（参见第 235 页中的第 14 章“[Access Manager](#)”）

Access Manager 在验证和授权（包括单点登录）方面起着核心作用，如果升级，应在依靠它来提供这些服务的组件之前进行升级。

14. Portal Server（参见第 279 页中的第 15 章“[Portal Server](#)”）

Portal Server 依赖于上述许多组件（Directory Server、Web 容器和 Access Manager），如果升级，应在这些组件之后进行升级。

15. Portal Server Secure Remote Access（参见第 345 页中的第 16 章“[Portal Server Secure Remote Access](#)”）

Portal Server Secure Remote Access 必须在升级 Portal Server 后进行升级。

特殊情况

在进行 Java ES 组件到发行版本 5 的升级规划时，有几种特殊情况需要注意。下面将对此进行介绍。

选择性升级：Application Server 未升级

如果要在运行发行版本 3 或发行版本 4 Application Server (8.1) 的计算机上执行任何 Java ES 组件到 Java ES 5 的选择性升级，而不将 Application Server 升级到发行版本 5，则必须先处理以下几种情况，Application Server 方可继续正常工作：

- **JSP 编译错误。**在执行选择性升级之前，应该首先应用下表中所示的 Application Server 修补程序。

表 1-11 Application Server 未升级到发行版本 5 时所需的修补程序¹

说明	修补程序 ID: Solaris 9 和 10	修补程序 ID: Linux
修复针对的是发行版本 3 和发行版本 4 Application Server	119166-17 (SPARC) 119166-17 (x86)	119168-17

1. 修补程序修订版本号是所需的最低版本号。如果有较新的修订版本可用，则使用较新版本替代表中所示的那些版本。

如果应用修补程序失败，则 Application Server 将会遭遇 JSP 编译错误。（还可以通过应用[表 1-11](#)中的修补程序来解决问题。）

- **ANT 共享组件二进制文件在 Linux 上的重新定位。**发行版本 5 ANT 所处的路径与先前版本不同。必须将 *AppServer8-base/config/asenv.conf* 文件中所指定的、指向 ANT 的 Application Server 环境变量从：

`AS_ANT_LIB="/opt/sun/lib"`

更改为：

`AS_ANT_LIB="/opt/sun/share/lib"`

然后必须重新启动 Application Server。

将 Portal Server 过渡功能版本 (IFR) 7.0 升级到 Java ES 5

如果要在 Web Server 环境中将 Portal Server 从过渡功能版本 (Interim Feature Release, IFR) 7.0 2005Q4 升级到发行版本 5，请查阅[第 326 页](#)中的“[从过渡功能版本 7.0 升级 Portal Server](#)”，以了解[第 50 页](#)中的“[升级先后顺序指导原则](#)”中所提出的那些指导原则的几种例外情况。

Java ES 5 升级与 Solaris 10 区域

本节主要讨论在 Solaris 10 区域中升级 Java ES 软件所涉及的问题，并且提供了在此类环境中的建议升级做法。本节还就《Java Enterprise System 5 安装规划指南》(<http://docs.sun.com/doc/820-0876>) 中的 Java ES 5 和 Solaris 10 区域做了补充说明。

其中包括以下主题：

- [Java ES 安装程序中的区域支持](#)
- [建议的升级做法](#)
- [特殊情况或例外情况](#)

Java ES 安装程序中的区域支持

Java ES 5 安装程序为 Java ES 产品组件的升级（以及安装）和共享组件的同步提供了合格的区域支持。已在安装程序中实现的策略有助于防止升级方案出现问题。

产品组件的升级

如第 31 页中的“使用 [Java ES 安装程序的升级功能](#)”中所述，Java ES 安装程序只能用于升级有限数量的产品组件及其相应的共享组件。升级功能适用于全局区域 (global zone) 和所有的非全局区域 (non-global zone)。

但是，此行为存在三种与区域相关的例外情况：

- 在稀疏根区域中，无法安装或升级某些共享组件，因为它们驻留在只读目录中。在这种情况下，产品组件的升级会暂停，直到在全局区域中安装或升级了此类共享组件之时才会继续。安装程序会提供以下消息：“无法在稀疏根区域中安装或升级所选组件所需的以下共享组件。请先在全局区域中安装或升级这些共享组件，然后再继续。使用“所有共享组件”选项。”
- Application Server 和 Message Queue 均与 Solaris 操作系统捆绑在一起。这两个版本都不能在稀疏根区域中直接进行升级。有关这两个捆绑组件的详细信息，参见第 56 页中的“[产品组件特殊情况](#)”。
- 在全局区域中，如果存在非全局区域，则安装程序会将所有 Java ES 共享组件都同步到发行版本 5（无论是否有任何特定的产品组件需要它们），而不是升级当前安装的所有共享组件并安装所选组件所需的任何缺少的共享组件。这样便可以将所有发行版本 5 共享组件都传播到非全局区域，从而确保在非全局区域中不会出现共享组件版本混杂的情况。

注

有若干特殊或例外情况可能会妨碍产品组件在非全局区域中的安装或升级。在第 56 页中的“[特殊情况或例外情况](#)”中对此进行了介绍。

同步所有共享组件

所有共享组件都必须同步到其在发行版本 5 中的版本，为了应对这种情况，在发行版本 5 中提供了一个共享组件同步选项。当选择了“所有共享组件”选项时，安装程序将会升级当前安装的所有共享组件并安装任何缺少的共享组件，而不管是否有任何特定的产品组件需要它们。此选项适用于全局区域和完全根区域（但不适用于稀疏根区域）。

在以下两种基于区域的升级方案中需要使用“所有共享组件”选项（在第 61 页中的“同步所有共享组件”中对该选项进行了更加详细地介绍）：

- **手动升级产品组件。** 当所要升级的产品组件无法使用 Java ES 安装程序进行升级时，需要使用“所有共享组件”选项来执行此时所需的共享组件安装和升级。
- **稀疏根区域中的升级。** 某些共享组件无法在默认的稀疏根区域中进行安装或升级。因此，当使用 Java ES 安装程序在稀疏根区域中升级产品组件时，可能需要首先根据所涉及的共享组件，在全局区域中同步共享组件。然后在全局区域中使用“所有共享组件”选项执行这种情况下所要求的共享组件安装和升级。

有关 Java ES 安装程序在共享组件方面的区域行为的摘要，参见《Java Enterprise System 5 安装规划指南》(<http://docs.sun.com/doc/820-0876>) 中就 Java ES 5 和 Solaris 10 区域所提供的信息。

建议的升级做法

在制定升级计划时，应对 Java ES 软件任何现有的多区域部署做个调查，同时牢记《Java Enterprise System 5 安装规划指南》(<http://docs.sun.com/doc/820-0876>) 中所概括的区域安装及管理策略。在某些情况下，可能需要在在一个或多个区域中卸载组件并在其他区域中重新安装，方可实施下面建议的做法：

- 避免混合采用多种策略。特别是：
 - 使 Java ES 区域部署和管理策略尽可能保持简单。不要在同一台计算机上对 Java ES 组件混合进行完全根部署和稀疏根部署。（为支持稀疏根区域部署所需的过程和做法，可能会妨碍完全根区域部署。）
 - 不要在全局区域和非全局区域中安装同一 Java ES 产品组件，即使它们的版本不同也如此。（升级全局区域安装所需的过程可能会破坏非全局区域安装。）
 - 当已在完全根区域中安装了发行版本 4（或更早的）Java ES 组件时，不要在全局区域中将 Java ES 组件升级到发行版本 5。全局区域中的升级可能会导致完全根区域中出现发行版本 4 文件和发行版本 5 文件相混合的情况。

- 升级做法：
 - 如果要将所有已安装的发行版本 4 产品组件都升级到发行版本 5，请在全局区域中同步所有的 Java ES 共享组件，然后在已安装了所需产品组件的区域中执行这些产品组件的升级。（发行版本 5 共享组件向后兼容。）
 - 如果已在非区域环境中安装了发行版本 4 或发行版本 5 产品组件，而且希望向环境中添加非全局区域并在新的非全局区域中安装产品组件，则可能需要在全局区域中卸载组件，并在非全局区域中重新安装它们。

特殊情况或例外情况

存在多种特殊情况，其中一些起因于这样一个事实：某些 Java ES 共享组件和某些 Java ES 产品组件是与 Solaris 10 捆绑在一起的。由于这种捆绑的缘故，这些 Java ES 组件自动存在于全局区域中，进而也存在于创建自全局区域的任何非全局区域中。

产品组件特殊情况

- **Message Queue。**Message Queue 与 Solaris 10 捆绑在一起，因而会在创建非全局区域时自动进行传播（除非事先已将 Message Queue 从全局区域中删除）。无法在稀疏根区域中安装或升级 Message Queue。与其他产品组件不同，当 Java ES 安装程序在全局区域中安装或升级 Message Queue 时，在默认情况下会将其传播到非全局区域。
- **Application Server。**Application Server 与 Solaris 10 捆绑在一起，因而会在创建非全局区域时自动进行传播（除非事先已将 Application Server 从全局区域中删除）。以这种方式进行传播时，Java ES 安装程序无法在稀疏根区域中升级安装在 /usr 中的、捆绑的 Application Server（默认情况下 /usr 是只读的）。为了解决这一问题，必须在稀疏根区域中安装发行版本 5 Application Server 之前，手动将捆绑的 Application Server 软件包从全局区域中删除。参见第 200 页中的[“仅限 Solaris OS: 手动删除操作系统捆绑的 Application Server 软件包”](#)。
- **Sun Cluster。**Sun Cluster 软件在非全局区域中不受支持。

共享组件特殊情况

- **Sun Java Web Console (SJWC)。** Java ES 安装程序无法删除与 Solaris 10 (Update 1 和 Update 2) 捆绑在一起的 SJWC 软件包。这些较旧 SJWC 软件包的 `SUNW_PKG_ALLZONES` 属性已被设置为 `True`，这就意味着该软件包在所有区域中都必须相同，且只能由全局管理员进行管理。因此，必须在全局区域中手动删除这些软件包，并以正确的软件包取而代之。

如果 Java ES 安装程序尝试在非全局区域中安装所选的产品组件，并检测到 SJWC 需要进行升级，则安装程序将会受阻。在 Solaris 10 (Update 1 和 2) 上安装时便会发生这种情况。

作为一种解决方法，现已开发了一个特殊的脚本，该脚本会将 SJWC 的旧软件包从全局区域中删除，并将其替换为发行版本 5 SJWC (它具有正确的区域传播属性值)。有关详细信息，参见《适用于 UNIX 的 Java Enterprise System 5 安装指南》。

- **Common Agent Container (CAC)。** 仅当安装了 Sun Cluster、Sun Cluster Geographic Edition 或 Sun Cluster Agents 时，才会安装版本 1.1。如果选择了“所有共享组件”选项，则不会安装该版本。在这种情况下只会安装版本 2.0。

升级 Java ES 共享组件

本章提供有关将 Java ES 共享组件升级到 Java ES 5（发行版本 5）的信息。

每个 Java ES 产品组件均依赖于一个或多个称为 Java ES 共享组件的本地共享库。共享组件在产品组件安装期间由 Java ES 安装程序自动安装，具体安装哪些共享组件视所安装的产品组件而定。部署 Java ES 产品组件期间并不显式选择、安装或配置这些组件。

同样，对于那些可使用 Java ES 安装程序进行升级的产品组件，相应共享组件也会自动升级。

但是，当手动升级产品组件时，如本升级指南中的若干章节所述，可能需要使用本章中所述的过程显式执行共享组件的升级。

本章包含以下各节：

- [第 60 页中的“共享组件升级概述”](#)
- [第 63 页中的“共享组件升级过程”](#)
- [第 64 页中的“特殊升级过程”](#)

请务必先阅读[第 64 页中的“特殊升级过程”](#)，然后再执行共享组件升级。

共享组件升级概述

将共享组件升级到 Java ES 5（发行版本 5）应作为较大规模升级计划的一部分来执行，如第 1 章“升级规划”中所述。为确保升级成功，请仔细阅读第 1 章，并制定一个符合需要的升级计划。

本节包括以下主题：

- [一般注意事项](#)
- [同步共享组件](#)
- [同步所有共享组件](#)
- [Solaris 10 区域注意事项](#)

一般注意事项

升级共享组件时，请考虑以下问题：

- **操作系统问题。** 如第 40 页中的“操作系统注意事项”中所述，执行任何操作系统升级。对于 Solaris 10 操作系统以外的所有平台，应先执行操作系统升级，然后再升级共享组件。
- **排序指导原则。** 查看第 50 页中的“升级先后顺序指导原则”中列出的排序指导原则。通常情况下，首先升级共享组件。但是，在开始升级之前，应了解升级到 Java ES 发行版本 5 的整个顺序。

同步共享组件

测试和支持 Java ES 共享组件与 Java ES 产品组件之间大量（约 30）复杂交互极为困难，这要求单一操作系统实例内的所有共享组件均同步到同一 Java ES 版本。操作系统实例是指运行 Solaris 9 或 Linux 操作系统的单台计算机，而在使用 Solaris 10 操作系统的情况下，它是指在单台计算机上运行的任何虚拟操作系统环境（区域）。

换言之，安装在操作系统实例内的所有 Java ES 共享组件必须是同一版本。此同步要求会对 Java ES 共享组件的安装和升级方式设置一些限制：

- 不同版本的 Java ES 共享组件只能驻留于不同的操作系统实例中。例如，可以在一个操作系统实例中安装 Java ES 发行版本 4 共享组件，在另一操作系统实例中安装 Java ES 发行版本 5 共享组件，但却不能将它们一起安装在同一个操作系统实例中。
- 如果升级操作系统实例中的任一共享组件或引入任一更高版本的新共享组件，则还必须同时升级该操作系统实例中的所有共享组件。（共享组件需要向后兼容，使发行版本 4 产品组件与发行版本 5 共享组件可以一起使用而不会出现问题。）

例如，假定将发行版本 5 产品组件安装在一或多个发行版本 4 产品组件所驻留的操作系统实例中。由于发行版本 5 产品组件需要一些发行版本 5 共享组件，因此该同步要求就意味着在安装发行版本 5 产品组件的同时，也必须将所有驻留在该操作系统实例中的发行版本 4 共享组件升级到发行版本 5。（即使所安装的发行版本 5 产品组件需要与那些已安装的共享组件不同的共享组件，亦是如此。）

同样，如果将发行版本 4 产品组件升级到发行版本 5，且该升级需要升级它所依赖的一些共享组件，则必须将安装在该操作系统实例中的所有共享组件升级到发行版本 5，无论所升级的特定发行版本 4 产品组件是否对所有这些共享组件都具有依赖性。

同步所有共享组件

Java ES 安装程序包括一个同步所有共享组件的功能，适用于所有共享组件必须同步到其发行版本 5 版本的情况。

在 Java ES 安装程序的组件选择页面中选择“所有共享组件”后，安装程序会升级任何现有的共享组件，并安装任何缺少的共享组件，而无论特定产品组件是否需要它们。

同步所有共享组件功能支持 Java ES 安装程序未显式升级的产品组件（即，除 Application Server、Message Queue、HADB 和 Java DB 之外的所有 Java ES 产品组件）的升级。例如，同步所有共享组件功能可用来将 Web Proxy Server 和 Portal Server 升级到发行版本 5。

在这种情况下同步所有共享组件的原因是，安装程序目前不知道在计算机上需要同步哪些共享组件。因此，该功能将所有 Java ES 共享组件安装或升级到发行版本 5 版本。

同步所有共享组件功能还支持许多区域方案，如下面的“[Solaris 10 区域注意事项](#)”所述。

Solaris 10 区域注意事项

阻碍在 Solaris 10 多区域环境中部署 Java ES 的一些限制源自共享组件的注意事项。

这些注意事项中最重要的一点就是，由于稀疏根区域中存在只读文件系统，而使许多共享组件无法安装在稀疏根区域中。此限制适用于那些基目录为 /usr（默认情况下全局区域共享的目录）的共享组件。

许多 Java ES 共享组件不能安装在稀疏根区域中意味着，要在稀疏根区域中成功安装或升级对此类共享组件有依赖性的产品组件，必须先在全局区域中安装或升级这些共享组件，然后再从全局区域传播至非全局区域。

由于与共享组件相关的同步要求（参见第 60 页中的“同步共享组件”），又由于在全局区域中安装或升级共享组件必须照顾到要在稀疏根区域中安装或升级的任何产品组件，因此，有必要将全局区域中的所有共享组件均同步到各自的发行版本 5 版本。这意味着将升级任何现有的共享组件并安装任何缺少的共享组件，而无论稀疏根区域中的任何特定产品组件是否需要它们。

通过在 Java ES 安装程序的组件选择页面中选择“所有共享组件”，可以执行全局区域中的同步共享组件工作。

将共享组件安装到全局区域并由全局区域向外传播时（例如，通过在全局区域中安装 Java ES 产品组件），必须特别小心地维持所有区域中共享组件的同步。否则，非全局区域中较早版本的共享组件可能会与传播自全局区域的发行版本 5 共享组件混合在一起。

升级共享组件的方法

可以使用两种方法来升级在升级特定 Java ES 产品组件时所需的共享组件。一种方法是确定所需的所有共享组件，然后手动将它们安装或升级到各自的发行版本 5 版本。另一种方法是使用 Java ES 安装程序的同步“所有共享组件”功能。

但是，由于与共享组件相关的同步要求（参见第 60 页中的“同步共享组件”）的关系，唯一可行的方法是使用 Java ES 安装程序的同步“所有共享组件”功能。

这是因为在大多数情况下，要确定任一特定情况下需要安装或升级哪些共享组件非常困难，而即使能够成功完成这项工作，手动升级所需共享组件也不是那么简单。有些共享组件可以修补为发行版本 5 版本，但有些则需要用发行版本 5 软件包替换先前的软件包。视升级方式而定，某些先前的软件包可能需要手动删除。

过去，手动升级共享组件的困难性为升级产品组件制造了巨大的障碍。尽管有安装那些可能不需要的共享组件的缺点，但相对于 Java ES 发行版本 3 和发行版本 4 中需要手动升级共享组件来说，同步“所有共享组件”功能依然是一次重大的进步。

共享组件升级过程

共享组件的升级过程使用 Java ES 安装程序的同步“所有共享组件”功能。

1. 以 root 身份登录，或成为超级用户。

```
su -
```

2. 如第 64 页中的“特殊升级过程”所述关闭服务或进程。
3. 启动 Java ES 安装程序。

```
cd Java ES Release 5 distribution/os_arch  
./installer
```

其中，*os_arch* 与您的平台匹配，如 *Solaris_sparc*。（对于命令行界面，使用 `installer -nodisplay` 选项。）

显示“欢迎”和“许可协议”页面之后，将向您显示一个组件选择页面。（如果检测到可由 Java ES 安装程序直接升级的已安装组件，它们的状态会显示为“可升级”。）

4. 在组件选择页面中选择“所有共享组件”。
5. 确认您的选择。

所有共享组件将同步到各自的发行版本 5 版本。

6. 退出 Java ES 安装程序。

有关任何后续过程，参见第 64 页中的“特殊升级过程”。

特殊升级过程

本节提供升级以下共享组件所需的特殊过程：

- [第 64 页中的“Java SE 升级过程”](#)
- [第 66 页中的“Common Agent Container 升级过程”](#)

Java SE 升级过程

Java ES 发行版本 5 经证实适用于 Java Platform, Standard Edition (Java SE) 版本 5.0 Update 9（此处标识为 Java SE 5.0 Update 9）。（Java SE 5.0 有时称为开发者版本 1.5.0）。

如同其他共享组件一样，Java SE 也由 Java ES 安装程序升级。不过，无论是否要安装或升级其他共享组件，安装程序都会将其升级到 Java SE 5.0 Update 9。因此，无需为安装 Java SE 5.0 Update 9 而显式选择“所有共享组件”。

注	升级 Java SE 时，可能要先关闭依赖于当前已安装 Java SE 的任何服务。这是为了避免因这些服务产生任何问题。如果不关闭依赖于 Java SE 的服务，则应在将 Java SE 升级到发行版本 5 版本后重新引导系统。
----------	--

当安装程序在计算机上检测到较旧版本的 Java SE 软件包或不完整的软件包集（完整的软件包集包括：SUNWj5rt、SUNWj5rtx、SUNWj5dev、SUNWj5dmo、SUNWj5dvx、SUNWj5man、SUNWj5cfg、SUNWj5dmx）时，将向您显示一个对话框。该对话框在组件选择页面之后出现，让您选择是自动将 Java SE 升级到发行版本 5 级别，还是绕过自动升级而改为手动执行升级。

- **手动升级。**如果选择手动升级 Java SE，请使用以下过程：

- a. 在安装或升级其他组件之前先退出 Java ES 安装程序。
- b. 手动安装适当版本的 Java SE。
- c. 重新启动 Java ES 安装程序。

如果安装程序检测到正确 / 完整版本的 Java SE，将允许您继续操作，否则将显示先前提及的对话框。

- **自动升级。**如果选择自动升级 Java SE，Java ES 安装程序会将 Java SE 升级到版本 5.0 Update 9。该升级具有以下行为：
 - 升级过程不会删除先前安装的主要 Java SE 发行版本（例如，Java SE 版本 1.4.2 或 Java SE 版本 1.6.x），因为其他应用程序可能会依赖该版本。但是，升级过程会设置一个符号链接（如下表所示），以引用发行版本 5 版本。

表 2-1 按平台而定的指向 Java SE 的符号链接

平台	符号链接	位置
Solaris	/usr/jdk/entsys-j2se	/usr/jdk/instances/jdk1.5.0
Linux	/usr/jdk/entsys-j2se	/usr/java/jdk1.5.0_09

对于需要早期版本的服务，应维护指向主要 Java SE 发行版本（例如 Java SE 1.4.2）的任何指针。有关如何维护指向早期版本 Java SE 的符号链接的信息，查阅相应的产品组件文档。

- 升级过程会删除先前安装的次要 Java SE 发行版本（例如，Java SE 版本 5.0 Update 5）并将其替换为 Java SE 5.0 Update 9。

检查 Java SE 符号链接

Java Enterprise System 会维护指向所支持 Java SE 平台版本的符号链接，以确保 Java ES 服务可以找到正确的 Java SE 运行时环境来使用。

按如下所示检查符号链接（例如，在 Solaris 操作系统上）：

```
ls -l /usr/jdk/entsys-j2se
lrwxrwxrwx 1 root  other  7 Jul 7 23:18 /usr/jdk/entsys-j2se ->
    /usr/jdk/instances/jdk1.5.0
```

其中，/usr/jdk/instances/jdk1.5.0 是默认位置。

检验当前 Java SE 版本

要确定 Java ES 安装所使用的 Java SE 的版本，运行以下命令，它将检验 Java SE 符号链接所引用的 Java SE 的版本：

```
/usr/jdk/entsys-j2se/bin/java -version
```

下表显示了输出结果。

表 2-2 Java SE 版本检验输出

Java ES 发行版本	Java SE 版本号	Java SE 版本字符串
发行版本 2	1.4.2 Update 5	1.4.2_05
发行版本 3	5.0 Update 1	1.5.0_01
发行版本 4	5.0 Update 4	1.5.0_04
发行版本 5	5.0 Update 9	1.5.0_09

Common Agent Container 升级过程

如果要升级 Common Agent Container 共享组件以为升级 Sun Cluster 软件（即修补版本 1.1）做准备，请遵循《Sun Cluster 软件安装指南（适用于 Solaris OS）》的“升级 Sun Cluster 软件”

(<http://docs.sun.com/app/docs/doc/819-2056/6n4cdufo1?l=zh&a=view>) 中的升级依赖软件过程。

有关上述指南的更新，参见《Sun Cluster 3.1 8/05 With Java Enterprise System 5 Special Instructions》(<http://docs.sun.com/doc/819-4351>) 中的 "Upgrading to Sun Cluster 3.1 8/05 Software"。

如果要升级 Common Agent Container 共享组件以为升级其他 Java ES 组件（即升级到版本 2.0）做准备，使用以下过程，路径名变量如下：

表 2-3 Common Agent Container 目录路径

路径名变量	Solaris OS	Linux OS
rel4CACbase-dir	/opt/SUNWcacao	/opt/sun/cacao
rel5CAC-admin-dir	/usr/lib/cacao	/opt/sun/cacao

1. 如果当前安装使用自定义配置设置，则使用以下命令捕获配置设置：

```
rel4CAC-base-dir/bin/cacoadm list-params
```

输出类似于以下形式：

```
java-flags=-Xms4M -Xmx64M
jmxmp-connector-port=10162
snmp-adaptor-port=10161
snmp-adaptor-trap-port=10162
commandstream-adaptor-port=10163
retries=4
```

上例显示的是默认值。注意第 67 页中的步骤 4 中使用的任何非默认设置。

2. 使用以下命令停止 Common Agent Container 进程：

```
rel4CAC-base-dir/bin/cacoadm stop
echo $?
```

如果退出代码不是 0，则强制停止：

```
rel4CAC-base-dir/bin/cacoadm stop -f
```

3. 使用 Java ES 安装程序的同步“所有共享组件”功能来升级 Common Agent Container。

参见第 63 页中的“共享组件升级过程”。

4. 应用先前在第 67 页中的步骤 1 中捕获的所有自定义配置设置。

```
rel5CAC-admin-dir/bin/cacoadm set-param java-flags=Value
rel5CAC-admin-dir/bin/cacoadm set-param jmxmp-connector-port=Value
rel5CAC-admin-dir/bin/cacoadm set-param snmp-adaptor-port=Value
rel5CAC-admin-dir/bin/cacoadm set-param
    snmp-adaptor-trap-port=Value
rel5CAC-admin-dir/bin/cacoadm set-param
    commandstream-adaptor-port=Value
rel5CAC-admin-dir/bin/cacoadm set-param retries=Value
```

5. 如果已将 Java SE 升级到 Java SE 版本 5，则运行 rebuild-dependencies 实用程序：

```
rel5CAC-admin-dir/bin/cacoadm rebuild-dependencies
```

此命令的输出将会是：

```
Property updated: [java-home].
Property updated: [jdkmk-home].
Property updated: [nss-lib-home].
Property updated: [nss-tools-home].
```

6. 重新启动 Common Agent Container 服务:

```
rel5CAC-admin-dir/bin/cacoadm stop  
rel5CAC-admin-dir/bin/cacoadm start
```

7. 检验 Common Agent Container 的升级:

```
rel5CAC-admin-dir/bin/cacoadm status  
rel5CAC-admin-dir/bin/cacoadm verify-configuration
```

Sun Cluster 软件

本章介绍如何将 Sun Cluster 软件升级到 Java ES 5（发行版本 5）：Sun Cluster 3.1 8/05 软件。

本章概述了将 Sun Cluster 软件升级到发行版本 5 的升级注意事项。

仅在 Solaris 平台上支持 Sun Cluster 软件。

本章所述的 Sun Cluster 软件的升级包括 Sun Cluster 框架软件和 Sun Cluster 数据服务软件（或代理）这两者的升级。

- [第 70 页中的“Sun Cluster 软件升级概述”](#)
- [第 73 页中的“将 Sun Cluster 软件升级到 Java ES 发行版本 5”](#)

Sun Cluster 软件升级概述

本节将在以下方面对 Sun Cluster 软件进行介绍，这些方面会对升级到 Java ES 5（发行版本 5）产生影响：

- [关于发行版本 5 Sun Cluster 软件](#)
- [Sun Cluster 软件升级指导说明](#)
- [Sun Cluster 数据](#)
- [Sun Cluster 升级策略](#)

关于发行版本 5 Sun Cluster 软件

相对于发行版本 4 Sun Cluster 软件而言，发行版本 5 Sun Cluster 软件只是一个次要升级（参见 Sun Cluster 发行说明

<http://docs.sun.com/doc/819-2084/6n4d8kv8c?l=zh&a=view>）。发行版本 5 Sun Cluster 在本质上与发行版本 4 相同，但发行版本 5 不支持 Solaris 8 操作系统。

有关 Sun Cluster 3.1 8/05 文档集的更改和新增内容，参见《Sun Cluster 3.1 8/05 With Java Enterprise System 5 Special Instructions》<http://docs.sun.com/doc/819-4351>。此外，Sun Cluster 3.1 8/05 文档集中的过程还适用于发行版本 5 Sun Cluster 软件。

注

如果 Solaris 8 OS 上需要 Sun Cluster 3.1 8/05 软件，则必须从 Java ES 发行版本 4 分发软件中获取 Sun Cluster 3.1 8/05 软件，其网址为 <http://www.sun.com/software/javaenterprisesystem/previous/index.xml>。

Java ES 5 安装程序不再支持在 Solaris 8 OS 上安装 Sun Cluster 3.1 8/05 软件。

要在 Solaris 8 OS 上安装 Sun Cluster 3.1 8/05 软件，请执行以下步骤：

1. 通过 Java ES 发行版本 4 分发软件安装 Sun Cluster 3.1 8/05 软件。
 2. 为 Sun Cluster 3.1 8/05 软件安装所有必需的修补程序。
-

Sun Cluster 软件升级指导说明

表 3-1 显示了所支持的、将 Sun Cluster 升级到 Java ES 5（发行版本 5）的升级方式。该表仅适用于 Solaris 操作系统。

Sun Cluster 版本与 Java ES 发行版本并非是一对一的映射关系。过去，Sun Cluster 软件的过渡功能版本 (interim feature releases, IFR) 已合并到 Java ES 正式发行版本之间的 Java ES 中。因此，如表 3-1 中所示，将 Java ES 发行版本 2、发行版本 3 和发行版本 4 Sun Cluster 升级到发行版本 5 Sun Cluster 包括将 Sun Cluster 3.1 4/04、Sun Cluster 3.1 9/04 和 Sun Cluster 3.1 8/05 软件升级到发行版本 5。

表 3-1 升级到 Java ES 5（发行版本 5）：Sun Cluster 3.1 8/05 软件的方式

Java ES 发行版本	Sun Cluster 软件版本	常规方法	必需的重新配置
发行版本 4	Sun Cluster 3.1 8/05 (2005Q4)	直接升级 ¹ ： 使用 Sun Cluster scinstall 实用程序执行。	无。
发行版本 3	Sun Cluster 3.1 9/04 或 Sun Cluster 3.1 8/05 (IFR)	直接升级： 使用 Sun Cluster scinstall 实用程序执行。	自动迁移到已升级版本的群集配置 将 IFR 升级到发行版本 5 时不需要重新配置。
发行版本 2	Sun Cluster 3.1 4/04 或 Sun Cluster 3.1 9/04 (IFR)	直接升级： 使用 Sun Cluster scinstall 实用程序执行。	自动迁移到已升级版本的群集配置
发行版本 1	Sun Cluster 3.1	直接升级未经验证： 但是可使用 scinstall 实用程序执行。	自动迁移到已升级版本的群集配置
Java ES 以前的发行版本	Sun Cluster 3.0	直接升级未经验证： 但是可使用 scinstall 实用程序执行。	自动迁移到已升级版本的群集配置

1. 除非 Sun Cluster 软件所使用的共享组件已升级到发行版本 5，否则没有必要从发行版本 4 升级到发行版本 5。

Sun Cluster 数据

下表显示了可能受 Sun Cluster 软件升级影响的数据类型。

表 3-2 Sun Cluster 数据用途

数据类型	位置	用途
群集配置数据	群集配置系统信息库，在所有群集节点范围内复制和同步（注意：切勿手动编辑 CCR 文件，这会导致一个节点或整个群集停止工作）	存储关于 Sun Cluster 所有操作方面的配置信息：群集节点配置、故障转移机制、资源管理等

Sun Cluster 升级策略

Sun Cluster 的升级策略通常取决于在第 1 章“升级规划”中讨论的许多注意事项：升级方式、Java ES 组件之间的依赖性、选择性升级与全部升级、多实例部署，等等。

本节通过提出可能影响 Sun Cluster 升级计划的问题，对 Sun Cluster 的一般注意事项进行逐一介绍。

兼容性问题

Java ES 发行版本 5 Sun Cluster 软件包括新的图形管理界面，但与早期发行版本的 Sun Cluster 代理向后兼容。

Sun Cluster 组件依赖性

Sun Cluster 对其他 Java ES 组件的依赖性会影响升级和重新配置 Sun Cluster 软件的过程。例如，Sun Cluster 接口或功能上的变化可能需要 Sun Cluster 软件所依赖的组件的已升级版本。是否需要升级此类组件取决于特定的升级方式。

Sun Cluster 对以下 Java ES 组件具有依赖性：

- **共享组件。** Sun Cluster 软件对特定的 Java ES 共享组件具有依赖性（参见第 43 页中的表 1-9）。
- **数据服务。** Sun Cluster 软件需要特定的数据服务（或代理），以使 Java ES 产品组件具有高可用性。对于在 Sun Cluster 环境中运行的每个产品组件，必须提供对应的数据服务来管理对应的群集资源。代理软件包通常在 Sun Cluster 升级过程中进行升级。

双重升级

在双重升级中，Sun Cluster 软件和操作系统都要进行升级（如第 41 页中的“双重升级：Java ES 和操作系统软件”中所述），双重升级可在 Sun Cluster 非滚动升级上下文中执行，而不能在滚动升级过程中执行。

有关该过程的详细信息，参见 Sun Cluster 安装指南的升级一章 (<http://docs.sun.com/app/docs/doc/819-2056/6n4cdufo7?l=zh&a=view>)。《Sun Cluster 3.1 8/05 With Java Enterprise System 5 Special Instructions》(<http://docs.sun.com/doc/819-4351>) 中说明了对发行版本 5 Sun Cluster 进行的修改。

该过程适用于将 Solaris 操作系统从 Solaris 8 或 Solaris 9 升级到 Solaris 10。

将 Sun Cluster 软件升级到 Java ES 发行版本 5

本节包含有关将 Sun Cluster 软件从 Java ES 2005Q4（发行版本 4）、Java ES 2005Q1（发行版本 3）和 Java ES 2004Q2（发行版本 2）升级到 Java ES 5（发行版本 5）的信息。上述 Java ES 发行版本中包含的三种 Sun Cluster 版本（Sun Cluster 3.1 4/04、Sun Cluster 3.1 9/04 和 Sun Cluster 3.1 8/05 软件）的升级过程相同。

如果已经安装了 Sun Cluster 3.1 8/05（发行版本 3 或发行版本 4）软件，则除非 Sun Cluster 软件所使用的共享组件已升级到发行版本 5，否则不必升级到发行版本 5。

本节涵盖了以下主题：

- [简介](#)
- [Sun Cluster 升级](#)

简介

在将 Sun Cluster 软件升级到 Java ES 发行版本 5 时，请考虑升级过程中的以下方面：

- **常规升级方法。** 通过运行 `scinstall` 脚本执行升级，该脚本将升级 Sun Cluster 软件，并在完成软件升级后应用先前的 Sun Cluster 配置。但必须将群集环境中的所有节点均升级到同一版本，方法有两种：一种是关闭群集然后升级所有节点，另外一种是通过滚动升级而不必关闭群集，期间将逐个升级各个节点，每次升级一个。
- **升级依赖性。** 虽然 Sun Cluster 软件对许多 Java ES 共享组件都具有依赖性（参见第 43 页中的表 1-9），但它们是软性升级依赖性：Java ES 发行版本 5 Sun Cluster 软件与这些组件的发行版本 4 的版本兼容。
- **向后兼容性。** 发行版本 5 Sun Cluster（框架）软件与早期的群集代理向后兼容，并且，发行版本 5 Sun Cluster 代理软件与 Java ES 发行版本 4 组件向后兼容。但是，一个群集中的所有节点必须运行同一版本的框架和代理软件。
- **升级回滚。** 不支持 Sun Cluster 软件的发行版本 5 到早期版本的升级回滚。
- **平台问题。** 在所有 Solaris 平台和硬件体系结构上，Sun Cluster 软件的升级方法都是相同的，但在 Linux 平台上不支持 Sun Cluster 软件。

Sun Cluster 升级

本节概述了如何执行 Sun Cluster 软件从 Java ES 发行版本 4 到 Java ES 发行版本 5 的升级：

- [升级前的任务](#)
- [升级 Sun Cluster 软件](#)
- [检验升级](#)
- [升级后的任务](#)
- [升级回滚](#)

本节涵盖了 Sun Cluster 非滚动升级的情况。滚动升级的情况略微有些不同，原因在于执行滚动升级时不关闭群集。但是如下所述，对于给定群集节点，这两种情况都包含了相同的常规过程。具体过程可在 [Sun Cluster 安装指南的升级一章](#)

(<http://docs.sun.com/app/docs/doc/819-2056/6n4cdufo1?l=zh&a=view>) 中找到。

《Sun Cluster 3.1 8/05 With Java Enterprise System 5 Special Instructions》(<http://docs.sun.com/doc/819-4351>) 中说明了对发行版本 5 Sun Cluster 进行的修改。

升级前的任务

升级 Sun Cluster 软件之前，应执行以下任务：

- [检验当前版本信息](#)
- [关闭 Sun Cluster Geographic Edition 基础结构](#)
- [为升级准备群集节点](#)
- [升级操作系统](#)
- [升级 Sun Cluster 依赖项](#)
- [获取所需的配置信息和密码](#)

检验当前版本信息

可通过输入以下命令检验 Sun Cluster 软件的当前版本：

```
scinstall -pv
```

该命令将返回 Sun Cluster 版本及所安装的各个软件包的版本。如果此命令返回 3.1 8/05 版本 3.1u4，则检查修补程序修订版本号，以确定拥有的是发行版本 4 还是发行版本 5 软件，如下表所示。

表 3-3 Sun Cluster 版本检验输出

Java ES 发行版本	Sun Cluster 版本号	修补程序修订版本号
发行版本 1 (Sun Cluster 3.1)	3.1	
发行版本 2 (Sun Cluster 3.1 4/04)	3.1u2	
发行版本 2 或 3 (Sun Cluster 3.1 9/04)	3.1u3	
发行版本 3 或 4 (Sun Cluster 3.1 8/05)	3.1u4	Solaris 9 sparc: 117949-15 Solaris 8 sparc: 117950-15 Solaris 9 x86: 117909-15
发行版本 5 (Sun Cluster 3.1 8/05)	3.1u4	Solaris 10 sparc: 120500-08 ¹ Solaris 9 sparc: 117949-23 Solaris 8 sparc: 117950-23 Solaris 10 x86: 120501-08 Solaris 9 x86: 117909-23

1. 修补程序修订版本号是在不同平台上升级到 Java ES 发行版本 5 所需的最低版本号。如果有较新的修订版本可用，请用较新版本替代该表中所示的版本。

关闭 Sun Cluster Geographic Edition 基础结构

如果要升级运行有 Sun Cluster Geographic Edition 软件的群集，必须先关闭 Sun Cluster Geographic Edition 基础结构，然后执行其他步骤以使群集做好升级准备。请遵循《Sun Cluster Geographic Edition Installation Guide》中的升级过程 (<http://docs.sun.com/doc/819-8004/6n9tmd19d?=view>)。这些过程包括在适当升级阶段升级 Sun Cluster 软件的步骤。

为升级准备群集节点

必须先从群集环境中删除群集节点，然后才能升级 Sun Cluster 软件：

- **非滚动升级。** 从群集环境中删除节点意味着关闭群集环境：将资源组切换到脱机状态、禁用资源组、关闭正在环境中运行的应用程序、备份共享数据、关闭群集、备份系统磁盘，然后重新引导节点，进入非群集模式。
- **滚动升级。** 从群集环境中删除节点意味着转移节点的所有资源组和设备组、备份共享数据和系统磁盘，然后重新引导节点，进入非群集模式。

有关这些操作以及可能需要在特定情况下执行的其他操作的详细信息，参见 Sun Cluster 安装指南的升级一章

(<http://docs.sun.com/app/docs/doc/819-2056/6n4cdufo1?l=zh&a=view>)。

升级操作系统

您可能想利用任何升级停机时间来将操作系统升级到其最新版本，同时也升级正在使用的卷管理器的版本。

有关这些操作的详细信息，参见 Sun Cluster 安装指南的升级一章

(<http://docs.sun.com/app/docs/doc/819-2056/6n4cdufo1?l=zh&a=view>)。

升级 Sun Cluster 依赖项

通常建议将计算机系统（以及计算环境）中的所有 Java ES 组件都升级到发行版本 5。Sun Cluster 软件所依赖的发行版本 4 共享组件的升级是可选的，并可按照第 2 章“升级 Java ES 共享组件”中所述的方式执行升级（而发行版本 2 共享组件到发行版本 5 的升级则是强制性的）。

注

如果 Java ES 共享组件已同步到发行版本 5，则 Sun Cluster 软件必须升级到发行版本 5。这是因为发行版本 4 Sun Cluster 与发行版本 5 Sun Java Web Console 共享组件不兼容。

如果 Sun Java Web Console 已升级到发行版本 5（版本 3.0），则应该应用最新的 Sun Cluster 修补程序或按照本章所述的方式升级 Sun Cluster。

获取所需的配置信息和密码

不需要任何与当前安装版本有关的特殊信息，但是必须以超级用户身份登录才能执行升级。

升级 Sun Cluster 软件

本节先讨论影响 Sun Cluster 软件升级过程的若干注意事项，随后对升级过程本身加以说明。

升级注意事项

将 Sun Cluster 软件升级到 Java ES 发行版本 5 时，应考虑以下注意事项：

- 在升级 Sun Cluster 框架软件时，最好也将管理高可用性 Java ES 组件所需的数据服务以及其他在群集环境中运行的应用程序升级。
- 升级 Sun Cluster 软件的同时也可以升级 Java ES 组件或者在群集环境中运行的其他应用程序。

升级过程

下面的过程适用于升级每个群集节点上的 Sun Cluster 软件。以下步骤只是概括性步骤，有关如何执行这些步骤的详细信息，参见 Sun Cluster 安装指南的升级一章 (<http://docs.sun.com/app/docs/doc/819-2056/6n4cduf01?l=zh&a=view>)。《Sun Cluster 3.1 8/05 With Java Enterprise System 5 Special Instructions》(<http://docs.sun.com/doc/819-4351>) 中说明了对发行版本 5 Sun Cluster 进行的修改。

1. 引导到非群集模式。
2. 以 root 身份登录，或成为超级用户。
3. 切换到 Java ES 发行版本 5 分发软件的以下目录：

```
su -
```

```
cd /os_arch/Product/sun_cluster/os-ver/Tools
```

其中，*os_arch* 与您的平台匹配（如 Solaris_sparc），*os-ver* 为 Solaris 9 或 Solaris 10。

4. 运行 scinstall 实用程序。

```
./scinstall
```

将显示用于执行群集安装、配置和升级任务的主菜单。

5. 升级 Sun Cluster 框架软件以及任何所需的数据服务。

已升级数据服务需要通过将对应的资源迁移到已升级资源类型来进行配置（参见第 78 页中的“升级后的任务”）。

6. 向 Sun Cluster 框架软件和数据服务应用所有必要的修补程序。

有关访问和应用相关修补程序的信息，参见《Sun Cluster 3.0-3.1 Release Notes Supplement》

(<http://docs.sun.com/app/docs/doc/816-3381/6m9lratq9?a=view#gcpom>)。

7. 重新引导节点，进入群集。

检验升级

可按如下步骤检验 Sun Cluster 软件的升级是否成功：

1. 输入以下命令。

```
scinstall -pv
```

该命令将返回 Sun Cluster 版本及所安装的各个软件包的版本。如果此命令返回 3.1 8/05 版本 3.1u4，则检查修补程序修订版本号，以确定拥有的是发行版本 4 还是发行版本 5 软件，如第 75 页中的表 3-3 所示。

2. 检查数据服务升级日志文件。

在升级输出消息的末尾提到了日志文件。

升级后的任务

在执行完 Sun Cluster 软件的升级后，可能还需要执行许多附加任务，这取决于您执行的是非滚动升级还是滚动升级。其中，完全恢复群集环境所需的任务有：

- 检验群集配置的状态
- 将资源迁移到新资源类型版本
- 升级安装在群集中的其他 Java ES 组件或者应用程序

有关这些安装后步骤的详细信息，参见 Sun Cluster 安装指南的升级一章

(<http://docs.sun.com/app/docs/doc/819-2056/6n4cdufo1?l=zh&a=view>)。

升级回滚

不支持 Sun Cluster 软件的回滚。升级过程中所做的更改无法轻易逆向恢复。

Sun Cluster Geographic Edition

本章介绍如何将 Sun Cluster Geographic Edition 升级到 Java ES 5（发行版本 5）：Sun Cluster Geographic Edition 3.1 2006Q4。

本章概要介绍了有关发行版本 5 所支持的不同升级方式的升级注意事项。其内容仅包括在 Solaris 操作系统上的升级：

- 第 80 页中的“Sun Cluster Geographic Edition 升级概述”
- 第 82 页中的“从版本 3.1 8/05 升级 Sun Cluster Geographic Edition”

注 本章中的文件位置是相对于一个称为 *SunClusterGeo-base* 的目录路径而指定的。在安装 Sun Cluster Geographic Edition 时，可能至少已将此路径的一部分指定为安装目录。否则，Geographic Edition 安装程序会分配一个默认值。

下表显示了此目录路径的默认值。

表 4-1 Sun Cluster Geographic Edition 目录路径

路径名变量	Solaris OS	Linux OS
<i>SunClusterGeo-base</i>	/opt/SUNWscgeo	不适用

Sun Cluster Geographic Edition 升级概述

本节将在以下方面对 Sun Cluster Geographic Edition 进行介绍，这些方面会对升级到 Java ES 5（发行版本 5）产生影响：

- [关于 Java ES 发行版本 5](#)
- [Java ES 发行版本 5 升级指导说明](#)
- [Sun Cluster Geographic Edition 数据](#)
- [Sun Cluster Geographic Edition 升级策略](#)

关于 Java ES 发行版本 5

Java ES 发行版本 5 Sun Cluster Geographic Edition 是第一个即将作为 Java ES 组件交付的发行版本；Sun Cluster Geographic Edition 3.1 首次是作为独立的产品发行的。

相对于 Sun Cluster Geographic Edition 3.1 8/05 而言，发行版本 5 Sun Cluster Geographic Edition 只是一个次要发行版本。它是第一个支持 Solaris x86 平台的发行版本。发行版本 5 Sun Cluster Geographic Edition 还修复了一些选定的错误，适合于各种硬件和软件组件，且支持其他数据复制产品。

Java ES 发行版本 5 升级指导说明

[表 4-2](#) 显示了所支持的、将 Sun Cluster Geographic Edition 升级到 Java ES 发行版本 5 的升级方式。该表仅适用于 Solaris 操作系统。

表 4-2 升级到 Java ES 5（发行版本 5）：Sun Cluster Geographic Edition 3.1 2006Q4 的方式

Java ES 发行版本	Sun Cluster Geographic Edition 版本	常规方法	必需的重新配置
Java ES 以前的发行版本	Sun Cluster Geographic Edition 3.1 8/05	直接升级： 用全新安装替代 Sun Cluster Geographic Edition 3.1 8/05	无

Sun Cluster Geographic Edition 数据

下表显示了可能受 Sun Cluster Geographic Edition 软件升级影响的数据类型。

表 4-3 Sun Cluster Geographic Edition 数据用途

数据类型	位置	用途
Sun Cluster Geographic Edition 配置数据	群集配置系统信息库，在所有群集节点范围内复制和同步（注意：切勿手动编辑 CCR 文件，这会导致一个节点或整个群集停止工作）	存储 Sun Cluster Geographic Edition 操作所有方面的配置信息。

Sun Cluster Geographic Edition 升级策略

Sun Cluster Geographic Edition 的升级策略通常取决于在第 1 章“升级规划”中讨论的许多注意事项：升级方式、Java ES 组件之间的依赖性、选择性升级与全部升级、多实例部署，等等。

本节通过提出可能影响 Sun Cluster Geographic Edition 升级计划的问题，对 Sun Cluster Geographic Edition 的一般注意事项进行逐一介绍。

兼容性问题

伙伴关系中的两个群集必须运行相同版本的 Sun Cluster Geographic Edition。因此，发行版本 5 Sun Cluster Geographic Edition 不能与 Sun Cluster Geographic Edition 3.1 8/05（无向后兼容性）并存于伙伴关系中。发行版本 5 Sun Cluster Geographic Edition 包含 Sun Cluster Geographic Edition 3.1 8/05 所无法读取的附加配置数据。

组件依赖性

Sun Cluster Geographic Edition 对其他 Java ES 组件的依赖性会影响升级和重新配置 Sun Cluster Geographic Edition 软件的过程。例如，Sun Cluster Geographic Edition 接口或功能上的变化可能要求 Sun Cluster Geographic Edition 所依赖组件的升级版本。是否需要升级此类组件取决于具体的升级方式。

Sun Cluster Geographic Edition 对以下 Java ES 组件具有依赖性：

- **共享组件。** Sun Cluster Geographic Edition 对特定的 Java ES 共享组件具有依赖性（参见第 43 页中的表 1-9）。
- **Sun Cluster。** Sun Cluster Geographic Edition 对 Sun Cluster 具有强制依赖性，后者提供基本功能。

双重升级

在双重升级中，Sun Cluster Geographic Edition 软件和操作系统都要进行升级（如第 41 页中的“双重升级：Java ES 和操作系统软件”中所述），但双重升级不适用于 Sun Cluster Geographic Edition。

由于 Sun Cluster Geographic Edition 对 Sun Cluster 具有硬性升级依赖性，因此，操作系统升级是在 Sun Cluster 升级上下文中执行的。有关 Sun Cluster 双重升级的信息，参见第 72 页中的“双重升级”。

从版本 3.1 8/05 升级 Sun Cluster Geographic Edition

本节包含关于将 Sun Cluster Geographic Edition 从版本 3.1 8/05 升级到 Java ES 5（发行版本 5）的信息。其中涵盖了以下主题：

- [简介](#)
- [版本 3.1 8/05 Sun Cluster Geographic Edition 升级](#)

简介

在将 Sun Cluster Geographic Edition 3.1 8/05 升级到发行版本 5 时，请考虑升级过程中的以下方面：

- **常规升级方法。** 要实现升级，先卸载版本 3.1 8/05，然后执行发行版本 5 Sun Cluster Geographic Edition 的全新安装。配置数据将保留。
- **升级依赖性。** Sun Cluster Geographic Edition 对许多 Java ES 共享组件都具有依赖性（参见第 43 页中的表 1-9），所有这些组件会在您升级 Sun Cluster Geographic Edition 时由 Java ES 安装程序将它们自动升级到发行版本 5。需特别指出的是，Sun Cluster Geographic Edition 对 Common Agent Container（版本 1.1）具有硬性升级依赖性。Sun Cluster Geographic Edition 还对 Sun Cluster 具有硬性升级依赖性：发行版本 4 Sun Cluster (3.1 8/05) 必须升级到发行版本 5 才能支持发行版本 5 Sun Cluster Geographic Edition。
- **向后兼容性。** 发行版本 5 Sun Cluster Geographic Edition 与 Sun Cluster Geographic Edition 3.1 8/05 不兼容。具有伙伴关系的所有群集都必须升级到发行版本 5。
- **升级回滚。** 不支持从发行版本 5 Sun Cluster Geographic Edition 到版本 3.1 8/05 的升级回滚。
- **平台问题。** 在所有 Solaris 平台和硬件体系结构上，Sun Cluster Geographic Edition 软件的升级方法都是相同的，但在 Linux 平台上不支持 Sun Cluster Geographic Edition 软件。

版本 3.1 8/05 Sun Cluster Geographic Edition 升级

本节介绍如何将 Sun Cluster Geographic Edition 从版本 3.1 8/05 升级到 Java ES 发行版本 5。其中涵盖了以下主题：

- [升级前的任务](#)
- [升级 3.1 8/05 Sun Cluster Geographic Edition](#)
- [检验升级](#)
- [升级后的任务](#)
- [升级回滚](#)

升级前的任务

升级 Sun Cluster Geographic Edition 软件之前，应执行以下任务：

- [检验当前版本信息](#)
- [为升级准备群集](#)
- [升级 Sun Cluster Geographic Edition 依赖项](#)
- [备份 Sun Cluster Geographic Edition 数据](#)
- [获取所需的配置信息和密码](#)

检验当前版本信息

可使用以下命令检验 Sun Cluster Geographic Edition 的当前版本：

```
/usr/cluster/bin/geoadm --version
```

版本字符串在版权文字之后出现，下表显示了对应的版本号：

表 4-4 Sun Cluster Geographic Edition 版本检验输出

Java ES 发行版本	Sun Cluster Geographic Edition 版本号
版本 3.1 8/05	1.0
发行版本 5	1.1

为升级准备群集

对于与您要升级的群集构成伙伴关系的所有群集，请执行以下步骤。此过程会从生产环境中删除 Sun Cluster Geographic Edition 层。

1. 确保群集正常起作用。

要查看群集的当前状态，可从任何节点运行以下命令。

```
scstat
```

2. 以 root 身份登录，或成为超级用户。

```
su -
```

3. 从保护组中删除所有应用程序资源组。

在 Sun Cluster Geographic Edition 升级期间，高可用性应用程序不会停机。

```
geopg remove-resource-group resourcegroup protectiongroupname
```

4. 对于与此群集构成伙伴关系的所有群集，执行步骤 1 到步骤 3。

5. 停止在该群集中处于活动状态的所有保护组。

```
geopg stop protectiongroupname -e local | global
```

6. 从两个伙伴群集的所有心跳 (heartbeat) 中删除 ICRM 插件。

```
geohp remove-plugin icrmplugin heartbeatname
```

升级 Sun Cluster Geographic Edition 依赖项

通常建议将计算机系统（以及计算环境）中的所有 Java ES 组件都升级到 Java ES 发行版本 5。Sun Cluster Geographic Edition 对许多共享组件以及 Sun Cluster 具有硬性升级依赖性。

升级 Sun Cluster Geographic Edition 的依赖项时，应在升级 Sun Cluster Geographic Edition 之前按以下顺序执行（跳过任何可能已升级的依赖项）。共享组件的升级通常是由 Java ES 安装程序自动实现的。

1. **共享组件。** 第 59 页中的“升级 Java ES 共享组件”中提供了将 Java ES 共享组件同步到发行版本 5 的说明。不过，在将 Sun Cluster Geographic Edition 升级到发行版本 5 时，Java ES 安装程序会自动升级 Sun Cluster Geographic Edition 所需的所有共享组件。

注

如果 Java ES 共享组件已升级到发行版本 5，则必须将 Sun Cluster Geographic Edition 软件也升级到发行版本 5。这是因为 Sun Cluster Geographic Edition 3.1 8/05 与发行版本 5 Sun Java Web Console 不兼容。

2. **Sun Cluster。** 第 69 页中的第 3 章“Sun Cluster 软件”中提供了将 Sun Cluster 升级至发行版本 5 的说明。

备份 Sun Cluster Geographic Edition 数据

Sun Cluster Geographic Edition 将所有数据存储在群集配置系统信息库中，因此，无需备份当前数据。

获取所需的配置信息和密码

不需要任何与当前安装版本有关的特殊信息，但是必须以超级用户身份登录才能执行升级。

升级 3.1 8/05 Sun Cluster Geographic Edition

本节讨论影响 Sun Cluster Geographic Edition 升级过程的一些注意事项，随后对升级过程本身加以说明。

升级注意事项

将 Sun Cluster Geographic Edition 软件升级到 Java ES 发行版本 5 不会干扰正在运行的应用程序。您可在运行中的群集上升级 Sun Cluster Geographic Edition 软件而不会出现中断；由于服务在运行，群集仍保持处于生产状态。同样，还可在不停机的情况下应用 Sun Cluster Geographic Edition 修补程序。配置数据在整个升级过程中将得到保留。

必须在与您要升级的群集构成伙伴关系的所有群集的所有节点上升级 Sun Cluster Geographic Edition 软件。

如果想要在 Sun Cluster Geographic Edition 软件升级过程中升级 Solaris 操作系统，则必须先删除 Sun Cluster Geographic Edition 软件包，然后再升级 Solaris 操作系统。

升级过程

下述过程适用于本地驻留在即将进行升级的计算机上的 Sun Cluster Geographic Edition 实例。

1. 以 root 身份登录，或成为超级用户。

```
su -
```

2. 关闭 Sun Cluster Geographic Edition 3.1 8/05。

```
/usr/cluster/bin/geoadm stop
```

要在关闭 Sun Cluster Geographic Edition 的同时使底层应用程序处于运行状态，参见《Sun Cluster Geographic Edition System Administration Guide》(<http://docs.sun.com/doc/819-8003>)。

3. 卸载 Sun Cluster Geographic Edition 3.1 8/05。

SunClusterGeo-base/install/uninstall/uninstaller

在卸载模式下使用 Sun Cluster Geographic Edition 安装程序，如《Sun Cluster Geographic Edition Installation Guide》(<http://docs.sun.com/doc/819-8004>) 中所述。

4. 如果 Sun Cluster 未运行，请将其启动。

Sun Cluster 必须处于运行状态才能安装 Sun Cluster Geographic Edition。可使用 `scstat` 命令对此进行检查。

5. 使用 Java ES 安装程序执行发行版本 5 Sun Cluster Geographic Edition 的全新安装。

- a. 在托管发行版本 4 Access Manager 的计算机上启动 Java ES 安装程序。

```
cd Java ES Release 5 distribution/os_arch
./installer
```

其中，*os_arch* 与您的平台匹配，如 *Solaris_sparc*。（对于命令行界面，使用 `installer -nodisplay` 选项。）

显示“欢迎”和“许可协议”页面之后，将向您显示一个组件选择页面。（如果检测到可由 Java ES 安装程序直接升级的已安装组件，它们的状态会显示为“可升级”。）

- b. 从组件选择页面中选择 Sun Cluster Geographic Edition。

- c. 选择“现在配置”或“以后再配置”。

是选择“现在配置”还是“以后再配置”并无差别，因为不需要任何重新配置。

- d. 如果需要，选择安装本地化软件包的选项。

- e. 安装完成后退出 Java ES 安装程序。

注	如果您要在 Solaris 8 OS 上升级 Sun Cluster Geographic Edition 软件，必须使用 <code>pkgadd</code> 命令通过 Java ES 发行版本 5 分发软件安装该软件。Java ES 安装程序不支持在 Solaris 8 OS 上安装 Sun Cluster Geographic Edition 软件。
----------	--

6. 安装所有必需的 Sun Cluster Geographic Edition 修补程序。

有关说明，参见《Sun Cluster Geographic Edition Installation Guide》(<http://docs.sun.com/doc/819-8004>)。

7. 对于与此群集构成伙伴关系的所有群集，执行第 85 页中的步骤 1 到步骤 6。
8. 重新启动发行版本 5 Sun Cluster Geographic Edition 软件。

```
/usr/cluster/bin/geoadm start
```

如果群集处于伙伴关系中，则必须先升级两个伙伴上的所有节点，然后再启动 Sun Cluster Geographic Edition 软件。

有关更多信息，参阅上文提到的安装指南以及《Sun Cluster Geographic Edition System Administration Guide》(<http://docs.sun.com/doc/819-8003>)。

9. 添加第 83 页中的“为升级准备群集”中所述的、准备群集升级时所删除的所有应用程序资源组。

```
geopg add-resource-group resourcegroup protectiongroupname
```

10. 启动已添加的所有保护组。

```
geopg start protectiongroupname -e local | global [-n]
```

检验升级

可按如下步骤检验 Sun Cluster Geographic Edition 软件的升级是否成功：

1. 运行以下命令：

```
/usr/cluster/bin/geoadm --version
```

参见第 83 页中的表 4-4 确认输出值。

2. 两个伙伴都升级后，在每个伙伴的一个群集节点上运行以下命令（如适用）。

```
/usr/cluster/bin/geoadm start
```

```
/usr/cluster/bin/geoadm show
```

该命令将显示 Sun Cluster Geographic Edition 软件在该节点上是否处于活动状态。

升级后的任务

除第 85 页中的“升级过程”中所述的步骤之外，没有任何升级后的任务。

升级回滚

不支持 Sun Cluster Geographic Edition 软件的回滚操作。

但是，可以使用 Java ES 安装程序的卸载功能删除发行版本 5 Sun Cluster Geographic Edition，然后使用版本 3.1 8/05 安装程序重新安装版本 3.1 8/05。由于发行版本 5 Sun Cluster Geographic Edition 会写入版本 3.1 8/05 所无法读取的数据，因此，任何此类回滚必须在启动发行版本 5 之前（即在启动已升级的 Sun Cluster Geographic Edition 软件之前）执行，如第 85 页中的“升级过程”中所述。

Directory Server

本章介绍如何将 Directory Server 升级到 Java ES 5（发行版本 5）：Sun Java System Directory Server 6.0。

本章概要介绍了有关发行版本 5 所支持的不同升级方式的升级注意事项。其内容涵盖了在 Solaris 和 Linux 操作系统上的升级：

- [第 90 页中的“Directory Server 升级概述”](#)
- [第 94 页中的“从 Java ES 发行版本 4 升级 Directory Server”](#)
- [第 102 页中的“从 Java ES 发行版本 3 升级 Directory Server”](#)
- [第 103 页中的“从 Java ES 发行版本 2 升级 Directory Server”](#)

注 本章中的文件位置是相对于两个目录路径指定的，分别称为 *serverRoot* (Directory Server 5.x) 和 *DirServer-base* (Directory Server 6.0)。在安装 Directory Server 时，可能至少已将这两个路径的一部分指定为安装目录，否则，Java ES 安装程序会分配一个默认值。

下表说明了这些目录路径的默认值。

表 5-1 Directory Server 目录路径

路径名变量	Solaris OS	Linux OS
<i>serverRoot</i> (Directory Server 5.x)	/var/opt/mps/serverroot	/var/opt/sun/directory-server
<i>DirServer-base</i> (Directory Server 6.0)	/opt/SUNWdsee	/opt/sun

Directory Server 升级概述

本节将在以下方面对 Directory Server 进行介绍，这些方面会对升级到 Java ES 5（发行版本 5）产生影响：

- [关于 Java ES 发行版本 5](#)
- [Java ES 发行版本 5 升级指导说明](#)
- [Directory Server 数据](#)
- [Directory Server 升级策略](#)

关于 Java ES 发行版本 5

Java ES 发行版本 5 Directory Server 是一个主要发行版本，它新增了许多功能并做了许多改进。有关详细信息，参见 Directory Server Enterprise Edition 6 发行说明 (<http://docs.sun.com/doc/820-0289>)。

Java ES 发行版本 5 升级指导说明

表 5-2 显示了所支持的、将 Directory Server 升级到 Java ES 发行版本 5 的升级方式。此表既适用于 Solaris，也适用于 Linux 操作系统。

表 5-2 升级到 Java ES 5（发行版本 5）：Directory Server 6.0 的方式

Java ES 发行版本	Directory Server 版本	常规方法	必需的重新配置
发行版本 4	Sun Java System Directory Server 5.2 2005Q4	直接升级： 执行全新安装并迁移所有数据。	将配置数据从先前的版本迁移到新安装的 Directory Server
	Sun Java System Administration Server 5.2 2005Q4	由 Directory Service Control Center 和 Directory Server EE 命令行实用程序替代 Administration Server 的功能。	
发行版本 3	Sun Java System Directory Server 5 2005Q1	直接升级： 执行全新安装并迁移所有数据。	将配置数据从先前的版本迁移到新安装的 Directory Server
	Sun Java System Administration Server 5 2005Q1	由 Directory Service Control Center 和 Directory Server EE 命令行实用程序替代 Administration Server 的功能。	

表 5-2 升级到 Java ES 5（发行版本 5）：Directory Server 6.0 的方式（续）

Java ES 发行版本	Directory Server 版本	常规方法	必需的重新配置
发行版本 2	Sun Java System Directory Server 5.2 2004Q2	直接升级： 执行全新安装并迁移所有数据。	将配置数据从先前的版本迁移到新安装的 Directory Server
	Sun Java System Administration Server 5.2 2004Q2	由 Directory Service Control Center 和 Directory Server EE 命令行实用程序替代 Administration Server 的功能。	
发行版本 1	Sun ONE Directory Server5.2	直接升级未经验证： 但可以使用发行版本 2 的升级方法。	将配置数据从先前的版本迁移到新安装的 Directory Server
	Sun ONE Administration Server 5.2		
Java ES 以前的发行版本	Sun ONE Directory Server5.2	直接升级未经验证： 但可以使用发行版本 2 的升级方法。	将配置数据从先前的版本迁移到新安装的 Directory Server
	Sun ONE Administration Server 5.2		
	Sun ONE Directory Server5.1	不可直接升级： 首先升级到发行版本 3。参阅 Java Enterprise System 2005Q1 升级与迁移指南	参阅 Java Enterprise System 2005Q1 升级与迁移指南 (http://docs.sun.com/doc/819-2237)。
	Sun ONE Administration Server 5.1	(http://docs.sun.com/doc/819-2237)。 然后从发行版本 3 升级到发行版本 5。	

Directory Server 数据

Directory Server 5.x 版本利用 Directory Server 本身来存储配置数据。数据存储在目录内部的一个特定树结构中。托管配置的 Directory Server 实例称为配置目录。配置目录可以与其他 Directory Server 实例驻留在同一台计算机上；但在大多数部署体系结构中，配置目录都会远离使用它来存储配置信息的其他组件。

Directory Server 6.0 不再将配置数据存储于配置目录中。使用 Directory Service Control Center（或 Directory Server EE 命令行实用程序）来执行配置，并且，应该通过此接口进行访问。Directory Service Control Center 将配置数据存储在其自身的本地 Directory Server 实例中。

下表显示了在 Directory Server 软件升级到发行版本 5 时受其影响的数据类型。

表 5-3 Directory Server 数据用途

数据类型	位置	用途
Directory Server 配置数据	Directory Server 5.1 和 5.2: 配置目录 Directory Server 6.0: 通过 Directory Service Control Center 和 Directory Server EE 命令行实用程序访问	Directory Server 实例的配置
Directory Server 模式		用于定义目录中数据的结构和语义
安全性数据	Directory Server 5.1 和 5.2: 通过 Directory Server 控制台配置 SSL。 Directory Server 6.0: 通过 Directory Service Control Center 和 Directory Server EE 命令行实用程序配置 SSL。	服务器证书
用户数据	Directory Server	通过用户特定的配置数据和用户配置文件支持应用程序

Directory Server 升级策略

Directory Server 的升级策略通常取决于在第 1 章“升级规划”中讨论的许多注意事项：升级方式、Java ES 组件之间的依赖性、选择性升级与全部升级、多实例部署，等等。

本节通过提出可能影响 Directory Server 升级计划的问题，对 Directory Server 的一般注意事项进行逐一介绍。

兼容性问题

Java ES 发行版本 5 Directory Server 未引入新的公共接口，因此与早期版本向后兼容；它支持发行版本 4 Directory Server 及早期版本所支持的所有组件。

但是，发行版本 5 引入了对专用管理接口的更改。发行版本 5 的接口与早期发行版本的 Directory Server 不兼容。需特别指出的是，用于配置早期 Directory Server 实例的 Administration Server 已被 Directory Service Control Center 和 Directory Server EE 命令行实用程序所取代，并删掉了用于存储 Directory Server 配置信息的 o=NetscapeRoot 目录后缀。有关详细信息，参阅 Directory Server Enterprise Edition 6 Migration Guide (<http://docs.sun.com/doc/819-0994>)。

组件依赖性

通常情况下，对其他 Java ES 组件的依赖性会影响 Directory Server 软件的升级过程。

Directory Server 对以下 Java ES 组件具有依赖性：

- **共享组件。** Directory Server 对特定的 Java ES 共享组件具有依赖性（参见第 43 页中的表 1-9）。Directory Server 升级可能要依赖于这些共享组件的已升级版本。
- **Directory Proxy Server。** Directory Server 对用于增进 LDAP 请求安全性和性能的 Directory Proxy Server 具有协作依赖性。

双重升级

在双重升级中，Directory Server 和操作系统都要进行升级（如第 41 页中的“双重升级：Java ES 和操作系统软件”中所述）。可使用以下两种方式之一执行双重升级：

- 操作系统全新安装
- 操作系统就地升级

操作系统全新安装

1. 备份现有 Directory Server 数据。

有关 Directory Server 5.x 信息，参见第 96 页中的“创建 Directory Server 映像（可选）”。

2. 安装新的操作系统。

可将操作系统安装在新系统上（或 Solaris 10 区域中），否则会擦除现有文件系统。

3. 恢复在步骤 1 中备份的 Directory Server 数据。
4. 安装发行版本 5 Directory Server。
5. 创建一个发行版本 5 Directory Server 实例，并将目录数据迁移到该新实例中。

参见第 96 页中的“升级发行版本 4 Directory Server”过程中的相关步骤。

操作系统就地升级

1. 备份现有 Directory Server 数据。

有关 Directory Server 5.x 信息，参见第 96 页中的“创建 Directory Server 映像（可选）”。

2. 升级操作系统。

升级会将现有文件系统留在原来的位置。

3. 将 Directory Server 升级到发行版本 5。

参见本章中的相关小节，这要视升级方式而定。

从 Java ES 发行版本 4 升级 Directory Server

本节包含关于将 Directory Server 从 Java ES 2005 Q4（发行版本 4）升级到 Java ES 5（发行版本 5）的信息。其中涵盖了以下主题：

- [简介](#)
- [发行版本 4 Directory Server 的升级](#)
- [多实例升级](#)

简介

将 Java ES 发行版本 4 Directory Server 升级到发行版本 5 时，请考虑升级过程中的以下方面：

- **常规升级方法。** 通过执行发行版本 5 Directory Server 的全新安装，然后使用迁移工具在截然不同的全新发行版本 5 Directory Server 实例中重新创建以前的 Directory Server 实例来实现升级。
- **升级依赖性。** Directory Server 对许多 Java ES 共享组件都具有依赖性（参见 [第 43 页中的表 1-9](#)），所有这些组件会在您升级 Directory Server 时，由 Java ES 安装程序自动将它们升级到发行版本 5。
- **向后兼容性。** 发行版本 5 Directory Server 与发行版本 4 的版本不向后兼容，如 [第 92 页中的“兼容性问题”](#) 中所述。但是，如果您要继续保留一组依赖于 Directory Server 5.x 管理框架的 Directory Server 实例，则可以使用迁移工具来迁移 o=NescapeRoot 后缀。
- **升级回滚。** 发行版本 5 的升级回滚通过恢复到前一个版本来实现，该版本在升级到发行版本 5 时会保持不变。
- **平台问题。** 在 Solaris 和 Linux 操作系统上，升级 Directory Server 的常规方法相同。

发行版本 4 Directory Server 的升级

本节介绍如何在 Solaris 和 Linux 平台上将 Directory Server 从 Java ES 发行版本 4 升级到 Java ES 发行版本 5。如果其中某个主题取决于具体平台的过程，则该主题将指明所适用的操作系统。本节包括以下主题：

- [升级前的任务](#)
- [升级发行版本 4 Directory Server](#)
- [检验升级](#)
- [升级后的任务](#)
- [升级回滚](#)

升级前的任务

升级 Directory Server 软件之前，应执行以下任务：

- [检验当前版本信息](#)
- [升级 Directory Server 依赖项](#)
- [获取所需的配置信息和密码](#)
- [创建 Directory Server 映像（可选）](#)

检验当前版本信息

可通过使用 -v 选项重新启动 Directory Server 守护进程来检验 Directory Server 的当前版本：

```
cd serverRoot/bin/slapd/server
./ns-slapd -v
```

表 5-4 Directory Server 版本检验输出

Java ES 发行版本	Directory Server 版本号
发行版本 2	Sun Java (TM) System Directory Server/5.2_Patch_2
发行版本 3	Sun Java (TM) System Directory Server/5.2_Patch_3
发行版本 4	Sun Java (TM) System Directory Server/5.2_Patch_4
发行版本 5	Sun Java (TM) System Directory Server/6.0

如果 ns-slapd 命令在 Solaris 10 平台上失败，请在运行该命令时将库路径设置为空：

```
LD_LIBRARY_PATH= ./ns-slapd -v
```

升级 Directory Server 依赖项

通常建议将计算机系统（以及计算环境）中的所有 Java ES 组件都升级到 Java ES 发行版本 5。但是，在执行 Directory Server 到发行版本 5 的升级时，将自动升级 Directory Server 所需的所有共享组件。

获取所需的配置信息和密码

您应知道当前安装版本的 Directory Server 管理员用户 ID 和密码。其他配置信息会在升级过程中得到保留。

创建 Directory Server 映像（可选）

如果要安装发行版本 5 Directory Server 的计算机不同于发行版本 4 的版本所驻留的计算机，应在要安装发行版本 5 Directory Server 的计算机上创建发行版本 4 版本的映像。需要该映像来实现数据到新的发行版本 5 Directory Server 实例的自动迁移（使用 `dsmig` 命令）。

发行版本 4 映像包括所有模式文件、配置文件、安全文件和数据库文件，其布局与原始 Directory Server 5.x *serverRoot* 文件结构完全相同。需要该映像才能执行数据到新的发行版本 5 Directory Server 实例的迁移。

升级发行版本 4 Directory Server

本节讨论影响 Directory Server 升级过程的一些注意事项，随后对升级过程本身加以说明。

升级注意事项

将 Directory Server 软件升级到 Java ES 发行版本 5 时，应考虑以下注意事项：

- 应将使用 Directory Server 实例的所有 Java ES 组件（如 Access Manager 或 Portal Server，或者 Sun Java Communications Suite 组件）全部关闭，并在需要时将它们重新配置为访问所对应的、新的发行版本 5 实例。
- 在其中有多个 Directory Server 实例在一台计算机上运行（所有实例均对应于同一已安装的 Directory Server 映像）的部署体系结构中，您只要必须升级 Directory Server 映像一次；但是，您必须分别迁移每个实例的数据。
- 在许多发行版本 4 Directory Server 部署体系结构中，配置目录是单独的 Directory Server 实例。这些实例是不需要升级的，因为配置目录在发行版本 5 中已被废弃。另一方面，升级可能会要求将发行版本 5 Directory Server 管理控制台 (Directory Service Control Center) 部署到单独的计算机上，您将通过它远程管理 Directory Server 实例。

- Directory Server 随附了一个命令行工具，该工具有助于实现模式数据、配置数据、安全性数据和用户数据的自动迁移。通过该迁移工具可一步一步地迁移这些不同的数据。即使是仅自动迁移了其中的一部分数据，大部分升级方案也会从中获益。

升级过程

下述过程适用于在即将升级的计算机上本地驻留的 Directory Server 实例，如果实例要移动到另一台计算机上，则适用于将在该目标计算机上运行的所有实例。

1. 以 root 身份登录，或成为超级用户。

```
su -
```

2. 关闭发行版本 4 Directory Server (5.2) 实例。

```
serverRoot/slapd-instanceName/stop-slapd
```

检查错误日志 (*serverRoot/slapd-hostName/logs/errors*) 是否报告已彻底关闭：

```
[23/Jan/2006:15:56:47 +0100] - All database threads now stopped

[23/Jan/2006:15:56:50 +0100] - slapd stopped.
```

3. 确保发行版本 5 Directory Server 的主机有足够的磁盘空间。

基本计算方法如下：

$2 * (\text{用于现有服务器的空间}) + (\text{用于 LDIF 文件的空间})$

遗憾的是，没有任何工具可用来预见从导出数据库创建的 LDIF 文件的大小。该大小将取决于数据条目的数目、其内部表示法、索引的数目，等等。

4. 对于发行版本 5 的远程安装，创建发行版本 4 映像并将其传送到相应的远程计算机。

参见第 96 页中的“创建 Directory Server 映像（可选）”。

5. 确保已升级 Directory Server 对其具有硬性升级依赖性的所有 Java ES 组件（参见第 96 页中的“升级 Directory Server 依赖项”）。

6. 执行发行版本 5 Directory Server 的全新安装。

执行以下步骤：

a. 启动 Java ES 安装程序。

```
cd Java ES Release 5 distribution/os_arch  
  
./installer
```

其中，*os_arch* 与您的平台匹配，如 *Solaris_sparc*。（对于命令行界面，使用 `installer -nodisplay` 选项。）

显示“欢迎”和“许可协议”页面之后，将向您显示一个组件选择页面。（如果检测到可由 Java ES 安装程序直接升级的已安装组件，它们的状态会显示为“可升级”。）

b. 选择 Directory Server Enterprise Edition 的 Directory Server 子组件。

还需要安装您想要使用的管理子组件（Directory Service Control Panel 或命令行实用程序）。

c. 指定一个不同于任何现有发行版本 4 Directory Server 的安装路径。

d. 选择“现在配置”或“以后再配置”。

选择“现在配置”还是“以后再配置”并不重要，因为实际上 Directory Server 不需要任何配置。但是，如果选择“现在配置”，**不要**选择创建新实例。

e. 确认您的安装选择。

将升级 Directory Server 软件包并显示升级摘要。

f. 退出 Java ES 安装程序。

7. 创建 Directory Server 实例。

```
DirServer-base/ds6/bin/dsadm create instancePath
```

其中，*instancePath* 是 Directory Server 实例的完整路径。

有关创建 Directory Server 实例的信息，参见 Directory Server Enterprise Edition 6 管理指南 (<http://docs.sun.com/doc/820-0293>)。

如果未能创建新实例，则在您使用 dsmig 命令迁移数据时将自动为您创建新实例（[步骤 8](#)）。

如果 dsadm 命令在 Solaris 10 平台上失败，请在运行该命令时将库路径设置为空：

```
LD_LIBRARY_PATH= ./dsadm create instancePath
```

8. 将发行版本 4 数据迁移到发行版本 5 Directory Server 实例。

使用 *DirServer-base*/ds6/bin/dsmig 命令。

dsmig 命令将修改发行版本 4 数据，使其适应发行版本 5 格式并将其写入适当的位置。例如，在具有一个 Directory Server 实例的单台计算机上进行的典型迁移可能会如下所示：

```
dsmig migrate-schema -v old_instancePath new_instancePath

dsmig migrate-config -v old_instancePath new_instancePath

dsmig migrate-security -v old_instancePath new_instancePath

dsmig migrate-data -v old_instancePath new_instancePath
```

注

- 如果 `dsmig migrate-config` 命令在 Solaris 10 平台上失败，请在运行该命令时将库路径设置为空：

`LD_LIBRARY_PATH= ./dsmig migrate-config ...`
 - 如果您要迁移的 Directory Server 实例中存储着其他 Java ES 组件（例如 Sun Java Communications Suite Messaging Server 组件）的配置数据，则可能必须迁移名为 `o=netscaperoot` 的目录信息树的一个特定部分。默认情况下，将不会迁移此根后缀。要迁移 `o=netscaperoot`，请使用 `dsmig migrate-config` 和 `dsmig migrate-data` 命令的 `-N` 选项。例如：

`dsmig migrate-config -v old_instancePath
new_instancePath -N`
 - 如果要从 32 位体系结构上的实例迁移到 64 位体系结构上的实例，则不能使用 `dsmig migrate-data` 命令（自动迁移工具）。您必须手动迁移数据，如下文提到的 **Migration Guide** 中所述。但是，您仍然可以自动迁移模式数据、配置数据和安全数据。
 - 在某些情况下，在迁移目录数据之后启动 Directory Server 时，新的发行版本 5 错误检查会在 Directory Server 组条目中检测到循环定义。这些循环定义在功能上是良性的，但可能会导致在错误文件中记录大量的错误。
-

有关迁移过程、`dsmig` 命令和手动迁移的详细信息，参见 Directory Server Enterprise Edition 6 Migration Guide (<http://docs.sun.com/doc/819-0994>)。

检验升级

可按如下步骤检验 Directory Server 的升级是否成功。

1. 启动新的 Directory Server 实例：

`DirServer-base/ds6/bin/dsadm -v`

参见第 95 页中的表 5-4 确认输出值。

2. 检查 Directory Server 错误日志中的启动消息：

instancePath/logs/errors

注 在启动时，发行版本 5 Directory Server 会立即检测到循环定义。这些循环定义在功能上是良性的，但在您从包含此类循环定义的先前版本升级时可能会导致生成大量的错误。

升级后的任务

除了第 97 页中的“升级过程”中所述的步骤之外，没有任何升级后的任务，但需要将依赖于 Directory Server 的所有 Java ES 组件重新配置为指向新的 Directory Server 实例。

升级回滚

发行版本 5 的升级回滚是通过复原到先前版本来实现的，该版本在升级到发行版本 5 后仍保持原样。

多实例升级

第 95 页中的“发行版本 4 Directory Server 的升级”中的过程没有对一些部署体系结构进行明确说明，在这些部署体系结构中复制了 Directory Server 以实现可用性或可伸缩性。这些体系结构可能包括 Directory Server 复制或在 Sun Cluster 环境中将 Directory Server 作为数据服务进行部署。

Directory Server 复制的滚动升级

不同计算机系统上的多个 Directory Server 实例（如在多主复制部署体系结构中使用的实例）可以依序升级，每次升级一个实例。在第一次同步所有 Directory Server 主节点之后，升级相应主机上的每个实例，而其他实例仍保持运行状态。这种滚动升级允许目录服务在升级用以提供该项服务的各个 Directory Server 实例的同时保持联机状态。

作为数据服务升级 Directory Server

有关在 Sun Cluster 环境中将 Directory Server 作为数据服务进行升级和回滚的信息目前正在准备之中。

从 Java ES 发行版本 3 升级 Directory Server

将 Java ES 2003Q1（发行版本 3）Directory Server 升级到发行版本 5 的过程与将发行版本 4 Directory Server 升级到发行版本 5 的过程相同。

要将发行版本 3 Directory Server 升级到发行版本 5，请按照第 94 页中的“从 Java ES 发行版本 4 升级 Directory Server”中的说明操作，但需要在提到发行版本 4 的地方用发行版本 3 代替。

从 Java ES 发行版本 2 升级 Directory Server

升级前的任务应该包括将所有共享组件（参见第 43 页中的表 1-9）升级到发行版本 5，除此之外，将 Java ES 2004Q2（发行版本 2）Directory Server 升级到发行版本 5 的过程与将发行版本 4 Directory Server 升级到发行版本 5 的过程相同。

第 59 页中的第 2 章“升级 Java ES 共享组件”中提供了将 Java ES 共享组件升级到发行版本 5 的说明。

要将发行版本 2 Directory Server 升级到发行版本 5，请按照第 94 页中的“从 Java ES 发行版本 4 升级 Directory Server”中的说明操作，但需要在提到发行版本 4 的地方用发行版本 2 代替。

注	如果要在 Linux 平台上从发行版本 2 Directory Server 升级，则必须执行双重升级，即 Directory Server 和操作系统都要升级（RHEL 2.1 上不支持发行版本 5 Directory Server）。有关更多信息，参见第 93 页中的“双重升级”。
----------	---

Directory Proxy Server

本章介绍如何将 Directory Proxy Server 升级到 Java ES 5（发行版本 5）：Sun Java System Directory Proxy Server 6.0。

本章概要介绍了有关发行版本 5 所支持的不同升级方式的升级注意事项。其内容涵盖了在 Solaris 和 Linux 操作系统上的升级：

- [第 106 页中的“Directory Proxy Server 升级概述”](#)
- [第 110 页中的“从 Java ES 发行版本 4 升级 Directory Proxy Server”](#)
- [第 116 页中的“从 Java ES 发行版本 3 升级 Directory Proxy Server”](#)
- [第 117 页中的“从 Java ES 发行版本 2 升级 Directory Proxy Server”](#)

注 本章中的文件位置是相对于两个目录路径指定的，分别称为 *serverRoot* (Directory Proxy Server 5.x) 和 *DirServer-base* (Directory Proxy Server 6.0)。在安装 Directory Proxy Server 时，可能至少已将这两个路径的一部分指定为安装目录，否则，Java ES 安装程序会分配一个默认值。

下表说明了这些目录路径的默认值。

表 6-1 Directory Proxy Server 目录路径

路径名变量	Solaris OS	Linux OS
<i>serverRoot</i> (Directory Proxy Server 5.x)	/var/opt/mps/serverroot	/var/opt/sun/directory-server
<i>DirServer-base</i> (Directory Proxy Server 6.0)	/opt/SUNWdsee	/opt/sun

Directory Proxy Server 升级概述

本节将在以下方面对 Directory Proxy Server 进行介绍，这些方面会对升级到 Java ES 5（发行版本 5）产生影响：

- [关于 Java ES 发行版本 5](#)
- [Java ES 发行版本 5 升级指导说明](#)
- [Directory Proxy Server 数据](#)
- [Directory Proxy Server 升级策略](#)

关于 Java ES 发行版本 5

相对于发行版本 4 Directory Proxy Server 和所有早期发行版本而言，Java ES 发行版本 5 Directory Proxy Server 是一个主要发行版本，同时也是一个新产品。

发行版本 5 Directory Proxy Server 仍是 LDAP 代理，但具有新的、可扩展路由选择功能。发行版本 5 还支持“虚拟目录”功能，即能够将多个数据视图整合到一个视图中。这些数据视图可以表示 LDAP 或 SQL 可以访问的数据存储库。

有关更多信息，参见 Directory Server Enterprise Edition 6 发行说明。

Java ES 发行版本 5 升级指导说明

表 6-2 显示了所支持的、将 Directory Proxy Server 升级到 Java ES 发行版本 5 的升级方式。此表既适用于 Solaris，也适用于 Linux 操作系统。

表 6-2 升级到 Java ES 5（发行版本 5）：Directory Proxy Server 6.0 的方式

Java ES 发行版本	Directory Proxy Server 版本	常规方法	必需的重新配置
发行版本 4	Sun Java System Directory Proxy Server 5.2 2005Q4	直接升级：用发行版本 5 的全新安装和配置来替换发行版本 4。	如果需要向后兼容性，请将以前的配置手动映射到新的配置属性。
发行版本 3	Sun Java System Directory Proxy Server 5 2005.2Q1	直接升级：用发行版本 5 的全新安装和配置来替换发行版本 3。	如果需要向后兼容性，请将以前的配置手动映射到新的配置属性。
发行版本 2	Sun Java System Directory Proxy Server 5.2 2004Q2	直接升级：用发行版本 5 的全新安装和配置来替换发行版本 2。	如果需要向后兼容性，请将以前的配置手动映射到新的配置属性。

表 6-2 升级到 Java ES 5（发行版本 5）：Directory Proxy Server 6.0 的方式（续）

Java ES 发行版本	Directory Proxy Server 版本	常规方法	必需的重新配置
发行版本 1	Sun ONE Directory Proxy Server5.2	直接升级未经验证：但可以使用发行版本 2 的升级方法。	如果需要向后兼容性，请将以前的配置手动映射到新的配置属性。
Java ES 以前的发行版本	Sun ONE Directory Proxy Server5.2	直接升级未经验证：但可以使用发行版本 2 的升级方法。	如果需要向后兼容性，请将以前的配置手动映射到新的配置属性。
	Sun ONE Directory Access Router 5.0 或 5.0 SP1	不可直接升级：首先升级到发行版本 3。参阅 Java Enterprise System 2005Q1 升级与迁移指南 (http://docs.sun.com/doc/819-2237)。然后从发行版本 3 升级到发行版本 5。	参阅 Java Enterprise System 2005Q1 升级与迁移指南 (http://docs.sun.com/doc/819-2237)。

Directory Proxy Server 数据

Directory Proxy Server 不再利用 Directory Server 来存储配置数据。使用新的 Directory Service Control Center 或 Directory Server EE 命令行实用程序执行配置。

下表显示了可能受 Directory Proxy Server 软件升级影响的数据类型。

表 6-3 Directory Proxy Server 数据用途

数据类型	位置	用途
Directory Proxy Server 配置数据	Directory Proxy Server 5.2: 配置目录 Directory Proxy Server 6.0: 通过 Directory Service Control Center 和 Directory Server EE 命令行实用程序访问。	Directory Proxy Server 的配置
安全性数据	Directory Proxy Server 5.2: 通过 Directory Proxy Server 控制台配置 SSL。 Directory Proxy Server 6.0: 通过 Directory Service Control Center 和 Directory Server EE 命令行实用程序配置 SSL。	服务器证书

Directory Proxy Server 升级策略

Directory Proxy Server 的升级策略通常取决于在第 1 章“升级规划”中讨论的许多注意事项：升级方式、Java ES 组件之间的依赖性、选择性升级与全部升级、多实例部署，等等。

本节通过提出可能影响 Directory Proxy Server 升级计划的问题，对 Directory Proxy Server 的一般注意事项进行逐一介绍。

兼容性问题

发行版本 5 Directory Proxy Server 引入了接口更改，从而与早期的 Directory Proxy Server 发行版本不兼容。发行版本 5 Directory Proxy Server 以全新的、基于 Java 的实现为基础，其配置与发行版本 4 Directory Proxy Server 以及早期发行版本完全不同。

但是，可以将发行版本 5 Directory Proxy Server 配置为向后兼容，即，使其行为类似发行版本 4 Directory Proxy Server 及早期发行版本。此配置要求您将以前的配置属性手动映射到发行版本 5 配置属性。有关详细信息，参见 Directory Server Enterprise Edition 6 Migration Guide (<http://docs.sun.com/doc/819-0994>)。

但发行版本 5 Directory Proxy Server 的默认行为不同于先前版本：它不允许 LDAP 控件通过代理。要再现早期版本的行为，可按第 114 页中的“升级后的任务”中所述取消对这些控件的阻塞。

组件依赖性

对其他 Java ES 组件的依赖性会影响升级和重新配置 Directory Proxy Server 软件的过程。

Directory Proxy Server 对以下 Java ES 组件具有依赖性：

- **共享组件。**Directory Proxy Server 对特定的 Java ES 共享组件具有依赖性（参见第 43 页中的表 1-9）。Directory Proxy Server 升级可能要依赖于这些共享组件的已升级版本。
- **Directory Server。**Directory Proxy Server 对用于增进 LDAP 请求安全性和性能的 Directory Server 具有协作依赖性。Directory Proxy Server 提供对 Directory Server 的前端访问，但除了这种功能关系之外，它对 Directory Server 没有依赖性。

双重升级

在双重升级中，Directory Proxy Server 和操作系统都要进行升级（如第 41 页中的“双重升级：Java ES 和操作系统软件”中所述）。可使用以下两种方式之一执行双重升级：

- 操作系统全新安装
- 操作系统就地升级

操作系统全新安装

1. 备份现有 Directory Proxy Server 数据。

有关基本数据的位置，参见第 107 页中的“Directory Proxy Server 数据”。

2. 安装新的操作系统。

可将操作系统安装在新系统上（或 Solaris 10 区域中），否则会擦除现有文件系统。

3. 安装发行版本 5 Directory Proxy Server。

4. 创建一个发行版本 5 Directory Proxy Server 实例，然后将配置属性映射到发行版本 5 Directory Proxy Server 属性。

参见第 112 页中的“升级发行版本 4 Directory Proxy Server”过程中的相关步骤。

操作系统就地升级

1. 备份现有 Directory Proxy Server 数据。

有关基本数据的位置，参见第 107 页中的“Directory Proxy Server 数据”。

2. 升级操作系统。

升级会将现有文件系统留在原来的位置。

3. 将 Directory Proxy Server 升级到发行版本 5。

参见本章中的相关小节，这要视升级方式而定。

从 Java ES 发行版本 4 升级 Directory Proxy Server

本节包含关于将 Directory Proxy Server 从 Java ES 2005Q4（发行版本 4）升级到 Java ES 5（发行版本 5）的信息。其中涵盖以下主题：

- [简介](#)
- [发行版本 4 Directory Proxy Server 的升级](#)
- [多实例升级](#)

简介

将 Java ES 发行版本 4 Directory Proxy Server 升级到发行版本 5 时，请考虑升级过程中的以下方面：

- **常规升级方法。** 通过执行发行版本 5 Directory Proxy Server 的全新安装，然后使用 Directory Service Control Center 或 Directory Server EE 命令行实用程序配置新的 Directory Proxy Server 实例来实现升级。
- **升级依赖性。** Directory Proxy Server 对许多 Java ES 共享组件都具有依赖性（参见第 43 页中的表 1-9），所有这些组件会在您升级 Directory Proxy Server 时，由 Java ES 安装程序自动将它们升级到发行版本 5。
- **向后兼容性。** 可将发行版本 5 Directory Proxy Server 配置为与发行版本 4 的版本向后兼容，如第 108 页中的“兼容性问题”中所述。
- **升级回滚。** 发行版本 5 的升级回滚通过恢复到前一个版本来实现，该版本在升级到发行版本 5 时会保持不变。
- **平台问题。** 在 Solaris 和 Linux 操作系统上，升级 Directory Proxy Server 的常规方法相同。

发行版本 4 Directory Proxy Server 的升级

本节介绍如何在 Solaris 和 Linux 平台上将 Directory Proxy Server 从 Java ES 发行版本 4 升级到 Java ES 发行版本 5。如果其中某个主题取决于具体平台的过程，则该主题将指明所适用的操作系统。本节包括以下主题：

- [升级前的任务](#)
- [升级发行版本 4 Directory Proxy Server](#)
- [检验升级](#)
- [升级后的任务](#)
- [升级回滚](#)

升级前的任务

升级 Directory Proxy Server 软件之前，应执行以下任务：

- [检验当前版本信息](#)
- [升级 Directory Proxy Server 依赖项](#)
- [获取所需的配置信息和密码](#)

检验当前版本信息

可使用以下命令检验 Directory Proxy Server 的当前版本：

```
cd serverRoot/bin/dps/server/bin
./ldapfwd -v
```

下表显示了输出结果：

表 6-4 Directory Proxy Server 版本检验输出

Java ES 发行版本	Directory Proxy Server 版本号
发行版本 2	Sun ONE Directory Proxy Server Version 5.2_Patch_2
发行版本 3	Sun ONE Directory Proxy Server Version 5.2_Patch_3
发行版本 4	Sun ONE Directory Proxy Server Version 5.2_Patch_4
发行版本 5 ¹	Sun ONE Directory Proxy Server Version 6.0

1. ldapfwd 命令不能用于返回发行版本 5 的版本号。参见第 114 页中的“检验升级”。

升级 Directory Proxy Server 依赖项

通常建议将计算机系统（以及计算环境）中的所有 Java ES 组件都升级到 Java ES 发行版本 5。Directory Proxy Server 仅对少数几个共享组件具有硬性升级依赖性。

升级 Directory Proxy Server 的依赖项时，应在升级 Directory Proxy Server 之前按以下顺序执行（跳过任何可能已升级的依赖项）。共享组件的升级通常是由 Java ES 安装程序自动实现的。

1. **共享组件。** 第 59 页中的“升级 Java ES 共享组件”中提供了将 Java ES 共享组件同步到发行版本 5 的说明。不过，在将 Directory Proxy Server 升级到发行版本 5 时，Java ES 安装程序会自动升级 Directory Proxy Server 所需的所有共享组件。
2. **Directory Server（软性升级依赖性）。** 第 89 页中的第 5 章“Directory Server”中提供了将 Directory Server 升级到发行版本 5 的说明。不过，发行版本 4 Directory Server 支持发行版本 5 Directory Proxy Server。

获取所需的配置信息和密码

配置信息在升级过程中会得到保留并可用于将发行版本 4 配置属性映射到发行版本 5 配置属性。参见第 108 页中的“兼容性问题”。

升级发行版本 4 Directory Proxy Server

本节讨论影响 Directory Proxy Server 升级过程的一些注意事项，随后对升级过程本身加以说明。

升级注意事项

将 Directory Proxy Server 软件升级到 Java ES 发行版本 5 时，应考虑以下注意事项：

- 应将使用 Directory Proxy Server 实例的所有 Java ES 组件（如 Access Manager、Communications Express、Messaging Server、Portal Server，等等）全部关闭，并重新配置为访问所对应的、新的发行版本 5 实例。
- 在其中有多个 Directory Proxy Server 实例在一台计算机上运行（所有实例都对应于同一已安装的 Directory Proxy Server 映像）的部署体系结构中，升级 Directory Proxy Server 映像将需要创建新的 Directory Proxy Server 实例。
- 在包含 Directory Proxy Server 的发行版本 4 部署体系结构中，使用 Administration Server 来配置和管理 Directory Proxy Server 实例。在发行版本 5 中，Directory Proxy Server 的升级可能会要求部署 Directory Service Control Center，用于配置和管理 Directory Proxy Server 实例。

升级过程

下述过程适用于本地驻留在即将进行升级的计算机上的 Directory Proxy Server 实例。

1. 以 root 身份登录，或成为超级用户。

```
su -
```

2. 关闭依赖于即将升级的 Directory Proxy Server 实例的所有 Java ES 组件。此步骤可能取决于 Directory Proxy Server 在部署体系结构中的复制方式。

有关如何关闭 Java ES 组件的信息，参见其各自的管理指南。

3. 执行发行版本 5 Directory Proxy Server 的全新安装。

执行以下步骤：

- a. 启动 Java ES 安装程序。

```
cd Java ES Release 5 distribution/os_arch
./installer
```

其中，*os_arch* 与您的平台匹配，如 *Solaris_sparc*。（对于命令行界面，使用 `installer -nodisplay` 选项。）

显示“欢迎”和“许可协议”页面之后，将向您显示一个组件选择页面。（如果检测到可由 Java ES 安装程序直接升级的已安装组件，它们的状态会显示为“可升级”。）

- b. 选择 Directory Server Enterprise Edition 的 Directory Proxy Server 子组件。

还需要安装您想要使用的管理子组件（Directory Service Control Panel 或命令行实用程序）。

- c. 指定一个不同于任何现有发行版本 4 Directory Proxy Server 的安装路径。

- d. 选择“现在配置”或“以后再配置”。

选择“现在配置”还是“以后再配置”并不重要，因为实际上 Directory Proxy Server 不需要配置。但是，如果选择“现在配置”，系统将询问您是否想要创建新实例。（安装组件后，您可以按照自己的意愿创建任意多的 Directory Proxy Server 实例。）

- e. 如果需要，选择安装本地化软件包的选项。

- f. 确认您的安装选择。

将安装 Directory Proxy Server 软件包并显示安装摘要。

- g. 退出 Java ES 安装程序。

如果尚未创建新实例，请继续执行 [步骤 4](#)，否则，继续执行 [步骤 5](#)。

4. 创建 Directory Proxy Server 实例。

```
DirServer-base/dps6/bin/dpadm create instancePath
```

其中，*instancePath* 是 Directory Proxy Server 实例的完整路径。

有关创建 Directory Proxy Server 实例的信息，参见 Directory Server Enterprise Edition 6 管理指南 (<http://docs.sun.com/doc/820-0293>)。

5. 如果需要，将发行版本 4 配置属性映射到发行版本 5 Directory Proxy Server 属性。

有关该映射过程的详细信息，参见 Directory Server Enterprise Edition 6 Migration Guide (<http://docs.sun.com/doc/819-0994>)。

检验升级

可按如下步骤检验 Directory Proxy Server 的升级是否成功。

1. 启动新的 Directory Proxy Server 实例。

```
DirServer-base/dps6/bin/dpadm start instancePath
```

2. 检查 Directory Proxy Server 的版本。

```
DirServer-base/dps6/bin/dpadm --version
```

第 111 页中的表 6-4 中显示了输出值。

升级后的任务

所有依赖于 Directory Proxy Server 的 Java ES 组件都需要重新配置为指向新的 Directory Proxy Server 实例。

此外，要再现先前版本的默认行为，必须明确允许 LDAP 控件通过代理。可通过设置 `allowed-ldap-controls` 属性来启用这些控件，如下所示：

```
cd DirServer-base/dps6/bin  
  
./dpconf set-server-prop  
allowed-ldap-controls:auth-request  
allowed-ldap-controls:chaining-loop-detection  
allowed-ldap-controls:manage-dsa  
allowed-ldap-controls:persistent-search  
allowed-ldap-controls:proxy-auth-v1  
allowed-ldap-controls:proxy-auth-v2  
allowed-ldap-controls:real-attributes-only  
allowed-ldap-controls:server-side-sorting
```

升级回滚

发行版本 5 的升级回滚是通过复原到先前版本来实现的，该版本在升级到发行版本 5 后仍保持原样。

多实例升级

在一些部署体系结构中，Directory Proxy Server 被部署在多个计算机系统上，以提供可伸缩性和提高可用性。例如，您可以在多台计算机上运行 Directory Proxy Server 组件，并使用一个负载均衡器来分配负载。

对于 Directory Proxy Server 的负载均衡实例，将不能执行滚动升级：需要关闭负载均衡器并重新配置为访问发行版本 5 实例。按第 111 页中的“[发行版本 4 Directory Proxy Server 的升级](#)”中所述执行每个实例的升级。

从 Java ES 发行版本 3 升级 Directory Proxy Server

将 Java ES 2005Q1（发行版本 3）Directory Proxy Server 升级到发行版本 5 的过程与将发行版本 4 Directory Proxy Server 升级到发行版本 5 的过程相同。

要将发行版本 3 Directory Proxy Server 升级到发行版本 5，请按照第 110 页中的[“从 Java ES 发行版本 4 升级 Directory Proxy Server”](#)中的说明操作，但需要在提到发行版本 4 的地方用发行版本 3 代替。

从 Java ES 发行版本 2 升级 Directory Proxy Server

升级前的任务应包括将所有共享组件（参见第 43 页中的表 1-9）以及 Directory Proxy Server 所依赖的所有本地驻留的产品组件同步到发行版本 5，除此之外，将 Java ES 2004Q2（发行版本 2）Directory Proxy Server 升级到发行版本 5 的过程与将发行版本 4 Directory Proxy Server 升级到发行版本 5 的过程相同。

第 59 页中的第 2 章“升级 Java ES 共享组件”中提供了将 Java ES 共享组件同步到发行版本 5 的说明。

要将发行版本 2 Directory Proxy Server 升级到发行版本 5，请按照第 110 页中的“从 Java ES 发行版本 4 升级 Directory Proxy Server”中的说明操作，但需要在提到发行版本 4 的地方用发行版本 2 代替。

注

如果要在 Linux 平台上从发行版本 2 Directory Proxy Server 升级，则必须执行双重升级，即 Directory Proxy Server 和操作系统都要升级（RHEL 2.1 上不支持发行版本 5 Directory Proxy Server）。有关更多信息，参见第 109 页中的“双重升级”。

Web Server

本章介绍如何将 Web Server 升级到 Java ES 5（发行版本 5）：Sun Java System Web Server 7.0。

本章概要介绍了有关发行版本 5 所支持的不同升级方式的升级注意事项。其内容涵盖了在 Solaris 和 Linux 操作系统上的升级：

- 第 120 页中的“Web Server 升级概述”
- 第 123 页中的“从 Java ES 发行版本 4 升级 Web Server”
- 第 139 页中的“从 Java ES 发行版本 3 升级 Web Server”
- 第 140 页中的“从 Java ES 发行版本 2 升级 Web Server”

注 本章中的文件位置是相对于若干目录路径指定的，分别称为 *WebServer6-base* (Web Server 6.x) 以及 *WebServer7-base* 和 *WebServer7Config-base* (Web Server 7.0)。在最初安装 Web Server 时，可能至少已将这些路径的一部分指定为安装目录。否则，Java ES 安装程序会分配一个默认值。

下表说明了这些目录路径的默认值。

表 7-1 Web Server 目录路径

路径名变量	Solaris OS	Linux OS
<i>WebServer6-base</i>	/opt/SUNWwbsvr	/opt/sun/webserver
<i>WebServer7-base</i>	/opt/SUNWwbsvr7	/opt/sun/webserver7
<i>WebServer7Config-base</i>	/var/opt/SUNWwbsvr7	/var/opt/sun/webserver7

Web Server 升级概述

本节将在以下方面对 Web Server 进行介绍，这些方面会对升级到 Java ES 5（发行版本 5）产生影响：

- [关于 Java ES 发行版本 5 Web Server](#)
- [Web Server 升级指导说明](#)
- [Web Server 数据](#)
- [Web Server 升级策略](#)

关于 Java ES 发行版本 5 Web Server

相对于发行版本 4 而言，Java ES 发行版本 5 Web Server 是一个主要发行版本。它新增了许多功能并且对接口也进行了多处增强。

发行版本 5 Web Server 采用了新的管理基础结构，其中带有多个新的管理工具。该管理基础结构包括一个 Administration Server 实例，该实例可托管任意多个 Web Server 实例的配置信息。还有一个新的命令行界面 (wadm) 和新的图形用户界面，用于在本地或在远程计算机上创建 Web Server 实例以及配置和管理这些实例。新的管理工具需要管理员用户名和密码。

有关新管理基础结构的更多信息，参见 Web Server 7.0 管理员指南 (<http://docs.sun.com/doc/820-0872>)。

Web Server 管理接口中的这些变化对升级具有重大影响。

Web Server 升级指导说明

[表 7-2](#) 显示了所支持的、将 Web Server 升级到 Java ES 发行版本 5 的升级方式。此表既适用于 Solaris，也适用于 Linux 操作系统。

表 7-2 升级到 Java ES 5（发行版本 5）： Web Server 7.0 的方式

Java ES 发行版本	Web Server 版本	常规方法	必需的重新配置
发行版本 4	Sun Java System Web Server 6.1 SP5 2005Q4	直接升级： 全新安装后紧接着进行数据迁移	将实例配置迁移到新的实例。
发行版本 3	Sun Java System Web Server 6 2005Q1 Update 1 SP 4	直接升级： 全新安装后紧接着进行数据迁移	将实例配置迁移到新的实例。

表 7-2 升级到 Java ES 5（发行版本 5）： Web Server 7.0 的方式（续）

Java ES 发行版本	Web Server 版本	常规方法	必需的重新配置
发行版本 2	Sun Java System Web Server 6 2004Q2 Update 1 SP 2 Platform Edition 和 Enterprise Edition	直接升级： 全新安装后紧接着进行数据迁移	将实例配置迁移到新的实例。
发行版本 1	Sun ONE Web Server 6.1 (2003Q4)	直接升级未经验证： 但这是通过在执行全新安装后紧接着进行数据迁移来实现的。	将实例配置迁移到新的实例。
Java ES 以前的发行版本		不可直接升级。	

Web Server 数据

下表显示了可能受 Web Server 软件升级影响的数据类型。

表 7-3 Web Server 数据用途

数据类型	位置	用途
配置数据	Web Server 6.x（Java ES 发行版本 2、3 和 4）： <i>WebServer6-base/https-instanceName/config/</i> Web Server 7.0（Java ES 发行版本 5）： 实例配置 <i>WebServer7Config-base/https-configName/config¹/</i> 中心配置存储库 通过 Web Server 控制台和 wadm 命令行界面进行访问。	Web Server 实例的配置

1. 请注意，*WebServer7Config-base* 路径与 *WebServer6-base* 路径大不相同。

Web Server 升级策略

Web Server 的升级策略通常取决于在第 1 章“升级规划”中讨论的许多注意事项：升级方式、Java ES 组件之间的依赖性、选择性升级与全部升级、多实例部署，等等。

本节通过提出可能影响 Web Server 升级计划的问题，对 Web Server 的一般注意事项进行逐一介绍。

兼容性问题

Java ES 发行版本 5 Web Server 未在公共接口中引入任何更改，因此从这点上讲，它与早期版本向后兼容。但是，新的管理接口与早期的管理接口不向后兼容。这就会影响 Web 应用程序（例如，包括 Java ES 组件）的升级和重新部署。

特别是，发行版本 5 Web Server 对实例目录和虚拟服务器名使用了不同的默认值，如下表所示。

表 7-4 Web Server 默认名称

项	Java ES 发行版本 2、3 和 4 Web Server 6.x 默认值	Java ES 发行版本 5 Web Server 7.0 默认值
配置名		<i>hostName.domainName</i>
实例目录路径	<i>WebServer6-base / https-hostName.domainName</i>	<i>WebServer7Config-base https-hostName.domainName</i>
虚拟服务器名	<i>https-hostName.domainName</i>	<i>hostName.domainName</i>

Web Server 组件依赖性

Web Server 对以下 Java ES 组件具有依赖性：

- **共享组件。** Web Server 对特定的 Java ES 共享组件具有依赖性（参见第 43 页中的表 1-9）。Web Server 升级可能要依赖于这些共享组件的已升级版本。
- **Directory Server。** Web Server 对用于提供基于 LDAP 验证的 Directory Server 具有可选依赖性。
- **Web Proxy Server。** Web Server 对用于增进 HTTP 请求安全性和性能的 Web Proxy Server 具有协作依赖性。

双重升级

在双重升级中，Web Server 和操作系统都要进行升级（如第 41 页中的“双重升级：Java ES 和操作系统软件”中所述）。可使用操作系统就地升级方法执行双重升级：

1. 备份现有 Web Server 数据。
有关基本数据的位置，参见第 121 页中的“Web Server 数据”。
2. 升级操作系统。
升级会将现有文件系统留在原来的位置。
3. 将 Web Server 升级到发行版本 5。
参见本章中的相应小节，这要视升级方式而定。

从 Java ES 发行版本 4 升级 Web Server

本节包含关于将 Web Server 从 Java ES 2005Q4（发行版本 4）升级到 Java ES 5（发行版本 5）的信息。其中涵盖以下主题：

- [简介](#)
- [发行版本 4 Web Server 的升级](#)

简介

将 Java ES 发行版本 4 Web Server 升级到发行版本 5 时，请考虑升级过程中的以下方面：

- **常规升级方法。** 这种执行升级的方法是先进行发行版本 5 Web Server 的全新安装，将发行版本 4 Web Server 实例配置信息迁移到发行版本 5 配置，然后创建与发行版本 4 实例相对应的发行版本 5 Web Server 实例。
- **升级依赖性。** Web Server 对许多 Java ES 共享组件都具有依赖性（参见 [第 43 页中的表 1-9](#)），在执行 Web Server 的升级时，Java ES 安装程序会自动将所有这些组件都升级到发行版本 5。Web Server 仅对 NSS 和 NSPR 共享组件具有硬性升级依赖性。
- **向后兼容性。** 发行版本 5 Web Server 管理接口与发行版本 4 的版本不向后兼容。
- **升级回滚。** 发行版本 5 的升级回滚是通过复原到保持原样的发行版本 4 安装来实现的。
- **平台问题。** 在 Solaris 和 Linux 操作系统上，升级 Web Server 的常规方法相同。

发行版本 4 Web Server 的升级

本节介绍如何在 Solaris 和 Linux 平台上将 Web Server 从 Java ES 发行版本 4 升级到 Java ES 发行版本 5。如果其中某个主题取决于具体平台的过程，则该主题将指明所适用的操作系统。本节包括以下主题：

- [升级前的任务](#)
- [升级发行版本 4 Web Server](#)
- [检验升级](#)
- [升级后的任务](#)
- [升级回滚](#)

升级前的任务

升级 Web Server 软件之前，应执行以下任务：

- [检验当前版本信息](#)
- [升级 Web Server 依赖项](#)
- [备份 Web Server 数据](#)
- [获取所需的配置信息和密码](#)

检验当前版本信息

通过使用 -version 选项运行 Web Server 实例服务器，可以检验 Web Server 的当前版本：

`WebServer6-base/https-hostName.domainName/start -version`

表 7-5 Web Server 版本检验输出

Java ES 发行版本	Web Server 版本号
发行版本 2	6.1SP2
发行版本 3	6.1SP4
发行版本 4	6.1SP5
发行版本 5	7.0

升级 Web Server 依赖项

通常建议将计算机系统上（和计算环境中）的所有 Java ES 组件都升级到 Java ES 发行版本 5。但是，在执行 Web Server 到发行版本 5 的升级时，Java ES 安装程序会自动升级 Web Server 所需的所有共享组件（参见第 43 页中的表 1-9）。

备份 Web Server 数据

在将 Web Server 从发行版本 4 升级到发行版本 5 时，不会修改现有配置数据，它们将保持原样。因此没有必要备份当前数据。

获取所需的配置信息和密码

您必须以超级用户身份登录才能执行升级，并且执行迁移的用户帐户应有权访问现有的 Web Server 安装目录。

升级发行版本 4 Web Server

本节讨论影响 Web Server 升级过程的一些注意事项，随后对升级过程本身加以说明。

升级注意事项

将 Web Server 软件升级到 Java ES 发行版本 5 时，应考虑以下注意事项：

- **“现在配置”或“以后再配置”。**执行升级时，使用“现在配置”或“以后再配置”选项指定是否安装发行版本 5 Web Server：
 - “现在配置”意味着安装程序将如指定的那样设置 **Administration Server** 或管理节点，还将创建默认配置及相应的 **Web Server** 实例。此方法对于单台计算机上的安装很有用，但在升级期间默认配置名可能会对现有实例配置的迁移产生影响。
 - “以后再配置”意味着安装程序将不执行任何配置：您必须在向输入文件提供属性值后手动运行 `configureServer` 脚本。如果要使用执行无提示安装脚本在多台计算机上自动进行安装，此方法很有用。还可以对配置名进行完全控制，从而避免在升级期间与现有实例配置的迁移发生冲突。
- **默认发行版本 4 实例配置的迁移。**执行升级时，将每个发行版本 4 Web Server 实例的配置数据均迁移到由 **Web Server Administration Server** 维护的中心配置存储库。该迁移是使用 `wadm migrate-server` 命令或发行版本 5 管理控制台实现的。

如果所迁移的实例为默认发行版本 4 Web Server (6.x) 实例，则它与默认发行版本 5 Web Server (7.0) 配置具有相同的名称 (`hostName.domainName`)，后者是通过“现在配置”选项自动创建的。

可以采用三种方法来执行默认发行版本 4 实例配置到发行版本 5 的迁移，每种方法最终会产生不同的配置名。

所选择的方法可能会对已部署 Web 应用程序的后续升级产生影响。例如，在升级已部署的 Java ES 组件（如 Access Manager 和 Portal Server）和 Sun Java Communications Suite 组件（如 Communications Express、Instant Messaging 和 Delegated Administrator）时，执行此类升级的人员通常需要知道发行版本 4 实例配置已迁移到的发行版本 5 配置的名称。

这三种方法如下：

- 指定不同于默认名称的新配置名，例如，"JavaESapps"。migrate-server 命令随即会创建一个名为 JavaESapps 的新的发行版本 5 配置。

随后，需要将已升级的 Web 应用程序重新部署到 JavaESapps 配置。

- 不指定新的配置名，而是在运行 migrate-server 命令之前删除默认的发行版本 5 实例及配置 (*hostName.domainName*)。migrate-server 命令随即会使用默认名称 (*hostName.domainName*) 创建一个新配置。顺序如下（有关详细信息，参见 wadm help）：

```
wadm delete-instance
wadm delete-config
wadm migrate-server
wadm create-instance
```

随后，需要将已升级的 Web 应用程序重新部署到 *hostName.domainName* 配置，该配置与默认发行版本 4 实例同名。

- 不指定新的配置名，也不删除默认发行版本 5 配置。migrate-server 命令随即会使用以下名称创建一个新配置：*hostName.domainName-1*。

随后，需要将已升级的 Web 应用程序重新部署到 *hostName.domainName-1* 配置。

无论在迁移发行版本 4 实例配置时采取哪种方法，无论何人执行已部署在该实例中的 Web 应用程序的后续升级，发行版本 4 实例配置所迁移到的发行版本 5 配置的名称都应该传达给该人。

- **配置数据的迁移。**迁移发行版本 4 实例配置时，会自动迁移以下信息：
 - 以下发行版本 4 Web Server 实例目录中的所有配置信息：*WebServer6-base/https-instanceName/config*。这包括发行版本 4 实例中部署的所有 Web 应用程序（例如，诸如 Access Manager 和 Portal Server 的 Java ES 组件）的配置信息。
 - 来自 *WebServer6-base/httpacl* 的 acl 信息。
 - 来自 *WebServer6-base/userdb* 的 auth-db 信息。
 - 来自 *WebServer6-base/https-admserv/config* 的调度程序信息。
 - 来自 *WebServer6-base/alias* 的证书信息。
 - 在执行迁移时所指定的搜索集合信息和索引文件。

自动迁移不包括以下数据：

- docroot 内容。新配置将转而指向旧的 docroot，并且将在迁移日志中记录日志消息。
- Webdav 数据。将会迁移 Webdav 集合信息。
- 不会迁移第三方 NSAPI 插件。它们将转而指向发行版本 4 文件，并且将在迁移日志中记录日志消息。
- 日志文件
- 对搜索集合 docroot 所做的更改
- 命令行脚本 (startsvr, startsvr.bat, stopsvr, stopsvr.bat, restart, reconfig, reconfig.bat)。

有关数据迁移的详细信息，参见 Web Server 7.0 Installation and Migration Guide (<http://docs.sun.com/doc/819-2625>)。

- **发行版本 5 Web Server 实例的创建。**迁移后，必须显式创建与所迁移发行版本 4 实例相对应的发行版本 5 Web Server 实例。不会自动完成此操作。

升级过程

下述过程适用于所有这样的 Web Server 实例，它们与即将进行升级的计算机上的同一个已安装 Web Server 映像相对应。

1. 以 root 身份登录，或成为超级用户。

```
su -
```

2. 停止所有运行的 Web Server 和 Administration Server 实例。

```
WebServer6-base/https-instanceName/stop
WebServer6-base/https-admserv/stop
```

3. 执行发行版本 5 Web Server 的全新安装。

执行以下步骤：

- a. 启动 Java ES 安装程序。

```
cd Java ES Release 5 distribution/os_arch
./installer
```

其中，os_arch 与您的平台匹配，如 Solaris_sparc。（对于命令行界面，使用 installer -nodisplay 选项。）

显示“欢迎”和“许可协议”页面之后，将向您显示一个组件选择页面。

（如果检测到可由 Java ES 安装程序直接升级的已安装组件，它们的状态会显示为“可升级”。）

- b. 在组件选择页面中选择 Web Server。
- c. 指定不同于发行版本 4 Web Server 的安装路径。
- d. 选择“现在配置”或“以后再配置”。
 - 如果选择“现在配置”，请转至[步骤 e](#)。
 - 如果选择“以后再配置”，请转至[步骤 f](#)。
- e. 如果选择“现在配置”，Java ES 安装程序会提供两种选择：
 - 将管理实例配置为 Administration Server
在将要托管 Administration Server 的计算机上使用此项选择，除了其他管理任务之外，执行发行版本 4 实例到发行版本 5 的迁移也需要 Administration Server。
 - 将管理实例配置为管理节点
在将要托管远离 Administration Server 的 Web Server 实例的计算机上使用此项选择。管理实例配置为与 Administration Server 交互的节点代理。
 - I. 指定所请求的配置值。
系统将要求您输入主机名、HTTP 端口、管理员用户名和管理员密码。
 - II. 确认您的安装选择。
将安装 Web Server 软件包并显示安装摘要。
Java ES 安装程序将创建一个名为 *hostName.domainName* 的默认配置以及相应的 Web Server 实例。
 - III. 退出 Java ES 安装程序，然后转至第 130 页中的[步骤 4](#)。
- f. 如果选择“以后再配置”，Java ES 安装程序将会创建一个 `configureServer` 脚本，您会在第 130 页中的[步骤 IV](#) 中运行该脚本。
 - I. 确认您的安装选择。
将安装 Web Server 软件包并显示安装摘要。
 - II. 退出 Java ES 安装程序。
 - III. 在 `WebServer7-base/setup/WSInstall.properties` 文件中设置值。

向下表中所有必需（非可选）的属性提供值。

表 7-6 WSInstall.properties 值

属性	说明
WS_DOCROOT	（可选）可以托管 Web 内容文件的文档位置
WS_SERVER_NAME	可用于为 HTTP 请求提供服务的主机名
WS_SERVER_USER	运行时 unix 用户。有效值可以是 root、任何有效的 UNIX 用户，或是 webservd（默认值）
WS_HTTP_PORT	可用于侦听 HTTP 请求的实例端口
WS_ADMIN_SSL_PORT	SSL 管理端口
WS_ADMIN_HOST	用于执行管理服务器任务的管理主机名
WS_CONFIG_NAME	此主机的配置名。此值可以与 WS_SERVER_NAME 中提供的值相同
WS_ADMIN_SERVER_USER	管理服务器运行时 UNIX 用户。有效值：'root' 或是与 WS_SERVER_USER 相同的用户
WS_ADMIN_LOGIN_USER	管理服务器登录用户名
WS_ADMIN_LOGIN_PASSWORD	管理服务器登录密码
WS_ADMIN_HTTP_PORT	（可选）非 SSL 管理端口。默认值：8800
WS_START_ON_BOOT	（可选）“引导时启动”功能 (true/false)。True 将允许服务器实例及其管理服务器在系统重新引导后自动启动。默认值：false
WS_64BIT_INSTALL	（可选）服务器运行时模式 (true/false)。True 将在 64 位模式下配置服务器。（仅适用于 Solaris）。False 将在 32 位模式下配置服务器。（仅适用于 Solaris）。默认值：false
WS_ADMIN_IS_SERVER_MODE	（可选）管理配置模式。(true/false)。True 将在管理服务器模式下配置服务器。False 将在管理代理模式下配置服务器。默认值：true
WS_REGISTER_ADMIN_AGENT	（可选）远程代理注册。(true/false)。仅当 WS_ADMIN_IS_SERVER_MODE 设置为 false 时才需要此属性。True 将要求您提供用于注册的远程管理服务器主机。默认值：true
WS_AGENT_SSL_PORT	（可选）管理代理 SSL 端口。仅当 WS_ADMIN_IS_SERVER_MODE 设置为 false 时才需要此属性
WS_AGENT_HOST=	（可选）管理代理主机名。仅当 WS_ADMIN_IS_SERVER_MODE 设置为 false 时才需要此属性

IV. 运行 configureServer 脚本。

```
WebServer7-base/setup/configureServer
-inputfile WebServer7-base/setup/WSInstall.properties
-logfile WebServer7-base/setup/WSInstall.log
-verbose
```

configureServer 脚本将创建一个名为 *hostName.domainName* 的默认配置以及相应的 Web Server 实例。

4. 启动 Web Server Administration Server 服务。

```
WebServer7Config-base/admin-server/bin/startserv
```

5. 将发行版本 4 Web Server 实例配置迁移到发行版本 5 配置。

可以使用命令行 (wadm)，也可以使用图形用户界面管理工具（登录到 Web Server Admin Server GUI）。以下步骤是基于 wadm 命令行界面进行介绍的。

例如，要将名为 myinstance 的实例迁移到新的配置：

```
WebServer7-base/bin/wadm migrate-server --user=admin
--host=localhost --server-root=/opt/SUNWwbsvr
--instance=https-myinstance --config=newconfigname
```

完整的命令语法如下：

```
WebServer7-base/bin/wadm migrate-server

--user=admin-user [--password-file=admin-pswd-file] [--host=admin-host]
[--echo] [--rcfile=rcfile] [--no-prompt] [--verbose]

[--search-collection-copy-path=searchCollectionPath]
[--log-dir=directory] --serverroot=path
([--all] | [--instance=https-instanceName] [--config=newconfigName])
```

上面的第一组命令选项是所有 wadm 命令的公共选项，在下面的表 7-7 中记述了这些选项。第二组命令选项是特定于 migrate-server 命令的选项，在表 7-8 中记述了这些选项。

仅使用第一组命令选项调用 wadm 时，您将处于 wadm 命令 shell 内。在此 shell 内调用命令不需要再次指定公共选项。

如果从 shell 外部调用完整的 wadm 命令，则至少必须指定 --user 和 --host 选项。（如果省略 --password-file 选项，系统将提示您输入密码；如果省略其他选项，将采用默认值。）但是，对于本章中用来对过程进行说明的命令，为简单起见，未包括 --user 和 --host 选项。

默认情况下，wadm 在端口 8989 使用 SSL 协议。

有关 wadm 命令和选项的完整信息，参见 Web Server 7.0 CLI Reference Manual (<http://docs.sun.com/doc/819-3283>)。

表 7-7 wadm 公共命令选项

选项	说明
user	授权的 Web Server 管理用户 ID。
password-file	一种文件，其中包含用于向 Administration Server 验证管理用户的密码。该密码文件必须包含这样一行：WADM_PASSWORD= <i>password</i> 。如果没有在命令中指定此选项，则系统会提示您输入密码。
host	运行 Administration Server 的计算机的名称。默认值：localhost。
echo	将此选项设置为 true 时，将会在执行命令之前在标准输出上回显命令行。默认值：false。
interactive	如果此选项设置为 true，则会提示输入所需的密码选项。默认值：true。
rcfile	用于在启动 wadm 时加载的启动文件。默认值：~/.wadmrc。
no-prompt	如果此选项设置为 true，则在任何情况下命令都不会要求任何用户输入。例如，如果在调用命令时缺少参数，则命令只是出错，而不会请求并等待用户输入。在与 shell 脚本一起使用 wadm 命令时，您可能需要将此选项设置为 true，以使命令始终返回而不是等待用户输入。默认值：false。
verbose	如果设置为 true，则会显示详细列表。默认值：false。

表 7-8 wadm migrate-server 命令选项和操作对象

选项 / 操作对象	说明
search-collection-copy-path	指定将要在迁移搜索集合时复制搜索集合索引文件的目标路径。可能会出现以下各种迁移情况： 如果 Web Server 6.x 搜索集合路径在 Web Server 6.x 实例外部，则所迁移搜索集合的路径将指向 Web Server 6.x 搜索集合路径，而此选项将会被忽略。 如果 Web Server 6.x 搜索集合路径在 Web Server 6.x 实例之内并且为此选项指定了有效路径，则会将搜索集合索引文件复制到以下目录：<i>searchCollectionPath/configName/virtualServerName/collectionName</i>。如果指定的路径无效，则会在日志中记录一条错误消息。 如果 Web Server 6.x 搜索集合路径在 Web Server 6.x 实例之内而没有为此选项指定任何路径，则不会复制搜索集合索引文件。将会向迁移日志写入一条消息，请求用户使用 <code>wadm add-documents</code> 命令手动复制搜索集合索引文件。在此情况下，所迁移搜索集合的路径将为以下路径：<i>WebServer7Config-base/https-configName/config/collections/virtualServerName/collectionName</i>。
log-dir	迁移日志的位置。 默认值：<i>WebServer7Config-base/admin-serv/logs</i>。
serverroot	安装 Web Server 6.x 版本的安装位置（目录）：与 <i>WebServer6-base</i> 相同。
all	如果设置为 <code>true</code> ，则会将所有 Web Server 6.x 实例配置迁移到与相应实例同名的 Web Server 7.0 配置。如果具有该名称的配置已存在，则会使用 <i>instanceName-1</i> 作为配置名。默认值： <code>false</code> 。
instance	如果要单个迁移实例配置 (<code>all=false</code>)，则是要迁移的 Web Server 6.x 实例配置的名称（形式为： <i>https-instanceName</i> ）。默认的 Web Server 6.x 实例名称为 <i>hostName.domainName</i> 。
config	指定的 Web Server 6.x 实例配置要迁移到的配置名称。默认情况下将使用 Web Server 6.x 实例配置的 <i>instanceName</i> 。但是，如果具有该名称的配置已存在，则命令会在该名称后面附加一个整数。如果要迁移默认的 Web Server 6.x 实例配置，便很可能会出现这种情况。

使用 migrate-server 命令时，请牢记以下注意事项：

- 如果要迁移多个发行版本 4 实例配置，可以使用不同的 `--instance` 值及相应的 `--config` 参数多次运行 `migrate-server` 命令，也可使用 `--all` 选项一齐进行迁移。
- 每次调用 `migrate-server` 命令进行迁移时，均会在 `--log-dir` 选项所指定的目录中（或是在默认的 *WebServer7Config-base/admin-server/logs* 目录中）创建具有以下名称的日志文件：

MIGRATION_YYYYMMDDHHMMSS.log

如果选择 `--all` 选项，则日志文件将会存储所有迁移实例的迁移信息。

- 对于不是通过 `migrate-server` 命令迁移的数据（参见第 125 页中的“升级注意事项”），必须手动执行迁移（参见第 134 页中的“升级后的任务”）。

6. 创建发行版本 5 Web Server 实例。

必须为步骤 5 中所迁移的每个发行版本 4 实例配置创建一个新的发行版本 5 实例。

- a. 在创建新实例之前，检验迁移日志并修复所迁移配置中的任何问题。
- b. 运行 `create-instance` 命令。

```
WebServer7-base/bin/wadm create-instance
--config=configName nodehost1 [nodehost2 ...nodehostN]
```

在第 131 页中的表 7-7 中记述了公共命令选项。在下表中记述了 `create-instance` 命令特定的选项。

表 7-9 wadm create-instance 命令选项和操作对象

选项 / 操作对象	说明
config	实例应指向的发行版本 5 配置名称。
nodehost	即将在其上创建实例的计算机名称。可按空格分隔的 <i>hostName.domainName</i> 列表形式指定多台计算机，从而创建多个同样的实例。

`create-instance` 命令会在指定节点上的 `WebServer7Config-base/https-configName` 处创建一个实例目录，并将配置部署到相应的实例目录。

7. 启动每个发行版本 5 实例。

```
WebServer7Config-base/https-configName/bin/startserv
```

`startserv` 脚本是在创建实例时创建的。如果实例启动时没有出现任何问题，您会看到一条消息，显示“服务器启动成功”。随即会显示该实例的默认 URL。

检验升级

可通过执行以下步骤来检验 Web Server 到发行版本 5 的升级：

1. 检查新创建的迁移日志文件中是否有任何 ERROR 消息。

如果需要，进行手动更改（参见第 134 页中的“升级后的任务”）。

2. 检验发行版本 5 Web Server 实例。

从 Web 浏览器访问以下 URL 并确保到达欢迎页面：

`http://hostName.domainName:port`

其中，全限定主机名和端口对应于每个实例。

3. 使用 `-version` 选项运行 Web Server 实例：

`WebServer7Config-base/https-configName/bin/startserv -version`

有关版本输出值，参见第 124 页中的表 7-5。

升级后的任务

升级后的主要任务涉及执行某些发行版本 4 数据的手动迁移（如果需要）。这些数据通常与针对发行版本 4 配置的和在 `server.xml` 配置文件中指定的一个或多个虚拟服务器相关联。

请注意处理以下情况所需的升级后过程：

- [迁移 Web Server 6.1 docroot 内容](#)
- [迁移 webdav 集合信息](#)
- [迁移日志文件](#)
- [迁移第三方 NSAPI 插件](#)
- [更改搜索集合文档根目录](#)
- [自定义命令行脚本](#)

迁移 Web Server 6.1 docroot 内容

- 1. 将 Web Server 6.1 docroot 内容复制到您希望的任何地方。
- 2. 使用以下命令更新新的文档根目录路径：

```
WebServer7-base/bin/wadm set-virtual-server-prop
--config=configName --vs=virtualServerName
document-root=new docroot path
```

在第 131 页中的表 7-7 中记述了公共命令选项。在下表中记述了 set-virtual-server-prop 命令特定的选项。

表 7-10 wadm set-virtual-server-prop 命令选项和操作对象

选项 / 操作对象	说明
config	要设置新文档根目录路径的发行版本 5 配置名称。
vs	与所迁移文档根目录相对应的虚拟服务器名称。
document-root	新文档根目录的路径。

- 3. 将配置重新部署到相关的 Web Server 实例。

```
WebServer7-base/bin/wadm deploy-config
[--force] [--restart] [--no-reconfig]
configName
```

在第 131 页中的表 7-7 中记述了公共命令选项。在下表中记述了 deploy-config 命令特定的选项。

表 7-11 wadm deploy-cofig 命令选项和操作对象

选项 / 操作对象	说明
force	若为 true，则强制覆写自上次配置部署以来手动修改过的实例配置。默认值：false
restart	若为 true，则会重新启动正在运行的实例，以便在所部署的配置中选取配置设置。默认值：false
no-reconfig	若为 true，则正在运行的实例直到重新启动实例时才会 在所部署的配置中选取配置设置。默认值：false
configName	即将部署到某个 Web Server 实例的发行版本 5 配置名称，该实例的实例名与该配置名相对应。

迁移 webdav 集合信息

不需要另外进行任何手动迁移。只要更新 `docroot` 路径就足矣。

迁移日志文件

如果想要保存这些文件，请将其复制到某个已知位置（否则，假如删除发行版本 4 安装，就会删除这些文件）。

迁移第三方 NSAPI 插件

1. 将库文件从其发行版本 4 位置复制到 `WebServer7-base/lib` 目录。

2. 将 `magnus.conf` 和 `obj.conf` 配置文件导出到某个临时目录。

```
WebServer7-base/bin/wadm get-config-file --config=configName  
magnus.conf > /tmp/magnus.conf
```

```
WebServer7-base/bin/wadm get-config-file --config=configName  
obj.conf > /tmp/obj.conf
```

在第 131 页中的表 7-7 中记述了公共命令选项。

3. 如第三方 NSAPI 插件文档中所指定的那样，修改 `magnus.conf` 和 `obj.conf` 文件。

4. 从临时目录导入 `magnus.conf` 和 `obj.conf` 配置文件。

```
WebServer7-base/bin/wadm set-config-file --config=configName  
--upload-file=/tmp/magnus.conf magnus.conf
```

```
WebServer7-base/bin/wadm set-config-file --config=configName  
--upload-file=/tmp/obj.conf obj.conf
```

在第 131 页中的表 7-7 中记述了公共命令选项。

5. 将经过修改的配置重新部署到相关的 Web Server 实例。

```
WebServer7-base/bin/wadm deploy-config  
[--force] [--restart] [--no-reconfig]  
configName
```

在第 135 页中的表 7-11 中记述了命令选项。

更改搜索集合文档根目录

migrate-server 命令有一个用于迁移搜索集合信息的选项，但是，您可能需要更改搜索集合文档根目录，如下所示：

- 1. 如果搜索集合的文档根目录不同于发行版本 4 所用的文档根目录，则使用以下命令设置搜索集合的文档根目录：

```
WebServer7-base/bin/wadm set-search-collection-prop
--config=configName --vs=virtualServerName
--collection-name=searchCollectionName document-root= 搜索集合的新
docroot 路径
```

在第 131 页中的表 7-7 中记述了公共命令选项。在下表中记述了 set-search-collection-prop 命令特定的选项。

表 7-12 wadm set-search-collection-prop 命令选项和操作对象

选项 / 操作对象	说明
config	要设置搜索集合文档根目录的发行版本 5 配置名称。
vs	与搜索集合相对应的虚拟服务器名称。
collection-name	要设置新文档根目录路径的搜索集合名称。
document-root	搜索集合的新文档根目录的路径。

- 2. 将配置重新部署到相关的 Web Server 实例。

```
WebServer7-base/bin/wadm deploy-config
[--force] [--restart] [--no-reconfig]
configName
```

在第 135 页中的表 7-11 中记述了命令选项。

自定义命令行脚本

如果已对脚本进行了自定义，如 startsvr、startsvr.bat、stopsvr、stopsvr.bat、restart、reconfig 和 reconfig.bat，则须对位于以下目录的发行版本 5 默认脚本执行同样的自定义：WebServer7-base/bin。

升级回滚

通过全新安装发行版本 5 并随后迁移 Web Server 实例配置，发行版本 4 Web Server 仍旧保持原样。因此，发行版本 5 Web Server 的回滚包括以下步骤，用以回到发行版本 4。

1. 以 root 身份登录，或成为超级用户。

```
su -
```

2. 逐个停止所有正在运行的 Web Server 实例。

```
WebServer7Config-base/https-configName/bin/stopserv
```

如果正确停止了服务器，则会看到一条消息“服务器已关闭”。

3. 删除发行版本 5 Web Server 安装。

必须删除所有的发行版本 5 实例及已迁移的配置：

- a. 删除所有发行版本 5 实例。

```
WebServer7-base/bin/wadm delete-instance --user ...  
--config=configName hostName.domainName
```

- b. 删除所有发行版本 5 配置。

```
WebServer7-base/bin/wadm delete-config --user ... configName
```

4. 如第 127 页中的“升级过程”中所述，重新启动在升级 Web Server 时停止的 Web Server 实例。

从 Java ES 发行版本 3 升级 Web Server

将 Java ES 2005Q1（发行版本 3）Web Server 升级到发行版本 5 的过程与将发行版本 4 Web Server 升级到发行版本 5 的过程相同。

要将发行版本 3 Web Server 升级到发行版本 5，请按照第 123 页中的“从 Java ES 发行版本 4 升级 Web Server”中的说明操作，但需要在提到发行版本 4 的地方用发行版本 3 代替。

从 Java ES 发行版本 2 升级 Web Server

将 Java ES 2004Q2（发行版本 2）Web Server 升级到发行版本 5 的过程与将发行版本 4 Web Server 升级到发行版本 5 的过程相同。

要将发行版本 2 Web Server 升级到发行版本 5，请按照第 123 页中的“从 Java ES 发行版本 4 升级 Web Server”中的说明操作，但需要在提到发行版本 4 的地方用发行版本 2 代替。

注	如果要在 Linux 平台上从发行版本 2 Web Server 升级，则必须执行双重升级，即 Web Server 和操作系统都要升级（RHEL 2.1 上不支持发行版本 5 Web Server）。有关更多信息，参见第 122 页中的“双重升级”。
----------	--

Java DB

本章介绍如何将 Java DB 升级到 Java ES 5（发行版本 5）：Java DB 10.1.3。

本章概要介绍了有关发行版本 5 所支持的不同升级方式的升级注意事项。其内容涵盖了在 Solaris 和 Linux 操作系统上的升级：

- 第 142 页中的“Java DB 升级概述”
- 第 145 页中的“从 Java ES 发行版本 4 升级 Java DB”

注 本章中的文件位置是相对于一个称为 *JavaDB-base* 的目录路径而指定的。此路径由 Java ES 安装程序在安装 Java DB 时所设置。

 此目录路径的值显示在下表中。

表 8-1 Java DB 目录路径		
路径名变量	Solaris OS	Linux OS
<i>JavaDB-base</i>	/opt/SUNWjavadb	/opt/sun/javadb

Java DB 升级概述

本节将在以下方面对 Java DB 进行介绍，这些方面会对升级到 Java ES 5（发行版本 5）产生影响：

- [关于 Java ES 发行版本 5](#)
- [Java ES 发行版本 5 升级指导说明](#)
- [Java DB 数据](#)
- [Java DB 升级策略](#)

关于 Java ES 发行版本 5

Java ES 发行版本 5 Java DB 是第一个即将作为 Java ES 产品组件而交付的发行版本；Java DB 首次是作为共享组件（称为 DerbyDatabase）发行并包括在 Java ES 发行版本 4 中的。

相对于发行版本 4 的版本而言，发行版本 5 Java DB 只是一个次要发行版本。它改进了一些功能，对接口进行了更新，还修复了选定的错误。

Java ES 发行版本 5 升级指导说明

[表 8-2](#) 显示了所支持的、将 Java DB 升级到 Java ES 发行版本 5 的升级方式。此表既适用于 Solaris，也适用于 Linux 操作系统。

表 8-2 升级到 Java ES 5（发行版本 5）：Java DB 10.1.3 的方式

Java ES 发行版本	Java DB 版本	常规方法	必需的重新配置
发行版本 4	Derby Database 10.0.2	直接升级： 通过全新安装替换发行版本 4。不会影响到持久数据。	无

Java DB 数据

下表显示了可能受 Java DB 软件升级影响的数据类型。

表 8-3 Java DB 数据用途

数据类型	位置	用途
配置数据	实例配置是特定于应用程序的，存储在 Java DB 数据库中。	Java DB 实例的配置
持久数据	数据库目录及其内容是特定于应用程序的。其位置由数据库连接 URL <code>jdbc:derby:full path to database</code> 指定。	数据库和用户证书

Java DB 升级策略

Java DB 的升级策略通常取决于在第 1 章“升级规划”中讨论的许多注意事项：升级方式、Java ES 组件之间的依赖性、选择性升级与全部升级、多实例部署，等等。

本节通过提出可能影响 Java DB 升级计划的问题，对 Java DB 的一般注意事项进行逐一介绍。

兼容性问题

发行版本 5 Java DB 与发行版本 4 的版本向后兼容。

组件依赖性

Java DB 仅依赖于 J2SE 共享组件（参见第 43 页中的表 1-9）。

双重升级

在双重升级中，Java DB 和操作系统都要进行升级（如第 41 页中的“双重升级：Java ES 和操作系统软件”中所述）。可使用以下两种方式之一执行双重升级：

- 操作系统全新安装
- 操作系统就地升级

操作系统全新安装

1. 备份现有 Java DB 数据。

有关基本数据的位置，参见第 143 页中的“Java DB 数据”。

2. 安装新的操作系统。

可将操作系统安装在新系统上（或 Solaris 10 区域中），否则会擦除现有文件系统。

3. 安装发行版本 5 Java DB。

4. 恢复在步骤 1 中备份的 Java DB 数据。

操作系统就地升级

1. 备份现有 Java DB 数据。

有关基本数据的位置，参见第 143 页中的“Java DB 数据”。

2. 升级操作系统。

升级会将现有文件系统留在原来的位置。

3. 将 Java DB 升级到发行版本 5。

参见第 145 页中的“从 Java ES 发行版本 4 升级 Java DB”。

Java DB 数据应不受升级的影响。

从 Java ES 发行版本 4 升级 Java DB

本节包含关于将 Java DB 从 Java ES 2005Q4（发行版本 4）升级到 Java ES 5（发行版本 5）的信息。其中涵盖以下主题：

- [简介](#)
- [发行版本 4 Java DB 的升级](#)
- [多实例升级](#)

简介

将 Java ES 发行版本 4 Java DB 升级到发行版本 5 时，请考虑升级过程中的以下方面：

- **常规升级方法。** 通过执行发行版本 5 Java DB 的全新安装以替换发行版本 4 的版本来实现升级。发行版本 4 数据和配置保持不变。
- **升级依赖性。** Java DB 对 J2SE 共享组件具有硬性升级依赖性（参见[第 43 页中的表 1-9](#)），该共享组件会在您升级 Java DB 时，由 Java ES 安装程序自动升级到发行版本 5。
- **向后兼容性。** 发行版本 5 Java DB 与发行版本 4 完全兼容。
- **升级回滚。** 发行版本 5 的升级回滚只有通过复原到先前备份的发行版本 4 安装才能实现。
- **平台问题。** 在 Solaris 和 Linux 操作系统上，升级 Java DB 的常规方法相同。

发行版本 4 Java DB 的升级

本节介绍如何在 Solaris 和 Linux 平台上将 Java DB 从 Java ES 发行版本 4 升级到 Java ES 发行版本 5。其中涵盖以下主题：

- [升级前的任务](#)
- [升级发行版本 4 Java DB](#)
- [检验升级](#)
- [升级后的任务](#)
- [升级回滚](#)

升级前的任务

升级 Java DB 软件之前，应执行以下任务：

- [检验当前版本信息](#)
- [升级 Java DB 依赖项](#)
- [备份 Java DB 数据](#)
- [获取所需的配置信息和密码](#)

检验当前版本信息

可使用以下命令检验 Java DB 的当前版本：

```
java -cp JavaDB-base/lib/derby.jar org.apache.derby.tools.sysinfo
```

版本信息（对于 `derby.jar` 文件）显示在下表中：

表 8-4 Java DB 版本检验输出

Java ES 发行版本	Java DB 版本号
发行版本 4	10.0.2.1
发行版本 5	10.1.3.1

升级 Java DB 依赖项

通常建议将计算机系统上（以及计算环境中）的所有 Java ES 组件都升级到 Java ES 发行版本 5。但是，在执行 Java DB 到发行版本 5 的升级时，Java ES 安装程序会自动升级 Java DB 所需的所有共享组件。

备份 Java DB 数据

将 Java DB 从发行版本 4 升级到发行版本 5 不会修改配置数据或持久数据。不过，为安全起见，您应备份整个 Java DB 安装和数据。参见第 143 页中的表 8-3。

获取所需的配置信息和密码

升级 Java DB 不需要任何配置信息或密码。

升级发行版本 4 Java DB

本节介绍 Solaris 和 Linux 平台上的升级过程。

升级过程

下述过程适用于本地驻留在即将进行升级的计算机上的所有 Java DB 实例。

1. 以 root 身份登录，或成为超级用户。

```
su -
```

2. 停止发行版本 4 Java DB。

如果网络服务器正在运行，则使用以下命令：

```
java -cp JavaDB-base/lib/derby.jar:JavaDB-base/lib/derbynet.jar  
org.apache.derby.drda.NetworkServerControl shutdown
```

否则，只需关闭所有使用 Java DB 的应用程序。

3. 执行发行版本 5 Java DB 的全新安装。

执行以下步骤：

- a. 在托管发行版本 4 Java DB 的计算机上启动 Java ES 安装程序。

```
cd Java ES Release 5 distribution/os_arch  
./installer
```

其中，*os_arch* 与您的平台匹配，如 *Solaris_sparc*。（对于命令行界面，使用 *installer -nodisplay* 选项。）

显示“欢迎”和“许可协议”页面之后，将向您显示一个组件选择页面。（如果检测到可由 Java ES 安装程序直接升级的已安装组件，它们的状态会显示为“可升级”。）

- b. 从组件选择页面中选择 Java DB。

先前的安装将被覆盖。

- c. 选择“以后再配置”选项。

不支持“现在配置”选项。

- d. 如果需要，选择安装本地化软件包的选项。

- e. 安装完成后退出 Java ES 安装程序。

4. 启动发行版本 5 Java DB。

如果正在使用网络服务器，则使用以下命令：

```
java -jar <JavaDB-base>/lib/derbynet.jar start
```

否则，只需在嵌入式模式下启动任何使用 Java DB 的应用程序。

检验升级

可使用以下命令检验 Java DB 的升级是否成功：

```
java -cp JavaDB-base/lib/derby.jar org.apache.derby.tools.sysinfo
```

参见第 146 页中的表 8-4 确认输出值（对于 *derby.jar* 文件的版本）。

升级后的任务

将 Java DB 从发行版本 4 升级到发行版本 5 时，必须将数据从 Java DB 10.0 磁盘格式转换为 10.1 格式。要执行此转换，请将 `upgrade=true` 附加到 JDBC URL 以连接到数据库。例如：

```
java -cp JavaDB-base/lib/derbytools.java:JavaDB-base/lib/derby.jar
org.apache.derby.tools.ij
```

```
ij version 10.1
```

```
ij> connect 'jdbc:derby:/databasePath;upgrade=true';
```

```
ij> exit;
```

升级回滚

发行版本 5 的升级回滚只有通过复原到备份的发行版本 4 安装和数据才能实现。

多实例升级

在一些部署体系结构中，Java DB 被部署在多个计算机系统上，以提供可伸缩性和提高可用性。例如，您可以在多台计算机上运行 Java DB 实例，并使用一个负载平衡器来分配负载。

如第 146 页中的“发行版本 4 Java DB 的升级”中所述，在每台计算机上升级 Java DB。

High Availability Session Store

本章介绍如何将 High Availability Session Store 升级到 Java ES 5（发行版本 5）：High Availability Session Store (HADB) 4.4.3。

本章在介绍发行版本 5 所支持的各种不同升级方式之前概括介绍了一些升级问题。其内容涵盖了在 Solaris 和 Linux 操作系统上的升级：

- 第 152 页中的“HADB 升级概述”
- 第 154 页中的“从 Java ES 发行版本 4 升级 HADB”
- 第 160 页中的“从 Java ES 发行版本 3 升级 HADB”

注

本章中的文件位置是相对于一个称为 *HADB-base* 的目录路径而指定的。在最初安装 HADB 时，可能至少已将此路径的一部分指定为安装目录。否则，安装程序会分配一个默认值。
HADB-base 的默认值不依赖于操作系统平台，如下表所示。

表 9-1	HADB 目录路径	
路径名变量	Solaris OS	Linux OS
<i>HADB-base</i>	/opt/SUNWhadb/ <i>version_number</i>	/opt/SUNWhadb/ <i>version_number</i>
Java ES 安装程序		
<i>HADB-base</i>	/opt/SUNWappserver/appserver	/opt/SUNWappserver/appserver
独立的 Application Server	/hadb/ <i>version_number</i>	/hadb/ <i>version_number</i>
8.2 EE 安装程序		

HADB 升级概述

本节将在以下方面对 HADB 进行介绍，这些方面会对升级到 Java ES 5（发行版本 5）产生影响：

- [关于 Java ES 发行版本 5 HADB](#)
- [HADB 升级指导说明](#)
- [HADB 数据](#)
- [HADB 升级策略](#)

关于 Java ES 发行版本 5 HADB

相对于发行版本 4 HADB 而言，Java ES 发行版本 5 版本的 HADB 只是一个次要的用户功能增强版本。

HADB 升级指导说明

[表 9-2](#) 显示了所支持的、将 HADB 升级到 Java ES 发行版本 5 的升级方式。此表既适用于 Solaris，也适用于 Linux 操作系统。

表 9-2 升级到 Java ES 5（发行版本 5）：HADB 4.4.3 的方式

Java ES 发行版本	HADB 版本	常规方法	必需的重新配置
发行版本 4	HADB 4.4.2 (2005Q4)	直接升级： 联机升级和脱机升级方法均可。	无
发行版本 3	HADB 4.4.1 (2005Q1)	直接升级： 联机升级和脱机升级方法均可。	无
发行版本 2	HADB 4.4.0-14 (2004Q2)	不支持升级。	无
发行版本 1	不可用	无升级。	无
Java ES 以前的发行版本	不可用	无升级。	无

HADB 数据

下表显示了可能受 HADB 软件升级影响的数据类型。

表 9-3 HADB 数据用途

数据类型	位置	用途
动态应用程序数据	/var/opt/SUNWhadb	High availability session store
配置数据	/etc/opt/SUNWhadb /etc/init.d/ma-initd	高可用性服务器配置

HADB 升级策略

HADB 的升级策略通常取决于在[第 1 章“升级规划”](#)中讨论的许多注意事项：升级方式、Java ES 组件之间的依赖性、选择性升级与全部升级、多实例部署，等等。

本节通过提出可能影响 HADB 升级计划的问题，对 HADB 的一般注意事项进行逐一介绍。

兼容性问题

发行版本 5 HADB 与 Java ES 发行版本 4 随附的 HADB 向后兼容。

HADB 组件依赖性

发行版本 5 HADB 仅对 J2SE 共享组件具有依赖性，即 Java™ 2 Platform, Standard Edition (J2SE™) 版本 1.4 或更新版本。

双重升级

在双重升级中，HADB 软件和操作系统都要进行升级（如[第 41 页](#)中的“[双重升级：Java ES 和操作系统软件](#)”中所述）。双重升级在 Application Server 双重升级上下文中执行。有关 Application Server 双重升级的信息，参见[第 190 页](#)中的“[双重升级](#)”。

从 Java ES 发行版本 4 升级 HADB

本节包含关于将 HADB 从 Java ES 2005Q4（发行版本 4）升级到 Java ES 5（发行版本 5）的信息。其中涵盖以下主题：

- [简介](#)
- [发行版本 4 HADB 的升级](#)

简介

将 Java ES 发行版本 4 HADB 升级到 Java ES 发行版本 5 时，请考虑升级过程中的以下方面：

- **常规升级方法。** 升级包括删除 Java ES 发行版本 4 HADB 软件包和添加 Java ES 发行版本 5 软件包。提供了以下两种升级方法：
 - **联机升级。** 使用联机升级可避免 HADB 服务中断。
 - **脱机升级。** 如果在用更新版本替换 HADB 软件包时可以中断 HADB 服务，则使用脱机升级。
- **升级依赖性。** HADB 没有硬性升级依赖性。HADB 要求 J2SE 版本 1.4 或更新版本，这意味着它对 J2SE 具有软性升级依赖性。
- **向后兼容性。** Java ES 发行版本 5 随附的 HADB 与 Java ES 发行版本 4 随附的 HADB 向后兼容。
- **升级回滚。** 从 Java ES 发行版本 5 到 Java ES 发行版本 4 的升级回滚是通过恢复发行版本 4 版本的软件包实现的。
- **平台问题。** 在 Solaris 和 Linux 操作系统上，升级 HADB 的常规方法相同。

发行版本 4 HADB 的升级

本节介绍如何在 Solaris 和 Linux 平台上将 HADB 从 Java ES 发行版本 4 升级到 Java ES 发行版本 5。其中涵盖以下主题：

- [升级前的任务](#)
- [升级发行版本 4 HADB](#)
- [检验升级](#)
- [升级后的任务](#)
- [升级回滚](#)

升级前的任务

升级 HADB 软件之前，应执行以下任务：

- [检验当前版本信息](#)
- [升级 HADB 依赖项](#)
- [备份目录数据和配置文件](#)
- [获取所需的配置信息和密码](#)

检验当前版本信息

可使用标准版检查实用程序来检验 HADB 的当前版本。例如：

在 Solaris 上：

```
pkgparam -v SUNWhadba
```

在 Linux 上：

```
rpm -qi sun-hadb-a-4.4.3-5.i386.rpm
```

表 9-4 HADB 版本检验输出

Java ES 发行版本	HADB 版本号
发行版本 2	VERSION=4.4.0,REV=14 SUNW_PRODVERS=4.4.0
发行版本 3	VERSION=4.4.1,REV=7 SUNW_PRODVERS=4.4.1
发行版本 4	VERSION=4.4.2,REV=7 SUNW_PRODVERS=4.4.2
发行版本 5	VERSION=4.4.3,REV=5 SUNW_PRODVERS=4.4.3

升级 HADB 依赖项

通常建议将计算机系统（以及计算环境）中的所有 Java ES 组件都升级到 Java ES 发行版本 5。但是，在执行 HADB 到发行版本 5 的升级时，Java ES 安装程序会自动升级 HADB 所需的所有共享组件（即 J2SE）。

备份目录数据和配置文件

将 HADB 从 Java ES 发行版本 4 升级到 Java ES 发行版本 5 本身并不会修改 HADB 动态数据。但是，您可以备份 Java ES 发行版本 4 软件包，以备您需要回滚升级时使用。

此外，如果自上次安装 HADB 以来您对以下文件进行过修改，则备份这些文件。

```
/etc/opt/SUNWhadb/mgt.cfg
/etc/init.d/ma-initd
```

获取所需的配置信息和密码

HADB 升级要求您知道超级用户密码。

升级发行版本 4 HADB

本节讨论影响 HADB 升级过程的一些注意事项，随后对升级过程本身加以说明。

升级注意事项

将 HADB 软件升级到 Java ES 发行版本 5 时，应考虑以下注意事项：

- 您需要根据生产要求确定联机升级或脱机升级哪一个更合适。
- 下表显示了用于 Solaris 和 Linux 平台的 Java ES 发行版本 5 升级软件包。Solaris 软件包按其安装顺序列出。

表 9-5 用于在 Solaris 平台升级 HADB 的软件包版本

Solaris 软件包	Linux 软件包
SUNWhadba	sun-hadb-a-4.4.3-5.i386.rpm
SUNWhadbc	sun-hadb-c-4.4.3-5.i386.rpm
SUNWhadbe	sun-hadb-e-4.4.3-5.i386.rpm
SUNWhadbi	sun-hadb-i-4.4.3-5.i386.rpm
SUNWhadbj	sun-hadb-j-4.4.3-5.i386.rpm
SUNWhadbm	sun-hadb-m-4.4.3-5.i386.rpm
SUNWhadbs	sun-hadb-s-4.4.3-5.i386.rpm
SUNWhadbv	sun-hadb-v-4.4.3-5.i386.rpm
SUNWhadbx	sun-hadb-x-4.4.3-5.i386.rpm

HADB 的联机升级

执行 HADB 的联机升级时，应首先在要升级的群集中的每台服务器上安装、启动和检验发行版本 5 HADB。然后，每台服务器从早期安装的 HADB 中注销，并注册到新安装的 HADB 版本中。

有关执行联机升级的详细信息，参阅《Sun Java System Application Server Enterprise Edition 8.2 高可用性管理指南》中的相应章节 (<http://docs.sun.com/doc/820-0851/6ncihgipa?a=view>)。

HADB 的脱机升级

从 Java ES 发行版本 3 或发行版本 4 升级时，可以使用 HADB 的脱机升级。

要执行脱机升级，请关闭 HADB 服务，用 Java ES 5（发行版本 5）分发包中提供的更新版本替换现有 HADB 软件包，如第 156 页中的表 9-5 中所示。

1. 以 root 身份登录，或成为超级用户。

```
su -
```

2. 关闭所有 HADB 服务。

- a. 列出正在运行的所有数据库。

```
HADB-base/bin/hadbm list
```

- b. 关闭列出的每个数据库。

```
HADB-base/bin/hadbm stop databaseName
```

如果某个数据库已处于停止状态，则略过该消息。

- c. 在运行管理代理的每台主机上，关闭 HADB 管理代理：

```
/etc/init.d/ma-initd stop
```

（如果在安装 HADB 时，使用的是独立的 Application Server 安装程序而非 Java ES 安装程序，则 ma-initd 脚本位于 HADB-base/bin 中。）

3. 启动 Java ES 安装程序。

```
cd Java ES Release 5 distribution/os_arch
./installer
```

其中，os_arch 与您的平台匹配，如 Solaris_sparc。（对于命令行界面，使用 installer -nodisplay 选项。）

显示“欢迎”和“许可协议”页面之后，将向您显示一个组件选择页面。（如果检测到可由 Java ES 安装程序直接升级的已安装组件，它们的状态会显示为“可升级”。）

4. 在组件选择页面中选择 High Availability Session Store 4.4。
如果已经选择了 Application Server Enterprise Edition 8.2, 则会自动选中 HADB。
5. 确认您的升级选择。
将升级 HADB 软件包并显示升级摘要。
6. 退出 Java ES 安装程序。
7. 恢复在[第 156 页](#)中的“[备份目录数据和配置文件](#)”中所备份的文件。
8. 检验符号链接 /opt/SUNWhadb/4 现在是否指向 *HADB-base*。

例如, 对于默认的 *HADB-base*:

```
ls -l /opt/SUNWhadb/4
lrwxrwxrwx 1 root other 7 Jul 7 23:18 /opt/SUNWhadb/4 ->
4.4.3-5/
```

9. 重新启动[步骤 2](#)中关闭的 HADB 管理代理:

```
/etc/init.d/ma-initd start
```

(如果在安装 HADB 时, 使用的是独立的 Application Server 安装程序而非 Java ES 安装程序, 则 ma-initd 脚本位于 *HADB-base/bin* 中。)

检验升级

联机升级完成后, 使用以下过程检验升级。在核实升级成功之后, 可删除旧安装软件包。

要检验当前运行的进程是否在使用已升级的 HADB 服务, 可执行以下步骤。

1. 对于所有正在运行的 HADB 服务, 发出以下两个命令之一:

```
HADB-base/bin/ma -V
HADB-base/bin/hadbm -V
```

例如,

```
HADB-base/bin/ma -V
Sun Java System High Availability Database 4.4 Database Management
Agent
Version      : 4.4.3.5 [V4-5-3-5 2006-03-31 13:59:50 pakker@astra07]
(SunOS_5.9_sparc)
```

2. 检查数据库是否在运行。

```
HADB-base/bin/hadbm status -n databaseName
```

例如，对于名为 Example DB 的数据库，输入以下命令。

```
HADB-base/bin/hadbm list
```

```
Database
ExampleDB
```

```
HADB-base/bin/hadbm status ExampleDB
```

```
Database      Status
ExampleDB     FaultTolerant
```

```
HADB-base/bin/hadbm status -n ExampleDB
```

NodeNo	HostName	Port	NodeRole	NodeState	MirrorNode
0	sungod012	15000	active	running	1
1	sungod012	15020	active	running	0

所列节点的所有 HADB 服务都应处于 "running" 状态。

3. 检验使用 HADB 的所有产品是否都在使用新的 HADB 路径。

```
HADB-base/bin/hadbm get PackageName databaseName
```

例如，对于名为 Example DB 的数据库，输入以下命令。

```
HADB-base/bin/hadbm get PackageName ExampleDB
```

```
Attribute      Value
PackageName    V4.4.3.5
```

以上命令会显示 HADB 的当前版本。要获取详细列表，请发出以下命令：

```
HADB-base/bin/hadbm get --all ExampleDB
```

升级后的任务

除第 154 页中的“从 Java ES 发行版本 4 升级 HADB”中所述的步骤之外，没有任何升级后的任务。

升级回滚

要回滚 HADB 的升级，请使用先前备份的版本替换您安装的 HADB 软件包的更新版本，如第 156 页中的“备份目录数据和配置文件”中所述。

从 Java ES 发行版本 3 升级 HADB

将 Java ES 2005Q1（发行版本 3）HADB 升级到发行版本 5 的过程与将发行版本 4 HADB 升级到发行版本 5 的过程相同。

要将发行版本 3 HADB 升级到发行版本 5，请按照第 154 页中的“从 Java ES 发行版本 4 升级 HADB”中的说明操作，但需要在提到发行版本 4 的地方用发行版本 3 代替。

Message Queue

本章介绍如何将 Message Queue 软件从先前的 Java ES 版本升级到 Java ES 5（发行版本 5）：Sun Java System Message Queue 3.7 UR1。

本章概要介绍了有关发行版本 5 所支持的不同升级方式的升级注意事项。其内容涵盖了在 Solaris 和 Linux 操作系统上的升级：

- [第 162 页中的“Message Queue 升级概述”](#)
- [第 167 页中的“从 Java ES 发行版本 4 升级 Message Queue”](#)
- [第 173 页中的“从 Java ES 发行版本 3 升级 Message Queue”](#)
- [第 174 页中的“从 Java ES 发行版本 2 升级 Message Queue”](#)

注 本章中的文件位置是相对于一个称为 *MessageQueue-base* 的固定目录路径而指定的。

MessageQueue-base 的值视操作系统平台而定，如下表所示。

表 10-1 Message Queue 目录路径

路径名变量	Solaris OS	Linux OS
<i>MessageQueue-base</i>	/usr/bin	/opt/sun/mq/bin

Message Queue 升级概述

本节将在以下方面对 Message Queue 进行介绍，这些方面会对升级到 Java ES 5（发行版本 5）产生影响：

- [关于 Java ES 发行版本 5 Message Queue](#)
- [Message Queue 升级指导说明](#)
- [Message Queue 数据](#)
- [Message Queue 升级策略](#)

关于 Java ES 发行版本 5 Message Queue

相对于发行版本 4 而言，Java ES 发行版本 5 Message Queue 只是次要升级。它主要修复了代码错误，并无任何小的功能增强。

过去，Message Queue 软件包括两种版本，即 Platform Edition 和 Enterprise Edition，每种版本分别对应不同的功能集和许可能力。Enterprise Edition 用于在企业生产环境中部署和运行消息传送应用程序。Platform Edition 主要用于开发、调试及加载测试消息传送应用程序和组件。对于发行版本 5 Message Queue，Platform Edition 已被废弃，Message Queue 包括所有的 Enterprise Edition 功能。从早期 Java ES 发行版本升级到发行版本 5 会将所有已安装的 Platform Edition 转换为完整的 Message Queue 企业级功能。

Message Queue 升级指导说明

表 10-2 显示了所支持的、将 Message Queue 升级到 Java ES 发行版本 5 的升级方式。此表既适用于 Solaris，也适用于 Linux 操作系统。

表 10-2 升级到 Java ES 5（发行版本 5）：Message Queue 3.7 UR1 的方式

Java ES 发行版本	Message Queue 版本	常规方法	必需的重新配置
发行版本 4	Sun Java System Message Queue 2005Q4 (3.6 SP3) 仅限 Enterprise Edition	直接升级： 使用 Java ES 安装程序执行。	自动执行数据转换。
发行版本 3	Sun Java System Message Queue 2005Q1 (3.6) 仅限 Enterprise Edition	直接升级： 使用 Java ES 安装程序执行。	自动执行数据转换。
发行版本 2	Sun Java System Message Queue 2004Q2 (3.5 SP1) Platform Edition 和 Enterprise Edition	直接升级： 使用 mqupgrade 脚本执行。	在 Solaris 平台自动执行，在 Linux 平台可使用 mqmigrate 脚本。
发行版本 1	Sun Java System Message Queue 2003Q4 (3.0.1 SP2) Platform Edition 和 Enterprise Edition	直接升级未经验证： 但可以使用 mqupgrade 脚本执行。 ¹	在 Solaris 平台自动执行，在 Linux 平台可使用 mqmigrate 脚本。
Java ES 以前的发行版本	Sun Java System Message Queue 3.0.x 和早期版本 Platform Edition 和 Enterprise Edition	直接升级未经验证： 但可以使用 Java ES 安装程序执行。	

1. 可能需要在运行 mqupgrade 脚本之前备份以下文件并在运行该脚本之后将这些文件恢复：例如（在 Solaris OS 上）：将 /etc/imq/passwd 和 /etc/imq/accesscontrol.properties 恢复为 /var/imq/instances/instanceName/etc/

除了在表 10-2 中所示的 Java ES 发行版本的 Message Queue 之外，Message Queue 还与 Solaris OS 软件捆绑在一起。要将捆绑版本的 Message Queue 升级到发行版本 5，可使用 Java ES 安装程序来执行。

Message Queue 数据

与其他 Java ES 组件一样，Message Queue 将利用对于任一特定升级而言可能需要迁移至升级版本的各种数据。下表显示了可能受 Message Queue 软件升级影响的数据的类型。

表 10-3 显示了数据在 Solaris 系统上的位置。数据在 Linux 系统上的位置与之相似，但取决于 Message Queue 的版本：

- 发行版本 2：用 /opt/imq 替换表中的 /imq
- 发行版本 3 及更高版本：用 /opt/sun/mq 替换表中的 /imq

有关更多信息，参见 Message Queue 3.7 UR1 管理指南 (<http://docs.sun.com/app/docs/doc/820-0843/6nci9seed?a=view>)。

在表 10-3 中，*instanceName* 用于标识与数据关联的 Message Queue 代理实例的名称。

表 10-3 Message Queue 数据用途 (Solaris OS)

数据类别	位置（在 Solaris 上）	用途
代理实例配置属性	/var/imq/instances/ <i>instanceName</i> /props/config.properties	代理和相关服务配置
动态应用程序数据的持久存储库	发行版本 2、发行版本 3 和发行版本 4： /var/imq/instances/ <i>instanceName</i> /fs350/ 发行版本 5： /var/imq/instances/ <i>instanceName</i> /fs370/ 或 JDBC 可以访问的数据存储库	存储消息、目标、持久的订阅、事务和其他动态数据
受管对象（对象存储）	所选的本地目录 或 LDAP Directory Server	用于配置客户机 / 代理连接的对象
安全：用户信息库	/var/imq/instances/ <i>instanceName</i> /etc/passwd 或 LDAP 目录服务器	存储用于验证和授权的用户数据
安全：访问控制文件（默认位置）	/var/imq/instances/ <i>instanceName</i> /etc/accesscontrol.properties	设置授予用户对目标和相关功能访问权的规则
安全：密码文件目录（默认位置）	/var/imq/instances/ <i>instanceName</i> /etc/	存储加密的密码信息
安全：代理的密钥库文件位置	/etc/imq/	存储用于安全消息传送的加密证书信息

Message Queue 升级策略

Message Queue 的升级策略通常取决于在第 1 章“升级规划”中讨论的许多问题：升级方式、Java ES 组件之间的依赖性、选择性升级与全部升级、多实例部署，等等。

本节通过提出可能影响 Message Queue 升级计划的问题，对 Message Queue 的一般注意事项进行逐一介绍。

兼容性问题

发行版本 5 Message Queue 未在发行版本 3 或发行版本 4 的基础上引入任何新的不兼容性。不过，相对于发行版本 2 及早期版本，发行版本 5 却存在着明显的兼容性问题。这些问题在第 175 页中的“发行版本 2 兼容性问题”中进行了讨论。

此外，一般而言，如果在一个群集中既有发行版本 4 及早期版本的 Message Queue 代理，又有发行版本 5 Message Queue 代理，则主代理一定是早期发行版本代理，且群集将作为早期发行版本的 Message Queue 群集运行。

Message Queue 组件依赖性

Message Queue 对其他 Java ES 组件的依赖性会影响升级和重新配置 Message Queue 软件的过程。例如，Message Queue 接口或功能上的变化可能要求 Message Queue 所依赖组件的升级版本。是否需要升级此类组件取决于具体的升级方式。

Message Queue 对以下 Java ES 组件具有依赖性：

- **共享组件。** Message Queue 对特定的 Java ES 共享组件具有依赖性（参见第 43 页中的表 1-9）。
- **Directory Server。** Message Queue 对 Directory Server 具有可选依赖性：可将 Message Queue 配置为在 LDAP 目录 (Directory Server) 中存储受管对象和 / 或用户数据，而非在本地存储。
- **Web 容器。** Message Queue 对 Web Server、Application Server 或第三方 Web 容器具有可选依赖性，以支持客户机与代理之间的 HTTP 消息传送。
- **数据库。** Message Queue 对 Java DB（或第三方数据库）具有可选依赖性，以便为 Message Queue 持久层提供 JDBC 可以访问的数据存储库，而非平面文件消息存储库。
- **Sun Cluster。** Message Queue 对 Sun Cluster 具有可选依赖性，以提供高可用性支持。

双重升级

在双重升级中，Message Queue 和操作系统都要进行升级（如第 41 页中的“[双重升级：Java ES 和操作系统软件](#)”中所述）。可使用以下两种方式之一执行双重升级：

- 操作系统全新安装
- 操作系统就地升级

操作系统全新安装

1. 备份现有 Message Queue 数据。

有关基本数据的位置，参见第 164 页中的“[Message Queue 数据](#)”。

2. 安装新的操作系统。

可将操作系统安装在新系统上（或 Solaris 10 区域中），否则会擦除现有文件系统。

3. 安装发行版本 5 Message Queue。

4. 恢复或迁移在[步骤 1](#)中备份的 Message Queue 数据。

在 Linux 上从发行版本 2 Message Queue 升级时，数据将恢复到发行版本 5 位置。

操作系统就地升级

1. 备份现有 Message Queue 数据。

有关基本数据的位置，参见第 164 页中的“[Message Queue 数据](#)”。

2. 升级操作系统。

升级会将现有文件系统留在原来的位置。

3. 将 Message Queue 升级到发行版本 5。

参见本章中的相应小节，这要视升级方式而定。经过升级后，现有的 Message Queue 数据应保留在原来的位置。

不过，在 Linux 上从发行版本 2 Message Queue 升级时，必须将数据移动到发行版本 5 位置。

从 Java ES 发行版本 4 升级 Message Queue

本节包含关于将 Message Queue 从 Java ES 2005Q4（发行版本 4）升级到 Java ES 5（发行版本 5）的信息。其中涵盖以下主题：

- [简介](#)
- [发行版本 4 Message Queue 的升级](#)
- [多实例升级](#)

简介

将 Java ES 发行版本 4 Message Queue 升级到发行版本 5 时，请考虑升级过程中的以下方面：

- **常规升级方法。** 使用 Java ES 安装程序执行升级。安装程序会自动从发行版本 4 迁移配置数据。此外，在首次运行 `imqbrokerd` 时，会自动转换与发行版本 4 关联的任何动态应用程序数据。对于基于文件的存储库，这意味着 `fs350` 目录的内容将会复制到新的 `fs370` 目录中。对于 JDBC 存储库，将会对现有的数据库表进行简单的版本更新。
- **升级依赖性。** Message Queue 对许多 Java ES 共享组件都具有依赖性（参见 [第 43 页中的表 1-9](#)），所有这些组件会在您升级 Message Queue 时，由 Java ES 安装程序自动将它们升级到发行版本 5。

此外，发行版本 5 Message Queue 还依赖于 Java ES 产品组件，如 [第 165 页中的“Message Queue 组件依赖性”](#) 中所述。不过，在将 Message Queue 升级到发行版本 5 时，不必一定升级这些组件。

- **向后兼容性。** 就协议、代理兼容性、受管对象、管理工具以及客户机应用程序而言，发行版本 5 Message Queue 与发行版本 4 完全兼容。
- **升级回滚。** 没有用于将 Message Queue 升级回滚至发行版本 4 的实用程序。您必须删除已升级的组件，手动恢复先前的版本和配置数据。
- **平台问题。** 在 Solaris 和 Linux 操作系统上，升级 Message Queue 的方法相同。

发行版本 4 Message Queue 的升级

本节介绍如何执行 Message Queue 从 Java ES 发行版本 4 到 Java ES 发行版本 5 的升级：

- [升级前的任务](#)
- [升级发行版本 4 Message Queue](#)
- [检验 Message Queue 升级](#)
- [升级后的任务](#)
- [升级回滚](#)

升级前的任务

升级 Message Queue 软件之前，应执行以下任务：

- [检验当前版本信息](#)
- [升级 Message Queue 依赖项](#)
- [备份 Message Queue](#)

检验当前版本信息

可通过使用 `-version` 选项启动 Message Queue 代理，确定系统上安装的 Message Queue 的版本和版 (edition)：

```
imqbrokerd -version
```

表 10-4 Message Queue 版本检验输出

Java ES 发行版本	Message Queue 版本号
发行版本 2	Sun Java (tm) System Message Queue 3 2004Q2 版本： 3.5
发行版本 3	Sun Java (tm) System Message Queue 3 2005Q1 版本： 3.6
发行版本 4	Sun Java (tm) System Message Queue 3 2005Q4 版本： 3.6 SP3
发行版本 5	Sun Java (tm) System Message Queue 3.7 版本： 3.7 UR1

升级 Message Queue 依赖项

通常建议将计算机系统（以及计算环境）中的所有 Java ES 组件都升级到 Java ES 发行版本 5。Message Queue 仅对少数几个共享组件具有硬性升级依赖性。

升级 Message Queue 依赖项时，应在升级 Message Queue 之前按以下顺序执行（跳过任何可能已升级的依赖项）。共享组件的升级是由 Java ES 安装程序自动实现的。

1. **共享组件。** 第 59 页中的“升级 Java ES 共享组件”中提供了将 Java ES 共享组件同步到发行版本 5 的说明。不过，在将 Message Queue 升级到发行版本 5 时，Java ES 安装程序会自动升级 Message Queue 所需的所有共享组件。
2. **Sun Cluster（软性升级依赖性）。** 第 69 页中的第 3 章“Sun Cluster 软件”提供了将 Sun Cluster 升级到发行版本 5 的说明。
3. **Directory Server（软性升级依赖性）。** 第 89 页中的第 5 章“Directory Server”提供了将 Directory Server 升级到发行版本 5 的说明。
4. **Java DB（软性升级依赖性）。** 在升级 Message Queue 时，您需要执行发行版本 5 Java DB 的全新安装。
5. **Web 容器软件（软性升级依赖性）。** 第 119 页中的第 7 章“Web Server”和第 183 页中的第 11 章“Application Server”分别提供了有关升级 Web Server 和 Application Server 的说明。

备份 Message Queue

最好在执行升级之前备份生产环境中的应用程序数据。记下第 164 页中的表 10-3 中所指出的动态应用程序数据持久存储库的位置。

升级发行版本 4 Message Queue

升级过程由下列步骤组成：

1. 停止运行任何 Message Queue 客户机应用程序。

如果正在 Application Server 环境中使用 Message Queue，则还要关闭 Application Server。

2. 停止运行任何代理。将提示您输入管理员用户名和密码。

```
imqcmd shutdown bkr [-b hostName:port]
```

3. 如果不想保留动态数据、Message Queue 平面文件用户信息库以及与每个代理实例关联的 Message Queue 访问控制文件，请使用以下命令删除。

```
imqbrokerd -name instanceName -remove instance
```

否则，将会保留动态数据和配置信息供发行版本 5 的 Message Queue 使用。

4. 以 Root 身份登录。

```
su -
```

5. 启动 Java ES 安装程序。

```
cd Java ES Release 5 distribution/os_arch  
./installer
```

其中, *os_arch* 与您的平台匹配, 如 *Solaris_sparc*。(对于命令行界面, 使用 `installer -nodisplay` 选项。)

显示“欢迎”和“许可协议”页面之后, 将向您显示一个组件选择页面。(如果检测到可由 Java ES 安装程序直接升级的已安装组件, 它们的状态会显示为“可升级”。)

6. 在组件选择页面中选择 Message Queue。

7. 确认您的升级选择。

将升级 Message Queue 软件包并显示升级摘要。

8. 退出 Java ES 安装程序。

检验 Message Queue 升级

完成升级过程后, 通过使用 `-version` 选项启动 Message Queue 代理, 检验升级是否成功。

```
imqbrokerd -version
```

该命令会返回 Java ES 版本号和特定于 Message Queue 的版本号。

升级后的任务

如果升级了 Web 容器并要使用 Message Queue HTTP 隧道 servlet, 则您可能需要新的 Web 容器中重新部署它。与发行版本 4 相比, 发行版本 5 的 HTTP 隧道 servlet 没有任何变化。有关 HTTP 支持的更多信息, 参见 Message Queue 3.7 UR1 管理指南 (<http://docs.sun.com/doc/820-0843>)。

另外, 如果您确定不会再执行升级回滚, 可删除位于 `fs350` 目录中的发行版本 4 基于文件的数据存储库 (参见第 164 页中的表 10-3)。

升级回滚

尚未提供将 Message Queue 回滚到升级前状态的脚本。该过程必须按照以下步骤手动执行：

1. 停止运行任何 Message Queue 客户机应用程序。

2. 停止运行任何代理。将提示您输入管理员用户名和密码。

```
imqcmd shutdown bkr [-b hostName:port]
```

3. 如果想删除动态数据、Message Queue 平面文件用户信息库以及与每个代理实例关联的 Message Queue 访问控制文件，请使用以下命令删除。

```
imqbrokerd -name instanceName -remove instance
```

4. 以 root 身份登录，或成为超级用户。

```
su -
```

5. 使用以下命令获取安装的 Message Queue 软件包列表：

在 Solaris 上：

```
pkginfo | grep -i "message queue"
```

在 Linux 上：

```
rpm -qa | grep mq
```

6. 使用以下命令删除 Message Queue 软件包：

在 Solaris 上：

```
pkgrm packageName
```

其中，*packageName* 表示任意的 Message Queue 软件包。要删除多个软件包，请用空格分隔各软件包名称。

在 Linux 上：

```
rpm -e --nodeps RPMName
```

其中，*RPMName* 表示任意的 Message Queue rpm 组件。要删除多个组件，请用空格分隔各 RPM 名称。

由于其他产品可能要使用 Message Queue 软件包，因此删除它们时要慎重。pkgrm 命令会在删除软件包之前警告您任何依赖该软件包的产品。收到提示时，键入 **y**（是）确认删除请求。

7. 键入 "q" 退出。
8. 退出 root shell。
9. 重新安装发行版本 4 的 Message Queue。
使用 Java ES 发行版本 4 安装程序。
10. 恢复在[第 169 页](#)中的“[备份 Message Queue](#)”中备份的发行版本 4 Message Queue 数据。

发行版本 4 Message Queue 将使用在升级到发行版本 5 之前所备份的数据正常工作。

多实例升级

要升级一个 Message Queue 群集，且多个代理在此群集中进行交互以提供可伸缩的消息服务，您可以进行滚动升级，从而在每个 Message Queue 实例从发行版本 4 升级到发行版本 5 时，该群集仍保持在线。执行群集升级时，务必牢记以下两点：

- 因升级而关闭代理时，其存储的持久性消息要等到重新启动该代理后才可用。
- 主代理应在最后升级。

若不然，可直接执行以下过程：每次关闭、升级并重新启动一个代理，直到全部代理完成升级。

从 Java ES 发行版本 3 升级 Message Queue

将 Java ES 2005Q1（发行版本 3）Message Queue 升级到发行版本 5 的过程与将发行版本 4 Message Queue 升级到发行版本 5 的过程相同。

要将发行版本 3 Message Queue 升级到发行版本 5，请按照第 167 页中的“从 Java ES 发行版本 4 升级 Message Queue”中的说明操作，但需要在提到发行版本 4 的地方用发行版本 3 代替。

从 Java ES 发行版本 2 升级 Message Queue

本节包含关于将 Message Queue 从 Java ES 2004Q2（发行版本 2）升级到 Java ES 发行版本 5 的信息。其中涵盖以下主题：

- [简介](#)
- [发行版本 2 兼容性问题](#)
- [发行版本 2 Message Queue 的升级](#)
- [多实例升级](#)

注 如果要在 Linux 平台上从发行版本 2 Message Queue 升级，则必须执行双重升级，即 Message Queue 和操作系统都要升级（RHEL 2.1 上不支持发行版本 5 Message Queue）。有关更多信息，参见[第 166 页](#)中的“双重升级”。

简介

将 Java ES 发行版本 2 Message Queue 升级到发行版本 5 时，请考虑升级过程中的以下方面：

- **常规升级方法。** 使用 mqupgrade 脚本执行升级，该脚本会用新软件包替代先前的软件包并从发行版本 2 自动迁移配置数据。
- **升级依赖性。** 将一台计算机上的任一 Java ES 组件从发行版本 2 升级都需要升级由该计算机托管的所有其他 Java ES 组件；不支持有选择地将 Java ES 组件从发行版本 2 升级到发行版本 5。特别是，需要升级 Message Queue 所使用的所有 Java ES 共享组件。

此外，发行版本 5 Message Queue 对 Directory Server 和 Web Server（或 Application Server）具有可选依赖性，如[第 165 页](#)中的“[Message Queue 组件依赖性](#)”中所述。如果这些组件安装在同一台计算机上，则要求将它们也升级到发行版本 5。

- **向后兼容性。** 发行版本 5 Message Queue 与发行版本 2 不完全兼容，如下文的“[发行版本 2 兼容性问题](#)”中所述。
- **升级回滚。** 当前不支持从发行版本 5 到发行版本 2 的升级回滚（参见“[升级回滚](#)”）。

- **平台问题。** 在 Solaris 和 Linux 操作系统上，升级 Message Queue 的常规方法是相同的，但在 Linux 上还需要一些附加过程。以下过程将指出特定于平台的命令、文件位置或过程（如适用）。

发行版本 2 兼容性问题

相对于发行版本 2 及早期版本，发行版本 5 Message Queue 引入了以下一些常见的 Message Queue 兼容性问题。

协议兼容性

Message Queue 依赖 Web 容器在 Message Queue 客户机和代理之间提供 HTTP 协议支持。由于协议变化，使用 Sun Java System Web Server 为 Message Queue `imqhttp.war` 应用程序提供 Web 容器时，若不同时升级 Message Queue，将无法升级 Web Server 组件（参见第 170 页和第 181 页上的“升级后的任务”）。

代理兼容性

发行版本 5 Message Queue 代理将与发行版本 4、发行版本 3 或发行版本 2 的代理交互操作，但代理属性和持久存储库模式相对于发行版本 2 的更改会影响兼容性。

发行版本 5 Message Queue 可以使用发行版本 4、发行版本 3 和发行版本 2 的数据，但对于 Linux 系统，必须首先将发行版本 2 数据迁移到发行版本 5。

更新到发行版本 5 的 Message Queue 时，请考虑以下事项：

- 可以使用早期的 Message Queue `config.properties` 文件。也可以在配置发行版本 5 的 Message Queue 代理时将它们复制到另一个位置查看其包含的属性设置。
- 首次启动代理时，会根据需要将所有持久 Message Queue 数据（消息、目标、持久订阅）自动转换成发行版本 5 的 Message Queue 数据。例如，必要时，任何现有目标都会自动转换成发行版本 5 的 Message Queue 目标，同时保持现有属性并使用新属性的默认值。
- 如果将 Message Queue 发行版本 2 代理和 Message Queue 发行版本 5 代理混装在一个群集中，主代理必须是 Message Queue 发行版本 2 代理（取较旧者），该群集将运行 Message Queue 发行版本 2 群集。

受管对象兼容性

发行版本 5 Message Queue 受管对象与发行版本 3 及发行版本 4 的受管对象完全相同。但是，有些发行版本 3 受管对象相对于早期版本已经进行了重命名或由于采用新属性而得到了增强。因此，从发行版本 2 Message Queue 升级到发行版本 5 时，应考虑以下事项：

- 可以使用在发行版本 2 中创建的相同对象存储和受管对象；但最好是将受管对象迁移到发行版本 5。执行更新操作时，管理控制台 (imqadmin) 和 ObjectManager 命令行实用程序 (imqobjmgr) 会将发行版本 2 的受管对象转换成发行版本 5 的受管对象。
- 发行版本 5 客户机运行时将会查找并实例化发行版本 2 受管对象，然后转换它们以供发行版本 5 客户机使用。不过，这不会转换驻留在所查找的对象存储中的发行版本 2 受管对象。
- 现有的发行版本 2 客户机（应用程序和 / 或组件）（即直接实例化受管对象而非查找受管对象的客户机）与发行版本 5 兼容。不过，如果它们要使用新的受管对象属性（有关受管对象属性的信息，参见 Message Queue 3.7 UR1 管理指南 (<http://docs.sun.com/doc/820-0843>) 的第 16 章），则需要重写。（使用发行版本 5 重新编译发行版本 2 客户机时，将显示在发行版本 5 中已重命名的那些 Message Queue 发行版本 2 属性。旧名称仍然有效。）
- 用于启动 Java 客户机以及使用命令行选项设置受管对象属性值的脚本与发行版本 5 兼容。不过，如果它们要使用新的受管对象属性（有关受管对象属性的信息，参见 Message Queue 3.7 UR1 管理指南 (<http://docs.sun.com/doc/820-0843>) 的第 16 章），则需要重写。

管理工具兼容性

由于在发行版本 3 中添加了新的命令和新的管理功能，因此，发行版本 5 管理工具（管理控制台和命令行实用程序）只支持发行版本 3、发行版本 4 和发行版本 5 代理。但是，所有的发行版本 2 命令和命令选项仍受支持。

客户机兼容性

发行版本 3 和发行版本 4 客户机与发行版本 5 Message Queue 完全兼容。不过，在从发行版本 2 升级到发行版本 5 时，您应考虑以下与 Java 客户机相关的兼容性问题：

- 发行版本 5 代理将支持发行版本 2 客户机（但没有新增的发行版本 5 功能）。
- 发行版本 5 Java 客户机能够连接到发行版本 2 代理（但没有新增的发行版本 5 功能）。
- 发行版本 5 代理以及具有 Enterprise Edition 许可或 Platform Edition 试运行许可的发行版本 2、发行版本 3 或发行版本 4 代理支持 C 客户机程序。

发行版本 2 Message Queue 的升级

本节介绍如何执行 Message Queue 从 Java ES 发行版本 2 到 Java ES 发行版本 5 的升级：

- [升级前的任务](#)
- [升级发行版本 2 Message Queue \(Solaris\)](#)
- [升级发行版本 2 Message Queue \(Linux\)](#)
- [安装兼容性软件包 \(Linux\)](#)
- [检验 Message Queue 升级](#)
- [升级后的任务](#)
- [升级回滚](#)

升级前的任务

升级 Message Queue 软件之前，应执行以下任务：

- [检验当前版本信息](#)
- [升级 Message Queue 依赖项](#)
- [备份 Message Queue](#)

检验当前版本信息

可通过使用 `-version` 选项启动 Message Queue 代理，确定系统上安装的 Message Queue 的版本和版 (edition)：

```
imqbrokerd -version
```

表 10-5 Message Queue 版本检验输出

Java ES 发行版本	Message Queue 版本号
发行版本 2	Sun Java (tm) System Message Queue 3 2004Q2 版本： 3.5
发行版本 3	Sun Java (tm) System Message Queue 3 2005Q1 版本： 3.6
发行版本 4	Sun Java (tm) System Message Queue 3 2005Q4 版本： 3.6 SP3
发行版本 5	Sun Java (tm) System Message Queue 3.7 版本： 3.7 UR1

升级 Message Queue 依赖项

通常建议将计算机系统（以及计算环境）中的所有 Java ES 组件都升级到 Java ES 发行版本 5。Message Queue 仅对少数几个共享组件具有硬性升级依赖性。

升级 Message Queue 依赖项时，应在升级 Message Queue 之前按以下顺序执行（跳过任何可能已升级的依赖项）。

1. **共享组件。** 第 59 页中的第 2 章“升级 Java ES 共享组件”提供了将 Java ES 共享组件升级到发行版本 5 的说明。
2. **Sun Cluster（软性升级依赖性）。** 第 69 页中的第 3 章“Sun Cluster 软件”提供了将 Sun Cluster 升级到发行版本 5 的说明。
3. **Directory Server（软性升级依赖性）。** 第 89 页中的第 5 章“Directory Server”提供了将 Directory Server 升级到发行版本 5 的说明。
4. **Java DB（软性升级依赖性）。** 在升级 Message Queue 时，您需要执行发行版本 5 Java DB 的全新安装。
5. **Web 容器软件（软性升级依赖性）。** 第 119 页中的第 7 章“Web Server”和第 183 页中的第 11 章“Application Server”分别提供了有关升级 Web Server 和 Application Server 的说明。

备份 Message Queue

最好在执行升级之前备份生产环境中的应用程序数据。对于 Solaris OS，动态数据存储在以下目录中：

`/var/imq/instances/instanceName。`

对于其他操作系统，参见 Message Queue 3.7 UR1 管理指南 (<http://docs.sun.com/doc/820-0843>)。

升级发行版本 2 Message Queue (Solaris)

将 Message Queue 软件升级到 Java ES 发行版本 5 需要使用 `mqupgrade` 脚本，该脚本用于安装发行版本 5 软件包。

升级过程由下列步骤组成：

1. 停止运行任何 Message Queue 客户机应用程序。
如果正在 Application Server 环境中使用 Message Queue，则还要关闭 Application Server。
2. 停止运行任何代理。将提示您输入管理员用户名和密码。
`imqcmd shutdown bkr [-b hostName:port]`

3. 如果不想保留动态数据、Message Queue 平面文件用户信息库以及与每个代理实例关联的 Message Queue 访问控制文件，请使用以下命令删除。

```
imgbrokerd -name instanceName -remove instance
```

否则，将会保留动态数据和配置信息供发行版本 5 的 Message Queue 使用。

4. 以 Root 身份登录。

```
su -
```

5. 将目录更改为 Java ES 发行版本 5 分发软件 Tools 目录的位置。

在 Solaris SPARC 上：

```
cd Solaris_sparc/Product/message_queue/Tools
```

在 Solaris x86 上：

```
cd Solaris_x86/Product/message_queue/Tools
```

6. 运行 mqupgrade 脚本。

- a. 启动脚本：

```
./mqupgrade
```

mqupgrade 脚本会列出已安装的 Message Queue 组件。

- b. 输入 y（是）以升级 Message Queue 组件。

mqupgrade 脚本会检测并列出已经安装的本地化文件。

如果不想升级 Message Queue 组件，请输入 n（否）。mqupgrade 脚本将退出，不升级 Message Queue 组件。

- c. 收到提示时，输入 y（是）以升级本地化文件。

mqupgrade 会发送输出到以下位置的日志文件：

```
/var/sadm/install/logs/Message_Queue_upgrade_'date'.log
```

升级发行版本 2 Message Queue (Linux)

由于 Java ES 发行版本 2 仅在 RHEL 2.1 上受支持，但 RHEL 2.1 不支持 Java ES 发行版本 5，使得在 Linux 平台上将发行版本 2 Message Queue 升级到发行版本 5 变得更复杂。因此，需要进行双重升级：即操作系统和 Message Queue 都必须升级。参见第 41 页中的“双重升级：Java ES 和操作系统软件”。

基本过程是首先升级 Linux OS，然后升级所有的 Message Queue 依赖项，再升级 Message Queue。

从发行版本 2 Message Queue 到发行版本 5 的升级包括数据迁移步骤（即将代理实例数据迁移到相应的发行版本 5 位置），而 Solaris 系统则无需该步骤。如果要在升级到发行版本 5 时保留发行版本 2 数据，则使用 Message Queue 提供的迁移工具 (mqmigrate) 来执行此迁移。

要从发行版本 2 升级到发行版本 5，可按照第 178 页中的“升级发行版本 2 Message Queue (Solaris)”中的说明操作，只不过要在运行 mqupgrade 脚本（第 180 页中的步骤 7）之前先运行 mqmigrate 脚本（第 180 页中的步骤 6），这一点在以下过程中有详细说明。

1. 停止运行任何 Message Queue 客户机应用程序。

2. 停止运行任何代理。将提示您输入管理员用户名和密码。

```
imqcmd shutdown bkr [-b hostName:port]
```

3. 如果不想保留动态数据、Message Queue 平面文件用户信息库以及与每个代理实例关联的 Message Queue 访问控制文件，请使用以下命令删除。

```
imqbrokerd -name instanceName -remove instance
```

否则，将会保留动态数据和配置信息供发行版本 5 的 Message Queue 使用。

4. 以 root 身份登录，或成为超级用户。

```
su -
```

5. 将目录更改为 Java ES 发行版本 5 分发软件 Tools 目录的位置。

```
cd Linux_x86/Product/message_queue/Tools
```

6. 使用以下命令迁移代理实例数据：

```
./mqmigrate
```

mqmigrate 脚本会将发行版本 2 代理实例配置数据移动到相应的发行版本 4 位置。

7. 运行 mqupgrade 脚本。

- a. 启动脚本：

```
./mqupgrade
```

mqupgrade 脚本会列出已安装的 Message Queue 组件。

- b. 输入 y（是）以升级 Message Queue 组件。

mqupgrade 脚本会检测并列出已经安装的本地化文件。

如果不想升级 Message Queue 组件，请输入 n（否）。mqupgrade 脚本将退出，不升级 Message Queue 组件。

- c. 收到提示时，输入 y（是）以升级本地化文件。

mqupgrade 会发送输出到以下位置的日志文件：

```
/var/sadm/install/logs/Message_Queue_upgrade_'date'.log
```

安装兼容性软件包 (Linux)

如果您拥有取决于发行版本 5 安装文件位置的脚本或您的发行版本 2 客户机应用程序包含这样的脚本，就需要安装 sun-mq-compat 软件包，其中包含从发行版本 2 文件位置到发行版本 5 文件位置的符号链接。

sun-mq-compat 软件包位于以下位置，即解压缩 Java ES 发行版本 5 分发软件的位置。

```
Linux_x86/Product/message_queue/Packages
```

执行以下步骤，安装 sun-mq-compat 软件包：

1. 以 root 身份登录，或成为超级用户。

```
su -
```

2. 从 Packages 目录输入以下命令：

```
rpm -ivh --nodeps sun-mq-compat-3.7-RelNo.i386.rpm
```

检验 Message Queue 升级

完成升级过程后，通过使用 -version 选项启动 Message Queue 代理，检验升级是否成功。

```
imqbrokerd -version
```

该命令会返回 Java ES 版本号和特定于 Message Queue 的版本号。

升级后的任务

如果当前是使用 HTTP 隧道 servlet 提供 HTTP 连接服务支持，则 Message Queue 从发行版本 2 升级到发行版本 5 时便已升级了该 servlet。这便要求您在 Message Queue 升级到发行版本 5 之后重新部署该 servlet。有关 HTTP 支持的更多信息，参见 Message Queue 3.7 UR1 管理指南 (<http://docs.sun.com/doc/820-0843>)。

此外，您必须使用管理控制台 (imqadmin) 和 / 或 ObjectManager 命令行实用程序 (imqobjmgr) 将发行版本 2 受管对象迁移到它们的发行版本 5 的版本，以执行更新操作。

升级回滚

当前不支持将 Message Queue 从发行版本 2 升级到发行版本 5。通常，该过程应与从发行版本 5 到发行版本 4 的回滚过程相似（参见第 171 页中的“升级回滚”）。但是，由于将 Message Queue 从发行版本 2 升级到发行版本 5 不会更新 Java ES 产品注册表，因此，Java ES 安装程序无法重新安装发行版本 2 Message Queue。

有关此问题的解决方法，请咨询 Sun 服务部门。

多实例升级

要升级一个 Message Queue 群集，且多个代理在此群集中进行交互以提供可伸缩的消息服务，您可以进行滚动升级，从而在每个 Message Queue 实例从发行版本 2 升级到发行版本 5 时，该群集仍保持在线。执行群集升级时，务必牢记以下两点：

- 因升级而关闭代理时，其存储的持久性消息要等到重新启动该代理后才可用。
- 主代理应在最后升级。

若不然，可直接执行以下过程：每次关闭、升级并重新启动一个代理，直到全部代理完成升级。

Application Server

本章介绍如何将 Application Server 升级到 Java ES 5（发行版本 5）：Sun Java System Application Server Enterprise Edition 8.2。

本章概要介绍了有关发行版本 5 所支持的不同升级方式的升级注意事项。其内容涵盖了在 Solaris 和 Linux 操作系统上的升级：

- 第 185 页中的“Application Server 升级概述”
- 第 191 页中的“从 Java ES 发行版本 4 升级 Application Server”
- 第 197 页中的“从 Java ES 发行版本 3 升级 Application Server”
- 第 198 页中的“从 Java ES 发行版本 2 升级 Application Server”
- 第 207 页中的“在 Solaris 10 多区域环境中升级 Solaris 捆绑的 Application Server”

注	本章中的文件位置是相对于若干目录路径指定的，分别称为 <i>AppServer8-base</i> 和 <i>AppServer8Config-base</i> (Application Server 8.x) 以及 <i>AppServer7-base</i> 和 <i>AppServer7Config-base</i> (Application Server 7.x)。在安装 Application Server 时，可能至少已将这些路径的一部分指定为安装目录或域目录。否则，Java ES 安装程序会分配一个默认值。下表显示了这些目录路径的默认值。
----------	---

表 11-1 Application Server 目录路径

路径名变量	Solaris OS	Linux OS
<i>AppServer8-base</i>	/opt/SUNWappserver/appserver	/opt/sun/appserver
<i>AppServer8Install-base</i>	/opt/SUNWappserver	/opt/sun/appserver
<i>AppServer8Config-base</i>	/var/opt/SUNWappserver	/var/opt/sun/appserver
<i>AppServer7-base</i>	/opt/SUNWappserver7	/opt/SUNWappserver7
<i>AppServer7Config-base</i>	/var/opt/SUNWappserver7	/var/opt/SUNWappserver7

注 发行版本 5 Application Server (8.x) 的默认的 Application Server 域名是 domain1。换言之，domain1 是本章所使用的 *domainName* 变量的默认值。

此外，目录管理服务 (Directory Administration Service, DAS) 的默认实例名是 **server**。通常情况下，J2EE 应用程序并非部署到 DAS 实例中，而是部署到其他独立的实例中。

有关本章所使用的 **asadmin** 命令的更多信息，参阅《Sun Java System Application Server Enterprise Edition 8.2 Reference Manual》(<http://docs.sun.com/doc/819-4736>)。

Application Server 升级概述

本节将在以下方面对 Application Server 进行介绍，这些方面会对升级到 Java ES 5（发行版本 5）产生影响：

- [关于 Java ES 发行版本 5 Application Server](#)
- [Application Server 升级指导说明](#)
- [Application Server 数据](#)
- [Application Server 升级策略](#)

关于 Java ES 发行版本 5 Application Server

相对于发行版本 4 而言，Java ES 发行版本 5 Application Server 只是一个次要发行版本，其中仅修复了一些选定的错误。发行版本 5 Application Server 在功能上与发行版本 4 相同。

Application Server 升级指导说明

将 Application Server 升级到 Java ES 发行版本 5，有两种适合的升级方式：

- [表 11-2](#) 显示了所支持的、将 Java ES Application Server 升级到 Java ES 发行版本 5 的升级方式。此表既适用于 Solaris 操作系统，也适用于 Linux 操作系统。
- [表 11-3](#) 显示了所支持的、将 Solaris 捆绑的 Application Server 升级到 Java ES 发行版本 5 的升级方式。Application Server Platform Edition 与 Solaris OS 软件捆绑在一起。将捆绑版本的 Application Server 升级到发行版本 5 Enterprise Edition 可使用 Java ES 安装程序来执行，如[表 11-3](#)所示。

表 11-2 升级到 Java ES 5（发行版本 5）：Application Server Enterprise Edition 8.2 的方式

Java ES 发行版本	Application Server 版本	常规方法	必需的重新配置
发行版本 4	Sun Java System Application Server Enterprise Edition 8.1 2005Q4	直接升级： 使用 Java ES 安装程序执行。 ¹	无
发行版本 3	Sun Java System Application Server Enterprise Edition 8.1 2005Q1	直接升级： 使用 Java ES 安装程序执行。	无
发行版本 2	Sun Java System Application Server 7.0 Update 3 (2004Q2) Platform Edition 和 Enterprise Edition	直接升级： 使用 Java ES 安装程序升级软件包，再使用 postInstall 脚本执行重新配置，然后使用 asupgrade 实用程序迁移域信息。 ²	环境变量、域和其他配置数据。 需将 J2EE 组件和应用程序迁移到新的 Application Server 环境并重新部署。
发行版本 1	Sun ONE Application Server 7.0 Update 1 (2003Q4) Platform Edition 和 Enterprise Edition	直接升级未经验证： 使用发行版本 2 Application Server 的升级方法。	环境变量、域和其他配置数据。 需将 J2EE 组件和应用程序迁移到新的 Application Server 环境并重新部署。
Java ES 以前的发行版本		不可直接升级： 但可以按照 Java Enterprise System 2005Q1 升级与迁移指南 (http://docs.sun.com/doc/819-2237) 中的过程首先升级到发行版本 3。然后从发行版本 3 升级到发行版本 5。	

1. 如果要将 Application Server 从发行版本 4 升级到发行版本 5，但不保留配置信息或域信息，可使用 Java ES 发行版本 4 卸载程序卸载发行版本 4 Application Server，然后使用 Java ES 发行版本 5 安装程序执行发行版本 5 Application Server 的全新安装。但是，如果安装发行版本 4 Application Server 时使用了“以后再配置”选项，则必须在卸载发行版本 4 Application Server 之前，先创建一个 \$HOME/.asadminprefs 文件（其中，\$HOME 是安装并运行 Application Server 的用户的主目录）。该文件包括以下两行：
AS_ADMIN_PASSWORD=password
AS_ADMIN_USER=admin

2. 如果要在 Solaris 平台上从发行版本 2 Application Server 进行升级，必须要特别当心，因为 Solaris 捆绑的 Application Server 和 Java ES 发行版本 2 Application Server 会并存于您的计算机上。参见表 11-3 及其后的注意事项。

表 11-3 Solaris OS 捆绑的 Application Server 版本的升级方式

Solaris OS 版本	Application Server 版本	常规方法	必需的重新配置
Solaris 10	Sun Java System Application Server Platform Edition 8.0.0_01	直接升级： 使用 Java ES 安装程序执行。	无。
Solaris 9	Sun Java System Application Server 7.0.0_03c	直接升级： 如果 Application Server 已使用并创建了域，则需要迁移域信息。采用从发行版本 2 Application Server 进行升级时所介绍的方法，在此过程中，使用 Java ES 安装程序的“以后再配置”选项、postInstall 脚本以及 asupgrade 实用程序。 如果 Application Server 尚未使用，或者虽已使用但未创建域，也可采用从发行版本 2 Application Server 进行升级时所介绍的方法，不过，使用的是 Java ES 安装程序的“现在配置”选项，且不使用 postInstall 脚本和 asupgrade 实用程序。	环境变量、域（如果已创建）及其他配置数据。 需将 J2EE 组件和应用程序迁移到新的 Application Server 环境并重新部署。

注	如果要在 Solaris 平台上从发行版本 2 Application Server 进行升级，必须要特别当心，因为捆绑版本的 Application Server 和 Java ES 版本的 Application Server 会并存于您的计算机上。因此，必须先卸载捆绑版本（以及所有对应的域），然后再继续从发行版本 2 的版本进行升级。参见第 200 页中的“仅限 Solaris OS：手动删除操作系统捆绑的 Application Server 软件包”。
---	---

Application Server 数据

下表显示了可能受 Application Server 软件升级影响的数据类型。

表 11-4 Application Server 数据用途

数据类型	位置	用途
环境变量	<i>AppServer8-base/config/asenv.conf</i>	全局变量
配置数据	发行版本 3、4 和 5: domain.xml 和 server.policy 文件，位于 <i>AppServer8Config-base/domains/domainName/config</i> 发行版本 2: server.xml 和 server.policy 文件，位于 <i>AppServer7Config-base/domains/domainName/instanceName/config</i>	Application Server 实例的配置
部署数据	发行版本 3、4 和 5: <i>AppServer8Config-base/domains/domainName/applications</i> 发行版本 2: <i>AppServer7Config-base/domains/domainName/instanceName/applications</i>	特定 J2EE 组件和应用程序的 J2EE 容器的配置
访问日志文件	发行版本 3、4 和 5: <i>AppServer8Config-base/domains/domainName/logs/access/</i> 包含两个文件: server_acces_log 和 _asadmin_access_log 发行版本 2: <i>AppServer7Config-base/domains/domainName/instanceName/logs/access</i>	访问日志记录

Application Server 升级策略

Application Server 的升级策略通常取决于在第 1 章“升级规划”中讨论的许多注意事项：升级方式、Java ES 组件之间的依赖性、选择性升级与全部升级、多实例部署，等等。

本节通过提出可能影响 Application Server 升级计划的问题，对 Application Server 的一般注意事项进行逐一介绍。

兼容性问题

相对于发行版本 4 或发行版本 3，发行版本 5 Application Server 并未引入任何接口更改。但是，发行版本 5 和发行版本 2 之间却存在着很大的接口更改，这使得发行版本 5 与发行版本 2 不兼容。

但发行版本 5 Application Server 不支持发行版本 4 Service Registry。如果已将 Application Server 升级到发行版本 5，则必须将 Service Registry 也升级到发行版本 5。

Application Server 组件依赖性

Application Server 对其他 Java ES 组件的依赖性可能影响升级和重新配置 Application Server 软件的过程。例如 Application Server 接口或功能上的变化可能需要 Application Server 所依赖的组件的升级版本。是否需要升级此类组件取决于具体升级方式。

Application Server 对以下 Java ES 组件具有依赖性：

- **共享组件。** Application Server 对特定的 Java ES 共享组件具有依赖性（参见第 43 页中的表 1-9）。
- **Message Queue。** Application Server 依赖 Message Queue 提供符合 J2EE Java 消息服务标准的异步消息传送支持。
- **High Availability Session Store。** Application Server 依赖 High Availability Session Store (HADB) 来维护用以支持实例间故障转移的会话状态信息。

注 如果您拥有的是早期安装的、不需要 HADB 的 Application Server，则必须先安装 HADB，然后才能将 Application Server 升级到发行版本 5。例如，必须先安装发行版本 5 HADB，才能升级 Solaris OS 捆绑的 Application Server 版本。在这些情况下，Java ES 安装程序将自动执行 HADB 的安装。

- **Java DB。** Application Server 依赖 Java DB 作为默认的开发者数据库以及存储样例应用程序数据和 Enterprise Java Bean 计时器所需的数据。
- **Web 容器（可选依赖项）。** Application Server 因其可选的负载平衡插件而依赖于 Web 容器服务。该项支持可由 Java ES Web Server 提供，也可由第三方 Web 容器（如 Apache Web Server 和 Microsoft IIS）提供。

双重升级

在双重升级中，Application Server 和操作系统都要进行升级（如第 41 页中的“[双重升级：Java ES 和操作系统软件](#)”中所述）。可使用以下两种方式之一执行双重升级：

- 操作系统全新安装
- 操作系统就地升级

操作系统全新安装

1. 备份现有 Application Server 数据。

有关基本数据的位置，参见第 188 页中的“[Application Server 数据](#)”。

2. 安装新的操作系统。

只能将操作系统安装在同一台计算机上，并将清除现有文件系统。

3. 恢复在[步骤 1](#)中备份的 Application Server 数据。

4. 安装发行版本 5 Application Server。

使用本章中相关升级小节中所述的过程，这要视在[步骤 1](#)中备份的 Application Server 数据的版本而定。

操作系统就地升级

1. 备份现有 Application Server 数据。

有关基本数据的位置，参见第 188 页中的“[Application Server 数据](#)”。

2. 升级操作系统。

升级会将现有文件系统留在原来的位置。

3. 将 Application Server 升级到发行版本 5。

参见本章中的相关小节，这要视升级方式而定。

从 Java ES 发行版本 4 升级 Application Server

本节包含关于将 Application Server 从 Java ES 2005Q4（发行版本 4）升级到 Java ES 5（发行版本 5）的信息。其中涵盖以下主题：

- [简介](#)
- [发行版本 4 Application Server 的升级](#)

简介

将 Java ES 发行版本 4 Application Server 升级到发行版本 5 时，请考虑升级过程中的以下方面：

- **常规升级方法。** 使用 Java ES 安装程序执行升级。从发行版本 4 Application Server 升级到发行版本 5 时，不需要重新配置 Application Server，也不需要重新配置和迁移 J2EE 组件。
- **升级依赖性。** Application Server 对许多 Java ES 共享组件都具有依赖性（参见 [第 43 页中的表 1-9](#)），升级 Application Server 时，Java ES 安装程序会自动将所有这些组件升级到发行版本 5。Application Server 仅对 NSS 共享组件具有硬性升级依赖性。

此外，如 [第 189 页中的“Application Server 组件依赖性”](#) 中所述，发行版本 5 Application Server 还依赖于 Message Queue、HADB 和 Java DB。这些是硬性升级依赖性：必须将它们全部升级到发行版本 5。

Application Server 还对 Java ES Web Server 或第三方 Web 容器具有可选依赖性。但它们是软性升级依赖性，对于 Application Server 到发行版本 5 的升级，这些 Web 容器的升级是可选的。

- **向后兼容性。** 发行版本 5 的 Application Server 与发行版本 4 的版本向后兼容。
- **升级回滚。** 到发行版本 5 的升级无法回滚到发行版本 4。
- **平台问题。** 在 Solaris 和 Linux 操作系统上，升级 Application Server 的常规方法相同。

发行版本 4 Application Server 的升级

本节介绍如何在 Solaris 和 Linux 平台上将 Application Server 从 Java ES 发行版本 4 升级到 Java ES 发行版本 5。如果其中某个主题取决于具体平台的过程，则该主题将指明所适用的操作系统。本节包括以下主题：

- [升级前的任务](#)
- [升级发行版本 4 Application Server](#)
- [检验升级](#)
- [升级后的任务](#)
- [升级回滚](#)

升级前的任务

升级 Application Server 软件之前，应执行以下任务：

- [检验当前版本信息](#)
- [升级 Application Server 依赖项](#)
- [备份 Application Server 数据](#)
- [获取所需的配置信息和密码](#)

检验当前版本信息

可通过输入以下命令检验 Application Server 的当前版本：

```
AppServer8-base/bin/asadmin version --verbose
```

表 11-5 Application Server 版本检验输出

Java ES 发行版本	Application Server 版本号
发行版本 2	Sun ONE Application Server 7.0.0_03c
发行版本 3	Sun Java Enterprise System Application Server Enterprise Edition 8.1
发行版本 4	Sun Java Enterprise System Application Server Enterprise Edition 8.1_02
发行版本 5	Sun Java Enterprise System Application Server Enterprise Edition 8.2

升级 Application Server 依赖项

通常建议将计算机系统（以及计算环境）中的所有 Java ES 组件都升级到 Java ES 发行版本 5。Application Server 对 NSS 共享组件以及 Message Queue、HADB 和 Java DB 产品组件具有硬性升级依赖性。

升级 Application Server 依赖项时，应在升级 Application Server 之前，按如下顺序执行（跳过任何可能已经升级的依赖项）。但是，共享组件以及 Message Queue、HADB 和 Java DB 的升级通常是由 Java ES 安装程序在升级 Application Server 时自动实现的。

1. **共享组件。** 第 59 页中的“升级 Java ES 共享组件”中提供了将 Java ES 共享组件同步到发行版本 5 的说明。不过，在将 Application Server 升级到发行版本 5 时，Java ES 安装程序会自动升级 Application Server 所需的所有共享组件。
2. **Message Queue。** 第 161 页中的第 10 章“Message Queue”中提供了将 Message Queue 升级到发行版本 5 的说明。
3. **High Availability Session Store (HADB)。** 第 151 页中的第 9 章“High Availability Session Store”中提供了升级 HADB 的说明。
4. **Java DB。** 第 141 页中的第 8 章“Java DB”中提供了升级 Java DB 的说明。
5. **Web 容器软件（软性升级依赖性）。** 第 119 页中的第 7 章“Web Server”中提供了升级 Web Server 的说明。

备份 Application Server 数据

将 Application Server 从发行版本 4 升级到发行版本 5 并不修改配置数据，因此没有必要备份当前数据。

获取所需的配置信息和密码

您应当知道当前所安装版本的 Application Server 管理员用户 ID 和密码。

升级发行版本 4 Application Server

本节讨论影响 Application Server 升级过程的一些注意事项，随后对升级过程本身加以说明。

升级注意事项

将 Application Server 软件升级到 Java ES 发行版本 5 时，应考虑以下注意事项：

- 升级 Application Server 实例之前，应先关闭该实例中正在运行的所有 J2EE 组件。但是，如果负载平衡提供了高可用性或可伸缩性，则可以放宽此项要求。
- 在升级安装的 Application Server 映像期间，必须关闭在单台计算机上运行的所有 Application Server 实例（所有实例均对应于同一已安装的 Application Server 映像）。
- 在多节点部署中，需要对托管 Application Server 实例的每个节点或计算机执行升级过程。

升级过程

下述过程适用于本地驻留在即将进行升级的计算机上的 Application Server 实例。

1. 以 root 身份登录，或成为超级用户。

```
su -
```

2. 关闭要升级的 Application Server 实例中正在运行的所有 J2EE 组件。
3. 关闭要升级的计算机上的所有 Application Server 实例。

- a. 停止所有正在运行的节点代理。

```
AppServer8-base/bin/asadmin stop-node-agent --user admin_ID  
nodeagentName
```

其中，*nodeagentName* 的格式为 *hostName_domainName*，但默认情况下只是 *hostName*。

- b. 停止 Domain Administration Server (DAS)。

```
AppServer8-base/bin/asadmin stop-domain --user admin_ID  
domainName
```

- c. 停止 PointBase 数据库服务器（如果正在使用）。

```
AppServer8Config-base/appserver/pointbase/tools/stopserver.sh
```

4. 启动 Java ES 安装程序。

```
cd Java ES Release 5 distribution/os_arch
./installer
```

其中，*os_arch* 与您的平台匹配，如 *Solaris_sparc*。（对于命令行界面，使用 *installer -nodisplay* 选项。）

显示“欢迎”和“许可协议”页面之后，将向您显示一个组件选择页面。（如果检测到可由 Java ES 安装程序直接升级的已安装组件，它们的状态会显示为“可升级”。）

5. 在组件选择页面中选择 Application Server。

由于 Application Server 对 Message Queue、HADB 和 Java DB 具有硬性升级依赖性，因此还将自动选中这三个组件以进行升级。

6. 选择“现在配置”。

7. 指定所请求的配置值。

系统将显示许多配置面板。

8. 确认您的升级选择。

将升级 Application Server 软件包（必要时，还会升级 Message Queue、HADB 和 Java DB 的软件包）并显示升级摘要。

9. 退出 Java ES 安装程序。

10. 重新启动已升级的 Domain Administration Server (DAS)。

```
AppServer8-base/bin/asadmin start-domain --user admin_ID
domainName
```

11. 重新启动已升级的 Application Server 实例。

```
AppServer8-base/bin/asadmin start-node-agent --user admin_ID
nodeagentName
```

其中，*nodeagentName* 的格式为 *hostName_domainName*，但默认情况下只是 *hostName*。

检验升级

可使用以下命令检验升级是否成功：

```
AppServer8-base/bin/asadmin version --verbose
```

参见 第 192 页中的表 11-5 确认输出值。

升级后的任务

如果要继续使用嵌入式发行版本 4 PointBase 数据库而不使用 Java DB（一个新增的发行版本 5 Java ES 产品组件），则必须手动编辑升级后的

AppServer8-base/config/asenv.conf 文件。

从发行版本 4 Application Server 升级之后，*asenv.conf* 文件中的 PointBase 设置如下所示：

```
AS_POINTBASE="%POINTBASE_HOME%"  
AS_POINTBASE_SAMPLESDB="%POINTBASE_SAMPLESDB%"
```

将这些设置更改为以下值：

```
AS_POINTBASE="AppServer8-base/pointbase"  
AS_POINTBASE_SAMPLESDB="AppServerConfig8-base/var/appserver/pointbase"
```

升级回滚

不支持发行版本 5 的升级回滚。

从 Java ES 发行版本 3 升级 Application Server

将 Java ES 2005Q1（发行版本 3）Application Server 升级到发行版本 5 的过程与将发行版本 4 Application Server 升级到发行版本 5 的过程相同。

要将发行版本 3 Application Server 升级到发行版本 5，请按照第 191 页中的“从 [Java ES 发行版本 4 升级 Application Server](#)”中的说明操作，但需要在提到发行版本 4 的地方用发行版本 3 代替。

从 Java ES 发行版本 2 升级 Application Server

本节包含关于将 Application Server 从 Java ES 发行版本 2 升级到 Java ES 5（发行版本 5）的信息。其中涵盖了以下主题：

- [简介](#)
- [发行版本 2 Application Server 的升级](#)

注 如果要在 Linux 平台上从发行版本 2 Application Server 升级，则必须执行双重升级，即 Application Server 和操作系统都要升级（RHEL 2.1 上不支持发行版本 5 Application Server）。有关更多信息，参见第 190 页中的“双重升级”。

简介

将 Java ES 发行版本 2 Application Server 升级到发行版本 5 时，请考虑升级过程中的以下方面：

- **常规升级方法。** 通过使用 Java ES 安装程序并选择“以后再配置”选项安装发行版本 5 Application Server 来执行升级。随后使用 `asupgrade` 实用程序实现重新配置。在完成 Application Server 的升级之后，必须将发行版本 2 的 J2EE 组件和应用程序迁移到发行版本 5。
- **升级依赖性。** 将一台计算机上的任一 Java ES 组件从发行版本 2 升级都需要升级由该计算机托管的所有其他 Java ES 组件；不支持有选择地将 Java ES 组件从发行版本 2 升级到发行版本 5。

具体而言，必须升级 Application Server 所需的所有 Java ES 共享组件。对于 Message Queue，如果驻留在同一台计算机上，则必须将其也升级。另外，必须安装 High Availability Session Store (HADB) 和 Java DB，如果要使用 Web Server 实现负载平衡，必须将其也升级。

- **向后兼容性。** 发行版本 5 的 Application Server 与发行版本 2 版本不能向后兼容。需迁移 J2EE 组件和应用程序以在发行版本 5 的 Application Server 环境中运行。
- **升级回滚。** 只需恢复到发行版本 2 的安装即可实现从发行版本 5 到发行版本 2 的升级回滚（升级过程不会删除发行版本 2 的配置数据）。

- **平台问题。** 不能将在 Solaris OS 上升级 Application Server 的方法直接应用到 Linux OS: Java ES 发行版本 2 仅在 Linux2.1 上受支持, 而 Linux2.1 不支持 Java ES 发行版本 5。因此, 需要进行双重升级: 操作系统和 Application Server 都必须升级 (参见第 41 页中的“[双重升级: Java ES 和操作系统软件](#)”)。
- **数据位置。** 在发行版本 5 中, 配置数据、部署数据及访问日志文件位于新的位置中, 如第 188 页中的表 11-4 中所指定的那样。

发行版本 2 Application Server 的升级

本节介绍如何在 Solaris 和 Linux 平台上将 Application Server 从 Java ES 发行版本 2 升级到 Java ES 发行版本 5。如果其中某个主题取决于具体平台的过程, 则该主题将指明所适用的操作系统。本节包括以下主题:

- [升级前的任务](#)
- [升级发行版本 2 Application Server](#)
- [检验升级](#)
- [升级后的任务](#)
- [升级回滚](#)

升级前的任务

升级 Application Server 之前, 应执行以下任务:

- [仅限 Solaris OS: 手动删除操作系统捆绑的 Application Server 软件包](#)
- [检验当前版本信息](#)
- [升级 Application Server 依赖项](#)
- [备份 Application Server 数据](#)
- [获取所需的配置信息和密码](#)

仅限 Solaris OS: 手动删除操作系统捆绑的 Application Server 软件包

安装发行版本 2 Application Server 并不会删除 Solaris OS 捆绑的 Application Server 版本。要将发行版本 2 Application Server 正确升级到发行版本 5，必须执行以下步骤。

1. 通过手动删除位于 /usr/appserver 目录中的相应 Application Server 软件包，卸载捆绑版本：

```
pkgrm SUNWascmnse SUNWaslb SUNWasut ...
```

其中，可使用以下命令获得全套软件包：

```
pkginfo -i|grep -i "application server"
```

结果将包括如下一些软件包：

SUNWasacee、SUNWascmnse、SUNWasu、SUNWasuee、SUNWasac、SUNWascmn、SUNWasdb、SUNWasdem、SUNWasdemdb、SUNWasr、SUNWasut、SUNWasman、SUNWasjdoc

还可能会包括以下本地化软件包：

SUNWLocaleasacee、SUNWLocaleascmnse、SUNWLocaleasu、SUNWLocaleasuee

2. 如果已经使用了捆绑版本，则删除域信息：

```
rm /usr/appserver/domains
```

检验当前版本信息

可通过输入以下命令检验 Application Server 的当前版本：

```
AppServer7-base/bin/asadmin version --verbose
```

参见 第 192 页中的表 11-5 确认版本输出信息。

升级 Application Server 依赖项

升级 Application Server 依赖项需要将 Application Server 所依赖的所有本地驻留的发行版本 2 组件均升级到发行版本 5，具体而言就是 Message Queue 和共享组件。但共享组件会在升级过程中由 Java ES 安装程序自动升级（参见第 202 页中的“升级过程”）。

此外，升级 Application Server 还需要安装发行版本 5 Application Server 所依赖的 HADB 和 Java DB。但这两个产品组件会在升级过程中由 Java ES 安装程序自动选中并安装（参见第 202 页中的“升级过程”）。

一般而言，升级 Application Server 依赖项时，应在升级 Application Server 之前，按如下顺序执行（跳过任何可能已经升级的依赖项）。但共享组件的升级以及 HADB 和 Java DB 的安装是由 Java ES 安装程序在升级 Application Server 时自动完成的。

1. **共享组件。** 第 59 页中的“升级 Java ES 共享组件”中提供了将 Java ES 共享组件同步到发行版本 5 的说明。不过，在将 Application Server 升级到发行版本 5 时，Java ES 安装程序会自动升级 Application Server 所需的所有共享组件。

注 将共享组件升级到发行版本 5 包括将 J2SE 升级到 JDK 版本 1.5，而发行版本 2 Application Server 不支持 JDK 版本 1.5。因此，在升级共享组件之前，应关闭发行版本 2 Application Server。这样，只要使用 Java ES 安装程序安装或升级任何 Java ES 产品组件（如 Directory Server、Message Queue、HADB、Java DB，等等），就会将共享组件升级到发行版本 5。

2. **Message Queue。** 参见第 174 页中的“从 Java ES 发行版本 2 升级 Message Queue”。从发行版本 2 升级 Message Queue 不能通过使用 Java ES 安装程序来实现。
3. **High Availability Session Store (HADB)。** 升级 Application Server 时，Java ES 安装程序会执行 HADB 的全新安装。
4. **Java DB。** 升级 Application Server 时，Java ES 安装程序会执行 Java DB 的全新安装。
5. **Web 容器软件（软性升级依赖性）。** 参见第 140 页中的“从 Java ES 发行版本 2 升级 Web Server”。

备份 Application Server 数据

将 Application Server 从发行版本 2 升级到发行版本 5 不会覆写发行版本 2 的配置数据。但是，为了安全起见，应该先备份整个 domains 目录，然后再执行到发行版本 5 的升级：

```
AppServer7Config-base/domains
```

获取所需的配置信息和密码

您应知道有关当前所安装版本的以下信息：

- Application Server 管理员用户 ID 和密码
- 发行版本 2 的 Application Server 基本目录

升级发行版本 2 Application Server

本节讨论影响 Application Server 升级过程的一些注意事项，随后对升级过程本身加以说明。

升级注意事项

将 Application Server 软件升级到 Java ES 发行版本 5 时，应考虑以下注意事项：

- 升级 Application Server 实例之前，应先关闭该实例中正在运行的所有 J2EE 组件。但是，如果您使用负载均衡提供高可用性或可伸缩性，则可以放宽此项要求。
- 在升级安装的 Application Server 映像时，必须关闭在单台计算机上运行的所有 Application Server 实例（与同一已安装 Application Server 映像相对应的所有实例）。

升级过程

下述过程适用于本地驻留在即将进行升级的计算机上的所有 Application Server 实例。

1. 以 root 身份登录，或成为超级用户。

```
su -
```

2. 如果已将 Java ES 共享组件升级到了发行版本 5（特别是已将 J2SE 升级到了 JDK 1.5），而发行版本 2 Application Server 仍在运行，则编辑 asenv.conf 文件，使其直接指向 JDK 1.4。

```
AS_JAVA=/usr/java
```

如果 asenv.conf 引用的是 JDK 1.5，将无法关闭 Application Server（参见下面的步骤 3）。

3. 停止所有 Application Server 及相关进程。

```
AppServer7-base/bin/asadmin stop-appserv domainName
```

4. 安装发行版本 5 Application Server。

执行以下步骤：

- a. 在托管发行版本 2 Application Server 的计算机上启动 Java ES 安装程序。

```
cd Java ES Release 5 distribution/os_arch
./installer
```

其中，*os_arch* 与您的平台匹配，如 *Solaris_sparc*。（对于命令行界面，使用 `installer -nodisplay` 选项。）

显示“欢迎”和“许可协议”页面之后，将向您显示一个组件选择页面。（如果检测到可由 Java ES 安装程序直接升级的已安装组件，它们的状态会显示为“可升级”。）Application Server 不会显示为可从发行版本 2 升级。

- b. 从组件选择页面中选择 Application Server。

选择至少安装前三个子组件，包括 Node Agent 组件。

- c. 单击“下一步”。

如果 Message Queue 尚未升级到发行版本 5，将会显示一个错误消息窗口，要求您升级 Message Queue。在这种情况下，单击“确定”，然后升级 Message Queue。第 177 页中的“发行版本 2 Message Queue 的升级”介绍了该过程。

- d. 指定一个不同于发行版本 2 Application Server 的安装目录。
- e. 选择“以后再配置”选项。

注

如果您拥有的是 Solaris 9 OS 捆绑的 Application Server 版本，并且从未使用过此版本（未创建任何域信息），则可以选择“现在配置”选项并忽略步骤 5 及其后续步骤。

- f. 如果需要，选择安装本地化软件包的选项。
- g. 安装完成后退出 Java ES 安装程序。

5. 执行以下安装后过程：

- a. 在 Java ES 发行版本 5 分发软件的 Application Server Tools 目录中找到 `postInstall` 自述文件：

```
Java ES Release 5 distribution/os_arch/Product/application_svr/Tools
```

其中 *os_arch* 对应您的平台，例如 *Solaris_sparc*。

- b. 参阅 ReadMe 文件并运行 postInstall 脚本。

```
cd Java ES Release 5 distribution/os_arch/Product/application_svr/Tools
./postInstall AppServer8Install-base AppServer8Config-base
```

该脚本会通过安装期间所安装的模板，配置并创建 *AppServer8-base/bin/* shell* 脚本和一个 *config/asenv.conf* 文件。（通常情况下，Java ES 安装程序会创建 *bin/* shell* 脚本，但如果选择“以后再配置”，则必须按照所述方式创建它们。）

注 将 Application Server 升级到发行版本 5 时，会将非管理端口 8080（默认）和 8181（安全）分配给 Domain Administration Server (DAS)。这些端口分配可能会与分配给发行版本 2 Application Server 实例的端口相冲突。如果出现这种情况，则必须更改 DAS 端口，以免发生冲突。有关说明，参见《Sun Java System Application Server Enterprise Edition 8.2 管理指南》(<http://docs.sun.com/doc/820-0847>) 中关于如何修改 http 侦听器属性一节。

- c. 如有必要，在 *AppServer8-base/config/asenv.conf* 文件中修改环境设置。必须手动编辑该文件。

注 要配置 Application Server 以实现负载平衡，参阅《Sun Java System Application Server Enterprise Edition 8.2 高可用性管理指南》(<http://docs.sun.com/doc/820-0851>) “Application Server 高可用性功能”一章中的“配置 Web Server 以实现 HTTP 负载平衡”一节。

6. 运行 asupgrade 实用程序。

asupgrade 实用程序会创建一个发行版本 5 的节点代理，并将发行版本 2 的 Application Server 实例迁移到该节点代理下。

该实用程序位于 Application Server 目录下，例如：

- 升级向导模式：*AppServer8-base/bin/asupgrade*
- 升级控制台模式：*AppServer8-base/bin/asupgrade -c*

升级向导或升级控制台将指导您逐步升级。

a. 确定用于迁移域信息的源目录和目标目录：

- 发行版本 2: *AppServer7-base*
- 发行版本 5: *AppServerConfig8-base/domains*

b. 提供发行版本 2 Application Server 的管理员用户、管理员密码及主密码。

如果要求输入主密码，则指定 `changeit`；如果您当前使用的密钥库的主密码不是 `changeit`，应将密码更改为 `changeit`。

```
jdk-home/bin/keytool -storepasswd -new changeit -keystore keystore
-storepass oldpasswd
```

有关 Application Server `asupgrade` 实用程序的更多信息，参阅《Sun Java System Application Server Enterprise Edition 8.2 Reference Manual》(<http://docs.sun.com/doc/819-4736>) 的用户命令一节以及《Sun Java System Application Server Enterprise Edition 8.2 Upgrade and Migration Guide》(<http://docs.sun.com/doc/819-4737/6n6sao3ju?a=view>)。

7. 如果您在第 202 页中的步骤 2 中对 `asenv.conf` 文件进行了重定向，则将其恢复为指向 JDK 1.5。

```
AS_JAVA=/usr/jdk/entsys-j2se
```

如果 `asenv.conf` 引用的是 JDK 1.4，将无法启动 Application Server（参见下面的步骤 8 和步骤 9）。

8. 启动 Domain Administration Server (DAS)。

```
AppServer8-base/bin/asadmin start-domain --user admin_ID
domainName
```

9. 重新启动已升级的 Application Server 实例。

通过启动节点代理（升级后的 Application Server 实例已迁移到该节点代理下）来执行此操作：

```
AppServer8-base/bin/asadmin start-node-agent --user admin_ID
nodeagentName
```

其中，`nodeagentName` 的格式为 `hostName_domainName`，但默认情况下只是 `hostName`。

检验升级

启动管理控制台，检验是否启动了这些服务器。如果有任何服务器未运行，请检查以下日志文件，确定是否存在因端口冲突而引起的故障：

```
AppServer8Config-base/nodeagents/nodeagentName/instanceName/logs/server.log
```

如果存在因端口冲突引起的故障，请使用管理控制台修改端口号以消除冲突，然后停止并重新启动节点代理。

可通过输入以下命令检验 Application Server 到发行版本 5 的升级：

```
AppServer8-base/bin/asadmin version --verbose
```

参见 第 192 页中的表 11-5 确认输出值。

升级后的任务

除第 202 页中的“升级过程”中所述的步骤之外，还必须要执行以下两项升级后的任务：

- 更正对 JSS 的引用
- 迁移发行版本 2 J2EE 组件

更正对 JSS 的引用

要使 Application Server 引用正确版本的 JSS 共享组件，必须将 Application Server domain.xml 文件中的 jss3.jar 替换为 jss4.jar，该文件位于以下位置：

```
AppServer8Config-base/domains/domainName/config/domain.xml
```

迁移发行版本 2 J2EE 组件

必须迁移发行版本 2 J2EE 组件和应用程序，使其在发行版本 5 Application Server 环境中运行，并将它们重新部署到相应的 Application Server 实例中。有关迁移 J2EE 组件和应用程序的更多信息，参阅 Application Server Enterprise Edition 8.2 Upgrade and Migration Guide (<http://docs.sun.com/doc/819-4737>) 的第 4 章。

升级回滚

发行版本 5 的 Application Server 升级回滚过程仅仅是恢复到发行版本 2 的 Application Server 版本，在升级到发行版本 5 时并不会删除该发行版本。

在 Solaris 10 多区域环境中升级 Solaris 捆绑的 Application Server

Application Server（及其依赖的 Message Queue）与 Solaris 10 OS 捆绑在一起。只要不将该捆绑的 Application Server 从全局区域中删除，它就会在创建非全局区域时被传播到这些非全局区域当中。由于捆绑的 Application Server 存在于所有区域中，因此会影响随后 Application Server 到发行版本 5 的升级：

- 升级全局区域中的 Application Server 会删除全局区域中的捆绑版本，并自动删除所有非全局区域中的捆绑版本。
- 将全局区域中的 Application Server 升级到发行版本 5 不会升级非全局区域中的 Application Server，因为发行版本 5 Application Server 软件包不会被传播。

以下示例介绍了在 Solaris 10 多区域环境中升级 Application Server 时所涉及的一些技巧。（有关 Java ES 和 Solaris 10 区域的更详细论述，参见第 54 页中的“Java ES 5 升级与 Solaris 10 区域”。）

本示例的目标是将 Solaris 10 稀疏根区域中 Solaris 捆绑的 Application Server（版本 8.0.0_01）升级到发行版本 5。

您不能只是升级稀疏根区域中的 Application Server，因为 Solaris 捆绑的 Application Server 安装在从全局区域挂载的一个只读目录中。因此，要将稀疏根区域中的 Application Server 升级到发行版本 5，必须先将该捆绑版本从全局区域中删除。

此外，Message Queue 也安装在全局区域中，这违背了只能将共享组件（而非产品组件）安装在全局区域中的惯例。其原因是受只读目录的影响，不能在稀疏根区域中安装和升级 Message Queue。

将 Solaris 10 稀疏根区域中 Solaris 捆绑的 Application Server（版本 8.0.0_01）升级到发行版本 5 的过程如下所示：

1. 检验系统的初始状态。

本示例假定有一个已经由全局管理员配置、安装并引导了稀疏根区域的 Solaris 10 版本。

稀疏根区域中包括所有已安装在全局区域中的 Java ES 组件，即 Solaris 10 捆绑的 Message Queue 和 Application Server 版本。

此外，本示例还假定用户先前使用过稀疏根区域中的捆绑 Application Server，并创建了需要保留的管理域信息。

2. 升级全局区域中捆绑版本的 Application Server。

此操作会删除捆绑的 Application Server 软件包，并将它们替换为发行版本 5 软件包。对捆绑软件包的删除操作会传播到稀疏根区域中，从而能有效地卸载稀疏根区域中的 Application Server 软件包，但发行版本 5 软件包不会传播到非全局区域中。

- a. 在全局区域中运行 Java ES 安装程序。
- b. 在组件选择页面中选择 Application Server。

Message Queue、HADB 和 Java DB 将被自动选中，同时，Application Server 和 Message Queue 将被标记为可升级。

- c. 完成升级。

虽然会将发行版本 5 Message Queue 传播到稀疏根区域中，但不会传播 Application Server、HADB 和 Java DB。此外，所有共享组件都将同步到发行版本 5 并传播到稀疏根区域。

3. 在稀疏根区域中安装 Application Server。

- a. 在稀疏根区域中运行 Java ES 安装程序。
- b. 在组件选择页面中选择 Application Server。

如果 Message Queue 已被自动选中，则取消选择它；如果 HADB 和 Java DB 未自动选中，则选择它们。

- c. 完成 Application Server 的安装。

选择“以后再配置”，这样，域信息不会在安装过程中被覆写。

如果未创建任何域，则可以选择“现在配置”。

Service Registry

本章介绍如何将 Service Registry 升级到 Java ES 5（发行版本 5）：Service Registry 3.1。

本章概要介绍了有关发行版本 5 所支持的不同升级方式的升级注意事项。其内容涵盖了在 Solaris 和 Linux 操作系统上的升级：

- [第 210 页中的“Service Registry 升级概述”](#)
- [第 213 页中的“从 Java ES 发行版本 4 升级 Service Registry”](#)

注 本章中的文件位置是相对于称为 *ServiceRegistryR4-base* 和 *RegistryDomainR4-base*（Java ES 发行版本 4 Service Registry）以及 *ServiceRegistryR5-base* 和 *RegistryDomainR5-base*（Java ES 发行版本 5 Service Registry）的目录路径指定的。在安装 Service Registry 时，可能至少已将这些路径的一部分指定为安装目录。否则，Java ES 安装程序会分配一个默认值。

下表说明了这些目录路径的默认值。

表 12-1 Service Registry 目录路径

路径名变量	Solaris OS	Linux OS
<i>ServiceRegistryR4-base</i>	/opt/SUNWsoar	/opt/sun/SUNWsoar
<i>RegistryDomainR4-base</i>	/var/opt/SUNWsoar	/var/opt/sun/SUNWsoar
<i>ServiceRegistryR5-base</i>	/opt/SUNWsrcv-registry	/opt/sun/srcv-registry
<i>RegistryDomainR5-base</i>	/var/opt/SUNWsrcv-registry	/var/opt/sun/srcv-registry

Service Registry 升级概述

本节将在以下方面对 Service Registry 进行介绍，这些方面会对升级到 Java ES 5（发行版本 5）产生影响：

- [关于 Java ES 发行版本 5](#)
- [Java ES 发行版本 5 升级指导说明](#)
- [Service Registry 数据](#)
- [Service Registry 升级策略](#)

关于 Java ES 发行版本 5

相对于发行版本 4 Service Registry 而言，Java ES 发行版本 5 Service Registry 只是一个次要发行版本。它改进了一些功能，对接口进行了更新，还修复了选定的错误。

Java ES 发行版本 5 升级指导说明

[表 12-2](#) 显示了所支持的、将 Service Registry 升级到 Java ES 发行版本 5 的升级方式。此表既适用于 Solaris，也适用于 Linux 操作系统。

表 12-2 升级到 Java ES 5（发行版本 5）：Service Registry 3.1 的方式

Java ES 发行版本	Service Registry 版本	常规方法	必需的重新配置
发行版本 4	Sun Java System Service Registry 3.0 2005Q4	直接升级： 通过执行全新安装以及将注册表数据传送到发行版本 5 来替换发行版本 4。	无

Service Registry 数据

下表显示了可能受 Service Registry 软件升级影响的数据类型。

表 12-3 Service Registry 数据用途

数据类型	位置	用途
安装参数	<i>ServiceRegistryR4-base/install/install.properties</i>	Service Registry 的配置
受信证书	<i>ServiceRegistryR4-base/install/cacerts</i>	受 Service Registry 信任的证书，该证书并非是 Application Server 安装的一部分
配置数据	<i>RegistryDomainR4-base/domains/registry/applications/j2ee-modules/soar/WEB-INF/classes/*.properties</i>	Service Registry 实例的配置
注册表 / 系统信息库数据	<i>RegistryDomainR4-base/3.0/data</i>	数据库和用户证书
Web 接口配置	<i>RegistryDomainR4-base/3.0/jaxr-ebxml</i>	Web 接口的配置

Service Registry 升级策略

Service Registry 的升级策略通常取决于在第 1 章“升级规划”中讨论的许多注意事项：升级方式、Java ES 组件之间的依赖性、选择性升级与全部升级、多实例部署，等等。

本节通过提出可能影响 Service Registry 升级计划的问题，对 Service Registry 的一般注意事项进行逐一介绍。

兼容性问题

发行版本 5 Service Registry 与发行版本 4 Service Registry 向后兼容。

组件依赖性

Service Registry 对其他 Java ES 组件的依赖性会影响升级和重新配置 Service Registry 软件的过程。例如，Service Registry 接口或功能上的变化可能要求 Service Registry 所依赖组件的升级版本。是否需要升级此类组件取决于具体的升级方式。

Service Registry 对以下 Java ES 组件具有依赖性：

- **共享组件。** Service Registry 对特定的 Java ES 共享组件具有依赖性（参见第 43 页中的表 1-9）。
- **Application Server。** Service Registry 对 Application Server 具有强制依赖性，以便为 Service Registry 应用程序提供容器以及管理与联网注册表 / 系统信息库数据库之间的连接（对于 Java ES 发行版本 5）。
- **Java DB。** Service Registry 对 Java DB 具有强制依赖性，后者是用于存储服务以及描述这些服务的元数据的默认数据库。

双重升级

在双重升级中，Service Registry 和操作系统都要进行升级（如第 41 页中的“[双重升级：Java ES 和操作系统软件](#)”中所述）。可使用以下两种方式之一执行双重升级：

- 操作系统全新安装
- 操作系统就地升级

操作系统全新安装

1. 备份现有 Service Registry 数据。

有关基本数据的位置，参见第 211 页中的“[Service Registry 数据](#)”。

2. 安装新的操作系统。

可将操作系统安装在新系统上（或 Solaris 10 区域中），否则会擦除现有文件系统。

3. 恢复在[步骤 1](#)中备份的 Service Registry 数据。
4. 安装发行版本 5 Service Registry。

操作系统就地升级

1. 备份现有 Service Registry 数据。

有关基本数据的位置，参见第 211 页中的“[Service Registry 数据](#)”。

2. 升级操作系统。

升级会将现有文件系统留在原来的位置。

3. 将 Service Registry 升级到发行版本 5。

参见第 213 页中的“[从 Java ES 发行版本 4 升级 Service Registry](#)”。

从 Java ES 发行版本 4 升级 Service Registry

本节包含关于将 Service Registry 从 Java ES 2005Q4（发行版本 4）升级到 Java ES 5（发行版本 5）的信息。其中涵盖以下主题：

- [简介](#)
- [发行版本 4 Service Registry 的升级](#)
- [多实例升级](#)

简介

将 Java ES 发行版本 4 Service Registry 升级到发行版本 5 时，请考虑升级过程中的以下方面：

- **常规升级方法。** 要实现升级，先执行发行版本 5 Service Registry 的全新安装，将发行版本 4 数据和配置迁移到发行版本 5，然后删除发行版本 4 以节省磁盘空间。
- **升级依赖性。** Service Registry 对许多 Java ES 共享组件都具有依赖性（参见 [第 43 页中的表 1-9](#)），所有这些组件会在您升级 Service Registry 时，由 Java ES 安装程序自动将它们升级到发行版本 5。

Service Registry 对 Application Server 和 Java DB 具有硬性升级依赖性。

- **向后兼容性。** 发行版本 5 Service Registry 与发行版本 4 完全兼容。
- **升级回滚。** 发行版本 5 的升级回滚是通过恢复所保存的数据库和配置数据，然后复原到发行版本 4 实现的。
- **平台问题。** 在 Solaris 和 Linux 操作系统上，升级 Service Registry 的常规方法相同。

发行版本 4 Service Registry 的升级

本节介绍如何在 Solaris 和 Linux 平台上将 Service Registry 从 Java ES 发行版本 4 升级到 Java ES 发行版本 5。如果其中某个主题取决于具体平台的过程，则该主题将指明所适用的操作系统。本节包括以下主题：

- [升级前的任务](#)
- [升级发行版本 4 Service Registry](#)
- [检验升级](#)
- [升级后的任务](#)
- [升级回滚](#)

升级前的任务

升级 Service Registry 之前，应执行以下任务：

- [检验当前版本信息](#)
- [升级 Service Registry 依赖项](#)
- [备份 Service Registry 数据](#)
- [获取所需的配置信息和密码](#)

检验当前版本信息

可通过观察 Web Console 用户界面的特征来检验 Service Registry 的当前版本：

```
http://localhost:6060/soar
```

另外，也可检查 Service Registry 软件包名称。例如：

在 Solaris 上：

```
pkginfo -l|grep srvc
```

在 Linux 上：

```
rpm -qa|grep srvc
```

下表显示了区别特征和软件包名称：

表 12-4 Service Registry 版本检验输出

Java ES 发行版本	Service Registry 版本号	区别特征
发行版本 4	3.0	Web Console: 工具部分在左面板中 软件包名称包括字符串: soar
发行版本 5	3.1	Web Console: 左面板中有三个选项卡 软件包名称包括字符串: srvc-registry

升级 Service Registry 依赖项

通常建议将计算机系统（以及计算环境）中的所有 Java ES 组件都升级到 Java ES 发行版本 5。Service Registry 对许多共享组件、Application Server 以及 Java DB 具有硬性升级依赖性。

升级 Service Registry 的依赖项时，应在升级 Service Registry 之前按以下顺序执行（跳过任何可能已升级的依赖项）。共享组件的升级通常是由 Java ES 安装程序自动实现的。

- 1. **共享组件。** 第 59 页中的“升级 Java ES 共享组件”中提供了将 Java ES 共享组件同步到发行版本 5 的说明。不过，在将 Service Registry 升级到发行版本 5 时，Java ES 安装程序会自动升级 Service Registry 所需的所有共享组件。
- 2. **Java DB。** 第 141 页中的第 8 章“Java DB”中提供了将 Java DB 升级至发行版本 5 的说明。
- 3. **Application Server。** 第 183 页中的第 11 章“Application Server”中提供了将 Application Server 升级至发行版本 5 的说明。

修改 HTTP 端口号

编辑 *ServiceRegistryR4-base/install/install.properties* 文件，将 HTTP 端口从 6060 更改为 6480（6060 是保留端口）。有关设置此属性的信息，参见《Service Registry 3.1 管理指南》(<http://docs.sun.com/doc/820-0537>)。

备份 Service Registry 数据

将 Service Registry 从发行版本 4 升级到发行版本 5 不会修改配置数据，也不会修改注册表 / 系统信息库数据库。没有必要备份当前数据。

获取所需的配置信息和密码

需要知道发行版本 4 Service Registry 的用户 ID、密码、域名和端口号。

升级发行版本 4 Service Registry

本节介绍 Solaris 和 Linux 平台上的升级过程。

升级过程 (Solaris)

下述过程适用于本地驻留在即将进行升级的计算机上的 Service Registry 实例。

1. 以 root 身份登录，或成为超级用户。

```
su -
```

2. 确保 Jakarta ANT Java/XML-based build tool（ANT 共享组件）引用的是正确版本的 J2SE。

（在接下来的步骤中将使用 ant 命令。）

```
PATH=/usr/jdk/entsys-j2se/bin:$PATH
```

```
export PATH
```

3. 停止发行版本 4 Service Registry (Application Server) 域。

```
cd ServiceRegistryR4-base/install  
/usr/sfw/bin/ant -f build-install.xml appserver.domain.stop
```

该域与一个 Service Registry 实例相关联。

4. 执行发行版本 5 Service Registry 的全新安装。

执行以下步骤：

- a. 在托管发行版本 4 Service Registry 的计算机上启动 Java ES 安装程序。

```
cd Java ES Release 5 distribution/os_arch  
./installer
```

其中，*os_arch* 与您的平台匹配，如 *Solaris_sparc*。（对于命令行界面，使用 *installer -nodisplay* 选项。）

显示“欢迎”和“许可协议”页面之后，将向您显示一个组件选择页面。

（如果检测到可由 Java ES 安装程序直接升级的已安装组件，它们的状态会显示为“可升级”。）

- b. 从组件选择页面中选择 Service Registry。
- c. 指定一个不同于发行版本 4 的安装目录路径。

默认情况下，发行版本 5 安装路径 (*ServiceRegistryR5-base*) 不同于发行版本 4 安装路径 (*ServiceRegistryR4-base*)。

- d. 选择“以后再配置”选项。
不支持“现在配置”选项。
 - e. 如果需要，选择安装本地化软件包的选项。
 - f. 安装完成后退出 Java ES 安装程序。
5. 升级和配置发行版本 5 Service Registry 实例。

```
cd ServiceRegistryR5-base/install
/usr/sfw/bin/ant -f build-install.xml
-Dinstall.properties=ServiceRegistryR4-base/install/install.properties
upgrade
```

除了指向发行版本 4 `install.properties` 文件外，还可通过修改默认发行版本 5 `install.properties` 文件来再现任何发行版本 4 属性值。有关设置这些属性的信息，参见《Service Registry 3.1 管理指南》(<http://docs.sun.com/doc/820-0537>)。

如果要使用自定义属性值，但未将它们放入 `install.properties` 中，则需要在 `Ant` 命令行上指定此类属性值（所有值均位于一行），如下所示：

```
/usr/sfw/bin/ant -f build-install.xml
-Dregistry.install.RegistryServerKeystorePassword=password1
-Dregistry.install.AdministratorPassword=password2
-Dregistry.install.ApplicationServerKeystorePassword=password3
upgrade
```

但是，建议您将此类自定义属性值包括到具有受限权限的 `install.properties` 文件中，因为在使用命令行设置时可能会被未经授权人员看到。有关更多信息，参见 `Service Registry` 管理指南。

升级实用程序会创建一个新 `Application Server` 域，启动该域，并在该域中部署 `Service Registry` 实例。每个 `Service Registry` 实例均与其自身的 `Application Server` 域相关联。

6. 如果发行版本 4 `Service Registry` 的服务器属性文件已经修改，则可对发行版本 5 `Service Registry` 配置进行相应的更改，如下所示：

- a. 停止发行版本 5 `Service Registry` (`Application Server`) 域。

（该域已由步骤 5 中的 `upgrade` 命令自动启动。）

```
cd ServiceRegistryR5-base/install
/usr/sfw/bin/ant -f build-install.xml appserver.domain.stop
```

- b. 将发行版本 4 Service Registry 实例配置传送到发行版本 5。

将对发行版本 4 Service Registry 实例配置所做的任何修改：

```
RegistryDomainR4-base/domains/registry/applications/j2ee-modules/  
soar/WEB-INF/classes/*.properties
```

添加到相应的发行版本 5 配置中：

```
RegistryDomainR5-base/domains/registry/applications/j2ee-modules/  
soar/WEB-INF/classes/*.properties
```

7. 启动发行版本 5 Service Registry (Application Server) 域。

```
cd ServiceRegistryR5-base/install  
/usr/sfw/bin/ant -f build-install.xml appserver.domain.start
```

升级过程 (Linux)

在 Linux 上升级 Service Registry 与在 Solaris 上完全相同（参见第 216 页中的“[升级过程 \(Solaris\)](#)”），但对于在升级过程的各个步骤中所使用的 ant 命令，其在 Linux 平台上的位置与在 Solaris 平台上的位置不同：

```
/opt/sun/share/bin/ant
```

检验升级

可通过观察 Web Console 用户界面的特征来检验 Service Registry 升级是否成功：

```
http://localhost:6480/soar
```

另外，也可检查 Service Registry 软件包名称。例如：

在 Solaris 上：

```
pkginfo -l|grep soar
```

在 Linux 上：

```
rpm -qa|grep soar
```

区别特征和软件包名称显示在第 215 页中的表 12-4 中。

升级后的任务

以下步骤介绍如何删除发行版本 4 Service Registry，只有您确定不会对发行版本 4 的升级执行回滚操作，才能执行这些步骤。

1. 删除发行版本 4 Service Registry (Application Server) 域:

```
cd ServiceRegistryR4-base/install
```

在 Solaris 上:

```
/usr/sfw/bin/ant -f build-install.xml appserver.domain.delete
```

在 Linux 上:

```
/opt/sun/bin/ant -f build-install.xml appserver.domain.delete
```

2. 删除包含发行版本 4 Service Registry 域文件的目录。

```
rm -rf RegistryDomainR4-base
```

3. 删除包含发行版本 4 Service Registry 安装文件的目录。

```
rm -rf ServiceRegistryR4-base
```

升级回滚

发行版本 5 的升级回滚是通过复原到先前版本来实现的, 该版本在升级到发行版本 5 后仍保持原样。

1. 停止并删除发行版本 5 Service Registry (Application Server) 域:

```
cd ServiceRegistryR4-base/install
```

在 Solaris 上:

```
/usr/sfw/bin/ant -f build-install.xml appserver.domain.delete
```

在 Linux 上:

```
/opt/sun/bin/ant -f build-install.xml appserver.domain.delete
```

2. 运行 Java ES 发行版本 5 卸载程序以卸载发行版本 5 Service Registry。

3. 启动发行版本 4 Service Registry 域。

```
cd ServiceRegistryR4-base/install
```

在 Solaris 上:

```
/usr/sfw/bin/ant -f build-install.xml appserver.domain.start
```

在 Linux 上:

```
/opt/sun/bin/ant -f build-install.xml appserver.domain.start
```

4. 访问发行版本 4 Service Registry Web Console。

```
http://localhost:6480/soar
```

5. 确认该控制台显示的是发行版本 4 特征, 如第 215 页中的“Service Registry 版本检验输出”中所示。

多实例升级

在一些部署体系结构中，Service Registry 被部署在多个计算机系统上，以提供可伸缩性和提高可用性。例如，您可以在多台计算机上运行 Service Registry 实例，并使用一个负载均衡器来分配负载。

在这些体系结构中，注册表大多是只读的，并通过访问公共数据库响应大量查询负载。

如第 214 页中的“[发行版本 4 Service Registry 的升级](#)”中所述，在每台计算机上升级 Service Registry。

Web Proxy Server

本章介绍如何将 Web Proxy Server 升级到 Java ES 5（发行版本 5）：Sun Java System Web Proxy Server 4.0.4。

本章概要介绍了有关发行版本 5 所支持的不同升级方式的升级注意事项。其内容涵盖了在 Solaris 和 Linux 操作系统上的升级：

- 第 222 页中的“Web Proxy Server 升级概述”
- 第 225 页中的“从 Java ES 发行版本 4 升级 Web Proxy Server”
- 第 231 页中的“从版本 3.6 升级 Web Proxy Server”

注 本章中的文件位置是相对于一个称为 *WebProxyServer-base* 的目录路径而指定的。在最初安装 Web Proxy Server 时，可能至少已将此路径的一部分指定为安装目录。否则，Java ES 安装程序会分配一个默认值。

WebProxyServer-base 的默认值视操作系统平台而定，如下表所示。

表 13-1 Web Proxy Server 目录路径

路径名变量	Solaris OS	Linux OS
<i>WebProxyServer-base</i>	/opt/SUNWproxy	/opt/sun/webproxyserver

Web Proxy Server 升级概述

本节将在以下方面对 Web Proxy Server 进行介绍，这些方面会对升级到 Java ES 5（发行版本 5）产生影响：

- [关于 Java ES 发行版本 5 Web Proxy Server](#)
- [Web Proxy Server 升级指导说明](#)
- [Web Proxy Server 数据](#)
- [Web Proxy Server 升级策略](#)

关于 Java ES 发行版本 5 Web Proxy Server

相对于发行版本 4 而言，Java ES 发行版本 5 Web Proxy Server 只是一个次要发行版本，其中修复了一些错误。

但是，发行版本 5 Web Proxy Server 与其加入 Java Enterprise System 之前的 Sun One Web Proxy Server 3.6 相比，性能更佳、体系结构伸缩性更强、更符合标准，并具有新的管理界面。

Web Proxy Server 升级指导说明

表 13-2 显示了将 Web Proxy Server 升级到 Java ES 发行版本 5 的升级方式。先前的 Java ES 发行版本中不包括 Web Proxy Server。此表仅适用于 Solaris OS，因为在 Linux OS 上不支持 Web Proxy Server。

表 13-2 升级到 Java ES 5（发行版本 5）：Web Proxy Server 4.0.4 的方式

Java ES 发行版本	Web Proxy Server 版本	常规方法	必需的重新配置
发行版本 4	Sun Java System Web Proxy Server 4.0.1 2005Q4	直接升级： 使用修补程序执行	无
Java ES 以前的发行版本 （仅 Solaris OS）	Sun ONE Web Proxy Server3.6 （下文称为版本 3.6）	直接升级： 使用 Java ES 安装程序在新位置安装，然后使用管理工具迁移配置数据来执行	必须将配置信息迁移到新位置。

Web Proxy Server 数据

下表显示了可能受 Web Proxy Server 软件升级影响的数据类型。

表 13-3 Web Proxy Server 数据用途

数据类型	位置	用途
配置数据	<i>WebProxyServer-base/proxy-serverid/</i> config 目录 包含如 <i>server.xml</i> 、 <i>magnus.conf</i> 、 <i>obj.conf</i> 等文件	存储服务器、高速缓存、过滤器、路由及 Web Proxy Server 其他功能方面的配置信息

Web Proxy Server 升级策略

Web Proxy Server 的升级策略通常取决于在第 1 章“升级规划”中讨论的许多注意事项：升级方式、Java ES 组件之间的依赖性、选择性升级与全部升级、多实例部署，等等。

本节通过提出可能影响 Web Proxy Server 升级计划的问题，对 Web Proxy Server 的一般注意事项进行逐一介绍。

兼容性问题

发行版本 5 Web Proxy Server 未引入任何新的公共接口，并与发行版本 4 Web Proxy Server 向后兼容。发行版本 5 Web Proxy Server 还与版本 3.6 兼容，但使用版本 3.6 所支持的 NSAPI 接口所开发的插件除外，这些插件必须用发行版本 5 所支持的 NSAPI 接口重新编译。

Web Proxy Server 组件依赖性

Web Proxy Server 对以下 Java ES 组件具有依赖性：

- **共享组件。** Web Proxy Server 对特定的 Java ES 共享组件具有依赖性（参见第 43 页中的表 1-9）。
- **Directory Server。** Web Proxy Server 对用于提供基于 LDAP 验证的 Directory Server 具有可选依赖性。
- **Web Server。** Web Proxy Server 对用于增进 HTTP 请求安全性和性能的 Web Server 具有协作依赖性。

双重升级

在双重升级中，Web Proxy Server 和操作系统都要进行升级（如第 41 页中的“[双重升级：Java ES 和操作系统软件](#)”中所述）。可使用操作系统就地升级方法执行双重升级：

1. 备份现有 Web Proxy Server 数据。
有关基本数据的位置，参见第 223 页中的“[Web Proxy Server 数据](#)”。
2. 升级操作系统。
升级会将现有文件系统留在原来的位置。
3. 将 Web Proxy Server 升级到发行版本 5。
参见本章中的相应小节，这要视升级方式而定。

从 Java ES 发行版本 4 升级 Web Proxy Server

本节包含关于将 Web Proxy Server 从 Java ES 2005Q4（发行版本 4）升级到 Java ES 5（发行版本 5）的信息。其中涵盖以下主题：

- [简介](#)
- [发行版本 4 Web Proxy Server 的升级](#)

简介

在将 Web Proxy Server 升级到发行版本 5 时，请考虑升级过程中的以下方面：

- **常规升级方法。** 使用修补程序执行升级。不需要进行额外的重新配置。
- **升级依赖性。** Web Proxy Server 对许多 Java ES 共享组件都具有依赖性（参见 [第 43 页中的表 1-9](#)），但 Web Proxy Server 仅对 NSS 和 NSPR 共享组件具有硬性升级依赖性。
- **向后兼容性。** 发行版本 5 Web Proxy Server 与发行版本 4 向后兼容。
- **升级回滚。** Web Proxy Server 的发行版本 5 升级回滚可通过删除升级修补程序来实现。
- **平台问题。** 在 Solaris 和 Linux 操作系统上，升级 Web Proxy Server 的常规方法相同，但是修补技术却不同。因此，升级过程包括特定于平台的操作过程。

发行版本 4 Web Proxy Server 的升级

本节概述了如何将 Web Proxy Server 升级到 Java ES 发行版本 5。其中涵盖了以下主题：

- [升级前的任务](#)
- [升级发行版本 4 Web Proxy Server \(Solaris\)](#)
- [升级发行版本 4 Web Proxy Server \(Linux\)](#)
- [检验升级](#)
- [升级后的任务](#)
- [升级回滚 \(Solaris\)](#)

升级前的任务

升级 Web Proxy Server 之前，应执行以下任务：

- [检验当前版本信息](#)
- [升级 Web Proxy Server 依赖项](#)
- [备份 Web Proxy Server 数据](#)
- [获取所需的配置信息和密码](#)

检验当前版本信息

可通过输入以下命令检验 Web Proxy Server 的当前版本：

```
WebProxyServer-base/proxy-admserv/start -version
```

表 13-4 Web Proxy Server 版本检验输出

Java ES 发行版本	Web Proxy Server 版本号
非 Java ES 发行版本的版本 3.6	3.6
发行版本 4	4.0.1
发行版本 5	4.0.4

升级 Web Proxy Server 依赖项

通常建议将计算机系统（以及计算环境）中的所有 Java ES 组件都升级到 Java ES 发行版本 5。但是，将 Web Proxy Server 升级到发行版本 5 仅要求升级 NSS 和 NSPR 共享组件。如果这两个共享组件尚未升级，应使用“同步共享组件”选项将所有共享组件同步到各自的发行版本 5 的版本。[第 59 页中的第 2 章“升级 Java ES 共享组件”](#)中提供了相关说明。

备份 Web Proxy Server 数据

将 Web Proxy Server 升级到发行版本 5 不会修改发行版本 4 的配置数据。没有必要备份当前数据。

获取所需的配置信息和密码

不需要任何与当前安装版本有关的特殊信息，但是必须以超级用户身份登录才能执行升级。

升级发行版本 4 Web Proxy Server (Solaris)

本节讨论影响 Web Proxy Server 升级过程的一些注意事项，随后对升级过程本身加以说明。

升级注意事项 (Solaris)

将 Web Proxy Server 软件升级到 Java ES 发行版本 5 时，应考虑以下注意事项：

- 所有对应同一个已安装 Web Proxy Server 映像的 Web Proxy Server 实例将同时被升级。向安装的映像应用修补程序时，应关闭所有此类实例。
- 下表显示了用于 Solaris 操作系统的发行版本 5 的 Web Proxy Server 升级修补程序：

表 13-5 用于在 Solaris 上升级 Web Proxy Server 的修补程序¹

说明	修补程序 ID: SPARC Solaris 9 和 10	修补程序 ID: X86 Solaris 9 和 10
Web Proxy Server 核心组件	120981-10	120982-10
Web Proxy Server 本地化版本	122963-01	122964-01

1. 修补程序修订版本号是升级到 Java ES 发行版本 5 所需的最低版本号。如果有较新的修订版本可用，请用较新版本替代该表中所示的版本。

升级过程 (Solaris)

下述过程适用于要进行升级的计算机中的 Web Proxy Server。

1. 以 root 身份登录，或成为超级用户。

```
su -
```

2. 停止所有运行的 Web Proxy Server 和 Administration Server 实例。

```
WebProxyServer-base/proxy-instanceName/stop
```

```
WebProxyServer-base/proxy-admserv/stop
```

3. 将所有共享组件同步到发行版本 5（如果尚未执行此操作）。

第 59 页中的第 2 章“升级 Java ES 共享组件”中提供了有关说明。

4. 根据表 13-5，获取所需的修补程序。

可从以下网址将修补程序下载到 /tmp:

<http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access>

5. 应用适当的 Web Proxy Server 核心组件，如果需要，以表 13-5 中的顺序应用该表中的本地化修补程序。

```
patchadd /tmp/patch_ID
```

6. 确认修补程序升级成功:

```
showrev -p | grep proxy
```

输出应返回在步骤 5 中应用的修补程序 ID 的版本。

7. 重新启动在步骤 2 中停止的 Web Proxy Server 实例。

```
WebProxyServer-base/proxy-instanceName/start
```

升级发行版本 4 Web Proxy Server (Linux)

本节讨论影响 Web Proxy Server 升级过程的一些注意事项，随后对升级过程本身加以说明。

升级注意事项 (Linux)

除 Linux 发行版本 5 的升级修补程序与 Solaris 的修补程序不同外，在 Linux 平台上将 Web Proxy Server 软件升级到 Java ES 发行版本 5 的注意事项与在 Solaris 平台上相同（参见第 233 页中的“升级注意事项”）。

下表显示了 Linux 操作系统的发行版本 5 的 Web Proxy Server 升级修补程序：

表 13-6 用于在 Linux 上升级 Web Proxy Server 的修补程序¹

说明	修补程序 ID 和 RPM 名称
Web Proxy Server 核心组件	120983-10 sun-proxyserver-4.0-6.5.i386.rpm
Web Proxy Server 本地化版本	122965-01 sun-proxyserver-Locale-4.0.4.i386.rpm

1. 修补程序修订版本号是升级到 Java ES 发行版本 5 所需的最低版本号。如果有较新的修订版本可用，请用较新版本替代该表中所示的版本。

升级过程 (Linux)

下述过程适用于要进行升级的计算机中的 Web Proxy Server。

注意	在 Linux 上从 Java ES 发行版本 4 升级到 Java ES 发行版本 5 无法回滚。
-----------	--

1. 以 root 身份登录，或成为超级用户。

```
su -
```
2. 停止所有运行的 Web Proxy Server 和 Administration Server 实例。

```
WebProxyServer-base/proxy-instanceName/stop
WebProxyServer-base/proxy-admserv/stop
```
3. 将所有共享组件同步到发行版本 5（如果尚未执行此操作）。
参见第 227 页中的“升级 Web Proxy Server 依赖项”。
4. 使用表 13-6 中的修补程序号和 RPM 名称，获取所需的修补程序。使用此信息获取 RPM 的版本号。

可从以下网址将修补程序下载到 /tmp：
<http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access>
5. 应用核心组件，如果需要，以表 13-6 中的顺序应用该表中 Web Proxy Server 的本地化 RPM。

```
rpm -Fvh sun-proxyserver-version.i386.rpm
```

6. 确认修补程序升级成功:

```
rpm -qa | grep sun-proxyserver
```

应返回 RPM 的新版本号。

7. 重新启动在步骤 2 中停止的 Web Proxy Server 实例。

```
WebProxyServer-base/proxy-instanceName/start
```

检验升级

可通过用 `-version` 选项启动 Web Proxy Server 实例来检验 Web Proxy Server 到发行版本 5 的升级:

```
WebProxyServer-base/proxy-admserv/start -version
```

参见第 226 页中的表 13-4 确认输出值。

升级后的任务

除第 228 页中的“升级过程 (Solaris)”和第 229 页中的“升级过程 (Linux)”中所述的步骤之外, 没有任何升级后的任务。

升级回滚 (Solaris)

本节介绍影响 Web Proxy Server 升级回滚过程的一些注意事项, 随后对升级回滚过程本身加以说明。

回滚注意事项 (Solaris)

升级到发行版本 5 的 Web Proxy Server 的升级回滚过程几乎就是升级到发行版本 5 的相反过程。

回滚过程 (Solaris)

1. 以 root 身份登录, 或成为超级用户。

```
su -
```

2. 停止所有运行的 Web Proxy Server 和 Administration Server 实例。

```
WebProxyServer-base/proxy-instanceName/stop
```

```
WebProxyServer-base/proxy-admserv/stop
```

3. 删除第 227 页中的表 13-5 中的修补程序。

```
patchrm patch_ID
```

4. 重新启动在步骤 2 中停止的 Web Proxy Server 实例。

```
WebProxyServer-base/proxy-instanceName/start
```

从版本 3.6 升级 Web Proxy Server

本节包含关于将 Web Proxy Server 从版本 3.6 升级到 Java ES 5（发行版本 5）的信息。其中涵盖了以下主题：

- [简介](#)
- [版本 3.6 Web Proxy Server 升级](#)

简介

在将 Web Proxy Server 升级到发行版本 5 时，请考虑升级过程中的以下方面：

- **常规升级方法。** 通过使用 Java ES 安装程序在不同于版本 3.6 的目录中安装发行版本 5 Web Proxy Server 来执行升级。然后使用 Web Proxy Server Administration Server 将配置设置（非高速缓存中的内容）从版本 3.6 迁移到发行版本 5。
- **升级依赖性。** 将 Web Proxy Server 升级到发行版本 5 时，Java ES 安装程序会自动执行共享组件的安装。
- **向后兼容性。** 发行版本 5 的 Web Proxy Server 与版本 3.6 向后兼容，但必须用发行版本 5 支持的 NSAPI 接口重新编译用版本 3.6 支持的 NSAPI 接口所开发的插件。
- **升级回滚。** 发行版本 5 的 Web Proxy Server 的升级回滚通过恢复到版本 3.6 来实现，该版本在升级时保持不变。
- **平台问题。** 在所有 Solaris 平台上，Web Proxy Server 软件的升级方法都是相同的，但在 Linux 平台上不支持版本 3.6。

版本 3.6 Web Proxy Server 升级

本节概括介绍了如何将 Web Proxy Server 升级到 Java ES 发行版本 5。以前，在 Linux 平台上不支持 Web Proxy Server。因此，只能在 Solaris 平台上将 Web Proxy Server 升级到 Java ES 发行版本 5。本节涵盖了以下主题：

- [升级前的任务](#)
- [升级版本 3.6 Web Proxy Server](#)
- [检验升级](#)
- [升级后的任务](#)
- [版本 3.6 的升级回滚](#)

升级前的任务

升级 Web Proxy Server 之前，应执行以下任务：

- [检验当前版本信息](#)
- [升级 Web Proxy Server 依赖项](#)
- [备份 Web Proxy Server 数据](#)
- [获取所需的配置信息和密码](#)

检验当前版本信息

可通过输入以下命令检验 Web Proxy Server 的当前版本：

```
WebProxyServer-base/proxy-serverid/start -version
```

表 13-7 Web Proxy Server 版本检验输出

Java ES 发行版本	Web Proxy Server 版本号
非 Java ES 发行版本的版本 3.6	3.6
发行版本 4	4.0.1
发行版本 5	4.0.4

升级 Web Proxy Server 依赖项

通常建议将计算机系统（以及计算环境）中的所有 Java ES 组件都升级到 Java ES 发行版本 5。但是，用于将 Web Proxy Server 升级到发行版本 5 的 Java ES 安装程序会自动升级 Web Proxy Server 所依赖的所有共享组件（参见第 43 页中的表 1-9）。

备份 Web Proxy Server 数据

Web Proxy Server 升级到发行版本 5 时不会修改版本 3.6 的配置数据。但是，在升级之前，必须保存任何未保存的用管理界面对版本 3.6 配置数据所做的更改。

获取所需的配置信息和密码

要从版本 3.6 升级，需要知道该已安装版本的安装目录路径。

升级版本 3.6 Web Proxy Server

本节讨论影响 Web Proxy Server 升级过程的一些注意事项，随后对升级过程本身加以说明。

升级注意事项

对应于同一个已安装 Web Proxy Server 映像的所有 Web Proxy Server 实例都可以升级。但是，对于每个实例，配置数据的迁移必须分别进行。执行迁移时所有此类实例都应该关闭，以确保在启动已迁移的实例时不发生任何端口冲突。

升级过程

下述过程适用于要进行升级的计算机中的 Web Proxy Server 软件。

1. 以 root 身份登录，或成为超级用户。

```
su -
```
2. 安装 Web Proxy Server 发行版本 5。
 - a. 通过 Java ES 发行版本 5 分发软件运行 Java ES 安装程序。
 - b. 从组件选择页面中选择 Web Proxy Server。
 - c. 选择“现在配置”选项。
 - d. 安装完成后退出 Java ES 安装程序。

3. 将配置设置迁移到新安装版本中。

必须分别为每个 Web Proxy Server 实例执行此操作。

a. 启动 Web Proxy Server Administration Server。

WebProxyServer-base/proxy-admserv/start

b. 登录到图形管理界面。

c. 单击 "Server" 选项卡，然后单击 "Migrate Server"。

d. 指定版本 3.6 安装目录路径。

e. 选择要迁移的实例。

f. 单击 "Migrate" 按钮。

迁移成功后，迁移屏幕将提供附加配置的列表，必须手动执行这些配置。屏幕会提供需要添加的数据及相应的配置文件。

有关迁移配置设置的更多信息，参阅《Sun Java System Web Proxy Server 4.0.4 Installation and Migration Guide》(<http://docs.sun.com/doc/819-5492>)。

4. 执行步骤 f 中指定的任何附加配置更改。

有关更多信息，参阅《Sun Java System Web Proxy Server 4.0.4 Configuration File Reference》(<http://docs.sun.com/doc/819-5494>)。

检验升级

可通过用 `-version` 选项启动 Web Proxy Server 实例来检验 Web Proxy Server 到发行版本 5 的升级：

WebProxyServer-base/proxy-serverid/start -version

参见第 232 页中的表 13-7 确认输出值。

升级后的任务

除第 233 页中的“升级过程”中所述的步骤之外，没有任何升级后的任务。

版本 3.6 的升级回滚

第 233 页中的“升级版本 3.6 Web Proxy Server”中所述的 Web Proxy Server 到发行版本 5 的升级无法执行回滚操作。但是，您可以复原到版本 3.6，该版本在经过发行版本 5 升级过程后仍保持原样。

Access Manager

本章介绍如何将 Access Manager 软件从先前的 Java ES 版本升级到 Java ES 5（发行版本 5）：Sun Java System Access Manager 7.1。

本章概括介绍了有关发行版本 5 所支持的不同升级方式的 Access Manager 升级问题及过程。其内容涵盖了在 Solaris 和 Linux 操作系统上的升级：

- [第 236 页中的“Access Manager 升级概述”](#)
- [第 242 页中的“从 Java ES 发行版本 4 升级 Access Manager”](#)
- [第 262 页中的“从 Java ES 发行版本 3 升级 Access Manager”](#)
- [第 265 页中的“从 Java ES 发行版本 2 升级 Access Manager”](#)

注 本章中的文件位置是相对于两个目录路径指定的，分别称为 *AccessManager-base* 和 *AccessManagerConfig-base*。在最初安装 Access Manager 时，可能至少已将这两个路径的一部分指定为安装目录，否则，Java ES 安装程序会分配一个默认值。

下表说明了这些目录路径的默认值。

表 14-1 Access Manager 目录路径

路径名变量	Solaris OS	Linux OS
<i>AccessManager-base</i>	/opt/SUNWam	/opt/sun/identity
<i>AccessManagerConfig-base</i>	/etc/opt/SUNWam	/etc/opt/sun/identity

Access Manager 升级概述

本节将在以下方面对 Access Manager 进行介绍，这些方面会对升级到 Java ES 5（发行版本 5）产生影响：

- [关于 Java ES 发行版本 5 Access Manager](#)
- [Access Manager 升级指导说明](#)
- [Access Manager 数据](#)
- [Access Manager 升级策略](#)

注

在 Java ES 发行版本 3 之前的 Access Manager 版本名为 Identity Server。因此，当在本章提到 Identity Server 时，它是指 Java ES Access Manager 组件的早期版本。

关于 Java ES 发行版本 5 Access Manager

Java ES 发行版本 5 Access Manager 只是一个次要发行版本，其中包含针对 Java ES 发行版本 4 Access Manager 这个主要发行版本的许多错误修复和增强功能。在发行版本 5 的增强功能中，新增了一个基于 Java ES 监视框架的监视功能。有关发行版本 5 增强功能的更多信息，参见《Sun Java System Access Manager 7.1 发行说明》(<http://docs.sun.com/doc/820-0365>)。

与发行版本 4 类似，发行版本 5 Access Manager 支持多个身份系统信息库（或用户数据存储库）。因此，发行版本 5 Access Manager 不仅支持 LDAP 目录（如 Directory Server），而且还支持其他数据存储协议和格式。

在前端，通过使用 Access Manager 控制台来配置新的 Access Manager 服务和身份系统信息库。

为了提供与其他 Java ES 组件的向后兼容性，发行版本 5 可在传统模式下运行，传统模式支持依赖于发行版本 3 Access Manager 服务的 Java ES 组件（有关更多信息，参见第 239 页中的“兼容性问题”）。

Access Manager 升级指导说明

表 14-2 显示了所支持的、将 Access Manager 升级到 Java ES 发行版本 5 的升级方式。此表既适用于 Solaris，也适用于 Linux 操作系统。

表 14-2 升级到 Java ES 5（发行版本 5）：Access Manager 7.1 的方式

Java ES 发行版本	Access Manager 版本	常规方法	必需的重新配置
发行版本 4	Sun Java System Access Manager 7.0 2005Q4	直接升级： 执行方法是使用升级前脚本删除发行版本 4 的版本，然后执行发行版本 5 的完全安装和重新配置。	配置数据 用于 Access Manager 控制台和验证 UI 的自定义 JSP 目录模式
发行版本 3	Sun Java System Access Manager 5 2006.3Q1	直接升级： 执行方法是使用升级前脚本删除发行版本 3 的版本，然后执行发行版本 5 的完全安装和重新配置。	配置数据 用于 Access Manager 控制台和验证 UI 的自定义 JSP 目录模式
发行版本 2	Sun Java System Identity Server 6.2 2004Q2 及 6.2 SP1	直接升级： 执行方法是使用升级前脚本删除发行版本 2 的版本，然后执行发行版本 5 的完全安装和重新配置。	配置数据 用于 Access Manager 控制台和验证 UI 的自定义 JSP 目录模式
发行版本 1	Sun ONE Identity Server 6.1	不可直接升级： 但可以按照 Java Enterprise System 2005Q1 升级与迁移指南 (http://docs.sun.com/doc/819-2237) 中的过程首先升级到发行版本 3。然后从发行版本 3 升级到发行版本 5。	配置数据 用于 Access Manager 控制台和验证 UI 的自定义 JSP 目录模式
Java ES 以前的发行版本	Sun ONE Identity Server 6.0 或 6.0 SP 1，或 iPlanet Directory Server Access Management Edition (DSAME) 5.1	不可直接升级。	

Access Manager 数据

与其他 Java ES 组件一样，Access Manager 将利用对于任一特定升级而言可能需要迁移至升级版本的各种数据。下表显示了可能受 Access Manager 软件升级影响的数据的类型。

表 14-3 Access Manager 数据用途

数据类型	位置	用途
配置数据	<i>AccessManagerConfig-base/config/AMConfig.properties</i> <i>AccessManagerConfig-base/config/serverconfig.xml</i> 验证和自定义模块的 JAR 文件 <i>AccessManager-base/lib</i>	Access Manager 的配置及其与后端数据存储器的集成。
Web 容器访问控制和配置文件	Web Server 7.0 (Java ES 发行版本 5) server.policy 和 server.xml 文件，位于 <i>WebServer7Config-base/https-configName/config</i> Web Server 6.x (Java ES 发行版本 2、3 和 4) server.policy 和 server.xml 文件，位于 <i>WebServer6-base/https-hostname/config</i> Application Server 8.x (Java ES 发行版本 3、4 和 5) : server.policy 和 domain.xml 文件，位于 <i>AppServer8Config-base/domains/domainName/config</i> Application Server 7.x (Java ES 发行版本 2) : server.policy 和 server.xml 文件，位于 <i>AppServer7Config-base/domains/domainName/config</i> WebSphere 和 WebLogic: 为这两个 Web 容器配置 Access Manager 时修改它们各自的策略和配置文件。	Access Manager Web 容器实例的配置。
自定义数据 (Web 容器自定义 JSP 文件)	管理控制台: (Java ES 发行版本 2 和 3) : <i>AccessManager-base/web-src/applications</i> 管理控制台: (Java ES 发行版本 4 和 5) : <i>AccessManager-base/web-src/services</i> 验证 UI: <i>AccessManager-base/web-src/services</i>	Access Manager 管理接口的配置。
目录模式 服务配置 用户数据	Directory Server	Access Manager 根据服务配置、用户以及存储在目录中的策略数据，为最终用户提供验证和授权服务。
动态应用程序数据	无	Access Manager 并不永久存储应用程序数据 (如会话状态)。

Access Manager 升级策略

Access Manager 的升级策略通常取决于在第 1 章“升级规划”中讨论的许多注意事项：升级方式、Java ES 组件之间的依赖性、选择性升级与全部升级、多实例部署，等等。

本节通过提出可能影响 Access Manager 升级计划的问题，对 Access Manager 的一般注意事项进行逐一介绍。

兼容性问题

发行版本 5 Access Manager 与发行版本 4 Access Manager 向后兼容，但是，发行版本 4 Access Manager 是主要发行版本，只有将其配置为在传统模式下运行，它才会与早期发行版本兼容。同样，只有将发行版本 5 Access Manager 配置为在传统模式下运行，它才会与发行版本 3 Access Manager（或在传统模式下运行的发行版本 4 Access Manager）向后兼容。

此外，发行版本 5 Access Manager 在任何模式下都不能与发行版本 2 Access Manager 向后兼容；发行版本 5 Access Manager 不能与发行版本 2 Access Manager SDK 交互操作，反之亦然。

当配置为在更新的领域模式下运行时，发行版本 5 Access Manager 支持多个身份系统信息库和数据存储协议。目录数据必须迁移到新结构中以支持领域模式操作。此外，领域模式不支持其他 Java ES 组件（如 Portal Server）或 Sun Java Communications Suite 组件（如 Communications Express、Messaging Server，等等）。

当配置为在传统模式下运行时，发行版本 5 Access Manager 与发行版本 3 Access Manager 及相应的目录数据向后兼容，但有少数几种例外情况（参见《Sun Java System Access Manager 7.1 发行说明》(<http://docs.sun.com/doc/820-0365>)）。

要支持其他 Java ES 组件和旧版 Access Manager 策略代理（在领域模式下，这些策略代理无法与 Access Manager 进行交互操作），必须使用传统模式。这一不兼容性是升级时的一个重要注意事项，同时意味着在大多数的 Java ES 部署中，应该将 Access Manager 升级到发行版本 5 传统模式。

然而，即便配置为在传统模式下运行，发行版本 5 Access Manager 仍与发行版本 3 或早期 Sun Java Communications Suite 组件不兼容。如果 Access Manager 已升级到发行版本 5，则必须将发行版本 3 或早期 Delegated Administrator 也升级到发行版本 5，以便为 Messaging Server 和 Calendar Server 置备用户。但 Messaging Server 和 Calendar Server 本身则不必一定升级到发行版本 5。

与发行版本 4 Access Manager 控制台相同的是，发行版本 5 控制台也是既支持领域模式又支持传统模式。但是，如果已将 Access Manager 配置为在传统模式下运行，则仍可以使用在发行版本 2 和发行版本 3 中分发的“仅传统”控制台。

Access Manager 组件依赖性

Access Manager 对其他 Java ES 组件的依赖性会影响升级和重新配置 Access Manager 软件的过程。例如，Access Manager 接口或功能上的变化可能要求 Access Manager 所依赖组件的升级版本。是否需要升级此类组件取决于具体的升级方式。

Access Manager 对以下 Java ES 组件具有依赖性：

- **共享组件。** Access Manager 对特定的 Java ES 共享组件具有依赖性（参见第 43 页中的表 1-9）。Access Manager 升级可能要依赖于这些共享组件的已升级版本。
- **Web 容器。** Access Manager 对 Web 容器服务具有强制依赖性，这些服务可由 Java ES Web Server、Java ES Application Server 或第三方 Web 容器（来自 Weblogic 和 WebSphere）提供。Access Manager 升级可能需要将 Access Manager 控制台或验证 UI 的自定义 JSP 迁移到已升级 Access Manager 环境中。
- **Directory Server。** Access Manager 对 Directory Server 具有强制依赖性，后者用于存储配置数据和用户数据。因此，Access Manager 升级可能需要扩展目录模式。

Web 容器升级方案

Access Manager 可部署在由 Web Server 或 Application Server 提供的 Web 容器中。因此，Access Manager 到发行版本 5 的升级可能会复杂化，因为可能还需要将用于部署该组件的 Web 容器升级到发行版本 5。关于这一点，存在多种可能的 Web 容器升级方案，将在下表中一一列出。

表 14-4 用于 Access Manager 升级的 Web 容器升级方案

方案	最初部署 Access Manager 的 Web 容器	升级后部署 Access Manager 的 Web 容器	适用的 Access Manager 升级方式：升级始于
方案 1	Web Server 6.x	Web Server 6.x	发行版本 2 发行版本 3 发行版本 4
方案 2	Web Server 6.x	Web Server 7.0	发行版本 2 发行版本 3 发行版本 4
方案 3	Application Server 8.1	Application Server 8.1	发行版本 3 发行版本 4
方案 4	Application Server 8.1	Application Server 8.2	发行版本 3 发行版本 4
方案 5	Application Server 7x	Application Server 8.2	发行版本 2

升级 Access Manager（例如，使用 amconfig 脚本时）以提供适合于表 14-4 中适用升级方案的值时，特别是存在 Web 容器的主要版本升级时，必须要小心。

双重升级

在双重升级中，Access Manager 和操作系统都要进行升级（如第 41 页中的“[双重升级：Java ES 和操作系统软件](#)”中所述），但 Access Manager 不支持双重升级。

因此，如果您遇到需要双重升级的情况，必须先执行操作系统的安装或升级，然后再执行 Access Manager 的重新安装和全新配置。

从 Java ES 发行版本 4 升级 Access Manager

本节包含关于将 Access Manager 从 Java ES 2005Q4（发行版本 4）升级到 Java ES 5（发行版本 5）的信息。其中涵盖以下主题：

- [简介](#)
- [发行版本 4 Access Manager 的全升级](#)
- [多实例升级](#)
- [仅升级发行版本 4 Access Manager SDK](#)

简介

将 Java ES 发行版本 4 Access Manager 升级到发行版本 5 时，请考虑升级过程中的以下方面：

- **常规升级方法。** 通过删除先前的二进制版本然后重新安装发行版本 5 执行升级。随附了一个 `ampre71upgrade` 脚本以删除发行版本 4 的版本，然后使用 Java ES 安装程序来安装发行版本 5。随后使用 `amconfig` 脚本对 Access Manager 进行重新配置，并使用 `amupgrade` 脚本迁移目录模式。
- **升级依赖性。** Access Manager 对许多 Java ES 共享组件都具有依赖性（参见 [第 43 页中的表 1-9](#)），当升级 Access Manager 时，Java ES 安装程序会自动将所有这些组件升级到发行版本 5。其中包括对新增的 Java ES 监视框架的支持，该框架需要发行版本 4 Access Manager 所不需要的一些共享组件。

此外，如 [第 240 页中的“Access Manager 组件依赖性”](#) 中所述，发行版本 5 Access Manager 还依赖于 Directory Server 和 Web Server（或 Application Server 或第三方 Web 容器）。但它们是软性升级依赖性，对于 Access Manager 至发行版本 5 的升级，这些组件的升级是可选的。

- **向后兼容性。** 发行版本 5 Access Manager 与发行版本 4 兼容，但与更早的 Access Manager 发行版本不兼容（参见 [第 239 页中的“兼容性问题”](#)）。
- **升级回滚。** 没有用于回滚 Access Manager 升级的实用程序。事实上，将 Access Manager 回滚至其原始状态所需的重新配置的数字过多，从而导致回滚无法实现。回滚的最佳方法是使用备份的配置文件创建一个并行安装并在执行升级前测试此并行安装。这样，您就可以在需要时恢复为并行安装。
- **平台问题。** 对于 Solaris 和 Linux 操作系统，升级 Access Manager 的常规方法是相同的。随后的过程将指明特定于平台的命令或适当的文件位置。

发行版本 4 Access Manager 的全升级

本节介绍如何从 Java ES 发行版本 4 到 Java ES 发行版本 5 执行 Access Manager 的完全升级：

- 升级前的任务
- 升级发行版本 4 Access Manager
- 检验 Access Manager 升级
- 升级后的任务
- 升级回滚

升级前的任务

升级 Access Manager 之前，应执行以下任务：

- 检验当前版本信息
- 升级 Access Manager 依赖项
- 备份 Directory Server 数据
- 备份发行版本 4 Access Manager 配置信息
- 备份 Web 容器自定义文件
- 备份发行版本 4 Access Manager 日志和调试文件
- 备份自定义本地化文件
- 获取所需的配置信息和密码

检验当前版本信息

可使用以下命令检验 Access Manager 的当前版本：

```
AccessManager-base/bin/amadmin --version
```

表 14-5 Access Manager 版本检验输出

Java ES 发行版本	Access Manager 版本号
发行版本 2	6.2
发行版本 3	6 2005Q1
发行版本 4	7 2005Q4
发行版本 5	7.1

升级 Access Manager 依赖项

通常建议将计算机系统（以及计算环境）中的所有 Java ES 组件都升级到 Java ES 发行版本 5。Access Manager 对许多共享组件具有硬性升级依赖性（参见第 43 页中的表 1-9）。

如果选择升级 Access Manager 产品组件所依赖的组件，应在升级 Access Manager 之前按以下顺序执行（跳过任何可能已经升级的部分）。共享组件的升级通常是由 Java ES 安装程序自动实现的。

1. **共享组件。** 第 59 页中的“升级 Java ES 共享组件”中提供了将 Java ES 共享组件同步到发行版本 5 的说明。不过，在执行 Access Manager 至发行版本 5 的升级时，Java ES 安装程序会自动升级 Access Manager 所需的所有共享组件。
2. **Directory Server（软性升级依赖性）。** 第 89 页中的第 5 章“Directory Server”中提供了将 Directory Server 升级到发行版本 5 的说明。
3. **Web 容器软件（软性升级依赖性）。** 第 119 页中的第 7 章“Web Server”和第 183 页中的第 11 章“Application Server”中分别提供了有关升级 Web Server 和 Application Server 的说明。

如果未在升级 Access Manager 之前升级 Web 容器软件，升级过程（使用 amconfig 脚本）会将 Access Manager 配置和重新部署到现有 Web 容器中。

备份 Directory Server 数据

Access Manager 升级过程所使用的脚本会修改 Directory Server 模式。因此，在升级 Access Manager 之前，请使用 Directory Server 控制台或命令行实用程序（如 db2bak）备份 Directory Server 数据。您可以使用 db2ldif 来备份 Access Manager 模式和目录信息树 (directory information tree, DIT)。

有关备份 Directory Server 的更多信息，参见《Sun Java System Directory Server Enterprise Edition 6.0 管理指南》(<http://docs.sun.com/doc/820-0293>)。

备份发行版本 4 Access Manager 配置信息

由于重新配置发行版本 5 Access Manager 软件需要重新配置发行版本 4 的版本，因此，在一个已知位置备份配置文件是非常重要的。应该备份以下文件：

- AMConfig.properties 文件
AccessManagerConfig-base/config/AMConfig.properties
- serverconfig.xml 文件
AccessManagerConfig-base/config/serverconfig.xml

- Web 容器配置文件：
 - 对于 Web Server：参见第 238 页中的表 14-3 中 `server.policy` 和 `server.xml` 文件的位置
 - 对于 Application Server：参见第 238 页中的表 14-3 中 `server.policy` 和 `domain.xml` 文件的位置
 - 对于第三方 Web 容器：相应的配置文件
- 验证和自定义模块的 JAR 文件。

AccessManager-base/lib

备份 Web 容器自定义文件

如果 Access Manager 引用了任何 Web 容器自定义文件，应备份这些文件。此类自定义内容可能包括：

- 自定义的 Access Manager 控制台 JSP 页：
 - 领域 / 传统控制台（随 Java ES 发行版本 4 一起分发）
AccessManager-base/web-src/services/
 - 仅传统控制台（随 Java ES 发行版本 2 和 3 一起分发）
AccessManager-base/web-src/applications/
- 自定义的验证 UI JSP 页。
AccessManager-base/web-src/services/
- 自定义的 XML 文件。
AccessManagerConfig-base/config/xml/

提示 将自定义内容记录下来，以便可以在升级 Access Manager 之后使用备份的代码重新应用它们。

备份发行版本 4 Access Manager 日志和调试文件

要分析系统状态信息，最好的办法是备份日志和调试文件以使其不会丢失。这些文件位于以下位置：

- 调试文件
/var/AccessManager-base/debug
- 日志文件
/var/AccessManager-base/logs

备份自定义本地化文件

如果您对 Java ES 安装程序所安装的本地化文件进行了任何自定义或新增了 Java ES 安装程序所未安装的语言本地化文件，则应该备份这些自定义内容。这些自定义内容可能包括以下几项：

- 自定义的 Access Manager 用户界面本地化
*AccessManager-base/locale/*_Locale.properties*
- 自定义的验证 UI JSP 页
AccessManager-base/web-src/services/config/auth/default_Locale
- 自定义的联机帮助翻译
AccessManager-base/web-src/services/html/Locale

获取所需的配置信息和密码

要升级 Access Manager，必须提供具体的配置信息，其中包括：

- Access Manager 管理员用户 ID 和密码
- LDAP 用户 ID 和密码
- Access Manager 正在使用的 Directory Server 实例的目录管理员名称和密码

升级发行版本 4 Access Manager

将 Access Manager 软件升级到 Java ES 发行版本 5 包括重新配置 Access Manager 和迁移 Access Manager 数据的过程。

升级摘要

升级 Access Manager 的过程由下列步骤组成：

1. 升级 Access Manager mobile access 软件。
2. 删除 Java ES 发行版本 4 的 Access Manager 版本。使用 `ampre71upgrade` 脚本。
3. 如果到发行版本 5 的升级需要实现本地化，请删除发行版本 4 本地化软件包。必须手动执行此步骤。
4. 安装 Java ES 发行版本 5 的 Access Manager 版本。使用 Java ES 安装程序并选择“以后再配置”选项。
5. 重新自定义 Access Manager 的 JSP。
6. 取消部署 Access Manager，重新配置并重新部署到 Web 容器中。使用 `amconfig` 脚本。
7. 更新目录结构和模式。使用 `amupgrade` 脚本。

以下过程对这些步骤逐一进行说明。

升级过程

1. 升级 Access Manager mobile access 软件。

Access Manager mobile access 软件需要通过修补发行版本 4 的版本进行升级。
下表显示了所需的修补程序：

表 14-6 用于升级 Access Manager Mobile Access 软件的修补程序¹

说明	修补程序 ID: Solaris 9 和 10	修补程序 ID: Linux
Mobile	119530-05 (SPARC)	119532-05
Access 软件	119531-05 (x86)	• sun-identity-mobileaccess-6.2-25.3.i386.rpm • sun-identity-mobileaccess-config-6.2-25.3.i386.rpm

1. 修补程序修订版本号是升级到 Java ES 发行版本 5 所需的最低版本号。如果有较新的修订版本可用，请用较新版本替代该表中所示的版本。

- a. 利用表 14-6 中的修补程序号获取所需的修补程序。
可从以下网址将修补程序下载到 /tmp:
<http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access>
- b. 执行修补程序 README 文件中指示的任何修补前过程。
- c. 获取修补程序将要请求的以下参数的值：

表 14-7 Mobile Access 修补程序参数

参数	值
目录管理员的 DN	默认值: cn=Directory Manager
目录管理员的密码	

- d. 应用表 14-6 中的修补程序。

在 Solaris 上：
patchadd /tmp/patch_ID

在 Linux 上：
./update

执行修补程序 README 文件中指示的任何修补后过程。

2. 删除 Java ES 发行版本 4 的 Access Manager 版本。
- a. 以 root 身份登录到托管发行版本 4 Access Manager 的计算机或成为超级用户。

su -
- b. 将目录更改为 Java ES 发行版本 5 分发软件中的 `os_arch/Product/identity_svr/Tools` 目录，其中，`os_arch` 与您的平台匹配，如 `Solaris_sparc`。
- c. 获取 `ampre71upgrade` 脚本将要请求的以下参数的值：

表 14-8 Access Manager 配置参数: `ampre71upgrade`

参数	值
Directory Server 主机	设置全限定名称: <i>hostname.domian</i>
Directory Server 端口	指定一个非 SSL 端口号 ¹ 默认值: 389
顶级管理员 DN	默认值: <code>uid=amadmin,ou=People,default_org_DN</code>
顶级管理员密码	
存储备份文件的目录	默认值: <i>AccessManager-base</i>

1. 如果指定一个 Directory Server SSL 端口（如默认的 SSL 值 636），则无法成功完成升级前的过程。

- d. 确保 Directory Server 正在运行，否则请将其启动。
- e. 运行 `ampre71upgrade` 脚本。
- ./ampre71upgrade
- 该脚本将备份 Access Manager 配置文件，并删除发行版本 4 基本软件包（对于本地化软件包，必须按照下面的[步骤 3](#)手动删除）。

3. 如果到发行版本 5 的升级需要实现本地化，请删除发行版本 4 本地化软件包。

在上面 [步骤 2](#) 中运行的 `ampre71upgrade` 脚本并不删除本地化软件包，因此，您必须如下所示手动将其删除。

在 Solaris 上：

- a. 检查是否存在本地化软件包。

```
pkginfo | grep SUNWaml
pkginfo | grep SUNWamclnt
pkginfo | grep SUNWamdistauth
```

- b. 删除在上述 [步骤 a](#) 中找到的任何本地化软件包。

```
pkgrm SUNWamlLocale
pkgrm SUNWamclntLocale
pkgrm SUNWamdistauthLocale
```

在 Linux 上：

- a. 检查是否存在本地化 RPM。

```
rpm -qa | grep sun-identity-sdk-*
rpm -qa | grep sun-identity-clientsdk-*
rpm -qa | grep sun-identity-distauth-*
```

- b. 删除在上述 [步骤 a](#) 中找到的任何本地化 RPM。

```
rpm -e sun-identity-sdk-Locale-*
rpm -e sun-identity-clientsdk-Locale-*
rpm -e sun-identity-distauth-Locale-*
```

4. 安装 Java ES 发行版本 5 的 Access Manager 版本。

执行以下步骤：

- a. 在托管发行版本 4 Access Manager 的计算机上启动 Java ES 安装程序。

```
cd Java ES Release 5 distribution/os_arch
./installer
```

其中，`os_arch` 与您的平台匹配，如 `Solaris_sparc`。（对于命令行界面，使用 `installer -nodisplay` 选项。）

显示“欢迎”和“许可协议”页面之后，将向您显示一个组件选择页面。（如果检测到可由 Java ES 安装程序直接升级的已安装组件，它们的状态会显示为“可升级”。）

- b. 从组件选择页面中选择 Access Manager。

- c. 指定安装目录，需与发行版本 4 的安装目录相同。
 - d. 选择“以后再配置”选项。
 - e. 如果需要，选择安装本地化软件包的选项。
 - f. 安装完成后退出 Java ES 安装程序。
5. 重新自定义 Access Manager 的 JSP。

重新应用按第 245 页中的“备份 Web 容器自定义文件”中所述保存的、用于 Access Manager 控制台和验证用户界面 (User Interface, UI) 的发行版本 4 自定义 JSP 设置。

然后将自定义 JSP 文件复制到正确的目录：

- o 领域 / 传统 Access Manager 控制台
AccessManager-base/web-src/services/console
- o 仅传统 Access Manager 控制台
AccessManager-base/web-src/applications/console
- o 验证 UI: *AccessManager-base/web-src/services/config/auth/default* 或 *AccessManager-base/web-src/services/config/auth/default_Locale*（其中 *Locale* 是语言环境指示符，如 *ja*）

有关更多信息，参见《Sun Java System Access Manager 7.1 Developer's Guide》(<http://docs.sun.com/doc/819-4675>)。

6. 取消部署 Access Manager，重新配置并重新部署到 Web 容器中。

通过运行 *amconfig* 脚本为特定 Web 容器配置 Access Manager。*amconfig* 脚本（及相关联的 *amsamplesilent* 模板输入文件）位于以下目录中：

AccessManager-base/bin

有关 *amconfig* 脚本和 *amsamplesilent* 模板文件的信息，参见《Sun Java System Access Manager 7.1 管理指南》(<http://docs.sun.com/doc/820-0839>)。

执行以下步骤将 Access Manager 重新配置和重新部署到 Web 容器中：

- a. 如果选择升级 Web 容器软件（如第 244 页中的“升级 Access Manager 依赖项”中所述），需确保升级全部完成。
- b. 确保 Web 容器的管理实例正在运行，且处于 *amconfig* 脚本所支持的模式下，如下表所示：

表 14-9 amconfig 支持的管理服务器模式

Web 容器	支持的模式	默认端口号
Application Server (8.x): Java ES 发行版本 3、4 和 5	SSL (安全) 非 SSL	4849
Web Server (7.0): Java ES 发行版本 5	SSL (安全)	8989
Web Server (6.x): Java ES 发行版本 2、3 和 4	非 SSL	8888

- c. 如果 Web 容器在 SSL 模式下运行，请确保容器的 SSL 证书尚未过期并仍然有效。
- d. 如果在发行版本 5 Web Server 中部署 Access Manager，请禁用与 Access Manager 在同一实例中运行且依赖于 Access Manager 的所有 Java ES 组件。
这些组件可能是诸如 Portal Server 之类的组件，或诸如 Communications Express、Instant Messaging 或 Delegated Administrator 之类的 Sun Java Communications Suite 组件。

过程如下所示：

- I. 在 https://host:8989 上以管理员身份登录
- II. 转至“编辑虚拟服务器”。
- III. 选择“Web 应用程序”选项卡。
- IV. 选中所有依赖于 Access Manager 的应用程序。
- V. 单击“禁用”。
- VI. 单击“保存”。
- VII. 单击“部署暂挂” | “部署配置”。

配置更改将传播至 Web Server 实例。

- e. 检查 Directory Server 及相应的 Web 容器是否正在运行。
- f. 根据 amsamplesilent 模板输入文件创建一个 amconfig 输入文件：

```
cp amsamplesilent config-file
```


(在后续步骤中，假定 config-file 与 amsamplesilent 驻留在同一目录中。)

- g. 设置 *config-file* 中的配置参数。
- 需正确设置所有参数。一些参数的值可从 `AMConfig.properties` 文件迁移，相比之下，其他参数的值则是升级过程的专用值，如下表所示。

表 14-10 Access Manager 配置参数: `amconfig`

参数	值
升级参数	
DEPLOY_LEVEL	设置为 26 用于取消部署，或者 设置为 1 用于重新配置和部署
DIRECTORY_MODE	设置为 5
AM_REALM ¹	设置为 <code>disabled</code> ，如果启用传统模式 设置为 <code>enabled</code> ，如果启用领域模式 默认值: <code>enabled</code>
JAVA_HOME	设置为 JDK 发行版本 5 目录： <code>/usr/java/jdk1.5.0_04/</code>
WEB_CONTAINER	设置为 <code>WS</code> ，对于 Web Server 7.x 设置为 <code>WS6</code> ，对于 Web Server 6.x 设置为 <code>AS8</code> ，对于 Application Server 8.x 设置为 <code>WAS5</code> ，对于 IBM WebSphere 5.x 设置为 <code>WL8</code> ，对于 BEA WebLogic 8.x 且仅填写 <i>config-file</i> 的对应部分。
WS_INSTANCE (如果将 Web Server 7.x 用作 Web 容器)	设置为区分大小写的实例配置目录名： <code>https-configName/</code> 该目录位于以下路径中： <code>WebServer7Config-base/https-configName/</code>
WS61_INSTANCE (如果将 Web Server 6.x 用作 Web 容器)	设置为区分大小写的实例配置目录名： <code>https-instanceName</code> 该目录位于以下路径中： <code>WebServer6-base/https-instanceName/</code>
AS81_INSTANCE (将 Application Server 8.x 用 作 Web 容器)	设置为 Application Server 8.x <i>instanceName</i> 默认值: <code>server</code>
AS81_INSTANCE_DIR (将 Application Server 8.x 用 作 Web 容器)	设置为实例的 Application Server 8.x 域目录，其默认 值为 <code>AppServer8Config-base/domains/domain1</code>
AS81_DOCS_DIR (将 Application Server 8.x 用 作 Web 容器)	设置为实例的 Application Server 8.x 文档根目录，其 默认值为 <code>AppServer8Config-base/domains/domain1/docroot</code>

表 14-10 Access Manager 配置参数: amconfig (续)

参数	值
迁移自 AMConfig.properties	
SERVER_PROTOCOL	com.iplanet.am.server.protocol
SERVER_PORT	com.iplanet.am.server.port
SERVER_HOST	com.iplanet.am.server.host
DS_HOST	com.iplanet.am.directory.host
DS_PORT	com.iplanet.am.directory.port
ROOT_SUFFIX ²	com.iplanet.am.defaultOrg
CONSOLE_DEPLOY_URI	com.iplanet.am.console.deploymentDescriptor
SERVER_DEPLOY_URI	com.iplanet.am.services.deploymentDescriptor
PASSWORD_DEPLOY_URI	com.sun.identity.password.deploymentDescriptor
AM_ENC_PWD ²	am.encryption.pwd ³

- 1. 有关领域模式和传统模式的更多信息，参见第 239 页中的“兼容性问题”。
- 2. 此参数的值应与先前版本 Access Manager 中的参数值相同。
- 3. 如果同时部署了 Access Manager 和 Access Manager SDK，则对于 Access Manager 实例及其关联的 Access Manager SDK 实例，此属性的值必须相同。

对于其他参数，如果不更改 Web 容器和密码，请提供在要升级的发行版本 4 配置中使用的那些值。例如，如果已将 Web Server 升级到发行版本 5，请提供以下值：

表 14-11 amconfig 参数: 发行版本 5 Web Server

参数	值
WS_CONFIG	Web Server 配置的名称: <i>configName</i>
WS_INSTANCE	<i>https-configName</i>
WS_HOME	<i>WebServer7Config-base</i>
WS_PROTOCOL	http 或 https
WS_HOST	Web Server 实例侦听连接时使用的全限定主机名
WS_PORT	Web Server 实例侦听连接时使用的端口
WS_ADMINPORT	Web Server 管理实例侦听连接时使用的端口
WS_ADMIN	Web Server 管理员用户 ID
WS_ADMINPASSWD	Web Server 管理员密码

- h. 运行 `amconfig` 以取消部署 Access Manager。
将 `config-file` 中 `DEPLOY_LEVEL` 的值设置为 26。

```
cd /AccessManager-base/bin
./amconfig -s AccessManager-base/bin/config-file
```
- i. 检查以确保 Common Agent Container 正在运行。

```
netstat -an | grep 11163
```


如果 Common Agent Container 未运行，请将其启动。

```
/usr/sbin/cacoadm start
```
- j. 运行 `amconfig` 以重新配置 Access Manager 并将其部署到 Web 容器中。
将 `config-file` 中 `DEPLOY_LEVEL` 的值设置为 1。

```
cd /AccessManager-base/bin
./amconfig -s AccessManager-base/bin/config-file
```

7. 更新目录结构和模式。

发行版本 5 Access Manager 可与发行版本 4 的目录结构共存，但必须对该结构进行修改以使其支持发行版本 5 的功能。通过运行 `amupgrade` 脚本，将 Access Manager 目录结构和模式更新为发行版本 5，该脚本安装在以下目录中：

- o. 在 Solaris 上：
`AccessManager-base/upgrade/scripts`
- o. 在 Linux 上：
`AccessManager_base/identity/upgrade/scripts`
- a. 获取 `amupgrade` 脚本将要请求的以下参数的值：

表 14-12 Access Manager 配置参数: `amupgrade`

参数	值
Directory Server 全限定主机名	设置全限定名称: <i>hostname.domian</i>
Directory Server 端口	指定一个非 SSL 端口号 ¹ 默认值: 389
目录管理员 DN	默认值: <i>cn=Directory Manager</i>
目录管理员密码	
顶级管理员 DN	默认值: <i>uid=amadmin,ou=People,default_org_DN</i>
顶级管理员密码	

表 14-12 Access Manager 配置参数: amupgrade (续)

参数	值
启用领域模式 (从发行版本 4 领域模式升级时不请求此参 数值。)	Y/N: Yes 表明领域模式已启用且服务数据 已迁移到新的领域树 ² 。No (默认值) 表明 服务数据仍然处于传统模式下。

- 1. 必须指定一个与默认 SSL 值 636 不同的 Directory Server SSL 端口。
- 2. 参见第 256 页中的“迁移至领域模式”。

- b. 运行 amupgrade 脚本。

```
cd AccessManager-base/upgrade/scripts/  
./amupgrade
```

如果升级成功，脚本将显示 "Upgrade completed"。

- c. 查看以下升级日志文件以了解有关目录模式扩展的信息：

在 Solaris 上：

```
/var/sadm/install/logs/  
Sun_Java_System_Access_Manager_upgrade_dit_log.mmddhhmm
```

在 Linux 上：

```
/var/log/Sun_Java_System_Access_Manager_upgrade_dit_log.mmddhhmm
```

8. 启用在第 251 页中的步骤 d 中禁用的所有组件。

9. 重新启动部署了 Access Manager 的 Web 容器。

10. 启动 Access Manager。

重新启动部署了 Access Manager 的 Web 容器。

检验 Access Manager 升级

完成升级过程后，可按以下方法检验升级是否成功：

1. 使用以下命令检查 Access Manager 软件包的升级：

```
AccessManager-base/bin/amadmin --version
```

参见第 243 页中的表 14-5 确认输出值。

2. 通过检查 `/var/sadm/install/logs` 目录中的下列安装程序日志文件查看升级状态：

- `Java_Shared_Component_Install.timestamp`
- `Java_Enterprise_System_install.Atimestamp`
- `Java_Enterprise_System_install.Btimestamp`
- `Java_Enterprise_System_Summary_Report_install.timestamp`

3. 通过在运行 `amupgrade` 脚本的同时检查终端窗口中是否有错误来查看 Access Manager 的迁移状态。

另外，请检查 `/var/sadm/install/logs` 目录中的以下日志文件：

`Sun_Java_System_Access_Manager_upgrade_dit_log.timestamp`

4. 检查 Access Manager 错误诊断文件以了解错误。

这些文件的位置由 `AMConfig.properties` 文件的 `com.ipplanet.services.debug.directory` 属性指定。默认值为：

在 Solaris 上：

`/var/opt/SUNWam/debug`

在 Linux 上：

`/var/opt/sun/identity/debug`

升级后的任务

请注意处理以下情况所需要的升级后过程：

- [迁移至领域模式](#)
- [安全声明标记语言](#)

迁移至领域模式

如果在将 Access Manager 升级到发行版本 5 时，您已迁移到了领域模式（即，在使用 `amupgrade` 更新目录结构和模式时，对于是否启用领域模式，您的回答为 Yes），请执行以下步骤：

1. 打开 `AccessManagerConfig-base/config/AMConfig.properties` 文件。
2. 检查以下属性的值：
`com.sun.identity.sm.ldap.enableProxy`
3. 如果该属性未设置为 `false`，则手动将其设置为 `false`。

安全声明标记语言

如果要使用安全声明标记语言 (Security Assertion Markup Language, SAML) 服务，必须使用 Access Manager 控制台添加并启用一个 SAML 验证模块。有关如何创建 SAML 验证模块实例的信息，参阅《Sun Java System Access Manager 7.1 管理指南》(<http://docs.sun.com/doc/820-0839>)。

升级回滚

没有提供用于将 Access Manager 回滚至其升级前的状态的脚本，必须使用在执行升级前的任务过程中所备份的 Access Manager 数据来手动执行该过程（参见第 245 页中的“备份发行版本 4 Access Manager 日志和调试文件”）。升级回退过于困难，以至于无法实现。

一种回滚方法是重新安装发行版本 4 并将所有备份的配置文件迁移至它们的适当位置。另外一种方法是在尝试进行升级之前，先创建一个并行系统，然后使用备份的配置文件升级并测试该并行系统。

多实例升级

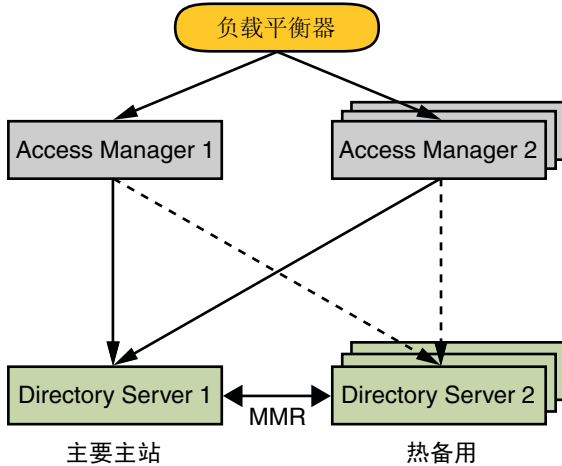
在一些部署体系结构中，Access Manager 被部署在多个计算机系统上，以提供高可用性和高可伸缩性。

用户常常希望在不中断服务的情况下按顺序升级 Access Manager 实例。本节将讨论如何执行从发行版本 4 Access Manager 至发行版本 5 的滚动升级过程。

注	当前发行版本不支持升级那些安装在同一主机系统上的多个 Access Manager 实例。如果在同一主机上有多个实例，在升级主实例之后，必须重新创建附加实例。
----------	---

下图所示的部署体系结构将用于说明滚动升级过程。

图 14-1 多 Access Manager 实例的部署体系结构示例



在此体系结构中，通过负载均衡器访问多个 Access Manager 实例，而这些实例又将访问为多主复制 (multi-master replication, MMR) 设置的目录。尽管可以使用其他 Directory Server 复制模式，但 MMR 是具有高可用性和高可伸缩性的目录服务。在图 14-1 中，对 Access Manager 和 Directory Server 的多个实例进行了编组以便于解释升级过程。例如，Access Manager 2 代表 Access Manager 的第 2 个到第 n 个实例。

从发行版本 4 Access Manager 至发行版本 5 的滚动升级过程基于以下互操作性：如果目录模式尚未更新为发行版本 5，则发行版本 5 Access Manager 和发行版本 4 Access Manager 实例可以共存并能够在同一个目录上同时运行。

因此，对于指向单一 Directory Server 实例的 Access Manager 实例，您可以延迟目录模式的更新直至所有 Access Manager 实例均完成升级为止，以此来执行滚动升级。

可按照以下过程执行从发行版本 4 Access Manager 至发行版本 5 的滚动升级：

1. 在所有 Access Manager 实例上备份发行版本 4 配置信息。

参见第 238 页中的表 14-3。

2. 升级 Access Manager 1。
 - a. 在负载均衡器中禁用 Access Manager 1。
请求不会再路由至 Access Manager 1。
 - b. 部分升级 Access Manager 1。
如第 246 页中的“升级发行版本 4 Access Manager”中所述升级 Access Manager，但不要按第 254 页中的步骤 7 中所述更新目录结构和模式。
 - c. 在负载均衡器中启用 Access Manager 1。
3. 将 Access Manager 2 到 Access Manager n 升级。
为简洁起见，在后续步骤中，“Access Manager 2”表示 Access Manager 2 到 Access Manager n。
 - a. 在负载均衡器中禁用 Access Manager 2。
请求不会再路由至 Access Manager 2。
 - b. 部分升级 Access Manager 2。
如第 246 页中的“升级发行版本 4 Access Manager”中所述升级 Access Manager 的每个实例，但不要按第 254 页中的步骤 7 中所述更新目录结构和模式。
 - c. 在负载均衡器中启用 Access Manager 2。
请求将重新路由至 Access Manager 2。
4. 更新 Directory Server 1 的目录结构和模式。
如第 254 页中的步骤 7 所述使用 amupgrade 脚本。当 Directory Server 1 的模式更新后，Access Manager 1 到 n 将继续发挥作用。

仅升级发行版本 4 Access Manager SDK

在一些部署体系结构中，Access Manager SDK 组件被安装在一个或多个计算机系统上，而在这些计算机上却未安装其他 Access Manager 组件。Access Manager SDK 相当于 Access Manager 的远程接口，且必须将其操作模式重新配置为与 Access Manager 相同：传统或领域。

Access Manager SDK 以及将 SDK 用作远程接口的整个 Access Manager 都应当升级到发行版本 5。但是，发行版本 5 Access Manager 与发行版本 4 Access Manager SDK 向后兼容，因此，通常应先升级 Access Manager，然后再升级其他计算机上的 Access Manager SDK。

作为 Access Manager 的远程接口，SDK 无需配置便能访问 Directory Server。如果正在使用 Access Manager SDK 来支持 Web 组件（如依赖于 Web 容器服务的 Portal Server），则必须为相应的 Web 容器配置 Access Manager SDK。但 Access Manager SDK 也能支持非 Web 组件，此时无需 Web 容器。

基于上述特点，升级 Access Manager SDK 的过程是 Access Manager 完全升级过程的一部分。

本节介绍如何从 Java ES 发行版本 4 到 Java ES 发行版本 5 执行仅限于 Access Manager SDK 的升级：

- [升级前的任务](#)
- [升级发行版本 4 Access Manager SDK](#)
- [检验 Access Manager SDK 升级](#)
- [升级回滚](#)

升级前的任务

Access Manager SDK 的升级前任务与 Access Manager 完全升级的升级前任务相同（参见第 243 页中的“升级前的任务”），但不包括那些与 Directory Server 和 Access Manager 管理工具 JSP 自定义相关的任务。Access Manager SDK 升级前需完成的任务如下：

- [第 244 页中的“升级 Access Manager 依赖项”](#)

但对于 Access Manager SDK 而言，对 Directory Server 没有依赖性，且仅当 Access Manager SDK 在 Web 容器中运行时才对 Web 容器软件具有依赖性。

- [第 244 页中的“备份发行版本 4 Access Manager 配置信息”](#)

但对于 Access Manager SDK 而言，仅当 Access Manager SDK 在 Web 容器中运行的情况下才需备份 Web 容器配置文件。

- [第 245 页中的“备份发行版本 4 Access Manager 日志和调试文件”](#)

要访问这些文件，还需获得管理员用户名和密码。

升级发行版本 4 Access Manager SDK

Access Manager SDK 的升级过程与 Access Manager 完全升级相同，但不包括那些与本地化、Access Manager 管理工具 JSP 自定义以及迁移目录模式相关的内容。

1. 删除 Java ES 发行版本 4 的 Access Manager SDK 版本。

按照第 248 页中的“删除 Java ES 发行版本 4 的 Access Manager 版本。”中的说明操作，只不过删除的是 Access Manager SDK。

2. 安装 Java ES 发行版本 5 的 Access Manager SDK 版本。

按照第 249 页中的“安装 Java ES 发行版本 5 的 Access Manager 版本。”中的说明操作，只不过安装的是 Access Manager SDK。

3. 重新配置 Access Manager SDK。

按照第 250 页中的“取消部署 Access Manager，重新配置并重新部署到 Web 容器中。”中的说明操作，不过要如下所示设置 `DEPLOY_LEVEL` 参数：

- 如果为 Web 容器配置了 Access Manager SDK：
`DEPLOY_LEVEL=4`（升级 SDK 并配置 Web 容器）
- 如果没有为 Web 容器配置 Access Manager SDK：
`DEPLOY_LEVEL=3`（仅升级 SDK）

检验 Access Manager SDK 升级

检验 Access Manager SDK 升级是否成功的方法有三种：

- 运行 Portal Server 或其他使用 Access Manager SDK 的组件以与 Access Manager 连接，并检查验证是否有效。
- 运行以下位置中提供的 Access Manager SDK 示例：

AccessManager-base/samples/sdk

- 检查 `com.ipplanet.am.version` 属性的值，该属性在 `AMConfig.properties` 文件中：

AccessManagerConfig-base/config/AMConfig.properties

升级回滚

没有提供用于将 Access Manager 回滚至其升级前的状态的脚本，必须使用在执行升级前的任务过程中所备份的 Access Manager 数据来手动执行该过程（参见第 245 页中的“备份发行版本 4 Access Manager 日志和调试文件”）。升级回退过于困难，以至于无法实现。

一种回滚方法是重新安装发行版本 4 并将所有备份的配置文件迁移至它们的适当位置。另外一种方法是在尝试进行升级之前，先创建一个并行系统，然后使用备份的配置文件升级并测试该并行系统。

从 Java ES 发行版本 3 升级 Access Manager

除了执行多实例升级的方法以外，将 Java ES 2003Q1（发行版本 3）Access Manager 或 Access Manager SDK 升级到发行版本 5 的过程与将发行版本 4 Access Manager 或 Access Manager SDK 升级到发行版本 5 的过程类似。

- [发行版本 3 Access Manager 的升级](#)
- [多实例升级](#)

注 发行版本 5 Access Manager 与发行版本 3 的某些 Sun Java Communications Suite 组件不兼容。如果 Access Manager 已升级到发行版本 5，则必须将发行版本 3 或早期 Delegated Administrator 也升级到发行版本 5，以便为 Messaging Server 和 Calendar Server 置备用户。但 Messaging Server 和 Calendar Server 本身则不必一定升级到发行版本 5。

发行版本 3 Access Manager 的升级

要将发行版本 3 Access Manager 或 Access Manager SDK 升级到发行版本 5，请按第 242 页中的“[从 Java ES 发行版本 4 升级 Access Manager](#)”中的说明操作，只不过要在提到发行版本 4 的地方替换为发行版本 3。

多实例升级

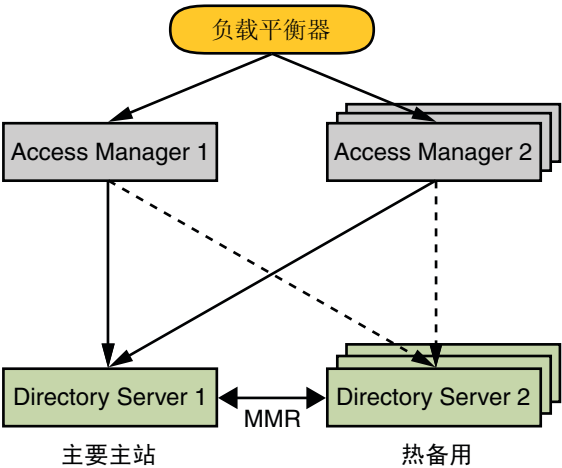
在一些部署体系结构中，Access Manager 被部署在多个计算机系统上，以提供高可用性和高可伸缩性。

用户常常希望在不中断服务的情况下按顺序升级多个 Access Manager 实例。本节将讨论如何执行从发行版本 3 Access Manager 至发行版本 5 的滚动升级过程。

注 当前发行版本不支持升级那些安装在同一主机系统上的多个 Access Manager 实例。如果在同一主机上有多个实例，在升级主实例之后，必须重新创建附加实例。

下图所示的部署体系结构将用于说明滚动升级过程。

图 14-2 多 Access Manager 实例的部署体系结构示例



在此体系结构中，通过负载平衡器访问多个 Access Manager 实例，而这些实例又将访问为多主复制 (multi-master replication, MMR) 设置的目录。尽管可以使用其他 Directory Server 复制模式，但 MMR 是具有高可用性和高可伸缩性的目录服务。在图 14-2 中，对 Access Manager 和 Directory Server 的多个实例进行了编组以便于解释升级过程。例如，Access Manager 2 代表 Access Manager 的第 2 个到第 n 个实例。

从发行版本 3 Access Manager 至发行版本 5 的滚动升级过程基于以下约束：发行版本 5 Access Manager 不能与发行版本 3 目录结构共存。但是，如果如图 14-2 中所示复制了 Directory Server 实例，则可以按照以下过程执行滚动升级：

1. 在所有 Access Manager 实例上备份发行版本 3 配置信息。
参见第 238 页中的表 14-3。
2. 修改 Access Manager 1 的配置。
 - a. 将 Access Manager 1 配置为指向 Directory Server 2 而不是 Directory Server 1。
 - b. 重新启动 Access Manager 1。

Access Manager 1 将继续处理请求，而 Access Manager 2 到 Access Manager n 将被升级。

3. 将 Access Manager 2 到 Access Manager n 升级。

为简洁起见，在后续步骤中，“Access Manager 2”表示 Access Manager 2 到 Access Manager n。

a. 在负载均衡器中禁用 Access Manager 2。

请求不会再路由至 Access Manager 2。

b. 部分升级 Access Manager 2。

如第 246 页中的“升级发行版本 4 Access Manager”中所述升级 Access Manager 的每个实例，但不要按第 254 页中的步骤 7 中所述更新目录结构和模式。

c. 禁用 Directory Server MMR。

d. 更新 Directory Server 1 的目录结构和模式。

如第 254 页中的步骤 7 所述使用 amupgrade 脚本。由于 Directory Server 2 的模式尚未更新，Access Manager 1 将继续发挥作用。

e. 重新启动 Access Manager 2。

f. 在负载均衡器中启用 Access Manager 2。

请求将重新路由至 Access Manager 2。

4. 升级 Access Manager 1。

a. 在负载均衡器中禁用 Access Manager 1。

请求不会再路由至 Access Manager 1。

b. 部分升级 Access Manager 1。

如第 246 页中的“升级发行版本 4 Access Manager”中所述升级 Access Manager，但不要按第 254 页中的步骤 7 中所述更新目录结构和模式。

c. 启用 Directory Server MMR。

现在，Directory Server 2 的模式（和数据）已经更新。

d. 将 Access Manager 1 的配置恢复为指向 Directory Server 1。

e. 重新启动 Access Manager 1。

f. 在负载均衡器中启用 Access Manager 1。

请求将重新路由至 Access Manager 1 以及其他所有已升级的 Access Manager 实例。

从 Java ES 发行版本 2 升级 Access Manager

如以下各节所示，将 Java ES 2004Q2（发行版本 2）Access Manager 升级到发行版本 5 的过程与将发行版本 4 Access Manager 升级到发行版本 5 的过程类似，只有少许差别：

- [升级前的任务](#)
- [发行版本 2 Access Manager 的升级](#)
- [多实例升级](#)

同样，将 Java ES 2004Q2（发行版本 2）Access Manager SDK 升级到发行版本 5 的过程与将发行版本 4 Access Manager SDK 升级到发行版本 5 的过程类似（参见第 259 页中的“[仅升级发行版本 4 Access Manager SDK](#)”），其例外情况也类似。仅 Access Manager SDK 升级不包括与本地化、Access Manager 管理工具 JSP 自定义以及迁移目录模式相关的过程。

发行版本 2 Access Manager SDK 以及将 SDK 用作远程接口的整个发行版本 2 Access Manager 必须全部升级到发行版本 5。不支持发行版本 2 组件与发行版本 5 组件并存。因此，必须将所有计算机上的所有发行版本 2 Access Manager 实例和所有发行版本 2 Access Manager SDK 实例全部升级到发行版本 5。

注 如果要在 Linux 平台上从发行版本 2 Access Manager 升级，则必须执行双重升级，即 Access Manager 和操作系统都要升级（RHEL 2.1 上不支持发行版本 5 Access Manager）。有关更多信息，参见第 241 页中的“[双重升级](#)”。

升级前的任务

在升级 Access Manager 之前，请执行第 243 页中的“[升级前的任务](#)”中所述的过程，但存在以下例外和补充：

- [升级 Access Manager 依赖项](#)
- [升级目录模式](#)
- [为目录重新建立索引](#)

升级 Access Manager 依赖项

与从发行版本 4 升级相比，从发行版本 2 到发行版本 5 的升级前任务需要将 Access Manager 所依赖的所有共享组件（参见第 43 页中的表 1-9）和所有本地驻留的产品组件均升级到发行版本 5。

升级 Access Manager 依赖项时，它们应按以下顺序升级，并且应在升级 Access Manager 之前进行。您可以跳过任何可能已完成升级的依赖项。

1. **共享组件。** 第 59 页中的第 2 章“升级 Java ES 共享组件”中提供了将 Java ES 共享组件同步到发行版本 5 的说明。不过，在执行发行版本 5 Access Manager 的全新安装时，安装程序会自动升级 Java ES 共享组件。
2. **Directory Server。** Directory Server 很少与 Access Manager 驻留在同一台计算机上，但是，第 103 页中的“从 Java ES 发行版本 2 升级 Directory Server”中提供了将 Directory Server 升级到发行版本 5 的说明。
3. **Web 容器软件。** 第 139 页中的“从 Java ES 发行版本 3 升级 Web Server”和第 198 页中的“从 Java ES 发行版本 2 升级 Application Server”中分别提供了升级 Web Server 和 Application Server 的说明。

升级目录模式

如果已用 Sun Java Communications Suite 的 Directory Preparation Tool (comm_dssetup.pl) 对 Directory Server 进行了配置，以支持 Communication Suite 组件（如 Messaging Server 和 Calendar Server），则必须先使用 Directory Preparation Tool 6.4 升级目录模式，然后再升级 Access Manager（参见《Sun Java Communications Suite 5 Upgrade Guide》(<http://docs.sun.com/doc/819-7561>)）。请在 Access Manager 依赖项升级完毕后执行这一升级前任务。

为目录重新建立索引

为了避免在完成目录模式的升级后执行 Access Manager 升级时更加复杂（参见上面的“升级目录模式”），需要为 Access Manager 目录根后缀手动重新建立索引，如下所示：

发行版本 2 - 发行版本 4 Directory Server:

1. `cd serverRoot/slapd-`hostname``
2. `./db2index.pl -D "cn=directory manager" -w passwordFile -n databaseName`

其中，默认的 *databaseName* 为 *userRoot*。

发行版本 5 Directory Server:

1. `cd DirServer-base/ds6/bin`
2. `./dsconf reindex -D "cn=Directory Manager" -e -w passwordFile suffix`

其中

-e 表示非安全连接

-D 为目录管理员

-w 为仅包含密码的密码文件

suffix 为 Access Manager 目录根后缀。

重新建立索引可能需要大量的时间，这取决于目录中的条目数。

发行版本 2 Access Manager 的升级

将 Access Manager 从发行版本 2 升级到发行版本 5 的过程视您在其中部署 Access Manager 软件的 Web 容器而定。

升级发行版本 2 Access Manager: Web Server Web 容器

当部署到 Web Server Web 容器中时，要将发行版本 2 Access Manager 升级到发行版本 5，请按第 246 页中的“升级发行版本 4 Access Manager”中的说明操作，只不过要在提到发行版本 4 的地方替换为发行版本 2。

升级发行版本 2 Access Manager: Application Server Web 容器

当部署到 Application Server Web 容器中时，要将发行版本 2 Access Manager 升级到发行版本 5，需分两种情况：

- **发行版本 5 Application Server 是新安装的。**这种情况下，要将发行版本 2 Access Manager 升级到发行版本 5，请按照第 246 页中的“升级发行版本 4 Access Manager”中的说明操作，只不过要在提到发行版本 4 的地方替换为发行版本 2。
- **发行版本 2 Application Server 已升级到发行版本 5。**在这种情况下，最初在其中部署 Access Manager 的发行版本 2 Application Server 实例 (instanceName) 会在升级到发行版本 5 之后，被迁移到升级过程所创建的一个节点代理下。对于此已升级 Application Server 实例中的 Access Manager 的升级，需要执行以下各节中的步骤：

升级摘要

升级 Access Manager 的过程由下列步骤组成：

- 1. 升级 Access Manager mobile access 软件。
- 2. 删除 Java ES 发行版本 2 的 Access Manager 版本。使用 ampre71upgrade 脚本。
- 3. 如果到发行版本 5 的升级需要实现本地化，请删除发行版本 2 本地化软件包。必须手动执行此步骤。
- 4. 安装 Java ES 发行版本 5 的 Access Manager 版本。使用 Java ES 安装程序并选择“以后再配置”选项。
- 5. 重新自定义 Access Manager 的 JSP。
- 6. 检查 Directory Server 是否正在运行。
- 7. 启动以下 Application Server 实例：Domain Administration Server (DAS)、节点代理以及在其中部署 Access Manager 的服务器实例。
- 8. 取消部署 Access Manager，重新配置并重新部署到 Application Server 实例中。使用 amconfig 脚本。
- 9. 检验 Access Manager 的 classpath-suffix 和 server-classpath 信息是否已迁移至发行版本 5 Application Server 的 domain.xml 文件。
- 10. 停止 Domain Administration Server (DAS)、节点代理和服务器实例。
- 11. 重新启动 Domain Administration Server (DAS)、节点代理和服务器实例。
- 12. 更新目录结构和模式。使用 amupgrade 脚本。

以下过程对上述步骤进行了说明。

升级过程

- 1. 升级 Access Manager mobile access 软件。

Access Manager mobile access 软件需要通过修补发行版本 2 的版本进行升级。下表显示了所需的修补程序：

表 14-13 用于升级 Access Manager Mobile Access 软件的修补程序¹

说明	修补程序 ID: Solaris 9 和 10	修补程序 ID: Linux
Mobile Access 软件	119530-05 (SPARC)	119532-05
	119531-05 (x86)	• sun-identity-mobileaccess-6.2-25.3.i386.rpm • sun-identity-mobileaccess-config-6.2-25.3.i386.rpm

1. 修补程序修订版本号是升级到 Java ES 发行版本 5 所需的最低版本号。如果有较新的修订版本可用，请用较新版本替代该表中所示的版本。

- a. 利用表 14-6 中的修补程序号获取所需的修补程序。
可从以下网址将修补程序下载到 /tmp:
<http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access>
- b. 执行修补程序 README 文件中指示的任何修补前过程。
- c. 获取修补程序将要请求的以下参数的值:

表 14-14 Mobile Access 修补程序参数

参数	值
目录管理员的 DN	默认值: cn=Directory Manager
目录管理员的密码	

- d. 应用表 14-6 中的修补程序。

在 Solaris 上:
`patchadd patch_ID`

在 Linux 上:
`./update`

执行修补程序 README 文件中指示的任何修补后过程。

- 2. 删除 Java ES 发行版本 2 的 Access Manager 版本。
 - a. 以 root 身份登录到托管发行版本 4 Access Manager 的计算机或成为超级用户。

`su -`
 - b. 将目录更改为 Java ES 发行版本 5 分发软件中的 `os_arch/Product/identity_svr/Tools` 目录, 其中, `os_arch` 与您的平台匹配, 如 `Solaris_sparc`。
 - c. 获取 `ampre71upgrade` 脚本将要请求的以下参数的值:

表 14-15 Access Manager 配置参数: ampre71upgrade

参数	值
Directory Server 主机	设置全限定名称: <i>hostname.domian</i>
Directory Server 端口	指定一个非 SSL 端口号 ¹ 默认值: 389
顶级管理员 DN	默认值: <i>uid=amadmin,ou=People,default_org_DN</i>
顶级管理员密码	
存储备份文件的目录	默认值: <i>AccessManager-base</i>

1. 必须指定一个与默认 SSL 值 636 不同的 Directory Server SSL 端口。

- d. 确保 Directory Server 正在运行，否则请将其启动。
- e. 运行 ampre71upgrade 脚本。

`./ampre71upgrade`

该脚本将备份 Access Manager 配置文件，并删除发行版本 4 基本软件包（对于本地化软件包，必须按照下面的步骤 3 手动删除）。

- 3. 如果到发行版本 5 的升级需要实现本地化，请删除发行版本 2 本地化软件包。

在上面步骤 2 中运行的 ampre71upgrade 脚本并不删除本地化软件包，因此，您必须如下所示手动将其删除。

在 Solaris 上：

- a. 检查是否存在本地化软件包。

```
pkginfo | grep SUNWaml
pkginfo | grep SUNWamclnt
pkginfo | grep SUNWamdistauth
```

- b. 删除在上述步骤 a 中找到的任何本地化软件包。

```
pkgrm SUNWamlLocale
pkgrm SUNWamclntLocale
pkgrm SUNWamdistauthLocale
```

在 Linux 上:

- a. 检查是否存在本地化 RPM。

```
rpm -qa | grep sun-identity-sdk-*
rpm -qa | grep sun-identity-clientsdk-*
rpm -qa | grep sun-identity-distauth-*
```

- b. 删除在上述步骤 a 中找到的任何本地化 RPM。

```
rpm -e sun-identity-sdk-Locale-*
rpm -e sun-identity-clientsdk-Locale-*
rpm -e sun-identity-distauth-Locale-*
```

4. 安装 Java ES 发行版本 5 的 Access Manager 版本。

执行以下步骤:

- a. 在托管发行版本 2 Access Manager 的计算机上启动 Java ES 安装程序。

```
cd Java ES Release 5 distribution/os_arch
./installer
```

其中, *os_arch* 与您的平台匹配, 如 *Solaris_sparc*。(对于命令行界面, 使用 *installer -nodisplay* 选项。)

显示“欢迎”和“许可协议”页面之后, 将向您显示一个组件选择页面。
(如果检测到可由 Java ES 安装程序直接升级的已安装组件, 它们的状态会显示为“可升级”。)

- b. 从组件选择页面中选择 Access Manager。
 - c. 指定安装目录, 需与发行版本 2 的安装目录相同。
 - d. 选择“以后再配置”选项。
 - e. 如果需要, 选择安装本地化软件包的选项。
 - f. 安装完成后退出 Java ES 安装程序。
5. 重新自定义 Access Manager 的 JSP。

重新应用按第 245 页中的“备份 Web 容器自定义文件”中所述保存的、用于 Access Manager 控制台和验证用户界面 (User Interface, UI) 的发行版本 2 自定义 JSP 设置。

然后将自定义 JSP 文件复制到正确的目录:

- o 仅传统 Access Manager 控制台
AccessManager-base/web-src/applications/console

- 验证 UI: *AccessManager-base/web-src/services/config/auth/default* 或 *AccessManager-base/web-src/services/config/auth/default_Locale* (其中 *Locale* 是语言环境指示符, 如 *ja*)

有关更多信息, 参见《Sun Java System Access Manager 7.1 Developer's Guide》(<http://docs.sun.com/doc/819-4675>)。

6. 检查 Directory Server 是否正在运行。

7. 启动以下 Application Server 实例:

在以下命令及后续步骤中, 将使用下面的约定:

- 其中, *nodeagentName* 的格式为 *hostName_domainName*, 但默认情况下只是 *hostName*
- 默认的 *domainName* 为 *domain1*
- 默认的 *instanceName* 为 *server1*

注 确保在启动服务器实例之前使用 `startinstances=false` 选项单独启动节点代理, 如下所示。

a. 启动 Domain Administration Server (DAS)

```
AppServer8-base/bin/asadmin start-domain --user admin_ID  
domainName
```

b. 启动其下迁移了已升级 Application Server 实例的节点代理

```
AppServer8-base/bin/asadmin start-node-agent  
--startinstances=false --user admin_ID nodeagentName
```

c. 启动其中部署了 Access Manager 的服务器实例 (*instanceName*) (如果该服务器实例尚未运行)。

```
AppServer8-base/bin/asadmin start-instance --user admin_ID  
instanceName
```

8. 取消部署 Access Manager, 重新配置并重新部署到 Application Server 实例中。

a. 如果 Web 容器在 SSL 模式下运行, 请确保容器的 SSL 证书尚未过期并仍然有效。

b. 根据 `amsamplesilent` 模板输入文件创建一个 `amconfig` 输入文件:

```
cp amsamplesilent config-file
```

(在后续步骤中, 假定 *config-file* 与 `amsamplesilent` 驻留在同一目录中。)

c. 设置 *config-file* 中的配置参数。

需正确设置所有参数。一些参数的值可从 `AMConfig.properties` 文件迁移，相比之下，其他参数的值则是升级过程的专用值，如下表所示。

表 14-16 Access Manager 配置参数: `amconfig`

参数	值
升级参数	
DEPLOY_LEVEL	设置为 26 用于取消部署，或者 设置为 1 用于重新配置和部署
DIRECTORY_MODE	设置为 5
AM_REALM ¹	设置为 disabled，如果启用传统模式 设置为 enabled，如果启用领域模式 默认值: enabled
JAVA_HOME	设置为 JDK 发行版本 5 目录: <code>/usr/java/jdk1.5.0_04/</code>
WEB_CONTAINER	设置为 AS8，对于 Application Server 8.x 且仅填写 <i>config-file</i> 的对应部分。
AS81_INSTANCE (将自 Application Server 7.x 升级而来的 Application Server 8.x 用作 Web 容器)	设置为 Application Server 7.x <i>instanceName</i> ，其默认值 为 <code>server1</code>
AS81_INSTANCE_DIR (将 Application Server 8.x 用 作 Web 容器)	设置为实例的 Application Server 8.x 域目录，其默认 值为 <code>AppServer8Config-base/domains/domain1</code>
AS81_DOCS_DIR (将 Application Server 8.x 用 作 Web 容器)	设置为实例的 Application Server 8.x 文档根目录，其 默认值为 <code>AppServer8Config-base/domains/domain1/docroot</code>
迁移自 <code>AMConfig.properties</code>	
SERVER_PROTOCOL	<code>com.iplanet.am.server.protocol</code>
SERVER_PORT	<code>com.iplanet.am.server.port</code>
SERVER_HOST	<code>com.iplanet.am.server.host</code>
DS_HOST	<code>com.iplanet.am.directory.host</code>
DS_PORT	<code>com.iplanet.am.directory.port</code>
ROOT_SUFFIX ²	<code>com.iplanet.am.defaultOrg</code>
CONSOLE_DEPLOY_URI	<code>com.iplanet.am.console.deploymentDescriptor</code>
SERVER_DEPLOY_URI	<code>com.iplanet.am.services.deploymentDescriptor</code>

表 14-16 Access Manager 配置参数: amconfig (续)

参数	值
PASSWORD_DEPLOY_URI	com.sun.identity.password.deploymentDescriptor
AM_ENC_PWD ²	am.encryption.pwd ³

- 1. 有关领域模式和传统模式的更多信息，参见第 239 页中的“兼容性问题”。
- 2. 此参数的值应与先前版本 Access Manager 中的参数值相同。
- 3. 如果同时部署了 Access Manager 和 Access Manager SDK，则对于 Access Manager 实例及其关联的 Access Manager SDK 实例，此属性的值必须相同。

对于其他参数，如果不更改 Web 容器和密码，请提供在要升级的发行版本 2 配置中使用的那些值。

- d. 运行 amconfig 以取消部署 Access Manager。

将 config-file 中 DEPLOY_LEVEL 的值设置为 26。

```
cd /AccessManager-base/bin
./amconfig -s AccessManager-base/bin/config-file
```

- e. 检查以确保 Common Agent Container 正在运行。

```
netstat -an | grep 11163
```

如果 Common Agent Container 未运行，请将其启动。

```
/usr/sbin/cacoadm start
```

- f. 运行 amconfig 以重新配置 Access Manager 并将其部署到 Web 容器中。

将 config-file 中 DEPLOY_LEVEL 的值设置为 1。

```
cd /AccessManager-base/bin
./amconfig -s AccessManager-base/bin/config-file
```

- 9. 检验 Access Manager 的 classpath-suffix 和 server-classpath 信息是否已迁移至发行版本 5 Application Server 的 domain.xml 文件。

- a. 记录 Access Manager 的 classpath-suffix 和 server-classpath 信息，它们位于最初部署 Access Manager 的发行版本 2 Application Server 实例的 server.xml 文件中：

```
AppServer7Config-base/domains/domainName/instanceName/config/server.xml
```

- b. 检查 classpath-suffix 和 server-classpath 条目是否已附加到其中部署了 Access Manager 的已升级 Application Server 实例的 domain.xml 文件：

```
AppServer8Config-base/nodeagents/nodeagentName/instanceName/  
config/domain.xml
```

应将类路径信息添加到发行版本 5 Application Server domain.xml 文件的 instanceName-config 块中。该块的起始行是：

```
<config dynamic-reconfiguration-enabled="true"  
name="instanceName-config">
```

10. 更新目录结构和模式。

发行版本 5 Access Manager 可与发行版本 4 的目录结构共存，但必须对该结构进行修改以使其支持发行版本 5 的功能。通过运行 amupgrade 脚本，将 Access Manager 目录结构和模式更新为发行版本 5，该脚本安装在以下目录中：

- o 在 Solaris 上：
AccessManager-base/upgrade/scripts
- o 在 Linux 上：
AccessManager_base/identity/upgrade/scripts
- a. 获取 amupgrade 脚本将要请求的以下参数的值：

表 14-17 Access Manager 配置参数：amupgrade

参数	值
Directory Server 主机	设置全限定名称：hostname.domian
Directory Server 端口	指定一个非 SSL 端口号 ¹ 默认值：389
目录管理员 DN	默认值：cn=Directory Manager
目录管理员密码	
顶级管理员 DN	默认值： uid=amadmin,ou=People,default_org_DN
顶级管理员密码	
启用领域模式 (从发行版本 4 领域模式升级时不请求此参 数值。)	Y/N: Yes 表明“领域模式”已启用且服务数 据已迁移到新的“领域”树。No (默认值) 表明服务数据仍然处于“传统模式”下。

1. 如果指定一个 Directory Server SSL 端口（如默认的 SSL 值 636），则无法成功完成升级过程。

- b. 运行 amupgrade 脚本。

```
cd AccessManager-base/upgrade/scripts
./amupgrade
```

如果升级成功，脚本将显示 "Upgrade completed"。

- c. 查看以下升级日志文件以了解有关目录模式扩展的信息：

在 Solaris 上：

```
/var/sadm/install/logs/
    Sun_Java_System_Access_Manager_upgrade_dit_log.mmddhhmm
```

在 Linux 上：

```
/var/log/Sun_Java_System_Access_Manager_upgrade_dit_log.mmddhhmm
```

11. 停止 Domain Administration Server (DAS)、节点代理和服务器实例。

它们是在第 272 页中的步骤 7 中启动的实例。

```
AppServer8-base/bin/asadmin stop-domain --user admin_ID
    domainName
```

```
AppServer8-base/bin/asadmin stop-node-agent --user admin_ID
    nodeagentName
```

12. 重新启动 Domain Administration Server (DAS)、节点代理和服务器实例。

注 确保在启动服务器实例之前使用 `startinstances=false` 选项单独启动节点代理，如下所示。

```
AppServer8-base/bin/asadmin start-domain --user admin_ID
    domainName
```

```
AppServer8-base/bin/asadmin start-node-agent --port DASportNumber
    --startinstances=false --user admin_ID --password password
    nodeagentName
```

```
AppServer8-base/bin/asadmin start-instance --port DASportNumber
    --user admin_ID --password password instanceName
```

DASportNumber 的默认值为 4848。

检验 Access Manager 升级

完成升级过程后，可按第 255 页中的“检验 Access Manager 升级”中所述检验升级是否成功。

升级后的任务

如果要使用安全声明标记语言 (Security Assertion Markup Language, SAML) 服务，必须使用 Access Manager 控制台添加并启用一个 SAML 验证模块。有关如何创建 SAML 验证模块实例的信息，参阅《Sun Java System Access Manager 7.1 管理指南》(<http://docs.sun.com/doc/820-0839>)。

升级回滚

未提供用于将 Access Manager 回滚至其升级前的状态的脚本。必须使用在执行升级前的任务过程中所备份的 Access Manager 数据手动执行该过程（参见第 245 页中的“备份发行版本 4 Access Manager 日志和调试文件”）。回滚过程过于困难以至于无法实现。

多实例升级

在一些部署体系结构中，Access Manager 被部署在多个计算机系统上，以提供高可用性和高可伸缩性。

用户常常希望在不中断服务的情况下按顺序升级 Access Manager 实例。从发行版本 2 Access Manager 至发行版本 5 的滚动升级过程基于以下约束：发行版本 5 Access Manager 不能与发行版本 2 目录结构共存。但是，如果如图 14-2 中所示复制了 Directory Server 实例，则可以按照第 262 页中的“多实例升级”中所述执行滚动升级。

Portal Server

本章介绍如何将 Portal Server 升级到 Java ES 5（发行版本 5）：Sun Java System Portal Server 7.1。

本章概要介绍了有关发行版本 5 所支持的不同升级方式的升级注意事项。其内容涵盖了在 Solaris 和 Linux 操作系统上的升级：

- 第 281 页中的“Portal Server 升级概述”
- 第 287 页中的“从 Java ES 发行版本 4 升级 Portal Server”
- 第 313 页中的“从 Java ES 发行版本 3 升级 Portal Server”
- 第 316 页中的“从 Java ES 发行版本 2 升级 Portal Server”
- 第 326 页中的“从过渡功能版本 7.0 升级 Portal Server”

注 本章中的文件位置是相对于称为 *PortalServer6-base* 与 *PortalServer6Config-base* (Portal Server 6.x) 以及 *PortalServer7-base* 与 *PortalServer7Config-base* (Portal Server 7.x) 的目录路径指定的。在最初安装 Portal Server 时，可能至少已将这些路径的一部分指定为安装目录，否则，Java ES 安装程序会分配一个默认值。

下表说明了这些目录路径的默认值。

表 15-1 Portal Server 目录路径

路径名变量	Solaris OS	Linux OS
<i>PortalServer6-base</i>	/opt/SUNWps	/opt/sun/portal
<i>PortalServer6Config-base</i>	/etc/opt/SUNWps	/etc/opt/sun/portal
<i>PortalServer7-base</i>	/opt/SUNWportal	/opt/sun/portal
<i>PortalServer7Config-base</i>	/etc/opt/SUNWportal	/etc/opt/sun/portal
<i>PortalServer7Data-base</i>	/var/opt/SUNWportal	/var/opt/sun/portal

Portal Server 升级概述

本节将在以下方面对 Portal Server 进行介绍，这些方面会对升级到 Java ES 5（发行版本 5）产生影响：

- [关于 Java ES 发行版本 5 Portal Server](#)
- [Portal Server 升级指导说明](#)
- [Portal Server 数据](#)
- [Portal Server 升级策略](#)

关于 Java ES 发行版本 5 Portal Server

相对于发行版本 4 而言，Java ES 发行版本 5 Portal Server 是一个主要发行版本，包含了许多增强功能和新增功能。其中的许多更改都是继发行版本 4 之后在过渡功能版本 (Interim Feature Release, IFR) 中做出的。相对于 IFR，发行版本 5 只进行了较少的功能更改。有关 IFR 增强功能和新增功能的信息，参见《Sun Java System Portal Server 7.1 发行说明》(<http://docs.sun.com/app/docs/doc/820-0897/6ncjtcci5?a=view>)。需特别指出的是，发行版本 4 命令行管理界面已替换为 psadmin 命令。

Portal Server 升级指导说明

表 15-2 显示了所支持的、将 Portal Server 升级到 Java ES 发行版本 5 的升级方式。此表既适用于 Solaris，也适用于 Linux 操作系统。

表 15-2 升级到 Java ES 5（发行版本 5）：Portal Server 7.1 的方式

Java ES 发行版本	Portal Server 版本	常规方法	必需的重新配置
过渡功能版本 (Interim Feature Release, IFR)	Sun Java System Portal Server IFR 7.0 2005Q4	直接升级： 通过应用修补程序然后使用升级脚本来执行。	自定义内容需要手动重新应用。
发行版本 4	Sun Java System Portal Server 6.3.1 2005Q4	直接升级： 使用升级脚本执行。	自定义内容需要手动重新应用。
发行版本 3	Sun Java System Portal Server 6.3.1 2005Q1	直接升级： 使用升级脚本执行。	自定义内容需要手动重新应用。

表 15-2 升级到 Java ES 5（发行版本 5）： Portal Server 7.1 的方式（续）

Java ES 发行版本	Portal Server 版本	常规方法	必需的重新配置
发行版本 2	Sun Java System Portal Server 6.3 2004Q2	直接升级： 使用升级脚本执行。	自定义内容需要手动重新应用。
发行版本 1	Sun ONE Portal Server 6.2 (2003Q4)	不可直接升级： 但可通过先升级到发行版本 3 然后从发行版本 3 升级到发行版本 5 来执行。	配置数据
Java ES 以前的发行版本		不可直接升级。	

Portal Server 数据

下表显示了可能受 Portal Server 软件升级影响的数据类型。

表 15-3 Portal Server 数据用途

数据类型	位置	用途
配置数据	<i>PortalServer6Config-base/</i>	Portal Server 的配置。
Web 容器访问控制和配置文件	Web Server 7.0（Java ES 发行版本 5） server.policy 和 server.xml 文件，位于 <i>WebServer7Config-base/https-configName/config</i> Web Server 6.x（Java ES 发行版本 2、3 和 4） server.policy 和 server.xml 文件，位于 <i>WebServer6-base/https-hostname/config</i> Application Server 8.x（Java ES 发行版本 3、4 和 5）： server.policy 和 domain.xml 文件，位于 <i>AppServer8Config-base/domains/domainName/config</i> Application Server 7.x（Java ES 发行版本 2）： server.policy 和 server.xml 文件，位于 <i>AppServer7Config-base/domains/domainName/config</i>	Portal Server Web 容器实例的配置。
自定义数据	<i>PortalServer6Config-base/desktop</i>	自定义模块的 JAR 文件 自定义的 Portal Server 桌面样例
目录模式 服务配置 用户数据	Directory Server	Portal Server 依赖于服务配置（如门户桌面）和存储在目录中的用户配置文件数据。
动态应用程序数据	无	Portal Server 并不永久存储应用程序数据（如会话状态）。

Portal Server 升级策略

Portal Server 的升级策略通常取决于在第 1 章“升级规划”中讨论的许多注意事项：升级方式、Java ES 组件之间的依赖性、选择性升级与全部升级、多实例部署，等等。

本节通过提出可能影响 Portal Server 升级计划的问题，对 Portal Server 的一般注意事项进行逐一介绍。

兼容性问题

发行版本 5 Portal Server 在用于管理 Portal Server 和 Portal Server Secure Remote Access 组件的 psadmin 命令中引入了公共接口更改。参见《Sun Java System Portal Server 7.1 Command-Line Reference》(<http://docs.sun.com/doc/819-5030>)。

因此，发行版本 5 Portal Server 与早期版本或早期版本的 Portal Server Secure Remote Access 组件（包括 SRA Gateway、Rewriter Proxy 和 Netlet Proxy）不向后兼容，但过渡时期除外，在这一时期，多实例部署正在进行滚动升级。所有 Portal Server 实例及 Portal Server Secure Remote Access 组件实例都需要在 Java ES 发行版本 5 同步。

并且，个别 Portal Server 组件（包括 mobile access 组件）与早期版本不向后兼容，需要将它们全部同步到 Java ES 发行版本 5。

此外，发行版本 5 Portal Server 与早期版本 Portal Server 所使用的 Directory Server 数据结构之间存在着不兼容性。这种不兼容性影响了使用相同 Directory Server 数据对多个 Portal Server 实例执行的滚动升级。

Portal Server 组件依赖性

Portal Server 对其他 Java ES 组件的依赖性会影响升级和重新配置 Portal Server 软件的过程。例如，Portal Server 接口或功能上的变化可能要求 Portal Server 所依赖组件的升级版本。是否需要升级此类组件取决于具体的升级方式。

Portal Server 对以下 Java ES 组件具有依赖性：

- **共享组件。** Portal Server 对特定的 Java ES 共享组件具有依赖性（参见第 43 页中的表 1-9）。
- **Web 容器。** Portal Server 对 Web 容器服务具有强制依赖性，这些服务可由 Java ES Web Server、Java ES Application Server 或来自 Weblogic 和 WebSphere 的第三方 Web 容器提供。

注	对于在第三方 Web 容器中的部署，不支持将 Portal Server 升级到发行版本 5。对于在来自 Weblogic 和 WebSphere 的 Web 容器中的部署，必须执行发行版本 5 Portal Server 的全新安装。
----------	--

- **Access Manager（或 Access Manager SDK）。** Portal Server 对 Access Manager 具有强制依赖性，以便为最终用户提供验证和授权服务（包括单点登录）。如果 Access Manager 运行在远程计算机上，则 Access Manager SDK 必须在本地可用。
- **Directory Server。** Portal Server 对 Directory Server 具有强制依赖性，后者用于存储通过 Access Manager 访问的用户数据。因此，Portal Server 升级可能需要扩展目录模式。
- **Portal Server Secure Remote Access。** Portal Server 对 Portal Server Secure Remote Access 具有可选依赖性，后者通过 Gateway、Rewriter Proxy 和 Netlet Proxy 组件提供安全的远程访问。
- **Java DB。** Portal Server 对 Java DB 具有可选依赖性，后者为多个 portlet 应用程序提供支持。
- **Service Registry。** Portal Server 对 Service Registry 具有强制依赖性，后者提供执行编译操作所需的库。
- **Communications Express。** Portal Server 对 Communications Express 具有可选依赖性，后者是一个 Sun Java Communications Suite 组件，用于向最终用户提供消息传送和日历频道。Communications Express 不再是 Java ES 产品组件。

选择性升级问题

一般而言，Java ES 发行版本 5 支持对一台计算机上的所有组件执行选择性升级，但由于 Portal Server 所依赖的其他 Java ES 组件过多，因此，验证不同 Java ES 发行版本的任意组件组合是非常困难的。

因此，Portal Server 支持一组与 Access Manager 和 Web 容器相关的受限升级方案。

- **如果要从 Java ES 发行版本 4 升级 Portal Server。** 可在升级 Portal Server 之前将 Directory Server、Access Manager 和 Web 容器（Web Server 或 Application Server）均升级到发行版本 5，也可以只将 Portal Server 升级到发行版本 5（将其他组件保留为发行版本 4），但不能将一些依赖项保留为发行版本 4 而将另一些依赖项升级到发行版本 5。
- **如果要从 Java ES 发行版本 3 升级 Portal Server。** 必须在升级 Portal Server 之前将 Directory Server、Access Manager 和 Web 容器（Web Server 或 Application Server）均升级到发行版本 4 或发行版本 5，但不能将任何依赖项保留为发行版本 3，也不能将一些依赖项升级到发行版本 4 而将另一些依赖项升级到发行版本 5。
- **如果要从 Java ES 发行版本 2 升级 Portal Server。** 必须在升级 Portal Server 之前将 Directory Server、Access Manager 和 Web 容器（Web Server 或 Application Server）均升级到发行版本 4 或发行版本 5，但不能将任何依赖项保留为发行版本 2，也不能将一些依赖项升级到发行版本 4 而将另一些依赖项升级到发行版本 5。

Web 容器升级方案

Portal Server 可部署在由 Web Server 或 Application Server 提供的 Web 容器中。因此，Portal Server 到发行版本 5 的升级可能会复杂化，因为可能还需要将用于部署该组件的 Web 容器升级到发行版本 5。关于这一点，存在多种可能的 Web 容器升级方案，将在下表中一一列出。

表 15-4 用于 Portal Server 升级的 Web 容器升级方案

方案	最初部署 Portal Server 的 Web 容器	升级后部署 Portal Server 的 Web 容器	适用的 Portal Server 升级方式：升级始于
方案 1	Web Server 6.x	Web Server 6.x	发行版本 2 发行版本 3 发行版本 4 IFR 7.0
方案 2	Web Server 6.x	Web Server 7.0	发行版本 2 发行版本 3 发行版本 4
方案 3	Application Server 8.1	Application Server 8.1	发行版本 3 发行版本 4 IFR 7.0
方案 4	Application Server 8.1	Application Server 8.2	发行版本 3 发行版本 4 IFR 7.0
方案 5	Application Server 7x	Application Server 8.2	发行版本 2

升级 Portal Server（例如，使用 `psupgrade` 脚本时）以提供适合于表 15-4 中适用升级方案的值时，特别是存在 Web 容器的主要版本升级时，必须要小心。

双重升级

在双重升级中，Portal Server 和操作系统都要进行升级（如第 41 页中的“双重升级：Java ES 和操作系统软件”中所述）。可使用操作系统就地升级方法执行双重升级：

1. 备份现有 Portal Server 数据。
有关基本数据的位置，参见第 282 页中的“Portal Server 数据”。
2. 升级操作系统。
升级会将现有文件系统留在原来的位置。
3. 将 Portal Server 升级到发行版本 5。
参见本章中的相应小节，这要视升级方式而定。

从 Java ES 发行版本 4 升级 Portal Server

本节包含关于将 Portal Server 从 Java ES 2005Q4（发行版本 4）升级到 Java ES 5（发行版本 5）的信息。

注 对于 Portal Server 已部署于 Application Server Web 容器中且在升级到发行版本 5 之前已从发行版本 2 升级到发行版本 3 或 4 的特殊情况，本节不适用。当前不支持上述升级方式。

本节涵盖了以下主题：

- [简介](#)
- [发行版本 4 Portal Server 的升级](#)
- [多实例升级](#)

简介

将 Java ES 发行版本 4 Portal Server 升级到发行版本 5 时，请考虑升级过程中的以下方面：

- **常规升级方法。** 使用升级脚本 `psupgrade` 执行升级。该脚本会安装新的软件包、在必要时迁移配置数据、更新本地化文件并将 Portal Server Web 应用程序重新部署到 Web 容器中。
- **升级依赖性。** Portal Server 对许多 Java ES 共享组件都具有依赖性（参见[第 43 页中的表 1-9](#)）。尽管发行版本 5 Portal Server 与发行版本 4 版本的这些共享组件兼容，但仍需要升级共享组件，因为用于升级 Portal Server 的 `psupgrade` 脚本要求发行版本 5 版本的 ANT 共享组件。

发行版本 5 Portal Server 还依赖于 Web 容器、Access Manager 和 Directory Server，如[第 284 页中的“Portal Server 组件依赖性”](#)中所述。支持使用两种方法来升级这些依赖项（参见[第 285 页中的“选择性升级问题”](#)）：

- 发行版本 4 组件满足的所有依赖性（**没有一项**升级到发行版本 5）
- 发行版本 5 组件满足的所有依赖性（**全部**升级到发行版本 5）。
- **向后兼容性。** 发行版本 5 Portal Server 与发行版本 4 的版本不向后兼容。
- **升级回滚。** 从发行版本 5 Portal Server 到发行版本 4 的升级回滚包括恢复发行版本 4 软件包、恢复发行版本 4 目录数据以及将 Portal Server Web 应用程序重新部署到 Web 容器中。

- **平台问题。** 对于 Solaris 和 Linux 操作系统，升级 Portal Server 的常规方法是相同的，但是，发行版本 5 Portal Server 在 Solaris OS 上是安装在新路径中，而在 Linux OS 上是安装在原来的发行版本 4 路径中。

发行版本 4 Portal Server 的升级

本节介绍如何在 Solaris 和 Linux 平台上将 Portal Server 从 Java ES 发行版本 4 升级到 Java ES 发行版本 5。如果其中某个主题取决于具体平台的过程，则该主题将指明所适用的操作系统。本节包括以下主题：

- [发行版本 4 升级前的任务](#)
- [升级发行版本 4 Portal Server \(Solaris\)](#)
- [升级发行版本 4 Portal Server \(Linux\)](#)
- [检验升级](#)
- [发行版本 4 升级后的任务](#)
- [升级回滚 \(Solaris\)](#)
- [升级回滚 \(Linux\)](#)

发行版本 4 升级前的任务

升级 Portal Server 之前，应执行以下任务：

- [检验当前版本信息](#)
- [升级 Portal Server 依赖项](#)
- [获取所需的配置信息和密码](#)
- [备份发行版本 4 Portal Server 配置信息](#)
- [记录 Java 虚拟机 \(Java Virtual Machine, JVM\) 设置](#)
- [删除负载均衡器的配置](#)
- [删除 Directory Proxy Server 的配置](#)

检验当前版本信息

可使用以下命令检验 Portal Server 的当前版本：

```
PortalServer6-base/bin/version
```

表 15-5 Portal Server 版本检验输出

Java ES 发行版本	Portal Server 版本号
发行版本 2	6.3
发行版本 3	6.3.1
发行版本 4	6.3.1 ¹
IFR 发行版本	7.0
发行版本 5	7.1

1. 发行版本 3 与发行版本 4 的不同之处仅在于修补程序。可通过使用 `showrev -p | grep patch_ID` (Solaris) 命令和 `rpm -qa sun-portal-core` (Linux) 命令并将版本与 Java ES 发行版本 4 升级指南中列出的那些版本相比较来检查是否为发行版本 4 修补程序。

升级 Portal Server 依赖项

通常建议将计算机系统（以及计算环境）中的所有 Java ES 组件都升级到 Java ES 发行版本 5。

虽然发行版本 5 Portal Server 与发行版本 4 版本的 Java ES 共享组件兼容，但仍需要升级共享组件，因为用于升级 Portal Server 的 `psupgrade` 脚本要求发行版本 5 版本的 ANT 共享组件。

如果您选择将其中的任一 Portal Server 产品组件依赖项升级到发行版本 5，则需要将它们全部升级（参见第 285 页中的“选择性升级问题”）。在升级 Portal Server 之前，应该按以下顺序升级这些依赖项（跳过任何可能已升级的依赖项）。

- 1. **共享组件。** 第 59 页中的“升级 Java ES 共享组件”提供了将 Java ES 共享组件同步到发行版本 5 的说明。
- 2. **Directory Server。** 第 89 页中的第 5 章“Directory Server”提供了将 Directory Server 升级至发行版本 5 的说明。
- 3. **Web 容器软件。** 第 119 页中的第 7 章“Web Server”和第 183 页中的第 11 章“Application Server”分别提供了升级 Web Server 和 Application Server 的说明。

注 升级诸如来自 Weblogic 和 WebSphere 的第三方 Web 容器时可能导致 Portal Server 中断，因为在容器升级时会覆写用于支持 Portal Server 的这些容器的自定义设置。

在此情况下，必须为升级后的 Web 容器环境重新安装和重新配置 Portal Server。

- 4. **Access Manager (Access Manager SDK)**。 第 235 页中的第 14 章 “Access Manager” 提供了将 Access Manager 升级至发行版本 5 的说明。
- 5. **Portal Server Secure Remote Access**。 第 345 页中的第 16 章 “Portal Server Secure Remote Access” 提供了将 Portal Server Secure Remote Access 升级至发行版本 5 的说明。
- 6. **Java DB**。 第 141 页中的第 8 章 “Java DB” 提供了将 Java DB 升级至发行版本 5 的说明。
- 7. **Service Registry**。 第 209 页中的第 12 章 “Service Registry” 提供了将 Service Registry 升级至发行版本 5 的说明。
- 8. **Communications Express**。 《Sun Java Communications Suite Upgrade Guide》(<http://docs.sun.com/doc/819-7561>) 中提供了将 Communications Express 升级到发行版本 5 的说明。

获取所需的配置信息和密码

根据 Web 容器升级方案（参见第 286 页中的表 15-4），psupgrade 脚本会要求您输入有关密码及其他 Web 容器配置数据的信息。不同 Web 容器升级方案所需的信息显示在表 15-6 中。请确保在开始升级 Portal Server 之前组合相关信息。

表 15-6 psupgrade 脚本根据 Web 容器升级方案所需的信息

信息	升级方案 ³	Web Server 7.x 示例值： 方案 2	Application Server 8.x 示例值：方案 5 ²
在 Web Server 7.0 上升级 Portal Server (yes/no)	2	是	不适用
Web 容器安装目录	2 和 5	WebServer7-base	AppServer8Install-base
Web 容器虚拟服务器实例名	2	https-configName ⁴	不适用
Web 容器实例名	5	不适用	server1
Web 容器实例目录	2	WebServer7Config-base/ https-configName ⁴ /	不适用
Portal 实例部署目录	5	不适用	AppServer8Config-base/ domains/domainName
Web 容器实例端口	2 和 5	80	80
Web 容器实例协议	2 和 5	http	http
Web 容器配置名称	2	configName ⁴	不适用
Web 容器域名	5	不适用	domain1

表 15-6 psupgrade 脚本根据 Web 容器升级方案所需的信息（续）

信息	升级方案 ³	Web Server 7.x 示例值： 方案 2	Application Server 8.x 示例值：方案 5 ²
Web 容器文档根目录	2 和 5	WebServer7Config-base/ https-configName ⁴ /docs/	AppServer8Config-base/ domains/domainName/ docroot
Web 容器管理主机名	2 和 5	localhost	localhost
Web 容器管理端口	2 和 5	8989	4848
Web 容器管理协议	2 和 5	https	https
Web 容器管理员用户 ID	2 和 5	admin	admin
Web 容器管理员密码	2 到 5		
Web 容器主密码	3 到 5	不适用	
目录管理员 (cn=Directory manager) 密码	1 到 5		
SRA 日志用户密码 ¹	1 到 5		
Access Manager 管理员密码	1 到 5		
Directory Server Idapuser 密码	1 到 5		
Portal 实例 ID ²	1 到 5		

- 1. 随 Portal Server 一起安装时需要此信息以配置 Portal Server Secure Remote Access 组件。
- 2. 必须为此参数提供一个唯一的非空值。值必须是字母数字，可以包括一个连字符 (-)。
- 3. Web 容器升级方案 5 适用于从发行版本 2 升级 Portal Server。
- 4. configName 的默认值为 hostName.domainName。

备份发行版本 4 Portal Server 配置信息

将 Portal Server 升级到发行版本 5 时，无需重新配置 Portal Server 软件。但是，为安全起见，psupgrade 脚本将备份其中存储有配置信息的下列目录：

PortalServer6Config-base/

注	建议您备份 Portal Server 在其中存储用户配置文件和其他数据的 Directory Server 实例。没有这些数据，将无法执行发行版本 5 Portal Server 的升级回滚。
---	---

记录 Java 虚拟机 (Java Virtual Machine, JVM) 设置

请在升级 Portal Server 之前，记录以下 Web 容器 JVM 设置（如果与默认值不同）：

```
<jvm-options>-XX:MaxPermSize=256m</jvm-options>
<jvm-options>-XX:+CMSPermGenSweepingEnabled</jvm-options>
<jvm-options>-XX:+CMSClassUnloadingEnabled</jvm-options>
```

如下表所示，JVM 设置的位置要视 Web 容器而定。

Web 容器	配置文件
发行版本 2、3、4 Web Server (6.x)	WebServer6-base/https-instanceName/config/server.xml
发行版本 5 Web Server (7.0)	WebServer7Config-base/https-configName/config/server.xml
发行版本 3、4、5 Application Server (8.x)	AppServer8Config-base/domains/domainName/config/domain.xml
Node Agent 管理的 Application Server 实例	AppServer8Config-base/nodeagents/nodeagentName/config/ domain.xml

以后，您需要检查并确认这些 JVM 设置并没有因为 Portal Server 升级而更改。

删除负载均衡器的配置

如果通过负载均衡器访问 Portal Server 实例，则用于配置此类访问的 LOAD_BALANCER_URL 属性的值可能会干扰 Portal Server 升级。因此，必须在执行升级前修改此设置。修改 LOAD_BALANCER_URL 属性设置：

1. 记录下列哪些配置文件驻留在本地（其中有些文件支持那些可能在本地上安装的 Portal Server Secure Remote Access 组件）：
PortalServer6Config-base/PSConfig.properties
PortalServer6Config-base/GWConfig.properties（如果 Gateway 在本地）
PortalServer6Config-base/RWPConfig.properties（如果 Rewriter Proxy 在本地）
PortalServer6Config-base/NLPConfig.properties（如果 Netlet Proxy 在本地）
2. 记录这些配置文件中 LOAD_BALANCER_URL 属性的当前值。
3. 修改 LOAD_BALANCER_URL 属性的值，使其指向相关 Portal Server 实例：
LOAD_BALANCER_URL=*portalHostName:port/portal*

删除 Directory Proxy Server 的配置

如果 Portal Server 实例通过 Directory Proxy Server 实例访问 Directory Server，则必须在执行升级前修改 Directory Proxy Server 的主机和端口号设置，并在升级完成后将其恢复为原始值。

修改相应的设置：

1. 记录以下 Access Manager 配置文件中 DS_HOST 和 DS_PORT 属性的当前值：
AccessManagerConfig-base/config/AMConfig.properties
2. 修改 DS_HOST 和 DS_PORT 属性的值，使其直接指向相关 Directory Server 实例。

升级发行版本 4 Portal Server (Solaris)

本节讨论影响 Portal Server 升级过程的一些注意事项，随后对升级过程本身加以说明。

升级注意事项 (Solaris)

将 Portal Server 软件升级到发行版本 5 时，应考虑以下注意事项：

- 同时升级对应于同一已安装 Portal Server 映像的所有 Portal Server 实例。
- Portal Server 软件由子组件组成，这些子组件扮演着许多不同的角色，但它们要一起升级：
 - **Portal-base**。包括管理 Mbean 和随附的管理软件 Logging Framework 以及监视相关的软件，所有这些软件一起组成软件包。
 - **Portal Server Web 应用程序**。包括部署在 Web 容器中的许多 Web 应用程序。其中至少有一些 Web 应用程序需要来自 Access Manager 的支持，进而需要来自 Directory Server 的支持。
 - **Secure Remote Access 核心组件**。支持 Portal Server Secure Remote Access 的软件：嵌入到 jar 文件中的一些 servlet 和 applet 以及无法部署在 Web 容器中的一些支持文件。
- psupgrade 脚本会自动检测主机上安装了哪些 Portal Server 子组件和哪些 Web 容器依赖项。例如，该脚本会查询系统以检测您要在其中部署 Portal Server Web 应用程序的 Application Server 或 Web Server 的版本，并根据它能够检测到的信息来调整它所请求的信息。

升级过程 (Solaris)

下述过程适用于要进行升级的计算机中的 Portal Server。

1. 以 root 身份登录，或成为超级用户。

```
su -
```

2. 将所有共享组件同步到发行版本 5（如果尚未执行此操作）。

第 59 页中的第 2 章“升级 Java ES 共享组件”中提供了有关说明。

此步骤是第 295 页中的步骤 8 中运行 `psupgrade` 脚本的必要先决条件。

3. 停止可能在本机运行的任何 Portal Server Secure Remote Access Gateway、Rewriter Proxy 或 Netlet Proxy 实例。

```
PortalServer6-base/bin gateway stop
PortalServer6-base/bin netletd stop
PortalServer6-base/bin rwproxysd stop
```

检查这些进程是否已停止：

```
Gateway: netstat -an | grep 443
Rewriter Proxy: netstat -an | grep 10443
Netlet Proxy: netstat -an | grep 10555
```

4. 确保 Access Manager 正在运行，如果它与 Portal Server 部署到不同的 Web 容器中。
5. 如果 Portal Server 尚未运行，请通过启动它所部署到的 Web 容器来启动它。

Web Server 6.x:

```
WebServer-base/https-instanceName/start
```

Web Server 7.0:

管理服务器 --

```
WebServer7Config-base/admin-server/bin/startserv
```

实例服务器 --

```
WebServer7Config-base/https-configName/bin/startserv
```

Application Server 8.x:

```
AppServer8-base/bin/asadmin start-domain --user admin_ID
--password password domainName
```

6. 设置 `psupgrade` 脚本所需的两个环境变量（`ANT_HOME` 和 `JAVA_HOME`）。例如，

```
export ANT_HOME=/usr/sfw
export JAVA_HOME=/usr/jdk/entsys-j2se
```

7. 确保计算机上有足够的交换空间。

指导原则是交换空间应设置为物理 RAM 大小的两倍。

8. 通过 Java ES 发行版本 5 分发软件运行 `psupgrade` 脚本。

```
cd os_arch/Products/portal_svr/Tools/upgrade/bin
./psupgrade
```

其中 `os_arch` 对应您的平台，例如 `Solaris_sparc`。

注	如果您无意中通过错误的 <code>os_arch</code> 目录运行了 <code>psupgrade</code> ，则需要如下所示回退该过程： a. 转到正确的 <code>os_arch</code> 目录。 b. 恢复对 Portal Server 数据的更改。 <pre>./psupgrade rollback</pre>提供请求的参数和密码。 c. 再次运行 <code>psupgrade</code>。
----------	---

`psupgrade` 脚本会检测已安装的 Portal Server 组件和本地化软件包，调用 Java ES 安装程序以安装新软件包，并查询系统以检测您要在其中部署 Portal Server Web 应用程序的 Web 容器的位置、端口号以及其他信息。根据 Web 容器升级方案（参见第 286 页中的表 15-4），该脚本会要求您输入将 Portal Server 部署到相应的 Web 容器时所需的其他信息。

第 290 页中的表 15-6 显示了表 15-4 中不同 Web 容器升级方案所请求的信息。

注	确保输入正确的 <code>psupgrade</code> 参数值，因为您不能返回并更改这些值，并且，对 <code>psupgrade</code> 脚本所做的更改执行回滚操作也是非常困难的。要恢复对 Portal Server 数据所做的更改，您必须在试图再次运行 <code>psupgrade</code> 之前运行 <pre>./psupgrade rollback</pre> 命令。
----------	--

9. 必要时，恢复 Web 容器 JVM 设置。

为确保 JVM 设置支持发行版本 5 Portal Server，请执行以下步骤：

- a. 检查并确认您在升级前所记录的 Portal Server 的 Web 容器 JVM 设置没有因为升级而更改。

参见第 292 页中的“记录 Java 虚拟机 (Java Virtual Machine, JVM) 设置”。

- b. 如果这些设置已更改，则将其恢复为您在升级前所记录的值。

确保包括以下 JVM 设置，即使先前没有对它们进行设置：

```
<jvm-options>-XX:MaxPermSize=256m</jvm-options>
<jvm-options>-XX:+CMSPermGenSweepingEnabled</jvm-options>
<jvm-options>-XX:+CMSClassUnloadingEnabled</jvm-options>
```

10. 停止并重新启动 Web 容器。

虽然并非在所有情况下都需要重新启动 Web 容器，但这样做可确保 Portal Server 在干净的状态下启动。

- a. 如下所示停止 Web 容器：

Web Server 6.x:
WebServer-base/https-instanceName/stop

Web Server 7.0:
管理服务器 --
WebServer7Config-base/admin-server/bin/stopserv
实例服务器 --
WebServer7Config-base/https-configName/bin/stopserv

Application Server 8.x:
AppServer8-base/bin/asadmin stop-domain --user admin_ID
--password password domainName

- b. 使用第 294 页中的步骤 5 中的命令重新启动 Web 容器。

升级发行版本 4 Portal Server (Linux)

本节讨论影响 Portal Server 升级过程的一些注意事项，随后对升级过程本身加以说明。

升级注意事项 (Linux)

在 Linux 平台上将 Portal Server 软件升级到发行版本 5 所考虑的注意事项与在 Solaris 平台上相同（参见第 293 页中的“升级注意事项 (Solaris)”），但在 Linux OS 上，安装发行版本 5 Portal Server 的路径与发行版本 4 相同。因此，在安装发行版本 5 RPM 时，psupgrade 脚本将删除先前的 RPM。

升级过程 (Linux)

下述过程适用于要进行升级的计算机中的 Portal Server。

注意	在 Linux 上，无法对从 Java ES 发行版本 4 到发行版本 5 的升级执行回滚操作。确保在执行以下过程之前备份您的系统。
-----------	--

1. 以 root 身份登录，或成为超级用户。

```
su -
```

2. 将所有共享组件同步到发行版本 5（如果尚未执行此操作）。

第 59 页中的第 2 章“升级 Java ES 共享组件”中提供了有关说明。

此步骤是第 298 页中的步骤 8 中运行 psupgrade 脚本的必要先决条件。

3. 停止可能在本地运行的任何 Portal Server Secure Remote Access Gateway、Rewriter Proxy 或 Netlet Proxy 实例。

```
PortalServer6-base/bin gateway stop
PortalServer6-base/bin netletd stop
PortalServer6-base/bin rwproxyd stop
```

检查这些进程是否已停止：

```
Gateway: netstat -an | grep 443
Rewriter Proxy: netstat -an | grep 10443
Netlet Proxy: netstat -an | grep 10555
```

4. 确保 Access Manager 正在运行，如果它与 Portal Server 部署到不同的 Web 容器中。

5. 如果 Portal Server 尚未运行，请通过启动它所部署到的 Web 容器来启动它。

Web Server 6.x:

```
WebServer-base/https-instanceName/start
```

Web Server 7.0:

管理服务器 --

```
WebServer7Config-base/admin-server/bin/startserv
```

实例服务器 --

```
WebServer7Config-base/https-configName/bin/startserv
```

Application Server 8.x:

```
AppServer8-base/bin/asadmin start-domain --user admin_ID  
--password password domainName
```

6. 设置 psupgrade 脚本所需的两个环境变量（ANT_HOME 和 JAVA_HOME）。例如，

```
export ANT_HOME=/opt/sun
```

```
export JAVA_HOME=/usr/jdk/entsys-j2se
```

7. 确保计算机上有足够的交换空间。

指导原则是交换空间应设置为物理 RAM 大小的两倍。

8. 通过 Java ES 发行版本 5 分发软件运行 psupgrade 脚本。

```
cd os_arch/Products/portal_svr/Tools/upgrade/bin
```

```
./psupgrade
```

其中 *os_arch* 对应您的平台，例如 *Linux_x86*。

psupgrade 脚本会检测已安装的 Portal Server 组件和本地化软件包，调用 Java ES 安装程序以安装新软件包，并查询系统以检测您要在其中部署 Portal Server Web 应用程序的 Web 容器的位置、端口号以及其他信息。根据 Web 容器升级方案（参见第 286 页中的表 15-4），该脚本会要求您输入将 Portal Server 部署到相应的 Web 容器时所需的其他信息。

第 290 页中的表 15-6 显示了表 15-4 中不同 Web 容器升级方案所请求的信息。

注	确保输入正确的 psupgrade 参数值，因为您不能返回并更改这些值，并且，对 psupgrade 脚本所做的更改执行回滚操作也是非常困难的。注意：请在运行 psupgrade 脚本之前备份您的系统。
----------	---

9. 修改 *PortalServer7Config-base/platform.conf.default* 配置文件。

从 *psupgrade* 备份的以下文件中复制带有 *gateway.logging.password* 的一行：

PortalServer6Config-base.bak/platform.conf.default

然后将该行置于 *PortalServer7Config-base/platform.conf.default* 中。

10. 必要时，恢复 Web 容器 JVM 设置。

为确保 JVM 设置支持发行版本 5 Portal Server，请执行以下步骤：

- a. 检查并确认您在升级前所记录的 Portal Server 的 Web 容器 JVM 设置没有因为升级而更改。

参见第 292 页中的“记录 Java 虚拟机 (Java Virtual Machine, JVM) 设置”。

- b. 如果这些设置已更改，则将其恢复为您在升级前所记录的值。

确保包括以下 JVM 设置，即使先前没有对它们进行设置：

```
<jvm-options>-XX:MaxPermSize=256m</jvm-options>
<jvm-options>-XX:+CMSPermGenSweepingEnabled</jvm-options>
<jvm-options>-XX:+CMSClassUnloadingEnabled</jvm-options>
```

11. 停止并重新启动 Web 容器。

虽然并非在所有情况下都需要重新启动 Web 容器，但这样做可确保 Portal Server 在干净的状态下启动。

- a. 如下所示停止 Web 容器：

Web Server 6.x:

```
WebServer-base/https-instanceName/stop
```

Web Server 7.0:

管理服务器 --

```
WebServer7Config-base/admin-server/bin/stopserv
```

实例服务器 --

```
WebServer7Config-base/https-configName/bin/stopserv
```

Application Server 8.x:

```
AppServer8-base/bin/asadmin stop-domain --user admin_ID
--password password domainName
```

- b. 使用第 298 页中的步骤 5 中的命令重新启动 Web 容器。

检验升级

可使用以下命令检验发行版本 5 软件包的安装：

```
PortalServer7-base/bin/psadmin --version --adminuser admin_ID  
-f adminpasswordfile
```

参见第 289 页中的表 15-5 确认输出值。

要检验完全升级，请确认 Portal 桌面出现且 psadmin 管理实用程序按照文档所述的那样起作用。

还可以检查以下升级日志文件：

```
/var/sadm/install/logs/Sun_Java_System_Portal_Server_upgrade.log*  
PortalServer7Data-base/logs/admin/  
PortalServer7Data-base/logs/config/
```

发行版本 4 升级后的任务

请注意处理以下情况所需要的升级后过程：

- [迁移自定义 web-src 数据](#)
- [重新部署自定义 Portlet 应用程序](#)
- [迁移自定义 Portlet 应用程序](#)
- [更正书签和应用程序频道中的链接](#)
- [正确访问搜索服务器](#)
- [恢复 Directory Proxy Server 的配置](#)
- [手动注册 Portal Server Secure Remote Access 组件](#)
- [启用 URLScrapper 频道](#)
- [注销页面中的更改](#)

迁移自定义 web-src 数据

如果您在以下目录中添加了自定义数据，如图像、javascript 文件或用于构建 portal.war 的任何其他文件：

```
PortalServer6-base/web-src
```

则必须将这些附加文件复制到发行版本 5 Portal Server 的对应目录中：

```
PortalServer7-base/web-src
```

重新部署自定义 Portlet 应用程序

如果您创建和部署了自定义 portlet 应用程序，则在升级到发行版本 5 Portal Server 之后必须手动重新部署这些 portlet。即使显示配置文件条目存在且频道名称将会显示出来，您也不会看到内容，直到重新部署自定义 portlet 为止。

使用以下命令重新部署 portlet:

```
PortalServer7-base/bin/psadmin deploy-portlet
```

可通过在以下位置查找对应的 .war 和 XML 文件来确认重新部署:

```
PortalServer7Data-base/portals/Upgraded/war
```

迁移自定义 Portlet 应用程序

对于基于由 Sun Java Web Console (SJWC) 提供的用户界面框架的 portlet 应用程序，您需要手动将其迁移至发行版本 5 并进行重新部署。

具体而言，这种需求适用于四个 Web 应用程序：filesharing、surveys、wiki 和 rssportlet，它们作为要自定义并安装到 portal 中的样例 portlet 应用程序随 Portal Server 一起分发。升级期间，这些样例 portlet 应用程序的已修复错误版本置于磁盘中的 portlet 应用程序区域。如果您已自定义这些 portlet 应用程序供您自己使用，则需要将它们手动迁移至发行版本 5 并重新进行部署；Portal Server 升级过程不会处理它们。

默认情况下，其中的某些 portlet 应用程序（filesharing 和 surveys）已部署且在创建了社区时可供用户使用。

可使用以下过程升级、自定义和重新部署基于 SJWC 的 portlet 应用程序。其中，以 filesharing portlet 应用程序为例：

1. 解压缩发行版本 5 SJWC jar 文件。
 - a. mkdir /tmp/lh
 - b. cd /tmp/lh
 - c. /usr/jdk/entsys-j2se/bin/jar xvf


```
PortalServer7-base/portlet/communityportlets.war
WEB-INF/lib/commons-beanutils.jar
WEB-INF/lib/commons-collections-3.1.jar
WEB-INF/lib/commons-digester.jar
WEB-INF/lib/commons-logging.jar
WEB-INF/lib/dataprovider.jar WEB-INF/lib/jsf-api.jar
WEB-INF/lib/jsf-impl.jar WEB-INF/lib/webui.jar
```

- d. 重命名其中一个文件。

```
mv WEB-INF/lib/commons-collections-3.1.jar  
WEB-INF/lib/commons-collections.jar
```

2. 找到 filesharing portlet 应用程序。

```
cd PortalServer7Config-base/portals/portall/portletapps/filesharing
```

3. 插入经过更新的 SJWC 库。

```
jar uvf src/filesharing.war.tokenized -C /tmp/lh WEB-INF
```

4. 自定义 filesharing portlet 应用程序。

```
ant customize
```

5. 重新部署 filesharing portlet 应用程序。

- a. *PortalServer7-base/bin/psadmin undeploy-portlet -u amadmin -f passwordfile -p portal_id -i instance_id -g filesharing*

- b. *ant deploy*

- c. 转到以下目录（视 Web 容器而定）：

Application Server 8.x:

*AppServer8Config-base/domains/domain1/applications/j2ee-modules/
communityportlet/WEB-INF*

Web Server 6.x:

*WebServer6-base/https-instanceName/webapps/https-instanceName/
communityportlet/WEB-INF*

Web Server 7.x:

*WebServer7Config-base/https-configName/web-app/https-configName/
communityportlet/WEB-INF*

- d. 打开 sun-web.xml 文件并在最后一行之前（即在 sun-web-app 结束标记之前）添加以下行：

```
<class-loader delegate="false"/>
```

- e. 对 filesharing/WEB-INF 下的 sun-web.xml 文件，重复步骤 c 和步骤 d。

6. 对 surveys 以及基于 SJWC 框架的任何其他自定义 portlet 应用程序，重复步骤 2 到步骤 5。

7. 重新启动 Web 容器。

更正书签和应用程序频道中的链接

将发行版本 4 Portal Server 升级到发行版本 5 时，书签和应用程序频道中存在许多重复和虚假的链接。要修复这些链接，执行以下过程。

1. 登录到 PS 控制台。
2. 在“常用任务”选项卡上，单击“管理频道和容器”。
3. 选择 "DeveloperSample [Org]" 作为 DN，然后单击“确定”。
4. 选择“JSPTabContainer [默认值]”作为“视图类型”。
5. 在 "MyFrontPageTabPanelContainer" 下，单击“应用程序”频道。

应用程序频道属性将显示在右框中。

要针对某个具体语言环境查看属性，单击“表首选项”，随后将提供“语言环境”值：de、fr、es、ja、ko、zh、zh_CN、zh_TW。

6. 编辑 "userApps" 属性。
 - a. 在 "userApps" 属性中，单击 “[编辑值 ...]” 链接。
将出现一个弹出式窗口，其中显示了现有应用程序。
 - b. 从列表中删除以下应用程序：

NetMail Lite
NetMail
 - c. 将以下应用程序添加到列表中：

NetFile
 - d. 单击“保存”，然后单击“关闭”。
7. 编辑 "target" 属性。
 - a. 在 "target" 属性中，单击 “[编辑值 ...]” 链接。
将出现一个弹出式窗口，其中显示了现有目标。
 - b. 从列表中删除以下目标：

```
NetMailLite|
NetMailServlet?nsid=newHTMLSessionNetMailLite|
NetMailServlet?nsid=newHTMLSession

NetMail|NetMailServlet?nsid=newAppletSession
```

- c. 删除以下重复出现的目标:

```
Instant Messenger (Java WebStart) |  
IMLaunch?provider=IMChannel&launch=jnlp&last=false  
  
Instant Messenger (Browser) |  
IMLaunch?provider=IMChannel&launch=plugin&last=false
```

- d. 将以下目标添加到列表中:

```
NetFile | /portal/NetFileApplet?Refer=java2
```

- e. 单击“保存”，然后单击“关闭”。

正确访问搜索服务器

在将发行版本 4 Portal Server 升级到发行版本 5 时，搜索服务器与 Portal Server 分离开来，因此，用于访问搜索服务器的 URL 也将发生更改。

- 在发行版本 4 中，URL 为 `http://hostName:port/portal/search`
- 在发行版本 5 中，URL 为 `http://hostName:port/UpgradedSearch/search`

因此，对于用于实现 `SearchProvider` 或 `DiscussionProvider` 接口的所有 portal 频道（如 `Search`、`DiscussionLite`、`Discussions` 和 `Instant Messaging` 频道），必须手动修改它们的显示配置文件。特别是，必须修改这些频道的 `searchServer` 属性（无论它们可能出现在显示配置文件的哪一组织或角色级别），以正确引用搜索服务器。如下所示修改 `searchServer` 值：

```
value="http://hostName:port/UpgradedSearch/search"
```

另外，对于 `Instant Messaging` 频道，需要用新的搜索服务器 URL 来更新 `Instant Messaging` 服务器配置属性 `iim_arch.portal.search`。

恢复 Directory Proxy Server 的配置

如果 Portal Server 实例已通过 Directory Proxy Server 实例访问 Directory Server，则必须在升级前将 Directory Proxy Server 的主机和端口号设置恢复为其原始值。参见第 293 页中的“删除 Directory Proxy Server 的配置”，在此过程中已修改了这些属性的值以备升级之用。

手动注册 Portal Server Secure Remote Access 组件

如果在升级过程结束时，您遇到报告给标准输出的空指针异常问题，则表明注册 Portal Server Secure Remote Access 组件（如果有）已失败。

在这种情况下，可通过执行以下命令来手动注册（启用）Portal Server Secure Remote Access 组件：

```
PortalServer7-base/bin/psadmin provision-sra -u amadminUser -f passwordFile
-p Portal_ID --gateway-profile profileName --enable
```

启用 URLScrapper 频道

从发行版本 4 升级到发行版本 5 时，必须启用 URLScrapper 频道。请使用以下过程：

1. 登录到 Portal Server 控制台

单击 "Portal" 选项卡，然后单击已升级的 portal。

2. 从“选择 DN”下拉菜单中，选择“顶层（全局）”并单击“下载显示配置文件”链接。

将下载的文件存储在某个临时位置

3. 找到 com.sun.portal.providers.urlscrapper.URLScrapperProvider。

4. 找到 XML 部分，它以下列内容开头：

```
<Provider advanced="false"
class="com.sun.portal.providers.urlscrapper.URLScrapperProvider"
```

并以下列内容结尾：

```
</Provider>
```

5. 用以下内容替换步骤 4 中的 XML 部分：

```
<Provider advanced="false"
class="com.sun.portal.providers.urlscrapper.URLScrapperProvider"
container="false" lock="false" merge="fuse"
name="URLScrapperProvider" version="2">

<Properties advanced="false" lock="false" merge="fuse"
name="_properties" propagate="true">

<String advanced="false" lock="false" merge="replace" name="title"
propagate="true" value="UrlScrapper Channel"/>

<String advanced="false" lock="false" merge="replace"
name="description" propagate="true" value="This is a test for
urlscrapper"/>

<Boolean advanced="true" lock="false" merge="replace"
name="isEditable" propagate="true" value="false"/>
```

```

<Boolean advanced="true" lock="false" merge="replace"
name="isTopLevel" propagate="true" value="false"/>

<String advanced="true" lock="false" merge="replace" name="editType"
propagate="true" value="edit_subset"/>

<Boolean advanced="true" lock="false" merge="replace"
name="enableUBT" propagate="true" value="false"/>

<String advanced="false" lock="false" merge="replace"
name="urlScraperRulesetID" propagate="true"
value="default_ruleset"/>

<String advanced="false" lock="false" merge="replace" name="width"
propagate="true" value="thick"/>

<String advanced="true" lock="false" merge="replace"
name="refreshTime" propagate="true" value="0"/>

<String advanced="true" lock="false" merge="replace" name="helpURL"
propagate="true" value="en/desktop/urlscrpr.htm"/>

<String advanced="false" lock="false" merge="replace" name="url"
propagate="true" value=""/>

<String advanced="false" lock="false" merge="replace"
name="fontFacel" propagate="true" value="Sans-serif"/>

<String advanced="false" lock="false" merge="replace"
name="productName" propagate="true" value="Sun Java™ System Portal
Server 7"/>

<Boolean advanced="false" lock="false" merge="replace"
name="cookiesToForwardAll" propagate="true" value="true"/>

<String advanced="false" lock="false" merge="replace"
name="inputEncoding" propagate="true" value="UTF-8"/>

<Collection advanced="false" lock="false" merge="fuse"
name="cookiesToForwardList" propagate="true"/>

<Integer advanced="false" lock="false" merge="replace"
name="timeout" propagate="true" value="100"/>

<String advanced="true" lock="false" merge="replace" name="formData"
propagate="true" value=""/>

<Boolean advanced="true" lock="false" merge="replace"
name="isHttpAuth" propagate="true" value="false"/>

```

```

<String advanced="true" lock="false" merge="replace" name="loginUrl"
propagate="true" value=""/>

<String advanced="true" lock="false" merge="replace"
name="loginFormData" propagate="true" value=""/>

<String advanced="true" lock="false" merge="replace" name="uid"
propagate="true" value=""/>

<String advanced="true" lock="false" merge="replace" name="password"
propagate="true" value=""/>

<ConditionalProperties advanced="false" condition="client"
lock="false" merge="fuse" propagate="true" value="HTML">

<ConditionalProperties advanced="false" condition="locale"
lock="false" merge="fuse" propagate="true" value="en">

<String advanced="true" lock="false" merge="replace" name="helpURL"
propagate="true" value="en/desktop/urlscrpr.htm"/>

<String advanced="false" lock="false" merge="replace" name="url"
propagate="true" value=""/>

</ConditionalProperties>

<String advanced="true" lock="false" merge="replace" name="helpURL"
propagate="true" value="en/desktop/urlscrpr.htm"/>

<String advanced="false" lock="false" merge="replace" name="url"
propagate="true" value=""/>

</ConditionalProperties>

<ConditionalProperties advanced="false" condition="locale"
lock="false" merge="fuse" propagate="true" value="en">

<String advanced="false" lock="false" merge="replace" name="title"
propagate="true" value="UrlScraper Channel"/>

<String advanced="false" lock="false" merge="replace"
name="description" propagate="true" value="This is a test for
urlscraper"/>

</ConditionalProperties>

</Properties>

</Provider>

```

6. 保存并上载修改的文件。

注销页面中的更改

发行版本 5 Portal Server 注销页面已不同于先前的 Access Manager 注销页面。请注意，此更改并不代表该软件中有缺陷。

升级回滚 (Solaris)

本节介绍影响 Portal Server 升级回滚过程的一些注意事项，随后对升级回滚过程本身加以说明。

回滚注意事项 (Solaris)

发行版本 5 的升级回滚过程包括在 *PortalServer6-base* 处恢复为发行版本 4 的安装以及重新部署发行版本 4 Web 应用程序。

回滚过程 (Solaris)

1. 以 root 身份登录，或成为超级用户。

```
su -
```

2. 将 Directory Server 恢复为升级前的状态。

使用 Directory Server 备份 / 恢复命令行和 GUI 实用程序。参见《Sun Java System Directory Server Enterprise Edition 6.0 管理指南》(<http://docs.sun.com/doc/820-0293>) 中的“目录服务器备份和恢复”一章。

3. 通过停止其 Web 容器，停止 Portal Server。

Web Server 6.x:

```
WebServer-base/https-instanceName/stop
```

Web Server 7.0:

管理服务器 --

```
WebServer7Config-base/admin-server/bin/stopserv
```

实例服务器 --

```
WebServer7Config-base/https-configName/bin/stopserv
```

Application Server 8.x:

```
AppServer8-base/bin/asadmin stop-domain --user admin_ID  
--password password domainName
```

4. 删除发行版本 5 Portal Server 软件包。

- a. 启动 Java ES 卸载程序。

```
/var/sadm/prod/SUNWentsys5/uninstall
```

- b. 选择所有已安装的 Portal Server 组件。

- c. 确认您的卸载选择。
 - d. 退出 Java ES 卸载程序。
5. 通过重新启动其 Web 容器，重新启动 Portal Server。

Web Server 6.x:

WebServer-base/https-instanceName/start

Web Server 7.0:

管理服务器 --

WebServer7Config-base/admin-server/bin/startserv

实例服务器 --

WebServer7Config-base/https-configName/bin/startserv

Application Server 8.x:

AppServer8-base/bin/asadmin start-domain --user admin_ID

--password password domainName

6. 通过 Java ES 发行版本 5 分发软件使用以下命令重新部署发行版本 4 Portal Server Web 应用程序:

```
cd os_arch/Products/portal_svr/Tools/upgrade/bin
```

```
./psupgrade rollback
```

其中 *os_arch* 对应您的平台，例如 *Solaris_sparc*。

psupgrade rollback 命令用于取消部署发行版本 5 Portal Server Web 应用程序并重新部署发行版本 4 Portal Server Web 应用程序。

该命令会将内容从 *PortalServer6-base/web-src* 重新部署至

/var/PortalServe6-base/https-hostName/deploy-dir/web-apps。因此，应首先在 */web-src* 中进行 Portal Server Web 应用程序的任何自定义设置，然后再将其部署至 */web-apps*。运行 *psupgrade rollback* 命令之前，应在 */web-src* 中复制可能在 */web-apps* 下所做的任何更改，否则，此类更改将会被覆写。

7. 停止并重新启动 Web 容器。

虽然并非在所有情况下都需要重新启动 Web 容器，但这样做可确保 Portal Server 在干净的状态下启动。

升级回滚 (Linux)

由于到发行版本 5 的升级需要删除发行版本 4 的二进制文件，因此在 Linux 上执行升级回滚是非常困难的。

一种回滚方法是在尝试进行升级之前，先创建一个并行系统，然后升级并测试该并行系统。如果需要对升级执行回滚操作，可恢复为该并行系统。

多实例升级

在一些部署体系结构中，Portal Server 被部署在多个计算机系统上，以提供可伸缩性和提高可用性。例如，您可以在多台计算机上运行 Portal Server 实例，并使用一个负载均衡器来分配负载。

就 Portal Server 负载均衡的实例而言，可执行滚动升级，在此期间可依序升级各个 Portal Server 实例而不中断服务，如下所述。该过程要考虑以下约束：发行版本 4 Portal Server 不能使用发行版本 5 的 Portal Server 目录数据。

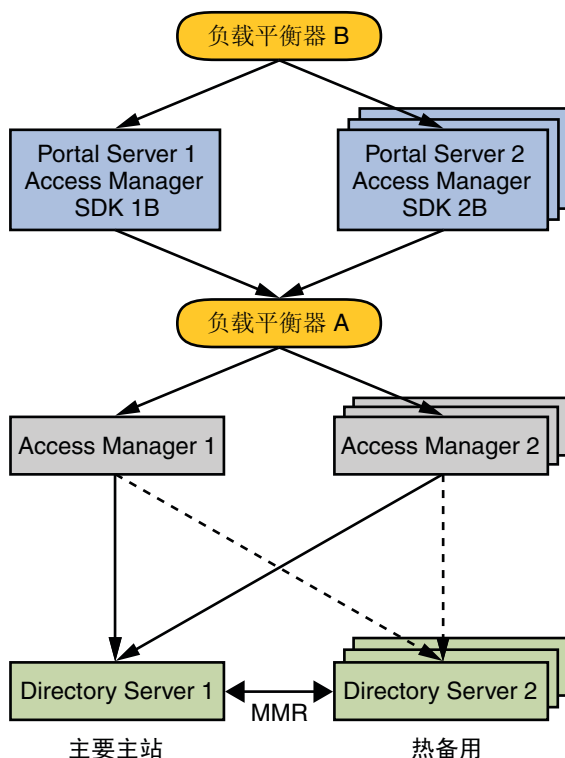
第 311 页中的图 15-1 中所示的部署体系结构用于说明将发行版本 4 Portal Server 实例滚动升级到发行版本 5 的过程。

注 有关包括 Portal Server Secure Remote Access 组件的体系结构，参见第 364 页中的“多实例升级”。

在图 15-1 的体系结构中，通过负载均衡器访问多个 Portal Server 实例以提供可用性和可伸缩性。而 Portal Server 实例又通过负载均衡器访问 Access Manager 实例。Access Manager 和 Access Manager SDK 实例将访问为多主复制 (multi-master replication, MMR) 设置的目录。尽管可以使用其他 Directory Server 复制方案，但 MMR 是具有高可用性和高可伸缩性的目录服务。

在图 15-1 中，对 Portal Server、Access Manager 和 Directory Server 的多个实例进行了编组以便于解释升级过程。例如，Portal Server 2 代表 Portal Server 的第 2 个到第 n 个实例。

图 15-1 多 Portal Server 实例的部署体系结构示例



从发行版本 4 Portal Server 到发行版本 5 的滚动升级过程如下：

1. 如果您要将发行版本 4 Access Manager 升级到发行版本 5，请如第 257 页中的“多实例升级”中所述执行滚动升级。请注意，在将发行版本 4 Portal Server 升级到发行版本 5 时，不必一定将发行版本 4 Access Manager 升级到发行版本 5。
2. 将 Portal Server 2 配置为指向 Directory Server 2 而不是 Directory Server 1。
为简洁起见，在此步骤和后续步骤中，“Portal Server 2”表示 Portal Server 2 到 Portal Server n。
3. 升级 Portal Server 1。
 - a. 在负载均衡器 B 中禁用 Portal Server 1。
请求不会再路由至 Portal Server 1。

b. 禁用 Directory Server MMR。

Directory Server 2 将不再与 Directory Server 1 同步。

c. 将 Access Manager SDK 1B 升级到发行版本 5。

按照第 259 页中的“仅升级发行版本 4 Access Manager SDK”中的过程操作。

d. 将 Portal Server 1 升级到发行版本 5。

按照第 288 页中的“发行版本 4 Portal Server 的升级”中所述执行 Portal Server 实例的升级，但要注意以下事项：

- 特别注意以下升级前任务：第 292 页中的“删除负载均衡器的配置”。
- 在执行升级之前，确认
AccessManagerConfig-base/config/AMConfig.properties 文件中
am.encrypted.pwd 的值对于本地 Access Manager SDK 以及与其相关联的远程 Access Manager 实例都是相同的。
- 对于 psupgrade 为您要升级的每个 Portal Server 实例所请求的 Portal Instance ID 参数，确保为其提供一个唯一的非空值。

Directory Server 1 的 Portal Server 数据已更新为发行版本 5。

e. 在负载均衡器 B 中启用 Portal Server 1。

请求将重新路由至 Portal Server 1。

4. 升级 Portal Server 2。

a. 在负载均衡器 B 中禁用 Portal Server 2。

请求不会再路由至 Portal Server 2。

b. 将 Portal Server 2 的配置恢复为指向 Directory Server 1。

c. 将 Access Manager SDK 2B 升级到发行版本 5。

使用第 312 页中的步骤 c 中的同一过程。

d. 将 Portal Server 2 升级到发行版本 5。

使用第 312 页中的步骤 d 中的同一过程。

e. 在负载均衡器 B 中启用 Portal Server 2。

请求将重新路由至 Portal Server 2。

5. 启用 Directory Server MMR。

现在，Directory Server 2 的 Portal Server 数据与 Directory Server 1 同步。

从 Java ES 发行版本 3 升级 Portal Server

将 Java ES 2005Q1（发行版本 3）Portal Server 升级到发行版本 5 的过程与将发行版本 4 Portal Server 升级到发行版本 5 的过程相同，但存在以下例外：

- 发行版本 3 升级前的任务：升级 Portal Server 依赖项
- 升级发行版本 3 Portal Server
- 多实例升级

发行版本 3 升级前的任务：升级 Portal Server 依赖项

从发行版本 3 升级 Portal Server 时，必须在升级 Portal Server 之前将 Access Manager 和 Web 容器（Web Server 或 Application Server）同时升级到发行版本 4 或发行版本 5，但不能将任何依赖项保留在发行版本 3，也不能将一些依赖项升级到发行版本 4 而将另一些依赖项升级到发行版本 5。有关更多信息，参见第 285 页中的“选择性升级问题”。

以下依赖项需要按照下面所示的顺序进行升级。

1. **共享组件。** 第 59 页中的第 2 章“升级 Java ES 共享组件”提供了将 Java ES 共享组件升级至发行版本 5 的说明。
2. **Directory Server。** 第 89 页中的第 5 章“Directory Server”提供了将 Directory Server 升级至发行版本 5 的说明。
3. **Web 容器软件。** 第 119 页中的第 7 章“Web Server”和第 183 页中的第 11 章“Application Server”分别提供了升级 Web Server 和 Application Server 的说明。

注

升级诸如来自 Weblogic 和 WebSphere 的第三方 Web 容器时可能导致 Portal Server 中断，因为在容器升级时会覆写用于支持 Portal Server 的这些容器的自定义设置。

在此情况下，必须为升级后的 Web 容器环境重新安装和重新配置 Portal Server。

4. **Access Manager (Access Manager SDK)。** 第 235 页中的第 14 章“Access Manager”提供了将 Access Manager 升级至发行版本 5 的说明。

升级发行版本 3 Portal Server

要将发行版本 3 Portal Server 升级到发行版本 5，请按照第 287 页中的“从 Java ES 发行版本 4 升级 Portal Server”中的说明操作，但需要在提到发行版本 4 的地方用发行版本 3 代替。

发行版本 3 升级后的任务

将 Portal Server 从发行版本 3 升级到发行版本 5 时，除了第 300 页中的“发行版本 4 升级后的任务”中所述的升级后过程以外，还必须执行处理以下情况所必需的升级后过程：

- [订阅讨论](#)

订阅讨论

只有先编辑全局显示配置文件顶层属性以在其中添加以下 String 属性，在社区中订阅讨论才会成功：

```
helpURL=en/desktop/usedesk.htm
```

请按以下过程操作：

1. 创建显示配置文件 XML 代码片段文件 helpUrl.xml：

```
<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE DisplayProfile SYSTEM "jar://resources/psdp.dtd">
  <Properties>
    <String name="helpURL" value="en/desktop/usedesk.htm" />
  </Properties>
```

2. 使用以下命令运行全局显示配置文件属性：

```
./psadmin modify-dp -u amadminUser -f /tmp/passwordFile -p portal_ID
-m -g helpUrl.xml
```

其中，-m 选项是必需的，以保证不会覆写整个全局显示配置文件。

多实例升级

在一些部署体系结构中，Portal Server 被部署在多个计算机系统上，以提供可伸缩性和提高可用性。例如，您可以在多台计算机上运行 Portal Server 实例，并使用一个负载均衡器来分配负载。

就 Portal Server 负载均衡的实例而言，可执行滚动升级，在此期间可依序升级各个 Portal Server 实例而不中断服务，如下所述。该过程要考虑以下约束：发行版本 3 Portal Server 不能使用发行版本 5 的 Portal Server 目录数据。

要执行从发行版本 3 Portal Server 到发行版本 5 的滚动升级，请使用第 310 页中的“多实例升级”中所述的同一过程，但需要在提到发行版本 4 的地方用发行版本 3 代替。此外，还必须如第 311 页中的步骤 1 中所述升级 Access Manager。

从 Java ES 发行版本 2 升级 Portal Server

本节包括有关将 Java ES 2004Q2（发行版本 2）Portal Server 升级到发行版本 5 的信息。该升级过程与将发行版本 4 Portal Server 升级到发行版本 5 的过程类似，但其中有一些更改，如以下几节所述：

- [发行版本 2 升级前的任务](#)
- [升级发行版本 2 Portal Server](#)
- [发行版本 2 升级后的任务](#)
- [多实例升级](#)

注 如果要在 Linux 平台上从发行版本 2 Portal Server 升级，则必须执行双重升级，即 Portal Server 和操作系统都要升级（RHEL 2.1 上不支持发行版本 5 Portal Server）。有关更多信息，参见[第 286 页中的“双重升级”](#)。

发行版本 2 升级前的任务

除了升级 Portal Server 依赖项以外，从发行版本 2 升级 Portal Server 的升级前任务与[第 288 页中的“发行版本 4 升级前的任务”](#)中所述的那些任务相同。

从发行版本 2 升级 Portal Server 时，必须在升级 Portal Server 之前将 Access Manager 和 Web 容器（Web Server 或 Application Server）同时升级到发行版本 4 或发行版本 5，但不能将任何依赖项保留在发行版本 2，也不能将一些依赖项升级到发行版本 4 而将另一些依赖项升级到发行版本 5。有关更多信息，参见[第 285 页中的“选择性升级问题”](#)。

需特别指出的是，Web 容器软件必须从发行版本 2 升级，这意味着在运行 `psupgrade` 脚本时只支持[第 286 页中的表 15-4](#)中的方案 2 和方案 5。

以下依赖项需要按照下面所示的顺序进行升级。

1. **共享组件。** [第 59 页中的第 2 章“升级 Java ES 共享组件”](#)提供了将 Java ES 共享组件升级至发行版本 5 的说明。
2. **Directory Server。** [第 103 页中的“从 Java ES 发行版本 2 升级 Directory Server”](#)提供了将 Directory Server 升级至发行版本 5 的说明。

3. **Web 容器软件。** 第 119 页中的第 7 章 “Web Server” 和第 183 页中的第 11 章 “Application Server” 分别提供了升级 Web Server 和 Application Server 的说明。

注 升级诸如来自 Weblogic 和 WebSphere 的第三方 Web 容器时可能导致 Portal Server 中断，因为在容器升级时会覆写用于支持 Portal Server 的这些容器的自定义设置。在这些情况下，必须针对已升级的 Web 容器环境重新安装和重新配置 Portal Server。

4. **Access Manager (Access Manager SDK)。** 第 235 页中的第 14 章 “Access Manager” 提供了将 Access Manager 升级至发行版本 5 的说明。

升级发行版本 2 Portal Server

如以下各节所述，将 Portal Server 从发行版本 2 升级到发行版本 5 的过程视您在其中部署 Portal Server 软件的 Web 容器而定。

升级发行版本 2 Portal Server: Web Server Web 容器

当部署到 Web Server Web 容器中时，要将发行版本 2 Portal Server 升级到发行版本 5，请按第 287 页中的 “从 Java ES 发行版本 4 升级 Portal Server” 中的说明操作，只不过要在提到发行版本 4 的地方替换为发行版本 2。

但是，如果 Portal Server 部署到了发行版本 5 Web Server (Web Server 7.0)，则必须在升级发行版本 2 Web Server 之前执行以下步骤：

1. 登录到 Web Server 管理控制台。
2. 单击 “编辑虚拟服务器” > “Web 应用程序”。
3. 删除其 URI 包括 /portal 或 /portalsamples 的所有已部署 Web 应用程序。
4. 单击 “保存”。
5. 单击 “部署暂挂”。

升级发行版本 2 Portal Server: Application Server Web 容器

在将发行版本 2 Portal Server 升级到发行版本 5 时，如果部署到了 Application Server Web 容器中，则 Application Server 已从发行版本 2 升级到发行版本 5。

最初在其中部署 Portal Server 的发行版本 2 Application Server 实例 (*instanceName*) 会在升级到发行版本 5 之后，被迁移到 Application Server 升级过程所创建的一个节点代理下。对于此已升级 Application Server 实例中的 Portal Server 的升级，需要执行以下步骤：

1. 以 root 身份登录，或成为超级用户。

```
su -
```

2. 将所有共享组件同步到发行版本 5（如果尚未执行此操作）。

[第 59 页中的第 2 章“升级 Java ES 共享组件”](#)中提供了有关说明。

此步骤是[第 319 页中的步骤 9](#)中运行 `psupgrade` 脚本的必要先决条件。

3. 停止可能在本地运行的任何 Portal Server Secure Remote Access Gateway、Rewriter Proxy 或 Netlet Proxy 实例。

```
PortalServer6-base/bin gateway stop
PortalServer6-base/bin netletd stop
PortalServer6-base/bin rwproxyd stop
```

检查这些进程是否已停止：

```
Gateway: netstat -an | grep 443
Rewriter Proxy: netstat -an | grep 10443
Netlet Proxy: netstat -an | grep 10555
```

4. 确保 Access Manager 正在运行，如果它与 Portal Server 部署到不同的 Web 容器中。
5. 如果 Portal Server 尚未运行，请通过启动它所部署到的 Web 容器来启动它。
 - a. 启动 Domain Administration Server (DAS)（如果尚未启动）。

```
AppServer8-base/bin/asadmin start-domain --user admin_ID
--password password domainName
```

- b. 启动其中部署了 Portal Server 的已升级 Application Server 实例 (*instanceName*) (如果该服务器实例尚未运行)。

通过启动其下迁移了已升级 Application Server 实例的节点代理来实现此操作：

```
AppServer8-base/bin/asadmin start-node-agent --user admin_ID
--password password nodeagentName
```

在上述命令及后续步骤中，将使用下面的约定：

- 其中，*nodeagentName* 的格式为 *hostName_domainName*，但默认情况下只是 *hostName*。
- 默认的 *domainName* 为 *domain1*
- 默认的 *instanceName* 为 *server1*

6. 取消部署 Portal Server 组件。

```
AppServer8-base/bin/asadmin undeploy --user admin_ID
--password password --target instanceName portal
```

```
AppServer8-base/bin/asadmin undeploy --user admin_ID
--password password --target instanceName portletsamples
```

7. 设置 *psupgrade* 脚本所需的两个环境变量 (*ANT_HOME* 和 *JAVA_HOME*)。例如，Solaris OS：

```
export ANT_HOME=/usr/sfw
export JAVA_HOME=/usr/jdk/entsys-j2se
```

Linux 操作系统：

```
export ANT_HOME=/opt/sun
export JAVA_HOME=/usr/jdk/entsys-j2se
```

8. 确保计算机上有足够的交换空间。

指导原则是交换空间应设置为物理 RAM 大小的两倍。

9. 通过 Java ES 发行版本 5 分发软件运行 *psupgrade* 脚本。

```
cd os_arch/Products/portal_svr/Tools/upgrade/bin
./psupgrade
```

其中 *os_arch* 对应您的平台，例如 *Solaris_sparc*。

注 如果您无意中通过错误的 `os_arch` 目录运行了 `psupgrade`，则需要如下所示回退该过程：

- a. 转到正确的 `os_arch` 目录。
- b. 恢复对 Portal Server 数据的更改。

```
./psupgrade rollback
```

提供请求的参数和密码。

- c. 再次运行 `psupgrade`。
-

`psupgrade` 脚本会调用 Java ES 安装程序以安装新软件包，并查询系统以检测您要在其中部署 Portal Server Web 应用程序的 Web 容器的位置、端口号以及其他信息。根据 Web 容器升级方案（参见第 286 页中的表 15-4；在本例中为方案 5），该脚本会要求您输入将 Portal Server 部署到相应的 Web 容器时所需的信息。

第 290 页中的表 15-6 显示了将发行版本 2 Application Server 升级到发行版本 5 时所请求的信息（方案 5）。

注 确保输入正确的 `psupgrade` 参数值，因为您不能返回并更改这些值，并且，对 `psupgrade` 脚本所做的更改执行回滚操作也是非常困难的。要恢复对 Portal Server 数据所做的更改，您必须在试图再次运行 `psupgrade` 之前运行

```
./psupgrade rollback
```

命令。

10. 停止在第 318 页中的步骤 5 中启动的 Domain Administration Server (DAS) 和节点代理。

```
AppServer8-base/bin/asadmin stop-domain --user admin_ID
--password password domainName
```

```
AppServer8-base/bin/asadmin stop-node-agent --user admin_ID
--password password nodeagentName
```

11. 重新启动在[步骤 10](#)中停止的 Domain Administration Server (DAS)、节点代理和服务器实例。

注 确保在启动服务器实例之前使用 `startinstances=false` 选项单独启动节点代理，如下所示。

```
AppServer8-base/bin/asadmin start-domain --user admin_ID
--password password domainName
```

```
AppServer8-base/bin/asadmin start-node-agent --port DASportNumber
--startinstances=false --user admin_ID --password password
nodeagentName
```

```
AppServer8-base/bin/asadmin start-instance --port DASportNumber
--user admin_ID --password password instanceName
```

`DASportNumber` 的默认值为 4848。

发行版本 2 升级后的任务

将 Portal Server 从发行版本 2 升级到发行版本 5 时，除了第 300 页中的“[发行版本 4 升级后的任务](#)”中所述的升级后过程以外，还必须执行处理以下情况所必需的升级后过程：

- [单点登录配置](#)
- [启用 URLScrapper 频道](#)
- [删除网关服务条目](#)

单点登录配置

从发行版本 2 升级 Portal Server 后，使用 `ssoadapter` 元模板访问后端服务器的 Portal 桌面通信频道（如邮件、日历和通讯录）将失败。

例如，如果已使用特定于 Messaging Server 的设置修改了发行版本 2 邮件 `ssoadapter` 元模板 `SUN-UWC-MAIL`，则在升级到发行版本 5 以后，将存在两个 `SUN-UWC-MAIL ssoadapter` 元模板：一个属于发行版本 2 的版本（尚未进行更改），另一个属于新的发行版本 5 的版本。在 Portal Server 控制台和 `psadmin` 命令行界面中，您将看到两个重复的 `ssoadapter` 元模板，二者的名称相同。

使用 `ssoadapter` 元模板的频道将无法与后端服务器建立连接并无法检索数据。

要解决此问题，必须检索 `ssoadapter` 元模板数据、重命名重复的条目，然后替换已修改的数据。按照以下过程进行操作：

1. 导出 `ssoadapter` 元模板数据。

使用 `amadmin` 实用程序导出 Access Manager 服务数据，如下所示：

a. 创建一个 `amadmin` 请求文件： `/tmp/ssoadapter-template-gets.xml`。

实用程序将使用此文件检索 `ssoadapter` 元模板数据：

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE Requests
  PUBLIC "-//iPlanet//iDSAME 5.0 Admin CLI DTD//EN"
  "jar://com/iplanet/am/admin/cli/amAdmin.dtd"
>
<Requests>
  <SchemaRequests serviceName="SunSSOAdapterService"
    SchemaType="global">
    <GetServiceDefaultValues>
      <Attribute name="sunConfigurationTemplates"/>
    </GetServiceDefaultValues>
  </SchemaRequests>
</Requests>
```

b. 执行以下 `asadmin` 命令：

```
AccessManager-base/bin/amadmin -u amadminUser -w password
-t ssoadapter-templates-get.xml > /tmp/ssoadapter-templates.xml
```

该命令的输出将保存到 `/tmp/ssoadapter-templates.xml` 中。

`/tmp/ssoadapter-templates.xml` 文件的格式如下：

```
sunConfigurationTemplates=
[<ssoadapter meta-template1>, <ssoadapter meta-template2>, ...]
```

每个 `<ssoadapter meta-template>` 的语法如下：

```
default|imap:/?configName=SUN-UWC-MAIL
&proxyAdminPassword=%5BPROXY-ADMIN_PASSWORD%5D&subType=sun-one
&enableProxyAuth=false ...
```

2. 修改 `/tmp/ssoadapter-templates.xml` 文件以重命名重复的 `ssoadapter` 元模板。

a. 在 `/tmp/ssoadapter-templates.xml` 文件中查找每个模板。

查找 `default|imap:/?configName=` 字符串。

- b. 将重复的 ssoadapter 元模板名称替换为唯一的值。

例如，如果有两个 SUN-UWC-MAIL ssoadapter 元模板，将其中一个元模板的 configName 值替换为 SUN-UWC-MAIL2，这样就有了两个名称唯一的模板：

```
default|imap:/?configName=SUN-UWC-MAIL ...
default|imap:/?configName=SUN-UWC-MAIL2 ...
```

3. 创建一个 amadmin 请求文件，它将导入已修改的 ssoadapter 元模板，从而覆写原始数据。

- a. 将 /tmp/ssoadapter-templates.xml 复制到 /tmp/ssoadapter-new-templates.xml

- b. 在 /tmp/ssoadapter-new-templates.xml 中，将以下字符串：

```
sunConfigurationTemplates=[
```

替换为：

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE Requests
  PUBLIC "-//iPlanet//iDSAME 5.0 Admin CLI DTD//EN"
  "jar://com/iplanet/am/admin/cli/amAdmin.dtd"
>
<Requests>
  <SchemaRequests serviceName="SunSSOAdapterService"
    SchemaType="Global">
    <ModifyDefaultValues>
      <AttributeValuePair>
        <Attribute name="sunConfigurationTemplates"/>
      </AttributeValuePair>
    </ModifyDefaultValues>
  </SchemaRequests>
</Requests>
```

- c. 将所有“和符号”("&") 替换为 "&"。

例如，下面的行：

```
default|imap:/?configName=SUN-UWC-MAIL
&proxyAdminPassword=%5BPROXY-ADMIN_PASSWORD%5D
&subType=sun-one&enableProxyAuth=false...
```

应变成：

```
default|imap:/?configName=SUN-UWC-MAIL
&amp;proxyAdminPassword=%5BPROXY-ADMIN_PASSWORD%5D
&amp;subType=sun-one&amp;enableProxyAuth=false ...
```

- d. 删除每个 ssoadapter 元模板末尾处的逗号(",")。

- e. 用起始标记 <Value> 和结束标记 </Value> 环绕每个 ssoadapter 元模板。

例如：

```
<Value>default|imap:/?configName=SUN-UWC-MAIL
&amp;proxyAdminPassword=%5BPROXY-ADMIN_PASSWORD%5D
&amp;subType=sun-one&amp;enableProxyAuth=false ...</Value>
```

- f. 从最后一个 ssoadapter 元模板中删除右方括号 ("]")。

- g. 在文件的末尾处添加以下几行：

```
</AttributeValuePair>
</ModifyDefaultValues>
</SchemaRequests>
</Requests>
```

以上步骤中使用的单个模板的最终 ssoadapter-new-templates.xml 文件应如下所示：

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE Requests
  PUBLIC "-//iPlanet//iDSAME 5.0 Admin CLI DTD//EN"
  "jar://com/iplanet/am/admin/cli/amAdmin.dtd"
>
<Requests>
  <SchemaRequests serviceName="SunSSOAdapterService"
    SchemaType="Global">
    <ModifyDefaultValues>
      <AttributeValuePair>
        <Attribute name="sunConfigurationTemplates"/>
        <Value>default|imap:/?configName=SUN-UWC-MAIL
&amp;proxyAdminPassword=%5BPROXY-ADMIN_PASSWORD%5D
&amp;subType=sun-one&amp;enableProxyAuth=false ...</Value>
      </AttributeValuePair>
    </ModifyDefaultValues>
  </SchemaRequests>
</Requests>
```

4. 导入新的 ssoadapter-new-templates.xml 文件。

```
AccessManager-base/bin/amadmin -u amadminUser -w password -v
-t ssoadapter-new-templates.xml
```

此时，您可在 Portal Server 控制台中通过访问 ssoadapter 选项卡来查看已更新的 ssoadapter。

启用 URLScrapper 频道

从发行版本 2 升级到发行版本 5 时，必须启用 URLScrapper 频道。参见第 305 页中的“启用 URLScrapper 频道”。

删除网关服务条目

从发行版本 2 升级 Portal Server 时，必须手动删除 amService-srapGateway 用户条目，否则，Portal Server Secure Remote Access Gateway 组件（如果使用的话）将无法在升级后启动。执行以下步骤：

1. 登录到 Access Manager 控制台。
2. 列出组织 DN 中的所有用户。
3. 删除 amService-srapGateway 用户。

多实例升级

在将发行版本 2 Portal Server 升级到发行版本 5 的过程中，不支持多实例滚动升级（参见第 310 页中的“多实例升级”）。

从过渡功能版本 7.0 升级 Portal Server

本节包含关于将 Portal Server 从过渡功能版本 (Interim Feature Release, IFR) 7.0 2005Q4 升级到 Java ES 5（发行版本 5）的信息。

本节涵盖了以下主题：

- [Portal Server IFR 升级简介](#)
- [Portal Server IFR 7.0 升级](#)
- [多实例升级](#)

Portal Server IFR 升级简介

在将 Portal Server IFR 7.0 升级到发行版本 5 Portal Server 时，请考虑升级过程中的以下方面：

- Application Server 7.x 上不支持 Portal Server IFR，因此，[第 286 页中的表 15-4](#) 中的升级方案 5 不适用。
- 用于将 Portal Server IFR 升级到发行版本 5 的 psupgrade 脚本并不安装新的软件包，这与从发行版本 4 进行升级时的情况一样。升级过程将改为要求您应用以下修补程序：

表 15-8 用于将 Portal Server IFR 升级到发行版本 5 的修补程序¹

说明	修补程序 ID: Solaris 9 和 10	修补程序 ID: Linux
Portal Server 7.1	121465-28 (SPARC)	121467-28
	121466-28 (x86)	
Portal Server 7.1 本地化版本	123254-02 (SPARC)	123255-02
	124590-02 (x86)	

1. 修补程序修订版本号是升级到 Java ES 发行版本 5 所需的最低修订版本号。如果有较新的修订版本可用，则使用较新的修订版本替代该表中所示的版本。

- 对于发行版本 5 Web Server，psupgrade 脚本不支持将 Portal Server IFR 升级到发行版本 5。这一限制隐含了两种含义：
 - 如果 Web Server 已升级到发行版本 5，则不能升级部署在早期 Web Server 容器中的 Portal Server IFR 软件。因此，不支持第 286 页中的表 15-4 中的 Web 容器升级方案 2。请注意第 53 页中的“特殊情况”中的警告事项。
 - 如果 Web Server 尚未升级到发行版本 5，则可以先将早期 Web Server (6.x) 容器中的 Portal Server IFR 软件升级到发行版本 5（如下文的“Portal Server IFR 7.0 升级”中所述），然后将 Web Server（必要时，连同 Access Manager）升级到发行版本 5。如果您采用了这种升级方案，另参见第 337 页中的“发行版本 5 Web Server (7.0) Web 容器中的升级”了解更多升级后说明。

Portal Server IFR 7.0 升级

本节介绍如何在 Solaris 和 Linux 平台上将 Portal Server 从 IFR 升级到 Java ES 发行版本 5。如果其中某个主题取决于具体平台的过程，则该主题将指明所适用的操作系统。本节包括以下主题：

- [IFR 7 升级前的任务](#)
- [升级 Portal Server IFR 7.0 \(Solaris\)](#)
- [升级 IFR 7 Portal Server \(Linux\)](#)
- [检验升级](#)
- [IFR 7 升级后的任务](#)
- [升级回滚 \(Solaris\)](#)
- [升级回滚 \(Linux\)](#)

IFR 7 升级前的任务

IFR 升级的升级前任务与发行版本 4 升级的升级前任务相同（参见第 288 页中的“发行版本 4 升级前的任务”），但存在以下例外：

- [获取所需的配置信息](#)
- [配置 Common Agent Container](#)

获取所需的配置信息

psupgrade 脚本所需的信息（如第 290 页中的“获取所需的配置信息和密码”中所述）不完全适用于从 Portal Server IFR 的升级。由于 Application Server 7.x 不支持 Portal Server IFR，因此，第 290 页中的表 15-6 中的 Web 容器升级方案 5 不适用。

配置 Common Agent Container

Common Agent Container 是一个共享组件，用于为 Java ES 监视和管理代理提供容器服务。Portal Server 管理工具（如 Portal Server 控制台和 psadmin 命令行界面）使用一组部署在 Common Agent Container 中的监视和管理代理（统称为 Portal Administration Server）。

如果在执行 Portal Server IFR 7.0 的升级之前，Java ES 共享组件已升级到发行版本 5，则您需要执行以下附加步骤才能登录到发行版本 5 Portal Server 控制台及使用 psadmin 命令行界面。（如果在执行 Portal Server IFR 7.0 的升级之前，Java ES 共享组件尚未升级到发行版本 5，则忽略以下附加步骤。）

1. 重新配置 Common Agent Container。

```
PortalServer7-base/bin/psconfig --config
PortalServer7-base/samples/example2.xml
```

example2.xml 文件将提供重新配置信息。您必须先编辑 example2.xml 文件以提供必要的密码，然后再运行 psconfig 命令。如果要使用非默认 Portal Server 位置，则还必须提供正确的目录。

2. 编辑 Web 容器的类路径以引用 Common Agent Container。

该 Web 容器的类路径将包含对早期 Common Agent Container 发行版本的位置的引用：(*rel4CAC-base-dir/lib/cacao_cacao.jar*)。

将此引用替换为发行版本 5 的位置：(*rel5CAC-admin-dir/lib/cacao_cacao.jar*)。

3. 检查在 PortalServer7Config-base 目录中是否创建了具有以下属性的属性文件 pasconnect.properties:

```
pas.host=
```

该属性值可为空、localhost 或实际 Portal Server 主机名。

4. 重新启动 Common Agent Container。

```
rel5CAC-admin-dir/bin/cacoadm start
```

升级 Portal Server IFR 7.0 (Solaris)

本节先讨论影响 Portal Server IFR 升级过程的若干注意事项，随后对升级过程本身加以说明。

IFR 7 升级注意事项 (Solaris)

将 Portal Server IFR 升级到 Java ES 发行版本 5 所考虑的注意事项与发行版本 4 升级相同（参见第 293 页中的“升级注意事项 (Solaris)”）。

此外，参见第 326 页中的“Portal Server IFR 升级简介”中所提出的问题。

IFR 7 升级过程 (Solaris)

下述过程适用于要进行升级的计算机中的 Portal Server。

1. 以 root 身份登录，或成为超级用户。

```
su -
```

2. 停止可能在本地运行的任何 Portal Server Secure Remote Access Gateway、Rewriter Proxy 或 Netlet Proxy 实例。

```
PortalServer7-base/bin/psadmin stop-sra-instance -u amadminUser  
-f passwordFile -t gateway -N gatewayProfileName
```

```
PortalServer7-base/bin/psadmin stop-sra-instance -u amadminUser  
-f passwordFile -t rwproxy -N gatewayProfileName
```

```
PortalServer7-base/bin/psadmin stop-sra-instance -u amadminUser  
-f passwordFile -t nlproxy -N gatewayProfileName
```

检查这些进程是否已停止：

```
Gateway: netstat -an | grep 443
```

```
Rewriter Proxy: netstat -an | grep 10443
```

```
Netlet Proxy: netstat -an | grep 10555
```

3. 确保 Access Manager 正在运行，如果它与 Portal Server 部署到不同的 Web 容器中。

4. 如果 Portal Server 尚未运行，请通过启动它所部署到的 Web 容器来启动它。

Web Server 6.x:

```
WebServer-base/https-instanceName/start
```

Application Server 8.x:

```
AppServer8-base/bin/asadmin start-domain --user admin_ID  
--password password domainName
```

5. 根据第 326 页中的表 15-8，获取所需的修补程序。

若未指明使用某个特定修订版本，始终应使用可用的最新修补程序修订版本。

可从以下网址将修补程序下载到 /tmp:

<http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access>

6. 应用相应的 Portal Server 修补程序，如果需要，应用表 15-8 中的本地化修补程序。

```
patchadd /tmp/patch_ID
```

7. 确认修补程序升级成功:

```
showrev -p | grep patch_ID
```

输出应返回在步骤 6 中应用的修补程序 ID 的版本。

8. 在步骤 6 中，如果本地化软件包已升级，将 Portal Server 控制台 JVM 的语言环境设置为 UTF-8。

```
export LC_ALL=ja_JP.UTF-8
```

```
export LANG=ja_JP.UTF-8
```

9. 设置 psupgrade 脚本所需的两个环境变量 (ANT_HOME 和 JAVA_HOME):

```
export ANT_HOME=/usr/sfw
```

```
export JAVA_HOME=/usr/jdk/entsys-j2se
```

10. 确保计算机上有足够的交换空间。

指导原则是交换空间应设置为物理 RAM 大小的两倍。

11. 运行 psupgrade 脚本。

```
cd PortalServer7-base/bin
```

```
./psupgrade
```

psupgrade 脚本并不通过 Java ES 发行版本 5 分发软件运行，而且，并不调用 Java ES 安装程序（软件包已进行修补）。

该脚本会查询系统以检测您要在其中部署 Portal Server Web 应用程序的 Web 容器的位置、端口号以及其他信息。根据 Web 容器升级方案（参见第 286 页中的表 15-4），该脚本会要求您输入将 Portal Server 部署到相应的 Web 容器时所需的其他信息。

第 290 页中的表 15-6 显示了第 286 页中的表 15-4 中不同 Web 容器升级方案所请求的信息。

注	确保输入正确的 psupgrade 参数值，因为您不能返回并更改这些值，并且，对 psupgrade 脚本所做的更改执行回滚操作也是非常困难的。
----------	--

12. 停止并重新启动 Web 容器。

虽然并非在所有情况下都需要重新启动 Web 容器，但这样做可确保 Portal Server 在干净的状态下启动。

升级 IFR 7 Portal Server (Linux)

本节讨论影响 Portal Server 升级过程的一些注意事项，随后对升级过程本身加以说明。

IFR 7 升级注意事项 (Linux)

在 Linux 平台上将 Portal Server IFR 软件升级到发行版本 5 需要考虑的注意事项与在 Solaris 平台上相同（参见第 329 页中的“IFR 7 升级注意事项 (Solaris)”），但在 Linux OS 上安装发行版本 5 修补程序时会删除先前的 RPM。

IFR 7 升级过程 (Linux)

下述过程适用于要进行升级的计算机中的 Portal Server。

注意	在 Linux 上，无法对从 Portal Server IFR 到发行版本 5 的升级执行回滚操作。确保在执行以下过程之前备份您的系统。
-----------	---

1. 以 root 身份登录，或成为超级用户。

su -

2. 停止可能在本地运行的任何 Portal Server Secure Remote Access Gateway、Rewriter Proxy 或 Netlet Proxy 实例。

```
PortalServer7-base/bin/psadmin stop-sra-instance -u amadminUser  
-f passwordFile -t gateway -N gatewayProfileName
```

```
PortalServer7-base/bin/psadmin stop-sra-instance -u amadminUser  
-f passwordFile -t rwproxy -N gatewayProfileName
```

```
PortalServer7-base/bin/psadmin stop-sra-instance -u amadminUser  
-f passwordFile -t nlproxy -N gatewayProfileName
```

检查这些进程是否已停止：

```
Gateway: netstat -an | grep 443
```

```
Rewriter Proxy: netstat -an | grep 10443
```

```
Netlet Proxy: netstat -an | grep 10555
```

3. 确保 Access Manager 正在运行，如果它与 Portal Server 部署到不同的 Web 容器中。
4. 如果 Portal Server 尚未运行，请通过启动它所部署到的 Web 容器来启动它。

Web Server 6.x:

```
WebServer-base/https-instanceName/start
```

Application Server 8.x:

```
AppServer8-base/bin/asadmin start-domain --user admin_ID  
--password password domainName
```

5. 使用第 326 页中的表 15-8 中的修补程序编号和 RPM 名称获取所需的修补程序。
若未指明使用某个特定修订版本，始终应使用可用的最新修补程序修订版本。

可从以下网址将修补程序下载到 /tmp:

<http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access>

6. 应用 Portal Server 修补程序，如果需要，以第 326 页中的表 15-8 中的顺序应用该表中 Portal Server 的本地化 RPM。

参见 Portal Server 修补程序的自述文件，该文件介绍了如何使用脚本应用修补程序的 RPM：

```
cd /tmp
```

其中，/tmp 是修补程序的下载目录。

```
./upgradportalrpm
```

更新脚本将安装 RPM。

对于本地化修补程序，使用以下命令安装每个 RPM：

```
rpm -Fvh patchName-version.rpm
```

7. 确认修补程序升级成功：

```
rpm -qa | grep sun-portal
```

应返回 RPM 的升级修订版本号。

8. 在步骤 6 中，如果本地化软件包已升级，将 Portal Server 控制台 JVM 的语言环境设置为 UTF-8。

```
export LC_ALL=ja_JP.UTF-8
```

```
export LANG=ja_JP.UTF-8
```

9. 设置 psupgrade 脚本所需的两个环境变量（ANT_HOME 和 JAVA_HOME）：

```
export ANT_HOME=/opt/sun
```

```
export JAVA_HOME=/usr/jdk/entsys-j2se
```

10. 确保计算机上有足够的交换空间。

指导原则是交换空间应设置为物理 RAM 大小的两倍。

11. 运行 psupgrade 脚本。

```
cd PortalServer7-base/bin
```

```
./psupgrade
```

psupgrade 脚本并不通过 Java ES 发行版本 5 分发软件运行，而且，并不调用 Java ES 安装程序（软件包已进行修补）。

该脚本会查询系统以检测您要在其中部署 Portal Server Web 应用程序的 Web 容器的位置、端口号以及其他信息。根据 Web 容器升级方案（参见第 286 页中的表 15-4），该脚本会要求您输入将 Portal Server 部署到相应的 Web 容器时所需的其他信息。

第 290 页中的表 15-6 显示了第 286 页中的表 15-4 中不同 Web 容器升级方案所请求的信息。

注 确保输入正确的 `psupgrade` 参数值，因为您不能返回并更改这些值，并且，对 `psupgrade` 脚本所做的更改执行回滚操作也是非常困难的。

12. 停止并重新启动 Web 容器。

虽然并非在所有情况下都需要重新启动 Web 容器，但这样做可确保 Portal Server 在干净的状态下启动。

检验升级

可使用以下命令检验是否将 Portal Server 软件包修补到发行版本 5:

```
PortalServer7-base/bin/psadmin --version --adminuser admin_ID
-f adminpasswordfile
```

参见第 289 页中的表 15-5 确认输出值。

要检验完全升级，请确认 Portal 桌面出现且 `psadmin` 管理实用程序按照文档所述的那样起作用。

还可以检查 `/var/sadm/install/logs` 中的以下升级日志文件：

- `Sun_Java_System_Portal_Server_upgrade.log`
- `Sun_Java_System_Portal_Server_upgrade.log_ant_xxx.log`

其中，`xxx` 可以为 `preupgrade`、`upgrade` 或 `postupgrade`。

IFR 7 升级后的任务

将 Portal Server 从 IFR 7 升级到发行版本 5 时，必须执行处理以下情况所必需的升级后过程：

- 启用 Java ES Monitoring Framework
- Application Server Web 容器中的升级
- 发行版本 5 Web Server (7.0) Web 容器中的升级
- 重新部署自定义 Portlet 应用程序
- 迁移自定义 Portlet 应用程序
- 更正书签和应用程序频道中的链接
- 手动迁移基于 Struts 的 Portlet

启用 Java ES Monitoring Framework

如果启用 Java ES Monitoring Framework (MFWK 共享组件)，Portal Server 管理工具（如 Portal Server 控制台和 psadmin 命令行界面）便能够报告诸如访问者数量及他们经常访问的 portal 等统计信息。启用 MFWK：

1. 找到以下两个文件：

```
MFWK-base/template/jesmf/desktopmfwk.properties
MFWK-base/template/jesmf/com.sun.cmm.ps.xml
```

其中，MFWK-base 为以下路径：

```
/opt/SUNWmfwk      (Solaris)
/opt/sun/mfwk       (Linux)
```

2. 将这两个文件复制到以下目录中：

```
PortalServer7Data-base/portals/portal_ID/config/Portal_Instance/
```

3. 在 desktopmfwk.properties 文件中，将

```
com.sun.portal.ProductCollectionId=%PS_DIR%
```

替换为

```
com.sun.portal.ProductCollectionId=Portal_Installed_Location
```

4. 将以下两个 jar 文件:

```
MFWK-base/lib/mfwk_instrum_tk.jar  
MFWK-base/lib/mfwk_agent.jar
```

添加到相应的 Web 容器类路径中（对于 Web Server 为 *server.xml* 文件，对于 Application Server 为 *domain.xml*）

5. 重新启动相应的 Web 容器。

Application Server Web 容器中的升级

如果 Portal Server 已部署在 Application Server Web 容器中，则必须执行以下附加过程才能成功地重新部署 Portal Server:

1. 在 *AppServer8Config-base/domains/domainName/config/server.policy* 文件中找到定义 *psconsole* 设置的那部分:
2. 在该部分的末尾添加下行:

```
permission java.lang.RuntimePermission "getProtectionDomain"
```

3. 重新启动 Application Server 实例。

```
AppServer8-base/bin/asadmin start-domain --user admin_ID  
--password passworddomainName
```

```
AppServer8-base/bin/asadmin start-node-agent --user admin_ID  
--password password nodeagentName
```

其中，*nodeagentName* 的格式为 *hostName_domainName*，但默认情况下只是 *hostName*。

Application Server Web 容器中的连接挂起

如果已升级的 Portal Server 已部署在 Application Server Web 容器中，portal 应用程序可能会挂起，以等待获得 Java DB 连接。要解决此问题，请执行以下步骤:

1. 在 *PortalServer7Data-base/derby/derby.properties* 中删除以下 2 个参数的设置:

```
derby.drda.maxThreads  
derby.drda.timeslice
```

2. 重新启动 Java DB。

```
ANT_HOME/bin/ant
-DPS_CONFIG=PortalServer7Config-base/PSConfig.properties
-buildfile PortalServer7-base/lib/derby.xml
[stop-instance|start-instance]
```

其中，ANT_HOME 为 /usr/sfw（在 Solaris 上）和 /opt/sun（在 Linux 上）。

3. 更改 Application Server 的 Java DB 配置设置。

使用 Application Server 控制台，更改以下连接池资源的属性值：
communitymcPool、FileSharingDBPool、PointBasePool、SurveyDBPool。

如下所示更改以下属性值：

```
将 Idle Timeout 更改为 300 或更大
将 Resource Type 更改为 javax.sql.ConnectionPoolDataSource
将 Datasource classname 更改为
    org.apache.derby.jdbc.ClientConnectionPoolDataSource
```

4. 重新启动在其中部署了 Portal Server 的 Application Server 实例。

发行版本 5 Web Server (7.0) Web 容器中的升级

如果您在尚未升级到发行版本 5 的 Web Server 中升级了 Portal Server IFR（因为 psupgrade 脚本不支持在发行版本 5 Web Server 中升级 Portal Server IFR；参见第 326 页中的“Portal Server IFR 升级简介”），则必须执行以下附加的升级后步骤：

1. 将 Web Server（必要时，连同 Access Manager）升级到发行版本 5。
2. 重新配置 Portal Server 控制台和 psadmin 命令行界面所需的 Web Server 容器值。
 - a. 打开 LDAP 浏览器。
配置值存储在 Directory Server 中。
 - b. 在 DN 下，查找：


```
sunPortalAdminPortalDomainID=defaultDomain
->sunPortalAdminPortalDomainPortalID=portal1
->sunPortalAdminPortalDomainPortalServerInstanceIn=host-port
```

c. 按照以下过程进行编辑。

注意，所有条目的开头均为以下字符串：

`sunPortalAdminPortalDomainPortalServerInstance`

- 删除 `WebContainerInstanceDir` 条目。
- 添加 `WebContainerDomainName` 条目，并将第 290 页中的表 15-6 中 Web 容器配置名称的值指定给该条目。
- 编辑 `InstallDir`、`WebContainerType`、`DocRoot` 等条目以及第 290 页中的表 15-6 中所示的其他参数，使它们与发行版本 5 Web Server (7.0) 的各个值一致。

3. 创建发行版本 5 Portal Server 实例。

```
PortalServer7-base/bin/psadmin create-instance newInstance_ID
```

如果 `newInstance_ID` 的值已存在，则会抛出一个错误，所以，在下面的步骤 4 之前执行此步骤是很有用的。

4. 删除 Portal Server IFR 实例。

```
PortalServer7-base/bin/psadmin delete-instance oldInstance_ID
```

重新部署自定义 Portlet 应用程序

如果您创建和部署了自定义 portlet 应用程序，则在升级到发行版本 5 Portal Server 之后必须手动重新部署这些 portlet。即使显示配置文件条目存在且频道名称将会显示出来，您也不会看到内容，直到重新部署自定义 portlet 为止。

使用以下命令重新部署 portlet：

```
PortalServer7-base/bin/psadmin deploy-portlet
```

可通过在以下位置查找对应的 .war 和 XML 文件来确认重新部署：

```
PortalServer7Data-base/portals/Upgraded/war
```

迁移自定义 Portlet 应用程序

对于基于由 Sun Java Web Console (SJWC) 提供的用户界面框架的 portlet 应用程序，您需要手动将其迁移至发行版本 5 并进行重新部署。

具体而言，这种需求适用于四个 Web 应用程序：`filesharing`、`surveys`、`wiki` 和 `rssportlet`，它们作为要自定义并安装到 portal 中的样例 portlet 应用程序随 Portal Server 一起分发。升级期间，这些样例 portlet 应用程序的已修复错误版本置于磁盘

中的 portlet 应用程序区域。如果您已自定义这些 portlet 应用程序供您自己使用，则需要将它们手动迁移至发行版本 5 并重新进行部署；Portal Server 升级过程不会处理它们。

默认情况下，其中的某些 portlet 应用程序（filesharing 和 surveys）已部署且在创建了社区时可供用户使用。

可使用以下过程升级、自定义和重新部署基于 SJWC 的 portlet 应用程序。其中，以 filesharing portlet 应用程序为例：

1. 解压缩发行版本 5 SJWC jar 文件。

- a. `mkdir /tmp/lh`
- b. `cd /tmp/lh`
- c. `/usr/jdk/entsys-j2se/bin/jar xvf`
`PortalServer7-base/portlet/communityportlets.war`
`WEB-INF/lib/commons-beanutils.jar`
`WEB-INF/lib/commons-collections-3.1.jar`
`WEB-INF/lib/commons-digester.jar`
`WEB-INF/lib/commons-logging.jar`
`WEB-INF/lib/dataprovider.jar WEB-INF/lib/jsf-api.jar`
`WEB-INF/lib/jsf-impl.jar WEB-INF/lib/webui.jar`

如果未找到 `PortalServer7-base/portlet/communityportlets.war`，则使用 `PortalServer7-base/portlet/core/communityportlets.war`。

d. 重命名其中一个文件。

```
mv WEB-INF/lib/commons-collections-3.1.jar
WEB-INF/lib/commons-collections.jar
```

2. 找到 filesharing portlet 应用程序。

```
cd PortalServer7Config-base/portals/portall/portletapps/filesharing
```

3. 插入经过更新的 SJWC 库。

```
jar uvf src/filesharing.war.tokenized -C /tmp/lh WEB-INF
```

4. 自定义 filesharing portlet 应用程序。

```
ant customize
```

5. 重新部署 filesharing portlet 应用程序。
 - a. `PortalServer7-base/bin/psadmin undeploy-portlet -u amadmin -f passwordfile -p portal_id -i instance_id -g filesharing`
 - b. `ant deploy`
 - c. 转到以下目录（视 Web 容器而定）：
Application Server 8.x:
`AppServer8Config-base/domains/domain1/applications/j2ee-modules/communityportlets/WEB-INF`
Web Server 6.x:
`WebServer6-base/https-instanceName/webapps/https-instanceName/communityportlets/WEB-INF`
Web Server 7.x:
`WebServer7Config-base/https-configName/web-app/https-configName/communityportlets/WEB-INF`
 - d. 打开 `sun-web.xml` 文件并在最后一行之前（即在 `sun-web-app` 结束标记之前）添加以下行：
`<class-loader delegate="false"/>`
 - e. 对 `filesharing/WEB-INF` 下的 `sun-web.xml` 文件，重复步骤 c 和步骤 d。
6. 对 `surveys` 以及基于 SJWC 框架的任何其他自定义 portlet 应用程序，重复步骤 2 到步骤 5。
7. 重新启动 Web 容器。

更正书签和应用程序频道中的链接

将发行版本 4 Portal Server 升级到发行版本 5 时，书签和应用程序频道中存在许多重复和虚假的链接。要修复这些链接，执行以下过程。

1. 登录到 PS 控制台。
2. 在“常用任务”选项卡上，单击“管理频道和容器”。
3. 选择 "DeveloperSample [Org]" 作为 DN，然后单击“确定”。
4. 选择“JSPTabContainer [默认值]”作为“视图类型”。

5. 在 `MyFrontPageTabPanelContainer` 下，单击“应用程序频道”。

应用程序频道属性将显示在右框中。

要针对某个具体语言环境查看属性，单击“表首选项”，随后将提供“语言环境”值：de、fr、es、ja、ko、zh、zh_CN、zh_TW。

6. 编辑 `userApps` 属性。

- a. 在 `userApps` 属性中，单击 “[编辑值 ...]” 链接。

将出现一个弹出式窗口，其中显示了现有应用程序。

- b. 将以下应用程序添加到列表中：

`NetFile`

- c. 单击“保存”，然后单击“关闭”。

7. 编辑 `target` 属性。

- a. 在 `target` 属性中，单击 “[编辑值 ...]” 链接。

将出现一个弹出式窗口，其中显示了现有目标。

- b. 将以下目标添加到列表中：

`NetFile|/portal/NetFileApplet?Refer=java2`

- c. 单击“保存”，然后单击“关闭”。

手动迁移基于 Strut 的 Portlet

如果您的自定义 portlet 使用基于 Struts 框架的代码，则这些 portlet 需要手动更新为使用发行版本 5 Portal Server 中随附的 `struts.jar` 文件。按照以下过程进行操作：

1. 取消部署基于 strut 的 portlet 应用程序。

```
PortalServer7-base/bin psadmin undeploy-portlet
```

2. 将 `.war` 文件更新为正确版本的 `struts.jar` 文件。

将 `PortalServer7-base/lib/struts.jar` 复制到
`strutsbasedPortlet/WEB-INF/lib/struts.jar`

其中，`strutsbasedPortlet` 是基于 strut 的 portlet 文件所驻留的目录。

3. 创建 `strutsbasedPortlet` 目录的 `.war` 归档。

4. 重新部署 portlet 应用程序。

```
PortalServer7-base/bin psadmin deploy-portlet
```

升级回滚 (Solaris)

本节介绍影响 Portal Server 升级回滚过程的一些注意事项，随后对升级回滚过程本身加以说明。

回滚注意事项 (Solaris)

发行版本 5 升级的回滚过程包括在 *PortalServer7-base* 处恢复为 IFR 安装以及重新部署 IFR Web 应用程序。

回滚过程 (Solaris)

1. 以 root 身份登录，或成为超级用户。

```
su -
```

2. 将 Directory Server 恢复为升级前的状态。

使用 Directory Server 备份 / 恢复命令行和 GUI 实用程序。参见《Sun Java System Directory Server Enterprise Edition 6.0 管理指南》(<http://docs.sun.com/doc/820-0293>) 中的“目录服务器备份和恢复”一章。

3. 取消部署那些在升级到发行版本 5 期间重新部署的发行版本 5 Portal Server Web 应用程序。

使用 Web 容器的管理实用程序（命令行或控制台）取消部署以下软件包：

```
portal  
psconsole  
search1  
wsssoportlet  
guessnumber  
portletsamples
```

4. 通过停止其 Web 容器，停止 Portal Server。

Web Server 6.x:

```
WebServer-base/https-instanceName/stop
```

Web Server 7.0:

管理服务器 --

```
WebServer7Config-base/admin-server/bin/stopserv
```

实例服务器 --

```
WebServer7Config-base/https-configName/bin/stopserv
```

Application Server 8.x:

```
AppServer8-base/bin/asadmin stop-domain --user admin_ID  
--password password domainName
```

5. 回退[表 15-8](#) 中的 Portal Server 7.1 修补程序。

```
patchrm patch_ID
```

6. 通过重新启动其 Web 容器，重新启动 Portal Server。

Web Server 6.x:

```
WebServer-base/https-instanceName/start
```

Web Server 7.0:

管理服务器 --

```
WebServer7Config-base/admin-server/bin/startserv
```

实例服务器 --

```
WebServer7Config-base/https-configName/bin/startserv
```

Application Server 8.x:

```
AppServer8-base/bin/asadmin start-domain --user admin_ID
```

```
--password password domainName
```

7. 部署那些在[第 342 页中的步骤 3](#) 中取消部署的发行版本 5 Portal Server Web 应用程序。

使用 Web 容器的管理实用程序（命令行或控制台）部署这些软件包。

8. 停止并重新启动 Web 容器。

虽然并非在所有情况下都需要重新启动 Web 容器，但这样做可确保 Portal Server 在干净的状态下启动。

升级回滚 (Linux)

在 Linux 上不能执行升级回滚。

不过，您可以在尝试进行升级之前，先创建一个并行系统，然后对该系统进行升级和测试。如果需要对升级执行回滚操作，可恢复为该并行系统。

多实例升级

在一些部署体系结构中，Portal Server 被部署在多个计算机系统上，以提供可伸缩性和提高可用性。例如，您可以在多台计算机上运行 Portal Server 组件，并使用一个负载均衡器来分配负载。

就 Portal Server 负载均衡的实例而言，可执行滚动升级，在此期间可依序升级各个 Portal Server 实例而不中断服务。在升级每个 Portal Server 实例时，其他实例仍可保持运行。要执行从 IFR 到发行版本 5 的滚动升级，可采用[第 310 页中的“多实例升级”](#) 中所述的同一过程，但需要在提到发行版本 4 的地方用 IFR 代替。

Portal Server Secure Remote Access

本章介绍如何将 Portal Server Secure Remote Access 升级到 Java ES 5（发行版本 5）：
Sun Java System Portal Server Secure Remote Access 7.1。

本章概要介绍了有关发行版本 5 所支持的不同升级方式的升级注意事项。其内容涵盖了在 Solaris 和 Linux 操作系统上的升级：

- 第 347 页中的“Portal Server Secure Remote Access 升级概述”
- 第 352 页中的“从 Java ES 发行版本 4 升级 Portal Server Secure Remote Access”
- 第 369 页中的“从 Java ES 发行版本 3 升级 Portal Server Secure Remote Access”
- 第 371 页中的“从 Java ES 发行版本 2 升级 Portal Server Secure Remote Access”
- 第 374 页中的“从过渡功能版本 7.0 升级 Portal Server Secure Remote Access”

注 本章中的文件位置是相对于称为 *PortalServer6-base* 与 *PortalServer6Config-base* (Portal Server 6.x) 以及 *PortalServer7-base* 与 *PortalServer7Config-base* (Portal Server 7.x) 的目录路径指定的。在最初安装 Portal Server 时，可能至少已将这些路径的一部分指定为安装目录，否则，Java ES 安装程序会分配一个默认值。

下表说明了这些目录路径的默认值。

表 16-1 Portal Server Secure Remote Access 目录路径

路径名变量	Solaris OS	Linux OS
<i>PortalServer6-base</i>	/opt/SUNWps	/opt/sun/portal
<i>PortalServer6Config-base</i>	/etc/opt/SUNWps	/etc/opt/sun/portal
<i>PortalServer6Data-base</i>	/var/opt/SUNWps	/var/opt/sun/portal
<i>ortalServer7-base</i>	/opt/SUNWportal	/opt/sun/portal
<i>PortalServer7Config-base</i>	/etc/opt/SUNWportal	/etc/opt/sun/portal
<i>PortalServer7Data-base</i>	/var/opt/SUNWportal	/var/opt/sun/portal

Portal Server Secure Remote Access 升级概述

本节将在以下方面对 Portal Server Secure Remote Access 进行介绍，这些方面会对升级到 Java ES 5（发行版本 5）产生影响：

- [关于 Java ES 发行版本 5 Portal Server Secure Remote Access](#)
- [Portal Server Secure Remote Access 升级指导说明](#)
- [Portal Server Secure Remote Access 数据](#)
- [Portal Server Secure Remote Access 升级策略](#)

关于 Java ES 发行版本 5 Portal Server Secure Remote Access

尽管部署 Portal Server Secure Remote Access（包括 Gateway、Rewriter Proxy、Netlet Proxy 组件）的计算机通常不是托管 Portal Server 的计算机，但 Portal Server Secure Remote Access 仍与 Portal Server 紧密耦合。Portal Server Secure Remote Access 组件与 Portal Server 使用相同的管理基础结构，并与托管 Portal Server 的计算机上所驻留的 servlet 和 applet 进行交互。

相对于发行版本 4 而言，Java ES 发行版本 5 Portal Server Secure Remote Access 是一个主要发行版本，包含了许多增强功能和新增功能。其中的许多更改都是继发行版本 4 之后在过渡功能版本 (Interim Feature Release, IFR) 中做出的。相对于 IFR，发行版本 5 只进行了较少的功能更改。有关 IFR 增强功能和新增功能的信息，参见《Sun Java System Portal Server 7.1 发行说明》(<http://docs.sun.com/app/docs/doc/820-0897/6ncjtcci5?a=view>)。需特别指出的是，发行版本 4 命令行管理界面已替换为 psadmin 命令。

Portal Server Secure Remote Access 升级指导说明

表 16-2 显示了所支持的、将 Portal Server Secure Remote Access 升级到 Java ES 发行版本 5 的升级方式。此表既适用于 Solaris，也适用于 Linux 操作系统。

表 16-2 升级到 Java ES 5（发行版本 5）：Portal Server Secure Remote Access 7.1 的方式

Java ES 发行版本	Portal Server Secure Remote Access 版本	常规方法	必需的重新配置
过渡功能版本 (Interim Feature Release, IFR)	Sun Java System Portal Server Secure Remote Access IFR 7.0 2005Q4	直接升级： 通过应用修补程序然后使用升级脚本来执行。	无。
发行版本 4	Sun Java System Portal Server Secure Remote Access 6.3.1 2005Q4	直接升级： 使用升级脚本执行。	无。
发行版本 3	Sun Java System Portal Server Secure Remote Access 6.3.1 2005Q1	直接升级： 使用升级脚本执行。	无。
发行版本 2	Sun Java System Portal Server Secure Remote Access 6.3 2004Q2	直接升级： 使用升级脚本执行。	无。
发行版本 1	Sun ONE Portal Server Secure Remote Access 6.2 (2003Q4)	不可直接升级： 但可通过先升级到发行版本 3 然后从发行版本 3 升级到发行版本 5 来执行。	配置数据
Java ES 以前的发行版本		不可直接升级。	

Portal Server Secure Remote Access 数据

下表显示了可能受 Portal Server Secure Remote Access 软件升级影响的数据类型。

表 16-3 Portal Server Secure Remote Access 数据用途

数据类型	位置	用途
配置数据	PortalServer6Config-base/	Portal Server Secure Remote Access 的配置。
目录模式 服务配置 用户数据	Directory Server	Portal Server Secure Remote Access 依赖于服务配置（如门户桌面）和存储在目录中的用户配置文件数据。
动态应用程序数据	无	Portal Server Secure Remote Access 并不永久存储应用程序数据（如会话状态）。

Portal Server Secure Remote Access 升级策略

Portal Server Secure Remote Access 的升级策略通常取决于在第 1 章“升级规划”中讨论的许多注意事项：升级方式、Java ES 组件之间的依赖性、选择性升级与全部升级、多实例部署，等等。

本节通过提出可能影响 Portal Server Secure Remote Access 升级计划的问题，对 Portal Server Secure Remote Access 的一般注意事项进行逐一介绍。

兼容性问题

发行版本 5 Portal Server Secure Remote Access 在用于启动和停止 Gateway、Rewriter Proxy 和 Netlet Proxy 组件的 psadmin 命令中引入了公共接口更改。参见《Sun Java System Portal Server 7.1 Command-Line Reference》(<http://docs.sun.com/doc/819-5030>)。

单独的 Portal Server Secure Remote Access 组件（包括 Gateway、Rewriter Proxy 和 Netlet Proxy）与早期版本不向后兼容；它们都需要与 Portal Server 一起在 Java ES 发行版本 5 进行同步。此要求适用于 Portal Server Secure Remote Access 本地组件和分布式组件。

此外，发行版本 5 Portal Server 与早期版本 Portal Server 所使用的 Directory Server 数据结构之间存在着不兼容性。这种不兼容性影响了使用相同 Directory Server 数据对多个 Portal Server 实例执行的滚动升级。

Portal Server Secure Remote Access 组件依赖性

Portal Server Secure Remote Access 与 Portal Server 紧密耦合，这取决于与 Portal Server 一起打包以及与 Portal Server 在同一台计算机上运行的软件。

但是，Portal Server Secure Remote Access 还依赖于其他 Java ES 组件。这些依赖性会影响升级和重新配置 Portal Server Secure Remote Access 软件的过程。例如，Portal Server Secure Remote Access 接口或功能上的变化可能要求使用 Portal Server Secure Remote Access 所依赖组件的升级版本。是否需要升级此类组件取决于具体的升级方式。

Portal Server Secure Remote Access 组件对以下 Java ES 组件具有依赖性：

- **共享组件。** Portal Server Secure Remote Access 组件对特定的 Java ES 共享组件具有依赖性（参见第 43 页中的表 1-9）。
- **Portal Server。** Portal Server Secure Remote Access 组件对 Portal Server（包括支持 Portal Server Secure Remote Access 功能所需的本地组件）具有强制依赖性。
- **Access Manager（或 Access Manager SDK）。** Portal Server Secure Remote Access 组件对 Access Manager 具有强制依赖性，从而为最终用户提供验证和授权服务（包括单点登录）。如果 Access Manager 运行在远程计算机上，则 Access Manager SDK 必须在本地可用。
- **Directory Server。** Portal Server Secure Remote Access 对 Directory Server 具有强制依赖性，后者用来存储用户数据。因此，Portal Server Secure Remote Access 升级可能需要扩展目录模式。

选择性升级问题

一般而言，Java ES 发行版本 5 支持对计算机上的所有组件执行选择性升级，但由于 Portal Server Secure Remote Access 与 Portal Server 紧密结合，因此，如果升级 Portal Server，则必须升级 Portal Server Secure Remote Access。同样，如果升级 Portal Server Secure Remote Access，则还需要升级 Portal Server。

因此，Portal Server Secure Remote Access 与 Portal Server 的升级限制相同（参见 Portal Server 第 285 页中的“选择性升级问题”）：可以将 Portal Server Secure Remote Access 及其所有产品组件依赖项都升级到发行版本 5，也可以仅将 Portal Server Secure Remote Access 和 Portal Server 升级到发行版本 5，而将其他产品组件依赖项保留为发行版本 4。

双重升级

在双重升级中，Portal Server Secure Remote Access 和操作系统都要进行升级（如第 41 页中的“双重升级：Java ES 和操作系统软件”中所述）。可使用操作系统就地升级方法执行双重升级：

1. 备份现有 Portal Server Secure Remote Access 数据。

有关基本数据的位置，参见第 349 页中的“Portal Server Secure Remote Access 数据”。

2. 升级操作系统。

升级会将现有文件系统留在原来的位置。

3. 将 Portal Server Secure Remote Access 升级到发行版本 5。

参见本章中的相应小节，这要视升级方式而定。

从 Java ES 发行版本 4 升级 Portal Server Secure Remote Access

本节包含关于将 Portal Server Secure Remote Access 从 Java ES 2005Q4（发行版本 4）升级到 Java ES 5（发行版本 5）的信息。

本节涵盖了以下主题：

- [简介](#)
- [发行版本 4 Portal Server Secure Remote Access 的升级](#)
- [多实例升级](#)

简介

将 Java ES 发行版本 4 Portal Server Secure Remote Access 升级到发行版本 5 时，请考虑升级过程中的以下方面：

- **常规升级方法。** 使用升级脚本 `psupgrade` 执行升级。该脚本会删除旧软件包、安装新软件包并在必要时迁移配置数据。
- **升级依赖性。** Portal Server Secure Remote Access 对许多 Java ES 共享组件都具有依赖性（参见第 43 页中的表 1-9）。尽管发行版本 5 Portal Server Secure Remote Access 与发行版本 4 版本的这些共享组件兼容，但仍需要升级共享组件，因为用于升级 Portal Server Secure Remote Access 的 `psupgrade` 脚本要求发行版本 5 版本的 ANT 共享组件。

发行版本 5 Portal Server Secure Remote Access 还依赖于 Portal Server、Access Manager 和 Directory Server，如第 350 页中的“[Portal Server Secure Remote Access 组件依赖性](#)”中所述。支持使用两种方法来升级这些依赖项（参见第 350 页中的“[选择性升级问题](#)”）：

- **升级依赖性。** Portal Server Secure Remote Access 对许多 Java ES 共享组件都具有依赖性（参见第 43 页中的表 1-9），但发行版本 5 Portal Server Secure Remote Access 与发行版本 4 版本的这些组件兼容。因此，对于 Portal Server Secure Remote Access 到发行版本 5 的升级，这些共享组件的升级是可选的。

但是，发行版本 5 Portal Server Secure Remote Access 仅对 Portal Server 具有硬性升级依赖性。发行版本 5 Portal Server Secure Remote Access 还对 Access Manager 和 Directory Server 具有软性升级依赖性，如第 350 页中的“[Portal Server Secure Remote Access 组件依赖性](#)”中所述。

支持两种升级这些产品组件依赖项的方法（参见第 350 页中的“选择性升级问题”）：

- 发行版本 4 组件满足的所有依赖性（除了 Portal Server 外，均未升级到发行版本 5）
- 发行版本 5 组件满足的所有依赖性（全部都升级到发行版本 5）。

Portal Server Secure Remote Access 采用的方法必须与 Portal Server 采用的方法相同。

- **向后兼容性。** 发行版本 5 的 Portal Server Secure Remote Access 与发行版本 4 的版本向后兼容。
- **升级回滚。** 从 Portal Server Secure Remote Access 发行版本 5 到发行版本 4 的升级回滚包括恢复发行版本 4 软件包和恢复发行版本 4 目录数据。
- **平台问题。** 对于 Solaris 和 Linux 操作系统，升级 Portal Server Secure Remote Access 的常规方法是相同的，但是，发行版本 5 Portal Server Secure Remote Access 在 Solaris OS 上是安装在新路径中，而在 Linux OS 上是安装在原来的发行版本 4 路径中。

发行版本 4 Portal Server Secure Remote Access 的升级

本节介绍如何在 Solaris 和 Linux 平台上将 Portal Server Secure Remote Access 从 Java ES 发行版本 4 升级到 Java ES 发行版本 5。如果其中某个主题取决于具体平台的过程，则该主题将指明所适用的操作系统。本节包括以下主题：

- [发行版本 4 升级前的任务](#)
- [升级发行版本 4 Portal Server Secure Remote Access \(Solaris\)](#)
- [升级发行版本 4 Portal Server Secure Remote Access \(Linux\)](#)
- [检验升级](#)
- [发行版本 4 升级后的任务](#)
- [升级回滚 \(Solaris\)](#)
- [升级回滚 \(Linux\)](#)

发行版本 4 升级前的任务

升级 Portal Server Secure Remote Access 之前，应执行以下任务：

- [检验当前版本信息](#)
- [升级 Portal Server Secure Remote Access 依赖项](#)
- [备份发行版本 4 Portal Server Secure Remote Access 配置信息](#)
- [删除负载均衡器的配置](#)
- [删除 Directory Proxy Server 的配置](#)
- [获取所需的配置信息和密码](#)

检验当前版本信息

可使用以下命令检验 Portal Server Secure Remote Access 的当前版本：

```
PortalServer6-base/bin/version
```

表 16-4 Portal Server Secure Remote Access 版本检验输出

Java ES 发行版本	Portal Server Secure Remote Access 版本号
发行版本 2	6.3
发行版本 3	6.3.1
发行版本 4	6.3.1 ¹
IFR 发行版本	7.0
发行版本 5	7.1

1. 发行版本 3 与发行版本 4 的不同之处仅在于修补程序。可通过使用 `showrev -p | grep patch_ID` (Solaris) 命令和 `rpm -qa sun-portal-core` (Linux) 命令并将版本与 Java ES 发行版本 4 升级指南中列出的那些版本相比较来检查是否为发行版本 4 修补程序。

升级 Portal Server Secure Remote Access 依赖项

通常建议将计算机系统（以及计算环境）中的所有 Java ES 组件都升级到 Java ES 发行版本 5。

虽然发行版本 5 Portal Server Secure Remote Access 与发行版本 4 版本的 Java ES 共享组件兼容，但仍需要升级共享组件，因为用于升级 Portal Server Secure Remote Access 的 `psupgrade` 脚本要求发行版本 5 版本的 ANT 共享组件。

此外，Portal Server Secure Remote Access 需要升级 Portal Server。但是，它不需要升级其所依赖的其他 Java ES 发行版本 4 产品组件。

事实上，您的依赖项升级方法与 Portal Server 采用的方法相同：如果将任意依赖项升级到发行版本 5，则所有依赖项都需要进行升级（参见第 285 页中的“选择性升级问题”）。但由于 Portal Server Secure Remote Access 依赖于 Portal Server，因此升级 Portal Server 时还要顾及到 Portal Server Secure Remote Access 依赖项（共享组件除外）。

将 Portal Server Secure Remote Access 依赖项升级到发行版本 5 时，在升级 Portal Server Secure Remote Access 之前，应按照下面的顺序升级依赖项（跳过任何可能已升级的依赖项）。

1. **共享组件。** 第 59 页中的“升级 Java ES 共享组件”提供了将 Java ES 共享组件同步到发行版本 5 的说明。
2. **Portal Server。** 第 279 页中的第 15 章“Portal Server”提供了升级 Portal Server 的说明。

备份发行版本 4 Portal Server Secure Remote Access 配置信息

将 Portal Server Secure Remote Access 升级到发行版本 5 时，无需重新配置 Portal Server Secure Remote Access 软件。但是，为安全起见，psupgrade 脚本将备份其中存储有配置信息的下列目录：

PortalServer6Config-base/

删除负载均衡器的配置

通过负载均衡器访问 Portal Server Secure Remote Access 实例时，用于配置此类访问的 LOAD_BALANCER_URL 属性的值可能会干扰 Portal Server Secure Remote Access 升级。因此，必须在升级任何 Portal Server Secure Remote Access 组件前修改此设置。要修改 LOAD_BALANCER_URL 属性设置，请执行下列步骤：

1. 记录下列哪些配置文件驻留在本地（其中有些文件支持那些可能在本地安装的 Portal Server 组件）：
 - PortalServer6Config-base/PSConfig.properties* （如果 Portal Server 在本地）
 - PortalServer6Config-base/GWConfig.properties* （如果 Gateway 在本地）
 - PortalServer6Config-base/RWPCConfig.properties* （如果 Rewriter Proxy 在本地）
 - PortalServer6Config-base/NLPConfig.properties* （如果 Netlet Proxy 在本地）
2. 记录这些配置文件中 LOAD_BALANCER_URL 属性的当前值。
3. 修改 LOAD_BALANCER_URL 属性的值，使其指向要升级的对应 Portal Server Secure Remote Access 实例：

`LOAD_BALANCER_URL=hostName:port/portal`

4. 确保以下配置属性（如果存在）引用相关 Portal Server Secure Remote Access 组件（而非负载平衡器），如下所示：

在 *PortalServer6Config-base/platform.conf.default* 文件中：

```
gateway.host=Gateway_hostName
```

在 *PortalServer6Config-base/GWConfig.properties* 和

PortalServer6Config-base/GWConfig-default.properties 文件中：

```
GW_HOST=Gateway_hostName
```

```
GW_IP=Gateway_hostIP
```

在 *PortalServer6Config-base/RWPConfig.properties* 和

PortalServer6Config-base/RWPConfig-default.properties 文件中：

```
RWP_HOST=RewriterProxy_hostName
```

```
RWP_IP=RewriterProxy_hostIP
```

在 *PortalServer6Config-base/NLPConfig.properties* 和

PortalServer6Config-base/NLPConfig-default.properties 文件中：

```
NLP_HOST=NetLetProxy_hostName
```

```
NLP_IP=NetLetProxy_hostIP
```

删除 Directory Proxy Server 的配置

如果 Portal Server Secure Remote Access 实例通过 Directory Proxy Server 实例访问 Directory Server，则必须在执行升级前修改 Directory Proxy Server 的主机和端口号设置，并在升级完成后将其恢复为原始值。

修改相应的设置：

1. 记录以下 Access Manager 配置文件中 DS_HOST 和 DS_PORT 属性的当前值：

AccessManagerConfig-base/config/AMConfig.properties

2. 修改 DS_HOST 和 DS_PORT 属性的值，使其直接指向相关 Directory Server 实例。

获取所需的配置信息和密码

根据升级方案的不同，psupgrade 脚本会要求您输入有关以下管理员帐户的信息：

- Directory Server 管理员 ID 和密码
- Access Manager 管理员 ID 和密码
- Directory Server amldapuser ID 和密码

升级发行版本 4 Portal Server Secure Remote Access (Solaris)

本节讨论影响 Portal Server Secure Remote Access 升级过程的一些注意事项，随后对升级过程本身加以说明。

升级注意事项 (Solaris)

将 Portal Server Secure Remote Access 软件升级到发行版本 5 时，应考虑以下注意事项：

- Portal Server Secure Remote Access 软件包括许多子组件，它们执行许多不同的任务，但必须一起升级到发行版本 5：
 - **Portal-base。**包括管理 Mbean 和随附的管理软件 Logging Framework 以及与监视相关的软件，所有这些软件都打包到 SUNWportal-base 软件包中。
 - **Secure Remote Access 应用程序。**包括 Gateway、Rewriter Proxy 和 Netlet Proxy。这些应用程序通常部署在托管 Portal Server 的计算机之外的一台或多台计算机上。Secure Remote Access 应用程序不需要 Web 容器。
- 如果 Gateway、Rewriter Proxy 和 Netlet Proxy 未部署在同一台计算机上，则应在升级 Gateway 之前升级 Rewriter Proxy 和 Netlet Proxy。
- 所有 Portal Server Secure Remote Access 子组件都对应于同一已安装的 Portal Server Secure Remote Access 映像，如果这些子组件位于要升级的计算机上，则它们同时升级。
- psupgrade 脚本自动检测主机上安装了哪些 Portal Server Secure Remote Access 子组件并升级这些组件。

升级过程 (Solaris)

下述过程适用于要进行升级的计算机中的 Portal Server Secure Remote Access 组件。

1. 以 root 身份登录，或成为超级用户。

```
su -
```

2. 将所有共享组件同步到发行版本 5（如果尚未执行此操作）。

第 59 页中的第 2 章“升级 Java ES 共享组件”中提供了有关说明。

此步骤是第 358 页中的步骤 8 中运行 psupgrade 脚本的必要先决条件。

3. 停止在本地运行的任何 Gateway、Rewriter Proxy 或 Netlet Proxy 实例。

```
PortalServer6-base/bin gateway stop
PortalServer6-base/bin netletd stop
PortalServer6-base/bin rwproxyd stop
```

检查这些进程是否已停止：

```
Gateway: netstat -an | grep 443
Rewriter Proxy: netstat -an | grep 10443
Netlet Proxy: netstat -an | grep 10555
```

4. 确保 Access Manager 正在运行。
5. 设置 psupgrade 脚本需要的两个环境变量（ANT_HOME 和 JAVA_HOME）。例如，

```
export ANT_HOME=/usr/sfw
export JAVA_HOME=/usr/jdk/entsys-j2se
```

6. 确保计算机上有足够的交换空间。

指导原则是交换空间应设置为物理 RAM 大小的两倍。

7. 如果要升级的 Portal Server Secure Remote Access 组件与 Portal Server 不在同一台计算机上，则将 dpadmin 可执行文件从托管 Portal Server 的计算机复制到托管 Portal Server Secure Remote Access 组件的计算机上。

可在以下位置找到 dpadmin 可执行文件：

PortalServer7-base/SUNWps.bak/bin/dpadmin（如果 Portal Server 已升级）。

PortalServer6-base/bin/dpadmin（如果 Portal Server 尚未升级）。

8. 通过 Java ES 发行版本 5 分发软件运行 psupgrade 脚本。

```
cd os_arch/Products/portal_svr/Tools/upgrade/bin
./psupgrade
```

其中 *os_arch* 对应您的平台，例如 *Solaris_sparc*。

psupgrade 脚本调用 Java ES 安装程序安装新软件包并请求以下信息：

- Directory Server 管理员 ID 和密码
- Access Manager 管理员 ID 和密码
- Directory Server amldapuser ID 和密码

9. 启动在[第 358 页中的步骤 3](#)中停止的 Gateway、Rewriter Proxy 或 Netlet Proxy 实例。

```
PortalServer7-base/bin/psadmin start-sra-instance -u amadminUser
-f passwordFile --name default --type gateway
```

```
PortalServer7-base/bin/psadmin start-sra-instance -u amadminUser
-f passwordFile --name default --type nlproxy
```

```
PortalServer7-base/bin/psadmin start-sra-instance -u amadminUser
-f passwordFile --name default --type rwproxy
```

如果上述命令失败，则必须首先注册（启用） Portal Server Secure Remote Access 组件：

```
PortalServer7-base/bin/psadmin provision-sra -u amadminUser
-f passwordFile -p Portal_ID --gateway-profile profileName --enable
```

升级发行版本 4 Portal Server Secure Remote Access (Linux)

本节讨论影响 Portal Server Secure Remote Access 升级过程的一些注意事项，随后对升级过程本身加以说明。

升级注意事项 (Linux)

在 Linux 平台上将 Portal Server Secure Remote Access 软件升级到发行版本 5 所考虑的注意事项与在 Solaris 平台上相同（参见[第 357 页中的“升级注意事项 \(Solaris\)”](#)），但在 Linux OS 上，安装发行版本 5 Portal Server Secure Remote Access 的路径与发行版本 4 相同。因此，在安装发行版本 5 RPM 时，psupgrade 脚本将删除先前的 RPM。

升级过程 (Linux)

下述过程适用于要进行升级的计算机中的 Portal Server Secure Remote Access。

注意	在 Linux 上，无法对从 Java ES 发行版本 4 到发行版本 5 的升级执行回滚操作。确保在执行以下过程之前备份您的系统。
-----------	--

1. 以 root 身份登录，或成为超级用户。

```
su -
```

2. 将所有共享组件同步到发行版本 5（如果尚未执行此操作）。

[第 59 页中的第 2 章“升级 Java ES 共享组件”](#)中提供了有关说明。

此步骤是[第 360 页中的步骤 8](#)中运行 psupgrade 脚本的必要先决条件。

3. 停止在本地运行的任何 Gateway、Rewriter Proxy 或 Netlet Proxy 实例。

```
PortalServer6-base/bin gateway stop
PortalServer6-base/bin netletd stop
PortalServer6-base/bin rwproxyd stop
```

检查这些进程是否已停止：

```
Gateway: netstat -an | grep 443
Rewriter Proxy: netstat -an | grep 10443
Netlet Proxy: netstat -an | grep 10555
```

4. 确保 Access Manager 正在运行。
5. 设置 psupgrade 脚本所需的两个环境变量（ANT_HOME 和 JAVA_HOME）。例如，

```
export ANT_HOME=/opt/sun
export JAVA_HOME=/usr/jdk/entsys-j2se
```

6. 确保计算机上有足够的交换空间。

指导原则是交换空间应设置为物理 RAM 大小的两倍。

7. 如果要升级的 Portal Server Secure Remote Access 组件与 Portal Server 不在同一台计算机上，则将 dpadmin 可执行文件从托管 Portal Server 的计算机复制到托管 Portal Server Secure Remote Access 组件的计算机上。

可在以下位置找到 dpadmin 可执行文件：

PortalServer7-base/SUNWps.bak/bin/dpadmin（如果 Portal Server 已升级）。

PortalServer6-base/bin/dpadmin（如果 Portal Server 尚未升级）。

8. 通过 Java ES 发行版本 5 分发软件运行 psupgrade 脚本。

```
cd os_arch/Products/portal_svr/Tools/upgrade/bin
./psupgrade
```

其中 *os_arch* 对应您的平台，例如 *Solaris_sparc*。

psupgrade 脚本调用 Java ES 安装程序安装新软件包并请求以下信息：

- Directory Server 管理员 ID 和密码
- Access Manager 管理员 ID 和密码
- Directory Server amldapuser ID 和密码

检验升级

如果要升级的 Portal Server Secure Remote Access 组件与 Portal Server 不在同一台计算机上，则可以通过在以下文件中检查版本信息来检验是否安装了发行版本 5 软件包：

```
PortalServer7-base/lib/PSversion.properties
```

但是，如果要升级的 Portal Server Secure Remote Access 组件与 Portal Server 驻留在同一台计算机上，则可以使用以下命令检验升级：

```
PortalServer7-base/bin/psadmin --version --adminuser admin_ID  
-f adminpasswordfile
```

参见第 354 页中的表 16-4 确认输出值。

还可以检查位于以下位置的升级日志文件：

```
/var/sadm/install/logs/Sun_Java_System_Portal_Server_upagrede.log
```

发行版本 4 升级后的任务

除以下情况之外，将 Portal Server Secure Remote Access 升级到发行版本 5 时，不需要执行任何升级后的任务：

- 恢复负载均衡器的配置
- 恢复 Directory Proxy Server 的配置
- 删除发行版本 4 本地化提供者

恢复负载均衡器的配置

如果已通过负载均衡器访问 Portal Server Secure Remote Access 实例，则需要在升级后执行以下步骤恢复负载均衡器配置：

1. 在 *PortalServer7Config-base/platform.conf.default* 文件中设置以下参数：

```
gateway.virtualhost=loadBalancer_hostName loadBalancer_hostIP  
gateway.external.ip=loadBalancer_hostIP  
gateway.dsame.agent=http\://loadBalancer_hostName\  
80/portal/RemoteConfigServlet
```

2. 在 *PortalServer7Config-base/GWConfig-default.properties* 文件中设置以下参数：

```
gateway.ipaddress=Gateway_hostIP
```

3. 如果 Rewriter Proxy 和 Netlet Proxy 实例部署在远离 Portal Server 主机的计算机上，则为它们设置步骤 1 和步骤 2 中的对应参数。
4. 重新启动 Portal Server 和负载均衡的 Portal Server Secure Remote Access 实例。

恢复 Directory Proxy Server 的配置

如果 Portal Server Secure Remote Access 实例已通过 Directory Proxy Server 实例访问了 Directory Server，则必须在升级前将 Directory Proxy Server 的主机和端口号设置恢复为其原始值。参见第 356 页中的“删除 Directory Proxy Server 的配置”，在此过程中已修改了这些属性的值以备升级之用。

删除发行版本 4 本地化提供者

只有删除发行版本 4 本地化提供者，才会加载本地化 Proxylet 服务，步骤如下：

1. 转至 `PortalServer7Data-base/portals/Upgraded/desktop` 目录。
2. 删除 `default_Locale` 中的所有目录和文件，以下目录和文件除外：
 - 您创建的文件和目录（未随 Portal Server Secure Remote Access 一同提供）
 - `message.properties` 文件
 - 以下目录：

```
AddressBookProvider
BookmarkProvider
CalendarProvider
LoginProvider
LotusNotesAddressBookProvider
LotusNotesCalendarProvider
LotusNotesMailProvider
MSEExchangeAddressBookProvider
MSEExchangeCalendarProvider
MSEExchangeMailProvidervMailProvider
NotesProvider
PersonalNoteProvider
Register
SampleRSS
SampleURLScrapper
SampleXML
TemplateEditContainerProvider
TemplateTabContainerProvider
URLScrapperProvider
UWCAddressBookProvider
UserInfo
UserInfoProvider
XMLProvider
error
```

3. 重新启动在其中部署了 Portal Server 的 Web 容器。

升级回滚 (Solaris)

本节介绍影响 Portal Server Secure Remote Access 升级回滚过程的一些注意事项，随后对升级回滚过程本身加以说明。

回滚注意事项 (Solaris)

发行版本 5 的升级回滚过程包括在 *PortalServer6-base* 处恢复为发行版本 4 的安装。

回滚过程 (Solaris)

1. 以 root 身份登录，或成为超级用户。

```
su -
```

2. 将 Directory Server 恢复为升级前的状态。

使用 Directory Server 备份 / 恢复命令行和 GUI 实用程序。参见《Sun Java System Directory Server Enterprise Edition 6.0 管理指南》(<http://docs.sun.com/doc/820-0293>) 中的“目录服务器备份和恢复”一章。

3. 删除发行版本 5 Portal Server Secure Remote Access 软件包。

- a. 启动 Java ES 卸载程序。

```
/var/sadm/prod/SUNWentsys5/uninstall
```

- b. 选择所有已安装的 Portal Server Secure Remote Access 组件。

- c. 确认您的卸载选择。

- d. 退出 Java ES 卸载程序。

4. 将 *PortalServer6-base* 和 *PortalServer6Config-base* 目录恢复至其原始位置。

在升级期间，它们被移到了带有 .bak 扩展名的目录中。

升级回滚 (Linux)

由于到发行版本 5 的升级需要删除发行版本 4 的二进制文件，因此在 Linux 上执行升级回滚是非常困难的。

一种回滚方法是在尝试进行升级之前，先创建一个并行系统，**然后**升级并测试该并行系统。如果需要对升级执行回滚操作，可恢复为该并行系统。

多实例升级

在一些部署体系结构中，Portal Server Secure Remote Access 组件（例如 Gateway）被部署在多个计算机系统上，以提供安全性和可伸缩性以及提高可用性。例如，您可以在多台计算机上运行 Gateway 组件，并使用一个负载均衡器来分配负载。

就 Gateway 负载均衡的实例而言，可执行滚动升级，期间可依序升级各个 Gateway 实例而不中断服务，如下所述。该过程需要考虑以下约束：单独的 Portal Server Secure Remote Access 组件与早期版本不向后兼容；它们都需要与 Portal Server 一起在 Java ES 发行版本 5 进行同步。不过，滚动升级期间，在升级 Portal Server 实例时，发行版本 4 Portal Server Secure Remote Access 实例可一直保持运行状态。

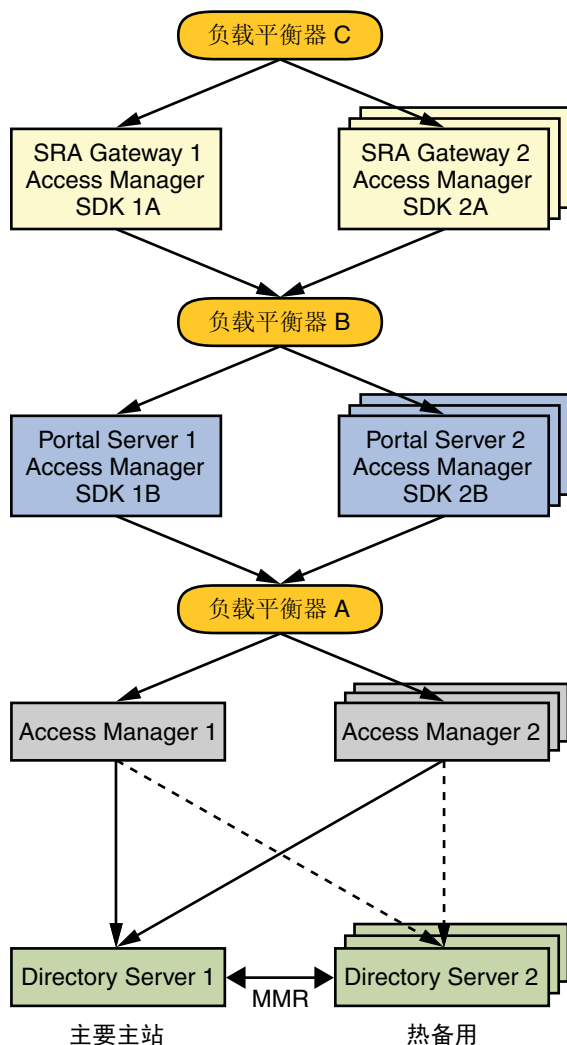
第 365 页中的图 16-1 所示的部署体系结构将用于说明滚动升级过程。

在此体系结构中，通过 Portal Server Secure Remote Access Gateway 实例访问多个 Portal Server 实例。Portal Server 实例和 Gateway 实例均达到负载平衡，以提供可用性和可伸缩性。

而 Portal Server 实例又将通过负载均衡器访问 Access Manager 实例。Access Manager 和 Access Manager SDK 实例将访问一个为多主复制 (multi-master replication, MMR) 设置的目录。尽管可以使用其他 Directory Server 复制模式，但 MMR 是具有高可用性和高可伸缩性的目录服务。

在图 16-1 中，对 Gateway、Portal Server、Access Manager 和 Directory Server 的多个实例进行了编组以便于解释升级过程。例如，Portal Server 2 代表第二个到第 n 个 Portal Server 实例。

图 16-1 多 Portal Server 实例的部署体系结构示例



从发行版本 4 Gateway（和 Portal Server）到发行版本 5 的滚动升级过程如下：

1. 如果您要将发行版本 4 Access Manager 升级到发行版本 5，请如第 257 页中的“多实例升级”中所述执行滚动升级。请注意，在将发行版本 4 Gateway 或发行版本 4 Portal Server 升级到发行版本 5 时，不必将发行版本 4 Access Manager 升级到发行版本 5。
2. 如下所示修改 Portal Server 和 Gateway 实例的配置。

- a. 将 Portal Server 2 配置为指向 Directory Server 2 而不是 Directory Server 1。

为简洁起见，在此步骤和后续步骤中，“Portal Server 2”表示 Portal Server 2 到 Portal Server n。

- b. 将 Gateway 2 配置为指向 Directory Server 2 而不是 Directory Server 1。

为简洁起见，在此步骤和后续步骤中，“Gateway 2”表示 Gateway 2 到 Gateway n。

3. 升级 Portal Server 1。

- a. 在负载均衡器 B 中禁用 Portal Server 1。

请求不会再路由至 Portal Server 1。

- b. 禁用 Directory Server MMR。

Directory Server 2 将不再与 Directory Server 1 同步。

- c. 将 Access Manager SDK 1B 升级到发行版本 5。

按照第 259 页中的“[仅升级发行版本 4 Access Manager SDK](#)”中的过程操作。

- d. 将 Portal Server 1 升级到发行版本 5。

按照第 353 页中的“[发行版本 4 Portal Server Secure Remote Access 的升级](#)”中所述执行 Portal Server 实例的升级，但要注意以下事项：

- 特别注意以下升级前任务：[第 355 页中的“删除负载均衡器的配置”](#)。
- 在执行升级之前，确认
AccessManagerConfig-base/config/AMConfig.properties 文件中
am.encrypted.pwd 的值对于本地 Access Manager SDK 和与其关联的
远程 Access Manager 实例都是相同的。
- 对于 psupgrade 为要升级的每个 Portal Server 实例所请求的 Portal Instance ID 参数，确保为其提供一个唯一的非空值。

Directory Server 1 的 Portal Server 数据已更新为发行版本 5。

4. 升级 Gateway 1。
 - a. 在负载均衡器 C 中禁用 Gateway 1。
请求不会再路由至 Gateway 1。
 - b. 将 Access Manager SDK 1A 升级到发行版本 5。
按照第 259 页中的“仅升级发行版本 4 Access Manager SDK”中的过程操作。
 - c. 将 Gateway 1 升级到发行版本 5。
按照第 353 页中的“发行版本 4 Portal Server Secure Remote Access 的升级”所述执行 Gateway 的升级，但要注意以下事项：
 - 特别注意以下升级前任务：第 355 页中的“删除负载均衡器的配置”。
 - 在执行升级之前，确认
AccessManagerConfig-base/config/AMConfig.properties 文件中
am.encrypted.pwd 的值对于本地 Access Manager SDK 和与其关联的
远程 Access Manager 实例都是相同的。
5. 在各自的负载均衡器中，启用以前禁用的 Portal Server 1 和 Gateway 1，步骤如下：
 - a. 在负载均衡器 B 中启用 Portal Server 1。
请求将重新路由至 Portal Server 1。
 - b. 在负载均衡器 C 中启用 Gateway 1。
请求将重新路由至 Gateway 1。
6. 在各自的负载均衡器中，禁用 Portal Server 2 和 Gateway 2，步骤如下：
 - a. 在负载均衡器 B 中禁用 Portal Server 2。
请求不会再路由至 Portal Server 2。
 - b. 在负载均衡器 C 中禁用 Gateway 2。
请求不会再路由至 Gateway 2。

7. 升级 Portal Server 2。

a. 将 Portal Server 2 的配置恢复为指向 Directory Server 1。

b. 将 Access Manager SDK 2B 升级到发行版本 5。

按照第 259 页中的“仅升级发行版本 4 Access Manager SDK”中的过程操作。

c. 将 Portal Server 2 升级到发行版本 5。

按照升级 Portal Server 1（第 366 页中的步骤 d）的过程操作。

d. 在负载均衡器 B 中启用 Portal Server 2。

请求将重新路由至 Portal Server 2。

8. 升级 Gateway 2。

a. 将 Gateway 2 的配置恢复为指向 Directory Server 1。

b. 将 Access Manager SDK 2A 升级到发行版本 5。

按照第 259 页中的“仅升级发行版本 4 Access Manager SDK”中的过程操作。

c. 将 Gateway 2 升级到发行版本 5。

按照升级 Gateway 1（第 367 页中的步骤 c）的过程操作。

d. 在负载均衡器 C 中启用 Gateway 2。

请求将重新路由至 Gateway 2。

9. 启用 Directory Server MMR。

现在，Directory Server 2 的 Portal Server 数据与 Directory Server 1 同步。

注

在滚动升级方案中（与上述过程中的情况不同，Portal Server 实例将升级到发行版本 5，而较早发行版本的 Gateway 组件保持活动状态；并且将通过负载均衡器访问 Gateway 实例），您应检查所有 Gateway 实例，以确认

PortalServer6Config-base/GWConfig.properties 文件和 *GWConfig-default.properties* 文件中的以下配置属性引用的是 Gateway 而非负载均衡器：

```
GW_IP=Gateway_hostIP
GW_HOST=Gateway_hostName
```

如果这些属性指向负载均衡器，则 Gateway 将不再访问已升级的 Portal Server 实例。

从 Java ES 发行版本 3 升级 Portal Server Secure Remote Access

将 Java ES 2005Q1（发行版本 3）Portal Server Secure Remote Access 升级到发行版本 5 的过程与将发行版本 4 Portal Server Secure Remote Access 升级到发行版本 5 的过程相同，但存在以下例外：

- 升级 Portal Server Secure Remote Access 依赖项
- 升级发行版本 3 Portal Server Secure Remote Access
- 多实例升级

升级 Portal Server Secure Remote Access 依赖项

从发行版本 3 升级 Portal Server Secure Remote Access 时，必须在升级 Portal Server Secure Remote Access 之前将 Access Manager 升级到发行版本 4 或发行版本 5，而且不能将任何其他依赖项保留在发行版本 3，也不能将一些依赖项升级到发行版本 4，而将另一些依赖项升级到发行版本 5。有关更多信息，参见第 350 页中的“选择性升级问题”。

以下依赖项需要按照下面所示的顺序进行升级。

1. **共享组件。** 第 59 页中的第 2 章“升级 Java ES 共享组件”提供了将 Java ES 共享组件升级到发行版本 5 的说明。
2. **Directory Server。** 第 103 页中的“从 Java ES 发行版本 2 升级 Directory Server”提供了将 Directory Server 升级到发行版本 5 的说明。
3. **Access Manager (Access Manager SDK)。** 第 235 页中的第 14 章“Access Manager”提供了将 Access Manager 升级到发行版本 5 的说明。
4. **Portal Server。** 第 279 页中的第 15 章“Portal Server”提供了升级 Portal Server 的说明。

升级发行版本 3 Portal Server Secure Remote Access

要将发行版本 3 Portal Server Secure Remote Access 升级到发行版本 5，请按照第 352 页中的“从 Java ES 发行版本 4 升级 Portal Server Secure Remote Access”中的说明操作，但需要在提到发行版本 4 的地方用发行版本 3 代替。

多实例升级

在一些部署体系结构中，Portal Server Secure Remote Access 组件（例如 Gateway）被部署在多个计算机系统上，以提供安全性和可伸缩性以及提高可用性。例如，您可以在多台计算机上运行 Gateway 组件，并使用一个负载均衡器来分配负载。

要从发行版本 3 Portal Server Secure Remote Access 执行多实例升级，请按照第 364 页中的“多实例升级”中所述的过程操作，但需要在提到“发行版本 4”的地方用“发行版本 3”代替。此外，还必须升级 Access Manager，如第 365 页中的步骤 1 所述。

从 Java ES 发行版本 2 升级 Portal Server Secure Remote Access

本节包括有关将 Java ES 2004Q2（发行版本 2）Portal Server Secure Remote Access 升级到发行版本 5 的信息。该升级过程与将发行版本 4 Portal Server Secure Remote Access 升级到发行版本 5 的过程类似，但其中有一些更改，如以下几节所述：

- [发行版本 2 升级前的任务](#)
- [升级发行版本 2 Portal Server Secure Remote Access](#)
- [发行版本 2 升级后的任务](#)
- [多实例升级](#)

注	如果要在 Linux 平台上从发行版本 2 Portal Server Secure Remote Access 升级，则必须执行双重升级，即 Portal Server Secure Remote Access 和操作系统都要升级（RHEL 2.1 上不支持发行版本 5 Portal Server Secure Remote Access）。有关更多信息，参见第 351 页中的“双重升级”。
----------	--

发行版本 2 升级前的任务

除了以下任务外，从发行版本 2 升级 Portal Server Secure Remote Access 的升级前任务与第 354 页中的“[发行版本 4 升级前的任务](#)”中所述的那些任务相同：

- [升级 Portal Server Secure Remote Access 依赖项](#)
- [删除网关服务条目](#)

升级 Portal Server Secure Remote Access 依赖项

从发行版本 2 升级 Portal Server Secure Remote Access 时，必须在升级 Portal Server Secure Remote Access 之前将 Access Manager 升级到发行版本 4 或发行版本 5，而且不能将任何其他依赖项保留在发行版本 2，也不能将一些依赖项升级到发行版本 4，而将另一些依赖项升级到发行版本 5。有关更多信息，参见第 350 页中的“[选择性升级问题](#)”。

以下依赖项需要按照下面所示的顺序进行升级。

1. **共享组件。** 第 59 页中的第 2 章“升级 Java ES 共享组件”提供了将 Java ES 共享组件升级到发行版本 5 的说明。但是，如果共享组件尚未升级，则它们将由 `psupgrade` 脚本自动升级。
2. **Directory Server。** 第 103 页中的“从 Java ES 发行版本 2 升级 Directory Server”提供了将 Directory Server 升级到发行版本 5 的说明。
3. **Access Manager (Access Manager SDK)。** 第 235 页中的第 14 章“Access Manager”提供了将 Access Manager 升级到发行版本 5 的说明。
4. **Portal Server。** 第 279 页中的第 15 章“Portal Server”提供了升级 Portal Server 的说明。

要将发行版本 2 Portal Server Secure Remote Access 升级到发行版本 5，请按照第 352 页中的“从 Java ES 发行版本 4 升级 Portal Server Secure Remote Access”中的说明操作，但需要在提到发行版本 4 的地方用发行版本 2 代替。

删除网关服务条目

从发行版本 2 升级 Portal Server 时，必须手动删除 `amService-srapGateway` 用户条目，否则 Gateway 将无法在升级后启动。执行以下步骤：

1. 登录到 Access Manager 控制台。
2. 列出组织 DN 中的所有用户。
3. 删除 `amService-srapGateway` 用户。

升级发行版本 2 Portal Server Secure Remote Access

将 Java ES 2004Q2（发行版本 2）Portal Server Secure Remote Access 升级到发行版本 5 的过程与将发行版本 4 Portal Server Secure Remote Access 升级到发行版本 5 的过程相同。

要将发行版本 2 Portal Server Secure Remote Access 升级到发行版本 5，请按照第 352 页中的“从 Java ES 发行版本 4 升级 Portal Server Secure Remote Access”中的说明操作，但需要在提到发行版本 4 的地方用发行版本 2 代替。

发行版本 2 升级后的任务

除了以下任务外，从发行版本 2 升级的升级后任务与第 361 页中的“发行版本 4 升级后的任务”中所述的那些任务相同：

- 为 Proxylet 服务设置 Portal Server 域

为 Proxylet 服务设置 Portal Server 域

将发行版本 2 Portal Server Secure Remote Access 升级到发行版本 5 后，必须设置正确的 Portal Server 域值。

1. 登录到 Portal Server 控制台，并导航到 "Secure Remote Access" 下的 "Proxylet" 选项卡。
2. 选择可在其中找到 Proxylet 服务的组织的标识名 (Distinguished Name, DN)。
3. 在 Proxylet 规则的“域”字段下，将 SERVER_DOMAIN 替换为在其中安装了 Portal Server 的域的名称。
4. 对于可在其中找到 Proxylet 服务的所有组织，重复上述步骤。

多实例升级

在将发行版本 2 Portal Server Secure Remote Access 组件（或 Portal Server）升级到发行版本 5 的过程中，不支持多实例滚动升级（参见第 364 页中的“多实例升级”）。

从过渡功能版本 7.0 升级 Portal Server Secure Remote Access

本节包含关于将 Portal Server Secure Remote Access 从过渡功能版本 (Interim Feature Release, IFR) 7.0 2005Q4 升级到 Java ES 5（发行版本 5）的信息。

本节涵盖了以下主题：

- [简介](#)
- [Portal Server Secure Remote Access IFR 7.0 升级](#)
- [多实例升级](#)

简介

在将 Portal Server Secure Remote Access IFR 7.0 升级到发行版本 5 时，请考虑升级过程中的以下方面：

与从发行版本 4 进行升级时的情况一样，用于将 Portal Server Secure Remote Access IFR 升级到发行版本 5 的 psupgrade 脚本并不安装新软件包。升级过程将要求您应用以下修补程序：

表 16-5 用于将 Portal Server Secure Remote Access IFR 升级到发行版本 5 的修补程序¹

说明	修补程序 ID: Solaris 9 和 10	修补程序 ID: Linux
Portal Server 7.1	121465-10 (SPARC)	121467-10
	121466-10 (x86)	
Portal Server 7.1 本地化版本	123254-02 (SPARC)	123255-02
	124590-02 (x86)	

1. 修补程序修订版本号是升级到 Java ES 发行版本 5 所需的最低版本号。如果有较新的修订版本可用，请用较新版本替代该表中所示的版本。

Portal Server Secure Remote Access IFR 7.0 升级

本节介绍如何在 Solaris 和 Linux 平台上将 Portal Server Secure Remote Access 从 IFR 升级到 Java ES 发行版本 5。如果其中某个主题取决于具体平台的过程，则该主题将指明所适用的操作系统。本节包括以下主题：

- [升级前的任务](#)
- [升级 Portal Server Secure Remote Access IFR 7.0 \(Solaris\)](#)
- [升级 Portal Server Secure Remote Access IFR 7.0 \(Linux\)](#)
- [检验升级](#)
- [升级后的任务](#)
- [升级回滚 \(Solaris\)](#)
- [升级回滚 \(Linux\)](#)

升级前的任务

IFR 升级的升级前任务与发行版本 4 升级的升级前任务相同（参见第 354 页中的“[发行版本 4 升级前的任务](#)”）。

升级 Portal Server Secure Remote Access IFR 7.0 (Solaris)

本节讨论影响 Portal Server Secure Remote Access 升级过程的一些注意事项，随后对升级过程本身加以说明。

IFR 7 升级注意事项 (Solaris)

将 Portal Server Secure Remote Access IFR 升级到发行版本 5 时，需要考虑的注意事项与发行版本 4 升级相同（参见第 357 页中的“[升级注意事项 \(Solaris\)](#)”）。

IFR 7 升级过程 (Solaris)

下述过程适用于要进行升级的计算机中的 Portal Server Secure Remote Access。

1. 以 root 身份登录，或成为超级用户。

```
su -
```

2. 停止在本地运行的任何 Gateway、Rewriter Proxy 或 Netlet Proxy 实例。

```
PortalServer7-base/bin/psadmin stop-sra-instance -u amadminUser  
-f passwordFile -t gateway -N gatewayProfileName
```

```
PortalServer7-base/bin/psadmin stop-sra-instance -u amadminUser  
-f passwordFile -t rwproxy -N gatewayProfileName
```

```
PortalServer7-base/bin/psadmin stop-sra-instance -u amadminUser  
-f passwordFile -t nlproxy -N gatewayProfileName
```

检查这些进程是否已停止：

```
Gateway: netstat -an | grep 443  
Rewriter Proxy: netstat -an | grep 10443  
Netlet Proxy: netstat -an | grep 10555
```

3. 确保 Access Manager 正在运行。
4. 根据第 374 页中的表 16-5，获取所需的修补程序。

若未指明使用某个特定修订版本，始终应使用可用的最新修补程序修订版本。

可从以下网址将修补程序下载到 /tmp：

<http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access>

5. 应用相应的 Portal Server 修补程序，如果需要，应用表 16-5 中的本地化修补程序。

```
patchadd patch_ID
```

6. 确认修补程序升级成功：

```
showrev -p | grep patch_ID
```

输出应返回在步骤 5 中应用的修补程序 ID 的版本。

7. 在步骤 5 中，如果本地化软件包已升级，将 Portal Server 控制台 JVM 的语言环境设置为 UTF-8。

```
export LC_ALL=ja_JP.UTF-8  
export LANG=ja_JP.UTF-8
```

8. 设置 psupgrade 脚本所需的两个环境变量（ANT_HOME 和 JAVA_HOME）：

```
export ANT_HOME=/usr/sfw  
export JAVA_HOME=/usr/jdk/entsys-j2se
```

9. 确保计算机上有足够的交换空间。

指导原则是交换空间应设置为物理 RAM 大小的两倍。

10. 运行 psupgrade 脚本。

```
cd PortalServer7-base/bin
./psupgrade
```

psupgrade 脚本并不通过 Java ES 发行版本 5 分发软件运行，而且，并不调用 Java ES 安装程序（软件包已进行修补）。

升级 Portal Server Secure Remote Access IFR 7.0 (Linux)

本节讨论影响 Portal Server Secure Remote Access 升级过程的一些注意事项，随后对升级过程本身加以说明。

IFR 7 升级注意事项 (Linux)

在 Linux 平台上将 Portal Server Secure Remote Access 软件升级到发行版本 5 需要考虑的注意事项与在 Solaris 平台上相同（参见第 357 页中的“升级注意事项 (Solaris)”），但在 Linux OS 上安装发行版本 5 修补程序时会删除先前的 RPM。

IFR 7 升级过程 (Linux)

下述过程适用于要进行升级的计算机中的 Portal Server Secure Remote Access。

注意	在 Linux 上，无法对从 Portal Server Secure Remote Access IFR 到发行版本 5 的升级执行回滚操作。确保在执行以下过程之前备份您的系统。
-----------	--

1. 以 root 身份登录，或成为超级用户。

```
su -
```

2. 停止在本地运行的任何 Gateway、Rewriter Proxy 或 Netlet Proxy 实例。

```
PortalServer7-base/bin/psadmin stop-sra-instance -u amadminUser
-f passwordFile -t gateway -N gatewayProfileName
```

```
PortalServer7-base/bin/psadmin stop-sra-instance -u amadminUser
-f passwordFile -t rwproxy -N gatewayProfileName
```

```
PortalServer7-base/bin/psadmin stop-sra-instance -u amadminUser
-f passwordFile -t nlproxy -N gatewayProfileName
```

检查这些进程是否已停止：

```
Gateway: netstat -an | grep 443
Rewriter Proxy: netstat -an | grep 10443
Netlet Proxy: netstat -an | grep 10555
```

3. 确保 Access Manager 正在运行。
4. 使用第 374 页中的表 16-5 中的修补程序编号和 RPM 名称获取所需的修补程序。
若未指明使用某个特定修订版本，始终应使用可用的最新修补程序修订版本。

可从以下网址将修补程序下载到 /tmp:

<http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access>

5. 应用 Portal Server 修补程序，如果需要，以表 16-5 中的顺序应用该表中 Portal Server 的本地化 RPM。

参见 Portal Server 修补程序的自述文件，该文件介绍了如何使用脚本应用修补程序的 RPM:

```
cd /tmp
```

其中，/tmp 是修补程序的下载目录。

```
./update
```

更新脚本将安装 RPM。

对于本地化修补程序，使用以下命令安装每个 RPM:

```
rpm -Fvh patchName-version.rpm
```

6. 确认修补程序升级成功:

```
rpm -qa | grep sun-portal-core
```

应返回 RPM 的升级修订版本号。

7. 在步骤 5 中，如果本地化软件包已升级，将 Portal Server 控制台 JVM 的语言环境设置为 UTF-8。

```
export LC_ALL=ja_JP.UTF-8
```

```
export LANG=ja_JP.UTF-8
```

8. 设置 psupgrade 脚本所需的两个环境变量 (ANT_HOME 和 JAVA_HOME):

```
export ANT_HOME=/opt/sun
```

```
export JAVA_HOME=/usr/jdk/entsys-j2se
```

9. 确保计算机上有足够的交换空间。

指导原则是交换空间应设置为物理 RAM 大小的两倍。

10. 运行 psupgrade 脚本。

```
cd PortalServer7-base/bin
./psupgrade
```

psupgrade 脚本并不通过 Java ES 发行版本 5 分发软件运行，而且，并不调用 Java ES 安装程序（软件包已进行修补）。

检验升级

可使用以下命令检验是否将 Portal Server Secure Remote Access 软件包修补到发行版本 5:

```
PortalServer7-base/bin/psadmin --version --adminuser admin_ID
-f adminpasswordfile
```

参见第 354 页中的表 16-4 确认输出值。

还可以检查位于以下位置的升级日志文件:

```
/var/sadm/install/logs/Sun_Java_System_Portal_Server_upagrede.log
```

升级后的任务

将 Portal Server Secure Remote Access 升级到发行版本 5 时，不需要执行任何升级后的任务。

升级回滚 (Solaris)

本节介绍影响 Portal Server Secure Remote Access 升级回滚过程的一些注意事项，随后对升级回滚过程本身加以说明。

回滚注意事项 (Solaris)

发行版本 5 的升级回滚过程包括在 *PortalServer7-base* 处恢复为 IFR 的安装。

回滚过程 (Solaris)**1. 以 root 身份登录，或成为超级用户。**

```
su -
```

2. 将 Directory Server 恢复为升级前的状态。

使用 Directory Server 备份 / 恢复命令行和 GUI 实用程序。参见《Sun Java System Directory Server Enterprise Edition 6.0 管理指南》(<http://docs.sun.com/doc/820-0293>) 中的“目录服务器备份和恢复”一章。

3. 回退第 374 页中的表 16-5 中的 Portal Server 7.1 修补程序。

```
patchrm patch_ID
```

升级回滚 (Linux)

在 Linux 平台上，不存在升级回滚过程。

多实例升级

在一些部署体系结构中，Portal Server Secure Remote Access 被部署在多个计算机系统上，以提供可伸缩性和提高可用性。例如，您可以在多台计算机上运行 Portal Server Secure Remote Access 组件，并使用一个负载均衡器来分配负载。

就 Portal Server Secure Remote Access 负载均衡的实例而言，可执行滚动升级，期间您可依序升级各个 Portal Server Secure Remote Access 实例而不中断服务。在升级每个 Portal Server Secure Remote Access 实例时，其他实例仍可保持运行。可按第 375 页中的“Portal Server Secure Remote Access IFR 7.0 升级”中所述升级每个实例。

要从 IFR Portal Server Secure Remote Access 执行多实例升级，请按照第 364 页中的“多实例升级”中所述的过程操作，但需要在提到发行版本 4 的地方用 "IFR" 代替。此外，还必须升级 Access Manager，如第 365 页中的步骤 1 所述。

Java Enterprise System 发行版本内容

本附录列出了各个 Java Enterprise System 发行版本的内容。包括以下各节：

- 第 382 页中的“Java ES 2003Q4（发行版本 1）”
- 第 384 页中的“Java ES 2004Q2（发行版本 2）”
- 第 387 页中的“Java ES 2005Q1（发行版本 3）”
- 第 392 页中的“Java ES 2005Q4（发行版本 4）”
- 第 397 页中的“Java ES 5（发行版本 5）”

Java ES 2003Q4（发行版本 1）

本节列出了 Java Enterprise System 2003Q4 的内容。

发行版本 1 安装程序可选组件

Sun Open Network Environment (Sun ONE) 和 Sun Cluster 组件产品提供了支持分布式企业应用程序所需的基础结构服务。这些组件产品包括：

- Sun Cluster 3.1 和 Sun Cluster Agents for Sun ONE
- Sun ONE Administration Server 5.2
- Sun ONE Application Server 7 Update 1
- Sun ONE Calendar Server 6.0
- Sun ONE Directory Server 5.2
- Sun ONE Directory Proxy Server 5.2
- Sun ONE Identity Server 6.1
- Sun ONE Instant Messaging 6.1
- Sun ONE Message Queue 3.0.1 Service Pack 2
- Sun ONE Messaging Server 6.0
- Sun ONE Portal Server 6.2
- Sun ONE Portal Server, Secure Remote Access 6.2
- Sun ONE Web Server 6.1

发行版本 1 共享组件

共享组件提供组件产品所依赖的本地服务和技术支持。安装组件产品时，Java ES 安装程序会自动安装所需的共享组件（如果尚未安装）。

Java Enterprise System 包括以下共享组件：

- ANT (Jakarta ANT Java/XML-based build tool)
- Apache Commons Logging
- ICU (International Components for Unicode)
- J2SE™ platform 1.4.1_06 (Java 2 Platform, Standard Edition)
- JAF (JavaBeans™ Activation Framework)
- JATO (Sun ONE Application Framework)
- JavaHelp™ Runtime
- JAXM (Java API for XML Messaging) Client Runtime
- JAXP (Java API for XML Processing)
- JAXR (Java API for XML Registries)
- JAX-RPC (Java APIs for XML-based Remote Procedure Call)
- JSS (Java Security Services)
- KT search engine
- LDAP C Language SDK
- NSPR (Netscape Portable Runtime)
- NSS (Network Security Services)
- SAAJ (SOAP with Attachments API for Java)
- SASL (Simple Authentication and Security Layer)
- XML C Library (libxml)

注 Application Server 和 Directory Server 还要求在系统中安装 Perl，但 Perl 不会作为 Java ES 共享组件自动安装。

Java ES 2004Q2（发行版本 2）

本节列出了 Java Enterprise System 2004Q2 的内容。

发行版本 2 安装程序可选组件

组件产品提供支持分布式企业应用程序所需的基础结构服务。在特定主机上安装 Java Enterprise System 时，可根据总体部署体系结构来选择要将哪些组件产品安装到该主机。

Java Enterprise System 2004Q2 包括以下组件产品：

通信和协作服务

- Sun Java System Messaging Server 6 2004Q2
- Sun Java System Calendar Server 6 2004Q2
- Sun Java System Instant Messaging 6 2004Q2
- Sun Java System Portal Server 2004Q2
- Sun Java System Portal Server Mobile Access 2004Q2
- Sun Java System Portal Server Secure Remote Access 2004Q2
- Sun Java System Communications Express 6 2004Q2

Web 和应用程序服务

- Sun Java System Application Server 7.0 Update 3 (Standard and Platform Editions)
- Sun Java System Web Server 6 2004Q2 Update 1 Service Pack 2
- Sun Java System Message Queue 3.5 SP1 (Platform and Enterprise Editions)

目录和身份认证服务

- Sun Java System Identity Server 6.2 2004Q2, 包括
Sun Java System Communications Services 6 2004Q2 User Management Utility
- Sun Java System Directory Server 5 2004Q2
- Sun Java System Directory Proxy Server 5 2004Q2

可用性服务

- Sun Cluster 3.1 4/04 和 Sun Cluster Agents for Sun Java System

管理服务

- Sun Java System Administration Server 5 2004Q2
- Sun Remote Services Net Connect 3.5

注意, Sun Cluster、Sun Cluster Agents 和 Sun Remote Services Net Connect 在 Linux OS 上不可用。

发行版本 2 共享组件

共享组件提供组件产品所依赖的本地服务和技术支持。安装组件产品时, Java ES 安装程序会自动安装所需的共享组件(如果尚未安装)。

Java Enterprise System 2004Q2 包括以下共享组件:

- Ant (Jakarta ANT Java/XML-based build tool)
- Apache Commons Logging
- Apache SOAP (Simple Object Access Protocol)
- ICU (International Components for Unicode)
- J2SE™ platform 1.4.2_04 (Java 2 Platform, Standard Edition)
- JAF (JavaBeans™ Activation Framework)
- JATO (Java Application Framework)
- JavaHelp™ Runtime
- JAXB (Java Architecture for XML Binding)
- JAXM (Java API for XML Messaging) Client Runtime
- JAXP (Java API for XML Processing)

- JAXR (Java API for XML Registries)
- JAX-RPC (Java APIs for XML-based Remote Procedure Call)
- JCAPI (Java Calendar API)
- JSS (Java Security Services)
- KT search engine
- LDAP C Language SDK
- LDAP Java SDK
- NSPR (Netscape Portable Runtime)
- NSS (Network Security Services)
- Perl LDAP, 包括 NSPERL
- SAAJ (SOAP with Attachments API for Java)
- SAML (Security Assertions Markup Language)
- SASL (Simple Authentication and Security Layer)
- SNMP (Simple Network Management Protocol) Peer
- Sun Explorer Data Collector
- XML C Library (libxml)

Java ES 2005Q1（发行版本 3）

本节列出了 Java Enterprise System 2005Q1 的内容。

发行版本 3 安装程序可选组件

在 Java ES 安装程序的组件选择页面中，可选组件按其帮助提供的服务进行了分组。下面的列表还显示了随每个组件一起安装的子组件。

通信和协作服务

- Sun Java System Messaging Server 5 2006Q1
- Sun Java System Calendar Server 5 2006Q1
- Sun Java System Instant Messaging 5 2007Q1
 - Instant Messaging Server Core ; 包括服务器和多路复用器软件
 - Instant Messaging Resources
 - Access Manager Instant Messaging Service
- Sun Java System Portal Server 6 2005Q1
- Sun Java System Portal Server Secure Remote Access 5 2006Q1
 - Secure Remote Access Core
 - Gateway
 - Netlet Proxy
 - Rewriter Proxy
- Sun Java System Communications Express 2005Q1
- Sun Java System Directory Preparation Tool

Web 和应用程序服务

- Sun Java System Application Server Enterprise Edition 8.1 2005Q1

- Domain Administration Server
- Application Server Node Agent
- Command Line Administration Tool
- Load Balancing Plugin

可以与 Web Server 或 Apache Web Server 一起使用，在配置时可以选择。
默认情况下与 Web Server 一起使用。

- PointBase
- 样例应用程序
- Sun Java System Web Server 6 2005Q1 Update 1 Service Pack 4
- Sun Java System Message Queue 3 2005Q1

目录和身份认证服务

- Sun Java System Access Manager 5 2006.3Q1

Portal Server 和 Messaging Server 的 Delegated Administrator 置备工具是随 Access Manager 一起自动安装的。

- Identity Management and Policy Services Core (包括 Delegated Administrator Utility)
- Access Manager Administration Console
- Common Domain Services for Federation Management
- Access Manager SDK
- Sun Java System Directory Server 5 2005Q1
- Sun Java System Directory Proxy Server 5 2005Q1

可用性服务

- Sun Cluster 3.1 9/04
 - Sun Cluster Core
- Sun Cluster Agents for Sun Java System
 - HA/Scalable Sun Java System Web Server
 - HA Sun Java System Message Queue
 - HA Sun Java System Portal Server
 - HA Sun Java System Administration Server
 - HA Sun Java System Directory Server
 - HA Sun Java System Messaging Server
- HADB (用于高可用性会话存储)

管理服务

- Sun Java System Administration Server 5 2005Q1
- SunSM Remote Services Net Connect 3.1.1

注

Sun Cluster、Sun Cluster Agents 和 Sun Remote Services Net Connect 在 Solaris 10 或 Linux 操作系统上不可用。

Sun Remote Services Net Connect 在 Solaris x86 平台上不可用。

发行版本 3 共享组件

共享组件为可选组件提供本地服务和技术支持。安装 Java ES 组件时，安装程序会自动安装所需的共享组件（如果尚未安装）。

此版本的 Java ES 包括以下共享组件：

- Ant (Jakarta ANT Java/XML-based build tool)
- Apache SOAP (Simple Object Access Protocol) Runtime
- Berkeley Database
- Common agent container
- ICU (International Components for Unicode)
- J2SE™ (Java 2 Platform, Standard Edition) platform 5.0
- JAF (JavaBeans™ Activation Framework)
- JATO (Java Studio Web Application Framework)
- JavaHelp™ Runtime
- JavaMail™ Runtime
- JAXB (Java Architecture for XML Binding) Runtime
- JAXP (Java API for XML Processing)
- JAXR (Java API for XML Registries) Runtime
- JAX-RPC (Java API for XML-based Remote Procedure Call) Runtime
- JCAPI (Java Calendar API)
- JDMK (Java Dynamic Management™ Kit) Runtime
- JSS (Java Security Services)
- KTSE (KT Search Engine)
- LDAP C SDK
- LDAP Java SDK
- NSPR (Netscape Portable Runtime)
- NSS (Network Security Services)
- Perl LDAP，包括 NSPERL

- SAAJ (SOAP with Attachments API for Java)
- SAML (Security Assertions Markup Language)
- SASL (Simple Authentication and Security Layer)
- SNMP (Simple Network Management Protocol) Peer
- Sun Explorer Data Collector (仅限 Solaris)
- Sun Java Monitoring Framework
- Sun Java Web Console
- Tomcat Servlet JSP Container
- XML C Library (libxml)
- WSCL (Web services Common Library)

Java ES 2005Q4（发行版本 4）

本节列出了 Java Enterprise System 2005Q4 的内容。

发行版本 4 安装程序可选组件

在 Java ES 安装程序的组件选择页面中，可选组件按其帮助提供的服务进行了分组。下面的列表还显示了随每个组件一起安装的子组件。

通信和协作服务

- Sun Java System Messaging Server 6.2 2005Q4
- Sun Java System Calendar Server 6.2 2005Q4
- Sun Java System Instant Messaging 7.0.1 2005Q4
 - Instant Messaging Server Core ; 包括服务器和多路复用器软件
 - Instant Messaging Resources
 - Access Manager Instant Messaging Service
- Sun Java System Portal Server 6.3.1 2005Q4
- Sun Java System Portal Server Secure Remote Access 6.3.1 2005Q4
 - Secure Remote Access Core
 - Gateway
 - Netlet Proxy
 - Rewriter Proxy
- Sun Java System Communications Express 6.2 2005Q4
- Sun Java System Directory Preparation Tool 6.3 2005Q4
- Sun Java System Communications Services Delegated Administrator 6.3 2005Q4
 - Delegated Administrator Console and Utility
 - Delegated Administrator Server

Web 和应用程序服务

- Sun Java System Application Server Enterprise Edition 8.1 2005Q4
 - Domain Administration Server
 - Application Server Node Agent
 - Command Line Administration Tool
 - Load Balancing Plugin

可以与 Web Server 或 Apache Web Server 一起使用，在配置时可以选择。
默认情况下与 Web Server 一起使用。

- PointBase Database
- 样例应用程序
- Sun Java System Web Server 6.1 Service Pack 5 2005Q4
- Sun Java Web Proxy Server 4.0.1 2005Q4
- Sun Java System Message Queue Enterprise Edition 3.6 SP3 2005Q4
- Sun Java Service Registry 3.0

目录和身份认证服务

- Sun Java System Access Manager 7.0 2005Q4
 - Identity Management and Policy Services Core
 - Access Manager Administration Console
 - Common Domain Services for Federation Management
 - Access Manager SDK
- Sun Java System Directory Server 5.2 2005Q4
- Sun Java System Directory Proxy Server 5.2 2005Q4

可用性服务

- Sun Cluster 3.1 8/05
 - Sun Cluster Core
 - Sun Cluster Agents for Sun Java System
 - HA Sun Java System Directory Server
 - HA Sun Java System Administration Server
 - HA/Scalable Sun Java System Web Server
 - HA Sun Java System Message Queue
 - HA Sun Java System Application Server
 - HA Sun Java System Messaging Server
 - HA Sun Java System Calendar Server
 - HA Sun Java System Instant Messaging
- High Availability Session Store (HADB) 4.4.2

管理服务

- Sun Java System Administration Server 5.2 2005Q4

注	Solaris OS 支持 Sun Cluster 和 Sun Cluster Agents, 但 Linux OS 不支持它们。
----------	---

发行版本 4 共享组件

共享组件为可选组件提供本地服务和技术支持。安装 Java ES 组件时，安装程序会自动安装所需的共享组件（如果尚未安装）。

此版本的 Java ES 包括以下共享组件：

- ANT (Jakarta ANT Java/XML-based build tool)
- ACL (Apache Commons Logging)
- BDB (Berkeley Database)
- CAC (Common agent container)
- Derby Database
- ICU (International Components for Unicode)
- IM-SDK (Instant Messenger SDK)
- J2SE™ (Java 2 Platform, Standard Edition) platform 5.0
- JAF (JavaBeans™ Activation Framework)
- JATO (Java Studio Web Application Framework)
- JavaHelp™ Runtime
- JavaMail™ Runtime
- JAXB (Java Architecture for XML Binding) Runtime
- JAXP (Java API for XML Processing)
- JAXR (Java API for XML Registries) Runtime
- JAX-RPC (Java API for XML-based Remote Procedure Call) Runtime
- JCAPI (Java Calendar API)
- JDMK (Java Dynamic Management™ Kit) Runtime
- JSS (Java Security Services)
- KTSE (KT Search Engine)
- LDAP C SDK
- LDAP Java SDK
- MA (Mobile Access) Core

- NSPR (Netscape Portable Runtime)
- NSS (Network Security Services)
- SAAJ (SOAP runtime with Attachments API for Java)
- SASL (Simple Authentication and Security Layer)
- SEDC (Sun Explorer Data Collector, 仅限 Solaris)
- MFWK (Java ES Monitoring Framework)
- SJWC (Sun Java Web Console)
- WSCL (Web services Common Library)

Java ES 5（发行版本 5）

本节列出了 Java Enterprise System 5 的内容。

发行版本 5 安装程序可选组件

在 Java ES 安装程序的组件选择页面中，可选组件按其帮助提供的服务进行了分组。下面的列表还显示了随每个组件一起安装的子组件。注意，Java ES 安装程序不再支持通信服务组件。

协作服务

- Sun Java System Portal Server 7.1
- Sun Java System Portal Server Secure Remote Access 7.1
 - Gateway
 - Netlet Proxy
 - Rewriter Proxy

Web 和应用程序服务

- Sun Java System Application Server Enterprise Edition 8.2
 - Domain Administration Server
 - Application Server Node Agent
 - Command Line Administration Tool
 - Load Balancing Plugin

可以与 Web Server 或 Apache Web Server 一起使用，在配置时可以选择。
默认情况下与 Web Server 一起使用。

 - 样例应用程序
- Sun Java System Web Server 7.0
 - Web Server 7.0 CLI
 - Web Server 7.0 Core
 - Web Server 7.0 Samples
- Sun Java Web Proxy Server 4.0.4
- Sun Java System Message Queue 3.7UR1

- Service Registry 3.1
 - Service Registry Client Support
 - Service Registry Deployment Support

目录和身份认证服务

- Sun Java System Access Manager 7.1
 - Identity Management and Policy Services Core
 - Access Manager Administration Console
 - Common Domain Services for Federation Management
 - Access Manager SDK
 - Distributed Authentication
 - Client SDK
 - Session Failover Client
- Sun Java System Directory Server Enterprise Edition 6
 - Java Enterprise System Directory Server 6 Core Server
 - Java Enterprise System Directory Service Control Center
 - Sun Java System Directory Server Enterprise Edition 6 Command-Line Utilities
 - Sun Java System Directory Proxy Server 6 Core Server

可用性服务

- Sun Cluster 3.1
 - Sun Cluster Core
- Sun Cluster Geographic Edition 3.1 2006Q4
- Sun Cluster Agents 3.1
 - HA Sun Java System Application Server
 - HA Sun Java System Message Queue
 - HA Sun Java System Directory Server
 - HA Sun Java System Messaging Server
 - HA Sun Java System Application Server EE (HADB)

- HA/Scalable Sun Java System Web Server
- HA Instant Messaging
- HA Sun Java System Calendar Server
- ...
- Sun Java System High Availability Session Store 4.4.3

共享服务

- 所有共享组件
参见第 400 页中的“发行版本 5 共享组件”
- Sun Java System Monitoring Console 1.0
- Java DB
 - Java DBClient
 - Java DB Server

注	Solaris OS 支持 Sun Cluster 和 Sun Cluster Agents, 但 Linux OS 不支持它们。
----------	---

发行版本 5 共享组件

共享组件为可选组件提供本地服务和技术支持。安装 Java ES 组件时，安装程序会自动安装所需的共享组件（如果尚未安装）。

此版本的 Java ES 包括以下共享组件：

- ANT (Jakarta ANT Java/XML-based build tool)
- ACL (Apache Commons Logging)
- BDB (Berkeley Database)
- 仅适用于 Sun Cluster 的 CAC (Common agent container)
- CAC (Common agent container)
- FIS (FastInfoSet)
- ICU (International Components for Unicode)
- IM-SDK (Instant Messenger SDK)
- J2SE™ (Java 2 Platform, Standard Edition) platform 5.0
- JAF (JavaBeans™ Activation Framework)
- JATO (Java Studio Web Application Framework)
- JavaHelp™ Runtime
- JavaMail™ Runtime
- JAXB (Java Architecture for XML Binding) Runtime
- JAXP (Java API for XML Processing)
- JAXR (Java API for XML Registries) Runtime
- JAX-RPC (Java API for XML-based Remote Procedure Call) Runtime
- JAXWS (Java API for Web Services) Runtime
- JDMK (Java Dynamic Management™ Kit) Runtime
- JSS (Java Security Services)
- JSTL (JSP Standard Library Template)
- KTSE (KT Search Engine)
- LDAP C SDK

- LDAP Java SDK
- MA (Mobile Access) Core
- NSPR (Netscape Portable Runtime)
- NSS (Network Security Services)
- SAAJ (SOAP runtime with Attachments API for Java)
- SASL (Simple Authentication and Security Layer)
- SEDC (Sun Explorer Data Collector, 仅限 Solaris)
- MFWK (Java ES Monitoring Framework)
- SJWC (Sun Java Web Console)
- WSCL (Web services Common Library)
- XWSS (XML Web Services Security)

索引

A

Access Manager

- 产品组件依赖性 47
- 共享组件依赖性 43
- 升级方法 33
- 缩写 28, 42
- 子组件 393, 398

AccessManager-base 路径 235

AccessManagerConfig-base 路径 235

ACL 共享组件

- 全名 29
- 依赖性表中 43

amconfig 脚本 (Access Manager) 250

ampre71upgrade 脚本 (Access Manager) 248, 269

amupgrade 脚本 (Access Manager) 254, 275

ANT 共享组件

- 全名 29
- 依赖性表中 43

ANT_HOME 环境变量 294, 298, 319, 330, 333, 358, 360, 376, 378

Apache Commons Logging, 参见 ACL

Application Server

- 安装后配置 203
- 产品组件依赖性 47
- 共享组件依赖性 43
- 升级方法 33
- 缩写 28, 42
- 子组件 393, 397

AppServer7-base 路径 184

AppServer7Config-base 路径 184

AppServer8-base 路径 184

AppServer8Config-base 路径 184

AppServer8Install-base 路径 184

安全声明标记语言, 参见 SAML

B

BDB 共享组件

- 全名 29
- 依赖性表中 43

Berkeley Database, 参见 BDB

C

CAC 共享组件

- 全名 29
- 依赖性表中 43

Common agent container, 参见 CAC

操作系统

- 规划注意事项 40
- 双重升级 41
- 修补程序 40
- 支持矩阵 41

D

产品组件

对共享组件的依赖性 43

Java ES 2003Q4 （发行版本 1） 382

Java ES 2004Q2 （发行版本 2） 384

Java ES 2005Q1 （发行版本 3） 387

Java ES 2005Q4 （发行版本 4） 392

Java ES 5 （发行版本 5） 397

相互依赖性 47

D

db2bak 实用程序 244

Derby Database 共享组件 395

Directory Proxy Server

产品组件依赖性 47

共享组件依赖性 43

升级方法 33

缩写 28, 42

Directory Server

产品组件依赖性 47

共享组件依赖性 43

升级方法 33

缩写 28, 42

DirServer-base 路径 (Directory Server 6.0) 89, 105

F

FastInfoSet, 参见 FIS

FIS 共享组件

全名 29

依赖性表中 43

符号约定 21

G

共享组件

ava ES 2005Q4 （发行版本 4） 395

Java ES 2003Q4 （发行版本 1） 383

Java ES 2004Q2 （发行版本 2） 385

Java ES 2005Q1 （发行版本 3） 390

Java ES 5 （发行版本 5） 400

相互依赖性 45

依赖产品组件 43

H

HADB-base 路径 151

High Availability Session Store (HADB)

产品组件依赖性 47

共享组件依赖性 43

升级方法 33

缩写 28, 42

J

JAF 共享组件

全名 29

依赖性表中 43

Jakarta ANT Java/XML-based build tool, 参见 ANT

JATO 共享组件

全名 29

依赖性表中 43

Java 2 Platform, Standard Edition, 参见 J2SE

Java 2 Platform, Standard Edition, 参见 Java SE

Java API for XML Processing, 参见 JAXP

Java API for XML Registries, 参见 JAXR

Java API for XML-based Remote Procedure Call, 参见 JAX-RPC

Java Architecture for XML Binding, 参见 JAXB

Java Calendar API, 参见 JCAPI

Java DB

- 产品组件依赖性 47
- 共享组件依赖性 43
- 升级方法 33
- 缩写 28, 42

Java Dynamic Management Kit, 参见 JDMK

Java ES 2003Q4 (发行版本 1)

- 产品组件 382
- 共享组件 383

Java ES 2004Q2 (发行版本 2)

- 产品组件 384
- 共享组件 385

Java ES 2005Q1 (发行版本 3)

- 产品组件 387
- 共享组件 390

Java ES 2005Q4 (发行版本 4)

- 产品组件 392
- 共享组件 395

Java ES 5 (发行版本 5)

- 产品组件 397
- 共享组件 400

Java ES Monitoring Framework, 参见 MFWK

Java SE 共享组件

- 全名 29
- 依赖性表中 43

Java Security Services, 参见 JSS

Java Studio Web Application Framework, 参见 JATO

JAVA_HOME 319, 330, 333, 358, 360, 376, 378

JAVA_HOME 环境变量 252, 273, 294, 298

JavaBeans Activation Framework, 参见 JAF

JavaHelp 共享组件

- 全名 29
- 依赖性表中 44

JavaMail 共享组件

- 全名 29
- 依赖性表中 44

JAXB 共享组件

- 全名 29
- 依赖性表中 44

JAXP 共享组件

- 全名 29
- 依赖性表中 44

JAXR 共享组件

- 全名 29
- 依赖性表中 44

JAX-RPC 共享组件

- 全名 30
- 依赖性表中 44

JAXWS 共享组件

- 全名 30
- 依赖性表中 44

JCAPI 共享组件

- 全名 30
- 依赖性表中 44

ICU 共享组件

- 全名 29
- 依赖性表中 43

JDMK 共享组件

- 全名 30
- 依赖性表中 44

IM-SDK 共享组件

- 全名 29
- 依赖性表中 43

Instant Messenger SDK, 参见 IM-SDK

International Components for Unicode, 参见 ICU

JSP Standard Library Template, 参见 JSTL

JSP 文件, 自定义 238, 245, 250, 271

JSS 共享组件

- 全名 30
- 依赖性表中 44

JSTL 共享组件

- 全名 30
- 依赖性表中 44

脚本

- amconfig (Access Manager) 250
- ampre71upgrade (Access Manager) 248, 269
- amupgrade (Access Manager) 254, 275
- mqmigrate (Message Queue) 180
- mqupgrade (Message Queue) 179
- postInstall (Application Server) 204

K

K

KT Search Engine, 参见 KTSE

KTSE 共享组件

全名 [30](#)

依赖性表中 [44](#)

L

LDAP C Language SDK 共享组件 [400](#)

LDAP C SDK 共享组件

全名 [30](#)

依赖性表中 [44](#)

LDAP J SDK 共享组件 [401](#)

全名 [30](#)

依赖性表中 [44](#)

M

MA Core 共享组件

全名 [30](#)

依赖性表中 [44](#)

Message Queue

产品组件依赖性 [48](#)

共享组件依赖性 [43](#)

升级方法 [33](#)

缩写 [28, 42](#)

MessageQueue-base 路径 [161](#)

MFWK 共享组件

全名 [30](#)

依赖性表中 [44](#)

Mobile Access Core, 参见 MA core

Monitoring Console

产品组件依赖性 [48](#)

共享组件依赖性 [43](#)

缩写 [28](#)

mqupgrade 脚本 (Message Queue) [179, 180](#)

N

Netscape Portable Runtime, 参见 NSPR

Network Security Services, 参见 NSS

NSPR 共享组件

全名 [30](#)

依赖性表中 [44](#)

NSS 共享组件

全名 [30](#)

依赖性表中 [44](#)

P

Portal Server

产品组件依赖性 [48](#)

共享组件依赖性 [43](#)

升级方法 [34](#)

缩写 [28, 42](#)

Portal Server Secure Remote Access

产品组件依赖性 [48](#)

共享组件依赖性 [43](#)

升级方法 [34](#)

缩写 [29, 42](#)

PortalServer6-base 路径 [280, 346](#)

PortalServer6Config-base 路径 [280, 346](#)

PortalServer7-base 路径 [280, 346](#)

PortalServer7Config-base 路径 [280, 346](#)

PortalServer7Data-base 路径 [280, 346](#)

postInstall 脚本 (Application Server) [204](#)

R

RegistryDomainR4-base 路径 [209](#)

RegistryDomainR5-base 路径 [209](#)

rel4CAC-base-dir 路径 [66](#)

rel5CAC-admin-dir 路径 [66](#)

S

SAAJ 共享组件

全名 [30](#)

依赖性表中 [44](#)

SAML [257, 277](#)

SASL 共享组件

全名 [30](#)

依赖性表中 [44](#)

SEDC 共享组件

全名 [30](#)

依赖性表中 [44](#)

serverRoot 路径 (Directory Server 5.x) [89, 105](#)

Service Registry

产品组件依赖性 [49](#)

共享组件依赖性 [43](#)

升级方法 [34](#)

缩写 [29, 42](#)

ServiceRegistryR4-base 路径 [141, 209](#)

ServiceRegistryR5-base 路径 [209](#)

Simple Authentication and Security Layer, 参见 SASL

SJWC 共享组件

全名 [30](#)

依赖性表中 [44](#)

SML Web Services Security, 参见 XWSS

SOAP runtime with Attachments API for Java, 参见 SAAJ

Sun Cluster

产品组件依赖性 [49](#)

共享组件依赖性 [43](#)

升级方法 [34](#)

缩写 [29, 42](#)

Sun Cluster Geographic Edition

产品组件依赖性 [49](#)

共享组件依赖性 [43](#)

升级方法 [34](#)

缩写 [29, 42](#)

Sun Explorer Data Collector, 参见 SEDC

Sun Java Web Console, 参见 SJWC

SunClusterGeo-base 路径 [79](#)

双重升级

定义的 [41](#)

方法 [41](#)

组件支持摘要 [42](#)

W

Web Proxy Server

产品组件依赖性 [49](#)

共享组件依赖性 [43](#)

升级方法 [34](#)

缩写 [29, 42](#)

Web Server

产品组件依赖性 [49](#)

共享组件依赖性 [43](#)

升级方法 [34](#)

缩写 [29, 42](#)

Web services Common Library, 参见 WSCL

WebProxyServer-base 路径 [221](#)

WebServer6-base 路径 [119](#)

WebServer7-base 路径 [119](#)

WebServer7Config-base 路径 [119](#)

WSCL 共享组件

全名 [30](#)

依赖性表中 [44](#)

X

XWSS 共享组件

全名 [30](#)

依赖性表中 [44](#)

Y

Y

依赖性

- 产品组件, 对共享组件 [43](#)

- 产品组件之间 [47](#)

- 共享组件之间 [45](#)

印刷约定 [21](#)

约定

- 符号 [21](#)

- 印刷 [21](#)