Sun Java™ System

Sun Java Enterprise System 5 アップグレードガイド (UNIX 版)

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

Part No: 820-0893
2007 年 6 月

Copyright © 2007 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. All rights reserved.

Sun Microsystems, Inc. は、この製品に含まれるテクノロジに関する知的所有権を保持しています。特に限定されることなく、これらの知的所有権は http://www.sun.com/patents に記載されている 1 つ以上の米国特許および米国およびその他の国における 1 つ以上の追加特許または特許出願中のものが含まれている場合があります。

このソフトウェアは SUN MICROSYSTEMS, INC. の機密情報と企業秘密を含んでいます。SUN MICROSYSTEMS, INC. の書面による許諾を受けることなく、このソフトウェアを使用、開示、複製することは禁じられています。

U.S. Government Rights - Commercial software. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements.

この配布には、第三者が開発したソフトウェアが含まれている可能性があります。

本製品の一部は、カリフォルニア大学からライセンスされている Berkeley BSD システムに基づいていることがあります。UNIX は、X/Open Company, Ltd が独占的にライセンスしている米国およびその他の国における登録商標です。

Sun、Sun Microsystems、Sun のロゴマーク、Java、Solaris、JDK、Java Naming and Directory Interface、JavaMail、JavaHelp、J2SE、iPlanet、Duke のロゴマーク、Java Coffee Cup のロゴ、Solaris のロゴ、SunTone 認定ロゴマークおよび Sun ONE ロゴマークは、米国およびその他の国における米国 Sun Microsystems, Inc. (以下、米国 Sun Microsystems 社とします) の商標もしくは登録商標です。

すべての SPARC 商標は、米国 SPARC International, Inc. のライセンスを受けて使用している同社の米国およびその他の国における商標または登録商標です。SPARC 商標が付いた製品は、米国 Sun Microsystems 社が開発したアーキテクチャに基づくものです。

Legato および Legato のロゴマークは Legato Systems, Inc. の商標であり、Legato NetWorker は同社の商標または登録商標です。

Netscape Communications Corp のロゴマークは Netscape Communications Corporation の商標または登録商標です。

OPEN LOOK および Sun Graphical User Interface は、米国 Sun Microsystems 社が自社のユーザおよびライセンス実施権者向けに開発しました。米国 Sun Microsystems 社は、コンピュータ産業用のビジュアルまたはグラフィカルユーザインタフェースの概念の研究開発における米国 Xerox 社の先駆者としての成果を認めるものです。米国 Sun Microsystems 社は米国 Xerox 社から Xerox Graphical User Interface の非独占的ライセンスを取得しており、このライセンスは米国 Sun Microsystems 社のライセンス実施権者にも適用されます。

この製品は、米国の輸出規制に関する法規の適用および管理下にあり、また、米国以外の国の輸出および輸入規制に関する法規の制限を受ける場合があります。核、ミサイル、生物化学兵器もしくは原子力船に関連した使用またはかかる使用者への提供は、直接的にも間接的にも、禁止されています。このソフトウェアを、米国の輸出禁止国へ輸出または再輸出すること、および米国輸出制限対象リスト (輸出が禁止されている個人リスト、特別に指定された国籍者リストを含む) に指定された、法人、または団体に輸出または再輸出することは一切禁止されています。

本書は、「現状のまま」をベースとして提供され、商品性、特定目的への適合性または第三者の権利の非侵害の黙示の保証を含みそれに限定されない、明示的であるか黙示的であるかを問わない、なんらの保証も行われないものとします。

目 次

表目次 **15**

はじめに **19**

対象読者 20
本書で使用する表記規則 20
　管理インタフェース 20
　表記上の規則 20
　記号 21
　シェルプロンプト 22
関連マニュアル 22
　Java ES ドキュメントセット内のマニュアル 22
　Sun 製品ドキュメントの検索 24
オンライン上の Sun リソースへのアクセス 25
Sun テクニカルサポートへの問い合わせ 25
サードパーティーの Web サイト 25
ご意見、ご要望の送付先 26

第 1 章 アップグレードの計画 **27**
Java ES 5 のコンポーネント 28
　Release 5 の製品コンポーネント 28
　Release 5 の共有コンポーネント 29
Java ES アップグレードテクノロジ 31
　製品コンポーネントのアップグレード 31
　　製品コンポーネントのアップグレード手順 32
　　各製品コンポーネントに使用するアップグレード手順 34
　共有コンポーネントのアップグレード 35
アップグレードプロセス 36

アップグレード計画の考慮事項 37
アップグレードの依存関係 38
サポートされているアップグレードパスとアップグレード方針 39
選択的なアップグレードとすべてのアップグレード 42
複数インスタンスのアップグレード 43
オペレーティングシステムの考慮事項 44
必須のオペレーティングシステムパッチ 44
二重のアップグレード : Java ES ソフトウェアとオペレーティングシステムソフトウェア 44
オペレーティングシステムのアップグレード 47
Solaris 10 の複数ゾーン環境 47
Java ES コンポーネントの依存関係 47
共有コンポーネントに対する依存関係 48
製品コンポーネントに対する依存関係 52
アップグレード手順に関するガイドライン 58
特殊な場合 61
選択的なアップグレード : Application Server をアップグレードしない場合 61
Portal Server の IFR 7.0 の Java ES 5 へのアップグレード 62
Java ES 5 のアップグレードと Solaris 10 ゾーン 62
Java ES インストーラのゾーンサポート 63
製品コンポーネントのアップグレード 63
すべての共有コンポーネントの同期 64
推奨されるアップグレード方法 64
特殊な場合または例外 65
製品コンポーネントの特殊な場合 65
共有コンポーネントの特殊な場合 66

第 2 章 Java ES 共有コンポーネントのアップグレード 67
共有コンポーネントのアップグレードの概要 68
一般的な注意点 68
共有コンポーネントの同期 68
すべての共有コンポーネントの同期 70
Solaris 10 ゾーンに関する注意点 70
共有コンポーネントのアップグレード方法 71
共有コンポーネントのアップグレード手順 72
特別なアップグレード手順 73
Java SE のアップグレード手順 73
Java SE のシンボリックリンクの確認 74
現在の Java SE バージョンの確認 75
共通エージェントコンテナのアップグレード手順 75

第 3 章 Sun Cluster ソフトウェア 79
Sun Cluster ソフトウェアアップグレードの概要 80

Release 5 Sun Cluster ソフトウェアについて 80
Sun Cluster ソフトウェアのアップグレードロードマップ 81
Sun Cluster のデータ 82
Sun Cluster のアップグレード方針 82
互換性の問題 82
Sun Cluster の依存関係 82
二重のアップグレード 83
Sun Cluster ソフトウェアの Java ES Release 5 へのアップグレード 84
はじめに 84
Sun Cluster のアップグレード 85
アップグレード前の作業 85
Sun Cluster ソフトウェアのアップグレード 88
アップグレードの確認 89
アップグレード後の作業 89
アップグレードのロールバック 90

第 4 章 Sun Cluster Geographic Edition 91
Sun Cluster Geographic Edition のアップグレードの概要 92
Java ES Release 5 について 92
Java ES Release 5 へのアップグレードロードマップ 92
Sun Cluster Geographic Edition のデータ 93
Sun Cluster Geographic Edition のアップグレード方針 93
互換性の問題 93
依存関係 94
二重のアップグレード 94
Sun Cluster Geographic Edition の Version 3.1 8/05 からのアップグレード 95
はじめに 95
Version 3.1 8/05 の Sun Cluster Geographic Edition のアップグレード 96
アップグレード前の作業 96
3.1 8/05 の Sun Cluster Geographic Edition のアップグレード 98
アップグレードの確認 101
アップグレード後の作業 101
アップグレードのロールバック 101

第 5 章 Directory Server 103
Directory Server のアップグレードの概要 104
Java ES Release 5 について 104
Java ES Release 5 へのアップグレードロードマップ 104
Directory Server のデータ 106
Directory Server のアップグレード方針 107
互換性の問題 107
依存関係 107

二重のアップグレード 108
Java ES Release 4 からの Directory Server のアップグレード 109
はじめに 109
Release 4 の Directory Server のアップグレード 110
アップグレード前の作業 110
Release 4 の Directory Server のアップグレード 111
アップグレードの確認 115
アップグレード後の作業 116
アップグレードのロールバック 116
複数インスタンスのアップグレード 116
Directory Server レプリケーションの順次アップグレード 116
データサービスとしての Directory Server のアップグレード 116
Java ES Release 3 からの Directory Server のアップグレード 117
Java ES Release 2 からの Directory Server のアップグレード 117

第 6 章 Directory Proxy Server 119
Directory Proxy Server のアップグレードの概要 120
Java ES Release 5 について 120
Java ES Release 5 へのアップグレードロードマップ 121
Directory Proxy Server のデータ 122
Directory Proxy Server のアップグレード方針 122
互換性の問題 122
依存関係 123
二重のアップグレード 123
Java ES Release 4 からの Directory Proxy Server のアップグレード 124
はじめに 125
Release 4 の Directory Proxy Server のアップグレード 125
アップグレード前の作業 126
Release 4 の Directory Proxy Server のアップグレード 127
アップグレードの確認 129
アップグレード後の作業 129
アップグレードのロールバック 130
複数インスタンスのアップグレード 130
Java ES Release 3 からの Directory Proxy Server のアップグレード 130
Java ES Release 2 からの Directory Proxy Server のアップグレード 131

第 7 章 Web Server 133
Web Server のアップグレードの概要 134
Java ES Release 5 の Web Server について 134
Web Server のアップグレードロードマップ 135
Web Server のデータ 135
Web Server のアップグレード方針 136

互換性の問題 . . . 136
Web Server の依存関係 . . . 137
二重のアップグレード . . . 137
Java ES Release 4 からの Web Server のアップグレード . . . 138
はじめに . . . 138
Release 4 の Web Server のアップグレード . . . 139
アップグレード前の作業 . . . 139
Release 4 の Web Server のアップグレード . . . 140
アップグレードの確認 . . . 150
アップグレード後の作業 . . . 151
アップグレードのロールバック . . . 154
Java ES Release 3 からの Web Server のアップグレード . . . 155
Java ES Release 2 からの Web Server のアップグレード . . . 155

第 8 章 Java DB . . . 157
Java DB のアップグレードの概要 . . . 158
Java ES Release 5 について . . . 158
Java ES Release 5 へのアップグレードロードマップ . . . 158
Java DB のデータ . . . 159
Java DB のアップグレード方針 . . . 159
互換性の問題 . . . 159
依存関係 . . . 159
二重のアップグレード . . . 159
Java ES Release 4 からの Java DB のアップグレード . . . 161
はじめに . . . 161
Release 4 の Java DB のアップグレード . . . 162
アップグレード前の作業 . . . 162
Release 4 の Java DB のアップグレード . . . 163
アップグレードの確認 . . . 164
アップグレード後の作業 . . . 165
アップグレードのロールバック . . . 165
複数インスタンスのアップグレード . . . 165

第 9 章 High Availability Session Store . . . 167
HADB のアップグレードの概要 . . . 168
Java ES Release 5 の HADB について . . . 168
HADB のアップグレードロードマップ . . . 168
HADB のデータ . . . 169
HADB のアップグレード方針 . . . 169
互換性の問題 . . . 169
HADB の依存関係 . . . 169
二重のアップグレード . . . 169

Java ES Release 4 からの HADB のアップグレード 170
はじめに 170
Release 4 の HADB のアップグレード 171
アップグレード前の作業 171
Release 4 の HADB のアップグレード 173
アップグレードの確認 175
アップグレード後の作業 176
アップグレードのロールバック 176
Java ES Release 3 からの HADB のアップグレード 177

第 10 章 Message Queue 179
Message Queue のアップグレードの概要 180
Java ES Release 5 の Message Queue について 180
Message Queue のアップグレードロードマップ 181
Message Queue のデータ 182
Message Queue のアップグレード方針 183
互換性の問題 183
Message Queue の依存関係 183
二重のアップグレード 184
Java ES Release 4 からの Message Queue のアップグレード 185
はじめに 186
Release 4 の Message Queue のアップグレード 186
アップグレード前の作業 187
Release 4 の Message Queue のアップグレード 188
Message Queue のアップグレードの確認 189
アップグレード後の作業 189
アップグレードのロールバック 190
複数インスタンスのアップグレード 191
Java ES Release 3 からの Message Queue のアップグレード 191
Java ES Release 2 からの Message Queue のアップグレード 192
はじめに 192
Release 2 の互換性の問題 193
プロトコルの互換性 193
ブローカの互換性 193
管理対象オブジェクトの互換性 194
管理ツールの互換性 194
クライアントの互換性 195
Release 2 の Message Queue のアップグレード 195
アップグレード前の作業 195
Release 2 の Message Queue のアップグレード (Solaris) 197
Release 2 の Message Queue のアップグレード (Linux) 198
互換性パッケージのインストール (Linux) 200
Message Queue のアップグレードの確認 200

アップグレード後の作業 200
アップグレードのロールバック 201
複数インスタンスのアップグレード 201

第 11 章 Application Server 203
Application Server のアップグレードの概要 204
Java ES Release 5 の Application Server について 204
Application Server のアップグレードロードマップ 205
Application Server のデータ 208
Application Server のアップグレード方針 209
互換性の問題 209
Application Server の依存関係 209
二重のアップグレード 210
Java ES Release 4 からの Application Server のアップグレード 211
はじめに 211
Release 4 の Application Server のアップグレード 212
アップグレード前の作業 212
Release 4 の Application Server のアップグレード 214
アップグレードの確認 216
アップグレード後の作業 216
アップグレードのロールバック 216
Java ES Release 3 からの Application Server のアップグレード 216
Java ES Release 2 からの Application Server のアップグレード 217
はじめに 217
Release 2 の Application Server のアップグレード 218
アップグレード前の作業 218
Release 2 の Application Server のアップグレード 221
アップグレードの確認 225
アップグレード後の作業 225
アップグレードのロールバック 226
Solaris 10 の複数ゾーン環境における Solaris 付属の Application Server のアップグレード 226

第 12 章 Service Registry 229
Service Registry のアップグレードの概要 230
Java ES Release 5 について 230
Java ES Release 5 へのアップグレードロードマップ 230
Service Registry のデータ 231
Service Registry のアップグレード方針 231
互換性の問題 231
依存関係 232
二重のアップグレード 232
Java ES Release 4 からの Service Registry のアップグレード 233

はじめに 233
Release 4 の Service Registry のアップグレード 234
アップグレード前の作業 234
Release 4 の Service Registry のアップグレード 236
アップグレードの確認 239
アップグレード後の作業 239
アップグレードのロールバック 239
複数インスタンスのアップグレード 240

第 13 章 Web Proxy Server 241
Web Proxy Server のアップグレードの概要 242
Java ES Release 5 の Web Proxy Server について 242
Web Proxy Server のアップグレードロードマップ 242
Web Proxy Server のデータ 243
Web Proxy Server のアップグレード方針 243
互換性の問題 243
Web Proxy Server の依存関係 243
二重のアップグレード 244
Java ES Release 4 からの Web Proxy Server のアップグレード 244
はじめに 244
Release 4 の Web Proxy Server のアップグレード 245
アップグレード前の作業 245
Release 4 の Web Proxy Server のアップグレード (Solaris) 246
Release 4 の Web Proxy Server のアップグレード (Linux) 248
アップグレードの確認 249
アップグレード後の作業 249
アップグレードのロールバック (Solaris) 249
Version 3.6 からの Web Proxy Server のアップグレード 251
はじめに 251
Version 3.6 の Web Proxy Server のアップグレード 252
アップグレード前の作業 252
Version 3.6 の Web Proxy Server のアップグレード 253
アップグレードの確認 254
アップグレード後の作業 254
Version 3.6 アップグレードのロールバック 255

第 14 章 Access Manager 257
Access Manager のアップグレードの概要 258
Java ES Release 5 の Access Manager について 258
Access Manager のアップグレードロードマップ 259
Access Manager のデータ 260
Access Manager のアップグレード方針 262

互換性の問題 . . . 262
Access Manager の依存関係 . . . 263
Web コンテナのアップグレードシナリオ . . . 264
二重のアップグレード . . . 264
Java ES Release 4 からの Access Manager のアップグレード . . . 265
はじめに . . . 265
Release 4 の Access Manager の完全アップグレード . . . 266
アップグレード前の作業 . . . 266
Release 4 の Access Manager のアップグレード . . . 270
Access Manager のアップグレードの確認 . . . 280
アップグレード後の作業 . . . 280
アップグレードのロールバック . . . 281
複数インスタンスのアップグレード . . . 281
Release 4 の Access Manager SDK のみのアップグレード . . . 284
アップグレード前の作業 . . . 284
Release 4 の Access Manager SDK のアップグレード . . . 285
Access Manager SDK のアップグレードの確認 . . . 285
アップグレードのロールバック . . . 286
Java ES Release 3 からの Access Manager のアップグレード . . . 286
Release 3 の Access Manager のアップグレード . . . 287
複数インスタンスのアップグレード . . . 287
Java ES Release 2 からの Access Manager のアップグレード . . . 290
アップグレード前の作業 . . . 291
Access Manager が依存するコンポーネントのアップグレード . . . 291
ディレクトリスキーマをアップグレードする . . . 291
ディレクトリのインデックスを再生成する . . . 292
Release 2 の Access Manager のアップグレード . . . 293
Release 2 の Access Manager のアップグレード : Web Server Web コンテナ . . . 293
Release 2 の Access Manager のアップグレード : Application Server Web コンテナ . . . 293
Access Manager のアップグレードの確認 . . . 303
アップグレード後の作業 . . . 303
アップグレードのロールバック . . . 303
複数インスタンスのアップグレード . . . 304

第 15 章 Portal Server . . . 305
Portal Server のアップグレードの概要 . . . 306
Java ES Release 5 の Portal Server について . . . 306
Portal Server のアップグレードロードマップ . . . 306
Portal Server のデータ . . . 307
Portal Server のアップグレード方針 . . . 309
互換性の問題 . . . 309
Portal Server の依存関係 . . . 309
選択的なアップグレードに関する問題 . . . 310

Web コンテナのアップグレードシナリオ 311
二重のアップグレード 312
Java ES Release 4 からの Portal Server のアップグレード 313
はじめに 313
Release 4 の Portal Server のアップグレード 314
Release 4 のアップグレード前の作業 315
Release 4 の Portal Server のアップグレード (Solaris) 320
Release 4 の Portal Server のアップグレード (Linux) 324
アップグレードの確認 327
Release 4 のアップグレード後の作業 327
アップグレードのロールバック (Solaris) 335
アップグレードのロールバック (Linux) 336
複数インスタンスのアップグレード 337
Java ES Release 3 からの Portal Server のアップグレード 340
Release 3 アップグレード前の作業 : Portal Server の依存関係のアップグレード 340
Release 3 の Portal Server のアップグレード 341
Release 3 のアップグレード後の作業 341
複数インスタンスのアップグレード 342
Java ES Release 2 からの Portal Server のアップグレード 343
Release 2 のアップグレード前の作業 343
Release 2 の Portal Server のアップグレード 344
Release 2 の Portal Server のアップグレード : Web Server Web コンテナ 344
Release 2 の Portal Server のアップグレード : Application Server Web コンテナ 345
Release 2 のアップグレード後の作業 348
複数インスタンスのアップグレード 352
Portal Server の IFR 7.0 からのアップグレード 352
Portal Server IFR のアップグレードの概要 353
Portal Server IFR 7.0 のアップグレード 354
IFR 7 のアップグレード前の作業 354
Portal Server IFR 7.0 のアップグレード (Solaris) 355
IFR 7 の Portal Server のアップグレード (Linux) 358
IFR 7 のアップグレード手順 (Linux) 358
アップグレードの確認 360
IFR 7 のアップグレード後の作業 361
アップグレードのロールバック (Solaris) 367
アップグレードのロールバック (Linux) 369
複数インスタンスのアップグレード 369

第 16 章 Portal Server Secure Remote Access 371
Portal Server Secure Remote Access のアップグレードの概要 373
Java ES Release 5 の Portal Server Secure Remote Access について 373
Portal Server Secure Remote Access のアップグレードロードマップ 374
Portal Server Secure Remote Access のデータ 375

Portal Server Secure Remote Access のアップグレード方針 375
互換性の問題 375
Portal Server Secure Remote Access の依存関係 376
選択的なアップグレードに関する問題 377
二重のアップグレード 377
Java ES Release 4 からの Portal Server Secure Remote Access のアップグレード 378
はじめに 378
Release 4 の Portal Server Secure Remote Access のアップグレード 379
Release 4 のアップグレード前の作業 380
Release 4 の Portal Server Secure Remote Access のアップグレード (Solaris) 383
Release 4 の Portal Server Secure Remote Access のアップグレード (Linux) 386
アップグレードの確認 388
Release 4 のアップグレード後の作業 388
アップグレードのロールバック (Solaris) 390
アップグレードのロールバック (Linux) 391
複数インスタンスのアップグレード 391
Java ES Release 3 からの Portal Server Secure Remote Access のアップグレード 396
Portal Server Secure Remote Access が依存するコンポーネントのアップグレード 396
Release 3 の Portal Server Secure Remote Access のアップグレード 397
複数インスタンスのアップグレード 397
Java ES Release 2 からの Portal Server Secure Remote Access のアップグレード 398
Release 2 のアップグレード前の作業 398
Portal Server Secure Remote Access が依存するコンポーネントのアップグレード 398
ゲートウェイサービスエントリの削除 399
Release 2 の Portal Server Secure Remote Access のアップグレード 399
Release 2 のアップグレード後の作業 400
プロキシレットサービスのための Portal Server ドメインの設定 400
複数インスタンスのアップグレード 400
Portal Server Secure Remote Access の IFR 7.0 からのアップグレード 401
はじめに 401
Portal Server Secure Remote Access IFR 7.0 のアップグレード 402
アップグレード前の作業 402
Portal Server Secure Remote Access IFR 7.0 のアップグレード (Solaris) 402
Portal Server Secure Remote Access IFR 7.0 のアップグレード (Linux) 404
IFR 7 のアップグレード手順 (Linux) 404
アップグレードの確認 406
アップグレード後の作業 406
アップグレードのロールバック (Solaris) 406
アップグレードのロールバック (Linux) 407
複数インスタンスのアップグレード 407

付録 A　Java Enterprise System リリースの内容 409
Java ES 2003Q4 (Release 1) 410

Release 1 のインストーラで選択可能なコンポーネント 410
Release 1 の共有コンポーネント 411
Java ES 2004Q2 (Release 2) 412
Release 2 のインストーラで選択可能なコンポーネント 412
Release 2 の共有コンポーネント 413
Java ES 2005Q1 (Release 3) 415
Release 3 のインストーラで選択可能なコンポーネント 415
Release 3 の共有コンポーネント 418
Java ES 2005Q4 (Release 4) 420
Release 4 のインストーラで選択可能なコンポーネント 420
Release 4 の共有コンポーネント 423
Java ES 5 (Release 5) 425
Release 5 のインストーラで選択可能なコンポーネント 425
Release 5 の共有コンポーネント 428

索引 431

表目次

表 1 表記上の規則 21
表 2 記号の表記規則 21
表 3 シェルプロンプト 22
表 4 Java Enterprise System のドキュメント 23
表 1-1 Java ES 5 の製品コンポーネント 28
表 1-2 Java ES 5 の共有コンポーネント 29
表 1-3 製品コンポーネントのアップグレード手順 34
表 1-4 アップグレードプロセスのフェーズ 36
表 1-5 Java ES 5 (Release 5) へのアップグレードパス 39
表 1-6 選択的なアップグレードとすべてのアップグレードの比較 42
表 1-7 Java ES/ オペレーティングシステムサポートマトリックス 44
表 1-8 Java ES 5 製品コンポーネントでの二重アップグレードのサポート 46
表 1-9 Java ES 5 (Release 5) の製品コンポーネントの共有コンポーネントに対する依存関係 . 48
表 1-10 Java ES 製品コンポーネントの依存関係 53
表 1-11 Application Server を Release 5 にアップグレードしない場合に必要なパッチ 61
表 2-1 プラットフォームごとの Java SE へのシンボリックリンク 74
表 2-2 Java SE のバージョン確認の出力 75
表 2-3 共通エージェントコンテナのディレクトリパス 75
表 3-1 Java ES 5 (Release 5): Sun Cluster 3.1 8/05 ソフトウェアへのアップグレードパス 81
表 3-2 Sun Cluster のデータの使用法 82
表 3-3 Sun Cluster のバージョン確認の出力 86
表 4-1 Sun Cluster Geographic Edition ディレクトリパス 91
表 4-2 Java ES 5 (Release 5): Sun Cluster Geographic Edition 3.1 2006Q4 へのアップグレードパス 92
表 4-3 Sun Cluster Geographic Edition のデータの使用法 93
表 4-4 Sun Cluster Geographic Edition のバージョン確認の出力 96
表 5-1 Directory Server ディレクトリパス 103

表 5-2 Java ES 5 (Release 5): Directory Server 6.0 へのアップグレードパス 104
表 5-3 Directory Server のデータの使用法 106
表 5-4 Directory Server のバージョン確認の出力 110
表 6-1 Directory Proxy Server ディレクトリパス 119
表 6-2 Java ES 5 (Release 5): Directory Proxy Server 6.0 へのアップグレードパス 121
表 6-3 Directory Proxy Server のデータの使用法 122
表 6-4 Directory Proxy Server のバージョン確認の出力 126
表 7-1 Web Server ディレクトリパス 133
表 7-2 Java ES 5 (Release 5): Web Server 7.0 へのアップグレードパス 135
表 7-3 Web Server のデータの使用法 136
表 7-4 Web Server のデフォルト名 136
表 7-5 Web Server のバージョン確認の出力 139
表 7-6 WSInstall.properties の値 144
表 7-7 wadm 共通コマンドオプション 147
表 7-8 wadm migrate-server コマンドのオプションとオペランド 148
表 7-9 wadm create-instance コマンドのオプションとオペランド 150
表 7-10 wadm set-virtual-server-prop コマンドのオプションとオペランド 151
表 7-11 wadm deploy-cofig コマンドのオプションとオペランド 152
表 7-12 wadm set-search-collection-prop コマンドのオプションとオペランド 153
表 8-1 Java DB ディレクトリパス 157
表 8-2 Java ES 5 (Release 5): Java DB 10.1.3 へのアップグレードパス 158
表 8-3 Java DB のデータの使用法 159
表 8-4 Java DB のバージョン確認の出力 162
表 9-1 HADB ディレクトリパス 167
表 9-2 Java ES 5 (Release 5): HADB 4.4.3 へのアップグレードパス 168
表 9-3 HADB のデータの使用法 169
表 9-4 HADB のバージョン確認の出力 172
表 9-5 Solaris プラットフォーム用 HADB アップグレードパッケージのバージョン 173
表 10-1 Message Queue ディレクトリパス 179
表 10-2 Java ES 5 (Release 5): Message Queue 3.7 UR1 へのアップグレードパス 181
表 10-3 Message Queue のデータの使用法 (Solaris OS) 182
表 10-4 Message Queue のバージョン確認の出力 187
表 10-5 Message Queue のバージョン確認の出力 196
表 11-1 Application Server ディレクトリパス 203
表 11-2 Java ES 5 (Release 5): Application Server Enterprise Edition 8.2 へのアップグレードパス 205
表 11-3 Solaris オペレーティングシステムに付属する Application Server バージョンのアップグレードパス 206

表 11-4 Application Server のデータの使用法 208
表 11-5 Application Server のバージョン確認の出力 213
表 12-1 Service Registry ディレクトリパス 229
表 12-2 Java ES 5 (Release 5): Service Registry 3.1 へのアップグレードパス 230
表 12-3 Service Registry のデータの使用法 231
表 12-4 Service Registry のバージョン確認の出力 235
表 13-1 Web Proxy Server ディレクトリパス 241
表 13-2 Java ES 5 (Release 5): Web Proxy Server 4.0.4 へのアップグレードパス 242
表 13-3 Web Proxy Server のデータの使用法 243
表 13-4 Web Proxy Server のバージョン確認の出力 246
表 13-5 Solaris で Web Proxy Server をアップグレードするためのパッチ 247
表 13-6 Linux で Web Proxy Server をアップグレードするためのパッチ 248
表 13-7 Web Proxy Server のバージョン確認の出力 252
表 14-1 Access Manager ディレクトリパス 257
表 14-2 Java ES 5 (Release 5): Access Manager 7.1 へのアップグレードパス 259
表 14-3 Access Manager のデータの使用法 260
表 14-4 Access Manager のアップグレード用 Web コンテナのアップグレードシナリオ 264
表 14-5 Access Manager のバージョン確認の出力 267
表 14-6 Access Manager モバイルアクセスソフトウェアをアップグレードするためのパッチ . 270
表 14-7 モバイルアクセスのパッチパラメータ 271
表 14-8 Access Manager 設定パラメータ : ampre71upgrade 272
表 14-9 amconfig でサポートされる管理サーバーモード 274
表 14-10 Access Manager 設定パラメータ : amconfig 275
表 14-11 amconfig のパラメータ : Release 5 の Web Server 277
表 14-12 Access Manager 設定パラメータ : amupgrade 279
表 14-13 Access Manager モバイルアクセスソフトウェアをアップグレードするためのパッチ . 294
表 14-14 モバイルアクセスのパッチパラメータ 295
表 14-15 Access Manager 設定パラメータ : ampre71upgrade 295
表 14-16 Access Manager 設定パラメータ : amconfig 299
表 14-17 Access Manager 設定パラメータ : amupgrade 302
表 15-1 Portal Server ディレクトリパス 305
表 15-2 Java ES 5 (Release 5): Portal Server 7.1 へのアップグレードパス 307
表 15-3 Portal Server のデータの使用法 307
表 15-4 Portal Server のアップグレード用 Web コンテナのアップグレードシナリオ 311
表 15-5 Portal Server のバージョン確認の出力 315
表 15-6 Web コンテナのアップグレードシナリオ別の psupgrade スクリプトで必要な情報 . . 317
表 15-7 JVM 設定の場所 319

表 15-8 Portal Server IFR を Release 5 にアップグレードするためのパッチ 353
表 16-1 Portal Server Secure Remote Access ディレクトリパス 372
表 16-2 Java ES 5 (Release 5): Portal Server Secure Remote Access 7.1 へのアップグレードパス 374
表 16-3 Portal Server Secure Remote Access のデータの使用法 375
表 16-4 Portal Server Secure Remote Access のバージョン確認の出力 380
表 16-5 Portal Server Secure Remote Access IFR を Release 5 にアップグレードするためのパッチ 401

はじめに

『Java Enterprise System アップグレードガイド (UNIX 版)』には、Sun Solaris™ オペレーティングシステム (Solaris OS) または Red Hat Enterprise Linux (RHEL) オペレーティングシステム環境での Sun Java™ Enterprise System (Java ES) ソフトウェアのアップグレードに必要な情報が含まれています。HP-UX 環境でのアップグレードについては取り上げていません。

本書では、Java ES 2004Q2 (Release 2)、Java ES 2005Q1 (Release 3)、および Java ES 2005Q4 (Release 4) から Java ES5 (Release 5) へのアップグレードについて説明します。

この章で説明する項目は次のとおりです。

- 20 ページの「対象読者」
- 20 ページの「本書で使用する表記規則」
- 22 ページの「関連マニュアル」
- 25 ページの「オンライン上の Sun リソースへのアクセス」
- 25 ページの「Sun テクニカルサポートへの問い合わせ」
- 25 ページの「サードパーティーの Web サイト」
- 26 ページの「ご意見、ご要望の送付先」

対象読者

本書は、システム管理者、および Java ES ソフトウェアのアップグレードが必要なソフトウェア技術者を対象としています。

本書は、次の事項に習熟している方を対象に記述されています。

- エンタープライズレベルのソフトウェア製品のインストール
- ご使用の環境に現在配備されている Java ES コンポーネント
- サポートする Java ES プラットフォーム上のシステム管理およびネットワーク
- クラスタリングモデル (クラスタリングソフトウェアをインストールする場合)

本書で使用する表記規則

次の表は、本書で使用する表記規則について説明します。

管理インタフェース

この『アップグレードガイド』で説明しているほとんどのアップグレード手順では、グラフィカルユーザーインタフェース (GUI) とコマンド行インタフェースの 2 つの管理インタフェースが使用できます。

この『アップグレードガイド』では、Java ES コンポーネントのアップグレード手順について説明するときは、ほとんどの場合、コマンド行インタフェースを使用しています。コマンド行インタフェースは、Java ES 配備をアップグレードするスクリプトに使用できるので、必要に応じて簡単に手順を繰り返すことができます。

ただし、手順の中で Java ES インストーラを使用するときは、対話型のテキストベースのインタフェースではなく、GUI インタフェースについて説明します。Java ES インストーラセッションは、状態ファイルに保存できます。必要な場合は、このファイルを使用して手順を繰り返すことができます。

表記上の規則

次の表は、本書で使用する表記上の規則について説明したものです。

表 1　表記上の規則

表記	意味	例
AaBbCc123 (モノスペース)	API および言語の要素、HTML タグ、Web サイトの URL、コマンド名、ファイル名、ディレクトリパス名、画面出力の表示、サンプルコード。	.login ファイルを編集します。 ls -a を使用してすべてのファイルを表示します。 % You have mail.
AaBbCc123 (太字のモノスペース)	ユーザーが入力するテキストを強調し、コンピュータからの画面出力と区別します。	% **su** Password:
AaBbCc123 (斜体)	実際の名前または値によって置き換えられるコマンドまたはパス名の可変部分。	これらを *class* オプションと呼びます。 ファイルは *install-dir*/bin ディレクトリにあります。

記号

次の表は、このマニュアルで使用される記号の表記規則を示しています。

表 2　記号の表記規則

記号	説明	例	意味
[ ]	省略できるコマンドオプションを含みます。	ls [-l]	-l オプションは省略可能です。
	必須コマンドオプションの選択肢を含みます。	-d {y|n}	-d オプションは、引数 y または n のいずれかを使用する必要があります。
-	同時に押すキーを連結します。	Control-A	Ctrl キーと A キーを同時に押します。
+	連続して押すキーを連結します。	Ctrl+A+N	Ctrl キーを押し、放してから、以後のキーを続けて押します。
>	グラフィカルユーザーインタフェースで選択するメニュー項目を示します。	「ファイル」>「新規」>「テンプレート」	「ファイル」メニューから「新規」を選択します。「新規」サブメニューから、「テンプレート」を選択します。

シェルプロンプト

次の表は、このマニュアルで使用されるシェルプロンプトを示しています。

表 3　　シェルプロンプト

シェル	プロンプト
UNIX または Linux の C シェル	*machine-name*%
UNIX または Linux の C シェルのスーパーユーザー	*machine-name*#
UNIX または Linux の Bourne シェルおよび Korn シェル	$
UNIX または Linux の Bourne シェルおよび Korn シェルのスーパーユーザー	#
Windows のコマンド行	C:¥

関連マニュアル

http://docs.sun.com℠ Web サイトでは、Sun テクニカルマニュアルにオンラインでアクセスできます。特定のマニュアルのタイトルまたは件名のアーカイブを参照し、検索できます。

Java ES ドキュメントセット内のマニュアル

Java ES マニュアルは、PDF (Portable Document Format) 形式および HTML (Hypertext Markup Language) 形式のオンラインファイルとして用意されています。どちらの形式のファイルも、障害を持つユーザーにも参照可能です。Sun™ のマニュアルには、次の Web サイトからアクセスできます。

http://docs.sun.com/app/docs?l=ja

Java ES マニュアルには、システム全体に関する情報とそのコンポーネントに関する情報が記載されています。このマニュアルには次の場所からアクセスできます。

http://docs.sun.com/coll/1657.1

次の表は、Java ES ドキュメントセットに含まれる、システムレベルのマニュアルを示しています。左の列は各ドキュメントの名称とアクセスできる URL、右の列はドキュメントの簡単な説明を示しています。

表 4　Java Enterprise System のドキュメント

マニュアル名	内容
『Java Enterprise System 5 リリースノート (UNIX 版)』 http://docs.sun.com/app/docs/doc/820-0453?l=ja	既知の問題など、Java ES に関する最新の情報が記載されています。これ以外に、コンポーネントごとにリリースノートがあります。
『Sun Java Enterprise System 5 技術の概要』 http://docs.sun.com/doc/820-0886?l=ja	Java ES の技術的および概念的な基礎について説明します。コンポーネント、アーキテクチャー、プロセス、および機能について説明しています。
『Sun Java Enterprise System 配備計画ガイド』 http://docs.sun.com/doc/819-3449?l=ja	Java ES に基づく企業向け配備ソリューションの計画および設計に関する情報を提供します。配備の計画および設計に関する基本的概念と原則を示し、ソリューションのライフサイクルについて説明し、Java ES に基づくソリューションを計画する際に使用する高度な例と方針を提供します。
『Sun Java Enterprise System 5 インストール計画ガイド』 http://docs.sun.com/doc/820-0879?l=ja	Java ES の配備に関し、ハードウェア、オペレーティングシステム、およびネットワーク面の実装仕様の開発に役立ちます。インストール計画および設定計画で検討すべきコンポーネントの依存関係などの問題について説明します。
『Java Enterprise System 5 インストールガイド (UNIX 版)』 http://docs.sun.com/doc/820-0101?l=ja	Solaris オペレーティングシステムまたは Linux オペレーティングシステムでの Java ES のインストール手順について説明します。インストール後にコンポーネントを設定する方法、および設定したコンポーネントが正常に機能するかどうかを確認する方法についても説明します。
『Sun Java Enterprise System 5 インストールリファレンス (UNIX 版)』 http://docs.sun.com/doc/820-0587?l=ja	設定パラメータに関する追加情報、設定計画に使用できるワークシート、およびデフォルトのディレクトリやポート番号などの参照情報の一覧を提供します。
『Java Enterprise System 5 アップグレードガイド (UNIX 版)』 http://docs.sun.com/doc/820-0893?l=ja	Solaris オペレーティングシステムまたは Linux オペレーティングシステムでの Java ES のアップグレード手順について説明します。

表 4　Java Enterprise System のドキュメント (続き)

マニュアル名	内容
『Sun Java Enterprise System 5 監視ガイド』 http://docs.sun.com/doc/820-1590?l=ja	製品コンポーネントごとの Monitoring Framework の設定、および Monitoring Console を使用したリアルタイムデータの表示としきい値アラームの設定の手順について説明します。
『Sun Java Enterprise System 用語集』 http://docs.sun.com/doc/819-4629?l=ja	Java ES のドキュメントで使用される用語について説明します。

Sun 製品ドキュメントの検索

docs.sun.com Web サイトから Sun 製品ドキュメントを検索するだけではなく、ほかの検索エンジンを使用することもできます。その際、検索フィールドに次の構文を入力します。

検索する用語 site:docs.sun.com

たとえば、「ブローカ」を検索するには、次のように入力します。

ブローカ site:docs.sun.com

検索範囲にほかの Sun Web サイト (たとえば、java.sun.com、www.sun.com、developers.sun.com など) を含めるには、検索フィールドで docs.sun.com の代わりに sun.com を使用します。

オンライン上の Sun リソースへのアクセス

製品のダウンロード、プロフェショナルサービス、パッチとサポート、および開発者向け追加情報については、次の Web サイトにアクセスしてください。

- ダウンロードセンター
  http://www.sun.com/software/download/
- クライアントソリューション
  http://www.sun.com/service/sunjavasystem/sjsservicessuite.html
- Sun Enterprise サービス、Solaris パッチ、およびサポート
  http://sunsolve.sun.com/
- 開発者向け情報
  http://jp.sun.com/developers/

次の場所には、Java Enterprise System およびそのコンポーネントに関する情報が用意されています。

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

Sun テクニカルサポートへの問い合わせ

製品のドキュメントで解決できない、本製品に関する技術的な質問の問い合わせ先については、http://jp.sun.com/service/support/ を参照してください。

サードパーティーの Web サイト

Sun は、このマニュアルに記載されているサードパーティー Web サイトの利用の可能性について責任を負いません。Sun は、このようなサイトまたはリソースで得られるあらゆる内容、広告、製品、およびその他の資料を保証するものではなく、責任または義務を負いません。Sun は、このようなサイトまたはリソースで得られるあらゆるコンテンツ、製品、またはサービスによって生じる、または生じたと主張される、または使用に関連して生じる、または信頼することによって生じる、いかなる損害または損失についても責任または義務を負いません。

ご意見、ご要望の送付先

Sun ではマニュアルの品質向上のため、お客様のご意見、ご要望をお受けしております。

コメントをお送りになる場合は、http://docs.sun.com/app/docs?l=ja にアクセスして「コメントの送信」をクリックしてください。オンラインフォームで、ドキュメントのタイトルと Part No. を入力します。Part No. は、マニュアルのタイトルページまたは最上部に記載されている 7 桁または 9 桁の番号です。

第 1 章

アップグレードの計画

この章では、Sun Solaris™ オペレーティングシステムまたは Red Hat Enterprise Linux (単に Linux と呼ぶ) オペレーティングシステム環境での Sun Java™ Enterprise System (Java ES) ソフトウェアの Java ES 5 へのアップグレード計画について説明します。

この章で説明する内容は、次のとおりです。

- 28 ページの「Java ES 5 のコンポーネント」
- 31 ページの「Java ES アップグレードテクノロジ」
- 36 ページの「アップグレードプロセス」
- 37 ページの「アップグレード計画の考慮事項」
- 47 ページの「Java ES コンポーネントの依存関係」
- 58 ページの「アップグレード手順に関するガイドライン」
- 61 ページの「特殊な場合」
- 62 ページの「Java ES 5 のアップグレードと Solaris 10 ゾーン」

Java ES 5 のコンポーネント

この節では、Java ES ソフトウェアのアップグレード計画の概説として、Java ES 5 (Release 5) に含まれるコンポーネントについて検討します。アップグレードシナリオによっては、これらのコンポーネントの 1 つ以上を Release 5 バージョンにアップグレードする必要が生じる可能性があります。

『Sun Java Enterprise System 5 技術の概要』(http://docs.sun.com/doc/819-3583?l=ja) で説明されるように、Java ES コンポーネントは数種類のタイプに分類されます。

- **製品コンポーネント** : Java ES 製品コンポーネントは、次の 2 種類から構成されます。
  - 主な Java ES インフラストラクチャーサービスを提供するシステムサービスコンポーネント
  - システムサービスを機能拡張するサービス品質コンポーネント

  製品コンポーネントは、Java ES インストーラで選択できます。
- **共有コンポーネント** : Java ES 共有コンポーネントは Java ES 製品コンポーネントが依存するライブラリで、ローカルで共有されます。共有コンポーネントは、Java ES インストーラによって自動的にインストールされます。インストールされる共有コンポーネントは、インストールされる製品コンポーネントによって異なります。

Release 5 の製品コンポーネント

次の表は、Release 5 の製品コンポーネントを、以降の表で使用する略称とともにアルファベット順に示しています。サービス品質コンポーネントについては、それらが提供するサービス拡張のタイプを示しています。

表 1-1 Java ES 5 の製品コンポーネント

製品コンポーネント	略称	バージョン	タイプ
Access Manager	AM	7.1	システムサービスコンポーネント
Application Server	AS	8.2	システムサービスコンポーネント
Directory Proxy Server	DPS	6.0	サービス品質 : アクセスコンポーネント
Directory Server	DS	6.0	システムサービスコンポーネント
High Availability Session Store	HADB	4.4.3	サービス品質 : 可用性コンポーネント
Java DB	JavaDB	10.2	システムサービスコンポーネント
Message Queue	MQ	3.7 UR1	システムサービスコンポーネント

表 1-1　Java ES 5 の製品コンポーネント (続き)

製品コンポーネント	略称	バージョン	タイプ
Monitoring Console	MC	1.0	サービス品質 : 管理コンポーネント
Portal Server	PS	7.1	システムサービスコンポーネント
Portal Server Secure Remote Access	PSRA	7.1	サービス品質 : アクセスコンポーネント
Service Registry	SR	3.1	システムサービスコンポーネント
Sun Cluster	SC	3.1 8/05	サービス品質 : 可用性コンポーネント
Sun Cluster Geographic Edition	SCG	2006Q4	サービス品質 : 可用性コンポーネント
Web Proxy Server	WPS	4.0.4	サービス品質 : アクセスコンポーネント
Web Server	WS	7.0	システムサービスコンポーネント

Release 5 の共有コンポーネント

次の表は、Release 5 の共有コンポーネントを、以降の表で使用する略称とともにアルファベット順で示しています。

表 1-2　Java ES 5 の共有コンポーネント

共有コンポーネント	バージョン	略称
Apache Commons Logging	1.0.3	ACL
Jakarta ANT Java/XML ベースの構築ツール	1.6.5	ANT
Berkeley Database	4.2.52	BDB
共通エージェントコンテナ	1.1 および 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

表 1-2　Java ES 5 の共有コンポーネント (続き)

共有コンポーネント	バージョン	略称
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
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 および 3.1.11	JSS および 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 コア	6.2	MA コア
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 (Solaris のみ)	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 アップグレードテクノロジ

単独で Java ES のすべてのコンポーネントをアップグレードできるシステムユーティリティーはありません。また、以降の節で説明するように、製品コンポーネントと共有コンポーネントのアップグレードでは、特性もアップグレードテクノロジもそれぞれ異なります。

製品コンポーネントのアップグレード

Java ES 製品コンポーネントの Release 5 へのアップグレードは、この『アップグレードガイド』で説明するコンポーネント固有のアップグレード手順を使用して、コンポーネントごと、コンピュータごとに実行します。

製品コンポーネントのアップグレードは、場合によっては以前のバージョンのコンポーネントと互換性がなくなる主要な機能アップグレードから、以前のバージョンと完全互換性のあるバグ修正アップグレードまで、多岐に渡ります。Java ES コンポーネント間の依存関係により、アップグレードの性質によっては、別のコンポーネントのアップグレードが必要になる場合もあります。

Java ES 製品コンポーネントのアップグレードには、次のように 2 種類の基本的な操作があります。これにより、Java ES 製品コンポーネントの初期インストールおよび初期設定が反映されます。

- **ソフトウェアアップグレードのインストール** : アップグレードされたソフトウェアにより、既存のソフトウェアが拡張、修復、または置換されます。ソフトウェアのインストールは、既存のソフトウェアパッケージへのパッチの適用、既存のパッケージの選択的な置換、新しいパッケージのインストール、またはコンポーネントソフトウェアの完全な再インストールによって行われます。
- **再設定** : 再設定は、アップグレードされたソフトウェアをサポートするのに必要な設定データ、ユーザーデータ、または動的なアプリケーションデータの変更をすべて網羅します。データの変更とは、データの追加、データ形式の変更 (プロパティーファイル内でもデータベーススキーマ内でも)、または新しい場所へのデータの移行を意味します。再設定には、ユーザーによる手順の実行が必要な場合と、自動的に実行される場合があります。時には、再設定のために、Web コンテナへのコンポーネントソフトウェアの再配備が必要になります。

さらに、通常 Java ES 製品コンポーネントのアップグレードには、アップグレード前の作業が含まれます。また場合によっては、アップグレード後、それを稼働させる前に必要となる手順も含まれます。

製品コンポーネントのアップグレード手順

アップグレードされたソフトウェアのインストールとコンポーネントの再設定の実行に使用するコンポーネント固有のアップグレード手順には、次のアップグレード手順が含まれます。

- Java ES インストーラのアップグレード機能の使用
- 製品コンポーネントの新規インストールの実行
- コンポーネント固有のアップグレードユーティリティーの実行
- 既存のコンポーネントパッケージへのパッチの適用

Java ES インストーラのアップグレード機能の使用

Release 5 のインストーラは、いくつかの特殊な場合 (Application Server、Message Queue、HADB、および Java DB) では製品コンポーネントのアップグレードを実行するアップグレード機能を備えています。Java ES インストーラは、以前にインストールされたこれらの製品コンポーネントのリリースバージョンを検出すると、それらのコンポーネントを「アップグレード可」として指定します。

これらのコンポーネントをアップグレードする前に、インストーラは共有コンポーネントの現在のバージョンと以前のバージョンを確認します。インストーラは、選択されたコンポーネントに必要な共有コンポーネントが以前のバージョンであるか、または欠落していることを検出すると、現在インストールされているすべての共有コンポーネントをアップグレードし、選択されたコンポーネントに必要な欠落している共有コンポーネントをインストールします。場合によっては (特に Application Server の場合)、インストーラはアップグレードされるコンポーネントが依存する製品コンポーネントもアップグレードします。

インストーラは、以前のバージョンのパッケージを削除し、Release 5 の製品コンポーネントパッケージをインストールして、(必要に応じて) アップグレードされた製品コンポーネントを再設定します。ただし、Solaris 9 オペレーティングシステムに付属する Application Server の場合、インストーラはパッケージを削除しません (218 ページの「Release 2 の Application Server のアップグレード」を参照)。

Solaris 10 オペレーティングシステムのゾーン機能を使用する場合は、特別な注意事項があります。63 ページの「Java ES インストーラのゾーンサポート」を参照してください。

製品コンポーネントの新規インストールの実行

一部の製品コンポーネントは、Java ES インストーラを使用してコンポーネントの新規インストールを実行することによりアップグレードされます。最初に以前のバージョンのパッケージを削除してから、同じパスに Release 5 をインストールします。または、並列パスに Release 5 をインストールして、以前のバージョンをそのまま残します。

どちらの場合も、以前のバージョンの設定データを新しいインストールに移行するか、新しい設定を実行するか、またはその両方を組み合わせることにより、製品コンポーネントを再設定します。一部の製品コンポーネントには、そのコンポーネントの設定データを再設定または移行するためのユーティリティーが用意されています。

コンポーネント固有のアップグレードユーティリティーの実行

一部の製品コンポーネントには、そのコンポーネントの Release 5 へのアップグレードを自動化するためのアップグレードユーティリティーまたはスクリプトが用意されています。このユーティリティーは通常、ソフトウェアパッケージのアップグレードとアップグレードの一部として必要な再設定の両方を実行します。Web コンテナに配備されているコンポーネントの場合、このユーティリティーは通常、アップグレードされたコンポーネントソフトウェアを Web コンテナに再配備します。

既存のコンポーネントパッケージへのパッチの適用

一部の製品コンポーネントについては、既存のソフトウェアパッケージに手動でパッチを適用することによってアップグレードを実行します。Solaris プラットフォームと Linux プラットフォームでは、インストールされているソフトウェアパッケージを管理したり、パッケージレジストリによってそれらのパッケージに対する変更を追跡したりするために、同様のテクノロジが採用されていますが、プラットフォーム間におけるパッチ適用テクノロジの違いは、アップグレード手順に影響を与えます。

- **Solaris プラットフォーム**: パッケージのインストールと削除は、Solaris の `pkgadd` コマンドと `pkgrm` コマンドによって行われます。パッケージをインストールすると、`patchadd` コマンドと `patchrm` コマンドによって適用または削除されるパッチを使用して、パッケージの内容を修正できます。Solaris パッケージに対するパッチは、次の SunSolve Web サイトで配布されています。
  http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access

  Solaris パッチでは、1 つまたは複数のパッケージを処理できます。`patchadd` コマンドは、`patchrm` コマンドを使用したパッチの削除を容易に行えるように、パッチされるパッケージのバックアップを保存します。パッチは、パッチ番号とリビジョン番号で構成されるパッチ ID によって識別されます。リビジョン番号は、パッチが修正されるたびに増加します。

- **Linux プラットフォーム**: Red Hat Enterprise Linux パッケージ (RPM) は、`rpm` コマンドによってインストールまたは更新できます。ただし、パッケージをインストールすると、パッチを使用してパッケージの内容を修正することはできません。RPM パッケージを更新するには、現在のパッケージを新しいパッケージで置き換えるための `rpm -U` コマンドオプションを使用します。

  便宜上、多くの RPM パッケージアップグレードは Java ES Release 5 ディストリビューションだけではなく、次の SunSolve Web サイトでも配布されています。
  http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access

SunSolve による配布では、RPM パッケージはパッチとしてカプセル化され、Solaris パッチと類似したパッチ ID とリビジョン番号が割り当てられます。これらの Linux パッチには 1 つまたは複数の RPM パッケージが含まれることがあり、それぞれが一意の RPM 名、RPM 番号、およびリビジョン番号によって識別されます。リビジョン番号は、RPM パッケージが修正されるたびに増加します。

各製品コンポーネントに使用するアップグレード手順

次の表に、各製品コンポーネントを Release 5 にアップグレードするために使用するアップグレード手順を示します。

表 1-3 製品コンポーネントのアップグレード手順

製品コンポーネント	アップグレードするソフトウェアのインストール	再設定
Access Manager	パッケージを置換 : パッケージ削除スクリプトを使用してから、新規インストールを行います	`amconfig` スクリプトと `amupgrade` スクリプトを使用して再設定と Web コンテナへの再配備を行います
Application Server	パッケージを置換 : Java ES インストーラのアップグレード機能を使用します	なし。ただし、Release 2 からアップグレードする場合は `postinstall` スクリプトと `asupgrade` スクリプトを使用します
Directory Proxy Server	以前のパッケージを置換せずに新規インストールを実行します。	手動で再設定します
Directory Server	以前のパッケージを置換せずに新規インストールを実行します。	`dsmig` コマンドを使用してディレクトリデータを移行します
High Availability Session Store	パッケージを置換 : Java ES インストーラのアップグレード機能を使用するか、並行して新規インストールを実行します。	なし
Java DB	パッケージを置換 : Java ES インストーラのアップグレード機能を使用します	なし
Message Queue	パッケージを置換 : Java ES インストーラのアップグレード機能または `mqupgrade` スクリプト (Release 2 からの場合) を使用します	なし。ただし、Linux で Release 2 からアップグレードする場合は、`mqmigrate` スクリプトを使用します

表 1-3　製品コンポーネントのアップグレード手順 (続き)

製品コンポーネント	アップグレードするソフトウェアのインストール	再設定
Portal Server	パッケージを置換: psupgrade スクリプトを使用します	psupgrade スクリプトを使用して再設定と Web コンテナへの再配備を行います
Portal Server Secure Remote Access	パッケージを置換: psupgrade スクリプトを使用します	psupgrade スクリプトを使用して再設定します
Service Registry	以前のパッケージを置換せずに新規インストールを実行します。	手動で再設定し、ant upgrade スクリプトを使用して Application Server のドメインに再配備します
Sun Cluster	パッケージを置換: scinstall スクリプトを使用してバイナリを置換します	scinstall スクリプトを使用して設定を移行します
Sun Cluster Geo	パッケージを置換: uninstall スクリプトを使用してから、新規インストールを行います	なし
Web Proxy Server	バイナリにパッチを適用します	なし
Web Server	以前のパッケージを置換せずに新規インストールを実行します。	wadm migrate-server コマンドを使用してサーバーインスタンスの設定を移行します

共有コンポーネントのアップグレード

Java ES 共有コンポーネントに依存する製品コンポーネントのアップグレードでは、共有コンポーネントのアップグレードも必要です。

共有コンポーネントのアップグレードには、コンポーネントの再設定もアップグレード前後の手順も必要ありません。また、共有コンポーネントのアップグレードは、以前のバージョンにロールバックできません。

Java ES 共有コンポーネントの数が多く (30 個程度)、共有コンポーネントと製品コンポーネントの相互作用が複雑なため、単一のオペレーティングシステムインスタンス内のすべての共有コンポーネントを同じ Java ES リリースバージョンに同期する必要があります。オペレーティングシステムインスタンスとは、Solaris 9、Solaris 10、または Red Hat Enterprise Linux オペレーティングシステムを実行する 1 台のコンピュータ、あるいは Solaris 10 オペレーティングシステムを実行するいずれかの仮想オペレーティングシステム環境 (ゾーン) を意味します。

同期の要件のため、Java ES 共有コンポーネントを 1 つずつアップグレードするべきではなく、同時に Release 5 バージョンにアップグレードする必要があります。

共有コンポーネントの Release 5 への同期は、Java ES インストーラを使用して行います。このインストーラは、製品コンポーネントのアップグレードを実行するとき (32 ページの「Java ES インストーラのアップグレード機能の使用」を参照)、または製品コンポーネントの新規インストールを実行するときに、共有コンポーネントを同期します。このインストーラは、既存の共有コンポーネントをアップグレードし、不足している共有コンポーネントをインストールする同期機能も備えています。この機能の詳細については、70 ページの「すべての共有コンポーネントの同期」を参照してください。

アップグレードプロセス

Java ES のアップグレードプロセスにはいくつかのフェーズがあり、これらのフェーズは、通常、本稼働環境で実行される前にまずステージング環境で実行されます。ステージング環境を使用することで、各フェーズをテストして、IT 担当者が複雑な Java ES 配備をアップグレードするために使用するスクリプトを作成できます。

ステージング環境でアップグレードプロセスをテストし、アップグレードが正常に機能することを確認できれば、そのプロセスを本稼働環境で再現できます。

このプロセスには、次の表に示すフェーズが含まれます。これらのフェーズについては、この『アップグレードガイド』で説明しています。これらのフェーズは、個々のコンポーネントのアップグレードだけでなく、Java ES の配備全体にも適用されます。

表 1-4 アップグレードプロセスのフェーズ

アップグレードフェーズ	説明
計画	アップグレード計画を作成します。計画では、アップグレードする Java ES コンポーネントと、それらのコンポーネントを配備内のさまざまなコンピュータやオペレーティングシステムインスタンスでアップグレードする順序を指定します。
アップグレード前の準備	設定とアプリケーションデータをバックアップし、オペレーティングシステムにパッチを適用して、必要な依存関係をアップグレードし、個々のコンポーネントをアップグレードするのに必要なその他の準備作業を実行します。
アップグレード	アップグレードに必要なパッケージ、パッチ、およびツールをすべて入手します。アップグレードされたソフトウェアをインストールし、各コンポーネントを所定の方法で再設定します。これには、アップグレードされたシステムへのデータの移行も含まれます。

表 1-4　アップグレードプロセスのフェーズ (続き)

アップグレードフェーズ	説明
検証	所定の検証テストを使用してアップグレードが成功したことを確認します。これには、アップグレードされたソフトウェアコンポーネントの起動や、さまざまな使用方法を想定したテストも含まれます。
アップグレード後の手順	追加設定、カスタマイズ、またはアップグレードされたコンポーネントを運用するために必要になる可能性のあるその他の作業 (たとえば、新機能を組み込む) を実行します。
ロールバックと復元	アップグレードをロールバックし、ロールバックが成功していることを確認します。何らかの理由で本稼働環境を以前の状態に復元しなければならない場合もあるため、アップグレードのロールバックのテストは重要です。

アップグレード計画の考慮事項

アップグレード計画では、Release 5 にアップグレードする Java ES コンポーネントと、それらコンポーネントを Java ES 配備内のさまざまなコンピュータやオペレーティングシステムインスタンスでアップグレードする順序を指定します。

計画は、配備アーキテクチャーの範囲と複雑さだけでなく、アップグレードの目的や優先事項によっても異なります。

たとえば、1 台のコンピュータで実行される単一の Java ES コンポーネントで構成される Java ES 配備アーキテクチャーで、以前のソフトウェアリリースのバグを修正することがアップグレードの目的である場合があります。一方、相互に依存関係を持つ多数の Java ES コンポーネントが複数のコンピュータに配備された構成の Java ES 配備アーキテクチャーで、最小のダウンタイムで必要最小限のコンポーネントをアップグレードすることによって新しい機能を実現することがアップグレードの目的である場合もあります。

通常、Java ES コンポーネントの数と、配備アーキテクチャー内のコンピュータの数が多くなるほど、アップグレード計画は複雑になります。

しかし、アップグレード計画は配備アーキテクチャーの範囲と複雑さ以外のいくつかの考慮事項によって異なります。これらの考慮事項には、次の要素が含まれます。

- アップグレードの依存関係
- 選択的なアップグレードとすべてのアップグレード
- サポートされているアップグレードパスとアップグレード方針
- 複数インスタンスのアップグレード

- オペレーティングシステムの考慮事項

アップグレードの依存関係

いずれの Java ES 製品コンポーネントのアップグレード計画においても、主要な課題の 1 つとなるのは、1 つの製品コンポーネントについて別の Java ES コンポーネントに対する依存関係を理解し、依存するコンポーネントのアップグレードをサポートするためにほかのコンポーネントをアップグレードする必要があるかどうかを判断することです。

アップグレードの依存関係には次の 2 つのタイプがあります。

- **強いアップグレード依存関係** : 製品コンポーネントをアップグレードするには、依存先のコンポーネントをアップグレードする必要があります。この要件は、新しい機能、新しいインタフェース、または依存するコンポーネントに必要なバグの修正に起因する場合があります。強いアップグレード依存関係の場合は、先に依存先のコンポーネントをアップグレードしないと、アップグレードは成功せず、依存するコンポーネントを使用できません。
- **弱いアップグレード依存関係** : 製品コンポーネントをアップグレードするために、依存先のコンポーネントをアップグレードする必要はありません。弱いアップグレード依存関係の場合は、依存先のコンポーネントをアップグレードしなくても、コンポーネントのアップグレードは成功し、依存するコンポーネントを使用できます。

Java ES 製品コンポーネントのアップグレードでは、強いアップグレード依存関係の依存先となるすべてのコンポーネントをアップグレードする必要がありますが、(このマニュアルで示したいくつかの例外はあるものの) 弱いアップグレード依存関係の依存先となるコンポーネントをアップグレードする必要はありません。相互に依存関係を持つ複数のコンポーネントがアップグレードに含まれる場合、アップグレードする Java ES コンポーネントのうち 1 つでも、あるコンポーネントに対して強いアップグレード依存関係を持つ場合には、そのコンポーネントもアップグレードする必要があります。

いくつかの特殊な場合では、互換性が失われるため、コンポーネントをアップグレードするためにはそのコンポーネントがサポートするコンポーネントもアップグレードする必要があります。これらの特殊な場合については、このマニュアルで説明しています。

サポートされているアップグレードパスとアップグレード方針

アップグレード計画は、Release 5 にアップグレードする Java ES のリリースによって異なります。

以前のリリースの Java ES ソフトウェアはすべて Java ES 5 (Release 5) にアップグレードできますが、サポートされているのは Java ES 2005Q4 (Release 4)、Java ES 2005Q1 (Release 3)、および Java ES 2004Q2 (Release 2) からのアップグレードだけです。この『アップグレードガイド』では、Java ES 2003Q4 (Release 1) およびそれ以前の Java ES リリースからのアップグレードについて、その方針を示しますが、そのようなアップグレードを実行するための手順は示していません。

次の表に、Release 5 へのさまざまなアップグレードパス、それらの特性、およびアップグレードの実行時に使用するアップグレード方針を示します。

この表に示しているアップグレードパス間には違いがあり、製品コンポーネントのアップグレード手順はアップグレードするリリースによって異なることが多いため、この『アップグレードガイド』では、各製品コンポーネントのアップグレードについて説明する章が複数の節に分けられ、それぞれが異なるアップグレードパスを説明しています。

表 1-5 Java ES 5 (Release 5) へのアップグレードパス

製品バージョン	Java ES のリリース	システムの特性	アップグレード方針
2005Q4	Release 4	Java ES 5 (Release 5) では、1 台のコンピュータ上での Release 4 の製品コンポーネントと Release 5 の製品コンポーネントの共存がサポートされますが、共有コンポーネントを同じリリースに同期する必要があります。Release 4 と Release 5 の製品コンポーネント間の相互運用性はテスト済みであり、インタフェースに関する既知の非互換性については『Java Enterprise System 5 リリースノート (UNIX 版)』(http://docs.sun.com/doc/820-0453?l=ja) で説明されています。	Release 4 と Release 5 の製品コンポーネントが共存すると、1 台のコンピュータ上で、または複数のコンピュータで構成される配備アーキテクチャー内で、Release 4 の製品コンポーネントの Release 5 への選択的なアップグレードを行う可能性が生じます。 Release 5 の製品コンポーネントのいずれかが Release 5 の共有コンポーネントのサポートを必要とする場合は、コンピュータ上のすべての共有コンポーネントを Release 5 に同期してください。

表 1-5　Java ES 5 (Release 5) へのアップグレードパス (続き)

製品バージョン	Java ES のリリース	システムの特性	アップグレード方針
2005Q1	Release 3	上記の Release 4 のアップグレードパスと同様です。Java ES 5 (Release 5) では、1 台のコンピュータ上での Release 3 (および Release 4) の製品コンポーネントと Release 5 製品コンポーネントの共存がサポートされますが、共有コンポーネントを同じリリースに同期する必要があります。Release 3 と Release 5 のコンポーネント間の相互運用性はテスト済みであり、インタフェースに関する既知の非互換性については『Java Enterprise System 5 リリースノート (UNIX 版)』(http://docs.sun.com/doc/820-0453?l=ja) で説明されています。	上記の Release 4 のアップグレードパスと同様です。Release 3 と Release 5 のコンポーネントが共存すると、1 台のコンピュータ上で、または複数のコンピュータで構成される配備アーキテクチャー内で、Release 3 のコンポーネントの Release 5 への選択的なアップグレードを行う可能性が生じます。 Release 5 の製品コンポーネントのいずれかが Release 5 の共有コンポーネントのサポートを必要とする場合は、コンピュータ上のすべての共有コンポーネントを Release 5 に同期してください。
2004Q2	Release 2	上記の Release 4 や Release 3 のアップグレードパスとは異なります。Java ES 5 (Release 5) では、1 台のコンピュータ上での Release 2 のコンポーネントと Release 5 のコンポーネントの共存は、製品コンポーネントか共有コンポーネントかに関係なく、サポートされていません。これらのリリースバージョン間にはインタフェースに関する既知の非互換性があり、Release 2 と Release 5 のコンポーネント間の相互運用性は保証 (テスト) されていません。	1 台のコンピュータで Release 2 から Release 5 にコンポーネントをアップグレードする場合、Release 2 のすべてのコンポーネントを Release 5 にアップグレードしてください。ただし、場合によっては、配備アーキテクチャー内の異なるコンピュータ上で Release 2 と Release 5 のコンポーネントを共存させることができます。

表 1-5　Java ES 5 (Release 5) へのアップグレードパス (続き)

製品バージョン	Java ES のリリース	システムの特性	アップグレード方針
2003Q4 およびそれ以前のバージョン	Release 1 およびそれ以前の Java ES	上記の Release 2 のアップグレードパスと同様です。Java ES 5 (Release 5) では、1 台のコンピュータ上での Release 1 以前のリリースのコンポーネントと Release 5 のコンポーネントの共存は (製品コンポーネントか共有コンポーネントかに関係なく) サポートされていません。これらのリリースバージョン間にはインタフェースに関する既知の非互換性があり、Release 1 以前のリリースと Release 5 のコンポーネント間の相互運用性は保証 (テスト) されていません。	Java ES では、Release 1 以前のリリースから Release 5 への直接のアップグレードは動作保証されていません。 ただし、Release 3 の『Sun Java Enterprise System 2005Q1 アップグレードと移行』(http://docs.sun.com/source/819-2235?l=ja) に説明されているように、最初に Java ES Release 3 にアップグレードし、次に Release 3 から Release 5 へアップグレードすることで Release 1 からアップグレードできることもあります。その可能性については、この『アップグレードガイド』の該当するコンポーネントのアップグレードロードマップで説明します。 一方、Release 2 または Release 3 から Release 5 へのアップグレードと同じ方法で Release 1 から Release 5 へのアップグレードを実行できることがあります。この『アップグレードガイド』では、該当するコンポーネントのアップグレードロードマップでその可能性について説明しています。

注　公式な Java ES リリースの間に製品コンポーネントの暫定機能リリース (interim feature release、IFR) が公開された場合、IFR のアップグレードは、通常、先行する Java ES リリースと同じ手順を使用して実行されます。たとえば、Release 3 と Release 4 の間に IFR がある場合、Release 3 から Release 5 へのアップグレード手順を使用してコンポーネントをアップグレードします。これに当てはまらない場合 (たとえば、Portal Server や Portal Server Secure Remote Access の場合) については、この『アップグレードガイド』で IFR 固有のアップグレード手順について説明しています。

選択的なアップグレードとすべてのアップグレード

アップグレード計画では、強いアップグレード依存関係と弱いアップグレード依存関係を区別することにより、配備されたシステム内の Java ES 製品コンポーネントを選択的にアップグレードすることもできます。選択的なアップグレードは、1 台のコンピュータ上での Release 3 および Release 4 から Release 5 へのアップグレードに適用されます。1 台のコンピュータでの Release 2 から Release 5 への選択的なアップグレードはサポートされていません。

通常は、選択的なアップグレードを実行するか、コンピュータ上のすべての Java ES 製品コンポーネントをアップグレードするかを選択できます。

- **選択的なアップグレード**: この手順では、Release 5 にアップグレードする Java ES 製品コンポーネントを最初に選択します。そのコンポーネントと強いアップグレード依存関係があるコンポーネントを特定します。また、それらのコンポーネントはアップグレードする必要があります。このプロセスを強いアップグレード依存関係ごとに、アップグレードの必要があるコンポーネントがなくなるまで継続します。この作業によって、アップグレードが必要なすべての Java ES 製品コンポーネントが特定されます。
- **すべてのアップグレード**: この手順では、配備されているすべての Java ES 製品コンポーネントを Release 5 にアップグレードします。場合によっては、配備の複雑さが原因で、業務上の理由からシステム全体を一度にアップグレードできないこともあります。

アップグレードを実行するための 2 つの手順を、次の表で比較します。

表 1-6 選択的なアップグレードとすべてのアップグレードの比較

アップグレードの手順	利点	欠点
選択的なアップグレード	アップグレードするコンポーネントの数を最小限にとどめます。	配備されているシステムですべてのコンポーネントについてバージョンの一貫性がなくなります
すべてのアップグレード	配備されているシステムですべてのコンポーネントについてバージョンの一貫性を維持します。	アップグレードするコンポーネントの数が最大になります

選択的なアップグレードは、Java ES Release 4 でもサポートされていました。したがって、Release 3 の製品コンポーネントと Release 4 の製品コンポーネントの両方を 1 台のコンピュータ上に共存させることができ、両方を Release 5 に選択的にアップグレードできます。

複数インスタンスのアップグレード

アップグレード手順は、配備アーキテクチャーで冗長性がどの程度使用されているかによって異なります。Java ES コンポーネントの複数インスタンスを使用すると、可用性、スケーラビリティー、サービス性、あるいはこれらサービス品質の組み合わせを高めることができます。Java ES 配備アーキテクチャーで冗長性のあるコンポーネントを使用するテクノロジには、負荷分散 (Directory Proxy Server、Web Server、Web Proxy Server、Application Server、Access Manager、および Portal Server)、高可用性の手法 (Sun Cluster および High Availability Session Store)、および Directory Server レプリケーションの 3 種類があります。

冗長性が必要なほとんどの場合は、大きなダウンタイムを発生させずにアップグレードを実行してください。これらの順次アップグレードでは、コンポーネントが提供するサービスを停止させることなく、それらのコンポーネントの冗長なインスタンスを継続的にアップグレードすることが試みられます。

通常、冗長なインスタンスは複数のコンピュータに配備されます。アップグレード計画では、ダウンタイムを最小限に抑えるために、複製されたコンポーネントのアップグレードをその他のコンポーネントのアップグレードから分離しなければならない場合があります。順次アップグレードを実行する前に、各コンピュータで複製されたコンポーネントに対するアップグレード前の作業をすべて実行します。

各レプリケーションテクノロジには、Java ES コンポーネントアップグレードの順序全体に影響を与える可能性がある設定または再設定の手順が含まれます。たとえば、Sun Cluster 環境で実行するコンポーネントでは、Sun Cluster 環境で実行しているコンポーネントをアップグレードする前に、Sun Cluster のアップグレードを必要とすることがあります。

この『アップグレードガイド』では、各製品コンポーネントのアップグレードについて説明する章で、該当するコンポーネントに対する複数インスタンスのアップグレードの実行方法について説明します。

オペレーティングシステムの考慮事項

次に示すように、オペレーティングシステムに関するいくつかの考慮事項は、Java ES のアップグレード計画に影響を与えることがあります。

必須のオペレーティングシステムパッチ

Java ES 製品コンポーネントのアップグレードを成功させるためには、最初にオペレーティングシステムにパッチを適用するか、Java ES 5 の製品コンポーネントが必要とするレベルまでオペレーティングシステムを更新しなければならない場合があります。ただし、それぞれの場合に必要な特定のパッチや修正を適用するよりも、オペレーティングシステム全体を Java ES 5 が必要とするレベルにしてから、特定の製品コンポーネントをアップグレードすることをお勧めします。

- **Solaris プラットフォーム :** オペレーティングシステムのパッチは SunSolve Web サイトからパッチクラスタとして入手できます。パッチクラスタとは、まとめて適用できるオペレーティングシステムのパッチを集めたものです。Java ES Release 5 をサポートするのに必要な、Solaris 9 および 10 用のパッチクラスタは、http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access で入手できます。
- **Linux プラットフォーム :** 更新リリースは、https://www.redhat.com/apps/download/ で入手できます。ただし、Java ES のアップグレードを実行する前に Linux オペレーティングシステムを更新する必要はありません。

二重のアップグレード : Java ES ソフトウェアとオペレーティングシステムソフトウェア

Java ES ソフトウェアかオペレーティングシステムのどちらかをサポートされていないバージョンにアップグレードしようとした場合、オペレーティングシステムと Java ES ソフトウェアが整合しなくなる可能性があります。関係するサポートマトリックスを次の表に示します。

表 1-7 Java ES/ オペレーティングシステムサポートマトリックス

Java ES のリリース	Solaris			RHEL		
	8	9	10	2.1	3.0	4.0
2003Q4 (Release 1)	X	X		X		
2004Q2 (Release 2)	X	X		X		
2005Q1 (Release 3)	X	X	X	X	X	
2005Q5 (Release 4)	X	X	X	X	X	
5 (Release 5)		X	X		X	X

Java ES ソフトウェアまたはオペレーティングシステムソフトウェアのどちらか片方をアップグレードすることで、サポートされていない環境になってしまう場合、二重にアップグレードを実行する必要があります。この二重のアップグレードとは、Java ES とオペレーティングシステムの両方をアップグレードすることです。次の状況では、二重のアップグレードが必要になる可能性があります。

- インストールした Java ES ソフトウェアでサポートされていないバージョンにオペレーティングシステムをアップグレードした場合。

  たとえば、Java ES 2004Q2 (Release 2) は、Solaris 8 および 9 のオペレーティングシステムと Red Hat Enterprise Linux (RHEL) 2.1 でサポートされています。オペレーティングシステムのプラットフォームを Java ES Release 2 でサポートされていない Solaris 10 または RHEL 3.0 にアップグレードする場合、Java ES Release 2 を、アップグレードされたプラットフォームをサポートする Java ES リリースバージョンにアップグレードする必要もあります。この場合、Java ES 5 (Release 5) にアップグレードすることをお勧めします。

- 既存のオペレーティングシステムソフトウェアでサポートされていないバージョンに Java ES をアップグレードした場合。

  たとえば、Java ES 2005Q1 (Release 3) と Java ES 2005Q4 (Release 4) は、Solaris 8 および RHEL 2.1 でサポートされています。ただし、Java ES を Solaris 8 や RHEL 2.1 でサポートされていない Release 5 にアップグレードする場合は、オペレーティングシステムを、Java ES 5 (Release 5) でサポートされているバージョンにアップグレードしてください。この場合、Solaris 10 または RHEL 4.0 にアップグレードすることをお勧めします。

一般的に、二重のアップグレードは次の 2 とおりの方法で実行できます。

- **新規オペレーティングシステムのインストール** : 新しいオペレーティングシステムをインストールした後、Java ES Release 5 の新規インストールを実行します。これには、旧バージョンの製品コンポーネントデータ (設定データ、実行時データ、カスタマイズなど) の移行も含まれます。新しいシステム (または Solaris 10 ゾーン) でオペレーティングシステムのインストールを実行するか、または、既存のファイルシステムを消去することができます。後者の場合、コンポーネントデータを最初にバックアップしておき、オペレーティングシステムのインストール後にそれを復元する必要があります。

- **オペレーティングシステムのインプレースアップグレード** : 既存のファイルシステムをそのまま同じ場所に (インプレースで) 残したまま、オペレーティングシステムのアップグレードを実行したあと、Java ES 製品コンポーネントを Release 5 にアップグレードします。これを正常に実行するには、オペレーティングシステムのアップグレードがインストール済みの Java ES 製品コンポーネントとそのデータ、および必要な共有コンポーネントのアップグレードに影響を与えないようにする必要があります。

二重のアップグレードが Java ES 製品コンポーネントでサポートされていない場合、つまり、これらのどちらの方法も使用できない場合、オペレーティングシステムのインストールまたはアップグレードを実行した後、そのコンポーネントを再インストールし新しく設定する必要があります。

次の表は Java ES 製品コンポーネントがサポートしている二重のアップグレードの手順を示します。

表 1-8 Java ES 5 製品コンポーネントでの二重アップグレードのサポート

製品コンポーネント	新規オペレーティングシステムのインストール	オペレーティングシステムのインプレースアップグレード
Access Manager	非サポート	非サポート
Application Server	同じコンピュータでのみサポート	サポート
Directory Proxy Server	サポート	サポート
Directory Server	サポート	サポート
High Availability Session Store	Application Server の二重アップグレードの処理の中で実行	Application Server の二重アップグレードの処理の中で実行
Java DB	サポート	サポート
Message Queue	サポート	サポート
Portal Server	非サポート	サポート
Portal Server Secure Remote Access	非サポート	サポート
Service Registry	サポート	サポート
Sun Cluster	非サポート	サポート
Sun Cluster Geographic Edition	非サポート	Sun Cluster の二重アップグレードの処理の中で実行
Web Proxy Server	非サポート	サポート
Web Server	非サポート	サポート

オペレーティングシステムのアップグレード

Solaris オペレーティングシステムのアップグレードによって、既存の Java ES 共有コンポーネントが以前のバージョンで上書きされる場合があります。このような場合は、Solaris オペレーティングシステムに付属する Message Queue を Release 5 にアップグレードすることで、正しい Java ES バージョンを復元できます。Message Queue をアップグレードすると、存在するすべての共有コンポーネントも強制的にアップグレードされます。

Solaris 10 の複数ゾーン環境

複数ゾーン環境内の Java ES コンポーネントのインストールとアップグレードには、いくつかの問題があります。Solaris 10 ゾーンに Java ES を配備する際のメリットと制限事項、および複数ゾーン環境内の Java ES コンポーネントのアップグレードに関する推奨方法については、62 ページの「Java ES 5 のアップグレードと Solaris 10 ゾーン」を参照してください。

Java ES コンポーネントの依存関係

アップグレード計画での最重要検討項目の 1 つに、配備されたシステム内の各種 Java ES コンポーネントの間の依存関係があります。コンポーネントのアップグレードを実行する順序は、コンポーネント間の依存関係の性質によって影響を受けます。

この節では、アップグレード計画に影響を与える Java ES コンポーネントの依存関係について説明します。

- 共有コンポーネントに対する依存関係
- 製品コンポーネントに対する依存関係

共有コンポーネントに対する依存関係

48 ページの表 1-9 は、Java ES 5 (Release 5) 製品コンポーネントの Java ES 共有コンポーネントに対する依存関係を示しています。表 1-9 の列見出しに示されている製品コンポーネントの略称は、28 ページの表 1-1 に記載されているものです。共有コンポーネントの略称は、29 ページの表 1-2 に記載されています。

表 1-9 のマトリックスでは、Release 3 および Release 4 から Release 5 へのアップグレードについて、強いアップグレード依存関係は「H」で示され、弱いアップグレード依存関係は「S」で示されています。Release 2 から Release 5 へのアップグレードについては、すべての共有コンポーネントの依存関係は、当然ながら強いアップグレード依存関係です。つまり、すべての共有コンポーネントを Release 2 から Release 5 にアップグレードしてください。

表 1-9　Java ES 5 (Release 5) の製品コンポーネントの共有コンポーネントに対する依存関係

共有コンポーネント	AM	AS	DPS	DS	DS コンソール	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			
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

表 1-9　Java ES 5 (Release 5) の製品コンポーネントの共有コンポーネントに対する依存関係 (続き)

共有コンポーネント	AM	AS	DPS	DS	DS コンソール	HADB	JavaDB	MQ	MC	PS	PSRA	SC	SCG	SR	WPS	WS
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 コア	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. これは特に共通エージェントコンテナ (CAC) Version 1.1 に対する依存関係です。

表 1-9 に示したすべての製品コンポーネントの依存関係は、共有コンポーネントとの直接的な依存関係と間接的な依存関係の両方を表しています。ある製品コンポーネントが特定の共有コンポーネントに依存し (直接的な依存関係)、同様にその共有コンポーネントがほかの 1 つまたは複数の共有コンポーネントに依存する (間接的な依存関係) こともあります。51 ページの図 1-1 に、共有コンポーネント間の相互依存関係を示します。

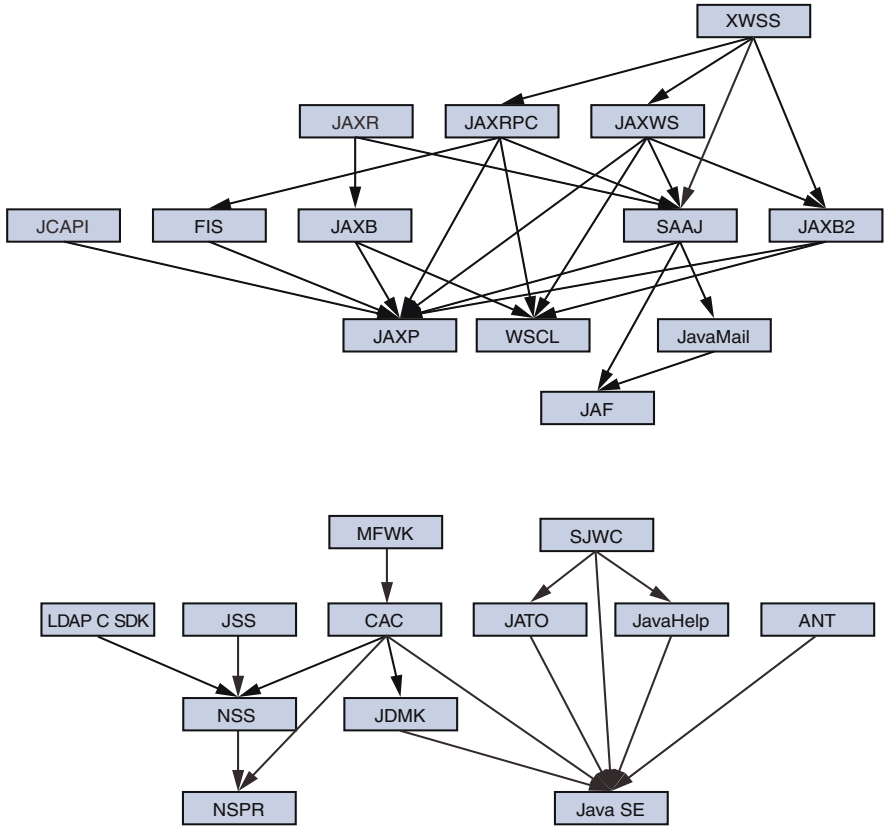
表 1-9 は、1 台のコンピュータ上で 1 つまたは複数の製品コンポーネントをアップグレードするときに、どの共有コンポーネントをアップグレードする必要があるかを示しています。

ただし、共有コンポーネントを同期させる必要があるため (35 ページの「共有コンポーネントのアップグレード」を参照)、Java ES 共有コンポーネントを 1 つずつアップグレードすることはできません。1 台のコンピュータ上、または 1 つのオペレーティングシステムインスタンス内のすべての共有コンポーネントを同時に Release 5 バージョンにアップグレードしてください。

強いアップグレード依存関係が含まれない場合、共有コンポーネントをアップグレードする必要はありません。ただし、配下の Java ES 共有コンポーネントベースを最新のバージョンにアップグレードすることをお勧めします。実際、Java ES インストーラによって製品コンポーネントをインストールまたはアップグレードした場合は、ホストコンピュータ上にあるすべての共有コンポーネントが自動的に Release 5 に同期します。

共有コンポーネントを手動でアップグレードする方法については、第 2 章「Java ES 共有コンポーネントのアップグレード」を参照してください。

図 1-1　共有コンポーネントの相互依存関係



製品コンポーネントに対する依存関係

製品コンポーネントに対する依存関係は、大きく分けて、実行時の依存関係と設定の依存関係の 2 つのカテゴリに分類されます。

- **実行時の依存関係** : ソフトウェアシステムの機能は、配備されているコンポーネント間の連携をベースとします。Java ES 製品コンポーネント間のインフラストラクチャーの依存関係については、『Sun Java Enterprise System 5 技術の概要』で説明されています。Release 5 の製品コンポーネントが別の製品コンポーネントに対して強いアップグレード依存関係を持つ場合は、依存先のコンポーネントもアップグレードされたときにのみ、依存するコンポーネントは正常にアップグレードされて意図したとおりに使用できます。
- **設定の依存関係** : 場合によっては、別のコンポーネントを設定するために、Java ES コンポーネントのインストール、設定、および実行が必要です。たとえば、Access Manager サービスを登録するには、Directory Server のユーザーディレクトリまたはグループディレクトリを実行している必要があります。通常、コンポーネントのアップグレード手順には、アップグレードされたコンポーネントの再設定や設定データの移行が関係しています。設定の依存関係は、アップグレード手順に影響を与えることがあります。

実行時の依存関係については、製品コンポーネント間の関係は次の 3 種類です。

- **必須** : サポートするコンポーネントがないとコンポーネントは動作しません。
- **省略可能** : サポートするコンポーネントがなくてもコンポーネントは動作しますが、コンポーネントの機能のサブセットにはサポートするコンポーネントが必要です。
- **共依存関係** : どちらのコンポーネントももう一方のサポートなしで動作しますが、両方のコンポーネントを使用することで特定の機能やパフォーマンスが向上します。

次の表は、28 ページの表 1-1 に示される Java ES 製品コンポーネント間の依存関係を示します。これらの情報を使用して、アップグレード計画に影響を与える強いアップグレード依存関係を特定できます。

1 列目は Release 5 の製品コンポーネント (アルファベット順)、2 列目は Release 5 コンポーネントが依存関係を持つほかの Java ES コンポーネント、3 列目は Release 5 の依存関係をサポートする Java ES のリリースバージョン、4 列目は依存関係の性質、5 列目は、サポートするコンポーネントを (リモートではなく) ローカルにインストールする必要があるか、ほかのサードパーティー製品が依存関係をサポートするか、など依存関係の特性を示します。

Release 5 にアップグレードする製品コンポーネントが (以前のリリースではなく) Release 5 のサポートするコンポーネントに対して依存関係を持つ場合は、サポートするコンポーネントは強いアップグレード依存関係を表すため、サポートするコンポーネントも Release 5 にアップグレードしてください。

表 1-10　Java ES 製品コンポーネントの依存関係

Release 5 製品コンポーネント	依存関係[1]	Java ES のリリース	依存関係の性質	特性
Access Manager	Directory Server	2-5	必須 : 設定データを保存し、ユーザーデータの検索を可能にします	
	J2EE Web コンテナ :		必須 : Web コンテナの実行時サービスを提供します	ローカルのみ
	- Application Server	4-5		ほかのサポート製品 : - Weblogic[2] - WebSphere[3]
	- Web Server	4-5		
Access Manager SDK	Access Manager	3-5	必須 : Access Manager サービスを提供します	
Access Manager 分散認証	Access Manager	4-5	必須 : Access Manager サービスを提供します	
	J2EE Web コンテナ :		必須 : Web コンテナの実行時サービスを提供します	ローカルのみ
	- Application Server	4-5		ほかのサポート製品 : - Weblogic[2] - WebSphere[3]
	- Web Server	4-5		
Access Manager セッションフェイルオーバー	Access Manager	5	必須 : Access Manager サービスを提供します	
	Message Queue	4-5	必須 : 信頼できる非同期メッセージングを提供します	

表 1-10　Java ES 製品コンポーネントの依存関係 (続き)

Release 5 製品コンポーネント	依存関係[1]	Java ES のリリース	依存関係の性質	特性
Application Server	Message Queue	3-5	必須 : 信頼できる非同期メッセージングを提供します	ローカルのみ
	High Availability Session Store (HADB)	5	必須 : インスタンス間のフェイルオーバーをサポートするために必要なセッション状態を保存します	ローカルのみ
	Java DB	5	必須 : デフォルトの開発者データベースおよびその他の持続的記憶領域を提供します。	ローカルのみ
	Web Server	3-5	省略可能 : インスタンス間の負荷分散を提供します	ローカルのみ
Directory Proxy Server	Directory Server	1-5	共依存関係 : ディレクトリ要求に関するセキュリティーとパフォーマンスを向上させます。Directory Proxy Server にデータを供給します	
Directory Server	Directory Proxy Server	1-5	共依存関係 : ディレクトリ要求に関するセキュリティーとパフォーマンスを向上させます。負荷を分散し、Directory Server からのデータをキャッシュします	
High Availability Session Store (HADB)	なし			
Java DB	なし			

表 1-10　Java ES 製品コンポーネントの依存関係 (続き)

Release 5 製品コンポーネント	依存関係[1]	Java ES のリリース	依存関係の性質	特性
Message Queue	Directory Server	2-5	省略可能 : 管理対象オブジェクトとユーザーデータを格納します	
	J2EE Web コンテナ :		省略可能 : クライアントと Message Queue ブローカ間の HTTP トランスポートをサポートします	
	- Application Server	2-5		
	- Web Server	2-5		
	Java DB	5	省略可能 : 永続メッセージを保存します。	ローカルのみ
	Sun Cluster	2-5	省略可能 : 高可用性をサポートします	
Monitoring Console	なし			

表 1-10　Java ES 製品コンポーネントの依存関係 (続き)

Release 5 製品コンポーネント	依存関係[1]	Java ES のリリース	依存関係の性質	特性
Portal Server	Directory Server	4-5	必須 : ユーザープロファイルを保存し、それらの検索を可能にします	
	J2EE Web コンテナ :		必須 : Web コンテナの実行時サービスを提供します	ローカルのみ
	- Application Server	4-5		
	- Web Server	4-5		
	Access Manager または Access Manager SDK	4-5	必須 : 認証サービス、承認サービス、およびシングルサインオンを提供します	ローカルのみ (Access Manager がリモートの場合は、Access Manager SDK をローカルで使用する)
	Portal Server Secure Remote Access	5	省略可能 : ゲートウェイ、Rewriter プロキシ、および Netlet プロキシのコンポーネントによるセキュリティー保護されたリモートアクセスを提供します	
	Service Registry クライアント	5	必須 : コンパイルに必要なライブラリを提供します	
	Java DB	5	必須 : いくつかのポートレットアプリケーションにサポートを提供します	
Portal Server Secure Remote Access ゲートウェイ	Portal Server	5	必須 : ゲートウェイ機能をサポートします	
	Access Manager または Access Manager SDK	4-5	必須 : 認証サービス、承認サービス、およびシングルサインオンを提供します	ローカルのみ (Access Manager がリモートの場合は、Access Manager SDK をローカルで使用する)
	Directory Server	4-5	必須 : ユーザーデータを保存し、それらの検索を可能にします	

表 1-10　Java ES 製品コンポーネントの依存関係 (続き)

Release 5 製品コンポーネント	依存関係[1]	Java ES のリリース	依存関係の性質	特性
Rewriter プロキシ	Portal Server	5	必須 : Rewriter プロキシ機能をサポートします	
Netlet プロキシ	Portal Server	5	必須 : Netlet プロキシ機能をサポートします	
Service Registry 配備	Application Server	5	必須 : コンテナの実行時サービスを提供します	ローカルのみ
	Java DB	5	必須 : サービスおよび関連するメタデータを格納するためのデフォルトデータベースを提供します	ローカルのみ
	Service Registry クライアント	5	必須 : 必要なクライアントライブラリを提供します	ローカルのみ
クライアント	なし			
Sun Cluster	なし			
Sun Cluster エージェント	Sun Cluster	4-5	必須 : Sun Cluster サービスへのアクセスを提供します	ローカルのみ
Sun Cluster Geographic Edition	Sun Cluster	4-5	必須 : Sun Cluster Geographic Edition の機能をサポートします。	ローカルのみ
Web Proxy Server	Directory Server	2-5	省略可能 : LDAP ベースの認証を提供します	
	Web Server	2-5	共依存関係 : HTTP 要求に関するセキュリティーとパフォーマンスを向上させます。Web Proxy Server にデータを供給します	ほかのサポート製品 : - Weblogic[2] - WebSphere[3]

表 1-10　Java ES 製品コンポーネントの依存関係 (続き)

Release 5 製品コンポーネント	依存関係[1]	Java ES のリリース	依存関係の性質	特性
Web Server	Directory Server	1-5	省略可能 : LDAP ベースの認証を提供します	
	Web Proxy Server	1-5	共依存関係 : HTTP 要求に関するセキュリティーとパフォーマンスを向上させます。負荷を分散し、Web Server からのデータをキャッシュします	

1. 各製品コンポーネントについて、通常のアップグレード順序に従って依存関係を示します。
2. BEA Weblogic Server
3. IBM WebSphere Application Server

アップグレード手順に関するガイドライン

選択的なアップグレードまたはすべてのアップグレードの選択、強いアップグレード依存関係、およびこれまでの節で説明したその他の要素は、いずれも、どの Java ES コンポーネントをアップグレードするかだけでなく、どのような順序でアップグレードする必要があるかに影響します。とはいえ、すべての場合には適用できませんが、手順に関する一般的なガイドラインがいくつか存在します。

1 台のコンピュータまたは 1 つの配備システムで Java ES コンポーネントを正常にアップグレードするための手順を次に示します。アップグレードを計画するときは、配備アーキテクチャーに含まれていないコンポーネントを省略できます。また、選択的なアップグレードを実行する場合は、アップグレード依存関係の弱いコンポーネントを省略できます。

この『アップグレードガイド』の各章は、次に示すコンポーネントの順序に従って構成されています。

注　Java ES コンポーネントをアップグレードする前に、オペレーティングシステムの必要な更新すべてを必ず適用してください (44 ページの「必須のオペレーティングシステムパッチ」を参照)。

また、61 ページの「特殊な場合」で、アップグレードのシナリオに適用される事項がないか確認してください。

1. **共有コンポーネント** (67 ページの第 2 章「Java ES 共有コンポーネントのアップグレード」を参照)

共有コンポーネントに依存するコンポーネントよりも前に、共有コンポーネントをアップグレードするようにしてください。ほとんどの場合、共有コンポーネントのアップグレードは Java ES インストーラによって処理されますが、Web Proxy Server および Portal Server の場合は、共有コンポーネントを明示的にアップグレードしてください。

2. **Sun Cluster ソフトウェア** (79 ページの第 3 章「Sun Cluster ソフトウェア」を参照)

   Sun Cluster 環境で実行されるコンポーネントがあり、Sun Cluster ソフトウェアをアップグレードする必要がある場合、Sun Cluster サービスを使用するコンポーネントよりも前にアップグレードしてください。Sun Cluster エージェントをアップグレードする場合、Sun Cluster のアップグレードの一部としてアップグレードしてください。

3. **Sun Cluster Geographic Edition ソフトウェア** (91 ページの第 4 章「Sun Cluster Geographic Edition」を参照)

   Sun Cluster Geographic Edition は、依存先の Sun Cluster ソフトウェアのあとにアップグレードするようにしてください。また、Sun Cluster サービスを使用するコンポーネントよりも前にアップグレードするようにしてください。

4. **Directory Server** (103 ページの第 5 章「Directory Server」を参照)

   多くのコンポーネントが Directory Server にユーザーデータや設定データを保存するため、通常は Directory Server に対して実行時の依存関係または設定の依存関係を持つコンポーネントをアップグレードする前に、Directory Server のアップグレードを実行してください。

5. **Directory Proxy Server** (119 ページの第 6 章「Directory Proxy Server」を参照)

   Directory Proxy Server は、Directory Server に対して弱いアップグレード依存関係を持っていて、いつでもアップグレードできます。ただし、一部のコンポーネントは Directory Proxy Server 経由で Directory Server にアクセスする可能性があるので、Directory Proxy Server をアップグレードする場合は、Directory Server の直後にアップグレードするようにしてください。

6. **Web Server** (133 ページの第 7 章「Web Server」を参照)

   多数の Java ES コンポーネントが Web コンテナのサポートを必要とするため、Web コンテナをアップグレードする場合、Web コンテナのサービスを必要とするコンポーネントよりも先に Web コンテナをアップグレードしてください。通常、Web コンテナのサービスは Web Server または Application Server によって提供されますが、その両方がアーキテクチャーに含まれる場合は、Application Server をアップグレードする前にまず Web Server をアップグレードしてください。

7. **Java DB** (157 ページの第 8 章「Java DB」を参照)

Java DB は、デフォルトデータベースとして Java DB を必要とする Application Server よりも前にアップグレードしてください。ただし、Java DB は Application Server をアップグレードするときに Java ES インストーラによって自動的にアップグレードされます。

8. **High Availability Session Store** (167 ページの第 9 章「High Availability Session Store」を参照)

   High Availability Session Store (HADB) は、高可用性のために High Availability Session Store を必要とする Application Server よりも前にアップグレードしてください。ただし、HADB は Application Server をアップグレードするときに Java ES インストーラによって自動的にアップグレードされます。

9. **Message Queue** (179 ページの第 10 章「Message Queue」を参照)

   Message Queue は、Java Enterprise Edition (Java EE) に準拠した Message Queue を必要とする Application Server よりも前にアップグレードしてください。ただし、Message Queue は Application Server をアップグレードするときに Java ES インストーラによって自動的にアップグレードされます。

10. **Application Server** (203 ページの第 11 章「Application Server」を参照)

    Application Server は、Message Queue と High Availability Session Store に依存しているため、アップグレードする場合はこれらのコンポーネントよりもあとにアップグレードするようにしてください。Application Server の負荷分散プラグインは Web Server に依存することもあるため、負荷分散の機能を使用する場合は Web Server のあとに Application Server をアップグレードするようにしてください。

11. **Service Registry** (229 ページの第 12 章「Service Registry」を参照)

    Service Registry は、Application Server をアップグレードしたあとであれば、いつでもアップグレードできます。これは、Service Registry が実行時のコンテナサービスを Application Server に依存するためです。

12. **Web Proxy Server** (**Web Proxy Server** を参照)

    Web Proxy Server はどの時点でもアップグレードできますが、通常はプロキシサービスの提供先である Web Server コンポーネントまたは Application Server コンポーネントのあとにアップグレードします。Web Proxy Server は新規の Java ES Release 5 コンポーネントですが、Java ES 以外の以前のリリースからアップグレードできます。

13. **Access Manager** (257 ページの第 14 章「Access Manager」を参照)

    Access Manager は、シングルサインオンを含み、認証および承認で中心的な役割を担うため、これらのサービスについて依存するコンポーネントよりも前にアップグレードしてください。

14. **Portal Server** (305 ページの第 15 章「Portal Server」を参照)

Portal Server は、先行するコンポーネント (Directory Server、Web コンテナ、および Access Manager) の多くに依存するため、アップグレードする場合はこれらのコンポーネントのあとにアップグレードするようにしてください。

15. **Portal Server Secure Remote Access** (371 ページの第 16 章「Portal Server Secure Remote Access」を参照)

    Portal Server Secure Remote Access は、Portal Server をアップグレードするときにアップグレードしてください。

特殊な場合

Java ES コンポーネントの Release 5 へのアップグレードを計画するときに認識しておくべき特殊な場合がいくつかあります。これらの場合について、次に説明します。

選択的なアップグレード : Application Server をアップグレードしない場合

Release 3 または Release 4 の Application Server (8.1) を実行中のコンピュータ上で、いずれかの Java ES コンポーネントの Java ES 5 への選択的なアップグレードを実行し、Application Server を Release 5 にアップグレードしない場合は、Application Server を引き続き正常に機能させるために、次のような状況を解決してください。

- **JSP のコンパイルエラー** : 選択的なアップグレードを実行する前に、次の表に示す Application Server のパッチをまず適用するようにしてください。

表 1-11 Application Server を Release 5 にアップグレードしない場合に必要なパッチ[1]

説明	パッチ ID: Solaris 9 および 10	パッチ ID: Linux
Application Server の Release 3 および Release 4 に対する修正	119166-17 (SPARC) 119166-17 (x86)	119168-17

1. パッチのリビジョン番号は最低要件です。表に記載されているものよりも新しいリビジョンのパッチが使用可能になった場合は、新しいほうのパッチを使用してください。

パッチの適用に失敗すると、Application Server で JSP のコンパイルエラーが発生します。問題を解決するために、表 1-11 のパッチを前に遡って適用することもできます。

- **Linux 上の ANT 共有コンポーネントバイナリのリロケーション**: Release 5 の ANT は、以前のバージョンとは異なるパスにあります。
  AppServer8-base/config/asenv.conf ファイルには、ANT をポイントする Application Server 環境変数が次のように指定されています。

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

  これを、次のように変更してください。

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

  変更後、Application Server を再起動してください。

Portal Server の IFR 7.0 の Java ES 5 へのアップグレード

Web Server 環境の Portal Server を 暫定機能リリース (Interim Feature Releases、IFR) 7.0 2005Q4 から Release 5 へアップグレードする場合は、58 ページの「アップグレード手順に関するガイドライン」のガイドラインの例外について、352 ページの「Portal Server の IFR 7.0 からのアップグレード」を参照してください。

Java ES 5 のアップグレードと Solaris 10 ゾーン

この節では、Solaris 10 ゾーンでの Java ES ソフトウェアのアップグレードに伴う問題を扱い、そのような環境での推奨事項を示します。ここではまた、『Sun Java Enterprise System 5 インストール計画ガイド』(http://docs.sun.com/doc/820-0879?l=ja) の Java ES 5 と Solaris 10 ゾーンに関する情報を補足します。

この節で説明する内容は、次のとおりです。

- Java ES インストーラのゾーンサポート
- 推奨されるアップグレード方法
- 特殊な場合または例外

Java ES インストーラのゾーンサポート

Java ES 5 インストーラは、Java ES 製品コンポーネントのアップグレード (およびインストール) と共有コンポーネントの同期に関して、ゾーンのためのサポートを提供します。インストーラには、問題のあるアップグレードシナリオを避けるためのポリシーが実装されています。

製品コンポーネントのアップグレード

32 ページの「Java ES インストーラのアップグレード機能の使用」で説明されているように、Java ES インストーラを使用すると、限定された数の製品コンポーネントとそれらに対応する共有コンポーネントをアップグレードできます。このアップグレード機能は、大域ゾーンとすべての非大域ゾーンに適用されます。

ただし、この動作にはゾーンに関して次の 3 つの例外があります。

- 疎ルートゾーンでは、一部の共有コンポーネントが読み取り専用ディレクトリにあるため、それらをインストールまたはアップグレードできません。このような場合は、該当する共有コンポーネントが大域ゾーンでインストールまたはアップグレードされるまで、製品コンポーネントのアップグレードは停止します。インストーラは、次のメッセージを表示します。「次の共有コンポーネントは、選択したコンポーネントに必要ですが、疎ルートゾーンでインストールまたはアップグレードできません。続行する前に、これらの共有コンポーネントを大域ゾーンでインストールまたはアップグレードしてください。「すべての共有コンポーネント」オプションを使用してください。」
- Application Server と Message Queue はいずれも Solaris オペレーティングシステムに付属しています。これらのバージョンは、いずれも疎ルートゾーンでは直接アップグレードできません。これら 2 つの付属コンポーネントの詳細については、65 ページの「製品コンポーネントの特殊な場合」を参照してください。
- 大域ゾーンでは、非大域ゾーンが存在する場合に、現在インストールされているすべての共有コンポーネントをアップグレードして、選択したコンポーネントに必要な不足している共有コンポーネントをインストールする代わりに、インストーラは すべての Java ES 共有コンポーネントを (それらが特定の製品コンポーネントに必要かどうかに関係なく) Release 5 に同期します。これにより、Release 5 のすべての共有コンポーネントは非大域ゾーンに伝播され、非大域ゾーンでは共有コンポーネントの複数のバージョンが混在しないことが保証されます。

注 非大域ゾーンでは、製品コンポーネントのインストールやアップグレードに支障を来たす可能性のある特殊な場合や例外がいくつかあります。これらについては、65 ページの「特殊な場合または例外」で説明します。

すべての共有コンポーネントの同期

Release 5 では、すべての共有コンポーネントを対応する Release 5 バージョンに同期しなければならない状況に対応するため、共有コンポーネントの同期オプションが提供されています。「すべての共有コンポーネント」オプションを選択すると、インストーラは、特定の製品コンポーネントに必要かどうかに関係なく、現在インストールされているすべての共有コンポーネントをアップグレードし、不足している共有コンポーネントがあればインストールします。このオプションは、大域ゾーンと完全ルートゾーンに適用されますが、疎ルートゾーンには適用されません。

「すべての共有コンポーネント」オプション (詳細については 70 ページの「すべての共有コンポーネントの同期」を参照) は、次に示す 2 つのゾーンベースのアップグレードシナリオで必要です。

- **製品コンポーネントの手動アップグレード** : 「すべての共有コンポーネント」オプションは、共有コンポーネントのインストールを実行するために必要です。また、Java ES インストーラを使用してアップグレードすることができない製品コンポーネントをアップグレードするときに必要となる、共有コンポーネントのアップグレードを実行するためにも必要です。
- **疎ルートゾーンでのアップグレード** : 一部の共有コンポーネントは、デフォルトの疎ルートゾーンでは、インストールやアップグレードができません。このため、Java ES インストーラを使用して疎ルートゾーンの製品コンポーネントをアップグレードする場合、関係する共有コンポーネントによっては、最初に大域ゾーンの共有コンポーネントを同期しなければならないことがあります。このような場合は、大域ゾーンで「すべての共有コンポーネント」オプションを使用して、必要な共有コンポーネントのインストールおよびアップグレードを実行します。

Java ES インストーラの共有コンポーネントに関するゾーン動作の概要については、『Sun Java Enterprise System 5 インストール計画ガイド』(http://docs.sun.com/doc/820-0879?l=ja) の Java ES 5 および Solaris 10 ゾーンに関する情報を参照してください。

推奨されるアップグレード方法

アップグレード計画を作成するときは、『Sun Java Enterprise System 5 インストール計画ガイド』(http://docs.sun.com/doc/820-0879?l=ja) で概説されているゾーンのインストールと管理の方針に留意しながら、Java ES ソフトウェアの既存の複数ゾーン配備を調査するようにしてください。場合によっては、次の推奨方法を実装するために、1 つまたは複数のゾーンでコンポーネントをアンインストールし、それらをほかのゾーンに再インストールしなければならないこともあります。

- 方針の混在を避けるため、特に次のことを推奨します。
  - Java ES ゾーンの配備と管理の方針をできるだけ単純にします。同じコンピュータ上で Java ES コンポーネントの完全ルートの配備と疎ルートの配備を混在させないでください。疎ルートゾーンの配備をサポートするのに必要な手順や方法によって、完全ルートゾーンの配備に支障が出る可能性があります。
  - 大域ゾーンと非大域ゾーンの両方に、同じ Java ES 製品コンポーネントを (異なるバージョンであっても) インストールしないでください。大域ゾーンのインストールをアップグレードするのに必要な手順によって、非大域ゾーンのインストールを破損させる可能性があります。
  - Release 4 (またはそれ以前) の Java ES コンポーネントが完全ルートゾーンにインストールされている場合は、大域ゾーンで Java ES コンポーネントを Release 5 にアップグレードしないでください。大域ゾーンでアップグレードを行うと、完全ルートゾーンで Release 4 と Release 5 のファイルの混在を招くことがあります。
- 次のアップグレード方法を推奨します。
  - インストールされている Release 4 のすべての製品コンポーネントを Release 5 にアップグレードする場合は、大域ゾーンにあるすべての Java ES 共有コンポーネントを同期してから、必要な製品コンポーネントのアップグレードをそれらのコンポーネントがインストールされているゾーンで実行します。Release 5 の共有コンポーネントは下位互換性があります。
  - Release 4 または Release 5 の製品コンポーネントがゾーン以外の環境にインストールされており、その環境に非大域ゾーンを追加して、新しい非大域ゾーンに製品コンポーネントをインストールする場合は、大域ゾーンのコンポーネントをアンインストールして、それらを非大域ゾーンに再インストールしなければならないこともあります。

特殊な場合または例外

特殊な場合はいくつかありますが、その一部は、Solaris 10 に一部の Java ES 共有コンポーネントと一部の Java ES 製品コンポーネントが付属していることが原因です。このため、これらの Java ES コンポーネントは自動的に大域ゾーンに存在し、その結果、大域ゾーンから作成されたすべての非大域ゾーンにも存在します。

製品コンポーネントの特殊な場合

- Message Queue: Message Queue は、Solaris 10 に付属するため、最初に大域ゾーンから Message Queue を削除していないかぎり、非大域ゾーンが作成されたときに自動的に伝播されます。Message Queue は、疎ルートゾーンではインストールまたはアップグレードできません。Message Queue は、ほかの製品コンポーネントと異なり、Java ES インストーラを使用して大域ゾーンでインストールまたはアップグレードしたときに、デフォルトで非大域ゾーンに伝播されます。

- Application Server: Application Server は、Solaris 10 に付属するため、最初に大域ゾーンから Application Server を削除していないかぎり、非大域ゾーンが作成されたときに自動的に伝播されます。/usr にインストールされる付属の Application Server は、このようにして伝播された場合、疎ルートゾーン (デフォルトで、/usr は読み取り専用) では Java ES インストーラを使用してアップグレードできません。この問題を解決するには、疎ルートゾーンに Release 5 の Application Server をインストールする前に、付属の Application Server パッケージを大域ゾーンから手動で削除してください。219 ページの「Solaris のみ : オペレーティングシステムに付属している Application Server パッケージを手動で削除します。」を参照してください。
- Sun Cluster: Sun Cluster ソフトウェアは、非大域ゾーンではサポートされていません。

共有コンポーネントの特殊な場合

- Sun Java Web Console (SJWC):Solaris 10 (Update 1 および Update 2) に付属する SJWC パッケージは、Java ES インストーラでは削除できません。これらの古い SJWC パッケージは、SUNW_PKG_ALLZONES 属性が True に設定されているため、すべてのゾーンで同一である必要があり、大域管理者のみが管理できます。このため、大域ゾーンでこれらのパッケージを手動で削除し、正しいパッケージに置き換えてください。

  Java ES インストーラは、選択した製品コンポーネントを非大域ゾーンにインストールしようとして、SJWC をアップグレードする必要があることを検出した場合、処理を中止します。これは、Solaris 10 の Update 1 および 2 にインストールするときに発生します。

  回避方法として、SJWC の古いパッケージを大域ゾーンから削除し、正しいゾーン伝播属性値を持つ Release 5 の SJWC に置き換える、専用のスクリプトが開発されています。詳細については、『Java Enterprise System 5 インストールガイド (UNIX 版)』を参照してください。

- 共通エージェントコンテナ (CAC): Version 1.1 は、Sun Cluster、Sun Cluster Geographic Edition または Sun Cluster エージェントがインストールされている場合にのみインストールされます。「すべての共有コンポーネント」オプションを選択した場合はインストールされません。その場合は、Version 2.0 のみがインストールされます。

第 2 章

Java ES 共有コンポーネントのアップグレード

この章では、Java ES 共有コンポーネントの Java ES 5 (Release 5) へのアップグレードについて説明します。

Java ES の各製品コンポーネントは、1 つ以上の Java ES 共有コンポーネントというローカルで共有されるライブラリに依存します。共有コンポーネントは、インストールされる製品コンポーネントの内容に基づいて、製品コンポーネントのインストール時に Java ES インストーラによって自動的にインストールされます。これらを Java ES 製品コンポーネントの配備中に明示的に選択、インストール、または設定することはありません。

同様に、Java ES インストーラを使用してアップグレードできる製品コンポーネントについては、対応する共有コンポーネントが自動的にアップグレードされます。

ただし、製品コンポーネントを手動でアップグレードする場合は、この『アップグレードガイド』のいくつかの章で説明しているように、この章で説明する手順を使用して、明示的に共有コンポーネントのアップグレードの実行を必要とする場合があります。

この章で説明する内容は、次のとおりです。

- 68 ページの「共有コンポーネントのアップグレードの概要」
- 72 ページの「共有コンポーネントのアップグレード手順」
- 73 ページの「特別なアップグレード手順」

共有コンポーネントのアップグレードを実行する前に、73 ページの「特別なアップグレード手順」を必ず読んでください。

共有コンポーネントのアップグレードの概要

第 1 章「アップグレードの計画」で説明したように、共有コンポーネントの Java ES 5 (Release 5) へのアップグレードは、より規模の大きいアップグレード計画の一部として実行するようにしてください。アップグレードを成功させるためには、第 1 章を熟読し、必要に応じたアップグレード計画を準備してください。

この節で説明する内容は、次のとおりです。

- 一般的な注意点
- 共有コンポーネントの同期
- すべての共有コンポーネントの同期
- Solaris 10 ゾーンに関する注意点

一般的な注意点

共有コンポーネントをアップグレードするときは、次の問題を考慮してください。

- **オペレーティングシステムの問題** : 44 ページの「オペレーティングシステムの考慮事項」で説明した手順に従って、オペレーティングシステムのアップグレードを実行します。Solaris 10 オペレーティングシステム以外のすべてのプラットフォームで、共有コンポーネントをアップグレードする前にオペレーティングシステムをアップグレードします。
- **順序付けのガイドライン** : 58 ページの「アップグレード手順に関するガイドライン」に説明されている順序付けのガイドラインを確認してください。通常、先に共有コンポーネントをアップグレードします。ただし、アップグレードプロセスを開始する前に、Java ES Release 5 にアップグレードするための全体の順番を理解する必要があります。

共有コンポーネントの同期

Java ES 共有コンポーネントと Java ES 製品コンポーネントの相互作用が数多く (30 個程度) あって複雑な場合は、そのテストやサポートが難しいため、単一のオペレーティングシステムインスタンスに含まれるすべての共有コンポーネントを同じ Java ES バージョンに同期する必要があります。オペレーティングシステムインスタンスとは、Solaris 9 または Linux オペレーティングシステムを実行する 1 台のコンピュータを意味します。また、Solaris 10 オペレーティングシステムの場合は、1 台のコンピュータで実行されているいずれかの仮想オペレーティングシステム環境 (ゾーン) を意味します。

つまり、1 つのオペレーティングシステムインスタンスにインストールされたすべての Java ES 共有コンポーネントは、同じバージョンでなければなりません。この同期の要件により、Java ES 共有コンポーネントのインストール方法やアップグレード方法に一定の制限が与えられます。

- バージョンが異なる Java ES 共有コンポーネントは、異なるオペレーティングシステムインスタンスにのみ配置できます。たとえば、あるオペレーティングシステムインスタンスに Java ES Release 4 の共有コンポーネントをインストールし、別のオペレーティングシステムインスタンスに Java ES Release 5 の共有コンポーネントをインストールすることはできますが、それらを同じオペレーティングシステムインスタンス内で組み合わせることはできません。
- 1 つのオペレーティングシステムインスタンス内のいずれかの共有コンポーネントをアップグレードした場合や、新しいバージョンの共有コンポーネントを導入した場合は、そのオペレーティングシステムインスタンス内のすべての共有コンポーネントを同時にアップグレードしてください。共有コンポーネントには下位互換性が必要とされるため、Release 4 の製品コンポーネントを Release 5 の共有コンポーネントと組み合わせて使用しても問題ありません。

たとえば、1 つ以上の Release 4 の製品コンポーネントが存在するオペレーティングシステムインスタンスに Release 5 の製品コンポーネントがインストールされたとします。Release 5 の製品コンポーネントにはいくつかの Release 5 の共有コンポーネントが必要なため、Release 5 の製品コンポーネントのインストールと同時に、そのオペレーティングシステムインスタンス内に存在するすべての Release 4 の共有コンポーネントを Release 5 にアップグレードすることが同期の要件になります。これは、インストールされる Release 5 の製品コンポーネントが、すでにインストールされているものとは異なる共有コンポーネントを必要とする場合にも当てはまります。

同様に、Release 4 の製品コンポーネントを Release 5 にアップグレードし、そのアップグレードに伴って、依存先であるいくつかの共有コンポーネントもアップグレードする必要がある場合は、アップグレードする特定の Release 4 の製品コンポーネントがすべての共有コンポーネントに対して依存関係を持っているかどうかに関係なく、そのオペレーティングシステムインスタンスにインストールされているすべての共有コンポーネントを Release 5 にアップグレードしてください。

すべての共有コンポーネントの同期

Java ES インストーラは、すべての共有コンポーネントを Release 5 バージョンに同期しなければならない場合のために、すべての共有コンポーネントの同期機能を備えています。

Java ES インストーラのコンポーネント選択ページで「すべての共有コンポーネント」を選択すると、インストーラは、特定の製品コンポーネントに必要かどうかに関係なく、既存のすべての共有コンポーネントをアップグレードし、不足している共有コンポーネントがあればインストールします。

すべての共有コンポーネントの同期機能は、Java ES インストーラでは明示的にアップグレードされない製品コンポーネント (つまり、Application Server、Message Queue、HADB、および Java DB を除くすべての Java ES 製品コンポーネント) のアップグレードをサポートしています。たとえば、すべての共有コンポーネントの同期機能は、Web Proxy Server や Portal Server を Release 5 にアップグレードするのに使用します。

この場合にすべての共有コンポーネントを同期する理由は、現時点ではインストーラがコンピュータ上のどの共有コンポーネントを同期する必要があるかを識別できないためです。したがって、この機能は、すべての Java ES 共有コンポーネントをインストールするか、または Release 5 バージョンにアップグレードします。

すべての共有コンポーネントの同期機能は、次の「Solaris 10 ゾーンに関する注意点」で説明されている、いくつかのゾーンのシナリオにも対応します。

Solaris 10 ゾーンに関する注意点

Solaris 10 の複数ゾーン環境に Java ES を配備する方法に関する制限事項の一部は、共有コンポーネントに関する注意事項から派生しています。

これらの注意事項の中でもっとも重要なのは、疎ルートゾーンの読み取り専用ファイルシステムのために、多くの共有コンポーネントが疎ルートゾーンにインストールできないことです。この制限事項は、ベースディレクトリが `/usr` (デフォルトで大域ゾーンによって共有されるディレクトリ) である共有コンポーネントに適用されます。

疎ルートゾーンに Java ES 共有コンポーネントの多くをインストールできないということは、そのような共有コンポーネントに対して依存関係を持つ製品コンポーネントを疎ルートゾーンに正常にインストールまたはアップグレードするためは、共有コンポーネントをまず大域ゾーンにインストールまたはアップグレードし、そこから非大域ゾーンに伝播させる必要があるということを意味します。

共有コンポーネントに関する同期の要件によって (68 ページの「共有コンポーネントの同期」を参照)、また大域ゾーンでの共有コンポーネントのインストールまたはアップグレードは、疎ルートゾーンでインストールまたはアップグレードされる製品コンポーネントと適応する必要があるため、大域ゾーン内のすべての共有コンポーネントを Release 5 バージョンに同期する必要があります。これは、疎ルートゾーン内の特定の製品コンポーネントに必要かどうかに関係なく、既存のすべての共有コンポーネントをアップグレードし、不足している共有コンポーネントをインストールすることを意味します。

このような大域ゾーン内の共有コンポーネントの同期は、Java ES インストーラのコンポーネント選択ページにある「すべての共有コンポーネント」を選択することによって実行できます。

(たとえば、Java ES 製品コンポーネントを大域ゾーンにインストールすることによって) 共有コンポーネントを大域ゾーンにインストールし、そこから伝播させる場合は、すべてのゾーンで共有コンポーネントの同期を維持するために特別な配慮が必要です。そうしないと、非大域ゾーン内の以前のバージョンの共有コンポーネントと大域ゾーンから伝播された Release 5 の共有コンポーネントが混在する可能性があります。

共有コンポーネントのアップグレード方法

特定の Java ES 製品コンポーネントをアップグレードするのに必要な共有コンポーネントをアップグレードするには、2 つの方法が考えられます。1 つは、必要なすべての共有コンポーネントを特定し、それらを Release 5 バージョンに手動でインストールまたはアップグレードする方法です。もう 1 つは、Java ES インストーラのすべての共有コンポーネントの同期機能を使用する方法です。

ただし、共有コンポーネントの同期の要件 (68 ページの「共有コンポーネントの同期」を参照) のため、現実的なのは Java ES インストーラのすべての共有コンポーネントの同期機能を使用する方法だけです。

これは、ほとんどの場合、特定の状況でどの共有コンポーネントをインストールまたはアップグレードする必要があるかを特定するのが非常に難しく、たとえそれがうまくできたとしても、必要な共有コンポーネントを手動でアップグレードするのは簡単でないためです。共有コンポーネントの中には、パッチを適用して Release 5 バージョンにアップグレードできるものもありますが、以前のパッケージを Release 5 のパッケージに置き換える必要があるものもあります。アップグレードパスによっては、以前のパッケージを手動で削除しなければならないこともあります。

以前は、共有コンポーネントの手動によるアップグレードの難しさが、製品コンポーネントのアップグレードの大きな障害になっていました。すべての共有コンポーネントの同期機能は、不要かもしれない共有コンポーネントがインストールされるという欠点はありますが、Java ES Release 3 および Release 4 で必要だった共有コンポーネントの手動アップグレードと比べて大幅に改善されています。

共有コンポーネントのアップグレード手順

共有コンポーネントのアップグレード手順では、Java ES インストーラのすべての共有コンポーネントの同期機能を使用します。

1. root としてログインするか、スーパーユーザーになります。

   ```
   su -
   ```

2. 73 ページの「特別なアップグレード手順」に従ってサービスまたはプロセスを停止します。

3. Java ES インストーラを起動します。

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

   ここで、*os_arch* は、`Solaris_sparc` などのプラットフォームに対応します。コマンド行インタフェースには、`installer -nodisplay` オプションを使用します。

   ようこそページと、ライセンス使用許諾契約ページが表示されてから、コンポーネント選択ページが表示されます。Java ES インストーラが直接アップグレードできるインストール済みのコンポーネントが検出された場合、それらのコンポーネントは「アップグレード可」の状態で表示されます。

4. コンポーネント選択ページで、「すべての共有コンポーネント」を選択します。

5. 選択した項目を確認します。

   すべての共有コンポーネントが、それぞれの Release 5 バージョンに同期します。

6. Java ES インストーラを終了します。

   その後の手順については、73 ページの「特別なアップグレード手順」を参照してください。

特別なアップグレード手順

この節では、次の共有コンポーネントをアップグレードするのに必要な特別な手順について説明します。

- 73 ページの「Java SE のアップグレード手順」
- 75 ページの「共通エージェントコンテナのアップグレード手順」

Java SE のアップグレード手順

Java ES Release 5 は Java Platform, Standard Edition (Java SE) Version 5.0 Update 9 (ここでは、Java SE 5.0 Update 9 と表記) で動作保証されています。Java SE 5.0 は、開発者バージョン 1.5.0 と呼ばれることもあります。

他の共有コンポーネントと同様、Java SE は Java ES インストーラによってアップグレードされます。ただし、インストーラは、ほかの共有コンポーネントがインストールまたはアップグレードされるかどうかに関係なく、Java SE 5.0 Update 9 にアップグレードします。Java SE 5.0 Update 9 をインストールするために「すべての共有コンポーネント」を明示的に選択する必要はありません。

注	Java SE のアップグレード時には、最初に現在インストールされている Java SE に依存するサービスをすべて停止する場合があります。これは、そのようなサービスによって生じる問題を回避するためです。Java SE に依存するサービスを停止しない場合は、Java SE を Release 5 バージョンにアップグレードしてから、システムを再起動するようにしてください。

インストーラがコンピュータ上で旧バージョンの Java SE パッケージまたは不完全なパッケージセットを検出した場合 (完全なセットは、`SUNWj5rt`、`SUNWj5rtx`、`SUNWj5dev`、`SUNWj5dmo`、`SUNWj5dvx`、`SUNWj5man`、`SUNWj5cfg`、および `SUNWj5dmx`)、ダイアログが表示されます。このダイアログの次にコンポーネント選択ページが表示され、Java SE を Release 5 レベルに自動的にアップグレードするか、手動でアップグレードを実行したい場合は自動アップグレードを回避することを選択できます。

- **手動アップグレード** : Java SE の手動アップグレードを選択した場合、次の手順に従います。
  a. 他のコンポーネントをインストールまたはアップグレードする前に Java ES インストーラを終了します。
  b. 適切なバージョンの Java SE を手動でインストールします。

c. Java ES インストーラを再起動します。

インストーラが正しいバージョンまたは完全なバージョンの Java SE を検出した場合は、次に進むことができますが、そうでない場合は前述のダイアログが表示されます。

- **自動アップグレード**: Java SE の自動アップグレードを選択した場合、Java ES インストーラは Java SE を Version 5.0 Update 9 にアップグレードします。アップグレードは次のように動作します。
  - 以前にインストールされた Java SE のメジャーリリース (たとえば、Java SE Version 1.4.2 や Java SE Version 1.6.*x*) は、ほかのアプリケーションがそのバージョンに依存している可能性があるため、アップグレードによって削除されません。ただし、次の表に示すように、アップグレードによって Release 5 バージョンを参照するシンボリックリンクが設定されます。

表 2-1 プラットフォームごとの Java SE へのシンボリックリンク

プラットフォーム	シンボリックリンク	場所
Solaris	/usr/jdk/entsys-j2se	/usr/jdk/instances/jdk1.5.0
Linux	/usr/jdk/entsys-j2se	/usr/java/jdk1.5.0_09

以前のバージョンを必要とするサービスのため、Java SE のメジャーリリース (たとえば、Java SE 1.4.2) へのあらゆるポインタを保持するようにしてください。以前のバージョンの Java SE へのシンボリックリンクを保持する方法については、該当する製品コンポーネントのマニュアルを参照してください。

  - 以前にインストールされた Java SE のマイナーリリース (たとえば、Java SE Version 5.0 Update 5) は、アップグレードによって削除され、Java SE 5.0 Update 9 に置き換えられます。

Java SE のシンボリックリンクの確認

Java Enterprise System は、Java ES サービスが使用する正しい Java SE ランタイムを確実に見つけられるように、サポートされるバージョンの Java SE プラットフォームへのシンボリックリンクを保持します。

シンボリックリンクを確認します。Solaris オペレーティングシステムでは次のようになります。

```
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
```

ここで、/usr/jdk/instances/jdk1.5.0 はデフォルトの場所です。

現在の Java SE バージョンの確認

Java ES の現在のインストールが使用している Java SE のバージョンを確認するには、次のコマンドを実行します。このコマンドは、Java SE のシンボリックリンクによって参照される Java SE のバージョンを検証します。

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

出力の一覧を次の表に示します。

表 2-2　Java SE のバージョン確認の出力

Java ES のリリース	Java SE のバージョン番号	Java SE のバージョン文字列
Release 2	1.4.2 Update 5	`1.4.2_05`
Release 3	5.0 Update 1	`1.5.0_01`
Release 4	5.0 Update 4	`1.5.0_04`
Release 5	5.0 Update 9	`1.5.0_09`

共通エージェントコンテナのアップグレード手順

Sun Cluster ソフトウェアのアップグレードに備えて共通エージェントコンテナの共有コンポーネントをアップグレード (つまり、Version 1.1 のパッチを適用) する場合は、『Sun Cluster Software Installation Guide for Solaris OS』(http://docs.sun.com/doc/819-0420/6n2rlnncr?a=view) の「Upgrading Sun Cluster Software」に記載されている、依存関係を持つソフトウェアのアップグレード手順に従います。

上記のガイドの更新については、『Sun Cluster 3.1 8/05 With Java Enterprise System 5 Special Instructions』(http://docs.sun.com/doc/819-4351) の「Upgrading to Sun Cluster 3.1 8/05 Software」を参照してください。

ほかの Java ES コンポーネントのアップグレード (つまり、Version 2.0 へのアップグレード) に備えて共通エージェントコンテナの共有コンポーネントをアップグレードする場合は、下記のパス名変数に注意しながら、次の手順に従います。

表 2-3　共通エージェントコンテナのディレクトリパス

パス名変数	Solaris OS	Linux OS
rel4CACbase-dir	/opt/SUNWcacao	/opt/sun/cacao
rel5CAC-admin-dir	/usr/lib/cacao	/opt/sun/cacao

1. 現在のインストールでカスタム設定が使用されている場合、次のコマンドを使用して設定を確認します。

   rel4CAC-base-dir/bin/cacaoadm list-params

   出力内容は次のようになります。

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

   上記の例は、デフォルト値を示しています。76 ページの手順 4 で使用するためにデフォルト設定以外をすべてメモします。

2. 次のコマンドを使用して、共通エージェントコンテナのプロセスを停止します。

   rel4CAC-base-dir/bin/cacaoadm stop
   echo $?

   終了コードが 0 以外の場合、次のように指定して強制的に停止します。

   rel4CAC-base-dir/bin/cacaoadm stop -f

3. Java ES インストーラのすべての共有コンポーネントの同期機能を使用して、共通エージェントコンテナをアップグレードします。

   72 ページの「共有コンポーネントのアップグレード手順」を参照してください。

4. 76 ページの手順 1 で確認したカスタム設定をすべて適用します。

   rel5CAC-admin-dir/bin/cacaoadm set-param java-flags= 値
   rel5CAC-admin-dir/bin/cacaoadm set-param jmxmp-connector-port= 値
   rel5CAC-admin-dir/bin/cacaoadm set-param snmp-adaptor-port= 値
   rel5CAC-admin-dir/bin/cacaoadm set-param
   snmp-adaptor-trap-port= 値
   rel5CAC-admin-dir/bin/cacaoadm set-param
   commandstream-adaptor-port= 値
   rel5CAC-admin-dir/bin/cacaoadm set-param retries= 値

5. Java SE から Java SE Version 5 へアップグレードした場合は、rebuild-dependencies ユーティリティーを実行します。

   rel5CAC-admin-dir/bin/cacaoadm rebuild-dependencies

   このコマンドの出力内容は、次のとおりです。

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

6. 共通エージェントコンテナのサービスを再起動します。

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

7. 共通エージェントコンテナがアップグレードされたことを確認します。

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

第 3 章

Sun Cluster ソフトウェア

この章では、Sun Cluster ソフトウェアを Java ES 5 (Release 5) Sun Cluster 3.1 8/05 ソフトウェアにアップグレードする方法について説明します。

ここでは、Sun Cluster ソフトウェアの Release 5 へのアップグレードに関する注意点の概要を示します。

Sun Cluster ソフトウェアは Solaris プラットフォームだけでサポートされます。

この章で説明する Sun Cluster ソフトウェアのアップグレードでは、Sun Cluster フレームワークソフトウェアおよび Sun Cluster データサービスソフトウェアつまりエージェントの両方を含みます。

- 80 ページの「Sun Cluster ソフトウェアアップグレードの概要」
- 84 ページの「Sun Cluster ソフトウェアの Java ES Release 5 へのアップグレード」

Sun Cluster ソフトウェアアップグレードの概要

この節では、Java ES 5 (Release 5) へのアップグレードに影響を与える次の Sun Cluster ソフトウェアの一般的な側面について説明します。

- Release 5 Sun Cluster ソフトウェアについて
- Sun Cluster ソフトウェアのアップグレードロードマップ
- Sun Cluster のデータ
- Sun Cluster のアップグレード方針

Release 5 Sun Cluster ソフトウェアについて

Release 5 の Sun Cluster ソフトウェアは、Release 4 の Sun Cluster ソフトウェアに対するマイナーアップグレードです (『Sun Cluster Release Notes』 (http://docs.sun.com/doc/819-1405/6n3p13hac?a=view) を参照)。Release 5 の Sun Cluster は、Solaris 8 オペレーティングシステムをサポートしない点を除いて、Release 4 と基本的に同じです。

Sun Cluster 3.1 8/05 ドキュメントセットへの変更や追加については、『Sun Cluster 3.1 8/05 With Java Enterprise System 5 Special Instructions』 (http://docs.sun.com/doc/819-4351) を参照してください。そのほかの点では、Sun Cluster 3.1 8/05 ドキュメントセットに記載された手順は、Release 5 の Sun Cluster ソフトウェアで有効です。

注

Solaris 8 OS 上で Sun Cluster 3.1 8/05 ソフトウェアが必要な場合は、http://www.sun.com/software/javaenterprisesystem/previous/index.xml にある Java ES Release 4 ディストリビューションから Sun Cluster 3.1 8/05 ソフトウェアを入手してください。

Sun Cluster 3.1 8/05 ソフトウェアの Solaris 8 OS へのインストールは、Java ES 5 インストーラではサポートされなくなりました。

Sun Cluster 3.1 8/05 ソフトウェアを Solaris 8 OS にインストールするには、次の手順を実行します。

1. Java ES Release 4 ディストリビューションから Sun Cluster 3.1 8/05 ソフトウェアをインストールします。
2. Sun Cluster 3.1 8/05 ソフトウェアに必要なすべてのパッチをインストールします。

Sun Cluster ソフトウェアのアップグレードロードマップ

表 3-1 は、Sun Cluster の、Java ES 5 (Release 5) へのサポートされているアップグレードパスを示します。この表は、Solaris オペレーティングシステムだけに適用されます。

Sun Cluster のバージョンは、Java ES のリリースと 1 対 1 に対応していません。以前は、Sun Cluster ソフトウェアの暫定機能リリース (Interim Feature Release、IFR) が Java ES の正式リリースの間に Java ES に組み込まれていました。このため、表 3-1 に示すように、Java ES Release 2、Release 3、および Release 4 Sun Cluster から Release 5 Sun Cluster へのアップグレードには、Sun Cluster 3.1 4/04、Sun Cluster 3.1 9/04、および Sun Cluster 3.1 8/05 から Release 5 へのアップグレードが含まれます。

表 3-1 Java ES 5 (Release 5): Sun Cluster 3.1 8/05 ソフトウェアへのアップグレードパス

Java ES のリリース	Sun Cluster ソフトウェアバージョン	一般的な手順	必要な再設定
Release 4	Sun Cluster 3.1 8/05 (2005Q4)	直接アップグレード[1]: Sun Cluster の `scinstall` ユーティリティーを使用して実行されます。	なし。
Release 3	Sun Cluster 3.1 9/04 または Sun Cluster 3.1 8/05 (IFR)	直接アップグレード: Sun Cluster の `scinstall` ユーティリティーを使用して実行されます。	クラスタ構成は、アップグレードされたバージョンに自動的に移行されます。 IFR の Release 5 へのアップグレードには、再設定は必要ありません。
Release 2	Sun Cluster 3.1 4/04 または Sun Cluster 3.1 9/04 (IFR)	直接アップグレード: Sun Cluster の `scinstall` ユーティリティーを使用して実行されます。	クラスタ構成は、アップグレードされたバージョンに自動的に移行されます。
Release 1	Sun Cluster 3.1	直接アップグレードは動作保証なし: ただし、`scinstall` ユーティリティーを使用することにより実行できます。	クラスタ構成は、アップグレードされたバージョンに自動的に移行されます。
それ以前の Java ES リリース	Sun Cluster 3.0	直接アップグレードは動作保証なし: ただし、`scinstall` ユーティリティーを使用することにより実行できます。	クラスタ構成は、アップグレードされたバージョンに自動的に移行されます。

1. Sun Cluster ソフトウェアによって使用される共有コンポーネントが Release 5 にアップグレードされていないかぎり、Release 4 から Release 5 へのアップグレードは必要ありません。

Sun Cluster のデータ

次の表は、Sun Cluster ソフトウェアのアップグレードにより影響を受ける可能性のあるデータの種類を示しています。

表 3-2　Sun Cluster のデータの使用法

データの種類	場所	使用法
クラスタ設定データ	クラスタ構成リポジトリ。すべてのクラスタノードに渡って複製および同期されます。**注意 :** CCR ファイルを手動で編集しないでください。ノードまたはクラスタ全体が機能を停止する原因になることがあります。	クラスタノードの設定、フェイルオーバー機構、リソース管理など、Sun Cluster 操作のすべての局面に対する設定情報を保存します。

Sun Cluster のアップグレード方針

Sun Cluster をアップグレードするための方針は一般に、第 1 章「アップグレードの計画」で説明されている多くの考慮事項 (アップグレードパス、Java ES コンポーネント間の依存関係、選択的なアップグレードかすべてのアップグレードか、複数インスタンスの配備など) によって異なります。

ここでは、Sun Cluster のアップグレード計画に影響する可能性のある問題を取り上げ、Sun Cluster の全般的な説明を詳細に記述します。

互換性の問題

Java ES Release 5 Sun Cluster ソフトウェアには新しいグラフィカル管理インタフェースが含まれます。ただし、以前のリリースの Sun Cluster エージェントと下位互換性があります。

Sun Cluster の依存関係

ほかの Java ES コンポーネントに対する Sun Cluster の依存関係が、Sun Cluster ソフトウェアのアップグレードおよび再設定の手順に影響を及ぼす可能性があります。たとえば、Sun Cluster のインタフェースや機能の変更が原因で、Sun Cluster ソフトウェアが依存するコンポーネントのバージョンのアップグレードが必要になる場合があります。このようなコンポーネントのアップグレードの必要性は、個々のアップグレードパスによって異なります。

Sun Cluster は、次の Java ES コンポーネントに依存します。

- **共有コンポーネント** : Sun Cluster ソフトウェアは、特定の Java ES 共有コンポーネントに依存します (48 ページの表 1-9 を参照)。
- **データサービス** : Sun Cluster ソフトウェアには、Java ES 製品コンポーネントの可用性を高めるために、特定のデータサービス (またはエージェント) が必要です。Sun Cluster 環境で実行している製品コンポーネントごとに、対応するクラスタリソースを管理するための対応するデータサービスが必要です。通常、エージェントパッケージは Sun Cluster アップグレードプロセスの一部としてアップグレードされます。

二重のアップグレード

二重のアップグレード (44 ページの「二重のアップグレード : Java ES ソフトウェアとオペレーティングシステムソフトウェア」で説明されているように、ここでは Sun Cluster とオペレーティングシステムの両方をアップグレードすること) は、順次アップグレードの一部ではなく、Sun Cluster の非順次アップグレードの処理の中で実行されます。

手順の詳細については、『Sun Cluster Installation Guide』(http://docs.sun.com/doc/819-0420/6n2rlnnd1?a=view) のアップグレードに関する章で説明されています。Release 5 の Sun Cluster に関する変更については、『Sun Cluster 3.1 8/05 With Java Enterprise System 5 Special Instructions』(http://docs.sun.com/doc/819-4351) で説明されています。

この手順は、Solaris 8 または Solaris 9 から Solaris 10 への Solaris オペレーティングシステムのアップグレードに適用されます。

Sun Cluster ソフトウェアの Java ES Release 5 へのアップグレード

この節では、Sun Cluster ソフトウェアの Java ES 2005Q4 (Release 4)、Java ES 2005Q1 (Release 3)、および Java ES 2004Q2 (Release 2) から Java ES 5 (Release 5) へのアップグレードに関する情報を提供します。これらの Java ES リリースに含まれる 3 つの Sun Cluster バージョン (Sun Cluster 3.1 4/04 ソフトウェア、Sun Cluster 3.1 9/04 ソフトウェア、および Sun Cluster 3.1 8/05 ソフトウェア) では、アップグレードの手順は同じです。

Sun Cluster 3.1 8/05 (Release 3 または Release 4) ソフトウェアがすでにインストールされている場合は、Sun Cluster ソフトウェアによって使用される共有コンポーネントが Release 5 にアップグレードされていないかぎり、Release 5 へアップグレードする必要はありません。

この節で説明する内容は、次のとおりです。

- はじめに
- Sun Cluster のアップグレード

はじめに

Sun Cluster ソフトウェアを Java ES Release 5 にアップグレードするときには、アップグレードプロセスに関する次の事項を検討します。

- **一般的なアップグレード手順** : アップグレードは、`scinstall` スクリプトを実行して行われます。このスクリプトは、Sun Cluster ソフトウェアをアップグレードし、ソフトウェアのアップグレード後に Sun Cluster の以前の設定を適用します。ただし、クラスタ環境内のすべてのノードを同じバージョンにアップグレードする必要があります。これは、クラスタを停止し、すべてのノードをアップグレードする方法、またはクラスタを停止せずに、ノードを 1 つずつ連続してアップグレードする順次アップグレードによって行います。
- **アップグレードの依存関係** : Sun Cluster ソフトウェアは多くの Java ES 共有コンポーネントに依存しますが (48 ページの表 1-9 を参照)、これらはアップグレード依存関係が弱いものであり、Java ES Release 5 の Sun Cluster ソフトウェアはこれらのコンポーネントの Release 4 バージョンと互換性があります。
- **下位互換性** : Release 5 の Sun Cluster (フレームワーク) ソフトウェアは以前のクラスタエージェントと下位互換性があり、Release 5 の Sun Cluster エージェントソフトウェアは Java ES Release 4 のコンポーネントと下位互換性があります。ただし、クラスタ内のすべてのノードは、同じバージョンのフレームワークとエージェントソフトウェアを実行する必要があります。

- **アップグレードのロールバック** : Sun Cluster ソフトウェアの Release 5 へのアップグレードを以前のバージョンにロールバックすることはサポートされていません。
- **プラットフォームの問題** : Sun Cluster ソフトウェアのアップグレード手順はすべての Solaris プラットフォームとハードウェアアーキテクチャーで同じですが、Sun Cluster ソフトウェアは Linux プラットフォームでサポートされていません。

Sun Cluster のアップグレード

この節では、Sun Cluster ソフトウェアを Java ES Release 4 から Java ES Release 5 にアップグレードする方法の概要を示します。

- アップグレード前の作業
- Sun Cluster ソフトウェアのアップグレード
- アップグレードの確認
- アップグレード後の作業
- アップグレードのロールバック

ここでは、Sun Cluster の非順次アップグレードについて説明します。順次アップグレードでは、クラスタを停止しないという点で異なります。ただし、次に説明するように、どちらの場合も 1 つのクラスタノードに対する一般的な手順は同じです。個別の手順については、『Sun Cluster Installation Guide』(http://docs.sun.com/doc/819-0420/6n2rlnncr?a=view) のアップグレードに関する章で説明されています。Release 5 の Sun Cluster に関する変更については、『Sun Cluster 3.1 8/05 With Java Enterprise System 5 Special Instructions』(http://docs.sun.com/doc/819-4351) で説明されています。

アップグレード前の作業

Sun Cluster ソフトウェアをアップグレードする前に、次の作業を実行してください。

- 現在のバージョン情報の確認
- Sun Cluster Geographic Edition インフラストラクチャーの停止
- アップグレードのためのクラスタノードの準備
- オペレーティングシステムのアップグレード
- Sun Cluster が依存するコンポーネントのアップグレード
- 必要な設定情報とパスワードの入手

現在のバージョン情報の確認

次のコマンドを入力することにより、Sun Cluster ソフトウェアの現在のバージョンを確認できます。

```
scinstall -pv
```

このコマンドは、Sun Cluster のバージョンと、インストールされている各ソフトウェアパッケージのバージョンを返します。このコマンドによって 3.1 8/05 のバージョンである `3.1u4` が返された場合は、パッチのリビジョン番号を調べて Release 4 または Release 5 のソフトウェアがインストールされているかどうかを確認します (次の表を参照)。

表 3-3 Sun Cluster のバージョン確認の出力

Java ES のリリース	Sun Cluster のバージョン番号	パッチのリビジョン番号
Release 1 (Sun Cluster 3.1)	`3.1`	
Release 2 (Sun Cluster 3.1 4/04)	`3.1u2`	
Release 2 または Release 3 (Sun Cluster 3.1 9/04)	`3.1u3`	
Release 3 または Release 4 (Sun Cluster 3.1 8/05)	`3.1u4`	Solaris 9 SPARC: 117949-15 Solaris 8 SPARC: 117950-15 Solaris 9 x86: 117909-15
Release 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. この表に示されているパッチのリビジョン番号は、各プラットフォームで Java ES Release 5 にアップグレードするための最低要件です。表に記載されているものよりも新しいリビジョンのパッチが使用可能になった場合は、新しいほうのパッチを使用してください。

Sun Cluster Geographic Edition インフラストラクチャーの停止

Sun Cluster Geographic Edition ソフトウェアを実行するクラスタをアップグレードする場合、最初に Sun Cluster Geographic Edition インフラストラクチャーを停止した後、クラスタのアップグレードを準備するための他の手順を実行する必要があります。『Sun Cluster Geographic Edition Installation Guide』(http://docs.sun.com/doc/819-8004/6n9tmd19d?=view) に記載されているアップグレードの手順に従ってください。記載されている手順のなかに、アップグレードの適切な段階で Sun Cluster ソフトウェアをアップグレードする手順が含まれます。

アップグレードのためのクラスタノードの準備

Sun Cluster ソフトウェアをアップグレードする前に、クラスタノードをクラスタ環境から削除する必要があります。

- **非順次アップグレード**: クラスタ環境からノードを削除することは、環境を停止することを意味します。つまり、リソースグループのオフライン化と無効化、環境内で実行中のアプリケーションの停止、共有データのバックアップ、クラスタの停止、システムディスクのバックアップ、および非クラスタモードでのノードの再起動を行います。
- **順次アップグレード**: クラスタ環境からノードを削除することは、ノードからのすべてのリソースグループとデバイスグループの移動、共有データとシステムディスクのバックアップ、および非クラスタモードでのノードの再起動を意味します。

これらの手順の詳細と、特定の状況で必要なその他の手順については、『Sun Cluster Installation Guide』(http://docs.sun.com/doc/819-0420/6n2rlnncr?a=view) のアップグレードに関する章で詳しく説明されています。

オペレーティングシステムのアップグレード

オペレーティングシステムの最新バージョンへのアップグレードや、使用しているボリュームマネージャーのバージョンのアップグレードに、アップグレードによるダウンタイムを利用することができます。

それらの手順ついては、『Sun Cluster Installation Guide』(http://docs.sun.com/doc/819-0420/6n2rlnncr?a=view) のアップグレードに関する章で詳しく説明されています。

Sun Cluster が依存するコンポーネントのアップグレード

一般的には、コンピュータシステム上 (および、コンピューティング環境内) のすべての Java ES コンポーネントを Release 5 にアップグレードすることが推奨されます。Sun Cluster ソフトウェアが依存する Release 4 共有コンポーネントのアップグレードは任意であり、第 2 章「Java ES 共有コンポーネントのアップグレード」の説明に従って実行できます。Release 2 共有コンポーネントの Release 5 へのアップグレードは必須です。

注

Java ES 共有コンポーネントが Release 5 に同期している場合は、Sun Cluster ソフトウェアを Release 5 にアップグレードしてください。これは、Release 4 の Sun Cluster は Release 5 Sun Java Web Console 共有コンポーネントと互換性がないためです。

Sun Java Web Console が Release 5 (Ver 3.0) にアップグレードされている場合は、Sun Cluster の最新のパッチを適用するか、この章の説明に従って Sun Cluster をアップグレードするようにしてください。

必要な設定情報とパスワードの入手

現在インストールされているバージョンに関する特別な情報は必要ありません。ただし、アップグレードを実行するには、スーパーユーザーとしてログインする必要があります。

Sun Cluster ソフトウェアのアップグレード

ここでは、Sun Cluster ソフトウェアのアップグレード手順に影響する注意点を示したあと、その手順自体について説明します。

アップグレードの注意点

Sun Cluster ソフトウェアの Java ES Release 5 へのアップグレードでは、次の注意点を考慮に入れてください。

- Sun Cluster フレームワークソフトウェアのアップグレード時には、高可用性の Java ES コンポーネントやクラスタ環境で実行するその他のアプリケーションを管理するために必要なデータサービスをアップグレードすることをお勧めします。
- Sun Cluster ソフトウェアのアップグレードにより、Java ES コンポーネントやクラスタ環境で実行するその他のアプリケーションをアップグレードする機会も得られます。

アップグレード手順

次の手順は、各クラスタノードでの Sun Cluster ソフトウェアのアップグレードに適用されます。これらの手順はごく一般的です。実行方法ついては、『Sun Cluster Installation Guide』(http://docs.sun.com/doc/819-0420/6n2rlnncr?a=view) のアップグレードに関する章で詳しく説明されています。Release 5 の Sun Cluster に関する変更については、『Sun Cluster 3.1 8/05 With Java Enterprise System 5 Special Instructions』(http://docs.sun.com/doc/819-4351) で説明されています。

1. 非クラスタモードで起動します。
2. root としてログインするか、スーパーユーザーになります。

   ```
   su -
   ```

3. Java ES Release 5 ディストリビューションで次のディレクトリに移動します。

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

   ここで、*os_arch* はプラットフォーム (`Solaris_sparc` など) に対応し、*os-ver* は `Solaris 9` または `Solaris 10` です。

4. `scinstall` ユーティリティーを実行します。

   ```
   ./scinstall
   ```

   クラスタのインストール、設定、およびアップグレードを行うためのメインメニューが表示されます。

5. Sun Cluster フレームワークソフトウェアと必要なデータサービスをアップグレードします。

   対応するリソースを、アップグレードされたリソースに移行することによって、アップグレードされたデータサービスを設定する必要があります (89 ページの「アップグレード後の作業」を参照)。

6. 必要なパッチをすべて Sun Cluster フレームワークソフトウェアとデータサービスに適用します。

   関連するパッチへのアクセスと適用については、『Sun Cluster 3.0-3.1 Release Notes Supplement』(http://docs.sun.com/app/docs/doc/816-3381) に説明されています。

7. クラスタでノードを再起動します。

アップグレードの確認

Sun Cluster ソフトウェアが正常にアップグレードされたかどうかは、次の手順に従って確認できます。

1. 次のコマンドを入力します。

   ```
   scinstall -pv
   ```

   このコマンドは、Sun Cluster のバージョンと、インストールされている各ソフトウェアパッケージのバージョンを返します。このコマンドによって 3.1 8/05 のバージョンである `3.1u4` が返された場合は、パッチのリビジョン番号を調べて Release 4 または Release 5 のソフトウェアがインストールされているかどうかを確認します (86 ページの表 3-3 を参照)。

2. データサービスのアップグレードログファイルをチェックします。

   ログファイルは、アップグレード出力メッセージの末尾に記載されています。

アップグレード後の作業

Sun Cluster ソフトウェアのアップグレード後、多くの作業を実行することが必要な場合があります。これは、非順次アップグレードと順次アップグレードのどちらを行ったかによって異なります。クラスタ環境を完全に復元するために必要なタスクには、次のようなものがあります。

- クラスタ構成の状態の確認。
- 新しいリソースタイプバージョンへのリソースの移行。
- クラスタにインストールされているその他の Java ES コンポーネントやアプリケーションのアップグレード。

インストール後の手順ついては、『Sun Cluster Installation Guide』(http://docs.sun.com/doc/819-0420/6n2rlnncr?a=view) のアップグレードに関する章で詳しく説明されています。

アップグレードのロールバック

Sun Cluster ソフトウェアのロールバックはサポートされていません。アップグレード手順の中で加えた変更を簡単にバックアウトすることはできません。

第 4 章

Sun Cluster Geographic Edition

この章では、Sun Cluster Geographic Edition を Java ES 5 (Release 5) の Sun Cluster Geographic Edition 3.1 2006Q4 にアップグレードする方法について説明します。

ここでは、Release 5 でサポートされているさまざまなアップグレードパスに関する注意点の概要を説明します。なお、ここで取り上げるのは Solaris オペレーティングシステムでのアップグレードのみです。

- 92 ページの「Sun Cluster Geographic Edition のアップグレードの概要」
- 95 ページの「Sun Cluster Geographic Edition の Version 3.1 8/05 からのアップグレード」

注 この章で説明するファイルの場所は、*SunClusterGeo-base* というディレクトリパスを基準に指定されています。このパスの少なくとも一部は、Sun Cluster Geographic Edition がインストールされたときにインストールディレクトリとして指定されている可能性があります。指定されていない場合は、Geographic Edition インストーラによってデフォルト値が割り当てられています。

次の表に、このディレクトリパスのデフォルト値を示します。

表 4-1 Sun Cluster Geographic Edition ディレクトリパス

パス名変数	Solaris OS	Linux OS
SunClusterGeo-base	`/opt/SUNWscgeo`	該当なし

Sun Cluster Geographic Edition のアップグレードの概要

ここでは、Java ES 5 (Release 5) へのアップグレードに影響する Sun Cluster Geographic Edition の一般的な側面について説明します。

- Java ES Release 5 について
- Java ES Release 5 へのアップグレードロードマップ
- Sun Cluster Geographic Edition のデータ
- Sun Cluster Geographic Edition のアップグレード方針

Java ES Release 5 について

Java ES Release 5 Sun Cluster Geographic Edition は、Java ES コンポーネントとして提供される最初のリリースです。Sun Cluster Geographic Edition 3.1 は、最初にスタンドアロン製品としてリリースされました。

Release 5 Sun Cluster Geographic Edition は、Sun Cluster Geographic Edition 3.1 8/05 に対するマイナーリリースです。これは、Solaris x86 プラットフォームをサポートする最初のリリースです。Release 5 Sun Cluster Geographic Edition には、選ばれたバグ修正、各種ソフトウェアおよびハードウェアコンポーネントの限定、および追加のデータレプリケーション製品のサポートも含まれています。

Java ES Release 5 へのアップグレードロードマップ

表 4-2 は、Sun Cluster Geographic Edition の、Java ES Release 5 へのサポートされているアップグレードパスを示しています。この表は、Solaris オペレーティングシステムだけに適用されます。

表 4-2 Java ES 5 (Release 5): Sun Cluster Geographic Edition 3.1 2006Q4 へのアップグレードパス

Java ES のリリース	Sun Cluster Geographic Edition のバージョン	一般的な手順	必要な再設定
それ以前の Java ES リリース	Sun Cluster Geographic Edition 3.1 8/05	直接アップグレード: 新規インストールにより、Sun Cluster Geographic Edition 3.1 8/05 を置き換えます。	なし

Sun Cluster Geographic Edition のデータ

次の表は、Sun Cluster Geographic Edition ソフトウェアのアップグレードにより影響を受ける可能性のあるデータの種類を示しています。

表 4-3 Sun Cluster Geographic Edition のデータの使用法

データの種類	場所	使用法
Sun Cluster Geographic Edition 設定データ	クラスタ構成リポジトリ。すべてのクラスタノードに渡って複製および同期されます。**注意** : CCR ファイルを手動で編集しないでください。ノードまたはクラスタ全体が機能を停止する原因になることがあります。	Sun Cluster Geographic Edition の操作のすべてに対する設定情報を格納します。

Sun Cluster Geographic Edition のアップグレード方針

Sun Cluster Geographic Edition をアップグレードするための方針は一般に、第 1 章「アップグレードの計画」で説明されている多くの考慮事項 (アップグレードパス、Java ES コンポーネント間の依存関係、選択的なアップグレードかすべてのアップグレードか、複数インスタンスの配備など) によって異なります。

ここでは、Sun Cluster Geographic Edition のアップグレード計画に影響する可能性のある問題を取り上げ、Sun Cluster Geographic Edition の全般的な説明を詳細に記述します。

互換性の問題

パートナー関係にある両方のクラスタは、同じバージョンの Sun Cluster Geographic Edition を実行する必要があります。このため、パートナー関係において Sun Cluster Geographic Edition 3.1 8/05 と Release 5 の Sun Cluster Geographic Edition を混在させることはできません。つまり、下位互換性はありません。Release 5 の Sun Cluster Geographic Edition には、Sun Cluster Geographic Edition 3.1 8/05 では読み取ることができない追加の設定データが含まれています。

依存関係

ほかの Java ES コンポーネントに対する Sun Cluster Geographic Edition の依存関係が、Sun Cluster Geographic Edition ソフトウェアのアップグレードおよび再設定の手順に影響を及ぼす可能性があります。たとえば、Sun Cluster Geographic Edition のインタフェースまたは関数が変更された場合は、Sun Cluster Geographic Edition が依存するコンポーネントのアップグレードされたバージョンが必要になる場合があります。このようなコンポーネントのアップグレードの必要性は、個々のアップグレードパスによって異なります。

Sun Cluster Geographic Edition は、次の Java ES コンポーネントに依存します。

- **共有コンポーネント** : Sun Cluster Geographic Edition は、特定の Java ES 共有コンポーネントに依存します (48 ページの表 1-9 を参照)。
- Sun Cluster: Sun Cluster Geographic Edition は、基本機能を提供する Sun Cluster に必ず依存しています。

二重のアップグレード

44 ページの「二重のアップグレード : Java ES ソフトウェアとオペレーティングシステムソフトウェア」で説明されているように、二重のアップグレードとは Sun Cluster Geographic Edition ソフトウェアとオペレーティングシステムの両方をアップグレードすることを言いますが、二重のアップグレードは Sun Cluster Geographic Edition には適用されません。

Sun Cluster Geographic Edition は Sun Cluster に対して強いアップグレード依存関係を持つため、Sun Cluster アップグレードの処理の中でオペレーティングシステムのアップグレードが実行されます。Sun Cluster の二重のアップグレードについては、83 ページの「二重のアップグレード」を参照してください。

Sun Cluster Geographic Edition の Version 3.1 8/05 からのアップグレード

この節では、Sun Cluster Geographic Edition の Version 3.1 から Java ES 5 (Release 5) へのアップグレードに関する情報を示します。この節で説明する内容は、次のとおりです。

- はじめに
- Version 3.1 8/05 の Sun Cluster Geographic Edition のアップグレード

はじめに

Sun Cluster Geographic Edition 3.1 8/05 を Release 5 にアップグレードするときは、アップグレードプロセスに関する次の事項を考慮してください。

- **一般的なアップグレード手順** : アップグレードは、Version 3.1 8/05 をアンインストールし、Release 5 の Sun Cluster Geographic Edition の新規インストールを実行することにより行われます。設定データは保持されます。
- **アップグレードの依存関係** : Sun Cluster Geographic Edition は、いくつかの Java ES 共有コンポーネントに依存しています (48 ページの表 1-9 を参照)。Sun Cluster Geographic Edition のアップグレードを実行すると、すべての共有コンポーネントは Java ES インストーラによって自動的に Release 5 にアップグレードされます。特に、Sun Cluster Geographic Edition は共通エージェントコンテナ Version 1.1 に対して強いアップグレード依存関係を持っています。Sun Cluster Geographic Edition は、Sun Cluster に対しても強いアップグレード依存関係を持っています。Release 5 の Sun Cluster Geographic Edition をサポートするには、Release 4 の Sun Cluster (3.1 8/05) を Release 5 にアップグレードしてください。
- **下位互換性** : Release 5 の Sun Cluster Geographic Edition は、Sun Cluster Geographic Edition 3.1 8/05 との互換性はありません。パートナー関係にあるすべてのクラスタを Release 5 にアップグレードしてください。
- **アップグレードのロールバック** : Sun Cluster Geographic Edition ソフトウェアの Release 5 へのアップグレードを Version 3.1 8/05 にロールバックすることはサポートされていません。
- **プラットフォームの問題** : Sun Cluster Geographic Edition ソフトウェアのアップグレード手順はすべての Solaris プラットフォームとハードウェアアーキテクチャーで同じですが、Sun Cluster Geographic Edition ソフトウェアは Linux プラットフォームでサポートされていません。

Version 3.1 8/05 の Sun Cluster Geographic Edition のアップグレード

この節では、Sun Cluster Geographic Edition を Version 3.1 8/05 から Java ES Release 5 にアップグレードする方法について説明します。この節で説明する内容は、次のとおりです。

- アップグレード前の作業
- 3.1 8/05 の Sun Cluster Geographic Edition のアップグレード
- アップグレードの確認
- アップグレード後の作業
- アップグレードのロールバック

アップグレード前の作業

Sun Cluster Geographic Edition ソフトウェアをアップグレードする前に、次の作業を実行してください。

- 現在のバージョン情報の確認
- アップグレードのためのクラスタの準備
- Sun Cluster Geographic Edition が依存するコンポーネントのアップグレード
- Sun Cluster Geographic Edition のデータのバックアップ
- 必要な設定情報とパスワードの入手

現在のバージョン情報の確認

Sun Cluster Geographic Edition の現在のバージョンは、次のコマンドを使用することにより確認できます。

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

出力されるバージョン文字列を次の表に示します。

表 4-4 Sun Cluster Geographic Edition のバージョン確認の出力

Java ES のリリース	Sun Cluster Geographic Edition のバージョン番号
Version 3.1 8/05	`1.0`
Release 5	`1.1`

アップグレードのためのクラスタの準備

アップグレードするクラスタとパートナー関係にあるすべてのクラスタ上で、次の手順を実行します。この手順を実行すると、製品から Sun Cluster Geographic Edition のレイヤーが削除されます。

1. クラスタが正常に動作していることを確認します。

   クラスタの現在の状態を表示するには、いずれかのノードで次のコマンドを実行します。

   ```
   scstat
   ```

2. root としてログインするか、スーパーユーザーになります。

   ```
   su -
   ```

3. 保護グループからすべてのアプリケーションリソースグループを削除します。

   高可用性のアプリケーションには、Sun Cluster Geographic Edition のアップグレード中の停止時間は発生しません。

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

4. このクラスタとパートナー関係にあるすべてのクラスタ上で、手順 1 〜手順 3 を実行します。

5. クラスタ上でアクティブなすべての保護グループを停止します。

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

6. 両方のパートナークラスタ上のすべてのハートビートから ICRM プラグインを削除します。

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

Sun Cluster Geographic Edition が依存するコンポーネントのアップグレード

一般的には、コンピュータシステム上 (および、コンピューティング環境内) のすべての Java ES コンポーネントを Java ES Release 5 にアップグレードすることが推奨されます。Sun Cluster Geographic Edition は、いくつかの共有コンポーネントと Sun Cluster に対して強いアップグレード依存関係を持っています。

Sun Cluster Geographic Edition が依存するコンポーネントをアップグレードするときは、Sun Cluster Geographic Edition をアップグレードする前に、(すでにアップグレードした可能性があるものを省いて) 次の順序で実行してください。共有コンポーネントのアップグレードは通常、Java ES インストーラによって自動的に実行されます。

1. **共有コンポーネント**: Java ES 共有コンポーネントを Release 5 に同期する手順は、67 ページの「Java ES 共有コンポーネントのアップグレード」で説明されています。ただし、Sun Cluster Geographic Edition の Release 5 へのアップグレードを実行すると、Sun Cluster Geographic Edition で必要な共有コンポーネントはすべて、Java ES インストーラによって自動的にアップグレードされます。

注	Java ES 共有コンポーネントが Release 5 にアップグレードされている場合は、Sun Cluster Geographic Edition ソフトウェアも Release 5 にアップグレードしてください。これは、Sun Cluster Geographic Edition 3.1 8/05 は Release 5 の Sun Java Web Console と互換性がないためです。

2. Sun Cluster: Sun Cluster を Release 5 にアップグレードする手順は、79 ページの第 3 章「Sun Cluster ソフトウェア」で説明されています。

Sun Cluster Geographic Edition のデータのバックアップ

Sun Cluster Geographic Edition はクラスタ構成リポジトリにすべてのデータを格納しているため、現在のデータをバックアップする必要はありません。

必要な設定情報とパスワードの入手

現在インストールされているバージョンに関する特別な情報は必要ありません。ただし、アップグレードを実行するには、スーパーユーザーとしてログインする必要があります。

3.1 8/05 の Sun Cluster Geographic Edition のアップグレード

ここでは、Sun Cluster Geographic Edition のアップグレード手順に影響する注意点を示したあと、その手順自体について説明します。

アップグレードの注意点

Sun Cluster Geographic Edition ソフトウェアは、実行されているアプリケーションに影響を与えることなく Java ES Release 5 にアップグレードできるはずです。動作中のクラスタ上の Sun Cluster Geographic Edition ソフトウェアを停止させずにアップグレードできます。クラスタはサービスが実行されている運用状態を維持します。同様に、Sun Cluster Geographic Edition のパッチも停止時間なしで適用できます。設定データは、アップグレードプロセスをまたいで保持されます。

アップグレードするクラスタとパートナー関係にある全クラスタの全ノードで、Sun Cluster Geographic Edition ソフトウェアをアップグレードしてください。

Sun Cluster Geographic Edition ソフトウェアのアップグレードプロセス中に Solaris オペレーティングシステムをアップグレードする場合は、Solaris オペレーティングシステムをアップグレードする前に Sun Cluster Geographic Edition パッケージを削除してください。

アップグレード手順

ここで説明する手順は、アップグレードを実施するコンピュータにローカルインストールされている Sun Cluster Geographic Edition インスタンスに適用されます。

1. root としてログインするか、スーパーユーザーになります。

   ```
   su -
   ```

2. Sun Cluster Geographic Edition 3.1 8/05 を停止します。

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

   Sun Cluster Geographic Edition の停止中に配下のアプリケーションを実行し続けるには、『Sun Cluster Geographic Edition Administration Guide』(http://docs.sun.com/doc/819-8003) を参照してください。

3. Sun Cluster Geographic Edition 3.1 8/05 をアンインストールします。

   SunClusterGeo-base`/install/uninstall/uninstaller`

   Sun Cluster Geographic Edition Installation Guide』(http://docs.sun.com/doc/819-8004) の説明に従って、Sun Cluster Geographic Edition インストーラをアンインストールモードで使用します。

4. Sun Cluster が実行されていない場合は起動します。

   Sun Cluster Geographic Edition をインストールするには、Sun Cluster が実行中である必要があります。これを確認するには、`scstat` コマンドを使用します。

5. Java ES インストーラを使用して、Release 5 の Sun Cluster Geographic Edition の新規インストールを実行します。

   a. Release 4 の Access Manager がホストされたコンピュータ上で Java ES インストーラを起動します。

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

      ここで、*os_arch* は、`Solaris_sparc` などのプラットフォームに対応します。コマンド行インタフェースには、`installer -nodisplay` オプションを使用します。

      ようこそページと、ライセンス使用許諾契約ページが表示されてから、コンポーネント選択ページが表示されます。Java ES インストーラが直接アップグレードできるインストール済みのコンポーネントが検出された場合、それらのコンポーネントは「アップグレード可」の状態で表示されます。

b. コンポーネント選択ページから Sun Cluster Geographic Edition を選択します。

c. 「今すぐ設定」または「あとで設定」を選択します。

再設定は必要ないため、「今すぐ設定」と「あとで設定」のどちらを選択しても違いはありません。

d. 必要に応じて、ローカライズ版のパッケージをインストールするオプションを選択します。

e. インストールが完了したら、Java ES インストーラを終了します。

注 Solaris 8 OS 上で Sun Cluster Geographic Edition ソフトウェアをアップグレードする場合は、pkgadd コマンドを使用して Java ES Release 5 ディストリビューションからインストールしてください。Java ES インストーラは、Solaris 8 OS 上での Sun Cluster Geographic Edition ソフトウェアのインストールをサポートしていません。

---

6. Sun Cluster Geographic Edition の必要なパッチをすべてインストールします。

手順については、『Sun Cluster Geographic Edition Installation Guide』(http://docs.sun.com/doc/819-8004) を参照してください。

7. このクラスタとパートナー関係にあるすべてのクラスタ上で、99 ページの手順 1 ～手順 6 を実行します。

8. Release 5 の Sun Cluster Geographic Edition ソフトウェアを再起動します。

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

クラスタがパートナー関係にある場合は、Sun Cluster Geographic Edition ソフトウェアを起動する前に、両方のパートナーのすべてのノードをアップグレードしてください。

詳細については、上記の『Installation Guide』および『Sun Cluster Geographic Edition Administration Guide』(http://docs.sun.com/doc/819-8003) を参照してください。

9. 97 ページの「アップグレードのためのクラスタの準備」 の説明に従って、クラスタのアップグレードの準備段階で削除したすべてのアプリケーションリソースグループを追加します。

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

10. 追加したすべての保護グループを開始します。

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

アップグレードの確認

Sun Cluster Geographic Edition ソフトウェアが正常にアップグレードされたかどうかは、次の手順に従って確認できます。

1. 次のコマンドを実行します。

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

   出力値については、96 ページの表 4-4 を参照してください。

2. 該当する場合は、両方のパートナーをアップグレードしたあと、各パートナーのいずれかのクラスタノード上で次のコマンドを実行します。

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

   このコマンドは、Sun Cluster Geographic Edition ソフトウェアがそのノードでアクティブであるかどうかを示します。

アップグレード後の作業

99 ページの「アップグレード手順」で説明している手順以外に、アップグレード後の作業はありません。

アップグレードのロールバック

Sun Cluster Geographic Edition ソフトウェアのロールバックはサポートされていません。

ただし、Java ES インストーラのアンインストール機能を使用して Release 5 の Sun Cluster Geographic Edition を削除してから、Version 3.1 8/05 のインストーラを使用して Version 3.1 8/05 を再インストールすることはできます。Release 5 の Sun Cluster Geographic Edition は Version 3.1 8/05 が読み取れないデータを書き込むため、このようなロールバックは、99 ページの「アップグレード手順」の説明に従って Release 5 を起動する前に (つまり、アップグレードされた Sun Cluster Geographic Edition ソフトウェアを起動する前に) 実行してください。

第 5 章

Directory Server

この章では、Directory Server を Java ES 5 (Release 5): Sun Java System Directory Server 6.0 にアップグレードする方法について説明します。

ここでは、Release 5 でサポートされている各種のアップグレードパスでのアップグレードの注意点の概要について説明します。また、Solaris オペレーティングシステムと Linux オペレーティングシステムの両方でのアップグレードを取り上げます。

- 104 ページの「Directory Server のアップグレードの概要」
- 109 ページの「Java ES Release 4 からの Directory Server のアップグレード」
- 117 ページの「Java ES Release 3 からの Directory Server のアップグレード」
- 117 ページの「Java ES Release 2 からの Directory Server のアップグレード」

注 この章で説明するファイルの場所は、*serverRoot* (Directory Server 5.*x*) と *DirServer-base* (Directory Server 6.0) というディレクトリパスを基準に指定されています。これらのパスの少なくとも一部は、Directory Server がインストールされたときにインストールディレクトリとして指定されている場合があります。指定されていない場合、Java ES インストーラによってデフォルト値が割り当てられています。

次の表に、これらのディレクトリパスのデフォルト値を示します。

表 5-1 Directory Server ディレクトリパス

パス名変数	Solaris OS	Linux OS
serverRoot (Directory Server 5.*x*)	`/var/opt/mps/serverroot`	`/var/opt/sun/directory-server`
DirServer-base (Directory Server 6.0)	`/opt/SUNWdsee`	`/opt/sun`

Directory Server のアップグレードの概要

ここでは、Java ES 5 (Release 5) へのアップグレードに影響する Directory Server の一般的な側面について説明します。

- Java ES Release 5 について
- Java ES Release 5 へのアップグレードロードマップ
- Directory Server のデータ
- Directory Server のアップグレード方針

Java ES Release 5 について

Java ES Release 5 Directory Server はメジャーリリースであり、各種の新機能と改善が追加されています。詳細は、『Directory Server Enterprise Edition 6.0 リリースノート』(http://docs.sun.com/doc/820-0287?l=ja) を参照してください。

Java ES Release 5 へのアップグレードロードマップ

表 5-2 は、Directory Server を Java ES Release 5 へアップグレードする際の、サポートされるアップグレードパスを示しています。この表は Solaris および Linux の両方のオペレーティングシステムに適用されます。

表 5-2　Java ES 5 (Release 5): Directory Server 6.0 へのアップグレードパス

Java ES のリリース	Directory Server のバージョン	一般的な手順	必要な再設定
Release 4	Sun Java System Directory Server 5.2 2005Q4 Sun Java System Administration Server 5.2 2005Q4	直接アップグレード : 新規インストールとすべてのデータの移行。 Administration Server の機能は、Directory Service Control Center と Directory Server EE コマンド行ユーティリティーに置き換えられます。	以前のバージョンから新しくインストールされた Directory Server に移行された設定データ

表 5-2　Java ES 5 (Release 5): Directory Server 6.0 へのアップグレードパス (続き)

Java ES のリリース	Directory Server のバージョン	一般的な手順	必要な再設定
Release 3	Sun Java System Directory Server 5 2005Q1 Sun Java System Administration Server 5 2005Q1	直接アップグレード : 新規インストールとすべてのデータの移行。 Administration Server の機能は、Directory Service Control Center と Directory Server EE コマンド行ユーティリティーに置き換えられます。	以前のバージョンから新しくインストールされた Directory Server に移行された設定データ
Release 2	Sun Java System Directory Server 5.2 2004Q2 Sun Java System Administration Server 5.2 2004Q2	直接アップグレード : 新規インストールとすべてのデータの移行。 Administration Server の機能は、Directory Service Control Center と Directory Server EE コマンド行ユーティリティーに置き換えられます。	以前のバージョンから新しくインストールされた Directory Server に移行された設定データ
Release 1	Sun ONE Directory Server 5.2 Sun ONE Administration Server 5.2	直接アップグレードは動作保証なし : ただし、Release 2 からのアップグレードと同じ手順を使用できます。	以前のバージョンから新しくインストールされた Directory Server に移行された設定データ
それ以前の Java ES リリース	Sun ONE Directory Server 5.2 Sun ONE Administration Server 5.2	直接アップグレードは動作保証なし : ただし、Release 2 からのアップグレードと同じ手順を使用できます。	以前のバージョンから新しくインストールされた Directory Server に移行された設定データ
	Sun ONE Directory Server 5.1 Sun ONE Administration Server 5.1	直接アップグレード不可 : まず Release 3 にアップグレードします。手順については、『Sun Java Enterprise System 2005Q1 アップグレードと移行』(http://docs.sun.com/source/819-2235?l=ja) を参照してください。 次に、Release 3 から Release 5 にアップグレードします。	手順については、『Sun Java Enterprise System 2005Q1 アップグレードと移行』(http://docs.sun.com/source/819-2235?l=ja) を参照してください。

Directory Server のデータ

Directory Server 5.*x* のバージョンでは、設定データの格納に Directory Server 自体を利用していました。データはディレクトリ内部の特定のツリー構造に格納されていました。設定を格納している Directory Server インスタンスのことを設定ディレクトリと呼んでいました。設定ディレクトリは、ほかの Directory Server インスタンスと同様、同じコンピュータ上に配置できました。ただし、ほとんどの配備アーキテクチャーでは、設定ディレクトリと、設定ディレクトリを使用して設定情報を格納するほかのコンポーネントは別々のコンピュータに配置していました。

Directory Server 6.0 では、設定ディレクトリに設定データを格納しなくなりました。設定は Directory Service Control Center (または Directory Server EE コマンド行ユーティリティー) を使用して行うため、このインタフェースを介して設定データにアクセスするようにしてください。Directory Service Control Center は、独自のローカル Directory Server インスタンスに設定データを格納します。

次の表は、Directory Server ソフトウェアを Release 5 にアップグレードすることによって影響を受けるデータの種類を示しています。

表 5-3 Directory Server のデータの使用法

データの種類	場所	使用法
Directory Server 設定データ	Directory Server 5.1 および 5.2: 設定ディレクトリ Directory Server 6.0: Directory Service Control Center および Directory Server EE コマンド行ユーティリティーを介してアクセスします	Directory Server インスタンスの設定
Directory Server スキーマ		ディレクトリ内のデータの構造とセマンティクスを定義します
セキュリティーデータ	Directory Server 5.1 および 5.2: Directory Server Console を介して SSL が設定されます。 Directory Server 6.0: Directory Service Control Center および Directory Server EE コマンド行ユーティリティーを介して SSL が設定されます。	サーバー証明書。
ユーザーデータ	Directory Server	ユーザー固有のデータやユーザープロファイルを持つアプリケーションをサポートします

Directory Server のアップグレード方針

Directory Server をアップグレードするための方針は一般に、第 1 章「アップグレードの計画」で説明されている多くの考慮事項 (アップグレードパス、Java ES コンポーネント間の依存関係、選択的なアップグレードかすべてのアップグレードか、複数インスタンスの配備など) によって異なります。

ここでは、Directory Server のアップグレード計画に影響する可能性のある問題を取り上げ、Directory Server の全般的な説明を詳細に記述します。

互換性の問題

Java ES Release 5 の Directory Server は、新しい公開インタフェースを導入していないため、以前のバージョンと下位互換性があり、Release 4 およびそれ以前の Directory Server がサポートするすべてのコンポーネントをサポートします。

ただし、Release 5 では非公開の管理インタフェースに変更が加えられています。Release 5 のインタフェースは、以前のバージョンの Directory Server と互換性がありません。具体的には、従来の Directory Server インスタンスを設定するために使用されていた Administration Server は Directory Service Control Center と Directory Server EE コマンド行ユーティリティーに置き換えられ、Directory Server の設定情報を格納するための `o=NetscapeRoot` ディレクトリサフィックスは廃止されました。詳細は、『Directory Server Enterprise Edition 6 Migration Guide』(`http://docs.sun.com/doc/819-0994`) に記載されています。

依存関係

一般的には、ほかの Java ES コンポーネントに対する依存関係によって、Directory Server ソフトウェアのアップグレード手順が影響を受ける場合があります。

Directory Server は、次の Java ES コンポーネントに依存します。

- **共有コンポーネント** : Directory Server は、特定の Java ES 共有コンポーネントに依存します (48 ページの表 1-9 を参照)。Directory Server をアップグレードするために、これらの共有コンポーネントのバージョンをアップグレードしなければならない場合があります。
- Directory Proxy Server: Directory Server は、LDAP 要求のセキュリティーとパフォーマンスを向上させるため、Directory Proxy Server に対して共依存関係を持っています。

二重のアップグレード

44 ページの「二重のアップグレード : Java ES ソフトウェアとオペレーティングシステムソフトウェア」で説明されているように、二重のアップグレードでは Directory Server とオペレーティングシステムの両方をアップグレードします。次の 2 つのどちらかの方法で二重のアップグレードを実行できます。

- 新規オペレーティングシステムのインストール
- オペレーティングシステムのインプレースアップグレード

新規オペレーティングシステムのインストール

1. 既存の Directory Server データをバックアップします。

   Directory Server 5.*x* の情報については、111 ページの「Directory Server イメージの作成 (省略可能)」を参照してください。

2. 新しいオペレーティングシステムをインストールします。

   新しいシステム (または Solaris 10 ゾーン) でオペレーティングシステムのインストールを実行するか、または、既存のファイルシステムを消去することができます。

3. 手順 1 でバックアップした Directory Server データを復元します。

4. Release 5 の Directory Server をインストールします。

5. Release 5 の Directory Server インスタンスを作成し、ディレクトリデータを新しいインスタンスに移行します。

   111 ページの「Release 4 の Directory Server のアップグレード」に記載されている手順の該当する節を参照してください。

オペレーティングシステムのインプレースアップグレード

1. 既存の Directory Server データをバックアップします。

   Directory Server 5.*x* の情報については、111 ページの「Directory Server イメージの作成 (省略可能)」を参照してください。

2. オペレーティングシステムをアップグレードします。

   このアップグレードでは、既存のファイルシステムはそのまま残されます。

3. Release 5 の Directory Server にアップグレードします。

   アップグレードパスに応じて、この章の該当する節を参照してください。

Java ES Release 4 からの Directory Server のアップグレード

この節では、Directory Server の Java ES 2005 Q4 (Release 4) から Java ES 5 (Release 5) へのアップグレードについて説明します。この節で説明する内容は、次のとおりです。

- はじめに
- Release 4 の Directory Server のアップグレード
- 複数インスタンスのアップグレード

はじめに

Java ES Release 4 Directory Server を Release 5 にアップグレードする場合は、アップグレード処理の次の側面を考慮してください。

- **一般的なアップグレード手順** : アップグレードは、Release 5 の Directory Server の新規インストールを実行し、移行ツールを使用して以前の Directory Server インスタンスを新しい別個の Release 5 の Directory Server インスタンスで再作成することによって行われます。
- **アップグレードの依存関係** : Directory Server は、いくつかの Java ES 共有コンポーネントに依存しています (48 ページの表 1-9 を参照)。Directory Server のアップグレードを実行すると、すべての共有コンポーネントは Java ES インストーラによって自動的に Release 5 にアップグレードされます。
- **下位互換性** : 107 ページの「互換性の問題」 で説明しているように、Release 5 の Directory Server は、いくつかの点で、Release 4 バージョンとの下位互換性がありません。ただし、Directory Server 5.*x* の管理フレームワークに依存する一連の Directory Server インスタンスを引き続き保持する場合は、移行ツールを使用して o=NescapeRoot サフィックスを移行できます。
- **アップグレードのロールバック** : Release 5 アップグレードのロールバックは、アップグレード時に元のまま保持されていた以前のバージョンを復元することによって行われます。
- **プラットフォームの問題** : Directory Server をアップグレードするための一般的な手順は、Solaris および Linux の両オペレーティングシステムで共通です。

Release 4 の Directory Server のアップグレード

ここでは、Solaris プラットフォームと Linux プラットフォームの両方で、Directory Server の Java ES Release 4 から Java ES Release 5 へのアップグレードを実行する方法について説明します。プラットフォーム固有の手順に依存する項目には、その手順が適用されるオペレーティングシステムが示されています。この節で説明する内容は、次のとおりです。

- アップグレード前の作業
- Release 4 の Directory Server のアップグレード
- アップグレードの確認
- アップグレード後の作業
- アップグレードのロールバック

アップグレード前の作業

Directory Server ソフトウェアをアップグレードする前に、次の作業を実行してください。

- 現在のバージョン情報の確認
- Directory Server が依存するコンポーネントのアップグレード
- 必要な設定情報とパスワードの入手
- Directory Server イメージの作成 (省略可能)

現在のバージョン情報の確認

Directory Server の現在のバージョンは、次のように -v オプションを使用して Directory Server デーモンを再起動することによって確認できます。

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

表 5-4　Directory Server のバージョン確認の出力

Java ES のリリース	Directory Server のバージョン番号
Release 2	`Sun Java(TM) System Directory Server/5.2_Patch_2`
Release 3	`Sun Java(TM) System Directory Server/5.2_Patch_3`
Release 4	`Sun Java(TM) System Directory Server/5.2_Patch_4`
Release 5	`Sun Java(TM) System Directory Server/6.0`

Solaris 10 プラットフォームで ns-slapd コマンドが失敗した場合は、コマンド実行時にライブラリパスを NULL に設定してください。

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

Directory Server が依存するコンポーネントのアップグレード

一般的には、コンピュータシステム上 (および、コンピューティング環境内) のすべての Java ES コンポーネントを Java ES Release 5 にアップグレードすることが推奨されます。ただし、Directory Server の Release 5 へのアップグレードを実行すると、Directory Server で必要な共有コンポーネントはすべて、自動的にアップグレードされます。

必要な設定情報とパスワードの入手

現在インストールされているバージョンの Directory Server の管理者ユーザー ID とパスワードを調べておきます。その他の設定情報は、アップグレードプロセスによって保持されます。

Directory Server イメージの作成 (省略可能)

Release 5 の Directory Server を Release 4 バージョンが存在するコンピュータとは別のコンピュータにインストールしている場合は、Release 5 Directory Server をインストールしているコンピュータ上で Release 4 バージョンのイメージを作成するようにしてください。このイメージは、(dsmig コマンドを使用した) 新しい Release 5 の Directory Server インスタンスへのデータ移行を自動化するために必要です。

Release 4 のイメージには、すべてのスキーマファイル、設定ファイル、セキュリティーファイル、およびデータベースファイルが、元の Directory Server 5.x の *serverRoot* ファイル構造と同じ配置で含まれています。このイメージは、新しい Release 5 の Directory Server インスタンスへのデータ移行を実行するために必要です。

Release 4 の Directory Server のアップグレード

ここでは、Directory Server のアップグレード手順に影響する注意点を示したあと、その手順自体について説明します。

アップグレードの注意点

Directory Server ソフトウェアの Java ES Release 5 へのアップグレードでは、次の注意点を考慮に入れてください。

- Directory Server インスタンスを使用する Java ES すべてのコンポーネント (Access Manager、Portal Server、Sun Java Communications Suite コンポーネントなど) をシャットダウンして、対応する新しい Release 5 インスタンスにアクセスするように再設定してください。

- Directory Server の複数のインスタンスが 1 台のコンピュータ上で稼働しており、インストールされた同じ Directory Server イメージにそれらすべてのインスタンスが対応しているような配備アーキテクチャーでは、Directory Server イメージのアップグレードは 1 回で済みますが、データはインスタンスごとに個別に移行する必要があります。
- Release 4 の Directory Server の多くの配備アーキテクチャーでは、設定ディレクトリは独立した Directory Server インスタンスです。Release 5 では設定ディレクトリが非推奨になったため、これらのインスタンスをアップグレードする必要はありません。一方、アップグレードの際に、Directory Server インスタンスをリモート管理するための別のコンピュータに Release 5 の Directory Server の管理コンソール (Directory Service Control Center) を配備しなければならない場合があります。
- Directory Server には、スキーマデータ、設定データ、セキュリティーデータ、およびユーザーデータの移行を自動化するのに役立つコマンド行ツールが用意されています。この移行ツールを使用すると、これらの各種データを段階的に移行できます。ほとんどのアップグレードシナリオでは、データの一部の移行を自動化するだけでもメリットがあります。

アップグレード手順

次に示す手順は、アップグレードを実施するコンピュータにローカルにインストールされている Directory Server インスタンスに適用されます。また、インスタンスを別のコンピュータに移動する場合は、移動先のコンピュータで実行されるすべてのインスタンスに適用されます。

1. root としてログインするか、スーパーユーザーになります。

   ```
   su -
   ```

2. Release 4 の Directory Server (5.2) インスタンスをシャットダウンします。

   serverRoot/slapd-*instanceName*/stop-slapd

   エラーログ (*serverRoot*/slapd-*hostName*/logs/errors) にシャットダウンが正常に行われた記録があることを確認します。

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

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

3. Release 5 の Directory Server のホストコンピュータに十分なディスク容量があることを確認します。

   基本的な計算は、次のとおりです。
   2 * (既存のサーバー用の容量) + (LDIF ファイル用の容量)

残念ながら、エクスポートされたデータベースから作成される LDIF ファイルのサイズを予想できるツールはありません。このサイズは、データエントリの数、データエントリの内部表現、インデックスの数などによって異なります。

4. Release 5 のリモートインストールの場合は、Release 4 のイメージを作成し、それをリモートコンピュータに転送します。

   111 ページの「Directory Server イメージの作成 (省略可能)」を参照してください。

5. Directory Server が強いアップグレード依存関係を持つ Java ES コンポーネント (111 ページの「Directory Server が依存するコンポーネントのアップグレード」を参照) がすべてアップグレード済みであることを確認します。

6. Release 5 の Directory Server の新規インストールを実行します。

   次の手順を実行します。

   a. Java ES インストーラを起動します。

   ```
   cd Java ES Release 5 distribution/os_arch

   ./installer
   ```

   ここで、*os_arch* は、`Solaris_sparc` などのプラットフォームに対応します。コマンド行インタフェースには、`installer -nodisplay` オプションを使用します。

   ようこそページと、ライセンス使用許諾契約ページが表示されてから、コンポーネント選択ページが表示されます。Java ES インストーラが直接アップグレードできるインストール済みのコンポーネントが検出された場合、それらのコンポーネントは「アップグレード可」の状態で表示されます。

   b. Directory Server Enterprise Edition の Directory Server サブコンポーネントを選択します。

   使用する管理サブコンポーネント (Directry Service Control Center やコマンド行ユーティリティー) もインストールする必要があります。

   c. 既存の Release 4 の Directory Server とは異なるインストールパスを指定します。

   d. 「今すぐ設定」または「あとで設定」を選択します。

   Directory Server に必要な設定は実際には存在しないため、「今すぐ設定」と「あとで設定」のどちらを選択しても問題ありません。ただし、「今すぐ設定」を選択した場合は、新しいインスタンスの作成を選択しないでください。

e. インストールに関して選択した項目を確認します。

Directory Server パッケージがアップグレードされ、アップグレードの概要が表示されます。

f. Java ES インストーラを終了します。

7. Directory Server インスタンスを作成します。

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

ここで、*instancePath* は Directory Server インスタンスへのフルパスです。

Directory Server インスタンスの作成については、『Directory Server Enterprise Edition 6.0 管理ガイド』(http://docs.sun.com/doc/820-0291?l=ja) を参照してください。

新しいインスタンスの作成に失敗した場合は、dsmig コマンドを使用してデータを移行するとき (手順 8) に新しいインスタンスが自動的に作成されます。

Solaris 10 プラットフォームで dsadm コマンドが失敗した場合は、コマンド実行時にライブラリパスを NULL に設定してください。

LD_LIBRARY_PATH= ./dsadm create *instancePath*

8. Release 4 のデータを Release 5 の Directory Server インスタンスに移行します。

DirServer-base/ds6/bin/dsmig コマンドを使用します。

dsmig コマンドは、Release 4 のデータを Release 5 の形式に適応させ、適切な場所に書き込みます。たとえば、単一のコンピュータに 1 つの Directory Server インスタンスがある場合の一般的な移行は、次のようになる可能性があります。

```
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

- Solaris 10 プラットフォームで dsmig migrate-config コマンドが失敗した場合は、コマンド実行時にライブラリパスを NULL に設定してください。

  ```
  LD_LIBRARY_PATH= ./dsmig migrate-config ...
  ```

- 移行する Directory Server インスタンスにほかの Java ES コンポーネント (たとえば、Sun Java Communications Suite Messaging Server コンポーネント) の設定データが格納されている場合は、o=netscaperoot という名前の、ディレクトリ情報ツリーの特定の部分を移行しなければならないこともあります。このルートサフィックスは、デフォルトでは移行されません。o=netscaperoot を移行するには、dsmig migrate-config コマンドと dsmig migrate-data コマンドの -N オプションを使用します。次に例を示します。

  ```
  dsmig migrate-config -v old_instancePath
      new_instancePath -N
  ```

- 32 ビットアーキテクチャー上のインスタンスから 64 ビットアーキテクチャー上のインスタンスへ移行する場合は、dsmig migrate-data コマンド (自動移行ツール) を使用できません。下記の『Migration Guide』の説明に従って、データを手動で移行する必要があります。ただし、スキーマデータ、設定データ、およびセキュリティーデータについては、自動的な移行を実行できます。
- ディレクトリデータの移行後に Directory Server を起動すると、新しい Release 5 のエラーチェックによって Directory Server のグループエントリ内に循環定義が検出される場合があります。これらの循環定義によって、機能上の問題は発生しませんが、エラーファイルに大量のエラーが記録されることがあります。

移行プロセス、dsmig コマンド、および手動による移行の詳細については、『Directory Server Enterprise Edition 6 Migration Guide』(http://docs.sun.com/doc/819-0994) を参照してください。

アップグレードの確認

Directory Server が正常にアップグレードされたかどうかは、次のようにして確認できます。

1. 新しい Directory Server インスタンスを起動します。

   DirServer-base/ds6/bin/dsadm -V

   出力値については、110 ページの表 5-4 を参照してください。

2. Directory Server のエラーログ内の起動メッセージを確認します。

instancePath/logs/errors

注	Release 5 の Directory Server は、起動時に循環定義を検出します。これらの循環定義によって、機能上の問題は発生しませんが、このような循環定義を含む以前のバージョンからアップグレードした場合は、エラーファイルに大量のエラーが記録されることがあります。

アップグレード後の作業

112 ページの「アップグレード手順」で説明している手順以外にアップグレード後の作業はありませんが、Directory Server に依存するすべての Java ES コンポーネントを、新しい Directory Server インスタンスをポイントするように再設定する必要があります。

アップグレードのロールバック

Release 5 アップグレードのロールバックは、Release 5 へのアップグレード時に未変更のまま残されていた以前のバージョンを復元することによって行われます。

複数インスタンスのアップグレード

110 ページの「Release 4 の Directory Server のアップグレード」の手順は、可用性またはスケーラビリティーのために Directory Server がレプリケートされる配備アーキテクチャーには対応していません。これらのアーキテクチャーには、Directory Server レプリケーションや、Sun Cluster 環境におけるデータサービスとしての Directory Server の配備が含まれる場合があります。

Directory Server レプリケーションの順次アップグレード

マルチマスターレプリケーション配備アーキテクチャーで使用されるような、複数のコンピュータシステム上に存在する複数の Directory Server インスタンスについては、一度に 1 つずつ連続してアップグレードできます。最初にすべての Directory Server マスターを同期したあと、ほかのインスタンスを稼働させたままで、各ホストコンピュータ上の各インスタンスをアップグレードします。これは順次アップグレードと呼ばれる仕組みであり、ディレクトリサービスをオンラインに保ったまま、サービスを提供する個別の Directory Server インスタンスを順次アップグレードすることができます。

データサービスとしての Directory Server のアップグレード

Sun Cluster 環境におけるデータサービスとしての Directory Server のアップグレードとロールバックに関する情報は、現在開発中です。

Java ES Release 3 からの Directory Server のアップグレード

Java ES 2003Q1 (Release 3) Directory Server を Release 5 にアップグレードする手順は、Release 4 Directory Server を Release 5 にアップグレードする場合と同じです。

Release 3 Directory Server を Release 5 にアップグレードするには、「Release 4」を「Release 3」と読み替えた上で、109 ページの「Java ES Release 4 からの Directory Server のアップグレード」の手順を使用してください。

Java ES Release 2 からの Directory Server のアップグレード

Java ES 2004Q2 (Release 2) Directory Server を Release 5 にアップグレードする手順は、Release 4 Directory Server を Release 5 にアップグレードする場合と同じです。ただし、アップグレードの前に、すべての共有コンポーネント (48 ページの表 1-9 を参照) を Release 5 へアップグレードする必要があります。

Java ES 共有コンポーネントを Release 5 にアップグレードする手順は、67 ページの第 2 章「Java ES 共有コンポーネントのアップグレード」で説明されています。

Release 2 Directory Server を Release 5 にアップグレードするには、「Release 4」を「Release 2」と読み替えた上で、109 ページの「Java ES Release 4 からの Directory Server のアップグレード」の手順を使用してください。

注　Linux プラットフォームで Release 2 の Directory Server からアップグレードする場合は、二重のアップグレード、つまり、Directory Server とオペレーティングシステムの両方をアップグレードする必要があります (Release 5 の Directory Server は RHEL 2.1 でサポートされていない)。詳細については、108 ページの「二重のアップグレード」を参照してください。

第 6 章

Directory Proxy Server

この章では、Directory Proxy Server を Java ES 5 (Release 5): Sun Java System Directory Proxy Server 6.0 にアップグレードする方法について説明します。

ここでは、Release 5 でサポートされている各種のアップグレードパスでのアップグレードの注意点の概要について説明します。また、Solaris オペレーティングシステムと Linux オペレーティングシステムの両方でのアップグレードを取り上げます。

- 120 ページの「Directory Proxy Server のアップグレードの概要」
- 124 ページの「Java ES Release 4 からの Directory Proxy Server のアップグレード」
- 130 ページの「Java ES Release 3 からの Directory Proxy Server のアップグレード」
- 131 ページの「Java ES Release 2 からの Directory Proxy Server のアップグレード」

注 この章で説明するファイルの場所は、*serverRoot* (Directory Proxy Server 5.*x*) と *DirServer-base* (Directory Proxy Server 6.0) というディレクトリパスを基準に指定されています。これらのパスの少なくとも一部は、Directory Proxy Server がインストールされたときにインストールディレクトリとして指定されている場合があります。指定されていない場合、Java ES インストーラによってデフォルト値が割り当てられています。

次の表に、これらのディレクトリパスのデフォルト値を示します。

表 6-1 Directory Proxy Server ディレクトリパス

パス名変数	Solaris OS	Linux OS
serverRoot (Directory Proxy Server 5.*x*)	`/var/opt/mps/serverroot`	`/var/opt/sun/directory-server`

表 6-1　Directory Proxy Server ディレクトリパス (続き)

パス名変数	Solaris OS	Linux OS
DirServer-base (Directory Proxy Server 6.0)	/opt/SUNWdsee	/opt/sun

Directory Proxy Server のアップグレードの概要

ここでは、Java ES 5 (Release 5) へのアップグレードに影響する Directory Proxy Server の一般的な側面について説明します。

- Java ES Release 5 について
- Java ES Release 5 へのアップグレードロードマップ
- Directory Proxy Server のデータ
- Directory Proxy Server のアップグレード方針

Java ES Release 5 について

Java ES Release 5 Directory Proxy Server はメジャーリリースであり、Release 4 以前のすべての Directory Proxy Server リリースに対する新規の製品です。

Release 5 の Directory Proxy Server は、これまでどおり LDAP プロキシですが、新しい拡張可能なルーティング機能を備えています。Release 5 では、複数のデータビューを 1 つのビューに集約する仮想ディレクトリ機能も使用可能です。これらのデータビューは、LDAP または SQL でアクセス可能なデータストアを表すことができます。

詳細については、『Directory Server Enterprise Edition 6 リリースノート』を参照してください。

Java ES Release 5 へのアップグレードロードマップ

表 6-2 は、Directory Proxy Server を Java ES Release 5 へアップグレードする際の、サポートされるアップグレードパスを示しています。この表は Solaris および Linux の両方のオペレーティングシステムに適用されます。

表 6-2　Java ES 5 (Release 5): Directory Proxy Server 6.0 へのアップグレードパス

Java ES のリリース	Directory Proxy Server のバージョン	一般的な手順	必要な再設定
Release 4	Sun Java System Directory Proxy Server 5.2 2005Q4	直接アップグレード : Release 5 の新規インストールと設定によって、Release 4 を置き換えます。	下位互換性が必要な場合は、以前の設定を新しい設定プロパティーに手動で対応付けます。
Release 3	Sun Java System Directory Proxy Server 5.2 2005Q1	直接アップグレード : Release 5 の新規インストールと設定によって、Release 3 を置き換えます。	下位互換性が必要な場合は、以前の設定を新しい設定プロパティーに手動で対応付けます。
Release 2	Sun Java System Directory Proxy Server 5.2 2004Q2	直接アップグレード : Release 5 の新規インストールと設定によって、Release 2 を置き換えます。	下位互換性が必要な場合は、以前の設定を新しい設定プロパティーに手動で対応付けます。
Release 1	Sun ONE Directory Proxy Server 5.2	直接アップグレードは動作保証なし : ただし、Release 2 からのアップグレードと同じ手順を使用できます。	下位互換性が必要な場合は、以前の設定を新しい設定プロパティーに手動で対応付けます。
それ以前の Java ES リリース	Sun ONE Directory Proxy Server 5.2	直接アップグレードは動作保証なし : ただし、Release 2 からのアップグレードと同じ手順を使用できます。	下位互換性が必要な場合は、以前の設定を新しい設定プロパティーに手動で対応付けます。
	Sun One Directory Access Router 5.0 または 5.0 SP1	直接アップグレード不可 : まず Release 3 にアップグレードします。手順については、『Java Enterprise System 2005Q1 アップグレードと移行』(http://docs.sun.com/source/819-2235?l=ja) を参照してください。 次に、Release 3 から Release 5 にアップグレードします。	手順については、『Sun Java Enterprise System 2005Q1 アップグレードと移行』(http://docs.sun.com/source/819-2235?l=ja) を参照してください。

Directory Proxy Server のデータ

Directory Proxy Server は、設定データの格納に Directory Server を利用しなくなりました。設定は、新しい Directory Service Control Center または Directory Server EE コマンド行ユーティリティーを使用して行います。

次の表は、Directory Proxy Server ソフトウェアのアップグレードにより影響を受ける可能性のあるデータの種類を示しています。

表 6-3 Directory Proxy Server のデータの使用法

データの種類	場所	使用法
Directory Proxy Server 設定データ	Directory Proxy Server 5.2 の設定ディレクトリ Directory Proxy Server 6.0: Directory Service Control Center および Directory Server EE コマンド行ユーティリティーを介してアクセスします。	Directory Proxy Server の設定
セキュリティーデータ	Directory Proxy Server 5.2: Directory Proxy Server Console を介して SSL が設定されます。 Directory Proxy Server 6.0: Directory Service Control Center および Directory Server EE コマンド行ユーティリティーを介して SSL が設定されます。	サーバー証明書

Directory Proxy Server のアップグレード方針

Directory Proxy Server をアップグレードするための方針は一般に、第 1 章「アップグレードの計画」で説明されている多くの考慮事項 (アップグレードパス、Java ES コンポーネント間の依存関係、選択的なアップグレードかすべてのアップグレードか、複数インスタンスの配備など) によって異なります。

ここでは、Directory Proxy Server のアップグレード計画に影響する可能性のある問題を取り上げ、Directory Proxy Server の全般的な説明を詳細に記述します。

互換性の問題

Release 5 の Directory Proxy Server ではインタフェースの変更があり、以前の Directory Proxy Server リリースとは互換性がありません。Release 5 の Directory Proxy Server は、完全に新しい Java ベースの実装に基づいており、その設定は Release 4 以前の Directory Proxy Server リリースとはまったく異なっています。

ただし、Release 5 の Directory Proxy Server を、下位互換性を持つ (つまり、Release 4 以前の Directory Proxy Server リリースのように動作する) ように設定することは可能です。このように設定するには、以前の設定属性を Release 5 の設定プロパティーに手動で対応付ける必要があります。詳細は、『Directory Server Enterprise Edition 6 Migration Guide』(http://docs.sun.com/doc/819-0994) に記載されています。

ただし、Release 5 の Directory Proxy Server のデフォルトの動作は以前のバージョンとは異なり、LDAP 制御がプロキシを通過することはできません。以前のバージョンの動作を再現するには、129 ページの「アップグレード後の作業」に説明されているように、これらの制御のブロックを解除します。

依存関係

ほかの Java ES コンポーネントに対する依存関係によって、Directory Proxy Server ソフトウェアのアップグレードおよび再設定の手順が影響を受ける場合があります。

Directory Proxy Server は、次の Java ES コンポーネントに依存します。

- **共有コンポーネント** : Directory Proxy Server は、特定の Java ES 共有コンポーネントに依存します (48 ページの表 1-9 を参照)。Directory Proxy Server をアップグレードするために、これらの共有コンポーネントのバージョンをアップグレードしなければならない場合があります。
- Directory Server: Directory Proxy Server は、LDAP 要求のセキュリティーとパフォーマンスを向上させるため、Directory Server に対して共依存関係を持っています。Directory Proxy Server は、Directory Server へのフロントエンドアクセスを提供しますが、この機能的関係を超える依存関係を Directory Server に対して持っていません。

二重のアップグレード

44 ページの「二重のアップグレード : Java ES ソフトウェアとオペレーティングシステムソフトウェア」で説明されているように、二重のアップグレードでは Directory Proxy Server とオペレーティングシステムの両方をアップグレードします。次の 2 つのどちらかの方法で二重のアップグレードを実行できます。

- 新規オペレーティングシステムのインストール
- オペレーティングシステムのインプレースアップグレード

新規オペレーティングシステムのインストール

1. 既存の Directory Proxy Server データをバックアップします。

   基本的なデータの場所については、122 ページの「Directory Proxy Server のデータ」を参照してください。

2. 新しいオペレーティングシステムをインストールします。

新しいシステム (または Solaris 10 ゾーン) でオペレーティングシステムのインストールを実行するか、または、既存のファイルシステムを消去することができます。

3. Release 5 の Directory Proxy Server をインストールします。
4. Release 5 の Directory Proxy Server インスタンスを作成し、設定属性を Release 5 の Directory Proxy Server プロパティーに対応付けます。

   127 ページの「Release 4 の Directory Proxy Server のアップグレード」に記載されている手順の該当する節を参照してください。

オペレーティングシステムのインプレースアップグレード

1. 既存の Directory Proxy Server データをバックアップします。

   基本的なデータの場所については、122 ページの「Directory Proxy Server のデータ」を参照してください。

2. オペレーティングシステムをアップグレードします。

   このアップグレードでは、既存のファイルシステムはそのまま残されます。

3. Release 5 の Directory Proxy Server にアップグレードします。

   アップグレードパスに応じて、この章の該当する節を参照してください。

Java ES Release 4 からの Directory Proxy Server のアップグレード

ここでは、Directory Proxy Server の Java ES 2005Q4 (Release 4) から Java ES 5 (Release 5) へのアップグレードについて説明します。この節で説明する内容は、次のとおりです。

- はじめに
- Release 4 の Directory Proxy Server のアップグレード
- 複数インスタンスのアップグレード

はじめに

Java ES Release 4 Directory Proxy Server を Release 5 にアップグレードする場合は、アップグレード処理の次の側面を考慮してください。

- **一般的なアップグレード手順** : アップグレードは、Release 5 の Directory Proxy Server の新規インストールを実行し、Directory Service Control Center または Directory Server EE コマンド行ユーティリティーを使用して新しい Directory Proxy Server インスタンスを設定することによって行われます。
- **アップグレードの依存関係** : Directory Proxy Server は、いくつかの Java ES 共有コンポーネントに依存しています (48 ページの表 1-9 を参照)。Directory Proxy Server のアップグレードを実行すると、すべての共有コンポーネントは Java ES インストーラによって自動的に Release 5 にアップグレードされます。
- **下位互換性** : 122 ページの「互換性の問題」 で説明しているように、Release 5 の Directory Proxy Server は、Release 4 バージョンとの下位互換性を持つように設定できます。
- **アップグレードのロールバック** : Release 5 アップグレードのロールバックは、アップグレード時に元のまま保持されていた以前のバージョンを復元することによって行われます。
- **プラットフォームの問題** : Directory Proxy Server をアップグレードするための一般的な手順は、Solaris および Linux の両オペレーティングシステムで共通です。

Release 4 の Directory Proxy Server のアップグレード

ここでは、Solaris プラットフォームと Linux プラットフォームの両方で、Directory Proxy Server の Java ES Release 4 から Java ES Release 5 へのアップグレードを実行する方法について説明します。プラットフォーム固有の手順に依存する項目には、その手順が適用されるオペレーティングシステムが示されています。この節で説明する内容は、次のとおりです。

- アップグレード前の作業
- Release 4 の Directory Proxy Server のアップグレード
- アップグレードの確認
- アップグレード後の作業
- アップグレードのロールバック

アップグレード前の作業

Directory Proxy Server ソフトウェアをアップグレードする前に、次の作業を実行してください。

- 現在のバージョン情報の確認
- Directory Proxy Server が依存するコンポーネントのアップグレード
- 必要な設定情報とパスワードの入手

現在のバージョン情報の確認

Directory Proxy Server の現在のバージョンは、次のコマンドを使用することにより確認できます。

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

出力の一覧を次の表に示します。

表 6-4　Directory Proxy Server のバージョン確認の出力

Java ES のリリース	Directory Proxy Server のバージョン番号
Release 2	`Sun ONE Directory Proxy Server Version 5.2_Patch_2`
Release 3	`Sun ONE Directory Proxy Server Version 5.2_Patch_3`
Release 4	`Sun ONE Directory Proxy Server Version 5.2_Patch_4`
Release 5[1]	`Sun ONE Directory Proxy Server Version 6.0`

1. `ldapfwd` コマンドを使用して Release 5 のバージョン番号を返すことはできません。129 ページの「アップグレードの確認」を参照してください。

Directory Proxy Server が依存するコンポーネントのアップグレード

Directory Proxy Server が強いアップグレード依存関係を持っている共有コンポーネントはわずかですが、一般的には、コンピュータシステム上 (および、コンピューティング環境内) のすべての Java ES コンポーネントを Java ES Release 5 にアップグレードすることが推奨されます。

Directory Proxy Server が依存するコンポーネントをアップグレードするときは、Directory Proxy Server をアップグレードする前に、(すでにアップグレードした可能性があるものを省いて) 次の順序で実行してください。共有コンポーネントのアップグレードは通常、Java ES インストーラによって自動的に実行されます。

1. **共有コンポーネント** : Java ES 共有コンポーネントを Release 5 に同期する手順は、67 ページの「Java ES 共有コンポーネントのアップグレード」で説明されています。ただし、Directory Proxy Server の Release 5 へのアップグレードを実行すると、Directory Proxy Server で必要な共有コンポーネントはすべて、Java ES インストーラによって自動的にアップグレードされます。
2. **Directory Server (弱いアップグレード依存関係)**: Directory Server を Release 5 にアップグレードする手順は、103 ページの第 5 章「Directory Server」で説明されています。ただし、Release 5 の Directory Proxy Server は Release 4 の Directory Server によってサポートされます。

必要な設定情報とパスワードの入手

設定情報は、アップグレードプロセスによって保持され、Release 4 の設定属性を Release 5 の設定プロパティーに対応付けるために使用できます。122 ページの「互換性の問題」を参照してください。

Release 4 の Directory Proxy Server のアップグレード

ここでは、Directory Proxy Server のアップグレード手順に影響する注意点を示したあと、その手順自体について説明します。

アップグレードの注意点

Directory Proxy Server ソフトウェアの Java ES Release 5 へのアップグレードでは、次の注意点を考慮に入れてください。

- Directory Proxy Server インスタンスを使用する すべての Java ES コンポーネント (Access Manager、Communications Express、Messaging Server、Portal Server など) をシャットダウンして、対応する新しい Release 5 インスタンスにアクセスするように再設定してください。
- Directory Proxy Server の複数のインスタンスが 1 台のコンピュータ上で稼働しており、インストールされた同じ Directory Proxy Server イメージにそれらすべてのインスタンスが対応しているような配備アーキテクチャーでは、Directory Proxy Server イメージをアップグレードするために新しい Directory Proxy Server インスタンスを作成する必要があります。
- Directory Proxy Server を含む Release 4 の配備アーキテクチャーでは、管理サーバーを使用して Directory Proxy Server インスタンスを設定および管理していました。Release 5 では、Directory Proxy Server のアップグレードの際には、Directory Proxy Server インスタンスを設定および管理するために使用する Directory Service Control Center の配備が必要な場合があります。

アップグレード手順

ここで説明する手順は、アップグレードを実施するコンピュータにローカルインストールされている Directory Proxy Server インスタンスに適用されます。

1. root としてログインするか、スーパーユーザーになります。

   ```
   su -
   ```

2. アップグレードする Directory Proxy Server インスタンスに依存するすべての Java ES コンポーネントをシャットダウンします。この手順は、配備アーキテクチャーの内部での Directory Proxy Server のレプリケート形態によって異なる場合があります。

   Java ES コンポーネントのシャットダウン方法については、各コンポーネントの管理ガイドを参照してください。

3. Release 5 の Directory Proxy Server の新規インストールを実行します。

   次の手順を実行します。

   a. Java ES インストーラを起動します。

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

      ここで、*os_arch* は、`Solaris_sparc` などのプラットフォームに対応します。コマンド行インタフェースには、`installer -nodisplay` オプションを使用します。

      ようこそページと、ライセンス使用許諾契約ページが表示されてから、コンポーネント選択ページが表示されます。Java ES インストーラが直接アップグレードできるインストール済みのコンポーネントが検出された場合、それらのコンポーネントは「アップグレード可」の状態で表示されます。

   b. Directory Proxy Server Enterprise Edition の Directory Server サブコンポーネントを選択します。

      使用する管理サブコンポーネント (Directry Service Control Center やコマンド行ユーティリティー) もインストールする必要があります。

   c. 既存の Release 4 の Directory Proxy Server とは異なるインストールパスを指定します。

   d. 「今すぐ設定」または「あとで設定」を選択します。

      Directory Proxy Server に必要な設定は実際には存在しないため、「今すぐ設定」と「あとで設定」のどちらを選択しても問題ありません。ただし、「今すぐ設定」を選択した場合は、新しいインスタンスを作成するかどうかを尋ねられます。この時点で作成しなくても、コンポーネントのインストール後に、Directory Proxy Server インスタンスを必要な数だけ作成できます。

   e. 必要に応じて、ローカライズ版のパッケージをインストールするオプションを選択します。

f. インストールに関して選択した項目を確認します。

Directory Proxy Server パッケージがインストールされ、インストールの概要が表示されます。

g. Java ES インストーラを終了します。

新しいインスタンスをまだ作成していない場合は手順 4 に進み、すでに作成した場合は手順 5 に進みます。

4. Directory Proxy Server インスタンスを作成します。

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

ここで、*instancePath* は Directory Proxy Server インスタンスへのフルパスです。

Directory Proxy Server インスタンスの作成については、『Directory Server Enterprise Edition 6.0 管理ガイド』(http://docs.sun.com/doc/820-0291?l=ja) を参照してください。

5. 必要であれば、Release 4 の設定属性を Release 5 の Directory Proxy Server のプロパティーに対応付けます。

対応付けの手順の詳細については、『Directory Server Enterprise Edition 6 Migration Guide』(http://docs.sun.com/doc/819-0994) を参照してください。

アップグレードの確認

Directory Proxy Server が正常にアップグレードされたかどうかは、次のようにして確認できます。

1. 新しい Directory Proxy Server インスタンスを起動します。

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

2. Directory Proxy Server のバージョンを確認します。

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

出力値は 126 ページの表 6-4 に示したとおりです。

アップグレード後の作業

Directory Proxy Server に依存するすべての Java ES コンポーネントを、新しい Directory Proxy Server インスタンスをポイントするように再設定する必要があります。

また、以前のバージョンのデフォルトの動作を再現するには、LDAP 制御にプロキシの通過を明示的に許可してください。allowed-ldap-controls プロパティーを次のように設定すると、これらの制御を有効にできます。

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
```

アップグレードのロールバック

Release 5 アップグレードのロールバックは、Release 5 へのアップグレード時に未変更のまま残されていた以前のバージョンを復元することによって行われます。

複数インスタンスのアップグレード

配備のアーキテクチャーによっては、スケーラビリティーを実現し、可用性を向上させるために、Directory Proxy Server が複数のコンピュータシステムに配備される場合があります。たとえば、ロードバランサを使用して Directory Proxy Server コンポーネントを複数のコンピュータで実行することにより、負荷を分散させることができます。

Directory Proxy Server のインスタンスが負荷分散されている場合は、順次アップグレードを実行できません。ロードバランサを停止して、Release 5 インスタンスにアクセスするように再設定する必要があります。125 ページの「Release 4 の Directory Proxy Server のアップグレード」の説明に従って、各インスタンスのアップグレードを実行します。

Java ES Release 3 からの Directory Proxy Server のアップグレード

Java ES 2005Q1 (Release 3) Directory Proxy Server を Release 5 にアップグレードする手順は、Release 4 Directory Proxy Server を Release 5 にアップグレードする場合と同じです。

Release 3 Directory Proxy Server を Release 5 にアップグレードするには、「Release 4」を「Release 3」と読み替えた上で、124 ページの「Java ES Release 4 からの Directory Proxy Server のアップグレード」の手順を使用してください。

Java ES Release 2 からの Directory Proxy Server のアップグレード

Java ES 2004Q2 (Release 2) Directory Proxy Server を Release 5 にアップグレードする手順は、Release 4 Directory Proxy Server を Release 5 にアップグレードする場合と同じです。ただし、アップグレードの前に、すべての共有コンポーネント (48 ページの表 1-9 を参照) と、Directory Proxy Server が依存するローカルにあるすべての製品コンポーネントを Release 5 へアップグレードする必要があります。

Java ES 共有コンポーネントを Release 5 に同期する手順は、67 ページの第 2 章「Java ES 共有コンポーネントのアップグレード」で説明されています。

Release 2 Directory Proxy Server を Release 5 にアップグレードするには、「Release 4」を「Release 2」と読み替えた上で、124 ページの「Java ES Release 4 からの Directory Proxy Server のアップグレード」の手順を使用してください。

注 Linux プラットフォームで Release 2 の Directory Proxy Server からアップグレードする場合は、二重のアップグレード、つまり、Directory Proxy Server とオペレーティングシステムの両方をアップグレードする必要があります (Release 5 の Directory Proxy Server は RHEL 2.1 でサポートされていない)。詳細については、123 ページの「二重のアップグレード」を参照してください。

第 7 章

Web Server

この章では、Web Server を Java ES 5 (Release 5): Sun Java System Web Server 7.0 にアップグレードする方法について説明します。

ここでは、Release 5 でサポートされている各種のアップグレードパスでのアップグレードの注意点の概要について説明します。また、Solaris オペレーティングシステムと Linux オペレーティングシステムの両方でのアップグレードを取り上げます。

- 134 ページの「Web Server のアップグレードの概要」
- 138 ページの「Java ES Release 4 からの Web Server のアップグレード」
- 155 ページの「Java ES Release 3 からの Web Server のアップグレード」
- 155 ページの「Java ES Release 2 からの Web Server のアップグレード」

注 この章で説明するファイルの場所は、*WebServer6-base* (Web Server 6.*x*) と *WebServer7-base* および *WebServer7Config-base* (Web Server 7.0) というディレクトリパスを基準に指定されています。これらのパスの少なくとも一部は、Web Server が最初にインストールされたときにインストールディレクトリとして指定されている可能性があります。指定されていない場合、Java ES インストーラによってデフォルト値が割り当てられています。

次の表に、これらのディレクトリパスのデフォルト値を示します。

表 7-1 Web Server ディレクトリパス

パス名変数	Solaris OS	Linux OS
WebServer6-*base*	/opt/SUNWwbsvr	/opt/sun/webserver
WebServer7-*base*	/opt/SUNWwbsvr7	/opt/sun/webserver7
WebServer7Config-base	/var/opt/SUNWwbsvr7	/var/opt/sun/webserver7

Web Server のアップグレードの概要

ここでは、Java ES 5 (Release 5) へのアップグレードに影響する Web Server の一般的な側面について説明します。

- Java ES Release 5 の Web Server について
- Web Server のアップグレードロードマップ
- Web Server のデータ
- Web Server のアップグレード方針

Java ES Release 5 の Web Server について

Java ES Release 5 の Web Server は Release 4 のメジャーリリースであり、多くの新機能やインタフェースの機能強化が追加されています。

Release 5 の Web Server は、新しい管理ツールを含む新しい管理インフラストラクチャーを装備しています。この管理インフラストラクチャーには、任意の数の Web Server インスタンスの設定情報をホストする Administration Server インスタンスが含まれています。新しいコマンド行インタフェース (`wadm`) と新しいグラフィカルユーザーインタフェースを使用して、ローカルコンピュータまたはリモートコンピュータのいずれかで Web Server インスタンスを作成し、これらのインスタンスを設定および管理します。この新しい管理ツールには、管理者ユーザー名とパスワードが必要です。

新しい管理インフラストラクチャーの詳細については、『Web Server 7.0 管理ガイド』(`http://docs.sun.com/doc/820-0875?l=ja`) を参照してください。

Web Server 管理インタフェースにおけるこれらの変更によって、アップグレードに重大な影響があります。

Web Server のアップグレードロードマップ

表 7-2 は、Web Server を Java ES Release 5 へアップグレードする際の、サポートされるアップグレードパスを示しています。この表は Solaris および Linux の両方のオペレーティングシステムに適用されます。

表 7-2 Java ES 5 (Release 5): Web Server 7.0 へのアップグレードパス

Java ES のリリース	Web Server のバージョン	一般的な手順	必要な再設定
Release 4	Sun Java System Web Server 6.1 SP5 2005Q4	直接アップグレード: 新規インストールのあと、データを移行します	インスタンス設定を新しいインスタンスに移行します。
Release 3	Sun Java System Web Server 6 2005Q1 Update 1 SP 4	直接アップグレード: 新規インストールのあと、データを移行します	インスタンス設定を新しいインスタンスに移行します。
Release 2	Sun Java System Web Server 6 2004Q2 Update 1 SP 2 Platform Edition および Enterprise Edition	直接アップグレード: 新規インストールのあと、データを移行します	インスタンス設定を新しいインスタンスに移行します。
Release 1	Sun ONE Web Server 6.1 (2003Q4)	直接アップグレードは動作保証なし: ただし、新規インストールのあとにデータを移行することによって実行されます。	インスタンス設定を新しいインスタンスに移行します。
それ以前の Java ES リリース		直接アップグレード不可。	

Web Server のデータ

次の表は、Web Server ソフトウェアのアップグレードにより影響を受ける可能性のあるデータの種類を示しています。

表 7-3　Web Server のデータの使用法

データの種類	場所	使用法
設定データ	Web Server 6.*x* (Java ES Release 2、3、および 4): *WebServer6-base*/https-*instanceName*/config/ Web Server 7.0 (Java ES Release 5): インスタンス設定 *WebServer7Config-base*/https-*configName*/config[1] / 中央の設定ストア Web Server コンソールを介して、および wadm コマンド行インタフェースを介してアクセスします。	Web Server インスタンスの設定

1. *WebServer7Config-base* のパスは *WebServer6-base* のパスと大幅に異なるので、注意が必要です。

Web Server のアップグレード方針

Web Server をアップグレードするための方針は一般に、第 1 章「アップグレードの計画」で説明されている多くの考慮事項 (アップグレードパス、Java ES コンポーネント間の依存関係、選択的なアップグレードかすべてのアップグレードか、複数インスタンスの配備など) によって異なります。

ここでは、Web Server のアップグレード計画に影響する可能性のある問題を取り上げ、Web Server の全般的な説明を詳細に記述します。

互換性の問題

Java ES Release 5 の Web Server では、公開インタフェースの変更は導入されていないので、この点では以前のバージョンと下位互換性があります。ただし、新しい管理インタフェースは、以前のバージョンの管理インタフェースとの下位互換性がありません。そのため、Web アプリケーション (Java ES コンポーネントなど) のアップグレードや再配備に影響を与えます。

特に、次の表に示すように、Release 5 の Web Server では、インスタンスのディレクトリや仮想サーバー名に対して異なるデフォルト値が使用されています。

表 7-4　Web Server のデフォルト名

項目	Java ES Release 2、3、および 4 Web Server 6.x のデフォルト	Java ES Release 5 Web Server 7.0 のデフォルト
設定名		*hostName.domainName*

表 7-4　Web Server のデフォルト名 (続き)

項目	Java ES Release 2、3、および 4 Web Server 6.x のデフォルト	Java ES Release 5 Web Server 7.0 のデフォルト
インスタンスのディレクトリパス	*WebServer6-base*/ `https-`*hostName.domainName*	*WebServer7Config-base* `https-`*hostName.domainName*
仮想サーバー名	`https-`*hostName.domainName*	*hostName.domainName*

Web Server の依存関係

Web Server は、次の Java ES コンポーネントに依存します。

- **共有コンポーネント** : Web Server は、特定の Java ES 共有コンポーネントに依存します (48 ページの表 1-9 を参照)。Web Server をアップグレードするために、これらの共有コンポーネントのバージョンをアップグレードしなければならない場合があります。
- Directory Server: Web Server は、LDAP ベースの認証サービスを提供するために、Directory Server に対して依存関係を持っていますが、これはオプションです。
- Web Proxy Server: Web Server は、HTTP 要求のセキュリティーとパフォーマンスを向上させるために、Web Proxy Server に対して共依存関係を持っています。

二重のアップグレード

44 ページの「二重のアップグレード : Java ES ソフトウェアとオペレーティングシステムソフトウェア」で説明されているように、二重のアップグレードでは Web Server とオペレーティングシステムの両方をアップグレードします。次のようなインプレースオペレーティングシステムアップグレードを使用して二重のアップグレードを実行できます。

1. 既存の Web Server データをバックアップします。

   基本的なデータの場所については、135 ページの「Web Server のデータ」を参照してください。

2. オペレーティングシステムをアップグレードします。

   このアップグレードでは、既存のファイルシステムはそのまま残されます。

3. Release 5 の Web Server にアップグレードします。

   アップグレードパスに応じて、この章の該当する節を参照してください。

Java ES Release 4 からの Web Server のアップグレード

ここでは、Web Server の Java ES 2005Q4 (Release 4) から Java ES 5 (Release 5) へのアップグレードについて説明します。この節で説明する内容は、次のとおりです。

- はじめに
- Release 4 の Web Server のアップグレード

はじめに

Java ES Release 4 Web Server を Release 5 にアップグレードする場合は、アップグレード処理の次の側面を考慮してください。

- **一般的なアップグレード手順** : このアップグレードは、Release 5 の Web Server の新規インストールを行い、Release 4 の Web Server のインスタンス設定情報を Release 5 設定に移行してから、Release 4 のインスタンスに対応する Release 5 の Web Server インスタンスを作成することによって実行されます。
- **アップグレードの依存関係** : Web Server は多くの Java ES 共有コンポーネントに依存しますが (48 ページの表 1-9 を参照)、これらはすべて、Web Server のアップグレードを実行したときに Java ES インストーラによって Release 5 に自動的にアップグレードされます。Web Server は、NSS および NSPR 共有コンポーネントに対してのみ強いアップグレード依存関係を持っています。
- **下位互換性** : Release 5 の Web Server 管理インタフェースは、Release 4 バージョンとの下位互換性はありません。
- **アップグレードのロールバック** : Release 5 へのアップグレードは、未変更のまま残されている Release 4 のインストールに復元することによってロールバックできます。
- **プラットフォームの問題** : Web Server をアップグレードするための一般的な手順は、Solaris および Linux の両オペレーティングシステムで共通です。

Release 4 の Web Server のアップグレード

ここでは、Solaris プラットフォームと Linux プラットフォームの両方で、Web Server の Java ES Release 4 から Java ES Release 5 へのアップグレードを実行する方法について説明します。プラットフォーム固有の手順に依存する項目には、その手順が適用されるオペレーティングシステムが示されています。この節で説明する内容は、次のとおりです。

- アップグレード前の作業
- Release 4 の Web Server のアップグレード
- アップグレードの確認
- アップグレード後の作業
- アップグレードのロールバック

アップグレード前の作業

Web Server ソフトウェアをアップグレードする前に、次の作業を実行してください。

- 現在のバージョン情報の確認
- Web Server が依存するコンポーネントのアップグレード
- Web Server のデータのバックアップ
- 必要な設定情報とパスワードの入手

現在のバージョン情報の確認

Web Server の現在のバージョンは、次のように、`-version` オプションを使用して Web Server インスタンスサーバーを実行することによって確認できます。

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

表 7-5　Web Server のバージョン確認の出力

Java ES のリリース	Web Server のバージョン番号
Release 2	6.1SP2
Release 3	6.1SP4
Release 4	6.1SP5
Release 5	7.0

Web Server が依存するコンポーネントのアップグレード

一般的には、コンピュータシステム上 (および、コンピューティング環境内) のすべての Java ES コンポーネントを Java ES Release 5 にアップグレードすることが推奨されます。ただし、Web Server の必要とするすべての共有コンポーネント (48 ページの表 1-9 を参照) は、Web Server の Release 5 へのアップグレードを実行するときに、Java ES インストーラによって自動的にアップグレードされます。

Web Server のデータのバックアップ

Web Server の Release 4 から Release 5 へのアップグレードでは、既存の設定データは変更されず、そのまま残されます。現在のデータをバックアップする必要はありません。

必要な設定情報とパスワードの入手

アップグレードを実行するには、スーパーユーザーとしてログインする必要があります。また、移行を実行するユーザーアカウントが、既存の Web Server インストールディレクトリに対してアクセス許可を持っているようにしてください。

Release 4 の Web Server のアップグレード

ここでは、Web Server のアップグレード手順に影響する注意点を示したあと、その手順自体について説明します。

アップグレードの注意点

Web Server ソフトウェアを Java ES Release 5 にアップグレードする場合は、次の注意点を考慮に入れるようにしてください。

- **「今すぐ設定」または「あとで設定」**: アップグレードを実行するときに、Release 5 の Web Server のインストールを、「今すぐ設定」オプションを使って行うか、または「あとで設定」オプションを使って行うかを指定します。
  - 「今すぐ設定」を選択すると、インストーラは指定どおりに Administration Server または管理ノードを設定し、デフォルト設定や対応する Web Server インスタンスも作成します。この方法は、1 台のコンピュータでのインストールには便利ですが、デフォルトの設定名が、アップグレード時の既存のインスタンス設定の移行に影響を与える可能性があります。
  - 「あとで設定」を選択すると、インストーラは何も設定を行いません。ユーザーは、入力ファイルにプロパティー値を設定したあと、手動で `configureServer` スクリプトを実行する必要があります。この方法は、サイレントインストールを実行するスクリプトを使用して、複数のコンピュータに自動的にインストールを行う場合に便利です。また、設定名を完全に制御できるので、アップグレード時の既存のインスタンス設定の移行による競合を回避できます。

- **デフォルトの Release 4 のインスタンス設定の移行** : アップグレードを実行するときに、Release 4 の 各 Web Server インスタンスの設定データを、Web Server Administration Server によって保持されている中央の設定ストアに移行します。この移行は、`wadm migrate-server` コマンドまたは Release 5 の管理コンソールを使用して行います。

  移行するインスタンスがデフォルトの Release 4 の Web Server (6.x) インスタンスである場合は、Release 5 の Web Server (7.0) のデフォルト設定 (「今すぐ設定」オプションによって自動的に作成される設定) と同じ名前 (*hostName.domainName*) になります。

  デフォルトの Release 4 インスタンス設定の Release 5 への移行を実行する場合、3 とおりの方法があり、それぞれ異なる設定名になります。

  *どの方法で移行するのかという選択は、配備済みの Web アプリケーションのその後のアップグレードに影響を与える可能性があります。*たとえば、配備済みの Java ES コンポーネント (Access Manager、Portal Server など) および Sun Java Communications Suite コンポーネント (Communications Express、Instant Messaging、Delegated Administrator など) のアップグレードを行う場合には通常、このようなアップグレードを実行するユーザーは、Release 4 のインスタンス設定の移行先となった Release 5 の設定の名前を知っている必要があります。

  3 とおりの移行方法は、次のとおりです。

  - デフォルト名とは異なる新しい設定名 (たとえば `JavaESapps`) を指定します。すると、`migrate-server` コマンドによって `JavaESapps` という名前の新しい設定が作成されます。

    あとからアップグレードする Web アプリケーションを、`JavaESapps` 設定に対して再配備する必要があります。

  - 新しい設定名を指定するのではなく、`migrate-server` コマンドを実行する前に、新しいデフォルトの Release 5 インスタンスと設定 (*hostName.domainName*) を削除します。次に、`migrate-server` コマンドによって、デフォルト名 (*hostName.domainName*) を持つ新しい設定を作成します。順序は次のようになります (詳細は `wadm help` を参照)。

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

    あとからアップグレードする Web アプリケーションを、デフォルトの Release 4 インスタンス名と同じである *hostName.domainName* 設定に対して再配備する必要があります。

  - 新しい設定名は指定しません。なお、デフォルトの Release 5 の設定は削除しないでください。次に、`migrate-server` コマンドによって、*hostName.domainName*`-1` という名前を持つ新しい設定を作成します。

あとからアップグレードする Web アプリケーションを、*hostName.domainName*-1 設定に対して再配備する必要があります。

どの方法で Release 4 インスタンス設定の移行を行なっても、Release 4 のインスタンス設定の移行先となった Release 5 の設定名を、そのインスタンスに配備される Web アプリケーションのアップグレードを移行後に実行するユーザーに伝達する必要があります。

- **設定データの移行** : Release 4 インスタンス設定の移行するときに、次の情報は自動的に移行されます。
  - Release 4 の Web Server インスタンスディレクトリ (*WebServer6-base*/https-*instanceName*/config) 内のすべての設定情報。これには、Release 4 インスタンスに配備されたすべての Web アプリケーションに関する設定情報 (たとえば、Access Manager、Portal Server などの Java ES コンポーネント) が含まれます。
  - *WebServer6-base*/httpacl からの ACL 情報
  - *WebServer6-base*/userdb からの auth-db 情報
  - *WebServer6-base*/https-admserv/config からのスケジューラ情報
  - *WebServer6-base*/alias からの証明書情報
  - 移行の実行時に指定した、検索コレクション情報とインデックスファイル

  自動的な移行には、次のデータは含まれません。
  - docroot の内容。その代わりに、新しい設定が古い docroot をポイントし、ログメッセージは移行ログに記録されます。
  - Webdav データ。Webdav コレクション情報は移行されます。
  - サードパーティーの NSAPI プラグインは移行されません。その代わりに、新しい設定が Release 4 のファイルをポイントし、ログメッセージは移行ログに記録されます。
  - ログファイル
  - 検索コレクションの docroot に対する変更
  - コマンド行スクリプト (startsvr、startsvr.bat、stopsvr、stopsvr.bat、restart、reconfig、reconfig.bat)

  データ移行の詳細については、『Web Server 7.0 Installation and Migration Guide』(http://docs.sun.com/doc/819-2625) を参照してください。
- **Release 5 の Web Server インスタンスの作成** : 移行後に、移行元の Release 4 インスタンスに対応する Release 5 の Web Server インスタンスを明示的に作成してください。この操作は、自動的には実行されません。

アップグレード手順

ここで説明する手順は、アップグレードを実施するコンピュータにインストールされている同じ Web Server イメージに対応するすべての Web Server インスタンスに適用されます。

1. root としてログインするか、スーパーユーザーになります。

   ```
   su -
   ```

2. Web Server および Administration Server の、稼働中のすべてのインスタンスを停止します。

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

3. Release 5 の Web Server の新規インストールを実行します。

   次の手順を実行します。

   a. Java ES インストーラを起動します。

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

      ここで、*os_arch* は、`Solaris_sparc` などのプラットフォームに対応します。コマンド行インタフェースには、`installer -nodisplay` オプションを使用します。

      ようこそページと、ライセンス使用許諾契約ページが表示されてから、コンポーネント選択ページが表示されます。Java ES インストーラが直接アップグレードできるインストール済みのコンポーネントが検出された場合、それらのコンポーネントは「アップグレード可」の状態で表示されます。

   b. コンポーネント選択ページで、Web Server を選択します。

   c. Release 4 の Web Server とは異なるインストールパスを指定します。

   d. 「今すぐ設定」または「あとで設定」を選択します。

      - 「今すぐ設定」を選択した場合は、手順 e に進みます。
      - 「あとで設定」を選択した場合は、手順 f に進みます。

   e. 「今すぐ設定」を選択した場合、Java ES インストーラから次の 2 つの選択肢が提供されます。

      - 「管理インスタンスをサーバーとして設定」
        ほかの管理タスクに加えて、Release 4 インスタンスの Release 5 への移行を実行する必要のある Administration Server をホストするコンピュータでは、こちらを選択します。

- 管理インスタンスを管理ノードとして設定
  Administration Server とは別のマシンにある Web Server インスタンスをホストするコンピュータでは、こちらを選択します。管理インスタンスは、Administration Server と相互作用するノードエージェントとして設定されます。

I. 要求される設定値を指定します。

ホスト名、HTTP ポート、管理ユーザー名および管理パスワードの入力を求められます。

II. インストールに関して選択した項目を確認します。

Web Server パッケージがインストールされ、インストールの概要が表示されます。

Java ES インストーラは、*hostName.domainName* という名前のデフォルト設定と、対応する Web Server インスタンスを作成します。

III. Java ES インストーラを終了して、146 ページの手順 4 に進みます。

f. 「あとで設定」を選択した場合、Java ES インストーラは、146 ページの手順 IV で実行する `configureServer` スクリプトを作成します。

I. インストールに関して選択した項目を確認します。

Web Server パッケージがインストールされ、インストールの概要が表示されます。

II. Java ES インストーラを終了します。

III. *WebServer7-base*`/setup/WSInstall.properties` ファイルに値を設定します。

次の表にあるすべての必須 (省略可能でない) プロパティーに値を設定します。

表 7-6 `WSInstall.properties` の値

プロパティー	説明
`WS_DOCROOT`	(省略可能) Web コンテンツファイルをホストできるドキュメントの場所
`WS_SERVER_NAME`	HTTP 要求を操作するために使用できるホスト名
`WS_SERVER_USER`	実行時の UNIX ユーザー。有効な値は、`root`、すべての有効な UNIX ユーザー、または `webservd` (デフォルト) です。
`WS_HTTP_PORT`	HTTP 要求を待機するために使用できるインスタンスポート
`WS_ADMIN_SSL_PORT`	SSL の管理ポート

表 7-6 WSInstall.properties の値 (続き)

プロパティー	説明
WS_ADMIN_HOST	管理サーバータスク用の管理ホスト名
WS_CONFIG_NAME	このホストの設定名。この値は、WS_SERVER_NAME に指定したのと同じ値にできます。
WS_ADMIN_SERVER_USER	管理サーバーの実行時の UNIX ユーザー。有効な値は、「root」または WS_SERVER_USER と同じユーザーです。
WS_ADMIN_LOGIN_USER	管理サーバーのログインユーザー名
WS_ADMIN_LOGIN_PASSWORD	管理サーバーのログインパスワード
WS_ADMIN_HTTP_PORT	(省略可能) 非 SSL の管理ポート。デフォルト : 8800
WS_START_ON_BOOT	(省略可能) ブート機能を起動 (true/false)。true に設定すると、サーバーインスタンスとその管理サーバーは、システムの再起動後に自動的に起動できるようになります。デフォルト : false
WS_64BIT_INSTALL	(省略可能) サーバーの実行時モード (true/false)。true に設定すると、サーバーが 64 ビットモードに設定されます (Solaris のみ)。false に設定すると、サーバーが 32 ビットモードに設定されます (Solaris のみ)。デフォルト : false
WS_ADMIN_IS_SERVER_MODE	(省略可能) 管理設定モード。(true/false)。true に設定すると、サーバーが管理サーバーモードに設定されます。false に設定すると、サーバーが管理エージェントモードに設定されます。デフォルト : true
WS_REGISTER_ADMIN_AGENT	(省略可能) リモートエージェント登録 (true/false)。WS_ADMIN_IS_SERVER_MODE を false に設定した場合にのみ必要です。true に設定すると、登録用のリモート管理サーバーホストを入力する必要があります。デフォルト : true
WS_AGENT_SSL_PORT	(省略可能) 管理エージェントの SSL ポート。WS_ADMIN_IS_SERVER_MODE を false に設定した場合にのみ必要です。

表 7-6　WSInstall.properties の値 (続き)

プロパティー	説明
WS_AGENT_HOST=	(省略可能) 管理エージェントのホスト名。WS_ADMIN_IS_SERVER_MODE を false に設定した場合にのみ必要です。

IV. configureServer スクリプトを実行します。

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

configureServer スクリプトは、*hostName.domainName* という名前のデフォルト設定と、対応する Web Server インスタンスを作成します。

4. Web Server の Administration Server サービスを起動します。

WebServer7Config-base/admin-server/bin/startserv

5. Release 4 の Web Server インスタンス設定を Release 5 設定に移行します。

コマンド行 (wadm) またはグラフィカルユーザーインタフェースの管理ツール (Web Server Admin Server GUI にログイン) のどちらを使用することもできます。次に示す手順は、wadm コマンド行インタフェースを基にしています。

たとえば、myinstance という名前のインスタンスを新しい設定に移行するには、次の手順に従います。

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

完全なコマンド構文は次のとおりです。

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*])

上記の最初の一連のコマンドオプションは、すべての wadm コマンドに共通で、下の表 7-7 で説明されています。2 番目の一連のコマンドオプションは、migrate-server コマンドに固有のもので、表 7-8 で説明されています。

最初の一連のコマンドオプションのみを使用して wadm を起動すると、wadm コマンドシェル内で操作を行うことになります。このシェル内でコマンドを起動する場合、共通オプションを再指定する必要はありません。

シェルの外側から完全な wadm コマンドを起動する場合は、少なくとも --user オプションと --host オプションを指定してください。--password-file オプションを省略すると、パスワードの入力が求められ、ほかのオプションを省略すると、デフォルト値が割り当てられます。ただし、説明を簡略化するために、この章で手順を説明するために使用するコマンドには、--user オプションと --host オプションは含まれていません。

デフォルトでは、wadm は、ポート 8989 で SSL プロトコルを使用します。

wadm コマンドおよびオプションの詳細については、『Web Server 7.0 CLI Reference Manual』(http://docs.sun.com/doc/819-3283) を参照してください。

表 7-7 wadm 共通コマンドオプション

オプション	説明
user	承認された Web Server 管理ユーザー ID。
password-file	管理ユーザーを Administration Server に対して認証するためのパスワードを格納しているファイル。このパスワードファイルには、WADM_PASSWORD=*password* という行を含めてください。コマンド中にこのオプションを指定していない場合は、パスワードの入力が求められます。
host	Administration Server を実行しているコンピュータの名前。デフォルト : localhost
echo	このオプションを true に設定すると、コマンドを実行する前に、コマンド行が標準出力上に表示されます。デフォルト : false
interactive	このオプションを true に設定すると、必要なパスワードオプションの入力が求められます。デフォルト : true
rcfile	wadm の起動時にロードするために使用される起動ファイル。デフォルト : ~/.wadmrc
no-prompt	このオプションを true に設定すると、どのような状況にあってもこのコマンドがユーザー入力を求めることはなくなります。たとえば、パラメータが不足している状態で起動された場合、ユーザー入力を求めて待機するのではなく、単にエラーアウトになります。wadm コマンドをシェルスクリプトで使用し、ユーザーの入力を待機するのではなくコマンドが常に返るようにする場合は、true に設定することをお勧めします。デフォルト : false
verbose	true に設定すると、冗長なリストが表示されます。デフォルト : false

表 7-8　wadm migrate-server コマンドのオプションとオペランド

オプションまたはオペランド	説明
search-collection-copy-path	検索コレクションを移行するときに、検索コレクションのインデックスファイルのコピー先のパスを指定します。次のような移行シナリオが考えられます。 Web Server 6.*x* 検索コレクションパスが Web Server 6.*x* インスタンスの外部にある場合、移行された検索コレクションパスは Web Server 6.*x* 検索コレクションパスをポイントし、このオプションは無視されます。 Web Server 6.*x* 検索コレクションパスが Web Server 6.*x* インスタンスの内部にあり、このオプションに対して有効なパスが指定されている場合、検索コレクションのインデックスファイルは、次のディレクトリにコピーされます。*searchCollectionPath*/*configName*/*virtualServerName*/*collectionName*。指定されたパスが有効でない場合、エラーメッセージが記録されます。 Web Server 6.*x* 検索コレクションパスが Web Server 6.*x* インスタンスの内部にあるものの、このオプションに対して何もパスが指定されていない場合、検索コレクションのインデックスファイルはコピーされません。`wadm add-documents` コマンドを使用して検索コレクションのインデックスファイルを手動でコピーするようにユーザーに求めるメッセージが、移行ログに書き込まれます。この場合、移行された検索コレクションパスは、次のようになります。*WebServer7Config-base*/`https-`*configName*`/config/collections/`*virtualServerName*/*collectionName*
log-dir	移行ログの場所。 デフォルト : *WebServer7Config-base*`/admin-serv/logs`
serverroot	Web Server 6.*x* バージョンをインストールする場所 (ディレクトリ)。*WebServer6-base* と同じ。
all	`true` に設定した場合、Web Server 6.*x* のすべてのインスタンス設定は、同じインスタンス名の Web Server 7.0 の設定に移行されます。この名前の設定がすでに存在している場合、設定名として *instanceName*`-1` が使用されます。デフォルト : `false`
instance	インスタンス設定が個別に移行される場合 (all=`false`) の、移行される Web Server 6.*x* インスタンス設定の名前 (形式 : `https-`*instanceName*)。Web Server6.x インスタンスのデフォルト名は *hostName.domainName* です。

表 7-8　wadm migrate-server コマンドのオプションとオペランド (続き)

オプションまたはオペランド	説明
config	指定した Web Server 6.*x* インスタンス設定の移行先の設定名。デフォルトは、Web Server 6.*x* インスタンス設定の *instanceName* を使用します。ただし、その名前の設定がすでに存在している場合は、コマンドによって整数が名前に付加されます。これは、デフォルトの Web Server 6.*x* インスタンス設定を移行する場合に起こり得るシナリオです。

migrate-server コマンドを使用する場合は、次の点に留意してください。

- 複数の Release 4 インスタンス設定を移行する場合、それぞれ異なる --instance 値とそれに対応する --config 引数を使用して migrate-server コマンドを何度も実行するか、--all オプションを使用して一度にすべてを移行することができます。
- migrate-server コマンドを起動するたびに、--log-dir オプションによって指定したディレクトリ (またはデフォルトの *WebServer7Config-base*/admin-server/logs ディレクトリ) に、次の名前のログファイルが作成されます。

  MIGRATION_*yyyymmddhhmmss*.log

  --all オプションを選択した場合、ログファイルには移行されたすべてのインスタンスに関する移行情報が格納されます。
- migrate-server コマンドでは移行されないデータ (140 ページの「アップグレードの注意点」を参照) に対しては、手動で移行を実行してください (151 ページの「アップグレード後の作業」を参照)。

6. Release 5 の Web Server インスタンスを作成します。

   手順 5 で移行した Release 4 インスタンス設定のそれぞれに対して、新しい Release 5 インスタンスを作成してください。

   a. 新しいインスタンスを作成する前に、移行ログを確認し、移行された設定における問題をすべて修正しておきます。

   b. create-instance コマンドを実行します。

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

      共通のコマンドオプションについては、147 ページの表 7-7 で説明されています。create-instance コマンドに固有のオプションについては、次の表で説明されています。

表 7-9　wadm create-instance コマンドのオプションとオペランド

オプションまたはオペランド	説明
config	インスタンスがポイントする Release 5 の設定の名前。
nodehost	インスタンスが作成されるコンピュータの名前。*hostName.domainName* のスペース区切りのリストで複数のコンピュータを指定できます。これによって複数の同一のインスタンスを作成します。

create-instance コマンドは、指定したノード上の *WebServer7Config-base*/https-*configName* にインスタンスディレクトリを作成し、対応するインスタンスディレクトリに設定を配備します。

7. 各 Release 5 インスタンスを起動します。

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

   インスタンスが作成されると、startserv スクリプトが作成されます。インスタンスが問題なく起動すると、サーバーが正常に起動したことを示すメッセージが表示され、インスタンスのデフォルトの URL が表示されます。

アップグレードの確認

Release 5 への Web Server のアップグレードが成功したかどうかは、次の手順を実行することによって確認できます。

1. 新しく作成された移行ログファイルをチェックして、ERROR メッセージがないかを確認します。

   必要に応じて、手動で変更を行います (151 ページの「アップグレード後の作業」を参照)。

2. Release 5 の Web Server インスタンスを確認します。

   Web ブラウザから次の URL にアクセスし、「ようこそ」ページが表示されることを確認します。

   http://*hostName.domainName*:*port*

   ここで、完全修飾ホスト名とポートは各インスタンスに対応します。

3. -version オプションを使用して、Web Server インスタンスを実行します。

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

   バージョンの出力値については、139 ページの表 7-5 を参照してください。

アップグレード後の作業

おもなアップグレード後の作業は、特定の Release 4 のデータを必要に応じて手動で移行することに関係します。通常、これは、Release 4 用に設定されて server.xml 設定ファイル内に指定された 1 つ以上の仮想サーバーに関連するデータです。

次の状況に対処するためにアップグレード後の手順に注意してください。

- Web Server 6.1 の docroot の内容の移行
- webdav コレクション情報の移行
- ログファイルの移行
- サードパーティーの NSAPI プラグインの移行
- 検索コレクションのドキュメントルートの変更
- コマンド行スクリプトのカスタマイズ

Web Server 6.1 の docroot の内容の移行

1. Web Server 6.1 の docroot の内容を任意の場所にコピーします。
2. 次のコマンドを使用して、新しいドキュメントルートのパスを更新します。

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

   共通のコマンドオプションについては、147 ページの表 7-7 で説明されています。set-virtual-server-prop コマンドに固有のオプションについては、次の表で説明されています。

   表 7-10　wadm set-virtual-server-prop コマンドのオプションとオペランド

オプションまたはオペランド	説明
config	新しくドキュメントルートのパスを設定する Release 5 の設定の名前。
vs	移行後のドキュメントルートが対応する仮想サーバーの名前。
document-root	新しいドキュメントルートディレクトリへのパス。

3. 関連する Web Server インスタンスに設定を再配備します。

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

共通のコマンドオプションについては、147 ページの表 7-7 で説明されています。deploy-config コマンドに固有のオプションについては、次の表で説明されています。

表 7-11 wadm deploy-cofig コマンドのオプションとオペランド

オプションまたはオペランド	説明
force	true の場合、前回の設定の配備以降に手動で変更を行ったインスタンス設定を、強制的に上書きします。デフォルト : false
restart	true の場合、実行中のインスタンスが再起動し、配備された設定から設定情報を取得します。デフォルト : false
no-reconfig	true に設定した場合、実行中のインスタンスは、再起動されるまで、配備された設定から設定情報を取得しません。デフォルト : false
configName	インスタンス名が設定名に対応する Web Server インスタンスに配備される Release 5 の設定の名前。

webdav コレクション情報の移行

手動での移行は必要ありません。docroot のパスを更新するだけです。

ログファイルの移行

ログファイルを保存する場合は、既知の場所にコピーします。そうしないと、Release 4 のインストールを削除するとログファイルも削除されます。

サードパーティーの NSAPI プラグインの移行

1. ライブラリファイルを Release 4 の場所から *WebServer7-base*/lib ディレクトリにコピーします。
2. magnus.conf および obj.conf 設定ファイルを一時ディレクトリにエクスポートします。

   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

   共通のコマンドオプションについては、147 ページの表 7-7 で説明されています。
3. サードパーティーの NSAPI プラグインのマニュアルに記載されているように、magnus.conf ファイルと obj.conf ファイルを変更します。

4. magnus.conf および obj.conf 設定ファイルを一時ディレクトリからインポートします。

   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

   共通のコマンドオプションについては、147 ページの表 7-7 で説明されています。

5. 関連する Web Server インスタンスに変更した設定を再配備します。

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

   コマンドオプションについては、152 ページの表 7-11 で説明されています。

検索コレクションのドキュメントルートの変更

migrate-server コマンドには、検索コレクション情報を移行するためのオプションがありますが、次のように検索コレクションのドキュメントルートを変更することもできます。

1. 検索コレクションのドキュメントルートが Release 4 で使用していたものと異なる場合、次のコマンドを使用して、検索コレクションのドキュメントルートを設定します。

   WebServer7-base/bin/wadm set-search-collection-prop --config=*configName* --vs=*virtualServerName* --collection-name=*searchCollectionName* document-root=*new docroot path for the search collection*

   共通のコマンドオプションについては、147 ページの表 7-7 で説明されています。set-search-collection-prop コマンドに固有のオプションについては、次の表で説明されています。

表 7-12 wadm set-search-collection-prop コマンドのオプションとオペランド

オプションまたはオペランド	説明
config	検索コレクションのドキュメントルートを設定する Release 5 の設定の名前。
vs	検索コレクションが対応する仮想サーバーの名前。
collection-name	新しくドキュメントルートのパスを設定する検索コレクションの名前。
document-root	検索コレクションの新しいドキュメントルートディレクトリへのパス。

2. 関連する Web Server インスタンスに設定を再配備します。

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

   コマンドオプションについては、152 ページの表 7-11 で説明されています。

コマンド行スクリプトのカスタマイズ

startsvr、startsvr.bat、stopsvr、stopsvr.bat、restart、reconfig、reconfig.bat などのスクリプトがカスタマイズされている場合、次のディレクトリにある Release 5 のデフォルトスクリプトに対しても同じカスタマイズを実行する必要があります。
WebServer7-base/bin

アップグレードのロールバック

Release 5 の新規インストールやその後の Web Server インスタンス設定の移行を行うことにより、Release 4 の Web Server は何も変更されずにそのまま残りました。したがって、Release 5 の Web Server のロールバックは、Release 4 へ復元する次の手順によって構成されます。

1. root としてログインするか、スーパーユーザーになります。

   su -

2. 実行中のすべての Web Server インスタンスを 1 つずつ停止します。

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

   サーバーが正常に停止したら、サーバーがシャットダウンされたことを示すメッセージが表示されます。

3. Release 5 の Web Server インストールを削除します。

   すべての Release 5 インスタンスと移行された設定を削除してください。

   a. Release 5 のインスタンスをすべて削除します。

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

   b. すべての Release 5 設定を削除します。

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

4. 143 ページの「アップグレード手順」の説明に従って、Web Server のアップグレード時に停止させた Web Server インスタンスを再起動します。

Java ES Release 3 からの Web Server のアップグレード

Java ES 2005Q1 (Release 3) Web Server を Release 5 にアップグレードする手順は、Release 4 Web Server を Release 5 にアップグレードする場合と同じです。

Release 3 Web Server を Release 5 にアップグレードするには、「Release 4」を「Release 3」と読み替えた上で、138 ページの「Java ES Release 4 からの Web Server のアップグレード」の手順を使用してください。

Java ES Release 2 からの Web Server のアップグレード

Java ES 2004Q2 (Release 2) の Web Server を Release 5 にアップグレードする手順は、Release 4 の Web Server を Release 5 にアップグレードする手順と同じです。

Release 2 Web Server を Release 5 にアップグレードするには、「Release 4」を「Release 2」と読み替えた上で、138 ページの「Java ES Release 4 からの Web Server のアップグレード」の手順を使用してください。

注 Linux プラットフォームで Release 2 の Web Server からアップグレードする場合は、二重のアップグレード、つまり、Web Server とオペレーティングシステムの両方をアップグレードする必要があります (Release 5 の Web Server は RHEL 2.1 でサポートされていない)。詳細については、137 ページの「二重のアップグレード」を参照してください。

第 8 章

Java DB

この章では、Java DB を Java ES 5 (Release 5): Java DB 10.1.3 にアップグレードする方法について説明します。

ここでは、Release 5 でサポートされている各種のアップグレードパスでのアップグレードの注意点の概要について説明します。また、Solaris オペレーティングシステムと Linux オペレーティングシステムの両方でのアップグレードを取り上げます。

- 158 ページの「Java DB のアップグレードの概要」
- 161 ページの「Java ES Release 4 からの Java DB のアップグレード」

注 この章で説明するファイルの場所は、*JavaDB-base* というディレクトリパスを基準に指定されています。このパスは、Java DB がインストールされたときに Java ES インストーラによって設定されています。

次の表に、このディレクトリパスの値を示します。

表 8-1 Java DB ディレクトリパス

パス名変数	Solaris OS	Linux OS
JavaDB-base	`/opt/SUNWjavadb`	`/opt/sun/javadb`

Java DB のアップグレードの概要

ここでは、Java ES 5 (Release 5) へのアップグレードに影響する Java DB の一般的な側面について説明します。

- Java ES Release 5 について
- Java ES Release 5 へのアップグレードロードマップ
- Java DB のデータ
- Java DB のアップグレード方針

Java ES Release 5 について

Java ES Release 5 Java DB は、Java ES 製品コンポーネントとして提供される最初のリリースです。Java DB は、Java ES Release 4 に含まれる DerbyDatabase という名前の共有コンポーネントとして、最初にリリースされました。

Release 5 Java DB は、Release 4 バージョンに対するマイナーリリースです。このリリースでは、いくつかの機能強化、インタフェースの更新、選ばれたバグ修正が行われています。

Java ES Release 5 へのアップグレードロードマップ

表 8-2 は、Java DB を Java ES Release 5 へアップグレードする際の、サポートされるアップグレードパスを示しています。この表は Solaris および Linux の両方のオペレーティングシステムに適用されます。

表 8-2 Java ES 5 (Release 5): Java DB 10.1.3 へのアップグレードパス

Java ES のリリース	Java DB のバージョン	一般的な手順	必要な再設定
Release 4	Derby Database 10.0.2	直接アップグレード: 新規インストールにより、Release 4 を置き換えます。持続的データは影響されません。	なし

Java DB のデータ

次の表は、Java DB ソフトウェアのアップグレードにより影響を受ける可能性のあるデータの種類を示しています。

表 8-3　Java DB のデータの使用法

データの種類	場所	使用法
設定データ	インスタンス設定はアプリケーション固有であり、Java DB データベースに保存されます。	Java DB インスタンスの設定
持続的データ	データベースディレクトリとその内容は、アプリケーション固有です。それらの場所は、データベース接続 URL によって指定されます。`jdbc:derby:` データベースのフルパス。	データベースとユーザー証明書

Java DB のアップグレード方針

Java DB をアップグレードするための方針は一般に、第 1 章「アップグレードの計画」で説明されている多くの考慮事項 (アップグレードパス、Java ES コンポーネント間の依存関係、選択的なアップグレードかすべてのアップグレードか、複数インスタンスの配備など) によって異なります。

ここでは、Java DB のアップグレード計画に影響する可能性のある問題を取り上げ、Java DB の全般的な説明を詳細に記述します。

互換性の問題

Release 5 の Java DB は、Release 4 バージョンと下位互換性があります。

依存関係

Java DB は、J2SE 共有コンポーネントに対してのみ依存関係を持っています (48 ページの表 1-9 を参照)。

二重のアップグレード

44 ページの「二重のアップグレード : Java ES ソフトウェアとオペレーティングシステムソフトウェア」で説明されているように、二重のアップグレードでは Java DB とオペレーティングシステムの両方をアップグレードします。次の 2 つのどちらかの方法で二重のアップグレードを実行できます。

- 新規オペレーティングシステムのインストール

- オペレーティングシステムのインプレースアップグレード

新規オペレーティングシステムのインストール

1. 既存の Java DB データをバックアップします。

   基本的なデータの場所については、159 ページの「Java DB のデータ」を参照してください。

2. 新しいオペレーティングシステムをインストールします。

   新しいシステム (または Solaris 10 ゾーン) でオペレーティングシステムのインストールを実行するか、または、既存のファイルシステムを消去することができます。

3. Release 5 の Java DB をインストールします。

4. 手順 1 でバックアップした Java DB データを復元します。

オペレーティングシステムのインプレースアップグレード

1. 既存の Java DB データをバックアップします。

   基本的なデータの場所については、159 ページの「Java DB のデータ」を参照してください。

2. オペレーティングシステムをアップグレードします。

   このアップグレードでは、既存のファイルシステムはそのまま残されます。

3. Release 5 の Java DB にアップグレードします。

   161 ページの「Java ES Release 4 からの Java DB のアップグレード」を参照してください。

   Java DB データはアップグレードの影響を受けません。

Java ES Release 4 からの Java DB のアップグレード

ここでは、Java DB の Java ES 2005Q4 (Release 4) から Java ES 5 (Release 5) へのアップグレードについて説明します。この節で説明する内容は、次のとおりです。

- はじめに
- Release 4 の Java DB のアップグレード
- 複数インスタンスのアップグレード

はじめに

Java ES Release 4 Java DB を Release 5 にアップグレードする場合は、アップグレード処理の次の側面を考慮してください。

- **一般的なアップグレード手順** : このアップグレードは、Release 5 の Java DB の新規インストールを行い、Release 4 バージョンを置き換えることによって実行されます。Release 4 のデータと設定は、そのまま残ります。
- **アップグレードの依存関係** : Java DB は J2SE 共有コンポーネント (48 ページの表 1-9 を参照) に対して強いアップグレード依存関係を持っています。この共有コンポーネントは、Java DB のアップグレートを実行すると、Java ES インストーラによって Release 5 に自動的にアップグレードされます。
- **下位互換性** : Release 5 の Java DB は Release 4 と完全互換です。
- **アップグレードのロールバック** : Release 5 アップグレードのロールバックは、バックアップされた Release 4 インストールを戻すことによってのみ可能です。
- **プラットフォームの問題** : Java DB をアップグレードするための一般的な手順は、Solaris および Linux の両オペレーティングシステムで共通です。

Release 4 の Java DB のアップグレード

ここでは、Solaris プラットフォームと Linux プラットフォームの両方で、Java DB の Java ES Release 4 から Java ES Release 5 へのアップグレードを実行する方法について説明します。この節で説明する内容は、次のとおりです。

- アップグレード前の作業
- Release 4 の Java DB のアップグレード
- アップグレードの確認
- アップグレード後の作業
- アップグレードのロールバック

アップグレード前の作業

Java DB ソフトウェアをアップグレードする前に、次の作業を実行してください。

- 現在のバージョン情報の確認
- Java DB が依存するコンポーネントのアップグレード
- Java DB のデータのバックアップ
- 必要な設定情報とパスワードの入手

現在のバージョン情報の確認

Java DB の現在のバージョンは、次のコマンドを使用することにより確認できます。

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

バージョン情報 (`derby.jar` ファイル) を次の表に示します。

表 8-4　Java DB のバージョン確認の出力

Java ES のリリース	Java DB のバージョン番号
Release 4	`10.0.2.1`
Release 5	`10.1.3.1`

Java DB が依存するコンポーネントのアップグレード

一般的には、コンピュータシステム上 (および、コンピューティング環境内) のすべての Java ES コンポーネントを Java ES Release 5 にアップグレードすることが推奨されます。ただし、Java DB の必要とするすべての共有コンポーネントは、Java DB の Release 5 へのアップグレードを実行するときに、Java ES インストーラによって自動的にアップグレードされます。

Java DB のデータのバックアップ

Java DB の Release 4 から Release 5 へのアップグレードでは、設定データや持続的データは変更されません。ただし、安全のために、Java DB のインストール全体とデータをバックアップすることをお勧めします。159 ページの表 8-3 を参照してください。

必要な設定情報とパスワードの入手

Java DB をアップグレードするために、設定情報やパスワードは必要ありません。

Release 4 の Java DB のアップグレード

ここでは、Solaris プラットフォームと Linux プラットフォームでのアップグレード手順について説明します。

アップグレード手順

次に示す手順は、アップグレードを実施するコンピュータにローカルインストールされているすべての Java DB インスタンスに適用されます。

1. root としてログインするか、スーパーユーザーになります。

   ```
   su -
   ```

2. Release 4 の Java DB を停止します。

   ネットワークサーバーが実行中の場合は、次のコマンドを使用します。

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

   それ以外の場合は、すべての Java DB を使用しているアプリケーションを単にシャットダウンします。

3. Release 5 の Java DB の新規インストールを実行します。

   次の手順を実行します。

a. Release 4 の Java DB がホストされたコンピュータ上で Java ES インストーラを起動します。

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

ここで、*os_arch* は、`Solaris_sparc` などのプラットフォームに対応します。コマンド行インタフェースには、`installer -nodisplay` オプションを使用します。

ようこそページと、ライセンス使用許諾契約ページが表示されてから、コンポーネント選択ページが表示されます。Java ES インストーラが直接アップグレードできるインストール済みのコンポーネントが検出された場合、それらのコンポーネントは「アップグレード可」の状態で表示されます。

b. コンポーネント選択ページから Java DB を選択します。

以前のインストールは上書きされます。

c. 「あとで設定」オプションを選択します。

「今すぐ設定」はサポートされていません。

d. 必要に応じて、ローカライズ版のパッケージをインストールするオプションを選択します。

e. インストールが完了したら、Java ES インストーラを終了します。

4. Release 5 の Java DB を起動します。

ネットワークサーバーを使用している場合は、次のコマンドを使用します。

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

それ以外の場合は、Java DB を組み込みモードで使用して、任意のアプリケーションを単に起動します。

アップグレードの確認

Java DB のアップグレードが成功したかどうかは、次のコマンドを使用して確認できます。

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

出力値 (`derby.jar` ファイルのバージョン) については、162 ページの表 8-4 を参照してください。

アップグレード後の作業

Java DB を Release 4 から Release 5 にアップグレードするときに、データを Java DB 10.0 ディスク形式から 10.1 形式に変換してください。この変換を実行するには、JDBC の URL に `upgrade=true` を付加してデータベースに接続します。次に例を示します。

```
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;
```

アップグレードのロールバック

Release 5 アップグレードのロールバックは、バックアップされた Release 4 インストールとそのデータに戻すことによってのみ可能です。

複数インスタンスのアップグレード

配備のアーキテクチャーによっては、スケーラビリティーを実現し、可用性を向上させるために、Java DB が複数のコンピュータシステムに配備される場合があります。たとえば、ロードバランサを使用して Java DB インスタンスを複数のコンピュータで実行することにより、負荷を分散させることができます。

162 ページの「Release 4 の Java DB のアップグレード」の説明に従って、各コンピュータで Java DB のアップグレードを実行します。

第 9 章

High Availability Session Store

この章では、High Availability Session Store を Java ES 5 (Release 5): High Availability Session Store (HADB) 4.4.3 にアップグレードする方法について説明します。

この章では、アップグレードに伴う一般的な問題の概要を示し、Release 5 でサポートされている各種のアップグレードパスについて説明します。また、Solaris オペレーティングシステムと Linux オペレーティングシステムの両方でのアップグレードを取り上げます。

- 168 ページの「HADB のアップグレードの概要」
- 170 ページの「Java ES Release 4 からの HADB のアップグレード」
- 177 ページの「Java ES Release 3 からの HADB のアップグレード」

注 この章では、*HADB-base* というディレクトリパスを基準に指定されています。このパスの少なくとも一部は、HADB が最初にインストールされたときにインストールディレクトリとして指定されている可能性があります。指定されていない場合、インストーラによってデフォルト値が割り当てられています。

HADB-base のデフォルト値は、次の表に示すように、オペレーティングシステムプラットフォームに依存しません。

表 9-1 HADB ディレクトリパス

パス名変数	Solaris OS	Linux OS
*HADB-base*Java ES インストーラ	`/opt/SUNWhadb/`*version_number*	`/opt/SUNWhadb/`*version_number*
HADB-base スタンドアロンの Application Server 8.2 EE インストーラ	`/opt/SUNWappserver/appserver/hadb/`**`version_number`**	`/opt/SUNWappserver/appserver/hadb/`*version_number*

HADB のアップグレードの概要

ここでは、Java ES 5 (Release 5) へのアップグレードに影響する HADB の一般的な側面について説明します。

- Java ES Release 5 の HADB について
- HADB のアップグレードロードマップ
- HADB のデータ
- HADB のアップグレード方針

Java ES Release 5 の HADB について

Java ES Release 5 バージョンの HADB では、Release 4 の HADB に対してユーザーの拡張機能が若干強化されています。

HADB のアップグレードロードマップ

表 9-2 は、HADB を Java ES Release 5 へアップグレードする際の、サポートされるアップグレードパスを示しています。この表は Solaris および Linux の両方のオペレーティングシステムに適用されます。

表 9-2 Java ES 5 (Release 5): HADB 4.4.3 へのアップグレードパス

Java ES のリリース	HADB のバージョン	一般的な手順	必要な再設定
Release 4	HADB 4.4.2 (2005Q4)	直接アップグレード : オンラインアップグレードとオフラインアップグレードの両方が可能。	なし
Release 3	HADB 4.4.1 (2005Q1)	直接アップグレード : オンラインアップグレードとオフラインアップグレードの両方が可能。	なし
Release 2	HADB 4.4.0-14 (2004Q2)	アップグレードはサポート対象外	なし
Release 1	なし	アップグレード不可	なし
それ以前の Java ES リリース	なし	アップグレード不可	なし

HADB のデータ

次の表は、HADB ソフトウェアのアップグレードにより影響を受ける可能性のあるデータの種類を示しています。

表 9-3　HADB のデータの使用法

データの種類	場所	使用法
動的なアプリケーションデータ	/var/opt/SUNWhadb	High Availability Session Store
設定データ	/etc/opt/SUNWhadb /etc/init.d/ma-initd	高可用性サーバー設定

HADB のアップグレード方針

HADB をアップグレードするための方針は一般に、第 1 章「アップグレードの計画」で説明されている多くの考慮事項 (アップグレードパス、Java ES コンポーネント間の依存関係、選択的なアップグレードかすべてのアップグレードか、複数インスタンスの配備など) によって異なります。

ここでは、HADB のアップグレード計画に影響する可能性のある問題を取り上げ、HADB の全般的な説明を詳細に記述します。

互換性の問題

Release 5 の HADB は、Java ES Release 4 で提供される HADB と下位互換性があります。

HADB の依存関係

Release 5 の HADB は、J2SE 共有コンポーネント (Java™ 2 Platform, Standard Edition (J2SE™) バージョン 1.4 以降) に対してのみ依存します。

二重のアップグレード

二重のアップグレード (44 ページの「二重のアップグレード : Java ES ソフトウェアとオペレーティングシステムソフトウェア」で説明されているように、ここでは HADB ソフトウェアとオペレーティングシステムの両方をアップグレードすること) は、Application Server の二重のアップグレードの処理の中で実行されます。Application Server の二重のアップグレードについては、210 ページの「二重のアップグレード」を参照してください。

Java ES Release 4 からの HADB のアップグレード

ここでは、HADB の Java ES 2005Q4 (Release 4) から Java ES 5 (Release 5) へのアップグレードについて説明します。この節で説明する内容は、次のとおりです。

- はじめに
- Release 4 の HADB のアップグレード

はじめに

Java ES Release 4 HADB を Java ES Release 5 にアップグレードする場合は、アップグレード処理の次の側面を考慮してください。

- **一般的なアップグレード手順** : アップグレードは、Java ES Release 4 の HADB パッケージを削除し、Java ES Release 5 のパッケージを追加することによって行います。アップグレードは 2 とおりの方法で実行できます。
  - **オンラインアップグレード** : HADB サービスの中断を避けるには、オンラインアップグレードを使用します。
  - **オフラインアップグレード** : HADB パッケージを新しいバージョンに置き換えるときに HADB サービスを中断できる場合、オフラインアップグレードを使用します。
- **アップグレードの依存関係** : HADB には強いアップグレード依存関係はありません。HADB には J2SE バージョン 1.4 以降が必要です。つまり、J2SE に対して弱いアップグレード依存関係を持っています。
- **下位互換性** : Java ES Release 5 で提供される HADB は、Java ES Release 4 で提供される HADB と下位互換性があります。
- **アップグレードのロールバック** : Java ES の Release 5 アップグレードから Java ES Release 4 へのロールバックは、Release 4 バージョンのパッケージを復元することにより行われます。
- **プラットフォームの問題** : HADB をアップグレードするための一般的な手順は、Solaris および Linux の両オペレーティングシステムで共通です。

Release 4 の HADB のアップグレード

ここでは、Solaris プラットフォームと Linux プラットフォームの両方で、HADB の Java ES Release 4 から Java ES Release 5 へのアップグレードを実行する方法について説明します。この節で説明する内容は、次のとおりです。

- アップグレード前の作業
- Release 4 の HADB のアップグレード
- アップグレードの確認
- アップグレード後の作業
- アップグレードのロールバック

アップグレード前の作業

HADB ソフトウェアをアップグレードする前に、次の作業を実行してください。

- 現在のバージョン情報の確認
- HADB が依存するコンポーネントのアップグレード
- ディレクトリデータと設定ファイルのバックアップ
- 必要な設定情報とパスワードの入手

現在のバージョン情報の確認

HADB の現在のバージョンは、標準のバージョン確認ユーティリティーを使用することにより確認できます。次に例を示します。

Solaris の場合:

```
pkgparam -v SUNWhadba
```

Linux の場合:

```
rpm -qi sun-hadb-a-4.40.3-5.i386.rpm
```

表 9-4　HADB のバージョン確認の出力

Java ES のリリース	HADB のバージョン番号
Release 2	VERSION=4.4.0,REV=14
	SUNW_PRODVERS=4.4.0
Release 3	VERSION=4.4.1,REV=7
	SUNW_PRODVERS=4.4.1
Release 4	VERSION=4.4.2,REV=7
	SUNW_PRODVERS=4.4.2
Release 5	VERSION=4.4.3,REV=5
	SUNW_PRODVERS=4.4.3

HADB が依存するコンポーネントのアップグレード

一般的には、コンピュータシステム上 (および、コンピューティング環境内) のすべての Java ES コンポーネントを Java ES Release 5 にアップグレードすることが推奨されます。ただし、HADB の必要とするすべての共有コンポーネント (つまり J2SE) は、HADB の Release 5 へのアップグレードを実行するときに、Java ES インストーラによって自動的にアップグレードされます。

ディレクトリデータと設定ファイルのバックアップ

HADB を Java ES Release 4 から Java ES Release 5 にアップグレードしても、HADB の動的データは変更されません。ただし、アップグレードをロールバックする必要が生じた場合に備えて、Java ES Release 4 のパッケージをバックアップすることができます。

また、前回の HADB のインストール以降、次のファイルに対して何らかの変更を行なった場合は、これらをバックアップします。

```
/etc/opt/SUNWhadb/mgt.cfg
/etc/init.d/ma-initd
```

必要な設定情報とパスワードの入手

HADB のアップグレードには、スーパーユーザーのパスワードが必要です。

Release 4 の HADB のアップグレード

ここでは、HADB のアップグレード手順に影響する注意点を示したあと、その手順自体について説明します。

アップグレードの注意点

HADB ソフトウェアの Java ES Release 5 へのアップグレードでは、次の注意点を考慮に入れてください。

- 運用環境の要件に基づいて、オンラインアップグレードとオフラインアップグレードのどちらが適切かを決定する必要があります。
- 次の表に、Solaris および Linux の各プラットフォーム用の、Java ES Release 5 アップグレードパッケージを示します。Solaris 用のパッケージは、インストールする順に示しています。

表 9-5　Solaris プラットフォーム用 HADB アップグレードパッケージのバージョン

Solaris 用パッケージ	Linux 用パッケージ
SUNWhadba	sun-hadb-a--4.4.3-5.i386.rpm
SUNWhadbc	sun-hadb-c-4.40.3-5.i386.rpm
SUNWhadbe	sun-hadb-e-4.40.3-5.i386.rpm
SUNWhadbi	sun-hadb-i-4.40.3-5.i386.rpm
SUNWhadbj	sun-hadb-j-4.40.3-5.i386.rpm
SUNWhadbm	sun-hadb-m-4.40.3-5.i386.rpm
SUNWhadbs	sun-hadb-s-4.40.3-5.i386.rpm
SUNWhadbv	sun-hadb-v-4.40.3-5.i386.rpm
SUNWhadbx	sun-hadb-x-4.40.3-5.i386.rpm

HADB のオンラインアップグレード

HADB のオンラインアップグレードを実行するときは、アップグレード対象のクラスタ内の各サーバーに、Release 5 の HADB をインストールして、起動し、確認を行います。各サーバーはまず、HADB の以前のインストールから登録解除し、次に、新しくインストールされるバージョンの HADB に登録します。

オンラインアップグレードの実行の詳細については、『Sun Java System Application Server Enterprise Edition 8.2 高可用性 (HA) 管理ガイド』(http://docs.sun.com/app/docs/doc/820-0854/6ncikc4ei?l=ja&a=view) の該当する章を参照してください。

HADB のオフラインアップグレード

HADB のオフラインアップグレードは、Java ES Release 3 または Release 4 からのアップグレード時に実行できます。

オフラインアップグレードを実行するには、HADB サービスをシャットダウンし、既存の HADB パッケージを新しいバージョンと置き換えます。この新しいバージョンは、173 ページの表 9-5 に示しているように、Java ES 5 (Release 5) ディストリビューションに収められています。

1. root としてログインするか、スーパーユーザーになります。

   ```
   su -
   ```

2. すべての HADB サービスをシャットダウンします。

   a. 実行中のすべてのデータベースを一覧表示します。

      HADB-base`/bin/hadbm  list`

   b. 一覧表示された各データベースをシャットダウンします。

      HADB-base`/bin/hadbm  stop` *databaseName*

      データベースがすでに停止状態である場合は、メッセージを無視します。

   c. 管理エージェントを実行しているすべてのホストで、HADB 管理エージェントをシャットダウンします。

      `/etc/init.d/ma-initd  stop`

      (Java ES インストーラではなくスタンドアロンの Application Server インストーラを使用して HADB をインストールした場合、`ma-initd` スクリプトは *HADB-base*`/bin` にある)。

3. Java ES インストーラを起動します。

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

   ここで、*os_arch* は、`Solaris_sparc` などのプラットフォームに対応します。コマンド行インタフェースには、`installer -nodisplay` オプションを使用します。

   ようこそページと、ライセンス使用許諾契約ページが表示されてから、コンポーネント選択ページが表示されます。Java ES インストーラが直接アップグレードできるインストール済みのコンポーネントが検出された場合、それらのコンポーネントは「アップグレード可」の状態で表示されます。

4. コンポーネント選択ページで、「High Availability Session Store 4.4」を選択します。

   すでに「Application Server Enterprise Edition 8.2」を選択済みの場合は、「HADB」が自動的に選択されます。

5. アップグレードに関して選択した項目を確認します。

HADB パッケージがアップグレードされ、アップグレードの概要が表示されます。

6. Java ES インストーラを終了します。
7. 172 ページの「ディレクトリデータと設定ファイルのバックアップ」でバックアップしたファイルを復元します。
8. シンボリックリンク `/opt/SUNWhadb/4` が *HADB-base* をポイントしていることを確認します。

   たとえば、デフォルトの *HADB-base* は次のようになります。

   ```
   ls   -l   /opt/SUNWhadb/4
   lrwxrwxrwx  1 root   other  7 Jul 7 23:18 /opt/SUNWhadb/4 ->
       4.4.3-5/
   ```

9. 手順 2 でシャットダウンした HADB 管理エージェントを再起動します。

   ```
   /etc/init.d/ma-initd   start
   ```

   (Java ES インストーラではなくスタンドアロンの Application Server インストーラを使用して HADB をインストールした場合、`ma-initd` スクリプトは *HADB-base*`/bin` にある)。

アップグレードの確認

オンラインアップグレードの完了後、次の手順を使用してアップグレードを確認します。アップグレードの成功を確認したら、古いインストールパッケージは削除できます。

稼働中のプロセスがアップグレードされた HADB サービスを使用していることを確認するために、次の手順を実行できます。

1. 稼働中のすべての HADB サービスに対して、次のいずれかのコマンドを発行します。

   HADB-base`/bin/ma -V`
   HADB-base`/bin/hadbm -V`

   次に例を示します。

   ```
   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. データベースが稼働しているかどうかをチェックします。

   HADB-base`/bin/hadbm status -n` *databaseName*

たとえば、ExampleDB という名前のデータベースには、次のコマンドを入力します。

*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

一覧表示されたノードに対して、すべての HADB サービスが「running」状態であることを確認します。

3. HADB を使用するすべての製品が新しい HADB パスを使用していることを確認します。

   HADB-base/bin/hadbm get PackageName *databaseName*

   たとえば、ExampleDB という名前のデータベースには、次のコマンドを入力します。

   HADB-base/bin/hadbm get PackageName ExampleDB
   Attribute Value
   PackageName V4.4.3.5

   このコマンドは、HADB の現在のバージョンを表示します。詳細な一覧表示を得るには、次のコマンドを発行します。

   HADB-base/bin/hadbm get --all ExampleDB

アップグレード後の作業

170 ページの「Java ES Release 4 からの HADB のアップグレード」で説明している手順以外に、アップグレード後の作業はありません。

アップグレードのロールバック

HADB へのアップグレードをロールバックするには、172 ページの「ディレクトリデータと設定ファイルのバックアップ」の説明に従って、インストールした HADB パッケージの新しいバージョンを、以前にインストールしたバージョンと置き換えます。

Java ES Release 3 からの HADB のアップグレード

Java ES 2005Q1 (Release 3) HADB を Release 5 にアップグレードする手順は、Release 4 HADB を Release 5 にアップグレードする場合と同じです。

Release 3 HADB を Release 5 にアップグレードするには、「Release 4」を「Release 3」と読み替えた上で、170 ページの「Java ES Release 4 からの HADB のアップグレード」の手順を使用してください。

第 10 章

Message Queue

この章では、Message Queue ソフトウェアを、以前の Java ES バージョンから Java ES 5 (Release 5): Sun Java System Message Queue 3.7 UR1 にアップグレードする方法について説明します。

ここでは、Release 5 でサポートされている各種のアップグレードパスでのアップグレードの注意点の概要について説明します。また、Solaris オペレーティングシステムと Linux オペレーティングシステムの両方でのアップグレードを取り上げます。

- 180 ページの「Message Queue のアップグレードの概要」
- 185 ページの「Java ES Release 4 からの Message Queue のアップグレード」
- 191 ページの「Java ES Release 3 からの Message Queue のアップグレード」
- 192 ページの「Java ES Release 2 からの Message Queue のアップグレード」

注 この章で説明するファイルの場所は、*MessageQueue-base* というディレクトリパスを基準に指定されています。

MessageQueue-base の値は、次の表に示すように、オペレーティングシステムプラットフォームによって異なります。

表 10-1 Message Queue ディレクトリパス

パス名変数	Solaris OS	Linux OS
MessageQueue-*base*	`/usr/bin`	`/opt/sun/mq/bin`

Message Queue のアップグレードの概要

ここでは、Java ES 5 (Release 5) へのアップグレードに影響する Message Queue の一般的な側面について説明します。

- Java ES Release 5 の Message Queue について
- Message Queue のアップグレードロードマップ
- Message Queue のデータ
- Message Queue のアップグレード方針

Java ES Release 5 の Message Queue について

Java ES Release 5 Message Queue は、Release 4 に対するマイナーアップグレードです。大部分はコード修正で、マイナー機能強化はありません。

Message Queue ソフトウェアには、従来から Platform Edition と Enterprise Edition の 2 つのエディションがあり、エディション間で機能セットおよびライセンス範囲が異なります。Enterprise Edition は、企業の実運用環境でメッセージングアプリケーションを配備し、実行するためのエディションでした。Platform Edition は、メッセージングを伴うアプリケーションやコンポーネントの開発、デバッグ、および負荷テストを主な用途とするエディションでした。Release 5 の Message Queue では Platform Edition が非推奨で、Message Queue にはすべての Enterprise Edition 機能が含まれています。Java ES の旧リリースバージョンから Release 5 にアップグレードすると、インストール済みの Platform Edition はすべて Message Queue の完全なエンタープライズレベル機能に変換されます。

Message Queue のアップグレードロードマップ

表 10-2 は、Message Queue を Java ES Release 5 へアップグレードする際の、サポートされるアップグレードパスを示しています。この表は Solaris および Linux の両方のオペレーティングシステムに適用されます。

表 10-2 Java ES 5 (Release 5): Message Queue 3.7 UR1 へのアップグレードパス

Java ES のリリース	Message Queue のバージョン	一般的な手順	必要な再設定
Release 4	Sun Java System Message Queue 2005Q4 (3.6 SP3) Enterprise Edition のみ	直接アップグレード: Java ES インストーラを使用することにより実行されます。	自動的に実行されるデータ変換。
Release 3	Sun Java System Message Queue 2005Q1 (3.6) Enterprise Edition のみ	直接アップグレード: Java ES インストーラを使用することにより実行されます。	自動的に実行されるデータ変換。
Release 2	Sun Java System Message Queue 2004Q2 (3.5 SP1) Platform Edition および Enterprise Edition	直接アップグレード: `mqupgrade` スクリプトを使用することにより実行されます。	Solaris プラットフォームでは自動的に実行され、Linux プラットフォームでは `mqmigrate` スクリプトを使用できます。
Release 1	Sun Java System Message Queue 2003Q4 (3.0.1 SP2) Platform Edition および Enterprise Edition	直接アップグレードは動作保証なし: ただし、`mqupgrade` スクリプトを使用することにより実行されます。[1]	Solaris プラットフォームでは自動的に実行され、Linux プラットフォームでは `mqmigrate` スクリプトを使用できます。
それ以前の Java ES リリース	Sun Java System Message Queue 3.0.*x* およびそれ以前のバージョン Platform Edition および Enterprise Edition	直接アップグレードは動作保証なし: ただし、Java ES インストーラを使用することにより実行できます。	

1. `mqupgrade` スクリプトの実行前と実行後に、次のファイルをバックアップして復元する必要がある可能性があります。たとえば、(Solaris OS の場合): `/etc/imq/passwd` および `/etc/imq/accesscontrol.properties` を `/var/imq/instances/`*instanceName*`/etc/` に復元します。

表 10-2 に示した Java ES リリースの Message Queue に加えて、Message Queue は Solaris オペレーティングシステムソフトウェアにも付属します。付属するバージョンの Message Queue は、Java ES インストーラを使用して Release 5 にアップグレードできます。

Message Queue のデータ

Message Queue では、ほかの Java ES コンポーネントと同様に、さまざまな種類のデータを利用します。Message Queue のアップグレードでは、これらのデータをアップグレード後のバージョンに移行しなければならない場合があります。次の表は、Message Queue ソフトウェアのアップグレードにより影響を受ける可能性のあるデータの種類を示しています。

表 10-3 では、Solaris システムでのデータの位置を示しています。Linux システム上の場所も類似していますが、Message Queue のバージョンに依存します。

- Release 2: 表の `/imq` は `/opt/imq` に置き換えられます。
- Release 3 以降 : 表の `/imq` は `/opt/sun/mq` に置き換えられます。

詳細については、『Message Queue 3.7 UR1 管理ガイド』(http://docs.sun.com/app/docs/doc/820-0846/6ncico030?l=ja&a=view) を参照してください。

表 10-3 で、*instanceName* はデータの関連付けに使用される Message Queue ブローカインスタンスの名前を表します。

表 10-3 Message Queue のデータの使用法 (Solaris OS)

データカテゴリ	位置 (Solaris)	使用法
ブローカインスタンス設定プロパティー	`/var/imq/instances/`*instanceName*`/props/config.properties`	ブローカおよび関連サービスの設定
動的アプリケーションデータの持続的格納	Release 2、Release 3、および Release 4: `/var/imq/instances/`*instanceName*`/fs350/` Release 5: `/var/imq/instances/`*instanceName*`/fs370/` または JDBC アクセス可能なデータストア	メッセージ、宛先、永続サブスクリプション、トランザクション、およびその他の動的データを格納する
管理対象オブジェクト (オブジェクトストア)	任意のローカルディレクトリ または LDAP Directory Server	クライアントとブローカの接続の設定に使用されるオブジェクト
セキュリティー : ユーザーリポジトリ	`/var/imq/instances/`*instanceName*`/etc/passwd` または LDAP ディレクトリサーバー	認証と承認に使用されるユーザーデータを格納する
セキュリティー : アクセス制御ファイル (デフォルトの位置)	`/var/imq/instances/`*instanceName*`/etc/accesscontrol.properties`	宛先および関連機能へのユーザーアクセスを承認するルールを設定する
セキュリティー : passfile ディレクトリ (デフォルトの位置)	`/var/imq/instances/`*instanceName*`/etc/`	暗号化されたパスワード情報を格納する。

表 10-3　Message Queue のデータの使用法 (Solaris OS) (続き)

データカテゴリ	位置 (Solaris)	使用法
セキュリティー : ブローカのキーストアファイルの位置	/etc/imq/	セキュリティー保護されたメッセージングのための暗号化された証明書情報を格納する。

Message Queue のアップグレード方針

Message Queue をアップグレードするための方針は一般に、第 1 章「アップグレードの計画」で説明されている多くの問題 (アップグレードパス、Java ES コンポーネント間の依存関係、選択的なアップグレードかすべてのアップグレードか、複数インスタンスの配備など) によって異なります。

ここでは、Message Queue のアップグレード計画に影響する可能性のある問題を取り上げ、Message Queue の全般的な説明を詳細に記述します。

互換性の問題

Release 5 の Message Queue では、Release 3 および Release 4 との非互換性に関する新しい問題はありません。ただし、Release 2 およびそれ以前のバージョンについては重大な互換性の問題があります。これらについては、193 ページの「Release 2 の互換性の問題」で説明します。

さらに、一般的な規則として、Release 4 およびそれ以前の Message Queue ブローカと Release 5 の Message Queue のブローカをクラスタ内に混在させる場合は、マスターブローカを古いほうのリリースのブローカにし、クラスタは古いほうのリリースの Message Queue クラスタとして動作します。

Message Queue の依存関係

ほかの Java ES コンポーネントに対する Message Queue の依存関係が、Message Queue ソフトウェアのアップグレードおよび再設定の手順に影響を及ぼす可能性があります。たとえば、Message Queue のインタフェースまたは関数が変更された場合は、Message Queue が依存するコンポーネントのアップグレードされたバージョンが必要になる場合があります。このようなコンポーネントのアップグレードの必要性は、個々のアップグレードパスによって異なります。

Message Queue は、次の Java ES コンポーネントに依存します。

- **共有コンポーネント** : Message Queue は、特定の Java ES 共有コンポーネントに依存します (48 ページの表 1-9 を参照)。

- Directory Server: Message Queue は Directory Server に対して任意の依存関係を持っています。管理対象オブジェクトやユーザーデータをローカルではなく LDAP ディレクトリ (Directory Server) に格納するように Message Queue を設定できます。
- Web コンテナ : Message Queue は、クライアントとブローカの間で HTTP メッセージをサポートするために、Web Server、Application Server、またはサードパーティーの Web コンテナに対する任意の依存関係を持っています。
- データベース : Message Queue は、Message Queue 持続レイヤーに対して、フラットファイルのメッセージストアではなく JDBC アクセス可能なデータストアを提供するために、Java DB (またはサードパーティーのデータベース) に対する任意の依存関係を持っています。
- **Sun Cluster**: Message Queue は、高可用性のサポートを提供するために、Sun Cluster に対する任意の依存関係を持っています。

二重のアップグレード

44 ページの「二重のアップグレード : Java ES ソフトウェアとオペレーティングシステムソフトウェア」で説明されているように、二重のアップグレードでは Message Queue とオペレーティングシステムの両方をアップグレードします。次の 2 つのどちらかの方法で二重のアップグレードを実行できます。

- 新規オペレーティングシステムのインストール
- オペレーティングシステムのインプレースアップグレード

新規オペレーティングシステムのインストール

1. 既存の Message Queue データをバックアップします。

   基本的なデータの場所については、182 ページの「Message Queue のデータ」を参照してください。

2. 新しいオペレーティングシステムをインストールします。

   新しいシステム (または Solaris 10 ゾーン) でオペレーティングシステムのインストールを実行するか、または、既存のファイルシステムを消去することができます。

3. Release 5 の Message Queue をインストールします。
4. 手順 1 でバックアップした Message Queue データを復元または移行します。

   Linux で Release 2 の Message Queue からアップグレードする場合、データは Release 5 の場所に復元されます。

オペレーティングシステムのインプレースアップグレード

1. 既存の Message Queue データをバックアップします。

基本的なデータの場所については、182 ページの「Message Queue のデータ」を参照してください。

2. オペレーティングシステムをアップグレードします。

   このアップグレードでは、既存のファイルシステムはそのまま残されます。

3. Release 5 の Message Queue にアップグレードします。

   アップグレードパスについては、この章の該当する節を参照してください。アップグレードでは、既存の Message Queue データはそのまま残されます。

   Linux で Release 2 の Message Queue からアップグレードする場合、データは Release 5 の場所に移動されるはずです。

Java ES Release 4 からの Message Queue のアップグレード

ここでは、Message Queue の Java ES 2005Q4 (Release 4) から Java ES 5 (Release 5) へのアップグレードについて説明します。この節で説明する内容は、次のとおりです。

- はじめに
- Release 4 の Message Queue のアップグレード
- 複数インスタンスのアップグレード

はじめに

Java ES Release 4 Message Queue を Release 5 にアップグレードする場合は、アップグレード処理の次の側面を考慮してください。

- **一般的なアップグレード手順** : このアップグレードは、Java ES インストーラを使用することにより実行されます。インストーラは、設定データを Release 4 から自動的に移行します。また、Release 4 に関連付けられたすべての動的アプリケーションデータは、`imqbrokerd` を最初に実行したときに、自動的に変換されます。つまり、ファイルベースのストアの場合は、`fs350` ディレクトリの内容が新しい `fs370` ディレクトリにコピーされるということです。JDBC ストアの場合は、既存のデータベーステーブルに対して単純にバージョンのアップグレードが実行されます。
- **アップグレードの依存関係** : Message Queue は、いくつかの Java ES 共有コンポーネントに依存しています (48 ページの表 1-9 を参照)。Message Queue のアップグレードを実行すると、すべての共有コンポーネントは Java ES インストーラによって自動的に Release 5 にアップグレードされます。

  さらに、183 ページの「Message Queue の依存関係」で説明されているように、Release 5 の Message Queue は Java ES 製品コンポーネントに依存します。ただし、これらのコンポーネントのアップグレードは、Message Queue を Release 5 にアップグレードする上では必要ありません。
- **下位互換性** : Release 5 の Message Queue は、プロトコル、ブローカの互換性、管理対象オブジェクト、管理ツール、およびクライアントアプリケーションに関して、Release 4 と完全互換です。
- **アップグレードのロールバック** : Message Queue のアップグレードを Release 4 にロールバックするためのユーティリティーはありません。アップグレードされたコンポーネントを削除し、以前のバージョンおよび設定データを手動で復元する必要があります。
- **プラットフォームの問題** : Message Queue をアップグレードするための手順は、Solaris および Linux の両オペレーティングシステムで共通です。

Release 4 の Message Queue のアップグレード

この節では、Message Queue を Java ES Release 4 から Java ES Release 5 にアップグレードする方法について説明します。

- アップグレード前の作業
- Release 4 の Message Queue のアップグレード
- Message Queue のアップグレードの確認
- アップグレード後の作業

- アップグレードのロールバック

アップグレード前の作業

Message Queue ソフトウェアをアップグレードする前に、次の作業を実行してください。

- 現在のバージョン情報の確認
- Message Queue が依存するコンポーネントのアップグレード
- Message Queue のバックアップ

現在のバージョン情報の確認

システムにインストールされている Message Queue のバージョンおよびエディションは、次のように、-version オプションを使用して Message Queue ブローカを起動することによって確認できます。

```
imqbrokerd -version
```

表 10-4　Message Queue のバージョン確認の出力

Java ES のリリース	Message Queue のバージョン番号
Release 2	`Sun Java(tm) System Message Queue 3 2004Q2` `Version: 3.5`
Release 3	`Sun Java(tm) System Message Queue 3 2005Q1` `Version: 3.6`
Release 4	`Sun Java(tm) System Message Queue 3 2005Q4` `Version: 3.6 SP3`
Release 5	`Sun Java(tm) System Message Queue 3.7` `Version: 3.7 UR1`

Message Queue が依存するコンポーネントのアップグレード

一般には、コンピュータシステム上 (および、コンピューティング環境内) のすべての Java ES コンポーネントを Java ES Release 5 にアップグレードすることをお勧めします。Message Queue は、いくつかの共有コンポーネントに対してのみ強いアップグレード依存関係を持っています。

Message Queue が依存するコンポーネントをアップグレードするときは、Message Queue をアップグレードする前に、(すでにアップグレードした可能性があるものを省いて) 次の順序で実行してください。共有コンポーネントのアップグレードは、Java ES インストーラによって自動的に実行されます。

1. **共有コンポーネント** : Java ES 共有コンポーネントを Release 5 に同期する手順は、67 ページの「Java ES 共有コンポーネントのアップグレード」で説明されています。ただし、Message Queue の Release 5 へのアップグレードを実行すると、Message Queue で必要な共有コンポーネントはすべて、Java ES インストーラによって自動的にアップグレードされます。
2. **Sun Cluster (弱いアップグレード依存関係)**: Sun Cluster を Release 5 にアップグレードする手順は、79 ページの第 3 章「Sun Cluster ソフトウェア」で説明されています。
3. **Directory Server (弱いアップグレード依存関係)**: Directory Server を Release 5 にアップグレードする手順は、103 ページの第 5 章「Directory Server」で説明されています。
4. **Java DB (弱いアップグレード依存関係)**: Message Queue をアップグレードするときには、Release 5 Java DB の新規インストールを実行する必要があります。
5. **Web コンテナソフトウェア (弱いアップグレード依存関係)**: Web Server または Application Server をアップグレードする手順は、それぞれ133 ページの第 7 章「Web Server」および203 ページの第 11 章「Application Server」で説明されています。

Message Queue のバックアップ

アップグレードを実行する前には、常に本稼働環境のアプリケーションデータをバックアップすることをお勧めします。182 ページの表 10-3 で示している動的アプリケーションデータの持続的ストアの場所に、注意してください。

Release 4 の Message Queue のアップグレード

アップグレード手順は、次の手順で構成されます。

1. 稼働中の Message Queue クライアントアプリケーションをすべて停止します。

   Application Server 環境で Message Queue を使用している場合、Application Server もシャットダウンします。

2. 稼働中のすべてのブローカを停止します。管理ユーザー名とパスワードの入力を求められます。

   ```
   imqcmd shutdown bkr [-b hostName:port]
   ```

3. 各ブローカインスタンスに関連付けられた動的データ、Message Queue フラットファイルユーザーリポジトリ、および Message Queue アクセス制御ファイルを保持しない場合、次のコマンドを使ってこれらのデータを削除します。

   ```
   imqbrokerd -name instanceName -remove instance
   ```

   削除しない場合、動的データおよび設定情報は保持され、Release 5 の Message Queue で使用されます。

4. root としてログインします。

```
su -
```

5. Java ES インストーラを起動します。

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

   ここで、*os_arch* は、`Solaris_sparc` などのプラットフォームに対応します。コマンド行インタフェースには、`installer -nodisplay` オプションを使用します。

   ようこそページと、ライセンス使用許諾契約ページが表示されてから、コンポーネント選択ページが表示されます。Java ES インストーラが直接アップグレードできるインストール済みのコンポーネントが検出された場合、それらのコンポーネントは「アップグレード可」の状態で表示されます。

6. コンポーネント選択ページで、Message Queue を選択します。

7. アップグレードに関して選択した項目を確認します。

   Message Queue パッケージがアップグレードされ、アップグレードの概要が表示されます。

8. Java ES インストーラを終了します。

Message Queue のアップグレードの確認

アップグレード手順の完了後、`-version` オプションを使用して Message Queue ブローカを起動することにより、アップグレードが成功したことを確認します。

```
imqbrokerd -version
```

このコマンドは、Java ES のバージョン番号と、Message Queue 固有のバージョン番号を返します。

アップグレード後の作業

Message Queue HTTP トンネリングサーブレットを使用しており、Web コンテナをアップグレードした場合、サーブレットを新しい Web コンテナに再配備しなければならない場合があります。Release 4 と Release 5 で HTTP トンネリングサーブレットの変更はありません。HTTP サポートの詳細については、『Message Queue 3.7 UR1 管理ガイド』(`http://docs.sun.com/doc/820-0846?l=ja`) を参照してください。

また、アップグレードをロールバックする必要のないことが確認できている場合は、`fs350` ディレクトリにある Release 4 のファイルベースのデータストアを削除することもできます (182 ページの表 10-3 を参照)。

アップグレードのロールバック

Message Queue をアップグレード前の状態にロールバックするためのスクリプトは用意されていません。ロールバックは、次の手順に従って手動で実行する必要があります。

1. 稼働中の Message Queue クライアントアプリケーションをすべて停止します。
2. 稼働中のすべてのブローカを停止します。管理ユーザー名とパスワードの入力を求められます。

   `imqcmd shutdown bkr [-b` *hostName*`:`*port*`]`
3. 各ブローカインスタンスに関連付けられた動的データ、Message Queue フラットファイルユーザーリポジトリ、および Message Queue アクセス制御ファイルを削除する場合、次のコマンドを使ってこれらのデータを削除します。

   `imqbrokerd -name` *instanceName* `-remove instance`
4. root としてログインするか、スーパーユーザーになります。

   `su -`
5. インストールされている Message Queue パッケージの一覧を、次のコマンドを使って取得します。

   Solaris の場合:
   `pkginfo | grep -i "message queue"`

   Linux の場合:
   `rpm -qa | grep mq`
6. 次のコマンドを使って Message Queue パッケージを削除します。

   Solaris の場合:
   `pkgrm` *packageName*
   ここで、*packageName* は任意の Message Queue パッケージの名前です。複数のパッケージを削除するには、パッケージ名をスペースで区切って入力します。

   Linux の場合:
   `rpm -e --nodeps` *RPMName*
   ここで、*RPMName* は任意の Message Queue RPM コンポーネントの名前です。複数のコンポーネントを削除するには、RPM 名をスペースで区切って入力します。

   ほかの製品が Message Queue パッケージを使用している可能性があるため、パッケージの削除は慎重に行なってください。パッケージに依存関係が存在する場合、`pkgrm` コマンドはパッケージを削除する前にその旨をユーザーに警告します。プロンプトが表示されたら、**`y`** (yes) と入力して削除要求を確認します。
7. 「`q`」と入力して処理を終了します。
8. root シェルを終了します。
9. Release 4 の Message Queue を再インストールする

Java ES Release 4 のインストーラを使用します。

10. 188 ページの「Message Queue のバックアップ」でバックアップした Release 4 の Message Queue データを復元します。

    Release 4 の Message Queue は、Release 5 にアップグレードする前にバックアップしたデータに対しても正常に機能します。

複数インスタンスのアップグレード

Message Queue クラスタでは、複数のブローカを相互作用させることで、スケーラブルなメッセージサービスを実現できます。そのようなクラスタをアップグレードする場合、ローリングアップグレードと呼ばれる手法を使用できます。ローリングアップグレードでは、クラスタをオンラインに保ったまま、個々の Message Queue インスタンスを Release 4 から Release 5 に順次アップグレードできます。クラスタのアップグレードを実行するときは、次の 2 つの条件に留意してください。

- アップグレードのためにブローカをシャットダウンしている間、そのブローカが格納している永続メッセージは、ブローカが再起動されるまで取り出すことができません。
- マスターブローカは最後にアップグレードすることが推奨されます。

ローリングアップグレードを使用しない場合の手順は簡単です。すべてのブローカがアップグレードされるまで、ブローカを一度に 1 つずつシャットダウンし、アップグレードし、再起動します。

Java ES Release 3 からの Message Queue のアップグレード

Java ES 2005Q1 (Release 3) Message Queue を Release 5 にアップグレードする手順は、Release 4 Message Queue を Release 5 にアップグレードする場合と同じです。

Release 3 Message Queue を Release 5 にアップグレードするには、「Release 4」を「Release 3」と読み替えた上で、185 ページの「Java ES Release 4 からの Message Queue のアップグレード」の手順を使用してください。

Java ES Release 2 からの Message Queue のアップグレード

ここでは、Message Queue の Java ES 2004Q2 (Release 2) から Java ES Release 5 へのアップグレードについて説明します。ここで説明する内容は次のとおりです。

- はじめに
- Release 2 の互換性の問題
- Release 2 の Message Queue のアップグレード
- 複数インスタンスのアップグレード

注 Linux プラットフォームで Release 2 の Message Queue からアップグレードする場合は、二重のアップグレード、つまり、Message Queue とオペレーティングシステムの両方をアップグレードする必要があります (Release 5 の Message Queue は RHEL 2.1 でサポートされていない)。詳細については、184 ページの「二重のアップグレード」を参照してください。

はじめに

Java ES Release 2 Message Queue を Release 5 にアップグレードする場合は、アップグレード処理の次の側面を考慮してください。

- **一般的なアップグレード手順** : アップグレードは `mqupgrade` スクリプトを使用して行います。このスクリプトは、古いソフトウェアパッケージを新しいパッケージに置き換え、設定データを Release 2 から自動的に移行します。
- **アップグレードの依存関係** : コンピュータ上の任意の Java ES コンポーネントを Release 2 からアップグレードするには、そのコンピュータでホストされているほかのすべての Java ES コンポーネントをアップグレードする必要があります。Java ES コンポーネントの Release 2 から Release 5 への選択的なアップグレードはサポートされていません。特に、Message Queue で使用されるすべての Java ES 共有コンポーネントをアップグレードする必要があります。

  Release 5 の Message Queue は、183 ページの「Message Queue の依存関係」で説明するように、Directory Server および Web Server (または Application Server) に依存する場合があります。これらが同じコンピュータ上にインストールされている場合、これらのコンポーネントの Release 5 へのアップグレードも必要です。
- **下位互換性** : 次の「Release 2 の互換性の問題」で説明されているように、Release 5 の Message Queue は Release 2 と完全互換ではありません。
- **アップグレードのロールバック** : Release 5 から Release 2 へのロールバックは、現在サポートされていません (「アップグレードのロールバック」を参照)。

- **プラットフォームの問題** : Message Queue をアップグレードするための一般的な手順は、Solaris および Linux の両オペレーティングシステムで共通ですが、Linux 上ではいくつかの追加手順が必要です。以降の手順説明では、必要に応じてプラットフォーム別のコマンド、ファイル位置、または手順を示します。

Release 2 の互換性の問題

Release 5 の Message Queue では、Release 2 以前のバージョンに対する Message Queue の互換性に関して、次のような一般的な問題が存在しています。

プロトコルの互換性

Message Queue は、Message Queue クライアントとブローカの間で HTTP プロトコルサポートを提供するために Web コンテナに依存します。Message Queue の `imqhttp.war` アプリケーションに対して Web コンテナを提供する目的で Sun Java System Web Server を使用しているとき、Web Server コンポーネントをアップグレードするには Message Queue もアップグレードする必要があります。これは、プロトコルの変更が原因です (189 ページおよび 200 ページを参照)。

ブローカの互換性

Release 5 の Message Queue ブローカは Release 4、Release 3、および Release 2 のブローカと相互運用可能ですが、Release 2 と比較した場合、ブローカプロパティーと永続的ストアスキーマが変更されており、このことが互換性に影響を及ぼす可能性があります。

Release 5 の Message Queue は Release 4、Release 3、および Release 2 のデータを使用できます。ただし、Linux システムでは Release 2 のデータをまず Release 5 に移行する必要があります。

Release 5 の Message Queue にアップグレードするときは、次の事項を考慮してください。

- Message Queue の古い `config.properties` ファイルを使用できます。また、それらのファイルを別の場所にコピーし、Release 5 の Message Queue ブローカを設定するときに、それらのファイルのプロパティー設定を参考にすることもできます。
- メッセージ、宛先、永続サブスクリプションなどの持続的 Message Queue データはすべて、ブローカの初回起動時に必要に応じて Release 5 の Message Queue データに自動的に変換されます。たとえば、必要な場合、既存の宛先がすべて Release 5 の Message Queue 宛先に変換されます。その際、既存の属性は保持され、新しい属性についてはデフォルト値が使用されます。

- 1 つのクラスタ内に Message Queue Release 2 のブローカと Message Queue Release 5 のブローカを混在させる場合は、Message Queue Release 2 のブローカ (古いほう) をマスターブローカにする必要があり、そのようなクラスタは Message Queue Release 2 クラスタとして動作します。

管理対象オブジェクトの互換性

Release 5 の Message Queue の管理対象オブジェクトは、Release 3 および Release 4 の管理対象オブジェクトと同一です。ただし、Release 3 の管理対象オブジェクトの中には、以前のバージョンから名前が変更されているものや、新しい属性が追加されているものがあります。したがって、Release 2 の Message Queue から Release 5 にアップグレードするときは、次の事項を考慮する必要があります。

- Release 2 で作成したオブジェクトストアおよび管理対象オブジェクトはそのままでも使用できますが、管理対象オブジェクトを Release 5 に移行することをお勧めします。更新操作を実行すると、管理コンソール (`imqadmin`) および ObjectManager コマンド行ユーティリティー (`imqobjmgr`) によって、Release 2 の管理対象オブジェクトが Release 5 の管理対象オブジェクトに変換されます。
- Release 5 のクライアントランタイムは、Release 2 の管理対象オブジェクトを検索してインスタンス化し、Release 5 クライアントで使用できるようにそれらを変換します。ただし、検索実行元のオブジェクトストア内にある Release 2 の管理対象オブジェクトは変換対象にはなりません。
- 管理対象オブジェクトを検索するのではなく直接インスタンス化する、アプリケーションやコンポーネントなどの既存の Release 2 クライアントは、Release 5 と互換性があります。ただし、そのようなクライアントで管理対象オブジェクトの新しい属性を使用する場合、クライアントのコードを変更する必要があります。管理対象オブジェクトの属性の詳細については、『Message Queue 3.7 UR1 管理ガイド』(`http://docs.sun.com/doc/820-0846?l=ja`) の第 16 章を参照してください。Release 2 のクライアントを Release 5 で再コンパイルすると、Message Queue Release 2 の属性のうち、どの属性の名前が Release 5 で変更されているかが示されます。古い名前も引き続き正しく機能します。
- Java クライアントを起動するスクリプトや、コマンド行オプションを使用して管理対象オブジェクトの属性値を設定するスクリプトは、Release 5 と互換性があります。ただし、そのようなクライアントで管理対象オブジェクトの新しい属性を使用する場合、クライアントのコードを変更する必要があります。管理対象オブジェクトの属性の詳細については、『Message Queue 3.7 UR1 管理ガイド』(`http://docs.sun.com/doc/820-0846?l=ja`) の第 16 章を参照してください。

管理ツールの互換性

Release 3 で新しいコマンドおよび管理機能が追加されたことにより、Release 5 の管理ツール (管理コンソールとコマンド行ユーティリティー) は Release 3、Release 4 および Release 5 のブローカに対してのみ正常に機能します。ただし、Release 2 のすべてのコマンドおよびコマンドオプションも引き続きサポートされます。

クライアントの互換性

Release 3 および Release 4 のクライアントは Release 5 の Message Queue と完全に互換性があります。ただし、Release 2 から Release 5 にアップグレードするときは、Java クライアントの互換性に関する次の問題を考慮する必要があります。

- Release 5 のブローカは Release 2 のクライアントをサポートします (ただし、Release 5 で追加された機能はサポートしない)。
- Release 5 の Java クライアントは Release 2 のブローカに接続できます (ただし、Release 5 で追加された機能は使用できない)。
- C クライアントのプログラムは、Release 5 ブローカと、Enterprise Edition ライセンスまたは Platform Edition の試用ライセンスで実行されている Release 2、Release 3、または Release 4 ブローカでサポートされます。

Release 2 の Message Queue のアップグレード

この節では、Message Queue を Java ES Release 2 から Java ES Release 5 にアップグレードする方法について説明します。

- アップグレード前の作業
- Release 2 の Message Queue のアップグレード (Solaris)
- Release 2 の Message Queue のアップグレード (Linux)
- 互換性パッケージのインストール (Linux)
- Message Queue のアップグレードの確認
- アップグレード後の作業
- アップグレードのロールバック

アップグレード前の作業

Message Queue ソフトウェアをアップグレードする前に、次の作業を実行してください。

- 現在のバージョン情報の確認
- Message Queue が依存するコンポーネントのアップグレード
- Message Queue のバックアップ

現在のバージョン情報の確認

システムにインストールされている Message Queue のバージョンおよびエディションは、次のように、`-version` オプションを使用して Message Queue ブローカを起動することによって確認できます。

```
imqbrokerd -version
```

表 10-5　Message Queue のバージョン確認の出力

Java ES のリリース	Message Queue のバージョン番号
Release 2	`Sun Java(tm) System Message Queue 3 2004Q2 Version: 3.5`
Release 3	`Sun Java(tm) System Message Queue 3 2005Q1 Version: 3.6`
Release 4	`Sun Java(tm) System Message Queue 3 2005Q4 Version: 3.6 SP3`
Release 5	`Sun Java(tm) System Message Queue 3.7 Version: 3.7 UR1`

Message Queue が依存するコンポーネントのアップグレード

一般には、コンピュータシステム上 (および、コンピューティング環境内) のすべての Java ES コンポーネントを Java ES Release 5 にアップグレードすることをお勧めします。Message Queue は、いくつかの共有コンポーネントに対してのみ強いアップグレード依存関係を持っています。

Message Queue が依存するコンポーネントをアップグレードするときは、Message Queue をアップグレードする前に、(すでにアップグレードした可能性があるものを省いて) 次の順序で実行してください。

1. 共有コンポーネント : Java ES 共有コンポーネントを Release 5 にアップグレードする手順は、67 ページの第 2 章「Java ES 共有コンポーネントのアップグレード」で説明されています。
2. Sun Cluster (弱いアップグレード依存関係): Sun Cluster を Release 5 にアップグレードする手順は、79 ページの第 3 章「Sun Cluster ソフトウェア」で説明されています。
3. Directory Server (弱いアップグレード依存関係): Directory Server を Release 5 にアップグレードする手順は、103 ページの第 5 章「Directory Server」で説明されています。
4. Java DB (弱いアップグレード依存関係): Message Queue をアップグレードするときには、Release 5 Java DB の新規インストールを実行する必要があります。
5. Web コンテナソフトウェア (弱いアップグレード依存関係): Web Server または Application Server をアップグレードする手順は、それぞれ 133 ページの第 7 章「Web Server」および 203 ページの第 11 章「Application Server」で説明されています。

Message Queue のバックアップ

アップグレードを実行する前には、常に本稼働環境のアプリケーションデータをバックアップすることをお勧めします。Solaris OS の場合、動的データは次のディレクトリに格納されます。
/var/imq/instances/*instanceName*

その他のオペレーティングシステムについては、『Message Queue 3.7 UR1 管理ガイド』(http://docs.sun.com/doc/820-0846?l=ja) を参照してください。

Release 2 の Message Queue のアップグレード (Solaris)

Message Queue ソフトウェアの Java ES Release 5 へのアップグレードでは、mqupgrade スクリプトを利用します。このスクリプトは Release 5 パッケージをインストールします。

アップグレード手順は、次の手順で構成されます。

1. 稼働中の Message Queue クライアントアプリケーションをすべて停止します。

   Application Server 環境で Message Queue を使用している場合、Application Server もシャットダウンします。

2. 稼働中のすべてのブローカを停止します。管理ユーザー名とパスワードの入力を求められます。

   ```
   imqcmd shutdown bkr [-b hostName:port]
   ```

3. 各ブローカインスタンスに関連付けられた動的データ、Message Queue フラットファイルユーザーリポジトリ、および Message Queue アクセス制御ファイルを保持しない場合、次のコマンドを使ってこれらのデータを削除します。

   ```
   imqbrokerd -name instanceName -remove instance
   ```

   削除しない場合、動的データおよび設定情報は保持され、Release 5 の Message Queue で使用されます。

4. root としてログインします。

   su -

5. Java ES Release 5 ディストリビューションの Tools ディレクトリに移動します。

   Solaris SPARC の場合:
   ```
   cd Solaris_sparc/Product/message_queue/Tools
   ```

   Solaris x86 の場合:
   ```
   cd Solaris_x86/Product/message_queue/Tools
   ```

6. mqupgrade スクリプトを実行します。

a. スクリプトを起動します。

```
./mqupgrade
```

mqupgrade スクリプトは、インストールされている Message Queue コンポーネントを一覧表示します。

b. y (yes) と入力して、Message Queue コンポーネントをアップグレードします。

mqupgrade スクリプトは、インストールされているローカライズ版ファイルを検出して一覧表示します。

Message Queue コンポーネントをアップグレードしない場合は、n (no) と入力します。mqupgrade スクリプトは、Message Queue コンポーネントをアップグレードせずに終了します。

c. プロンプトに対し、y (yes) と入力してローカライズ版ファイルをアップグレードします。

mqupgrade スクリプトは、次の場所にあるログファイルに出力を送信します。

```
/var/sadm/install/logs/Message_Queue_upgrade_'date'.log
```

Release 2 の Message Queue のアップグレード (Linux)

Linux プラットフォームでの Release 2 の Message Queue から Release 5 へのアップグレードは複雑です。Java ES Release 2 は RHEL 2.1 でのみサポートされているのに、Java ES Release 5 は RHEL 2.1 でサポートされていないからです。そこで、二重のアップグレードが必要になります。つまりオペレーティングシステムと Message Queue の両方をアップグレードしてください。44 ページの「二重のアップグレード : Java ES ソフトウェアとオペレーティングシステムソフトウェア」を参照してください。

基本的な手順は、最初に Linux オペレーティングシステムをアップグレードして、次に Message Queue が依存するすべてのコンポーネントをアップグレードし、それから Message Queue をアップグレードします。

Release 2 の Message Queue から Release 5 へのアップグレードには、Solaris システムでは必要のないデータ移行手順、つまりブローカインスタンスデータを適切な Release 5 の場所に移行する手順が含まれます。Release 5 へのアップグレードで Release 2 のデータを保持したい場合、Message Queue には、この移行を実行するための移行ツール mqmigrate が用意されています。

Release 2 から Release 5 にアップグレードするには、197 ページの「Release 2 の Message Queue のアップグレード (Solaris)」で使用したのと同じ手順を使用します。ただし、以降の手順内で説明するように、mqupgrade スクリプト (199 ページの手順 6) を実行する前に mqmigrate スクリプト (199 ページの手順 7) を実行する必要があります。

1. 稼働中の Message Queue クライアントアプリケーションをすべて停止します。
2. 稼働中のすべてのブローカを停止します。管理ユーザー名とパスワードの入力を求められます。

   `imqcmd shutdown bkr [-b` *hostName*`:`*port*`]`
3. 各ブローカインスタンスに関連付けられた動的データ、Message Queue フラットファイルユーザーリポジトリ、および Message Queue アクセス制御ファイルを保持しない場合、次のコマンドを使ってこれらのデータを削除します。

   `imqbrokerd -name` *instanceName* `-remove instance`

   削除しない場合、動的データおよび設定情報は保持され、Release 5 の Message Queue で使用されます。
4. root としてログインするか、スーパーユーザーになります。

   su -
5. Java ES Release 5 ディストリビューションの `Tools` ディレクトリに移動します。

   `cd Linux_x86/Product/message_queue/Tools`
6. 次のコマンドを使ってブローカインスタンスデータを移行します。

   `./mqmigrate`

   `mqmigrate` スクリプトは、Release 2 のブローカインスタンス設定データを Release 4 の適切な位置に移動します。
7. `mqupgrade` スクリプトを実行します。
   a. スクリプトを起動します。

      `./mqupgrade`

      `mqupgrade` スクリプトは、インストールされている Message Queue コンポーネントを一覧表示します。
   b. `y` (yes) と入力して、Message Queue コンポーネントをアップグレードします。

      `mqupgrade` スクリプトは、インストールされているローカライズ版ファイルを検出して一覧表示します。

      Message Queue コンポーネントをアップグレードしない場合は、`n` (no) と入力します。`mqupgrade` スクリプトは、Message Queue コンポーネントをアップグレードせずに終了します。
   c. プロンプトに対し、`y` (yes) と入力してローカライズ版ファイルをアップグレードします。

      `mqupgrade` スクリプトは、次の場所にあるログファイルに出力を送信します。

      `/var/sadm/install/logs/Message_Queue_upgrade_'`*date*`'.log`

互換性パッケージのインストール (Linux)

Release 5 のインストール済みファイルの位置に依存するスクリプトがあるか、Release 2 のクライアントアプリケーションにそのようなスクリプトが含まれる場合、sun-mq-compat パッケージをインストールする必要があります。このパッケージには、Release 2 のファイル位置から Release 5 のファイル位置へのシンボリックリンクが含まれます。

sun-mq-compat パッケージは、Java ES Release 5 ディストリビューションの展開先の次の位置にあります。

```
Linux_x86/Product/message_queue/Packages
```

次の手順を実行して、sun-mq-compat パッケージをインストールします。

1. root としてログインするか、スーパーユーザーになります。

   ```
   su -
   ```

2. Packages ディレクトリから次のコマンドを入力します。

   ```
   rpm -ivh --nodeps sun-mq-compat-3.7-RelNo.i386.rpm
   ```

Message Queue のアップグレードの確認

アップグレード手順の完了後、-version オプションを使用して Message Queue ブローカを起動することにより、アップグレードが成功したことを確認します。

```
imqbrokerd -version
```

このコマンドは、Java ES のバージョン番号と、Message Queue 固有のバージョン番号を返します。

アップグレード後の作業

HTTP 接続サービスのサポートを提供するために HTTP トンネリングサーブレットを使用している場合、Release 2 から Release 5 への Message Queue のアップグレードによってサーブレットがアップグレードされています。そのため、Message Queue を Release 5 にアップグレードしたあとでサーブレットを再配備する必要があります。HTTP サポートの詳細については、『Message Queue 3.7 UR1 管理ガイド』(http://docs.sun.com/doc/820-0846?l=ja) を参照してください。

さらに、管理コンソール (imqadmin) や ObjectManager コマンド行ユーティリティー (imqobjmgr) を使用して更新操作を実行し、Release 2 の管理対象オブジェクトを Release 5 バージョンに移行する必要があります。

アップグレードのロールバック

Message Queue の Release 2 から Release 5 へのアップグレードは、現在サポートされていません。通常、このアップグレードの手順は Release 5 から Release 4 へのロールバック (190 ページの「アップグレードのロールバック」を参照) と同じになります。しかし、Message Queue の Release 2 から Release 5 へのアップグレードでは、Java ES 製品のレジストリが更新されないため、Java ES インストーラは Release 2 の Message Queue を再インストールできません。

この問題に対する回避策については、Sun Services までお問い合わせください。

複数インスタンスのアップグレード

Message Queue クラスタでは、複数のブローカを相互作用させることで、スケーラブルなメッセージサービスを実現できます。そのようなクラスタをアップグレードする場合、ローリングアップグレードと呼ばれる手法を使用できます。ローリングアップグレードでは、クラスタをオンラインに保ったまま、個々の Message Queue インスタンスを Release 2 から Release 5 に順次アップグレードできます。クラスタのアップグレードを実行するときは、次の 2 つの条件に留意してください。

- アップグレードのためにブローカをシャットダウンしている間、そのブローカが格納している永続メッセージは、ブローカが再起動されるまで取り出すことができません。
- マスターブローカは最後にアップグレードすることが推奨されます。

ローリングアップグレードを使用しない場合の手順は簡単です。すべてのブローカがアップグレードされるまで、ブローカを一度に 1 つずつシャットダウンし、アップグレードし、再起動します。

第 11 章

Application Server

この章では、Application Server を Java ES 5 (Release 5): Sun Java System Application Server Enterprise Edition 8.2 にアップグレードする方法について説明します。

ここでは、Release 5 でサポートされている各種のアップグレードパスでのアップグレードの注意点の概要について説明します。また、Solaris オペレーティングシステムと Linux オペレーティングシステムの両方でのアップグレードを取り上げます。

- 204 ページの「Application Server のアップグレードの概要」
- 211 ページの「Java ES Release 4 からの Application Server のアップグレード」
- 216 ページの「Java ES Release 3 からの Application Server のアップグレード」
- 217 ページの「Java ES Release 2 からの Application Server のアップグレード」
- 226 ページの「Solaris 10 の複数ゾーン環境における Solaris 付属の Application Server のアップグレード」

注 この章では、*AppServer8-base* および *AppServer8Config-base* (Application Server 8.*x* の場合)、または *AppServer7-base* および *AppServer7Config-base* (Application Server 7.*x* の場合) というディレクトリパスを基準に、ファイルの場所が指定されています。これらのパスの少なくとも一部は、Application Server がインストールされたときにインストールディレクトリまたはドメインディレクトリとして指定されている場合があります。指定されていない場合、Java ES インストーラによってデフォルト値が割り当てられています。次の表に、これらのディレクトリパスのデフォルト値を示します。

表 11-1 Application Server ディレクトリパス

パス名変数	Solaris OS	Linux OS
AppServer8-base	`/opt/SUNWappserver/appserver`	`/opt/sun/appserver`

表 11-1　Application Server ディレクトリパス (続き)

パス名変数	Solaris OS	Linux OS
AppServer8Install-base	/opt/SUNWappserver	/opt/sun/appserver
AppServer8Config-base	/var/opt/SUNWappserver	/var/opt/sun/appserver
AppServer7-base	/opt/SUNWappserver7	/opt/SUNWappserver7
AppServer7Config-base	/var/opt/SUNWappserver7	/var/opt/SUNWappserver7

注

Release 5 の Application Server (8.*x*) のデフォルトの Application Server のドメイン名は domain1 です。言い換えると、domain1 は、この章で使用する *domainName* 変数のデフォルト値ということになります。

また、デフォルトの Directory Administration Service (DAS) のインスタンス名は、server です。J2EE アプリケーションは通常、DAS インスタンスには配備されずに、ほかのスタンドアロンインスタンスに配備されます。

この章で使用する asadmin コマンドの詳細については、『Sun Java System Application Server Enterprise Edition 8.2 Reference Manual』(http://docs.sun.com/doc/819-4736) を参照してください。

Application Server のアップグレードの概要

ここでは、Java ES 5 (Release 5) へのアップグレードに影響する Application Server の一般的な側面について説明します。

- Java ES Release 5 の Application Server について
- Application Server のアップグレードロードマップ
- Application Server のデータ
- Application Server のアップグレード方針

Java ES Release 5 の Application Server について

Java ES Release 5 Application Server は、Release 4 に対するマイナーリリースで、選ばれたバグ修正のみが含まれています。Release 5 の Application Server は、機能的には Release 4 と同じです。

Application Server のアップグレードロードマップ

Application Server の Java ES Release 5 へのアップグレードに適用するアップグレードパスは、次の 2 とおりがあります。

- 表 11-2 は、Java ES Application Server の、Java ES Release 5 へのサポートされているアップグレードパスを示しています。この表は Solaris および Linux の両方のオペレーティングシステムに適用されます。
- 表 11-3 は、Solaris に付属する Application Server の、Java ES Release 5 へのサポートされているアップグレードパスを示しています。Application Server Platform Edition は、Solaris オペレーティングシステムソフトウェアに付属しています。表 11-3 のとおり、付属するバージョンの Application Server は、Java ES インストーラを使用して Release 5 Enterprise Edition にアップグレードできます。

表 11-2 Java ES 5 (Release 5): Application Server Enterprise Edition 8.2 へのアップグレードパス

Java ES のリリース	Application Server のバージョン	一般的な手順	必要な再設定
Release 4	Sun Java System Application Server Enterprise Edition 8.1 2005Q4	直接アップグレード: Java ES インストーラを使用することにより実行されます。[1]	なし
Release 3	Sun Java System Application Server Enterprise Edition 8.1 2005Q1	直接アップグレード: Java ES インストーラを使用することにより実行されます。	なし
Release 2	Sun Java System Application Server 7.0 Update 3 (2004Q2) Platform Edition および Enterprise Edition	直接アップグレード: Java ES インストーラを使用してパッケージをアップグレードし、`postInstall` スクリプトを使用して再設定を行い、さらに `asupgrade` ユーティリティーを使用してドメイン情報を移行します。[2]	環境変数、ドメイン、およびその他の設定データ。 J2EE コンポーネントおよび J2EE アプリケーションを新しい Application Server 環境に移行し、再配備する必要があります。
Release 1	Sun ONE Application Server 7.0 Update 1 (2003Q4) Platform Edition および Enterprise Edition	直接アップグレードは動作保証なし: Release 2 の Application Server からのアップグレードと同じ手順を使用します。	環境変数、ドメイン、およびその他の設定データ。 J2EE コンポーネントおよび J2EE アプリケーションを新しい Application Server 環境に移行し、再配備する必要があります。

表 11-2 Java ES 5 (Release 5): Application Server Enterprise Edition 8.2 へのアップグレードパス (続き)

Java ES のリリース	Application Server のバージョン	一般的な手順	必要な再設定
それ以前の Java ES リリース		直接アップグレード不可 : まず Release 3 にアップグレードする必要があります。手順については、『Sun Java Enterprise System 2005Q1 アップグレードと移行』(http://docs.sun.com/source/819-2235?l=ja) を参照してください。 次に、Release 3 から Release 5 にアップグレードします。	

1. 設定情報やドメイン情報を保持せずに Application Server を Release 4 から Release 5 にアップグレードする場合は、Java ES Release 4 アンインストーラを使用して Release 4 の Application Server をアンインストールしてから、Java ES Release 5 のインストーラを使用して Release 5 の Application Server を新規インストールできます。ただし、Release 4 の Application Server が「あとで設定」オプションを使用してインストールされている場合は、Release 4 の Application Server をアンインストールする前に、まず *$HOME/.asadminprefs* ファイル (この *$HOME* は、Application Server をインストールして実行するユーザー用のホームディレクトリ) を作成します。このファイルには、次の 2 行があります。
   ```
   AS_ADMIN_PASSWORD=password
   AS_ADMIN_USER=admin
   ```
2. Solaris プラットフォーム上で Release 2 の Application Server からアップグレードを行う場合は、特別な配慮が必要です。これは、Solaris 付属の Application Server と Java ES Release 2 の Application Server がコンピュータ上に共存するためです。表 11-3 と、それに続く注を参照してください。

表 11-3 Solaris オペレーティングシステムに付属する Application Server バージョンのアップグレードパス

Solaris OS バージョン	Application Server のバージョン	一般的な手順	必要な再設定
Solaris 10	Sun Java System Application Server Platform Edition 8.0.0_01	直接アップグレード : Java ES インストーラを使用することにより実行されます。	なし。

表 11-3 Solaris オペレーティングシステムに付属する Application Server バージョンのアップグレードパス (続き)

Solaris OS バージョン	Application Server のバージョン	一般的な手順	必要な再設定
Solaris 9	Sun Java System Application Server 7.0.0_03c	直接アップグレード : Application Server がすでに使用され、ドメインが作成済みの場合、ドメイン情報を移行する必要があります。Release 2 の Application Server からのアップグレードで説明されている手順を使用します。そこでは、Java ES インストーラの「あとで設定」オプション、`postInstall` スクリプト、および `asupgrade` ユーティリティーを使用します。 Application Server が未使用であるか、ドメインを作成せずに使用された場合、Release 2 の Application Server からのアップグレードで説明されている手順を使用しますが、Java ES インストーラの「今すぐ設定」オプションを使用して、`postInstall` スクリプトや `asupgrade` ユーティリティーを使用しない点が異なります。	環境変数、ドメイン (作成済みの場合)、およびその他の設定データ。 J2EE コンポーネントおよび J2EE アプリケーションを新しい Application Server 環境に移行し、再配備する必要があります。

注 Solaris プラットフォーム上で Release 2 の Application Server からアップグレードを行う場合は、特別な配慮が必要です。これは、Solaris 付属のバージョンと Java ES バージョンの両方の Application Server がコンピュータ上に共存するためです。そのため、Release 2 バージョンからのアップグレードを始める前に、まず Solaris 付属バージョン (および、それに対応するすべてのドメイン) をアンインストールしてください。219 ページの「Solaris のみ : オペレーティングシステムに付属している Application Server パッケージを手動で削除します。」を参照してください。

Application Server のデータ

次の表は、Application Server ソフトウェアのアップグレードにより影響を受ける可能性のあるデータの種類を示しています。

表 11-4 Application Server のデータの使用法

データの種類	場所	使用法
環境変数	*AppServer8-base*/config/asenv.conf	グローバル変数
設定データ	Release 3、Release 4、および Release 5: 次の場所にある domain.xml および server.policy ファイル *AppServer8Config-base*/domains/*domainName*/config	Application Server インスタンスの設定
	Release 2: 次の場所にある server.xml および server.policy ファイル *AppServer7Config-base*/domains/*domainName*/*instanceName*/config	
配備データ	Release 3、Release 4、および Release 5: *AppServer8Config-base*/domains/*domainName*/applications	特定の J2EE コンポーネントおよび J2EE アプリケーション用の J2EE コンテナの設定
	Release 2: *AppServer7Config-base*/domains/*domainName*/*instanceName*/applications	
アクセスログファイル	Release 3、Release 4、および Release 5: *AppServer8Config-base*/domains/*domainName*/logs/access/ 含まれる 2 つのファイル: server_acces_log および _asadmin_access_log	ロギングのアクセス
	Release 2: *AppServer7Config-base*/domains/*domainName*/*instanceName*/logs/access	

Application Server のアップグレード方針

Application Server をアップグレードするための方針は一般に、第 1 章「アップグレードの計画」で説明されている多くの考慮事項 (アップグレードパス、Java ES コンポーネント間の依存関係、選択的なアップグレードかすべてのアップグレードか、複数インスタンスの配備など) によって異なります。

ここでは、Application Server のアップグレード計画に影響する可能性のある問題を取り上げ、Application Server の全般的な説明を詳細に記述します。

互換性の問題

Release 5 の Application Server では、Release 4 または Release 3 からのインタフェースの変更はありません。ただし、Release 5 と Release 2 ではインタフェースが大きく変更されているため、Release 5 は Release 2 と互換性がありません。

Release 5 の Application Server は、しかしながら Release 4 の Service Registry をサポートしていません。Application Server を Release 5 にアップグレードする場合は、Service Registry も Release 5 にアップグレードしてください。

Application Server の依存関係

ほかの Java ES コンポーネントに対する Application Server の依存関係が、Application Server ソフトウェアのアップグレードおよび再設定の手順に影響を及ぼす可能性があります。たとえば、Application Server のインタフェースまたは機能の変更が原因で、Application Server が依存するコンポーネントのバージョンのアップグレードが必要になる場合があります。このようなコンポーネントのアップグレードの必要性は、個々のアップグレードパスによって異なります。

Application Server は、次の Java ES コンポーネントに依存します。

- **共有コンポーネント** : Application Server は、特定の Java ES 共有コンポーネントに依存します (48 ページの表 1-9 を参照)。
- Message Queue: Application Server は、J2EE Java Message Service 互換の非同期メッセージングサポートを提供するために Message Queue を利用します。
- High Availability Session Store: Application Server で、インスタンス間のフェイルオーバーをサポートするために必要なセッション状態情報を維持するためには High Availability Session Store (HADB) が必要です。

注	HADB を必要としない以前の Application Server をインストールしている場合は、Application Server を Release 5 にアップグレードする前に、HADB をインストールしてください。たとえば、Solaris オペレーティングシステムに付属の Application Server のバージョンをアップグレードするには、Release 5 の HADB をインストールしてください。これらの状況では、Java ES インストーラによって HADB のインストールが自動的に行われます。

- Java DB: Application Server は、デフォルトの開発者データベースとして Java DB に依存し、サンプルアプリケーションデータや Enterprise Java Beans タイマーによって必要とされるデータを格納します。
- Web コンテナ (任意の依存関係): Application Server でオプションのロードバランスプラグインを使用する場合、Web コンテナが必要です。Web コンテナには、Java ES Web Server またはサードパーティー製 Web コンテナ (Apache Web Server や Microsoft IIS など) を使用できます。

二重のアップグレード

44 ページの「二重のアップグレード : Java ES ソフトウェアとオペレーティングシステムソフトウェア」で説明されているように、二重のアップグレードでは Application Server とオペレーティングシステムの両方をアップグレードします。次の 2 つのどちらかの方法で二重のアップグレードを実行できます。

- 新規オペレーティングシステムのインストール
- オペレーティングシステムのインプレースアップグレード

新規オペレーティングシステムのインストール

1. 既存の Application Server データをバックアップします。

   基本的なデータの場所については、208 ページの「Application Server のデータ」を参照してください。

2. 新しいオペレーティングシステムをインストールします。

   オペレーティングシステムのインストールは同じコンピュータでのみ実行でき、既存のファイルシステムは消去されます。

3. 手順 1 でバックアップした Application Server データを復元します。
4. Release 5 の Application Server をインストールします。

   手順 1 でバックアップした Application Server データのバージョンに応じて、この章のアップグレードに関する該当する節に記載されている手順を使用します。

オペレーティングシステムのインプレースアップグレード

1. 既存の Application Server データをバックアップします。

   基本的なデータの場所については、208 ページの「Application Server のデータ」を参照してください。

2. オペレーティングシステムをアップグレードします。

   このアップグレードでは、既存のファイルシステムはそのまま残されます。

3. Release 5 の Application Server にアップグレードします。

   アップグレードパスに応じて、この章の該当する節を参照してください。

Java ES Release 4 からの Application Server のアップグレード

ここでは、Application Server の Java ES 2005Q4 (Release 4) から Java ES 5 (Release 5) へのアップグレードについて説明します。この節で説明する内容は、次のとおりです。

- はじめに
- Release 4 の Application Server のアップグレード

はじめに

Java ES Release 4 Application Server を Release 5 にアップグレードする場合は、アップグレード処理の次の側面を考慮してください。

- **一般的なアップグレード手順** : このアップグレードは、Java ES インストーラを使用することにより実行されます。Release 4 の Application Server から Release 5 へのアップグレードでは、Application Server の再設定および J2EE コンポーネントの再設定または移行は必要ありません。
- **アップグレードの依存関係** : Application Server は多くの Java ES 共有コンポーネントに依存しますが (48 ページの表 1-9 を参照)、これらはすべて、Application Server のアップグレードを実行したときに Java ES インストーラによって Release 5 に自動的にアップグレードされます。Application Server は、NSS 共有コンポーネントに対してのみ強いアップグレード依存関係を持っています。

  さらに、209 ページの「Application Server の依存関係」で説明されているように、Release 5 の Application Server は、Message Queue、HADB、および Java DB に依存します。これらは強いアップグレード依存関係なので、すべてを Release 5 にアップグレードしてください。

また、Application Server は任意で Java ES Web Server またはサードパーティー製の Web コンテナに依存します。ただし、これらはアップグレード依存関係が弱いものであり、Web コンテナのアップグレードは、Application Server の Release 5 へのアップグレードに関しては省略可能です。

- **下位互換性**: Release 5 の Application Server は、Release 4 バージョンと下位互換性があります。
- **アップグレードのロールバック**: Release 5 へのアップグレードは、Release 4 にロールバックできません。
- **プラットフォームの問題**: Application Server をアップグレードするための一般的な手順は、Solaris および Linux の両オペレーティングシステムで共通です。

Release 4 の Application Server のアップグレード

ここでは、Solaris プラットフォームと Linux プラットフォームの両方で、Application Server の Java ES Release 4 から Java ES Release 5 へのアップグレードを実行する方法について説明します。プラットフォーム固有の手順に依存する項目には、その手順が適用されるオペレーティングシステムが示されています。この節で説明する内容は、次のとおりです。

- アップグレード前の作業
- Release 4 の Application Server のアップグレード
- アップグレードの確認
- アップグレード後の作業
- アップグレードのロールバック

アップグレード前の作業

Application Server ソフトウェアをアップグレードする前に、次の作業を実行してください。

- 現在のバージョン情報の確認
- Application Server が依存するコンポーネントのアップグレード
- Application Server のデータのバックアップ
- 必要な設定情報とパスワードの入手

現在のバージョン情報の確認

Application Server の現在のバージョンは、次のコマンドを入力することにより確認できます。

AppServer8-base/bin/asadmin version --verbose

表 11-5　Application Server のバージョン確認の出力

Java ES のリリース	Application Server のバージョン番号
Release 2	Sun ONE Application Server 7.0.0_03c
Release 3	Sun Java Enterprise System Application Server Enterprise Edition 8.1
Release 4	Sun Java Enterprise System Application Server Enterprise Edition 8.1_02
Release 5	Sun Java Enterprise System Application Server Enterprise Edition 8.2

Application Server が依存するコンポーネントのアップグレード

一般には、コンピュータシステム上 (および、コンピューティング環境内) のすべての Java ES コンポーネントを Java ES Release 5 にアップグレードすることをお勧めします。Application Server は、NSS 共有コンポーネントと、Message Queue、HADB、および Java DB 製品コンポーネントに対して強いアップグレード依存関係を持っています。

Application Server が依存するコンポーネントをアップグレードするときは、Application Server をアップグレードする前に、(すでにアップグレードした可能性があるものを省いて) 次の順序で実行してください。ただし、共有コンポーネントや Message Queue、HADB、および Java DB のアップグレードは通常、Application Server をアップグレードするときに Java ES インストーラによって自動的に実行されます。

1. **共有コンポーネント** : Java ES 共有コンポーネントを Release 5 に同期する手順は、67 ページの「Java ES 共有コンポーネントのアップグレード」で説明されています。ただし、Application Server の Release 5 へのアップグレードを実行すると、Application Server で必要な共有コンポーネントはすべて、Java ES インストーラによって自動的にアップグレードされます。
2. Message Queue: Message Queue を Release 5 にアップグレードする手順は、179 ページの第 10 章「Message Queue」で説明されています。
3. High Availability Session Store (HADB): HADB をアップグレードする手順は、167 ページの第 9 章「High Availability Session Store」で説明されています。
4. Java DB: Java DB をアップグレードする手順は、157 ページの第 8 章「Java DB」で説明されています。

5. **Web コンテナソフトウェア (弱いアップグレード依存関係):** Web Server をアップグレードする手順は、133 ページの第 7 章「Web Server」 で説明されています。

Application Server のデータのバックアップ

Application Server の Release 4 から Release 5 へのアップグレードでは、設定データは変更されません。したがって、現在のデータをバックアップする必要はありません。

必要な設定情報とパスワードの入手

現在インストールされているバージョンの Application Server の管理者ユーザー ID とパスワードを調べておきます。

Release 4 の Application Server のアップグレード

ここでは、Application Server のアップグレード手順に影響する注意点を示したあと、その手順自体について説明します。

アップグレードの注意点

Application Server ソフトウェアの Java ES Release 5 へのアップグレードでは、次の注意点を考慮に入れてください。

- Application Server インスタンスをアップグレードする前に、そのインスタンス内で稼働しているすべての J2EE コンポーネントをシャットダウンすることが推奨されます。ただし、高可用性またはスケーラビリティーを実現するためにロードバランスを使用している場合、この要件は緩和できます。
- 1 台のコンピュータ上で複数の Application Server インスタンスが稼働しており、すべてのインスタンスがインストール済みの同じ Application Server イメージに対応している場合、インストール済みのイメージのアップグレード中はすべてのインスタンスをシャットダウンしてください。
- 複数ノード配備の場合、Application Server インスタンスが動作している個々のノードまたはコンピュータ上でアップグレード手順を実行します。

アップグレード手順

ここで説明する手順は、アップグレードを実施するコンピュータにローカルインストールされている Application Server インスタンスに適用されます。

1. root としてログインするか、スーパーユーザーになります。

   ```
   su -
   ```

2. アップグレードする Application Server インスタンス内のすべての J2EE コンポーネントをシャットダウンします。
3. アップグレードするコンピュータ上のすべての Application Server インスタンスをシャットダウンします。

a. 実行中のすべてのノードエージェントを停止します。

AppServer8-base/bin/asadmin stop-node-agent --user *admin_ID* *nodeagentName*

ここで、*nodeagentName* の形式は *hostName_domainName* ですが、デフォルトでは単に *hostName* です。

b. ドメイン管理サーバー (DAS) を停止します。

AppServer8-base/bin/asadmin stop-domain --user *admin_ID* *domainName*

c. PointBase データベースサーバーを停止します (使用されている場合)。

AppServer8Config-base/appserver/pointbase/tools/stopserver.sh

4. Java ES インストーラを起動します。

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

ここで、*os_arch* は、Solaris_sparc などのプラットフォームに対応します。コマンド行インタフェースには、installer -nodisplay オプションを使用します。

ようこそページと、ライセンス使用許諾契約ページが表示されてから、コンポーネント選択ページが表示されます。Java ES インストーラが直接アップグレードできるインストール済みのコンポーネントが検出された場合、それらのコンポーネントは「アップグレード可」の状態で表示されます。

5. コンポーネント選択ページで、Application Server を選択します。

強いアップグレード依存関係があるので、Message Queue、HADB、および Java DB も自動的にアップグレード対象に選択されます。

6. 「今すぐ設定」を選択します。

7. 要求される設定値を指定します。

いくつかの設定パネルが表示されます。

8. アップグレードに関して選択した項目を確認します。

Application Server パッケージ (および、必要に応じて Message Queue、HADB、および Java DB のパッケージ) がアップグレードされ、アップグレードの概要が表示されます。

9. Java ES インストーラを終了します。

10. アップグレードしたドメイン管理サーバー (DAS) を再起動します。

AppServer8-base/bin/asadmin start-domain --user *admin_ID* *domainName*

11. アップグレードした Application Server インスタンスを再起動します。

AppServer8-base/bin/asadmin start-node-agent --user *admin_ID*
nodeagentName

ここで、*nodeagentName* の形式は *hostName_domainName* ですが、デフォルトでは単に *hostName* です。

アップグレードの確認

アップグレードが成功したかどうかは、次のコマンドを使用して確認できます。

AppServer8-base/bin/asadmin version --verbose

出力値については、213 ページの表 11-5 を参照してください。

アップグレード後の作業

新しい Release 5 の Java ES 製品コンポーネントである Java DB ではなく、組み込まれた Release 4 の PointBase データベースを使用して作業を続行する場合、アップグレードした *AppServer8-base*/config/asenv.conf ファイルを手動で編集してください。

Application Server を Release 4 からアップグレードしたあと、asenv.conf 内の PointBase 設定は次のようになっています。

```
AS_POINTBASE="%POINTBASE_HOME%"
AS_POINTBASE_SAMPLESDB="%POINTBASE_SAMPLESDB%"
```

これらの設定を、次の値に変更します。

AS_POINTBASE="*AppServer8-base*/pointbase"
AS_POINTBASE_SAMPLESDB="*AppServerConfig8-base*/var/appserver/pointbase"

アップグレードのロールバック

Release 5 アップグレードのロールバックはサポートされていません。

Java ES Release 3 からの Application Server のアップグレード

Java ES 2005Q1 (Release 3) Application Server を Release 5 にアップグレードする手順は、Release 4 Application Server を Release 5 にアップグレードする場合と同じです。

Release 3 Application Server を Release 5 にアップグレードするには、「Release 4」を「Release 3」と読み替えた上で、211 ページの「Java ES Release 4 からの Application Server のアップグレード」の手順を使用してください。

Java ES Release 2 からの Application Server のアップグレード

ここでは、Java ES Release 2 から Java ES 5 (Release 5) への Application Server のアップグレードについて説明します。この節で説明する内容は、次のとおりです。

- はじめに
- Release 2 の Application Server のアップグレード

注 Linux プラットフォームで Release 2 の Application Server からアップグレードする場合は、二重のアップグレード、つまり、Application Server とオペレーティングシステムの両方をアップグレードする必要があります (Release 5 の Application Server は RHEL 2.1 でサポートされていない)。詳細については、210 ページの「二重のアップグレード」を参照してください。

はじめに

Java ES Release 2 Application Server を Release 5 にアップグレードする場合は、アップグレード処理の次の側面を考慮してください。

- **一般的なアップグレード手順** : Java ES インストーラで「あとで設定」オプションを選択して Release 5 の Application Server をインストールすることによってアップグレードを実行します。その後、`asupgrade` ユーティリティーを使用して再設定を行います。Application Server をアップグレードしたあとで、Release 2 の J2EE コンポーネントと J2EE アプリケーションを Release 5 に移行する必要があります。
- **アップグレードの依存関係** : コンピュータ上の任意の Java ES コンポーネントを Release 2 からアップグレードするには、そのコンピュータでホストされているほかのすべての Java ES コンポーネントをアップグレードする必要があります。Java ES コンポーネントの Release 2 から Release 5 への選択的なアップグレードはサポートされていません。

  特に、Application Server で必要とされるすべての Java ES 共有コンポーネントをアップグレードする必要があります。同じコンピュータに Message Queue がインストールされている場合は、これもアップグレードしてください。High Availability Session Store (HADB) および Java DB をインストールしてください。さらに、負荷分散用に Web Server を使用している場合、これもアップグレードしてください。
- **下位互換性** : Release 5 の Application Server は、Release 2 バージョンとの下位互換性はありません。Release 5 の Application Server 環境で実行するためには、J2EE コンポーネントおよび J2EE アプリケーションを移行する必要があります。

- **アップグレードのロールバック** : Release 5 へのアップグレードは、Release 2 のインストールを復元するだけで Release 2 にロールバックすることができます。アップグレードプロセスでは、Release 2 の設定データは削除されません。
- **プラットフォームの問題** : Solaris オペレーティングシステム上で Application Server をアップグレードするための手順を、直接 Linux オペレーティングシステムに適用することはできません。Java ES Release 2 は Linux2.1 でのみサポートされていて、Java ES Release 5 は Linux2.1 ではサポートされていないからです。そこで、二重のアップグレードが必要になります。つまりオペレーティングシステムと Application Server の両方をアップグレードしてください (44 ページの「二重のアップグレード : Java ES ソフトウェアとオペレーティングシステムソフトウェア」を参照)。
- **データの格納場所** : 208 ページの表 11-4 に記載されているとおり、Release 5 には、設定データ、配備データ、およびアクセスログファイルの新しい場所があります。

Release 2 の Application Server のアップグレード

ここでは、Solaris プラットフォームと Linux プラットフォームの両方で、Application Server の Java ES Release 2 から Java ES Release 5 へのアップグレードを実行する方法について説明します。プラットフォーム固有の手順に依存する項目には、その手順が適用されるオペレーティングシステムが示されています。この節で説明する内容は、次のとおりです。

- アップグレード前の作業
- Release 2 の Application Server のアップグレード
- アップグレードの確認
- アップグレード後の作業
- アップグレードのロールバック

アップグレード前の作業

Application Server をアップグレードする前に、次の作業を実行してください。

- Solaris のみ : オペレーティングシステムに付属している Application Server パッケージを手動で削除します。
- 現在のバージョン情報の確認
- Application Server が依存するコンポーネントのアップグレード
- Application Server のデータのバックアップ
- 必要な設定情報とパスワードの入手

Solaris のみ : オペレーティングシステムに付属している Application Server パッケージを手動で削除します。

Release 2 の Application Server のインストールでは、Solaris オペレーティングシステムに付属する Application Server のバージョンが削除されませんでした。Release 2 の Application Server を Release 5 に正しくアップグレードするには、次の手順を実行します。

1. /usr/appserver ディレクトリにある対応する Application Server パッケージを手動で削除することで、付属するバージョンをアンインストールします。

   ```
   pkgrm SUNWascmnse SUNWaslb SUNWasut ...
   ```

   ここで、次のコマンドを使用すると、パッケージの完全な一式を取得できます。

   ```
   pkginfo -i|grep -i "application server"
   ```

   この実行結果には、次のようなパッケージが含まれます。

   SUNWasacee、SUNWascmnse、SUNWasu、SUNWasuee、SUNWasac、SUNWascmn、SUNWasdb、SUNWasdem、SUNWasdemdb、SUNWasr、SUNWasut、SUNWasman、SUNWasjdoc

   さらに、次のようなローカライズ版パッケージが含まれることもあります。

   SUNW*Locale*asacee, SUNW*Locale*ascmnse, SUNW*Locale*asu, SUNW*Locale*asuee

2. 付属するバージョンがすでに使用されている場合は、ドメイン情報を削除します。

   ```
   rm /usr/appserver/domains
   ```

現在のバージョン情報の確認

Application Server の現在のバージョンは、次のコマンドを入力することにより確認できます。

```
AppServer7-base/bin/asadmin version --verbose
```

出力されるバージョン情報については、213 ページの表 11-5 を参照してください。

Application Server が依存するコンポーネントのアップグレード

Application Server が依存するアップグレードでは、Application Server が依存する、ローカルにあるすべての Release 2 のコンポーネント (特に Message Queue と共有コンポーネント) を Release 5 にアップグレードする必要があります。ただし、共有コンポーネントは、アップグレード手順の一部として、Java ES インストーラによって自動的にアップグレードされます (221 ページの「アップグレード手順」を参照)。

さらに、Application Server のアップグレードでは、Release 5 の Application Server が依存する HADB および Java DB がインストールされている必要があります。ただし、これら製品のコンポーネントは、アップグレード手順の一部として、Java ES インストーラによって自動的に選択され、アップグレードされます (221 ページの「アップグレード手順」を参照)。

一般的に、Application Server の依存関係をアップグレードする場合は、Application Server をアップグレードする前に、(すでにアップグレードした可能性があるものを省いて) 次の順序で実行してください。ただし、共有コンポーネントや HADB および Java DB のインストールのアップグレードは、Application Server をアップグレードするときに Java ES インストーラによって自動的に実行されます。

1. **共有コンポーネント** : Java ES 共有コンポーネントを Release 5 に同期する手順は、67 ページの「Java ES 共有コンポーネントのアップグレード」で説明されています。ただし、Application Server の Release 5 へのアップグレードを実行すると、Application Server で必要な共有コンポーネントはすべて、Java ES インストーラによって自動的にアップグレードされます。

   注 共有コンポーネントの Release 5 へのアップグレードには、Release 2 の Application Server ではサポートされない、J2SE の JDK バージョン 1.5 へのアップグレードも含まれます。したがって、共有コンポーネントをアップグレードする前に、Release 2 の Application Server をシャットダウンするようにしてください。共有コンポーネントの Release 5 へのアップグレードは、Java ES インストーラを使用していずれかの Java ES 製品コンポーネント (Directory Server、Message Queue, HADB、Java DB、およびその他のコンポーネント) をインストールまたはアップグレードすると、常に実行されます。

2. Message Queue: 192 ページの「Java ES Release 2 からの Message Queue のアップグレード」を参照してください。Java ES インストーラを使用して Message Queue を Release 2 からアップグレードすることはできません。
3. High Availability Session Store (HADB): Application Server をアップグレードすると、HADB の新規インストールが Java ES インストーラによって実行されます。
4. Java DB: Application Server をアップグレードすると、Java DB の新規インストールが Java ES インストーラによって実行されます。
5. **Web コンテナソフトウェア (弱いアップグレード依存関係)**: 155 ページの「Java ES Release 2 からの Web Server のアップグレード」を参照してください。

Application Server のデータのバックアップ

Application Server を Release 2 から Release 5 にアップグレードしても、Release 2 の設定データは上書きされません。ただし、不測の事態に備えて、Release 5 へのアップグレードを実行する前に、`domains` ディレクトリ全体をバックアップすることが推奨されます。

AppServer7Config-base`/domains`

必要な設定情報とパスワードの入手

現在インストールされているバージョンに関して、次の情報を確認してください。

- Application Server 管理者のユーザー ID とパスワード
- Release 2 の Application Server のベースディレクトリ

Release 2 の Application Server のアップグレード

ここでは、Application Server のアップグレード手順に影響する注意点を示したあと、その手順自体について説明します。

アップグレードの注意点

Application Server ソフトウェアの Java ES Release 5 へのアップグレードでは、次の注意点を考慮に入れてください。

- Application Server インスタンスをアップグレードする前に、そのインスタンス内で稼働しているすべての J2EE コンポーネントをシャットダウンすることが推奨されます。ただし、ロードバランスプラグインを使用して高可用性またはスケーラビリティーを提供する場合、この要件は緩和できる場合があります。
- 1 台のコンピュータ上で複数の Application Server インスタンスが稼働しており、すべてのインスタンスがインストール済みの同じ Application Server イメージに対応している場合、インストール済みのイメージをアップグレードしている間はすべてのインスタンスをシャットダウンする必要があります。

アップグレード手順

次に示す手順は、アップグレードを実施するコンピュータにローカルインストールされているすべての Application Server インスタンスに適用されます。

1. root としてログインするか、スーパーユーザーになります。

   ```
   su -
   ```

2. Java ES 共有コンポーネントが Release 5 にアップグレード済み (特に J2SE が JDK 1.5 にアップグレード済み) で、Release 2 の Application Server がまだ稼働中であった場合、`asenv.conf` ファイルを編集して、JDK 1.4 を直接ポイントするようにします。

   ```
   AS_JAVA=/usr/java
   ```

   `asenv.conf` が JDK 1.5 を参照している場合、Application Server をシャットダウン (下の手順 3) することはできません。

3. すべての Application Server と関連プロセスを停止します。

   AppServer7-base`/bin/asadmin stop-appserv` *domainName*

4. Release 5 の Application Server をインストールします。

   次の手順を実行します。

a. Release 2 の Application Server がホストされたコンピュータ上で Java ES インストーラを起動します。

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

ここで、*os_arch* は、`Solaris_sparc` などのプラットフォームに対応します。コマンド行インタフェースには、`installer -nodisplay` オプションを使用します。

ようこそページと、ライセンス使用許諾契約ページが表示されてから、コンポーネント選択ページが表示されます。Java ES インストーラが直接アップグレードできるインストール済みのコンポーネントが検出された場合、それらのコンポーネントは「アップグレード可」の状態で表示されます。Application Server は、Release 2 から「アップグレード可」であるとは表示されません。

b. コンポーネント選択ページから Application Server を選択します。

最低限、「ノードエージェント」コンポーネントを含む最初の 3 つのサブコンポーネントをインストールするように選択します。

c. 「次へ」をクリックします。

Message Queue がまだ Release 5 へアップグレードされていない場合は、Message Queue をアップグレードするように求めるエラーメッセージのウィンドウが表示されます。この場合は「OK」をクリックして、Message Queue をアップグレードします。アップグレードの手順は、195 ページの「Release 2 の Message Queue のアップグレード」で説明されています。

d. Release 2 の Application Server がインストールされていたのとは異なるインストールディレクトリを指定します。

e. 「あとで設定」オプションを選択します。

注	Solaris 9 オペレーティングシステムに付属する Application Server のバージョンがあり、このバージョンをまだ一度も使用していない (ドメイン情報が作成されていない) 場合は、「今すぐ設定」を選択し、手順 5 を省略して次の手順に進むことができます。

f. 必要に応じて、ローカライズ版のパッケージをインストールするオプションを選択します。

g. インストールが完了したら、Java ES インストーラを終了します。

5. 次のインストール後の作業を実行します。

a. Java ES Release 5 ディストリビューションの Application Server ツールディレクトリで、postInstall readme ファイルを探します。

Java ES Release 5 distribution/*os_arch*/Product/application_svr/Tools

ここで、*os_arch* は、Solaris_sparc などのプラットフォームに対応します。

b. ReadMe ファイルを参照して、postInstall スクリプトを実行します。

cd *Java ES Release 5 distribution*/*os_arch*/Product/application_svr/Tools
./postInstall *AppServer8Install-base* *AppServer8Config-base*

スクリプトは、インストールの間にインストールされたテンプレートを基に、*AppServer8-base*/bin/* シェルスクリプトおよび config/asenv.conf ファイルを構成および作成します。通常、bin/* シェルスクリプトは Java ES インストーラによって作成されますが、「あとで設定」オプションを選択した場合は説明に従って作成する必要があります。

注 Application Server を Release 5 にアップグレードするときに、非管理ポートの 8080 (デフォルト) と 8181 (セキュリティー保護) がドメイン管理サーバー (DAS) に割り当てられます。これらのポート割り当ては、Release 2 の Application Server インスタンスに対して割り当てられたポートと競合する可能性があります。その場合は、競合を回避するために DAS ポートを変更してください。手順については、『Sun Java System Application Server Enterprise 8.2 Edition 管理ガイド』(http://docs.sun.com/doc/820-0850?l=ja) の HTTP 待機属性の変更方法を示した節を参照してください。

---

c. 必要に応じて、*AppServer8-base*/config/asenv.conf ファイル内の環境設定を変更します。

ファイルを手動で編集する必要があります。

注 負荷分散のために Application Server を設定する方法については、『Sun Java System Application Server Enterprise Edition 8.2 高可用性 (HA) 管理ガイド』(http://docs.sun.com/doc/820-0854?l=ja) の「Application Server High Availability Features」の章の「Configuring Web Servers for HTTP Load Balancing」を参照してください。

---

6. asupgrade ユーティリティーを実行します。

asupgrade ユーティリティーは、Release 5 のノードエージェントを作成し、その下に Release 2 の Application Server インスタンスを移行します。

ユーティリティーは Application Server ディレクトリの配下にあります。次に例を示します。

- アップグレードウィザードモード : *AppServer8-base*/bin/asupgrade
- アップグレードコンソールモード : *AppServer8-base*/bin/asupgrade -c

アップグレードウィザードまたはアップグレードコンソールによって、アップグレード手順が案内されます。

a. ドメイン情報の移行のためのターゲットディレクトリとソースディレクトリの両方を識別します。

   - Release 2: *AppServer7-base*
   - Release 5: *AppServerConfig8-base*/domains

b. Release 2 の Application Server の管理ユーザー、管理パスワード、およびマスターパスワードを入力します。

   マスターパスワードの入力を求められたら、changeit と指定し、changeit 以外のマスターパスワードを持つキーストアを使用している場合は、パスワードを changeit に変更するようにしてください。

   jdk-home/bin/keytool -storepasswd -new changeit -keystore *keystore* -storepass *oldpasswd*

Application Server の asupgrade ユーティリティーの詳細については、『Sun Java System Application Server Enterprise Edition 8.2 Reference Manual』(http://docs.sun.com/doc/819-4736) のユーザーコマンドの節と、『Sun Java System Application Server Enterprise Edition 8.2 アップグレードと移行』(http://docs.sun.com/app/docs/doc/820-1612/6nd978eqa?l=ja&a=view) を参照してください。

7. 221 ページの手順 2 で asenv.conf ファイルをリダイレクトした場合は、JDK 1.5 をポイントするように戻します。

   AS_JAVA=/usr/jdk/entsys-j2se

   asenv.conf が JDK 1.4 を参照している場合、Application Server を起動できません (下の手順 8 および手順 9)。

8. ドメイン管理サーバー (DAS) を起動します。

   AppServer8-base/bin/asadmin start-domain --user *admin_ID* *domainName*

9. アップグレードした Application Server インスタンスを再起動します。

   これは、アップグレードした Application Server がその下に移行されたノードエージェントを起動することによって行います。

   AppServer8-base/bin/asadmin start-node-agent --user *admin_ID* *nodeagentName*

ここで、*nodeagentName* の形式は *hostName_domainName* ですが、デフォルトでは単に *hostName* です。

アップグレードの確認

管理コンソールを起動し、これらのサーバーが起動されていることを確認します。稼働していないサーバーがある場合、次のログをチェックして、ポートの競合によるエラーが発生していないかを調べます。

AppServer8Config-base/nodeagents/*nodeagentName*/*instanceName*/logs/server.log

ポートの競合によるエラーが発生していた場合、管理コンソールを使ってポート番号を変更して競合を解消したあと、ノードエージェントをいったん停止し、再起動します。

Release 5 への Application Server のアップグレードが成功したかどうかは、次のコマンドを入力することによって確認できます。

AppServer8-base/bin/asadmin version --verbose

出力値については、213 ページの表 11-5 を参照してください。

アップグレード後の作業

221 ページの「アップグレード手順」で説明している手順以外に、アップグレード後の作業が 2 つ必要です。

- JSS への参照の修正
- Release 2 の J2EE コンポーネントの移行

JSS への参照の修正

Application Server が JSS 共有コンポーネントの正しいバージョンを参照するために、次の場所にある Application Server domain.xml ファイル内の jss3.jar を jss4.jar に置き換えてください。

AppServer8Config-base/domains/*domainName*/config/domain.xml

Release 2 の J2EE コンポーネントの移行

Release 2 の J2EE コンポーネントおよび J2EE アプリケーションを Release 5 の Application Server 環境で動作するように移行し、適切な Application Server インスタンスにそれらを再配備してください。J2EE コンポーネントと J2EE アプリケーションの移行の詳細については、『Application Server Enterprise Edition 8.2 アップグレードと移行』(http://docs.sun.com/doc/820-1612?l=ja) の第 4 章を参照してください。

アップグレードのロールバック

Application Server の Release 5 へのアップグレードは、Release 5 へのアップグレードによって削除されなかった Application Server の Release 2 バージョンを復元するだけでロールバックできます。

Solaris 10 の複数ゾーン環境における Solaris 付属の Application Server のアップグレード

Application Server (および Application Server の依存する Message Queue) は、Solaris 10 オペレーティングシステムに付属しています。大域ゾーンから削除しない限り、付属する Application Server は、非大域ゾーンが作成されるとそこに伝播されます。すべてのゾーンでの付属する Application Server の存在は、その後の Application Server の Release 5 へのアップグレードに次のような影響を与えます。

- 大域ゾーンで Application Server をアップグレードすると、大域ゾーン内の付属するバージョンが削除され、すべての非大域ゾーンからその付属するバージョンが自動的に削除されます。
- 大域ゾーンで Application Server を Release 5 にアップグレードしても、非大域ゾーンでは Application Server はアップグレードされません。これは、Release 5 の Application Server パッケージが伝播されないためです。

次の例で、Solaris 10 の複数ゾーン環境における Application Server のアップグレードに関するいくつかの細かい点を説明します。Java ES および Solaris 10 のゾーンに関するさらに包括的な説明は、62 ページの「Java ES 5 のアップグレードと Solaris 10 ゾーン」を参照してください。

この例での目的は、Solaris 10 疎ルートゾーンにおいて、Solaris 付属の Application Server (バージョン 8.0.0_01) を Release 5 にアップグレードすることです。

Solaris 付属の Application Server は、大域ゾーンからマウントされた読み取り専用ディレクトリにインストールされているため、疎ルートゾーンでは Application Server を単にアップグレードすることができません。したがって、疎ルートゾーン内で Application Server を Release 5 にアップグレードするには、まず大域ゾーン内で付属するバージョンを削除してください。

また、Message Queue が大域ゾーンにインストールされていることは、大域ゾーンには共有コンポーネントのみがインストールされる (製品コンポーネントはインストールされない) という通常の方法から逸脱していることを表しています。これは、読み取り専用ディレクトリが原因で、Message Queue を疎ルートゾーン内でインストールまたはアップグレードできないからです。

Solaris 10 疎ルートゾーンにおいて、Solaris 付属の Application Server (バージョン 8.0.0_01) を Release 5 にアップグレードするための手順は、次のとおりです。

1. システムの最初の状態を確認します。

   この例では、グローバル管理者によって設定、インストール、および起動された疎ルートゾーンを含む Solaris 10 バージョンを想定します。

   この疎ルートゾーンには、大域ゾーンにすでにインストール済みのすべての Java ES コンポーネント (つまり、Solaris 10 付属バージョンの Message Queue および Application Server) が含まれています。

   さらにこの例では、ユーザーがこれまでに疎ルートゾーン内の付属する Application Server を使用したことがあるために、管理ドメイン情報が作成されていて、これを保持する必要があると仮定します。

2. 大域ゾーンで、付属するバージョンの Application Server をアップグレードします。

   この操作によって、付属する Application Server パッケージは削除され、Release 5 パッケージと置き換えられます。付属するパッケージの削除は疎ルートゾーンに伝播され、疎ルートゾーンでは Application Server パッケージのアンインストールが効果的に行われます。しかし、Release 5 パッケージは非大域ゾーンに伝播されません。

   a. 大域ゾーンで Java ES インストーラを実行します。

   b. コンポーネント選択ページで、Application Server を選択します。

      Message Queue、HADB、および Java DB が自動的に選択され、Application Server と Message Queue は「アップグレード可」であると指定されます。

   c. アップグレードを完了します。

      Release 5 の Message Queue は疎ルートゾーンに伝播されますが、Application Server、HADB、Java DB のいずれも伝播されません。さらに、すべての共有コンポーネントは Release 5 に同期され、疎ルートゾーンに伝播されます。

3. 疎ルートゾーンに Application Server をインストールします。

   a. 疎ルートゾーンで Java ES インストーラを実行します。

   b. コンポーネント選択ページで、Application Server を選択します。

      Message Queue が自動的に選択されている場合はこれを選択解除して、HADB と Java DB が自動的に選択されていない場合はこれらを選択します。

c. Application Server のインストールを完了します。

インストールプロセスによってドメイン情報を上書きされないように、「あとで設定」を選択します。

ドメインをこれまでに 1 つも作成していない場合は、「今すぐ設定」を選択できます。

第 12 章

Service Registry

この章では、Service Registry を Java ES 5 (Release 5): Service Registry 3.1 にアップグレードする方法について説明します。

ここでは、Release 5 でサポートされている各種のアップグレードパスでのアップグレードの注意点の概要について説明します。また、Solaris オペレーティングシステムと Linux オペレーティングシステムの両方でのアップグレードを取り上げます。

- 230 ページの「Service Registry のアップグレードの概要」
- 233 ページの「Java ES Release 4 からの Service Registry のアップグレード」

注 この章では、*ServiceRegistryR4-base* および *RegistryDomainR4-base* (Java ES Release 4 Service Registry の場合)、または *ServiceRegistryR5-base* および *RegistryDomainR5-base* (Java ES Release 5 Service Registry の場合) というディレクトリパスを基準に、ファイルの場所が指定されています。これらのパスの少なくとも一部は、Service Registry がインストールされたときにインストールディレクトリとして指定されている場合があります。指定されていない場合、Java ES インストーラによってデフォルト値が割り当てられています。

次の表に、これらのディレクトリパスのデフォルト値を示します。

表 12-1 Service Registry ディレクトリパス

パス名変数	Solaris OS	Linux OS
ServiceRegistryR4-base	`/opt/SUNWsoar`	`/opt/sun/SUNWsoar`
RegistryDomainR4-base	`/var/opt/SUNWsoar`	`/var/opt/sun/SUNWsoar`
ServiceRegistryR5-base	`/opt/SUNWsrvc-registry`	`/opt/sun/srvc-registry`
RegistryDomainR5-base	`/var/opt/SUNWsrvc-registry`	`/var/opt/sun/srvc-registry`

Service Registry のアップグレードの概要

ここでは、Java ES 5 (Release 5) へのアップグレードに影響する Service Registry の一般的な側面について説明します。

- Java ES Release 5 について
- Java ES Release 5 へのアップグレードロードマップ
- Service Registry のデータ
- Service Registry のアップグレード方針

Java ES Release 5 について

Java ES Release 5 の Service Registry は、Release 4 Service Registry に対するマイナーリリースです。このリリースでは、いくつかの機能強化、インタフェースの更新、選ばれたバグ修正が行われています。

Java ES Release 5 へのアップグレードロードマップ

表 12-2 は、Service Registry を Java ES Release 5 へアップグレードする際の、サポートされるアップグレードパスを示しています。この表は Solaris および Linux の両方のオペレーティングシステムに適用されます。

表 12-2 Java ES 5 (Release 5): Service Registry 3.1 へのアップグレードパス

Java ES のリリース	Service Registry のバージョン	一般的な手順	必要な再設定
Release 4	Sun Java System Service Registry 3.0 2005Q4	直接アップグレード: Release 4 を新規インストールによって置き換えて、レジストリデータを Release 5 に転送します。	なし

Service Registry のデータ

次の表は、Service Registry ソフトウェアのアップグレードにより影響を受ける可能性のあるデータの種類を示しています。

表 12-3 Service Registry のデータの使用法

データの種類	場所	使用法
インストールパラメータ	*ServiceRegistryR4-base*/install/install.properties	Service Registry の設定
信頼できる証明書	*ServiceRegistryR4-base*/install/cacerts	Application Server インストールの一部ではない Service Registry によって信頼される証明書
設定データ	*RegistryDomainR4-base*/domains/registry/applications/j2ee-modules/soar/WEB-INF/classes/*.properties	Service Registry インスタンスの設定
レジストリデータ / リポジトリデータ	*RegistryDomainR4-base*/3.0/data	データベースとユーザー証明書
Web インタフェース設定	*RegistryDomainR4-base*/3.0/jaxr-ebxml	Web インタフェースの設定

Service Registry のアップグレード方針

Service Registry をアップグレードするための方針は一般に、第 1 章「アップグレードの計画」で説明されている多くの考慮事項 (アップグレードパス、Java ES コンポーネント間の依存関係、選択的なアップグレードかすべてのアップグレードか、複数インスタンスの配備など) によって異なります。

ここでは、Service Registry のアップグレード計画に影響する可能性のある問題を取り上げ、Service Registry の全般的な説明を詳細に記述します。

互換性の問題

Release 5 の Service Registry は、Release 4 の Service Registry と下位互換性があります。

依存関係

ほかの Java ES コンポーネントに対する Service Registry の依存関係が、Service Registry ソフトウェアのアップグレードおよび再設定の手順に影響を及ぼす可能性があります。たとえば、Service Registry のインタフェースまたは関数が変更された場合は、Service Registry が依存するコンポーネントのアップグレードされたバージョンが必要になる場合があります。このようなコンポーネントのアップグレードの必要性は、個々のアップグレードパスによって異なります。

Service Registry は、次の Java ES コンポーネントに依存します。

- **共有コンポーネント** : Service Registry は、特定の Java ES 共有コンポーネントに依存します (48 ページの表 1-9 を参照)。
- Application Server: Service Registry は、Service Registry アプリケーションにコンテナを提供したり、Java ES Release 5 においてネットワーク化されたレジストリまたはリポジトリデータベースに対する接続を管理したりするために、Application Server に必ず依存しています。
- Java DB: Service Registry は、サービスやサービスを記述するメタデータを格納するためのデフォルトデータベースとして、Java DB に必ず依存しています。

二重のアップグレード

44 ページの「二重のアップグレード : Java ES ソフトウェアとオペレーティングシステムソフトウェア」で説明されているように、二重のアップグレードでは Service Registry とオペレーティングシステムの両方をアップグレードします。次の 2 つのどちらかの方法で二重のアップグレードを実行できます。

- 新規オペレーティングシステムのインストール
- オペレーティングシステムのインプレースアップグレード

新規オペレーティングシステムのインストール

1. 既存の Service Registry データをバックアップします。

   基本的なデータの場所については、231 ページの「Service Registry のデータ」を参照してください。

2. 新しいオペレーティングシステムをインストールします。

   新しいシステム (または Solaris 10 ゾーン) でオペレーティングシステムのインストールを実行するか、または、既存のファイルシステムを消去することができます。

3. 手順 1 でバックアップした Service Registry データを復元します。
4. Release 5 の Service Registry をインストールします。

オペレーティングシステムのインプレースアップグレード

1. 既存の Service Registry データをバックアップします。

   基本的なデータの場所については、231 ページの「Service Registry のデータ」を参照してください。

2. オペレーティングシステムをアップグレードします。

   このアップグレードでは、既存のファイルシステムはそのまま残されます。

3. Release 5 の Service Registry にアップグレードします。

   233 ページの「Java ES Release 4 からの Service Registry のアップグレード」を参照してください。

Java ES Release 4 からの Service Registry のアップグレード

ここでは、Service Registry の Java ES 2005Q4 (Release 4) から Java ES 5 (Release 5) へのアップグレードについて説明します。この節で説明する内容は、次のとおりです。

- はじめに
- Release 4 の Service Registry のアップグレード
- 複数インスタンスのアップグレード

はじめに

Java ES Release 4 Service Registry を Release 5 にアップグレードする場合は、アップグレード処理の次の側面を考慮してください。

- **一般的なアップグレード手順** : このアップグレードは、Release 5 の Service Registry の新規インストールを行い、Release 4 のデータと設定を Release 5 に移行し、さらに Release 4 を削除してディスク容量を確保することによって実行されます。
- **アップグレードの依存関係** : Service Registry は、いくつかの Java ES 共有コンポーネントに依存しています (48 ページの表 1-9 を参照)。Service Registry のアップグレードを実行すると、すべての共有コンポーネントは Java ES インストーラによって自動的に Release 5 にアップグレードされます。

  Service Registry は、Application Server および Java DB に対する強いアップグレード依存関係を持ちます。
- **下位互換性** : Release 5 の Service Registry は Release 4 と完全互換です。

- **アップグレードのロールバック**：Release 5 アップグレードのロールバックは、保存したデータベースと設定データを復元したあとで、Release 4 に戻すことにより行われます。
- **プラットフォームの問題**：.Service Registry をアップグレードするための一般的な手順は、Solaris および Linux の両オペレーティングシステムで共通です。

Release 4 の Service Registry のアップグレード

ここでは、Solaris プラットフォームと Linux プラットフォームの両方で、Service Registry の Java ES Release 4 から Java ES Release 5 へのアップグレードを実行する方法について説明します。プラットフォーム固有の手順に依存する項目には、その手順が適用されるオペレーティングシステムが示されています。この節で説明する内容は、次のとおりです。

- アップグレード前の作業
- Release 4 の Service Registry のアップグレード
- アップグレードの確認
- アップグレード後の作業
- アップグレードのロールバック

アップグレード前の作業

Service Registry をアップグレードする前に、次の作業を実行してください。

- 現在のバージョン情報の確認
- Service Registry が依存するコンポーネントのアップグレード
- Service Registry のデータのバックアップ
- 必要な設定情報とパスワードの入手

現在のバージョン情報の確認

Service Registry の現在のバージョンは、次の Web Console のユーザーインタフェースの特性を観察することによって確認できます。

```
http://localhost:6060/soar
```

また、Service Registry のパッケージ名を確認することもできます。次に例を示します。

Solaris の場合：

```
pkginfo -l|grep srvc
```

Linux の場合:

```
rpm -qa|grep srvc
```

特徴的な特性やパッケージ名を、次の表に示します。

表 12-4 Service Registry のバージョン確認の出力

Java ES のリリース	Service Registry のバージョン番号	特徴的な特性
Release 4	3.0	Web Console: 左側のパネルのツールセクション パッケージ名に含まれる文字列 : soar
Release 5	3.1	Web Console: 左側のパネルの 3 つのタブ パッケージ名に含まれる文字列 : srvc-registry

Service Registry が依存するコンポーネントのアップグレード

一般には、コンピュータシステム上 (および、コンピューティング環境内) のすべての Java ES コンポーネントを Java ES Release 5 にアップグレードすることをお勧めします。Service Registry は、いくつかの共有コンポーネント、Application Server、および Java DB に対して強いアップグレード依存関係を持っています。

Service Registry が依存するコンポーネントをアップグレードするときは、Service Registry をアップグレードする前に、(すでにアップグレードした可能性があるものを省いて) 次の順序で実行してください。共有コンポーネントのアップグレードは通常、Java ES インストーラによって自動的に実行されます。

1. **共有コンポーネント** : Java ES 共有コンポーネントを Release 5 に同期する手順は、67 ページの「Java ES 共有コンポーネントのアップグレード」で説明されています。ただし、Service Registry の Release 5 へのアップグレードを実行すると、Service Registry で必要な共有コンポーネントはすべて、Java ES インストーラによって自動的にアップグレードされます。
2. Java DB: Java DB を Release 5 にアップグレードする手順は、157 ページの第 8 章「Java DB」で説明されています。
3. Application Server: Application Server を Release 5 にアップグレードする手順は、203 ページの第 11 章「Application Server」で説明されています。

HTTP ポート番号の変更

ServiceRegistryR4-base/install/install.properties ファイルを編集して、HTTP ポートを 6060 から 6480 に変更します (6060 は予約ポート)。このプロパティーの設定方法については、『Service Registry 3.1 管理ガイド』(http://docs.sun.com/doc/820-0535?l=ja) を参照してください。

Service Registry のデータのバックアップ

Service Registry の Release 4 から Release 5 へのアップグレードでは、設定データやレジストリデータベース、リポジトリデータベースは変更されません。現在のデータをバックアップする必要はありません。

必要な設定情報とパスワードの入手

Release 4 の Service Registry のユーザー ID、パスワード、ドメイン名、ポート番号を知っている必要があります。

Release 4 の Service Registry のアップグレード

ここでは、Solaris プラットフォームと Linux プラットフォームでのアップグレード手順について説明します。

アップグレード手順 (Solaris)

ここで説明する手順は、アップグレードを実施するコンピュータにローカルインストールされている Service Registry インスタンスに適用されます。

1. root としてログインするか、スーパーユーザーになります。

   ```
   su -
   ```

2. Jakarta ANT Java/XML ベースの構築ツール (ANT 共有コンポーネント) が正しいバージョンの J2SE を参照していることを確認します。

   (その後の手順では、ant コマンドが使用される)

   ```
   PATH=/usr/jdk/entsys-j2se/bin:$PATH

   export PATH
   ```

3. Release 4 Service Registry (Application Server) ドメインを停止します。

   ```
   cd ServiceRegistryR4-base/install
   /usr/sfw/bin/ant -f build-install.xml appserver.domain.stop
   ```

   このドメインは Service Registry インスタンスに関連付けられています。

4. Release 5 の Service Registry の新規インストールを実行します。

   次の手順を実行します。

a. Release 4 の Service Registry がホストされたコンピュータ上で Java ES インストーラを起動します。

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

ここで、*os_arch* は、`Solaris_sparc` などのプラットフォームに対応します。コマンド行インタフェースには、`installer -nodisplay` オプションを使用します。

ようこそページと、ライセンス使用許諾契約ページが表示されてから、コンポーネント選択ページが表示されます。Java ES インストーラが直接アップグレードできるインストール済みのコンポーネントが検出された場合、それらのコンポーネントは「アップグレード可」の状態で表示されます。

b. コンポーネント選択ページから Service Registry を選択します。

c. Release 4 とは異なるインストールディレクトリパスを指定します。

デフォルトでは、Release 5 のインストールパス (*ServiceRegistryR5-base*) は Release 4 のインストールパス (*ServiceRegistryR4-base*) と異なります。

d. 「あとで設定」オプションを選択します。

「今すぐ設定」はサポートされていません。

e. 必要に応じて、ローカライズ版のパッケージをインストールするオプションを選択します。

f. インストールが完了したら、Java ES インストーラを終了します。

5. Release 5 の Service Registry インスタンスのアップグレードと設定を行います。

```
cd ServiceRegistryR5-base/install
/usr/sfw/bin/ant -f build-install.xml
    -Dinstall.properties=ServiceRegistryR4-base/install/install.properties
    upgrade
```

Release 4 の `install.properties` ファイルをポイントする代わりに、デフォルトの Release 5 の `install.properties` ファイルを変更して任意の Release 4 プロパティー値を再現することもできます。これらのプロパティーの設定方法については、『Service Registry 3.1 管理ガイド』(http://docs.sun.com/doc/820-0535?l=ja) を参照してください。

カスタムプロパティー値を使用しているが、それらが `install.properties` 内に含まれていない場合、これらのプロパティー値を Ant コマンド行で (全体を 1 行にして) 、次のように指定する必要があります。

```
/usr/sfw/bin/ant -f build-install.xml
    -Dregistry.install.RegistryServerKeystorePassword=passwd1
    -Dregistry.install.AdministratorPassword=passwd2
    -Dregistry.install.ApplicationServerKeystorePassword=passwd3
    upgrade
```

ただしこれらのカスタムプロパティー値は、権限のないユーザーに対して表示が行われるようなコマンド行の設定を避けるため、アクセス権の制限された `install.properties` ファイルに格納することをお勧めします。詳細については、『Service Registry 管理ガイド』を参照してください。

アップグレードユーティリティーは新しい Application Server ドメインを作成し、ドメインを起動し、そのドメイン内に Service Registry インスタンスを配備します。各 Service Registry インスタンスは、それぞれ独自の Application Server ドメインに関連付けられています。

6. Release 4 の Service Registry のサーバープロパティーファイルが変更された場合、それに対応する Release 5 の Service Registry の設定変更を、次のように行うことができます。

   a. Release 5 Service Registry (Application Server) ドメインを停止します。

      (このドメインは、手順 5 の `upgrade` コマンドによって自動的に起動されている)

      ```
      cd ServiceRegistryR5-base/install
      /usr/sfw/bin/ant -f build-install.xml appserver.domain.stop
      ```

   b. Release 4 の Service Registry インスタンス設定を Release 5 に転送します。

      Release 4 の Service Registry インスタンス設定に対して行なった任意の変更を選択します。

      ```
      RegistryDomainR4-base/domains/registry/applications/j2ee-modules/
      soar/WEB-INF/classes/*.properties
      ```

      これを、対応する Release 5 の設定に追加します。

      ```
      RegistryDomainR5-base/domains/registry/applications/j2ee-modules/
      soar/WEB-INF/classes/*.properties
      ```

7. Release 5 の Service Registry (Application Server) ドメインを起動します。

   ```
   cd ServiceRegistryR5-base/install
   /usr/sfw/bin/ant -f build-install.xml appserver.domain.start
   ```

アップグレード手順 (Linux)

Linux での Service Registry のアップグレードは Solaris での手順と同じです (236 ページの「アップグレード手順 (Solaris)」を参照)。ただし、さまざまなアップグレード手順で使用される `ant` コマンドの Linux プラットフォーム上での場所が、Solaris プラットフォーム上での場所と異なります。

```
/opt/sun/share/bin/ant
```

アップグレードの確認

Service Registry のアップグレードが正常に行われたことは、次の Web Console のユーザーインタフェースの特性を観察することによって確認できます。

```
http://localhost:6480/soar
```

また、Service Registry のパッケージ名を確認することもできます。次に例を示します。

Solaris の場合:

```
pkginfo -l|grep soar
```

Linux の場合:

```
rpm -qa|grep soar
```

特徴的な特性やパッケージ名は、235 ページの表 12-4 に示されています。

アップグレード後の作業

次の手順は Release 4 の Service Registry の削除方法を示すものですが、この手順は、アップグレードを Release 4 にロールバックしないことが確実になるまでは実行しないでください。

1. Release 4 の Service Registry (Application Server) ドメインを削除します。

   ```
   cd ServiceRegistryR4-base/install
   ```

   Solaris の場合:

   ```
   /usr/sfw/bin/ant -f build-install.xml appserver.domain.delete
   ```

   Linux の場合:

   ```
   /opt/sun/bin/ant -f build-install.xml appserver.domain.delete
   ```

2. Release 4 の Service Registry ドメインファイルを含むディレクトリを削除します。

   rm -rf *RegistryDomainR4-base*

3. Release 4 の Service Registry インストールファイルを含むディレクトリを削除します。

   rm -rf *ServiceRegistryR4-base*

アップグレードのロールバック

Release 5 アップグレードのロールバックは、Release 5 へのアップグレード時に未変更のまま残されていた以前のバージョンを復元することによって行われます。

1. Release 5 の Service Registry (Application Server) のドメインを停止して削除します。

```
cd ServiceRegistryR4-base/install
```

Solaris の場合:

```
/usr/sfw/bin/ant -f build-install.xml appserver.domain.delete
```

Linux の場合:

```
/opt/sun/bin/ant -f build-install.xml appserver.domain.delete
```

2. Java ES Release 5 のアンインストーラを実行して、Release 5 の Service Registry をアンインストールします。
3. Release 4 の Service Registry のドメインを起動します。

   ```
   cd ServiceRegistryR4-base/install
   ```

   Solaris の場合:

   ```
   /usr/sfw/bin/ant -f build-install.xml appserver.domain.start
   ```

   Linux の場合:

   ```
   /opt/sun/bin/ant -f build-install.xml appserver.domain.start
   ```

4. Release 4 の Service Registry の Web Console にアクセスします。

   ```
   http://localhost:6480/soar
   ```

5. コンソールに表示される Release 4 の特性 (235 ページの「Service Registry のバージョン確認の出力」を参照) を確認します。

複数インスタンスのアップグレード

配備のアーキテクチャーによっては、スケーラビリティーを実現し、可用性を向上させるために、Service Registry が複数のコンピュータシステムに配備される場合があります。たとえば、ロードバランサを使用して Service Registry インスタンスを複数のコンピュータで実行することにより、負荷を分散させることができます。

これらのアーキテクチャでは、レジストリは主に読み取り専用で、共通のデータベースにアクセスすることで、負荷の高いクエリーに応答しています。

234 ページの「Release 4 の Service Registry のアップグレード」の説明に従って、各コンピュータで Service Registry のアップグレードを実行します。

第 13 章

Web Proxy Server

この章では、Web Proxy Server を Java ES 5 (Release 5): Sun Java System Web Proxy Server 4.0.4 にアップグレードする方法について説明します。

ここでは、Release 5 でサポートされている各種のアップグレードパスでのアップグレードの注意点の概要について説明します。また、Solaris オペレーティングシステムと Linux オペレーティングシステムの両方でのアップグレードを取り上げます。

- 242 ページの「Web Proxy Server のアップグレードの概要」
- 244 ページの「Java ES Release 4 からの Web Proxy Server のアップグレード」
- 251 ページの「Version 3.6 からの Web Proxy Server のアップグレード」

注

この章では、*WebProxyServer-base* というディレクトリパスを基準に指定されています。このパスの少なくとも一部は、Web Proxy Server が最初にインストールされたときにインストールディレクトリとして指定されている可能性があります。指定されていない場合、Java ES インストーラによってデフォルト値が割り当てられます。

WebProxyServer-base のデフォルト値は、次の表に示すように、オペレーティングシステムプラットフォームによって異なります。

表 13-1 Web Proxy Server ディレクトリパス

パス名変数	Solaris OS	Linux OS
WebProxyServer-base	/opt/SUNWproxy	/opt/sun/webproxyserver

Web Proxy Server のアップグレードの概要

ここでは、Java ES 5 (Release 5) へのアップグレードに影響する Web Proxy Server の一般的な側面について説明します。

- Java ES Release 5 の Web Proxy Server について
- Web Proxy Server のアップグレードロードマップ
- Web Proxy Server のデータ
- Web Proxy Server のアップグレード方針

Java ES Release 5 の Web Proxy Server について

Java ES Release 5 の Web Proxy Server は、Release 4 に対するマイナーリリースで、バグ修正が行われています。

ただし、Release 5 の Web Proxy Server の特長には、パフォーマンスの向上、よりスケーラブルなアーキテクチャー、標準への準拠度の向上、Java Enterprise System に組み込まれる前の Sun One Web Proxy Server 3.6 から一新された新しい管理インタフェースなどがあります。

Web Proxy Server のアップグレードロードマップ

表 13-2 に、Java ES Release 5 への Web Proxy Server のアップグレードパスを示します。Web Proxy Server は以前の Java ES リリースには含まれていませんでした。Web Proxy Server はこれまで Linux オペレーティングシステムではサポートされていなかったため、この表は Solaris オペレーティングシステムにのみ対応します。

表 13-2 Java ES 5 (Release 5): Web Proxy Server 4.0.4 へのアップグレードパス

Java ES のリリース	Web Proxy Server のバージョン	一般的な手順	必要な再設定
Release 4	Sun Java System Web Proxy Server4.0.1 2005Q4	直接アップグレード : パッチを使用することにより実行されます。	なし
それ以前の Java ES リリース (Solaris OS のみ)	Sun ONE Web Proxy Server 3.6 (以後 Version 3.6 と表記)	直接アップグレード : Java ES インストーラを使用して新しい場所にインストールし、管理ツールを使用して設定データを移行することによって行います。	設定情報を新しい場所に移行する必要があります。

Web Proxy Server のデータ

次の表は、Web Proxy Server ソフトウェアのアップグレードにより影響を受ける可能性のあるデータの種類を示しています。

表 13-3　Web Proxy Server のデータの使用法

データの種類	場所	使用法
設定データ	*WebProxyServer-base*/proxy-server*id*/config ディレクトリ 含まれるファイルの例 : server.xml、magnus.conf、obj.conf など	サーバー、キャッシュ、フィルタ、ルーティング、および Web Proxy Server のその他の機能に関連する設定情報を格納します

Web Proxy Server のアップグレード方針

Web Proxy Server をアップグレードするための方針は一般に、第 1 章「アップグレードの計画」で説明されている多くの考慮事項 (アップグレードパス、Java ES コンポーネント間の依存関係、選択的なアップグレードかすべてのアップグレードか、複数インスタンスの配備など) によって異なります。

ここでは、Web Proxy Server のアップグレード計画に影響する可能性のある問題を取り上げ、Web Proxy Server の全般的な説明を詳細に記述します。

互換性の問題

Release 5 Web Proxy Server では新しい公開インタフェースは導入されておらず、Release 4 の Web Proxy Server と下位互換性があります。Release 5 の Web Proxy Server も Version 3.6 と互換性がありますが、Version 3.6 でサポートされていた NSAPI インタフェースを使用して開発したプラグインは、Release 5 でサポートされている NSAPI インタフェースを使用して再コンパイルしてください。

Web Proxy Server の依存関係

Web Proxy Server は、次の Java ES コンポーネントに依存します。

- **共有コンポーネント** : Web Proxy Server は、特定の Java ES 共有コンポーネントに依存します (48 ページの表 1-9 を参照)。
- Directory Server: Web Proxy Server は、LDAP ベースの認証サービスを提供するために、Directory Server に対して依存関係を持っていますが、これはオプションです。

- **Web Server:** Web Proxy Server は、HTTP 要求のセキュリティーとパフォーマンスを向上させるために、Web Server に対して共依存関係を持っています。

二重のアップグレード

44 ページの「二重のアップグレード : Java ES ソフトウェアとオペレーティングシステムソフトウェア」で説明されているように、二重のアップグレードでは Web Proxy Server とオペレーティングシステムの両方をアップグレードします。次のようなインプレースオペレーティングシステムアップグレードを使用して二重のアップグレードを実行できます。

1. 既存の Web Proxy Server データをバックアップします。

   基本的なデータの場所については、243 ページの「Web Proxy Server のデータ」を参照してください。

2. オペレーティングシステムをアップグレードします。

   このアップグレードでは、既存のファイルシステムはそのまま残されます。

3. Release 5 の Web Proxy Server にアップグレードします。

   アップグレードパスに応じて、この章の該当する節を参照してください。

Java ES Release 4 からの Web Proxy Server のアップグレード

ここでは、Web Proxy Server の Java ES 2005Q4 (Release 4) から Java ES 5 (Release 5) へのアップグレードについて説明します。この節で説明する内容は、次のとおりです。

- はじめに
- Release 4 の Web Proxy Server のアップグレード

はじめに

Web Proxy Server を Release 5 にアップグレードするときは、アップグレードプロセスに関する次の事項を考慮してください。

- **一般的なアップグレード手順 :** このアップグレードは、パッチを使用することにより実行されます。追加の再設定は必要ありません。

- **アップグレードの依存関係** : Web Proxy Server はいくつかの Java ES 共有コンポーネントに依存しますが (48 ページの表 1-9 を参照)、Web Proxy Server が強い依存関係を持っているのは、NSS および NSPR 共有コンポーネントに対してのみです。
- **下位互換性** : Release 5 Web Proxy Server は、Release 4 と下位互換性があります。
- **アップグレードのロールバック** : Web Proxy Server の Release 5 へのアップグレードのロールバックは、アップグレードパッチを削除することにより行われます。
- **プラットフォームの問題** : Web Proxy Server をアップグレードするための一般的な手順は、Solaris および Linux の両オペレーティングシステムで共通ですが、パッチの適用に関して技術的に異なる点があります。したがって、アップグレードプロセスにはプラットフォーム固有の手順が含まれます。

Release 4 の Web Proxy Server のアップグレード

この節では、Web Proxy Server を Java ES Release 5 にアップグレードする方法の概要を示します。ここで説明する内容は次のとおりです。

- アップグレード前の作業
- Release 4 の Web Proxy Server のアップグレード (Solaris)
- Release 4 の Web Proxy Server のアップグレード (Linux)
- アップグレードの確認
- アップグレード後の作業
- アップグレードのロールバック (Solaris)

アップグレード前の作業

Web Proxy Server をアップグレードする前に、次の作業を実行してください。

- 現在のバージョン情報の確認
- Web Proxy Server が依存するコンポーネントのアップグレード
- Web Proxy Server のデータのバックアップ
- 必要な設定情報とパスワードの入手

現在のバージョン情報の確認

Web Proxy Server の現在のバージョンは、次のコマンドを入力することにより確認できます。

WebProxyServer-base/proxy-admserv/start -version

表 13-4　Web Proxy Server のバージョン確認の出力

Java ES のリリース	Web Proxy Server のバージョン番号
Version 3.6 (非 Java ES リリース)	`3.6`
Release 4	`4.0.1`
Release 5	`4.0.4`

Web Proxy Server が依存するコンポーネントのアップグレード

一般的には、コンピュータシステム上 (および、コンピューティング環境内) のすべての Java ES コンポーネントを Java ES Release 5 にアップグレードすることが推奨されます。ただし、Release 5 への Web Proxy Server のアップグレードでは、NSS および NSPR 共有コンポーネントのアップグレードのみが必須です。これらの共有コンポーネントをまだアップグレードしていない場合は、共有コンポーネントの同期機能を使用して、すべての共有コンポーネントを、対応する Release 5 バージョンに同期するようにしてください。手順は 67 ページの第 2 章「Java ES 共有コンポーネントのアップグレード」で説明されています。

Web Proxy Server のデータのバックアップ

Release 5 への Web Proxy Server のアップグレードでは、Release 4 の設定データは変更されません。現在のデータをバックアップする必要はありません。

必要な設定情報とパスワードの入手

現在インストールされているバージョンに関する特別な情報は必要ありません。ただし、アップグレードを実行するには、スーパーユーザーとしてログインする必要があります。

Release 4 の Web Proxy Server のアップグレード (Solaris)

ここでは、Web Proxy Server のアップグレード手順に影響する注意点を示したあと、その手順自体について説明します。

アップグレードの注意点 (Solaris)

Web Proxy Server ソフトウェアの Java ES Release 5 へのアップグレードでは、次の注意点を考慮に入れてください。

- 同じ Web Proxy Server インストールイメージに対応するすべての Web Proxy Server インスタンスが同時にアップグレードされます。インストールイメージにパッチを適用している間は、このようなインスタンスをすべて停止するようにしてください。

- Solaris OS での Release 5 Web Proxy Server のアップグレードパッチを次の表に示します。

表 13-5　Solaris で Web Proxy Server をアップグレードするためのパッチ[1]

説明	パッチ ID: SPARC Solaris 9、10	パッチ ID: X86 Solaris 9、10
Web Proxy Server コア	120981-10	120982-10
Web Proxy Server ローカライズ版	122963-01	122964-01

1. この表に示されているパッチのリビジョン番号は、Java ES Release 5 にアップグレードするための最低要件です。表に記載されているものよりも新しいリビジョンのパッチが使用可能になった場合は、新しいほうのパッチを使用してください。

アップグレード手順 (Solaris)

次に示す手順は、アップグレードを実行するコンピュータ上の Web Proxy Server に適用されます。

1. root としてログインするか、スーパーユーザーになります。

   ```
   su -
   ```

2. Web Proxy Server および Administration Server の、稼働中のすべてのインスタンスを停止します。

   WebProxyServer-base/proxy-*instanceName*/stop
   WebProxyServer-base/proxy-admserv/stop

3. すべての共有コンポーネントの Release 5 への同期をまだ実行していない場合は、実行します。

   手順は 67 ページの第 2 章「Java ES 共有コンポーネントのアップグレード」で説明されています。

4. 表 13-5 に基づいて、必要なパッチを入手します。

   パッチは http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access から /tmp にダウンロードします。

5. 表 13-5 の適切な Web Proxy Server コアパッチと、必要に応じてローカライズ版パッチを、この順序で適用します。

   ```
   patchadd /tmp/patch_ID
   ```

6. パッチのアップグレードが成功したことを確認します。

   ```
   showrev -p | grep proxy
   ```

   この出力では、手順 5 で適用したパッチ ID のバージョンが返されるはずです。

7. 手順 2 で停止した Web Proxy Server インスタンスを再起動します。

   WebProxyServer-base/proxy-*instanceName*/start

Release 4 の Web Proxy Server のアップグレード (Linux)

ここでは、Web Proxy Server のアップグレード手順に影響する注意点を示したあと、その手順自体について説明します。

アップグレードの注意点 (Linux)

Linux プラットフォームで Web Proxy Server ソフトウェアを Java ES Release 5 にアップグレードするときは、Linux の Release 5 アップグレードパッチが Solaris のパッチとは異なる点を除き、Solaris プラットフォームのときと同じ注意点 (253 ページの「アップグレードの注意点」を参照) を考慮に入れてください。

Linux OS での Release 5 Web Proxy Server のアップグレードパッチを次の表に示します。

表 13-6　Linux で Web Proxy Server をアップグレードするためのパッチ[1]

説明	パッチ ID と RPM 名
Web Proxy Server コア	120983-10
	• sun-proxyserver-4.0-6.5.i386.rpm
Web Proxy Server ローカライズ版	122965-01
	• sun-proxyserver-*Locale*-4.0.4.i386.rpm

1. この表に示されているパッチのリビジョン番号は、Java ES Release 5 にアップグレードするための最低要件です。表に記載されているものよりも新しいリビジョンのパッチが使用可能になった場合は、新しいほうのパッチを使用してください。

アップグレード手順 (Linux)

次に示す手順は、アップグレードを実行するコンピュータ上の Web Proxy Server に適用されます。

注意　Linux での Java ES Release 4 から Java ES Release 5 へのアップグレードはロールバックできません。

1. root としてログインするか、スーパーユーザーになります。

   ```
   su -
   ```

2. Web Proxy Server および Administration Server の、稼働中のすべてのインスタンスを停止します。

   WebProxyServer-base/proxy-*instanceName*/stop
   WebProxyServer-base/proxy-admserv/stop

3. すべての共有コンポーネントの Release 5 への同期をまだ実行していない場合は、実行します。

   246 ページの「Web Proxy Server が依存するコンポーネントのアップグレード」を参照してください。

4. 表 13-6 のパッチ番号と RPM 名に基づいて、必要なパッチを入手します。この情報を参照して、RPM のバージョン番号を確認してください。

   パッチは http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access から /tmp にダウンロードします。

5. 表 13-6 に示されている Web Proxy Server のコアと、必要に応じてローカライズ版 PRM を、この順序で適用します。

   ```
   rpm -Fvh sun-proxyserver-version.i386.rpm
   ```

6. パッチのアップグレードが成功したことを確認します。

   ```
   rpm -qa | grep sun-proxyserver
   ```

   RPM の新しいバージョン番号が返されることを確認します。

7. 手順 2 で停止した Web Proxy Server インスタンスを再起動します。

   WebProxyServer-base/proxy-*instanceName*/start

アップグレードの確認

Release 5 への Web Proxy Server のアップグレードは、次のように、-version オプションを使用して Web Proxy Server インスタンスを起動することによって確認できます。

WebProxyServer-base/proxy-admserv/start -version

出力値については、246 ページの表 13-4 を参照してください。

アップグレード後の作業

247 ページの「アップグレード手順 (Solaris)」および 248 ページの「アップグレード手順 (Linux)」で説明している手順以外に、アップグレード後の作業はありません。

アップグレードのロールバック (Solaris)

ここでは、Web Proxy Server のアップグレードのロールバック手順に影響する注意点を示したあと、その手順自体について説明します。

ロールバックの注意点 (Solaris)

Web Proxy Server の Release 5 へのアップグレードをロールバックするには、Release 5 へのアップグレードと逆の手順を実行します。

ロールバック手順 (Solaris)

1. root としてログインするか、スーパーユーザーになります。

   ```
   su -
   ```

2. Web Proxy Server および Administration Server の、稼働中のすべてのインスタンスを停止します。

   WebProxyServer-base/proxy-*instanceName*/stop
   WebProxyServer-base/proxy-admserv/stop

3. 247 ページの表 13-5 のパッチを削除します。

   patchrm *patch_ID*

4. 手順 2 で停止した Web Proxy Server インスタンスを再起動します。

   WebProxyServer-base/proxy-*instanceName*/start

Version 3.6 からの Web Proxy Server のアップグレード

この節では、Version 3.6 から Java ES 5 (Release 5) への Web Proxy Server のアップグレードに関する情報を示します。この節で説明する内容は、次のとおりです。

- はじめに
- Version 3.6 の Web Proxy Server のアップグレード

はじめに

Web Proxy Server を Release 5 にアップグレードするときは、アップグレードプロセスに関する次の事項を考慮してください。

- **一般的なアップグレード手順** : アップグレードは、Java ES インストーラを使用して、Version 3.6 とは異なるディレクトリに Release 5 の Web Proxy Server をインストールすることによって行います。その後、Web Proxy Server Administration Server を使用して、キャッシュされた内容を除く設定情報を Version 3.6 から Release 5 に移行します。
- **アップグレードの依存関係** : 共有コンポーネントのインストールは、Web Proxy Server を Release 5 へアップグレードするときに、Java ES インストーラによって自動的に実行されます。
- **下位互換性** : Release 5 の Web Proxy Server は Version 3.6 と下位互換性がありますが、Version 3.6 でサポートされていた NSAPI インタフェースを使用して開発したプラグインは、Release 5 でサポートされている NSAPI インタフェースを使用して再コンパイルする必要があります。
- **アップグレードのロールバック** : Release 5 への Web Proxy Server のアップグレードは、アップグレード時に未変更のまま残されていた Version 3.6 を復元することによってロールバックできます。
- **プラットフォームの問題** : Web Proxy Server をアップグレードする手順はすべての Solaris プラットフォーム上で共通ですが、Version 3.6 は Linux プラットフォームではサポートされていません。

Version 3.6 の Web Proxy Server のアップグレード

この節では、Web Proxy Server を Java ES Release 5 にアップグレードする方法の概要を示します。Web Proxy Server はこれまで、Linux プラットフォームではサポートされていませんでした。そのため、Java ES Release 5 への Web Proxy Server のアップグレードは Solaris プラットフォームでのみ実行できます。この節で説明する内容は、次のとおりです。

- アップグレード前の作業
- Version 3.6 の Web Proxy Server のアップグレード
- アップグレードの確認
- アップグレード後の作業
- Version 3.6 アップグレードのロールバック

アップグレード前の作業

Web Proxy Server をアップグレードする前に、次の作業を実行してください。

- 現在のバージョン情報の確認
- Web Proxy Server が依存するコンポーネントのアップグレード
- Web Proxy Server のデータのバックアップ
- 必要な設定情報とパスワードの入手

現在のバージョン情報の確認

Web Proxy Server の現在のバージョンは、次のコマンドを入力することにより確認できます。

WebProxyServer-base/*proxy-serverid*/start -version

表 13-7　Web Proxy Server のバージョン確認の出力

Java ES のリリース	Web Proxy Server のバージョン番号
Version 3.6 (非 Java ES リリース)	`3.6`
Release 4	`4.0.1`
Release 5	`4.0.4`

Web Proxy Server が依存するコンポーネントのアップグレード

一般的には、コンピュータシステム上 (および、コンピューティング環境内) のすべての Java ES コンポーネントを Java ES Release 5 にアップグレードすることが推奨されます。ただし、Java ES インストーラを使用して Web Proxy Server を Release 5 にアップグレードするとき、Web Proxy Server が依存するすべての共有コンポーネント (48 ページの表 1-9 を参照) が自動的にアップグレードされます。

Web Proxy Server のデータのバックアップ

Release 5 への Web Proxy Server のアップグレードでは、Version 3.6 の設定データは変更されません。ただし、管理インタフェースを使用して Version 3.6 の設定データを変更し、変更をまだ保存していない場合、アップグレードを実行する前に保存する必要があります。

必要な設定情報とパスワードの入手

Version 3.6 からアップグレードするには、Version 3.6 がインストールされているディレクトリのパスがわかっている必要があります。

Version 3.6 の Web Proxy Server のアップグレード

ここでは、Web Proxy Server のアップグレード手順に影響する注意点を示したあと、その手順自体について説明します。

アップグレードの注意点

同じ Web Proxy Server のインストールイメージに対応するすべての Web Proxy Server インスタンスをアップグレードできます。ただし、設定データの移行は各インスタンスに対して個別に行う必要があります。移行を行うときには、これらのインスタンスをすべてシャットダウンして、移行したインスタンスを起動したときにポートの競合が発生させないようにする必要があります。

アップグレード手順

ここで説明する手順は、アップグレード対象のコンピュータ上の Web Proxy Server ソフトウェアに適用されます。

1. root としてログインするか、スーパーユーザーになります。

   ```
   su -
   ```

2. Web Proxy Server Release 5 をインストールします。

   a. Java ES Release 5 ディストリビューションから Java ES インストーラを実行します。

   b. コンポーネント選択ページから Web Proxy Server を選択します。

   c. 「今すぐ設定」オプションを選択します。

d. インストールが完了したら、Java ES インストーラを終了します。

3. 新しくインストールされたバージョンに設定情報を移行します。

この操作は、各 Web Proxy Server インスタンスに対して個別に実行する必要があります。

a. Web Proxy Server 管理サーバーを起動します。

WebProxyServer-base/proxy-admserv/start

b. 管理グラフィカルインタフェースにログインします。

c. 「Server」タブをクリックし、「Migrate Server」をクリックします。

d. Version 3.6 のインストールディレクトリパスを指定します。

e. 移行するインスタンスを選択します。

f. 「Migrate」ボタンをクリックします。

移行が成功したら、移行画面に、手動で実行する必要のある追加の設定が一覧表示されます。この画面には、追加する必要のあるデータと、対応する設定ファイルが示されます。

設定情報の移行の詳細については、『Sun Java System Web Proxy Server 4.0.4 Installation and Migration Guide』(http://docs.sun.com/doc/819-5492) を参照してください。

4. 手順 f で指定された追加の設定変更を行います。

詳細については、『Sun Java System Web Proxy Server 4.0.4 Configuration File Reference』(http://docs.sun.com/doc/819-5494) を参照してください。

アップグレードの確認

Release 5 への Web Proxy Server のアップグレードは、次のように、-version オプションを使用して Web Proxy Server インスタンスを起動することによって確認できます。

WebProxyServer-base/*proxy-serverid*/start -version

出力値については、252 ページの表 13-7 を参照してください。

アップグレード後の作業

253 ページの「アップグレード手順」で説明している手順以外に、アップグレード後の作業はありません。

Version 3.6 アップグレードのロールバック

253 ページの「Version 3.6 の Web Proxy Server のアップグレード」で説明しているとおり、Release 5 への Web Proxy Server のアップグレードはロールバックできません。ただし、Release 5 のアップグレード時にそのまま残されていた Version 3.6 に戻すことは可能です。

第 14 章

Access Manager

この章では、Access Manager ソフトウェアを、以前の Java ES バージョンから Java ES 5 (Release 5): Sun Java System Access Manager 7.1 にアップグレードする方法について説明します。

ここでは、Release 5 でサポートされている各種のアップグレードパスでの Access Manager アップグレードの問題と手順の概要について説明します。また、Solaris オペレーティングシステムと Linux オペレーティングシステムの両方でのアップグレードを取り上げます。

- 258 ページの「Access Manager のアップグレードの概要」
- 265 ページの「Java ES Release 4 からの Access Manager のアップグレード」
- 286 ページの「Java ES Release 3 からの Access Manager のアップグレード」
- 290 ページの「Java ES Release 2 からの Access Manager のアップグレード」

注 この章で説明するファイルの場所は、*AccessManager-base* と *AccessManagerConfig-base* という 2 つのディレクトリパスを基準に指定されています。これらのパスの少なくとも一部は、Access Manager が最初にインストールされたときにインストールディレクトリとして指定されている可能性があります。指定されていない場合、Java ES インストーラによってデフォルト値が割り当てられています。

次の表に、これらのディレクトリパスのデフォルト値を示します。

表 14-1 Access Manager ディレクトリパス

パス名変数	Solaris OS	Linux OS
AccessManager-*base*	`/opt/SUNWam`	`/opt/sun/identity`
AccessManagerConfig-*base*	`/etc/opt/SUNWam`	`/etc/opt/sun/identity`

Access Manager のアップグレードの概要

ここでは、Java ES 5 (Release 5) へのアップグレードに影響する Access Manager の一般的な側面について説明します。

- Java ES Release 5 の Access Manager について
- Access Manager のアップグレードロードマップ
- Access Manager のデータ
- Access Manager のアップグレード方針

注 Access Manager は、Java ES Release 3 以前のバージョンでは Identity Server という名称でした。そのため、この章での Identity Server という表記は、Java ES Access Manager コンポーネントの以前のバージョンを指しています。

Java ES Release 5 の Access Manager について

Java ES Release 5 の Access Manager は、マイナーリリースです。メジャーリリースであった Java ES Release 4 の Access Manager に対して、いくつかのバグ修正と機能拡張が行われています。Release 5 の拡張機能には、Java ES Monitoring Framework に基づいた新しい監視機能が含まれています。Release 5 の拡張機能の詳細については、『Sun Java System Access Manager 7.1 リリースノート』(http://docs.sun.com/app/docs/doc/820-0363?l=ja) を参照してください。

Release 4 と同様に、Release 5 の Access Manager は、複数のアイデンティティーリポジトリまたはユーザーデータストアをサポートしています。そのため、Release 5 の Access Manager は、Directory Server などの LDAP ディレクトリだけでなく、その他のデータ格納プロトコルおよび格納形式もサポートします。

フロントエンドでは、新しい Access Manager サービスおよびアイデンティティーリポジトリの設定に Access Manager コンソールが使用されます。

ほかの Java ES コンポーネントとの下位互換性を確保するために、Release 5 は旧バージョンモードで実行できます。このモードは、Release 3 の Access Manager サービスに依存する Java ES コンポーネントをサポートします (詳細については 262 ページの「互換性の問題」参照)。

Access Manager のアップグレードロードマップ

表 14-2 は、Access Manager を Java ES Release 5 へアップグレードする際の、サポートされるアップグレードパスを示しています。この表は Solaris および Linux の両方のオペレーティングシステムに適用されます。

表 14-2　Java ES 5 (Release 5): Access Manager 7.1 へのアップグレードパス

Java ES のリリース	Access Manager のバージョン	一般的な手順	必要な再設定
Release 4	Sun Java System Access Manager 7.0 2005Q4	直接アップグレード : アップグレード前実行用スクリプトを使用して Release 4 バージョンを削除してから、Release 5 の完全インストールおよび再設定を行うことによって実行します。	設定データ Access Manager コンソールおよび認証 UI 用にカスタマイズされた JSP ディレクトリスキーマ
Release 3	Sun Java System Access Manager 6.3 2005Q1	直接アップグレード : アップグレード前実行用スクリプトを使用して Release 3 バージョンを削除してから、Release 5 の完全インストールおよび再設定を行うことによって実行します。	設定データ Access Manager コンソールおよび認証 UI 用にカスタマイズされた JSP ディレクトリスキーマ
Release 2	Sun Java System Identity Server 6.2 2004Q2 および 6.2 SP1	直接アップグレード : アップグレード前実行用スクリプトを使用して Release 2 バージョンを削除してから、Release 5 の完全インストールおよび再設定を行うことによって実行します。	設定データ Access Manager コンソールおよび認証 UI 用にカスタマイズされた JSP ディレクトリスキーマ
Release 1	Sun ONE Identity Server 6.1	直接アップグレード不可 : まず Release 3 にアップグレードする必要があります。手順については、『Sun Java Enterprise System 2005Q1 アップグレードと移行』 (http://docs.sun.com/source/819-2235?l=ja) を参照してください。 次に、Release 3 から Release 5 にアップグレードします。	設定データ Access Manager コンソールおよび認証 UI 用にカスタマイズされた JSP ディレクトリスキーマ

表 14-2　Java ES 5 (Release 5): Access Manager 7.1 へのアップグレードパス (続き)

Java ES のリリース	Access Manager のバージョン	一般的な手順	必要な再設定
それ以前の Java ES リリース	Sun ONE Identity Server 6.0 または 6.0 SP 1 または iPlanet Directory Server Access Management Edition (DSAME) 5.1	直接アップグレード不可。	

Access Manager のデータ

Access Manager では、ほかの Java ES コンポーネントと同様に、さまざまな種類のデータを利用します。Message Queue のアップグレードでは、これらのデータをアップグレード後のバージョンに移行しなければならない場合があります。次の表は、Access Manager ソフトウェアのアップグレードにより影響を受ける可能性のあるデータの種類を示しています。

表 14-3　Access Manager のデータの使用法

データの種類	場所	使用法
設定データ	*AccessManagerConfig-base*/config/AMConfig.properties *AccessManagerConfig-base*/config/serverconfig.xml 認証モジュールとカスタマイズモジュール用の JAR ファイル *AccessManager-base*/lib	Access Manager の設定と、バックエンドデータストアとの統合。

表 14-3　Access Manager のデータの使用法 (続き)

データの種類	場所	使用法
Web コンテナのアクセス制御および設定ファイル	Web Server 7.0 (Java ES Release 5) 次の場所にある server.policy および server.xml ファイル *WebServer7Config-base*/https-*configName*/config Web Server 6.*x* (Java ES Release 2、3、および 4) 次の場所にある server.policy および server.xml ファイル *WebServer6-base*/https-*hostname*/config Application Server 8.*x* (Java ES Release 3、4、および 5): 次の場所にある server.policy および domain.xml ファイル *AppServer8Config-base*/domains/*domainName*/config Application Server 7.*x* (Java ES Release 2): 次の場所にある server.policy および server.xml ファイル *AppServer7Config-base*/domains/*domainName*/config WebSphere および WebLogic: それぞれのポリシーファイルおよび設定ファイルは、これらの Web コンテナ用に Access Manager が設定されるときに変更されます。	Access Manager Web コンテナインスタンスの設定
カスタマイズデータ (Web コンテナのカスタマイズされた JSP ファイル)	管理コンソール : (Java ES Release 2 および 3): *AccessManager-base*/web-src/applications 管理コンソール : (Java ES Release 4 および 5): *AccessManager-base*/web-src/services 認証 UI: *AccessManager-base*/web-src/services	Access Manager 管理インタフェースの設定。
ディレクトリスキーマ サービスの設定 ユーザーデータ	Directory Server	Access Manager は、ディレクトリに格納されたサービス設定データ、ユーザーデータ、およびポリシーデータに基づいて、エンドユーザー向けの認証および承認サービスを提供します。
動的なアプリケーションデータ	なし	Access Manager は、セッション状態などのアプリケーションデータを永続的に格納しません。

Access Manager のアップグレード方針

Access Manager をアップグレードするための方針は一般に、第 1 章「アップグレードの計画」で説明されている多くの考慮事項 (アップグレードパス、Java ES コンポーネント間の依存関係、選択的なアップグレードかすべてのアップグレードか、複数インスタンスの配備など) によって異なります。

ここでは、Access Manager のアップグレード計画に影響する可能性のある問題を取り上げ、Access Manager の全般的な説明を詳細に記述します。

互換性の問題

Release 5 の Access Manager は Release 4 の Access Manager と下位互換性がありますが、Release 4 の Access Manager はメジャーリリースであり、旧バージョンモードで動作するように設定された場合を除いて、以前のリリースとの互換性はなくなりました。同様に、Release 5 の Access Manager も、旧バージョンモードで動作するように設定されないかぎり、Release 3 の Access Manager (または旧バージョンモードで動作している Release 4 の Access Manager) とは下位互換性がありません。

さらに、Release 5 の Access Manager は、どのモードにおいても Release 2 の Access Manager とは下位互換性がなく、Release 5 の Access Manager と Release 2 の Access Manager SDK との間では相互運用できません。

Release 5 の Access Manager は、より新しいレルムモードで動作するように設定された場合、複数のアイデンティティーリポジトリおよびデータ格納プロトコルをサポートします。レルムモードでの運用をサポートするには、ディレクトリデータを新しい構造に移行する必要があります。加えて、レルムモードは、Portal Server などのほかの Java ES コンポーネントや、Communications Express、Messaging Server などの Sun Java Communications Suite コンポーネントをサポートしていません。

ただし、旧バージョンモードで動作するように設定された場合、Release 5 の Access Manager は、若干の例外を除き Release 3 の Access Manager および対応するディレクトリデータと下位互換性があります。例外については、『Sun Java System Access Manager 7.1 リリースノート』(http://docs.sun.com/doc/820-0363?l=ja) を参照してください。

旧バージョンモードは、レルムモードの Access Manager と相互運用できないほかのバージョンの Access Manager ポリシーエージェントや、その他の Java ES コンポーネントをサポートするために必要です。この相互運用性はアップグレードの重要な考慮事項であり、ほとんどの Java ES 配備で、Access Manager を Release 5 旧バージョンモードにアップグレードすることが望ましいことを意味します。

ただし、旧バージョンモードで動作するように設定された場合でも、Release 5 の Access Manager は Release 3 またはそれ以前の Sun Java Communications Suite コンポーネントと互換性がありません。Access Manager を Release 5 にアップグレードする場合、Messaging Server および Calendar Server のユーザーをプロビジョニングするためには、Release 3 またはそれ以前の Delegated Administrator も Release 5 にアップグレードする必要があります。ただし、Messaging Server および Calendar Server 自体は Release 5 にアップグレードする必要はありません。

Release 5 の Access Manager コンソールは、Release 4 のコンソールと同様に、レルムモードと旧バージョンモードの両方をサポートしています。ただし、Access Manager を旧バージョンモードで動作するように設定した場合は、Release 2 および Release 3 で配布された旧バージョンのコンソールを引き続き使用できます。

Access Manager の依存関係

ほかの Java ES コンポーネントに対する Access Manager の依存関係が、Access Manager ソフトウェアのアップグレードおよび再設定の手順に影響を及ぼす可能性があります。たとえば、Access Manager のインタフェースまたは関数が変更された場合は、Access Manager が依存するコンポーネントのアップグレードされたバージョンが必要になる場合があります。このようなコンポーネントのアップグレードの必要性は、個々のアップグレードパスによって異なります。

Access Manager は、次の Java ES コンポーネントに依存します。

- **共有コンポーネント**: Access Manager は、特定の Java ES 共有コンポーネントに依存します (48 ページの表 1-9 を参照)。Access Manager をアップグレードするために、これらの共有コンポーネントのバージョンをアップグレードしなければならない場合があります。
- **Web コンテナ**: Access Manager は、Java ES Web Server または Java ES Application Server のどちらか、あるいは (Weblogic や WebSphere の) サードパーティー製 Web コンテナで提供される Web コンテナサービスに必ず依存しています。Access Manager のアップグレードの際に、アップグレードされた Access Manager 環境への Access Manager コンソールまたは認証 UI 用にカスタマイズされた JSP の移行が必要な場合があります。
- **Directory Server**: Access Manager は、設定データおよびユーザーデータの格納に使用される Directory Server に必ず依存しています。そのため、Access Manager のアップグレードによって、ディレクトリスキーマの拡張が必要になる場合があります。

Web コンテナのアップグレードシナリオ

Access Manager は、Web Server または Application Server のどちらかで提供される Web コンテナに配備できます。その結果、配備先の Web コンテナも同時に Release 5 にアップグレードする可能性があるために、Access Manager の Release 5 へのアップグレードが複雑になることがあります。この点では、次の表に示すように、可能性のある Web コンテナのアップグレードシナリオがいくつかあります。

表 14-4　Access Manager のアップグレード用 Web コンテナのアップグレードシナリオ

シナリオ	Access Manager が当初配備されていた Web コンテナ	アップグレード後に Access Manager が配備される Web コンテナ	適用可能な Access Manager のアップグレードパス : アップグレード元
シナリオ 1	Web Server 6.*x*	Web Server 6.*x*	Release 2 Release 3 Release 4
シナリオ 2	Web Server 6.*x*	Web Server 7.0	Release 2 Release 3 Release 4
シナリオ 3	Application Server 8.1	Application Server 8.1	Release 3 Release 4
シナリオ 4	Application Server 8.1	Application Server 8.2	Release 3 Release 4
シナリオ 5	Application Server 7*x*	Application Server 8.2	Release 2

表 14-4 内にあるアップグレードシナリオの中でも、特に Web コンテナのメジャーバージョンのアップグレードがある場合は、Access Manager をアップグレードする (たとえば、`amconfig` スクリプトを使用する) 際に適切な値を指定するように注意してください。

二重のアップグレード

44 ページの「二重のアップグレード : Java ES ソフトウェアとオペレーティングシステムソフトウェア」で説明されているように、二重のアップグレードは Access Manager ソフトウェアとオペレーティングシステムの両方がアップグレードされることを意味しますが、Access Manager では二重のアップグレードはサポートされていません。

その結果、二重のアップグレードが必要になる状況では、オペレーティングシステムのインストールまたはアップグレードを実行したあと、Access Manager を再インストールし、新しく設定する必要があります。

Java ES Release 4 からの Access Manager のアップグレード

ここでは、Access Manager の Java ES 2005Q4 (Release 4) から Java ES 5 (Release 5) へのアップグレードについて説明します。この節で説明する内容は、次のとおりです。

- はじめに
- Release 4 の Access Manager の完全アップグレード
- 複数インスタンスのアップグレード
- Release 4 の Access Manager SDK のみのアップグレード

はじめに

Java ES Release 4 Access Manager を Release 5 にアップグレードする場合は、アップグレード処理の次の側面を考慮してください。

- **一般的なアップグレード手順** : アップグレードは、以前のバージョンのバイナリを削除し、Release 5 を新しくインストールすることによって行います。用意されている `ampre71upgrade` スクリプトを使用して Release 4 を削除してから、Java ES インストーラを使用して Release 5 をインストールします。その後、`amconfig` スクリプトを使用して Access Manager を再設定し、`amupgrade` スクリプトを使用してディレクトリスキーマを移行します。
- **アップグレードの依存関係** : Access Manager は、いくつかの Java ES 共有コンポーネントに依存しています (48 ページの表 1-9 を参照)。Access Manager のアップグレードを実行すると、すべての共有コンポーネントは Java ES インストーラによって自動的に Release 5 にアップグレードされます。これには、Release 4 の Access Manager では必要ないが、新しい Java ES Monitoring Framework のサポートに必要な共有コンポーネントも含まれます。

  加えて、263 ページの「Access Manager の依存関係」で説明されているように、Release 5 の Access Manager は Directory Server および Web Server (あるいは、Application Server またはサードパーティー製 Web コンテナ) に依存します。ただし、これらはアップグレード依存関係が弱いものであり、これらのコンポーネントのアップグレードは、Access Manager の Release 5 へのアップグレードに関しては省略可能です。
- **下位互換性** : Release 5 の Access Manager は Release 4 と互換性がありますが、それ以前の Access Manager リリースとは互換性がありません (262 ページの「互換性の問題」を参照)。

- **アップグレードのロールバック** : Access Manager のアップグレードをロールバックするためのユーティリティーはありません。実際、Access Manager を元の状態にロールバックするためには膨大な量の再設定を行う必要があるため、ロールバックは非効率的です。ロールバックするための最適な方法は、バックアップされた設定ファイルを使用して並列システムを作成し、この並列システムをテストしてからアップグレードを実行する方法です。これにより、必要に応じて並列システムに戻すことができます。
- **プラットフォームの問題** : Access Manager をアップグレードするための一般的な手順は、Solaris および Linux の両オペレーティングシステムで共通です。以降の手順説明では、必要に応じてプラットフォーム別のコマンドまたはファイル位置を示します。

Release 4 の Access Manager の完全アップグレード

この節では、Java ES Release 4 から Java ES Release 5 への、Access Manager の完全アップグレードを実行する方法について説明します。

- アップグレード前の作業
- Release 4 の Access Manager のアップグレード
- Access Manager のアップグレードの確認
- アップグレード後の作業
- アップグレードのロールバック

アップグレード前の作業

Access Manager をアップグレードする前に、次の作業を実行してください。

- 現在のバージョン情報の確認
- Access Manager が依存するコンポーネントのアップグレード
- Directory Server のデータのバックアップ
- Release 4 Access Manager の設定情報のバックアップ
- Web コンテナのカスタマイズファイルのバックアップ
- Release 4 の Access Manager のログファイルとデバッグファイルのバックアップ
- カスタマイズしたローカライズ版ファイルのバックアップ
- 必要な設定情報とパスワードの入手

現在のバージョン情報の確認

Access Manager の現在のバージョンは、次のコマンドを使用することにより確認できます。

```
AccessManager-base/bin/amadmin --version
```

表 14-5　Access Manager のバージョン確認の出力

Java ES のリリース	Access Manager のバージョン番号
Release 2	`6.2`
Release 3	`6 2005Q1`
Release 4	`7 2005Q4`
Release 5	`7.1`

Access Manager が依存するコンポーネントのアップグレード

一般的には、コンピュータシステム上 (および、コンピューティング環境内) のすべての Java ES コンポーネントを Java ES Release 5 にアップグレードすることが推奨されます。Access Manager は、いくつかの共有コンポーネントに対して強いアップグレード依存関係を持っています (48 ページの表 1-9 を参照)。

Access Manager 製品コンポーネントが依存するコンポーネントをアップグレードすることを選択した場合は、Access Manager をアップグレードする前に、(すでにアップグレードした可能性があるものを省いて) 次の順序で実行してください。共有コンポーネントのアップグレードは通常、Java ES インストーラによって自動的に実行されます。

1. **共有コンポーネント** : Java ES 共有コンポーネントを Release 5 に同期する手順は、67 ページの「Java ES 共有コンポーネントのアップグレード」で説明されています。ただし、Access Manager の Release 5 へのアップグレードを実行すると、Access Manager で必要な共有コンポーネントはすべて、Java ES インストーラによって自動的にアップグレードされます。
2. **Directory Server (弱いアップグレード依存関係)**: Directory Server を Release 5 にアップグレードする手順は、103 ページの第 5 章「Directory Server」で説明されています。
3. **Web コンテナソフトウェア (弱いアップグレード依存関係)**: Web Server または Application Server をアップグレードする手順は、それぞれ 133 ページの第 7 章「Web Server」および 203 ページの第 11 章「Application Server」で説明されています。

Access Manager よりも先に Web コンテナソフトウェアをアップグレードしない場合、アップグレード手順は (amconfig スクリプトを使用して) Access Manager を設定し、既存の Web コンテナに配備します。

Directory Server のデータのバックアップ

Access Manager のアップグレードプロセスでは、Directory Server のスキーマを変更するスクリプトを使用します。このため、Access Manager をアップグレードする前に、Directory Server コンソールや db2bak などのコマンド行ユーティリティーを使用して Directory Server データをバックアップしてください。db2ldif を使用して、Access Manager スキーマと DIT (ディレクトリ情報ツリー) をバックアップできます。

Directory Server のバックアップの詳細については、『Sun Java System Directory Server Enterprise Edition 6.0 管理ガイド』(http://docs.sun.com/doc/820-0291?l=ja) を参照してください。

Release 4 Access Manager の設定情報のバックアップ

Release 5 の Access Manager ソフトウェアの再設定には Release 4 バージョンの再設定が必要なため、設定ファイルをわかりやすい場所にバックアップすることは重要です。バックアップが推奨されるファイルは次のとおりです。

- AMConfig.properties ファイル
  AccessManagerConfig-base/config/AMConfig.properties
- serverconfig.xml ファイル
  AccessManagerConfig-base/config/serverconfig.xml
- Web コンテナ設定ファイル :
  - Web Server については、260 ページの表 14-3 に記載されている server.policy および server.xml ファイルの場所を参照してください。
  - Application Server については、260 ページの表 14-3 に記載されている server.policy および domain.xml ファイルの場所を参照してください。
  - サードパーティー製 Web コンテナ : 所定の設定ファイル
- 認証モジュールとカスタマイズモジュール用の JAR ファイル

  AccessManager-base/lib

Web コンテナのカスタマイズファイルのバックアップ

Access Manager によって参照される Web コンテナカスタマイズファイルがある場合、それらのファイルをバックアップすることが推奨されます。このようなカスタマイズの例には次のものがあります、

- カスタマイズした Access Manager コンソール JSP ページ :
  - レルムモード / 旧バージョンモードのコンソール (Java ES Release 4 で配布)
    AccessManager-base/web-src/services/

- 旧バージョンのコンソール (Java ES Release 2 および 3 で配布)
  AccessManager-base/web-src/applications/

- カスタマイズした認証 UI JSP ページ。
  AccessManager-base/web-src/services/
- カスタマイズした XML ファイル。
  AccessManagerConfig-base/config/xml/

ヒント Access Manager をアップグレードしたあとで、バックアップしたコードを使用してカスタマイズを再適用できるように、カスタマイズ内容を記録しておきます。

Release 4 の Access Manager のログファイルとデバッグファイルのバックアップ

システム状態情報の分析に使用できるように、ログファイルとデバッグファイルをバックアップして消失を防ぐことをお勧めします。これらのファイルは次の場所にあります。

- デバッグファイル
  /var/*AccessManager-base*/debug
- ログファイル
  /var/*AccessManager-base*/logs

カスタマイズしたローカライズ版ファイルのバックアップ

Java ES インストーラによってインストールされたローカライズ版ファイルをカスタマイズした場合、または Java ES インストーラではインストールされない新しい言語のローカライズ版を追加した場合は、これらのカスタマイズをバックアップすることをお勧めします。カスタマイズの例には次のものがあります。

- Access Manager のユーザーインタフェース用にカスタマイズしたローカライズ版のファイル *AccessManager-base*/locale/*_*Locale*.properties
- 認証 UI 用にカスタマイズした JSP ページ
  AccessManager-base/web-src/services/config/auth/default_*Locale*
- カスタマイズした翻訳版オンラインヘルプ
  AccessManager-base/web-src/services/html/*Locale*

必要な設定情報とパスワードの入手

Access Manager をアップグレードするには、次に示すような、特定の設定情報を提供する必要があります。

- Access Manager 管理者のユーザー ID とパスワード
- LDAP ユーザー ID とパスワード

- Access Manager が使用している Directory Server インスタンスのディレクトリマネージャー名およびパスワード

Release 4 の Access Manager のアップグレード

Java ES Release 5 への Access Manager ソフトウェアのアップグレードには、Access Manager を再設定する手順と、Access Manager データを移行する手順が含まれます。

アップグレードの概要

Access Manager をアップグレードするための手順は、次の手順で構成されます。

1. Access Manager モバイルアクセスソフトウェアをアップグレードします。
2. Java ES Release 4 バージョンの Access Manager を削除します。`ampre71upgrade` スクリプトを使用します。
3. Release 5 へのアップグレードをローカライズする必要がある場合は、Release 4 のローカライズ版パッケージを削除します。この手順は、手動で実行してください。
4. Java ES Release 5 バージョンの Access Manager をインストールします。Java ES インストーラを使用し、「あとで設定」オプションを指定します。
5. Access Manager 用 JSP を再カスタマイズします。
6. Access Manager の配備を取り消し、再設定し、Web コンテナに再配備します。`amconfig` スクリプトを使用します。
7. ディレクトリ構造とスキーマを更新します。`amupgrade` スクリプトを使用します。

以降の手順中で、これらの各手順について説明します。

アップグレード手順

1. Access Manager モバイルアクセスソフトウェアをアップグレードします。

   Release 4 バージョンにパッチを適用することによって、Access Manager モバイルアクセスソフトウェアをアップグレードする必要があります。必要なパッチを次の表に示します。

表 14-6 Access Manager モバイルアクセスソフトウェアをアップグレードするためのパッチ[1]

説明	パッチ ID: Solaris 9 および 10	パッチ ID: Linux
Mobile Access ソフトウェア	119530-05 (SPARC) 119531-05 (x86)	119532-05 • `sun-identity-mobileaccess-6.2-25.3.i386.rpm` • `sun-identity-mobileaccess-config-6.2-25.3.i386.rpm`

1. この表に示されているパッチのリビジョン番号は、Java ES Release 5 にアップグレードするための最低要件です。表に記載されているものよりも新しいリビジョンのパッチが使用可能になった場合は、新しいほうのパッチを使用してください。

a. 表 14-6 のパッチ番号を参照して、必要なパッチを入手します。

パッチは http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access から /tmp にダウンロードします。

b. パッチの README ファイルに示されているパッチ適用前の手順をすべて実行します。

c. パッチによって要求される、次のパラメータの値を入手します。

表 14-7 モバイルアクセスのパッチパラメータ

パラメータ	値
Directory Manager DN	デフォルト : cn=Directory Manager
ディレクトリマネージャーのパスワード	

d. 表 14-6 のパッチを適用します。

Solaris の場合 :
`patchadd /tmp/patch_ID`

Linux の場合 :
`./update`

パッチの README ファイルに示されているパッチ適用後の手順をすべて実行します。

2. Java ES Release 4 バージョンの Access Manager を削除します。

a. Release 4 の Access Manager がインストールされたコンピュータに root としてログインするか、スーパーユーザーになります。

```
su -
```

b. Java ES Release 5 配布パッケージ内の *os_arch*/Product/identity_svr/Tools ディレクトリに移動します。ここで、*os_arch* はプラットフォーム (Solaris_sparc など) に対応します。

c. ampre71upgrade スクリプトによって要求される、次のパラメータの値を入手します。

表 14-8　Access Manager 設定パラメータ : ampre71upgrade

パラメータ	値
Directory Server ホスト	完全修飾名を設定する : *hostname.domain*
Directory Server ポート	非 SSL ポート番号を指定する[1] デフォルト : 389
最上位管理者の DN	デフォルト : uid=amadmin,ou=People,*default_org_DN*
最上位管理者のパスワード	
バックアップファイルを格納するディレクトリ	デフォルト : *AccessManager-base*

1. Directory Server の SSL ポートを SSL のデフォルト値である 636 などに指定すると、アップグレード前プロセスは正しく完了しません。

d. Directory Server が稼働していることを確認し、稼働していない場合は起動します。

e. ampre71upgrade スクリプトを実行します。

```
./ampre71upgrade
```

このスクリプトは Access Manager の設定ファイルをバックアップし、Release 4 のベースパッケージを削除します。ローカライズ版のパッケージは、下記の手順 3 に従って、手動で削除する必要があります。

3. Release 5 へのアップグレードをローカライズする必要がある場合は、Release 4 のローカライズ版パッケージを削除します。

上記の手順 2 で実行した ampre71upgrade スクリプトではローカライズ版パッケージが削除されないため、次のように、手動で削除してください。

Solaris の場合 :

a. ローカライズ版パッケージを確認します。

```
pkginfo | grep SUNWaml
pkginfo | grep SUNWamclnt
pkginfo | grep SUNWamdistauth
```

b. 上記の手順 a でローカライズ版パッケージが見つかった場合は、削除します。

```
pkgrm SUNWaml<Locale>
pkgrm SUNWamclnt<Locale>
pkgrm SUNWamdistauth<Locale>
```

Linux の場合 :

a. ローカライズ版 RPM を確認します。

```
rpm -qa | grep sun-identity-sdk-*
rpm -qa | grep sun-identity-clientsdk-*
rpm -qa | grep sun-identity-distauth-*
```

b. 上記の手順 a でローカライズ版 RPM が見つかった場合は、削除します。

```
rpm -e sun-identity-sdk-Locale-*
rpm -e sun-identity-clientsdk-Locale-*
rpm -e sun-identity-distauth-Locale-*
```

4. Java ES Release 5 バージョンの Access Manager をインストールします。

次の手順を実行します。

a. Release 4 の Access Manager がホストされたコンピュータ上で Java ES インストーラを起動します。

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

ここで、*os_arch* は、`Solaris_sparc` などのプラットフォームに対応します。コマンド行インタフェースには、`installer -nodisplay` オプションを使用します。

ようこそページと、ライセンス使用許諾契約ページが表示されてから、コンポーネント選択ページが表示されます。Java ES インストーラが直接アップグレードできるインストール済みのコンポーネントが検出された場合、それらのコンポーネントは「アップグレード可」の状態で表示されます。

b. コンポーネント選択ページから Access Manager を選択します。

c. Release 4 がインストールされていたのと同じインストールディレクトリを指定します。

d. 「あとで設定」オプションを選択します。

e. 必要に応じて、ローカライズ版のパッケージをインストールするオプションを選択します。

f. インストールが完了したら、Java ES インストーラを終了します。

5. Access Manager 用 JSP を再カスタマイズします。

268 ページの「Web コンテナのカスタマイズファイルのバックアップ」で保存した、Access Manager コンソールおよび認証ユーザーインタフェース (UI) 用 JSP に、Release 4 のカスタマイズを再適用します。

続いて、カスタマイズ JSP ファイルを次の適切なディレクトリにコピーします。

- レルム / 旧バージョンの Access Manager コンソール
  AccessManager-base`/web-src/services/console`

- 旧バージョンのみの Access Manager コンソール
  AccessManager-base/web-src/applications/console
- 認証 UI: *AccessManager-base*/web-src/services/config/auth/default または *AccessManager-base*/web-src/services/config/auth/default_*Locale* (*Locale* は ja などのロケールインジケータ)

詳細については、『Sun Java System Access Manager 7.1 Developer's Guide』(http://docs.sun.com/doc/819-4675) を参照してください。

6. Access Manager の配備を取り消し、再設定し、Web コンテナに再配備します。

   amconfig スクリプトを実行して、使用している Web コンテナ用に Access Manager を設定します。amconfig スクリプト (および、関連付けられた amsamplesilent テンプレート入力ファイル) は次のディレクトリにあります。

   AccessManager-base/bin

   amconfig スクリプトおよび amsamplesilent テンプレートファイルの詳細については、『Sun Java System Access Manager 7.1 管理ガイド』(http://docs.sun.com/doc/820-0842?l=ja) を参照してください。

   次の手順を実行して、Access Manager を再設定し、Web コンテナに再配備します。

   a. 267 ページの「Access Manager が依存するコンポーネントのアップグレード」で説明しているように Web コンテナソフトウェアをアップグレードする場合、アップグレードが完了していることを確認します。

   b. Web コンテナの管理インスタンスが実行中であり、下記の表に示される、amconfig スクリプトでサポートされるモードであることを確認してください。

表 14-9 amconfig でサポートされる管理サーバーモード

Web コンテナ	サポートされるモード	デフォルトのポート番号
Application Server (8.*x*): Java ES Release 3、4、および 5	SSL (セキュリティー保護)	4849
	非 SSL	
Web Server (7.0): Java ES Release 5	SSL (セキュリティー保護)	8989
Web Server (6.*x*): Java ES Release 2、3、および 4	非 SSL	8888

   c. Web コンテナが SSL モードで実行されている場合は、コンテナの SSL 証明書の期限が切れていなくて、まだ有効であることを確認してください。

d. Access Manager が Release 5 の Web Server に配備されている場合は、Access Manager と同じインスタンスで実行されている、Access Manager に依存しているすべての Java ES コンポーネントを無効にします。

これらは、Portal Server などのコンポーネント、または Communications Express、Instant Messaging、Delegated Administrator などの Sun Java Communications Suite コンポーネントである可能性があります。

手順は次のとおりです。

I. `https://host:8989` に管理者としてログインします。

II. 「仮想サーバーを編集」に移動します。

III. 「Web アプリケーション」タブを選択します。

IV. Access Manager に依存するすべてのアプリケーションにチェックマークを付けます。

V. 「無効にする」をクリックします。

VI. 「保存」をクリックします。

VII. 「配備の保留中 I 設定を配備」をクリックします。

この設定変更は、Web Server インスタンスに伝播されます。

e. Directory Server と適切な Web コンテナが稼働していることを確認します。

f. `amsamplesilent` テンプレート入力ファイルを基に、`amconfig` 入力ファイルを作成します。

```
cp amsamplesilent config-file
```

以降の手順では、*config-file* が `amsamplesilent` と同じディレクトリに存在することを前提にしています。

g. *config-file* 内の設定パラメータを設定します。

すべてのパラメータを正確に設定する必要があります。次の表に示すように、設定値には、`AMConfig.properties` ファイルから移行できるものと、アップグレード手順の中で新しく設定するものがあります。

表 14-10 Access Manager 設定パラメータ : `amconfig`

パラメータ	値
アップグレードパラメータ	
DEPLOY_LEVEL	配備取り消しの場合は `26` を設定、または再設定と配備の場合は `1` に設定
DIRECTORY_MODE	`5` に設定

表 14-10　Access Manager 設定パラメータ : amconfig (続き)

パラメータ	値
AM_REALM[1]	旧バージョンモードを有効にする場合は disabled に設定 レルムモードを有効にする場合は enabled に設定 デフォルト : enabled
JAVA_HOME	JDK Release 5 ディレクトリに設定 : /usr/java/jdk1.5.0_04/
WEB_CONTAINER	Web Server 7.*x* の場合は WS に設定 Web Server 6.*x* の場合は WS6 に設定 Application Server 8.*x* の場合は AS8 に設定 IBM WebSphere 5.*x* の場合は WAS5 に設定 BEA WebLogic 8.*x* の場合は WL8 に設定 *config-file* の対応するセクションのみを設定する
WS_INSTANCE (Web Server 7.*x* を Web コンテナとして使用している場合)	大文字と小文字が区別されるインスタンス設定ディレクトリ名に設定 : https-*configName*/ このディレクトリは次のパスにある : *WebServer7Config-base*/https-*configName*/
WS61_INSTANCE (Web Server 6.*x* を Web コンテナとして使用している場合)	大文字と小文字が区別されるインスタンス設定ディレクトリ名に設定 : https-*instanceName* このディレクトリは次のパスにある : *WebServer6-base*/https-*instanceName*/
AS81_INSTANCE (Application Server 8.*x* を Web コンテナとして使用している)	Application Server 8.*x* *instanceName* に設定 デフォルト : server
AS81_INSTANCE_DIR (Application Server 8.*x* を Web コンテナとして使用している)	インスタンスの Application Server 8.*x* ドメインディレクトリに設定。デフォルトでは次の場所にある : *AppServer8Config-base*/domains/domain1
AS81_DOCS_DIR (Application Server 8.*x* を Web コンテナとして使用している)	インスタンスの Application Server 8.*x* docroot ディレクトリに設定。デフォルトでは次の場所にある : *AppServer8Config-base*/domains/domain1/docroot
AMConfig.properties から移行される値	
SERVER_PROTOCOL	com.iplanet.am.server.protocol
SERVER_PORT	com.iplanet.am.server.port

表 14-10　Access Manager 設定パラメータ : amconfig (続き)

パラメータ	値
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. レルムモードと旧バージョンモードの詳細については、262 ページの「互換性の問題」を参照してください。
2. このパラメータの値は、以前のバージョンの Access Manager と同様にしてください。
3. Access Manager と Access Manager SDK の両方が配備されている場合、このプロパティーの値は、Access Manager インスタンスとそれに関連付けられた Access Manager SDK インスタンスの両方で同じにしてください。

その他のパラメータについては、Web コンテナまたはパスワードを変更する場合を除いて、アップグレードする Release 4 の設定で使われていたのと同じ値を指定します。たとえば、Web Server を Release 5 にアップグレードした場合は、次の値を指定します。

表 14-11　amconfig のパラメータ : Release 5 の Web Server

パラメータ	値
WS_CONFIG	Web Server 設定の名前 : *configName*
WS_INSTANCE	https-*configName*
WS_HOME	*WebServer7Config-base*
WS_PROTOCOL	http または https
WS_HOST	Web Server インスタンスが接続を待機している完全修飾ホスト名
WS_PORT	Web Server インスタンスが接続を待機しているポート
WS_ADMINPORT	Web Server 管理インスタンスが接続を待機しているポート

表 14-11　amconfig のパラメータ : Release 5 の Web Server

パラメータ	値
WS_ADMIN	Web Server 管理者ユーザー ID
WS_ADMINPASSWD	Web Server 管理者パスワード

h. amconfig を実行して、Access Manager の配備を取り消します。

config-file 内の DEPLOY_LEVEL の値を 26 に設定します。

```
cd /AccessManager-base/bin
./amconfig -s AccessManager-base/bin/config-file
```

i. 共通エージェントコンテナが実行されていることを確認してください。

```
netstat -an | grep 11163
```

実行されていない場合は、起動します。

```
/usr/sbin/cacaoadm start
```

j. amconfig を実行して Access Manager を再設定し、Web コンテナに配備します。

config-file 内の DEPLOY_LEVEL の値を 1 に設定します。

```
cd /AccessManager-base/bin
./amconfig -s AccessManager-base/bin/config-file
```

7. ディレクトリ構造とスキーマを更新します。

Release 5 の Access Manager は Release 4 のディレクトリ構造と共存しますが、Release 5 の機能をサポートするためには構造を変更する必要があります。amupgrade スクリプトを実行して、Access Manager のディレクトリ構造とスキーマを Release 5 に更新します。このスクリプトは次のディレクトリにインストールされています。

- *Solaris の場合 :*
  AccessManager-base/upgrade/scripts
- *Linux の場合 :*
  AccessManager_base/identity/upgrade/scripts

a. amupgrade スクリプトによって要求される、次のパラメータの値を入手します。

表 14-12　Access Manager 設定パラメータ : amupgrade

パラメータ	値
Directory Server の完全修飾ホスト名	完全修飾名を設定する : *hostname.domain*
Directory Server ポート	非 SSL ポート番号を指定する[1] デフォルト : 389
Directory Manager DN	デフォルト : cn=Directory Manager
Directory Manager パスワード	
最上位管理者の DN	デフォルト : uid=amadmin,ou=People,*default_org_DN*
最上位管理者のパスワード	
Enable Realm Mode Release 4 のレルムモードからアップグレードする場合、このパラメータ値は要求されません。	Y/N: Yes は、レルムモードが有効であり、サービスデータが新しいレルムツリーに移行されたことを意味します。[2] No (デフォルト) はサービスデータが旧バージョンモードのままであることを意味します。

1. SSL のデフォルト値である 636 とは異なる Directory Server SSL ポートを指定してください。
2. 281 ページの「レルムモードへの移行」を参照してください。

b. amupgrade スクリプトを実行します。

```
cd AccessManager-base/upgrade/scripts
./amupgrade
```

スクリプトによるアップグレードが成功すると、「Upgrade completed」というメッセージが表示されます。

c. 次のアップグレードログファイルをチェックし、ディレクトリスキーマ拡張についての情報を確認します。

Solaris の場合 :

```
/var/sadm/install/logs/
        Sun_Java_System_Access_Manager_upgrade_dit_log.mmddhhmm
```

Linux の場合 :

```
/var/log/Sun_Java_System_Access_Manager_upgrade_dit_log.mmddhhmm
```

8. 275 ページの手順 d で無効にしたコンポーネントをすべて有効にします。
9. Access Manager が配備される Web コンテナを再起動します。

10. Access Manager を起動します。

    Access Manager が配備される Web コンテナを再起動します。

Access Manager のアップグレードの確認

アップグレード手順が完了したら、次のようにして、アップグレードが成功したかどうかを確認します。

1. 次のコマンドを使って Access Manager パッケージのアップグレードを確認します。

   AccessManager-base/bin/amadmin --version

   出力値については、267 ページの表 14-5 を参照してください。

2. /var/sadm/install/logs ディレクトリ内の次のインストーラログファイルをチェックして、アップグレードの状態を確認します。

   - Java_Shared_Component_Install.*timestamp*
   - Java_Enterprise_System_install.*Atimestamp*
   - Java_Enterprise_System_install.*Btimestamp*
   - Java_Enterprise_System_Summary_Report_install.*timestamp*

3. amupgrade スクリプト実行中のエラーを端末ウィンドウでチェックすることによって、Access Manager の移行の状態を確認します。

   また、/var/sadm/install/logs ディレクトリにある次のログファイルもチェックします。

   Sun_Java_System_Access_Manager_upgrade_dit_log.*timestamp*

4. Access Manager トラブルシューティングファイルで、エラーの有無を確認します。

   これらのファイルは、AMConfig.properties ファイルの com.iplanet.services.debug.directory プロパティーで指定された場所に格納されています。デフォルト値は次のとおりです。

   Solaris の場合:
   /var/opt/SUNWam/debug

   Linux の場合:
   /var/opt/sun/identity/debug

アップグレード後の作業

次の状況に対処するために必要なアップグレード後の手順に注意してください。

- レルムモードへの移行

- SAML (Security Assertion Markup Language)

レルムモードへの移行

Access Manager を Release 5 にアップグレードするときにレルムモードに移行した (つまり、amupgrade を使用してディレクトリ構造とスキーマを更新するときに、レルムモードの有効化に対して「Yes」を選択した) 場合は、次の手順を実行します。

1. *AccessManagerConfig-base*/config/AMConfig.properties ファイルを開きます。
2. 次のプロパティーの値を確認します。

   com.sun.identity.sm.ldap.enableProxy
3. このプロパティーが false に設定されていない場合は、手動で false に設定します。

SAML (Security Assertion Markup Language)

SAML (Security Assertion Markup Language) サービスを使用する場合、Access Manager コンソールを使って SAML 認証モジュールを追加および有効化する必要があります。SAML 認証モジュールインスタンスの作成の詳細については、『Sun Java System Access Manager 7.1 管理ガイド』(http://docs.sun.com/doc/820-0842?l=ja) を参照してください。

アップグレードのロールバック

Access Manager をアップグレード前の状態にロールバックするためのスクリプトは用意されていません。ロールバックプロセスは、アップグレード前の作業でバックアップした Access Manager データ (269 ページの「Release 4 の Access Manager のログファイルとデバッグファイルのバックアップ」を参照) を使用して手動で行う必要があります。アップグレードのバックアウトは非常に難しく、実際的ではありません。

ロールバックするための 1 つの方法は、Release 4 の再インストールを実行し、バックアップしたすべての設定ファイルを正しい場所に移行することです。別の方法としては、バックアップした設定ファイルを使用してアップグレードの前に並列システムを作成し、アップグレードを試みる前にその並列システムをテストすることもできます。

複数インスタンスのアップグレード

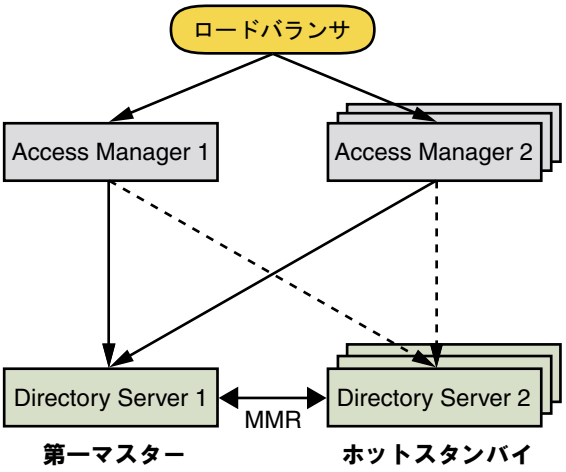
配備のアーキテクチャーによっては、高可用性とスケーラビリティーを実現するために、Access Manager が複数のコンピュータシステムに配備される場合があります。

通常は、サービスを中断させずに連続的に Access Manager インスタンスをアップグレードするのが理想的です。ここでは、Release 4 の Access Manager から Release 5 への、このような順次アップグレードを実行するための手順について説明します。

注	現在のリリースでは、同一ホストシステムにインストールされている Access Manager の複数インスタンスのアップグレードはサポートされていません。同一ホストに複数のインスタンスが存在する場合は、メインインスタンスをアップグレードしてから追加インスタンスを再作成する必要があります。

次の図に示されている配備アーキテクチャーは、順次アップグレードの手順を説明するために使用されます。

図 14-1 複数の Access Manager インスタンスに対する配備アーキテクチャーの例



このアーキテクチャーでは、複数の Access Manager インスタンスがロードバランサを介してアクセスされ、これらのインスタンスは順に、マルチマスターレプリケーション (MMR) 用に設定されたディレクトリにアクセスします。ほかの Directory Server レプリケーションスキーマも使用できますが、MMR は可用性が高くてスケーラブルな、代表的なディレクトリサービスです。図 14-1 では、アップグレード手順を説明しやすくするために、Access Manager と Directory Server の複数のインスタンスがグループ化されています。たとえば、Access Manager 2 は、Access Manager の 2 〜 n 番目のインスタンスを表しています。

Release 4 の Access Manager から Release 5 への順次アップグレードを実行する手順は、次の相互運用性に基づいています。ディレクトリスキーマがまだ Release 5 に更新されていなければ、Release 5 の Access Manager と Release 4 の Access Manager インスタンスは、共存して同じディレクトリに対して同時に動作できます。

そのため、単一の Directory Server インスタンスをポイントする Access Manager インスタンスの場合は、すべての Access Manager インスタンスがアップグレードされるまでディレクトリスキーマの更新を保留することによって、順次アップグレードを実行できます。

Release 4 の Access Manager から Release 5 への順次アップグレードは、次の手順を使用して実行できます。

1. すべての Access Manager インスタンスに関する Release 4 の設定情報をバックアップします。

   260 ページの表 14-3 を参照してください。

2. Access Manager1 をアップグレードします。

   a. ロードバランサで Access Manager 1 を無効にします。

      これで、要求は Access Manager 1 に経路指定されなくなります。

   b. Access Manager1 を部分的にアップグレードします。

      270 ページの「Release 4 の Access Manager のアップグレード」の説明に従って Access Manager をアップグレードします。ただし、ディレクトリ構造とスキーマの更新 (278 ページの手順 7) は除きます。

   c. ロードバランサで Access Manager 1 を有効にします。

3. Access Manager 2 ～ Access Manager n をアップグレードします。

   簡潔にするために、以降の手順では「Access Manager 2」は Access Manager 2 ～ Access Manager n を意味します。

   a. ロードバランサで Access Manager 2 を無効にします。

      これで、要求は Access Manager 2 に経路指定されなくなります。

   b. Access Manager2 を部分的にアップグレードします。

      270 ページの「Release 4 の Access Manager のアップグレード」の説明に従って Access Manager の各インスタンスをアップグレードします。ただし、ディレクトリ構造とスキーマの更新 (278 ページの手順 7) は除きます。

   c. ロードバランサで Access Manager 2 を有効にします。

      要求は、ふたたび Access Manager 2 に経路指定されるようになります。

4. Directory Server1 のディレクトリ構造とスキーマを更新します。

   278 ページの手順 7 の説明に従って、amupgrade スクリプトを使用します。Directory Server 1 のスキーマが更新された場合、Access Manager 1 から n は引き続き機能します。

Release 4 の Access Manager SDK のみのアップグレード

配備構成によっては、1 台以上のコンピュータシステムに、ほかの Access Manager コンポーネントをインストールせずに、Access Manager SDK コンポーネントがインストールされることがあります。Access Manager SDK は Access Manager に対するリモートインタフェースとして機能し、Access Manager と同じ運用モード (旧バージョンまたはレルム) に合わせて再設定する必要があります。

Access Manager SDK と、その SDK がリモートインタフェースとして機能する対象となる Access Manager の両方を Release 5 にアップグレードしてください。ただし、Release 5 の Access Manager は Release 4 の Access Manager SDK と下位互換性があるため、一般にはまず Access Manager をアップグレードしてから、ほかのコンピュータ上で Access Manager SDK をアップグレードしてください。

Access Manager へのリモートインタフェースであるこの SDK は、Directory Server にアクセスするように設定する必要はありません。Portal Server などの、Web コンテナサービスに依存する Web コンポーネントをサポートするために Access Manager SDK を使用する場合は、対応する Web コンテナに合わせて Access Manager SDK を設定する必要があります。ただし、Access Manager SDK は非 Web コンポーネントをサポートすることもでき、その場合 Web コンテナは必要ありません。

これまでに述べたような特性から、Access Manager SDK をアップグレードする手順は、Access Manager の完全アップグレードの手順の一部です。

この節では、Access Manager SDK のみを Java ES Release 4 から Java ES Release 5 にアップグレードする方法について説明します。

- アップグレード前の作業
- Release 4 の Access Manager SDK のアップグレード
- Access Manager SDK のアップグレードの確認
- アップグレードのロールバック

アップグレード前の作業

Access Manager SDK のアップグレード前作業は Access Manager の完全アップグレード (266 ページの「アップグレード前の作業」を参照) の場合と同じですが、Directory Server に関連する作業と、Access Manager 管理ツールの JSP のカスタマイズに関連する作業は除外します。Access Manager SDK に対して必要なアップグレード前作業は次のとおりです。

- 267 ページの「Access Manager が依存するコンポーネントのアップグレード」

ただし、Access Manager SDK については、Directory Server への依存関係はなく、Access Manager SDK が Web コンテナ内で動作する場合にのみ Web コンテナソフトウェアへの依存関係があります。

- 268 ページの「Release 4 Access Manager の設定情報のバックアップ」

  ただし、Access Manager SDK については、Access Manager SDK が Web コンテナ内で動作する場合にのみ、Web コンテナ設定ファイルをバックアップする必要があります。

- 269 ページの「Release 4 の Access Manager のログファイルとデバッグファイルのバックアップ」

  また、これらのファイルにアクセスするための管理ユーザー名とパスワードを入手する必要もあります。

Release 4 の Access Manager SDK のアップグレード

Access Manager SDK のアップグレード手順は Access Manager の完全アップグレードの場合と同じですが、ローカライズ版、Access Manager 管理ツールの JSP のカスタマイズ、およびディレクトリスキーマの移行に関連する作業は除外します。

1. Access Manager SDK の Java ES Release 4 バージョンを削除します。

   271 ページの「Java ES Release 4 バージョンの Access Manager を削除します。」の指示に従いますが、削除するのは Access Manager SDK だけです。

2. Java ES Release 5 バージョンの Access Manager SDK をインストールします。

   273 ページの「Java ES Release 5 バージョンの Access Manager をインストールします。」の指示に従いますが、インストールするのは Access Manager SDK だけです。

3. Access Manager SDK を再設定します。

   274 ページの「Access Manager の配備を取り消し、再設定し、Web コンテナに再配備します。」の指示に従いますが、異なる点として、`DEPLOY_LEVEL` パラメータを次のように設定します。

   - Access Manager SDK を Web コンテナ用に設定する場合 :
     `DEPLOY_LEVEL=4` (SDK をアップグレードし Web コンテナを設定する)
   - Access Manager SDK を Web コンテナ用に設定しない場合 :
     `DEPLOY_LEVEL=3` (SDK のみをアップグレードする)

Access Manager SDK のアップグレードの確認

Access Manager SDK のアップグレードが成功したかどうかは、3 とおりの方法で確認できます。

- Portal Server 、または Access Manager とのインタフェースに Access Manager SDK を使用するその他のコンポーネントを実行し、認証が機能することを確認します。
- 次の場所に用意されている、Access Manager SDK サンプルを実行します。

  AccessManager-base/samples/sdk
- AMConfig.properties ファイル内の com.iplanet.am.version プロパティーの値を確認します。

  AccessManagerConfig-base/config/AMConfig.properties

アップグレードのロールバック

Access Manager をアップグレード前の状態にロールバックするためのスクリプトは用意されていません。ロールバックプロセスは、アップグレード前の作業でバックアップした Access Manager データ (269 ページの「Release 4 の Access Manager のログファイルとデバッグファイルのバックアップ」を参照) を使用して手動で行う必要があります。アップグレードのバックアウトは非常に難しく、実際的ではありません。

ロールバックするための 1 つの方法は、R4 の再インストールを実行し、バックアップしたすべての設定ファイルを正しい場所に移行することです。別の方法としては、バックアップした設定ファイルを使用してアップグレードの前に並列システムを作成し、アップグレードを試みる前にその並列システムをテストすることもできます。

Java ES Release 3 からの Access Manager のアップグレード

Java ES 2003Q1 (Release 3) の Access Manager または Access Manager SDK を Release 5 にアップグレードする手順は、複数インスタンスのアップグレードを実行する方法を除き、Release 4 の Access Manager または Access Manager SDK を Release 5 にアップグレードする手順とほぼ同じです。

- Release 3 の Access Manager のアップグレード
- 複数インスタンスのアップグレード

注 Release 5 の Access Manager は、一部の Release 3 Sun Java Communications Suite コンポーネントと互換性がありません。Access Manager を Release 5 にアップグレードする場合、Messaging Server および Calendar Server のユーザーをプロビジョニングするためには、Release 3 またはそれ以前の Delegated Administrator も Release 5 にアップグレードする必要があります。ただし、Messaging Server および Calendar Server 自体は Release 5 にアップグレードする必要はありません。

Release 3 の Access Manager のアップグレード

Release 3 の Access Manager または Access Manager SDK を Release 5 にアップグレードするには、「Release 4」を「Release 3」と読み替えた上で、265 ページの「Java ES Release 4 からの Access Manager のアップグレード」の手順を使用してください。

複数インスタンスのアップグレード

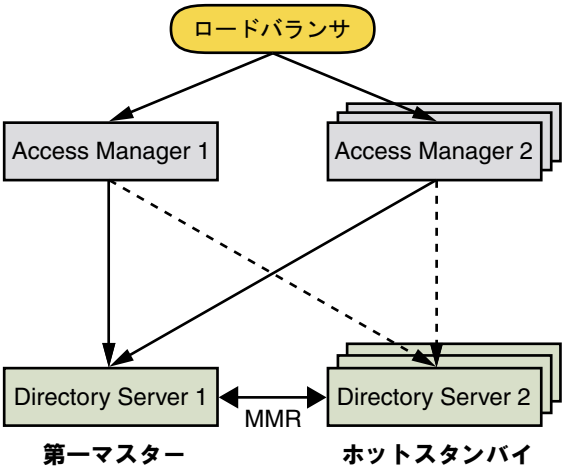
配備のアーキテクチャーによっては、高可用性とスケーラビリティーを実現するために、Access Manager が複数のコンピュータシステムに配備される場合があります。

通常は、サービスを中断させずに連続的に複数の Access Manager インスタンスをアップグレードするのが理想的です。ここでは、Release 3 の Access Manager から Release 5 への、このような順次アップグレードを実行するための手順について説明します。

注	現在のリリースでは、同一ホストシステムにインストールされている Access Manager の複数インスタンスのアップグレードはサポートされていません。同一ホストに複数のインスタンスが存在する場合は、メインインスタンスをアップグレードしてから追加インスタンスを再作成する必要があります。

次の図に示されている配備アーキテクチャーは、順次アップグレードの手順を説明するために使用されます。

図 14-2　複数の Access Manager インスタンスに対する配備アーキテクチャーの例



このアーキテクチャーでは、複数の Access Manager インスタンスがロードバランサを介してアクセスされ、これらのインスタンスは順に、マルチマスターレプリケーション (MMR) 用に設定されたディレクトリにアクセスします。ほかの Directory Server レプリケーションスキーマも使用できますが、MMR は可用性が高くてスケーラブルな、代表的なディレクトリサービスです。図 14-2 では、アップグレード手順を説明しやすくするために、Access Manager と Directory Server の複数のインスタンスがグループ化されています。たとえば、Access Manager 2 は、Access Manager の 2 〜 n 番目のインスタンスを表しています。

Release 3 の Access Manager から Release 5 への順次アップグレードを実行する手順は、Release 5 の Access Manager が Release 3 のディレクトリ構造と共存できないという制約に基づいています。ただし、図 14-2 のように Directory Server インスタンスがレプリケートされている場合は、次の手順を使用して順次アップグレードを実行できます。

1. すべての Access Manager インスタンスに関する Release 3 の設定情報をバックアップします。

   260 ページの表 14-3 を参照してください。

2. `Access Manager1` の設定を変更します。

   a. `Directory Server 1` ではなく `Directory Server 2` をポイントするように、`Access Manager 1` を設定します。

   b. `Access Manager1` を再起動します。

   `Access Manager 2` 〜 `Access Manager n` がアップグレードされている間は、`Access Manager 1` が引き続き要求を処理します。

3. Access Manager 2 ～ Access Manager n をアップグレードします。

   簡潔にするために、以降の手順では「Access Manager 2」は Access Manager 2 ～ Access Manager n を意味します。

   a. ロードバランサで Access Manager 2 を無効にします。

      これで、要求は Access Manager 2 に経路指定されなくなります。

   b. Access Manager2 を部分的にアップグレードします。

      270 ページの「Release 4 の Access Manager のアップグレード」の説明に従って Access Manager の各インスタンスをアップグレードします。ただし、ディレクトリ構造とスキーマの更新 (278 ページの手順 7) は除きます。

   c. Directory Server MMR を無効にします。

   d. Directory Server1 のディレクトリ構造とスキーマを更新します。

      278 ページの手順 7 の説明に従って、amupgrade スクリプトを使用します。Directory Server 2 のスキーマが更新されていないため、Access Manager 1 は引き続き機能します。

   e. Access Manager 2 を再起動します。

   f. ロードバランサで Access Manager 2 を有効にします。

      要求は、ふたたび Access Manager 2 に経路指定されるようになります。

4. Access Manager1 をアップグレードします。

   a. ロードバランサで Access Manager 1 を無効にします。

      これで、要求は Access Manager 1 に経路指定されなくなります。

   b. Access Manager1 を部分的にアップグレードします。

      270 ページの「Release 4 の Access Manager のアップグレード」の説明に従って Access Manager をアップグレードします。ただし、ディレクトリ構造とスキーマの更新 (278 ページの手順 7) は除きます。

   c. Directory Server MMR を有効にします。

      これで、Directory Server 2 のスキーマ (およびデータ) が更新されます。

   d. Directory Server 1 をポイントするように、Access Manager 1 の設定を復元します。

   e. Access Manager1 を再起動します。

   f. ロードバランサで Access Manager 1 を有効にします。

   要求はふたたび Access Manager 1 に、またアップグレードされたほかのすべての Access Manager インスタンスに、経路指定されるようになります。

Java ES Release 2 からの Access Manager のアップグレード

Java ES 2004Q2 (Release 2) の Access Manager を Release 5 にアップグレードする手順は、次の各節に示すように、多少の違いはありますが、Release 4 の Access Manager を Release 5 にアップグレードする手順とほぼ同じです。

- アップグレード前の作業
- Release 2 の Access Manager のアップグレード
- 複数インスタンスのアップグレード

同様に、Java ES 2004Q2 (Release 2) の Access Manager SDK を Release 5 にアップグレードする手順は、同様の例外を除き、Release 4 の Access Manager SDK を Release 5 にアップグレードする手順 (284 ページの「Release 4 の Access Manager SDK のみのアップグレード」を参照) と同じです。Access Manager SDK のみのアップグレードでは、ローカライズ版、Access Manager 管理ツールの JSP のカスタマイズ、およびディレクトリスキーマの移行に関連する手順は除外されます。

Release 2 の Access Manager SDK と、その SDK がリモートインタフェースとして機能する対象である完全な Release 2 の Access Manager の両方を Release 5 にアップグレードしてください。Release 2 と Release 5 のコンポーネントの混合はサポートされていません。そのため、すべてのコンピュータ上にある Release 2 の Access Manager と Release 2 の Access Manager SDK のすべてのインスタンスを Release 5 にアップグレードしてください。

注 Linux プラットフォームで Release 2 の Access Manager からアップグレードする場合は、二重のアップグレード、つまり、Access Manager とオペレーティングシステムの両方をアップグレードする必要があります (Release 5 の Access Manager は RHEL 2.1 でサポートされていない)。詳細については、264 ページの「二重のアップグレード」を参照してください。

アップグレード前の作業

Access Manager をアップグレードする前に、266 ページの「アップグレード前の作業」で説明されている手順を実行します。ただし、次の例外および追加があります。

- Access Manager が依存するコンポーネントのアップグレード
- ディレクトリスキーマをアップグレードする
- ディレクトリのインデックスを再生成する

Access Manager が依存するコンポーネントのアップグレード

Release 4 からのアップグレードとは異なり、Release 2 から Release 5 へのアップグレード前の作業には、すべての共有コンポーネント (48 ページの表 1-9 を参照) と、Access Manager が依存するローカルにあるすべての製品コンポーネントの Release 5 へのアップグレードが必要です。

Access Manager の依存関係をアップグレードする場合は、Access Manager をアップグレードする前に、そのすべてを次の順序でアップグレードするようにしてください。アップグレード済みのコンポーネントについては、その手順を省略できます。

1. 共有コンポーネント : Java ES 共有コンポーネントを Release 5 に同期する手順は、67 ページの第 2 章「Java ES 共有コンポーネントのアップグレード」で説明されています。ただし、Release 5 の Access Manager の新規インストールを実行すると、Java ES 共有コンポーネントはインストーラによって自動的にアップグレードされます。
2. Directory Server: Directory Server が Access Manager と同じコンピュータ上に存在することはほとんどありませんが、Directory Server を Release 5 にアップグレードする手順は、117 ページの「Java ES Release 2 からの Directory Server のアップグレード」で説明されています。
3. Web コンテナソフトウェア : Web Server または Application Server をアップグレードする手順は、それぞれ、155 ページの「Java ES Release 3 からの Web Server のアップグレード」および217 ページの「Java ES Release 2 からの Application Server のアップグレード」で説明されています。

ディレクトリスキーマをアップグレードする

Messaging Server や Calendar Server などの Communication Suite コンポーネントをサポートするために Sun Java Communications Suite の Directory Preparation Tool (`comm_dssetup.pl`) を使用して Directory Server を設定した場合、まず Directory Preparation Tool 6.4 を使用してディレクトリスキーマをアップグレードしてから

Access Manager をアップグレードしてください。『Sun Java Communications Suite 5 アップグレードガイド』(http://docs.sun.com/doc/820-0507?l=ja) を参照してください。このアップグレード前作業は、Access Manager が依存するコンポーネントをアップグレードしたあとに実行します。

ディレクトリのインデックスを再生成する

ディレクトリスキーマをアップグレードしたあとに Access Manager のアップグレードを実行する場合の問題 (上記の「ディレクトリスキーマをアップグレードする」を参照) を回避するには、次に示すように、Access Manager ディレクトリルートサフィックスのインデックスを手動で再生成する必要があります。

Release 2 ～ Release 4 の Directory Server:

1. `cd` *serverRoot*`/slapd-`\`hostname\`
2. `./db2index.pl -D "cn=directory manager" -w` *passwordFile* `-n` *databaseName*

   ここで、デフォルトの *databaseName* は `userRoot` です。

Release 5 の Directory Server:

1. `cd` *DirServer-base*`/ds6/bin`
2. `./dsconf reindex -D "cn=Directory Manager" -e -w` *passwordFile suffix*

   それぞれの意味は次のとおりです。

   `-e` は、セキュリティー保護されていない接続を示します。

   `-D` の後ろには Directory Manager を指定します。

   `-w` の後ろには、パスワード文字列を含んだファイルを指定します。

   suffix は、Access Manager ディレクトリルートサフィックスです。

ディレクトリ内のエントリの数によっては、インデックスの再生成が完了するまでに膨大な時間がかかることがあります。

Release 2 の Access Manager のアップグレード

Access Manager を Release 2 から Release 5 にアップグレードする手順は、Access Manager ソフトウェアを配備している Web コンテナによって異なります。

Release 2 の Access Manager のアップグレード : Web Server Web コンテナ

Release 2 の Access Manager を Release 5 にアップグレードするには、「Release 4」を「Release 2」と読み替えた上で、Web Server Web コンテナへの配備時に、270 ページの「Release 4 の Access Manager のアップグレード」の手順を使用します。

Release 2 の Access Manager のアップグレード: Application Server Web コンテナ

配備先 Web コンテナが Application Server のとき、Release 2 の Access Manager の Release 5 へのアップグレードには、次の 2 つのケースがあります。

- **Release 5 Application Server が新規インストールされている場合** : この場合、Release 2 Access Manager を Release 5 にアップグレードするには、「Release 4」を「Release 2」と読み替えた上で、270 ページの「Release 4 の Access Manager のアップグレード」の手順を使用してください。
- **Release 2 の Application Server が Release 5 にアップグレードされている場合** : この場合、Application Server が最初に配備されたときの Release 2 の Application Server インスタンス (*instanceName*) は、Release 5 にアップグレードされると、アップグレード処理によって作成されたノードエージェントの下へ移行されています。このアップグレードされた Application Server インスタンス内の Access Manager のアップグレードには、次の節以降の手順が必要です。

アップグレードの概要

Access Manager をアップグレードするための手順は、次の手順で構成されます。

1. Access Manager モバイルアクセスソフトウェアをアップグレードします。
2. Java ES Release 2 バージョンの Access Manager を削除します。`ampre71upgrade` スクリプトを使用します。
3. Release 5 へのアップグレードをローカライズする必要がある場合は、Release 2 のローカライズ版パッケージを削除します。この手順は、手動で実行してください。
4. Java ES Release 5 バージョンの Access Manager をインストールします。Java ES インストーラを使用し、「あとで設定」オプションを指定します。
5. Access Manager 用 JSP を再カスタマイズします。
6. Directory Server が稼働していることを確認します。

7. 次の Application Server インスタンスを起動します。ドメイン管理サーバー (DAS)、ノードエージェント、および Access Manager が配備されているサーバーインスタンス。
8. Access Manager の配備を取り消し、再設定し、Application Server インスタンスに再配備します。amconfig スクリプトを使用します。
9. Access Manager の classpath-suffix および server-classpath の情報が Release 5 の Application Server の domain.xml ファイルに移行されていることを確認します。
10. ドメイン管理サーバー (DAS)、ノードエージェント、およびサーバーインスタンスを停止します。
11. ドメイン管理サーバー (DAS)、ノードエージェント、およびサーバーインスタンスを再起動します。
12. ディレクトリ構造とスキーマを更新します。amupgrade スクリプトを使用します。

以降の手順中で、これらの手順について説明します。

アップグレード手順

1. Access Manager モバイルアクセスソフトウェアをアップグレードします。

   Release 2 バージョンにパッチを適用することによって、Access Manager モバイルアクセスソフトウェアをアップグレードする必要があります。必要なパッチを次の表に示します。

表 14-13 Access Manager モバイルアクセスソフトウェアをアップグレードするためのパッチ[1]

説明	パッチ ID: Solaris 9 および 10	パッチ ID: Linux
Mobile Access ソフトウェア	119530-05 (SPARC) 119531-05 (x86)	119532-05 • sun-identity-mobileaccess-6.2-25.3.i386.rpm • sun-identity-mobileaccess-config-6.2-25.3.i386.rpm

1. この表に示されているパッチのリビジョン番号は、Java ES Release 5 にアップグレードするための最低要件です。表に記載されているものよりも新しいリビジョンのパッチが使用可能になった場合は、新しいほうのパッチを使用してください。

   a. 表 14-6 のパッチ番号を参照して、必要なパッチを入手します。

      パッチは http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access から /tmp にダウンロードします。

b. パッチの README ファイルに示されているパッチ適用前の手順をすべて実行します。

c. パッチによって要求される、次のパラメータの値を入手します。

表 14-14　モバイルアクセスのパッチパラメータ

パラメータ	値
Directory Manager DN	デフォルト : cn=Directory Manager
ディレクトリマネージャーのパスワード	

d. 表 14-6 のパッチを適用します。

Solaris の場合 :
patchadd *patch_ID*

Linux の場合 :
./update

パッチの README ファイルに示されているパッチ適用後の手順をすべて実行します。

2. Java ES Release 2 バージョンの Access Manager を削除します。

a. Release 4 の Access Manager がインストールされたコンピュータに root としてログインするか、スーパーユーザーになります。

```
su -
```

b. Java ES Release 5 配布パッケージ内の *os_arch*/Product/identity_svr/Tools ディレクトリに移動します。ここで、*os_arch* はプラットフォーム (Solaris_sparc など) に対応します。

c. ampre71upgrade スクリプトによって要求される、次のパラメータの値を入手します。

表 14-15　Access Manager 設定パラメータ : ampre71upgrade

パラメータ	値
Directory Server ホスト	完全修飾名を設定する : *hostname.domain*
Directory Server ポート	非 SSL ポート番号を指定する[1] デフォルト : 389
最上位管理者の DN	デフォルト : uid=amadmin,ou=People,*default_org_DN*

表 14-15　Access Manager 設定パラメータ : ampre71upgrade (続き)

パラメータ	値
最上位管理者のパスワード	
バックアップファイルを格納するディレクトリ	デフォルト : *AccessManager-base*

1. SSL のデフォルト値である 636 とは異なる Directory Server SSL ポートを指定してください。

d. Directory Server が稼働していることを確認し、稼働していない場合は起動します。

e. ampre71upgrade スクリプトを実行します。

```
./ampre71upgrade
```

このスクリプトは Access Manager の設定ファイルをバックアップし、Release 4 のベースパッケージを削除します。ローカライズ版のパッケージは、下記の手順 3 に従って、手動で削除する必要があります。

3. Release 5 へのアップグレードをローカライズする必要がある場合は、Release 2 のローカライズ版パッケージを削除します。

上記の手順 2 で実行した ampre71upgrade スクリプトではローカライズ版パッケージが削除されないため、次のように、手動で削除してください。

Solaris の場合 :

a. ローカライズ版パッケージを確認します。

```
pkginfo | grep SUNWaml
pkginfo | grep SUNWamclnt
pkginfo | grep SUNWamdistauth
```

b. 上記の手順 a でローカライズ版パッケージが見つかった場合は、削除します。

```
pkgrm SUNWaml<Locale>
pkgrm SUNWamclnt<Locale>
pkgrm SUNWamdistauth<Locale>
```

Linux の場合:

a. ローカライズ版 RPM を確認します。

```
rpm -qa | grep sun-identity-sdk-*
rpm -qa | grep sun-identity-clientsdk-*
rpm -qa | grep sun-identity-distauth-*
```

b. 上記の手順 a でローカライズ版 RPM が見つかった場合は、削除します。

```
rpm -e sun-identity-sdk-Locale-*
rpm -e sun-identity-clientsdk-Locale-*
rpm -e sun-identity-distauth-Locale-*
```

4. Java ES Release 5 バージョンの Access Manager をインストールします。

次の手順を実行します。

a. Release 2 の Access Manager がホストされたコンピュータ上で Java ES インストーラを起動します。

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

ここで、*os_arch* は、`Solaris_sparc` などのプラットフォームに対応します。コマンド行インタフェースには、`installer -nodisplay` オプションを使用します。

ようこそページと、ライセンス使用許諾契約ページが表示されてから、コンポーネント選択ページが表示されます。Java ES インストーラが直接アップグレードできるインストール済みのコンポーネントが検出された場合、それらのコンポーネントは「アップグレード可」の状態で表示されます。

b. コンポーネント選択ページから Access Manager を選択します。

c. Release 2 がインストールされていたのと同じインストールディレクトリを指定します。

d. 「あとで設定」オプションを選択します。

e. 必要に応じて、ローカライズ版のパッケージをインストールするオプションを選択します。

f. インストールが完了したら、Java ES インストーラを終了します。

5. Access Manager 用 JSP を再カスタマイズします。

268 ページの「Web コンテナのカスタマイズファイルのバックアップ」で保存した、Access Manager コンソールおよび認証ユーザーインタフェース (UI) 用 JSP に、Release 2 のカスタマイズを再適用します。

続いて、カスタマイズ JSP ファイルを次の適切なディレクトリにコピーします。

- 旧バージョンのみの Access Manager コンソール *AccessManager-base*/web-src/applications/console
- 認証 UI: *AccessManager-base*/web-src/services/config/auth/default または *AccessManager-base*/web-src/services/config/auth/default_*Locale* (*Locale* は ja などのロケールインジケータ)

詳細については、『Sun Java System Access Manager 7.1 Developer's Guide』(http://docs.sun.com/doc/819-4675) を参照してください。

6. Directory Server が稼働していることを確認します。
7. 次の Application Server インスタンスを起動します。

   以降のコマンドおよび以降の手順では、次の規則が使用されています。

   - *nodeagentName* の形式は *hostName_domainName* ですが、デフォルトでは単に *hostName* です。
   - デフォルトの *domainName* は domain1 です。
   - デフォルトの *instanceName* は server1 です。

   注 サーバーインスタンスを起動する前に、下に示すように、startinstances=false オプションを使用してノードエージェントを別々に起動する必要があります。

   a. ドメイン管理サーバー (DAS) を起動します。

      AppServer8-base/bin/asadmin start-domain --user *admin_ID domainName*

   b. アップグレードされた Application Server インスタンスの移行先ノードエージェントを起動します。

      AppServer8-base/bin/asadmin start-node-agent --startinstances=false --user *admin_ID nodeagentName*

   c. Access Manager が配備されているサーバーインスタンス (*instanceName*) がまだ稼働していない場合は、起動します。

      AppServer8-base/bin/asadmin start-instance --user *admin_ID instanceName*

8. Access Manager の配備を取り消し、再設定し、Application Server インスタンスに再配備します。

   a. Web コンテナが SSL モードで実行されている場合は、コンテナの SSL 証明書の期限が切れていなくて、まだ有効であることを確認してください。

b. amsamplesilent テンプレート入力ファイルを基に、amconfig 入力ファイルを作成します。

```
cp amsamplesilent config-file
```

以降の手順では、*config-file* が amsamplesilent と同じディレクトリに存在することを前提にしています。

c. *config-file* 内の設定パラメータを設定します。

すべてのパラメータを正確に設定する必要があります。次の表に示すように、設定値には、AMConfig.properties ファイルから移行できるものと、アップグレード手順の中で新しく設定するものがあります。

表 14-16 Access Manager 設定パラメータ : amconfig

パラメータ	値
アップグレードパラメータ	
DEPLOY_LEVEL	配備取り消しの場合は 26 を設定、または再設定と配備の場合は 1 に設定
DIRECTORY_MODE	5 に設定
AM_REALM[1]	旧バージョンモードを有効にする場合は disabled に設定 レルムモードを有効にする場合は enabled に設定 デフォルト : enabled
JAVA_HOME	JDK Release 5 ディレクトリに設定 : /usr/java/jdk1.5.0_04/
WEB_CONTAINER	Application Server 8.*x* *の場合は* AS8 に設定 *config-file* の対応するセクションのみを設定する
AS81_INSTANCE (Application Server 7.*x* からアップグレードした Application Server 8.*x* を Web コンテナとして使用している)	Application Server 7.*x* *instanceName* に設定。デフォルトでは server1
AS81_INSTANCE_DIR (Application Server 8.*x* を Web コンテナとして使用している)	インスタンスの Application Server 8.*x* ドメインディレクトリに設定。デフォルトでは次の場所にある : *AppServer8Config-base*/domains/domain1

表 14-16　Access Manager 設定パラメータ : amconfig (続き)

パラメータ	値
AS81_DOCS_DIR (Application Server 8.*x* を Web コンテナとして使用している)	インスタンスの Application Server 8.*x* docroot ディレクトリに設定。デフォルトでは次の場所にある : *AppServer8Config-base*/domains/domain1/docroot
AMConfig.properties から移行される値	
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. レルムモードと旧バージョンモードの詳細については、262 ページの「互換性の問題」を参照してください。
2. このパラメータの値は、以前のバージョンの Access Manager と同様にしてください。
3. Access Manager と Access Manager SDK の両方が配備されている場合、このプロパティーの値は、Access Manager インスタンスとそれに関連付けられた Access Manager SDK インスタンスの両方で同じにしてください。

その他のパラメータについては、Web コンテナまたはパスワードを変更する場合を除いて、アップグレードする Release 2 の設定で使われていたのと同じ値を指定します。

d. amconfig を実行して、Access Manager の配備を取り消します。

config-file 内の DEPLOY_LEVEL の値を 26 に設定します。

```
cd /AccessManager-base/bin
./amconfig -s AccessManager-base/bin/config-file
```

e. 共通エージェントコンテナが実行されていることを確認してください。

```
netstat -an | grep 11163
```

実行されていない場合は、起動します。

```
/usr/sbin/cacaoadm start
```

f. amconfig を実行して Access Manager を再設定し、Web コンテナに配備します。

config-file 内の DEPLOY_LEVEL の値を 1 に設定します。

```
cd /AccessManager-base/bin
./amconfig -s AccessManager-base/bin/config-file
```

9. Access Manager の classpath-suffix および server-classpath の情報が Release 5 の Application Server の domain.xml ファイルに移行されていることを確認します。

a. Access Manager が当初配備されていた Release 2 の Application Server インスタンスの server.xml ファイルから、Access Manager の classpath-suffix および server-classpath の情報をメモします。

AppServer7Config-base/domains/*domainName*/*instanceName*/config/server.xml

b. classpath-suffix および server-classpath エントリが、Access Manager が配備されているアップグレードされた Application Server インスタンスの domain.xml ファイルに付加されていることを確認します。

AppServer8Config-base/nodeagents/*nodeagentName*/*instanceName*/config/domain.xml

クラスパス情報は、Release 5 の Application Server の domain.xml ファイルの、*instanceName*-config ブロックに追加することが推奨されます。このブロックは次の行で始まります。

```
<config dynamic-reconfiguration-enabled="true"
name="instanceName-config">
```

10. ディレクトリ構造とスキーマを更新します。

Release 5 の Access Manager は Release 4 のディレクトリ構造と共存しますが、Release 5 の機能をサポートするためには構造を変更する必要があります。amupgrade スクリプトを実行して、Access Manager のディレクトリ構造とスキーマを Release 5 に更新します。このスクリプトは次のディレクトリにインストールされています。

- *Solaris の場合:*
  AccessManager-base/upgrade/scripts
- *Linux の場合:*
  AccessManager_base/identity/upgrade/scripts

a. amupgrade スクリプトによって要求される、次のパラメータの値を入手します。

表 14-17　Access Manager 設定パラメータ : amupgrade

パラメータ	値
Directory Server ホスト	完全修飾名を設定する : *hostname.domain*
Directory Server ポート	非 SSL ポート番号を指定する[1] デフォルト : 389
Directory Manager DN	デフォルト : cn=Directory Manager
Directory Manager パスワード	
最上位管理者の DN	デフォルト : uid=amadmin,ou=People,*default_org_DN*
最上位管理者のパスワード	
Enable Realm Mode Release 4 のレルムモードからアップグレードする場合、このパラメータ値は要求されません。	Y/N: Yes は、レルムモードが有効であり、サービスデータが新しいレルムツリーに移行されたことを意味します。No (デフォルト) はサービスデータが旧バージョンモードのままであることを意味します。

1. Directory Server の SSL ポートを SSL のデフォルト値である 636 などに指定すると、アップグレードプロセスは正しく完了しません。

b. amupgrade スクリプトを実行します。

```
cd AccessManager-base/upgrade/scripts
./amupgrade
```

スクリプトによるアップグレードが成功すると、「Upgrade completed」というメッセージが表示されます。

c. 次のアップグレードログファイルをチェックし、ディレクトリスキーマ拡張についての情報を確認します。

Solaris の場合 :

```
/var/sadm/install/logs/
        Sun_Java_System_Access_Manager_upgrade_dit_log.mmddhhmm
```

Linux の場合 :

```
/var/log/Sun_Java_System_Access_Manager_upgrade_dit_log.mmddhhmm
```

11. ドメイン管理サーバー (DAS)、ノードエージェント、およびサーバーインスタンスを停止します。

    これらは 298 ページの手順 7 で起動されたインスタンスです。

    AppServer8-base/bin/asadmin stop-domain --user *admin_IDdomainName*

    AppServer8-base/bin/asadmin stop-node-agent --user *admin_ID*
    nodeagentName

12. ドメイン管理サーバー (DAS)、ノードエージェント、およびサーバーインスタンスを再起動します。

    注 サーバーインスタンスを起動する前に、下に示すように、startinstances=false オプションを使用してノードエージェントを別々に起動する必要があります。

    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 のデフォルト値は 4848 です。

Access Manager のアップグレードの確認

アップグレード手順が完了したら、280 ページの「Access Manager のアップグレードの確認」の説明に従って、アップグレードが成功したかどうかを確認します。

アップグレード後の作業

SAML (Security Assertion Markup Language) サービスを使用する場合、Access Manager コンソールを使って SAML 認証モジュールを追加および有効化する必要があります。SAML 認証モジュールインスタンスの作成の詳細については、『Sun Java System Access Manager7.1 管理ガイド』(http://docs.sun.com/doc/820-0842?l=ja) を参照してください。

アップグレードのロールバック

Access Manager をアップグレード前の状態にロールバックするためのスクリプトは用意されていません。ロールバックプロセスは、アップグレード前の作業でバックアップした Access Manager データ (269 ページの「Release 4 の Access Manager のログファイルとデバッグファイルのバックアップ」を参照) を使用して手動で行う必要があります。ロールバックは非常に難しく、実際的ではありません。

複数インスタンスのアップグレード

配備のアーキテクチャーによっては、高可用性とスケーラビリティーを実現するために、Access Manager が複数のコンピュータシステムに配備される場合があります。

通常は、サービスを中断させずに連続的に Access Manager インスタンスをアップグレードするのが理想的です。Release 2 の Access Manager から Release 5 への順次アップグレードを実行する手順は、Release 5 の Access Manager が Release 2 のディレクトリ構造と共存できないという制約に基づいています。ただし、図 14-2 のように Directory Server インスタンスがレプリケートされている場合は、287 ページの「複数インスタンスのアップグレード」の説明に従って順次アップグレードを実行できます。

第 15 章

Portal Server

この章では、Portal Server を Java ES 5 (Release 5): Sun Java System Portal Server 7.1 にアップグレードする方法について説明します。

ここでは、Release 5 でサポートされている各種のアップグレードパスでのアップグレードの注意点の概要について説明します。また、Solaris オペレーティングシステムと Linux オペレーティングシステムの両方でのアップグレードを取り上げます。

- 306 ページの「Portal Server のアップグレードの概要」
- 313 ページの「Java ES Release 4 からの Portal Server のアップグレード」
- 340 ページの「Java ES Release 3 からの Portal Server のアップグレード」
- 343 ページの「Java ES Release 2 からの Portal Server のアップグレード」
- 352 ページの「Portal Server の IFR 7.0 からのアップグレード」

注 この章では、*PortalServer6-base* と *PortalServer6Config-base* (Portal Server 6.*x* の場合)、および *PortalServer7-base* と *PortalServer7Config-base* (Portal Server 7.*x* の場合) というディレクトリパスを基準に、ファイルの場所が指定されています。これらのパスの少なくとも一部は、Portal Server が最初にインストールされたときにインストールディレクトリとして指定されている可能性があります。指定されていない場合、Java ES インストーラによってデフォルト値が割り当てられています。

次の表に、これらのディレクトリパスのデフォルト値を示します。

表 15-1 Portal Server ディレクトリパス

パス名変数	Solaris OS	Linux OS
PortalServer6-*base*	`/opt/SUNWps`	`/opt/sun/portal`
PortalServer6Config-*base*	`/etc/opt/SUNWps`	`/etc/opt/sun/portal`

表 15-1　Portal Server ディレクトリパス (続き)

パス名変数	Solaris OS	Linux OS
PortalServer7-*base*	/opt/SUNWportal	/opt/sun/portal
PortalServer7Config-*base*	/etc/opt/SUNWportal	/etc/opt/sun/portal
PortalServer7Data-*base*	/var/opt/SUNWportal	/var/opt/sun/portal

Portal Server のアップグレードの概要

ここでは、Java ES 5 (Release 5) へのアップグレードに影響する Portal Server の一般的な側面について説明します。

- Java ES Release 5 の Portal Server について
- Portal Server のアップグレードロードマップ
- Portal Server のデータ
- Portal Server のアップグレード方針

Java ES Release 5 の Portal Server について

Java ES Release 5 の Portal Server は Release 4 に対するメジャーリリースであり、多くの拡張機能や新機能が追加されています。これらの変更の多くは、Release 4 以降の暫定機能リリース (Interim Feature Releases、IFR) で加えられました。Release 5 では、IFR に対する若干の機能変更だけが行われています。IFR の拡張機能および新機能については、『Sun Java System Portal Server 7.1 リリースノート (UNIX 版)』(http://docs.sun.com/app/docs/doc/820-0900/6nck07u85?l=ja&a=view) を参照してください。特に、Release 4 のコマンド行管理インタフェースは psadmin コマンドに置き換えられています。

Portal Server のアップグレードロードマップ

表 15-2 は、Portal Server を Java ES Release 5 へアップグレードする際の、サポートされるアップグレードパスを示しています。この表は Solaris および Linux の両方のオペレーティングシステムに適用されます。

表 15-2　Java ES 5 (Release 5): Portal Server 7.1 へのアップグレードパス

Java ES のリリース	Portal Server バージョン	一般的な手順	必要な再設定
暫定機能リリース (IFR)	Sun Java System Portal Server IFR 7.0 2005Q4	直接アップグレード: パッチを適用してアップグレードスクリプトを使用することにより、実行されます。	カスタマイズを手動で再度適用する必要があります。
Release 4	Sun Java System Portal Server 6.3.1 2005Q4	直接アップグレード: アップグレードスクリプトを使用して行います。	カスタマイズを手動で再度適用する必要があります。
Release 3	Sun Java System Portal Server 6.3.1 2005Q1	直接アップグレード: アップグレードスクリプトを使用して行います。	カスタマイズを手動で再度適用する必要があります。
Release 2	Sun Java System Portal Server 6.3 2004Q2	直接アップグレード: アップグレードスクリプトを使用して行います。	カスタマイズを手動で再度適用する必要があります。
Release 1	Sun ONE Portal Server 6.2 (2003Q4)	直接アップグレード不可: ただし、まず Release 3 にアップグレードし、次に Release 3 から Release 5 にアップグレードすることにより実行できます。	設定データ
それ以前の Java ES リリース		直接アップグレード不可。	

Portal Server のデータ

次の表は、Portal Server ソフトウェアのアップグレードにより影響を受ける可能性のあるデータの種類を示しています。

表 15-3　Portal Server のデータの使用法

データの種類	場所	使用法
設定データ	*PortalServer6Config-base*/	Portal Server の設定

表 15-3　Portal Server のデータの使用法 (続き)

データの種類	場所	使用法
Web コンテナのアクセス制御および設定ファイル	Web Server 7.0 (Java ES Release 5) 次の場所にある server.policy および server.xml ファイル *WebServer7Config-base*/https-*configName*/config Web Server 6.*x* (Java ES Release 2、3、および 4) 次の場所にある server.policy および server.xml ファイル *WebServer6-base*/https-*hostname*/config Application Server 8.*x* (Java ES Release 3、4、および 5): 次の場所にある server.policy および domain.xml ファイル *AppServer8Config-base*/domains/*domainName*/config Application Server 7.*x* (Java ES Release 2): 次の場所にある server.policy および server.xml ファイル *AppServer7Config-base*/domains/*domainName*/config	Portal Server Web コンテナインスタンスの設定
カスタマイズデータ	*PortalServer6Config-base*/desktop	カスタマイズしたモジュールの JAR ファイル カスタマイズされたサンプル Portal Server デスクトップ
ディレクトリスキーマ サービスの設定 ユーザーデータ	Directory Server	Portal Server は、ポータルデスクトップなどのサービスの設定、およびディレクトリに格納されているユーザープロファイルデータに依存します。
動的なアプリケーションデータ	なし	Portal Server は、セッション状態などのアプリケーションデータを永続的に格納しません。

Portal Server のアップグレード方針

Portal Server をアップグレードするための方針は一般に、第 1 章「アップグレードの計画」で説明されている多くの考慮事項 (アップグレードパス、Java ES コンポーネント間の依存関係、選択的なアップグレードかすべてのアップグレードか、複数インスタンスの配備など) によって異なります。

ここでは、Portal Server のアップグレード計画に影響する可能性のある問題を取り上げ、Portal Server の全般的な説明を詳細に記述します。

互換性の問題

Release 5 の Portal Server では、Portal Server および Portal Server Secure Remote Access コンポーネントの管理に使用される `psadmin` コマンドに、公開インタフェースの変更が導入されました。『Sun Java System Portal Server 7.1 Command-Line Reference』(`http://docs.sun.com/doc/819-5030`) を参照してください。

そのため、Release 5 の Portal Server は、複数インスタンスの配備に順次アップグレードを実行している移行期間を除き、以前のバージョン、または Portal Server Secure Remote Access コンポーネント (SRA ゲートウェイ、Rewriter プロキシ、および Netlet プロキシを含む) の以前のバージョンとは下位互換性がありません。すべての Portal Server インスタンスを、Portal Server Secure Remote Access コンポーネントインスタンスも含めて Java ES Release 5 に合わせる必要があります。

また、個々の Portal Server コンポーネント (モバイルアクセスコンポーネントを含む) も以前のバージョンとは下位互換性がなく、すべてを Java ES Release 5 に同期する必要があります。

さらに、Release 5 の Portal Server で使用される Directory Server データ構造と以前のバージョンの Portal Server で使用される Directory Server データ構造の間にも互換性がありません。この非互換性は、同じ Directory Server データを使用した複数の Portal Server インスタンスの順次アップグレードに影響を与えます。

Portal Server の依存関係

ほかの Java ES コンポーネントに対する Portal Server の依存関係が、Portal Server ソフトウェアのアップグレードおよび再設定の手順に影響を及ぼす可能性があります。たとえば、Portal Server のインタフェースまたは関数が変更された場合は、Portal Server が依存するコンポーネントのアップグレードされたバージョンが必要になる場合があります。このようなコンポーネントのアップグレードの必要性は、個々のアップグレードパスによって異なります。

Portal Server は、次の Java ES コンポーネントに依存します。

- **共有コンポーネント** : Portal Server は、特定の Java ES 共有コンポーネントに依存します (48 ページの表 1-9 を参照)。
- **Web コンテナ** : Portal Server は、Java ES Web Server または Java ES Application Server のどちらか、あるいは Weblogic や WebSphere のサードパーティー製 Web コンテナで提供される Web コンテナサービスに必ず依存しています。

注	Portal Server の Release 5 へのアップグレードは、サードパーティー製 Web コンテナでの配備に対してはサポートされていません。Weblogic や WebSphere の Web コンテナでの配備に対しては、Release 5 の Portal Server の新規インストールを実行してください。

- Access Manager (または Access Manager SDK): Portal Server は、シングルサインオンを含む、エンドユーザーへの認証および承認サービスを提供するために Access Manager に必ず依存しています。Access Manager をリモートコンピュータ上で動作させる場合、Access Manager SDK がローカルで利用可能である必要があります。
- Directory Server: Portal Server は、Access Manager を介してアクセスされるユーザーデータを格納する Directory Server に必ず依存しています。そのため、Portal Server のアップグレードによって、ディレクトリスキーマの拡張が必要になる場合があります。
- Portal Server Secure Remote Access: Portal Server は、ゲートウェイ、Rewriter プロキシ、および Netlet プロキシコンポーネントによってセキュリティー保護されたリモートアクセスを提供する Portal Server Secure Remote Access に対する任意の依存関係を持っています。
- Java DB: Portal Server は、いくつかのポートレットアプリケーションにサポートを提供する Java DB に対する任意の依存関係を持っています。
- Service Registry: Portal Server は、コンパイルに必要なライブラリを提供する Service Registry に必ず依存しています。
- Communications Express: Portal Server は、エンドユーザーにメッセージングとカレンダのチャネルを提供するために使用される、Sun Java Communications Suite コンポーネントである Communications Express に対する任意の依存関係を持っています。Communications Express は、Java ES 製品コンポーネントではなくなりました。

選択的なアップグレードに関する問題

一般に、Java ES Release 5 はコンピュータ上のすべてのコンポーネントの選択的なアップグレードをサポートしていますが、Portal Server がほかの非常に多くの Java ES コンポーネントに依存しているために、さまざまな Java ES リリースバージョンにわたるコンポーネントの任意の組み合わせを保証することはきわめて困難です。

このため、Portal Server は、Access Manager および Web コンテナに対して、限られたアップグレードシナリオをサポートしています。

- **Portal Server を Java ES Release 4 からアップグレードする場合** : Portal Server をアップグレードする前に Directory Server、Access Manager、および Web コンテナ (Web Server または Application Server) を Release 5 にアップグレードします。または、ほかのコンポーネントを Release 4 レベルのままにして Portal Server のみを Release 5 にアップグレードすることが可能です。ただし、一部の依存関係を Release 4 のままにして、それ以外を Release 5 にアップグレードすることはできません。
- **Portal Server を Java ES Release 3 からアップグレードする場合** : Portal Server をアップグレードする前に Directory Server、Access Manager、および Web コンテナ (Web Server または Application Server) を Release 4 または Release 5 にアップグレードしてください。いずれかの依存関係を Release 3 のままにすることも、一部の依存関係を Release 4 に、それ以外を Release 5 にアップグレードすることもできません。
- **Portal Server を Java ES Release 2 からアップグレードする場合** : Portal Server をアップグレードする前に Directory Server、Access Manager、および Web コンテナ (Web Server または Application Server) を Release 4 または Release 5 にアップグレードしてください。いずれかの依存関係を Release 2 のままにすることも、一部の依存関係を Release 4 に、それ以外を Release 5 にアップグレードすることもできません。

Web コンテナのアップグレードシナリオ

Portal Server は、Web Server または Application Server のどちらかで提供される Web コンテナに配備できます。その結果、配備先の Web コンテナも同時に Release 5 にアップグレードする可能性があるために、Portal Server の Release 5 へのアップグレードが複雑になることがあります。この点では、次の表に示すように、可能性のある Web コンテナのアップグレードシナリオがいくつかあります。

表 15-4 Portal Server のアップグレード用 Web コンテナのアップグレードシナリオ

シナリオ	Portal Server が当初配備されていた Web コンテナ	アップグレード後に Portal Server が配備される Web コンテナ	適用可能な Portal Server のアップグレードパス : アップグレード元
シナリオ 1	Web Server 6.*x*	Web Server 6.*x*	Release 2 Release 3 Release 4 IFR 7.0
シナリオ 2	Web Server 6.*x*	Web Server 7.0	Release 2 Release 3 Release 4

表 15-4　Portal Server のアップグレード用 Web コンテナのアップグレードシナリオ

シナリオ	Portal Server が当初配備されていた Web コンテナ	アップグレード後に Portal Server が配備される Web コンテナ	適用可能な Portal Server のアップグレードパス : アップグレード元
シナリオ 3	Application Server 8.1	Application Server 8.1	Release 3 Release 4 IFR 7.0
シナリオ 4	Application Server 8.1	Application Server 8.2	Release 3 Release 4 IFR 7.0
シナリオ 5	Application Server 7*x*	Application Server 8.2	Release 2

特に Web コンテナのメジャーバージョンのアップグレードがある場合は、表 15-4 内の該当するアップグレードシナリオに適切な値を指定するように、Portal Server をアップグレードする (たとえば、`psupgrade` スクリプトを使用する) ときに注意してください。

二重のアップグレード

44 ページの「二重のアップグレード : Java ES ソフトウェアとオペレーティングシステムソフトウェア」で説明されているように、二重のアップグレードでは Portal Server とオペレーティングシステムの両方をアップグレードします。次のようなインプレースオペレーティングシステムアップグレードを使用して二重のアップグレードを実行できます。

1. 既存の Portal Server データをバックアップします。

   基本的なデータの場所については、307 ページの「Portal Server のデータ」を参照してください。

2. オペレーティングシステムをアップグレードします。

   このアップグレードでは、既存のファイルシステムはそのまま残されます。

3. Release 5 の Portal Server にアップグレードします。

   アップグレードパスに応じて、この章の該当する節を参照してください。

Java ES Release 4 からの Portal Server のアップグレード

ここでは、Portal Server の Java ES 2005Q4 (Release 4) から Java ES 5 (Release 5) へのアップグレードについて説明します。

注	ここでの説明は、Portal Server が Application Server Web コンテナに配備されており、Release 5 にアップグレードされる前に Release 2 から Release 3 または 4 にアップグレードされているという特殊なケースには適用されません。上記のアップグレードパスは、現在サポートされていません。

この節で説明する内容は、次のとおりです。

- はじめに
- Release 4 の Portal Server のアップグレード
- 複数インスタンスのアップグレード

はじめに

Java ES Release 4 Portal Server を Release 5 にアップグレードする場合は、アップグレード処理の次の側面を考慮してください。

- **一般的なアップグレード手順** : このアップグレードは、アップグレードスクリプト `psupgrade` を使用することにより実行されます。このスクリプトは新しいパッケージをインストールし、必要に応じて設定データを移行し、ローカライズ版ファイルを更新し、さらに Portal Server Web アプリケーションを Web コンテナに再配備します。
- **アップグレードの依存関係** : Portal Server は、いくつかの Java ES 共有コンポーネントに依存します (48 ページの表 1-9 を参照)。Release 5 の Portal Server はこれらの共有コンポーネントの Release 4 バージョンと互換性がありますが、Portal Server のアップグレードに使用される `psupgrade` スクリプトには ANT 共有コンポーネントの Release 5 バージョンが必要なため、共有コンポーネントのアップグレードはいずれにしても必要です。

  309 ページの「Portal Server の依存関係」で説明されているように、Release 5 の Portal Server は Web コンテナ、Access Manager、および Directory Server にも依存します。これらの依存関係のアップグレードに対して、次の 2 つの方法がサポートされています (310 ページの「選択的なアップグレードに関する問題」を参照)。

  - すべての依存関係が Release 4 コンポーネントによって満足される (何も Release 5 にアップグレードされない)。
  - すべての依存関係が Release 5 コンポーネントによって満足される (すべてが Release 5 にアップグレードされる)。
- **下位互換性** : Release 5 の Portal Server は、Release 4 バージョンとの下位互換性はありません。
- **アップグレードのロールバック** : Portal Server の Release 5 アップグレードの Release 4 へのロールバックは、Release 4 パッケージの復元、Release 4 ディレクトリデータの復元、および Portal Server Web アプリケーションの Web コンテナへの再配備で構成されます。
- **プラットフォームの問題** : Portal Server をアップグレードするための一般的な手順は Solaris および Linux の両オペレーティングシステムで共通ですが、Release 5 の Portal Server が、Solaris オペレーティングシステムでは新しいパスにインストールされるのに対して、Linux オペレーティングシステムでは Release 4 と同じパスにインストールされます。

Release 4 の Portal Server のアップグレード

ここでは、Solaris プラットフォームと Linux プラットフォームの両方で、Portal Server の Java ES Release 4 から Java ES Release 5 へのアップグレードを実行する方法について説明します。プラットフォーム固有の手順に依存する項目には、その手順が適用されるオペレーティングシステムが示されています。この節で説明する内容は、次のとおりです。

- Release 4 のアップグレード前の作業
- Release 4 の Portal Server のアップグレード (Solaris)
- Release 4 の Portal Server のアップグレード (Linux)
- アップグレードの確認
- Release 4 のアップグレード後の作業
- アップグレードのロールバック (Solaris)
- アップグレードのロールバック (Linux)

Release 4 のアップグレード前の作業

Portal Server をアップグレードする前に、次の作業を実行してください。

- 現在のバージョン情報の確認
- Portal Server が依存するコンポーネントのアップグレード
- 必要な設定情報とパスワードの入手
- Release 4 Portal Server の設定情報のバックアップ
- Java 仮想マシン (JVM) の設定の記録
- ロードバランサの設定の削除
- Directory Proxy Server の設定の削除

現在のバージョン情報の確認

Portal Server の現在のバージョンは、次のコマンドを使用することにより確認できます。

PortalServer6-base/bin/version

表 15-5 Portal Server のバージョン確認の出力

Java ES のリリース	Portal Server のバージョン番号
Release 2	6.3
Release 3	6.3.1
Release 4	6.3.1[1]
IFR リリース	7.0
Release 5	7.1

1. Release 3 と Release 4 の違いはパッチだけです。Release 4 のパッチは、Solaris の showrev -p | grep *patch_ID* コマンドおよび Linux の rpm -qa sun-portal-core コマンドを使用し、これらのバージョンを Java ES Release 4 の『アップグレードガイド』に示されているバージョンと比較することにより確認できます。

Portal Server が依存するコンポーネントのアップグレード

一般的には、コンピュータシステム上 (および、コンピューティング環境内) のすべての Java ES コンポーネントを Java ES Release 5 にアップグレードすることが推奨されます。

Release 5 の Portal Server は Java ES 共有コンポーネントの Release 4 バージョンと互換性がありますが、Portal Server のアップグレードに使用される psupgrade スクリプトには ANT 共有コンポーネントの Release 5 バージョンが必要なため、共有コンポーネントのアップグレードはいずれにしても必要です。

Portal Server 製品コンポーネントの依存関係のいずれかを Release 5 にアップグレードすることを選択した場合は、すべての依存関係をアップグレードする必要があります (310 ページの「選択的なアップグレードに関する問題」を参照)。Portal Server をアップグレードする前に、依存関係を (すでにアップグレードした可能性があるものを省いて) 次の順序でアップグレードするようにしてください。

1. **共有コンポーネント** : Java ES 共有コンポーネントを Release 5 に同期する手順は、67 ページの「Java ES 共有コンポーネントのアップグレード」で説明されています。
2. Directory Server: Directory Server を Release 5 にアップグレードする手順は、103 ページの第 5 章「Directory Server」で説明されています。
3. **Web コンテナソフトウェア** : Web Server または Application Server をアップグレードする手順は、それぞれ、133 ページの第 7 章「Web Server」および 203 ページの第 11 章「Application Server」で説明されています。

   注 Weblogic や WebSphere などのサードパーティー Web コンテナをアップグレードすると、Portal Server をサポートするためにこれらのコンテナに加えられたカスタマイズがコンテナのアップグレードによって上書きされるため、Portal Server が動作しなくなる場合があります。

   この場合は、アップグレードされた Web コンテナ環境で Portal Server を再インストールおよび再設定する必要があります。

4. Access Manager (Access Manager SDK): Access Manager を Release 5 にアップグレードする手順は、257 ページの第 14 章「Access Manager」で説明されています。
5. Portal Server Secure Remote Access: Portal Server Secure Remote Access を Release 5 にアップグレードする手順は、371 ページの第 16 章「Portal Server Secure Remote Access」で説明されています。
6. Java DB: Java DB を Release 5 にアップグレードする手順は、157 ページの第 8 章「Java DB」で説明されています。
7. Service Registry: Service Registry を Release 5 にアップグレードする手順は、229 ページの第 12 章「Service Registry」で説明されています。
8. Communications Express: Communications Express を Release 5 にアップグレードする手順は、『Sun Java Communications Suite アップグレードガイド』(http://docs.sun.com/app/docs/doc/820-0507?l=Ja) で説明されています。

必要な設定情報とパスワードの入手

Web コンテナのアップグレードシナリオ (311 ページの表 15-4 を参照) によっては、psupgrade スクリプトから、パスワードやその他の Web コンテナの設定データに関する情報を入力するように要求されます。Web コンテナの各種のアップグレードシナリオに必要な情報を表 15-6 に示します。Portal Server のアップグレードを開始する前に、関連する情報を集める必要があります。

表 15-6 Web コンテナのアップグレードシナリオ別の psupgrade スクリプトで必要な情報

情報	アップグレードシナリオ[3]	Web Server 7.*x* の値の例 : シナリオ 2	Application Server 8.*x* の値の例 : シナリオ 52
Web Server 7.0 での Portal Server のアップグレード (あり / なし)	2	あり	N/A
Web コンテナのインストールディレクトリ	2 および 5	WebServer7-*base*	*AppServer8Install-base*
Web コンテナの仮想サーバーインスタンス名	2	https-*configName*[4]	N/A
Web コンテナのインスタンス名	5	N/A	server1
Web コンテナのインスタンスディレクトリ	2	*WebServer7Config-base*/https-*configName*[4]/	N/A
ポータルインスタンスの配備ディレクトリ	5	N/A	*AppServer8Config-base*/domains/*domainName*
Web コンテナのインスタンスポート	2 および 5	80	80
Web コンテナのインスタンスプロトコル	2 および 5	http	http
Web コンテナの設定名	2	*configName*[4]	N/A
Web コンテナのドメイン名	5	N/A	domain1
Web コンテナのドキュメントルートディレクトリ	2 および 5	*WebServer7Config-base*/https-*configName*[4]/docs/	*AppServer8Config-base*/domains/*domainName*/docroot
Web コンテナの管理ホスト名	2 および 5	localhost	localhost
Web コンテナの管理ポート	2 および 5	8989	4848
Web コンテナの管理プロトコル	2 および 5	https	https
Web コンテナの管理ユーザー ID	2 および 5	admin	admin
Web コンテナの管理パスワード	2 から 5 まで		

表 15-6　Web コンテナのアップグレードシナリオ別の psupgrade スクリプトで必要な情報 (続き)

情報	アップグレードシナリオ[3]	Web Server 7.*x* の値の例 : シナリオ 2	Application Server 8.*x* の値の例 : シナリオ 52
Web コンテナのマスターパスワード	3 から 5 まで	N/A	
Directory Manager (cn=Directory manager) パスワード	1 から 5 まで		
SRA ログユーザーパスワード[1]	1 から 5 まで		
Access Manager 管理パスワード	1 から 5 まで		
Directory Server ldapuser パスワード	1 から 5 まで		
ポータルインスタンス ID[2]	1 から 5 まで		

1. この情報は、Portal Server にインストールされた Portal Server Secure Remote Access コンポーネントを設定するために必要です。
2. このパラメータには、NULL ではない一意の値を指定してください。この値は英数字にしますが、ハイフン (-) が含まれていてもかまいません。
3. Web コンテナのアップグレードシナリオ 5 は、Portal Server の Release 2 からのアップグレードに適用されます。
4. *configName* のデフォルト値は *hostName.domainName* です。

Release 4 Portal Server の設定情報のバックアップ

Portal Server の Release 5 へのアップグレードでは、Portal Server ソフトウェアの再設定は必要ありません。ただし、安全策として、設定情報が格納されている次のディレクトリは psupgrade スクリプトによってバックアップされます。

PortalServer6Config-base/

注　Portal Server がユーザープロファイルやその他のデータを格納している Directory Server インスタンスをバックアップすることをお勧めします。これらのデータがないと、Release 5 の Portal Server へのアップグレードをロールバックできません。

Java 仮想マシン (JVM) の設定の記録

Portal Server をアップグレードする前に、次の Web コンテナの JVM 設定がデフォルト値と異なっている場合は記録してください。

```
<jvm-options>-XX:MaxPermSize=256m</jvm-options>
<jvm-options>-XX:+CMSPermGenSweepingEnabled</jvm-options>
<jvm-options>-XX:+CMSClassUnloadingEnabled</jvm-options>
```

次の表に示すように、JVM 設定の場所は Web コンテナによって異なります。

表 15-7　JVM 設定の場所

Web コンテナ	設定ファイル
Release 2、3、4 Web Server (6.x)	*WebServer6-base*/https-*instanceName*/config/server.xml
Release 5 Web Server (7.0)	*WebServer7Config-base*/https-*configName*/config/server.xml
Release 3、4、5 の Application Server (8.x)	*AppServer8Config-base*/domains/*domainName*/config/domain.xml
Application Server インスタンス (ノードエージェントが管理)	*AppServer8Config-base*/nodeagents/*nodeagentName*/config/domain.xml

あとで、これらの JVM 設定が Portal Server のアップグレード手順の結果として変更されていないことを確認する必要があります。

ロードバランサの設定の削除

Portal Server インスタンスがロードバランサを介してアクセスされる場合は、このようなアクセスの設定に使用される LOAD_BALANCER_URL プロパティーの値によって Portal Server のアップグレードが妨げられる場合があります。そのため、アップグレードを実行する前に、この設定を変更してください。LOAD_BALANCER_URL プロパティー設定を変更するには、次の手順を実行します。

1. 次のどの設定ファイルがローカルに存在するかを確認してください (これらの設定ファイルの一部は、ローカルにインストールされる可能性のある Portal Server Secure Remote Access コンポーネントをサポートする)。

   PortalServer6Config-base/PSConfig.properties
   PortalServer6Config-base/GWConfig.properties (ゲートウェイがローカルの場合)
   PortalServer6Config-base/RWPConfig.properties (Rewriter プロキシがローカルの場合)
   PortalServer6Config-base/NLPConfig.properties (Netlet プロキシがローカルの場合)

2. これらの設定ファイル内の LOAD_BALANCER_URL プロパティーの現在の値を記録します。

3. 関連する Portal Server インスタンスをポイントするように、LOAD_BALANCER_URL プロパティーの値を変更します。

   LOAD_BALANCER_URL=*portalHostName*:*port*/portal

Directory Proxy Server の設定の削除

Portal Server インスタンスが Directory Proxy Server インスタンスを介して Directory Server にアクセスしている場合は、アップグレードを実行する前に Directory Proxy Server のホストおよびポート番号の設定を変更し、アップグレードが完了したあとに元の値に戻す必要があります。

該当する設定を変更するには、次の手順を実行します。

1. 次の Access Manager 設定ファイル内の `DS_HOST` および `DS_PORT` プロパティーの現在の値を記録します。

   AccessManagerConfig-base`/config/AMConfig.properties`

2. 関連する Directory Server インスタンスを直接ポイントするように、`DS_HOST` および `DS_PORT` プロパティーの値を変更します。

Release 4 の Portal Server のアップグレード (Solaris)

ここでは、Portal Server のアップグレード手順に影響する注意点を示したあと、その手順自体について説明します。

アップグレードの注意点 (Solaris)

Portal Server ソフトウェアの Release 5 へのアップグレードでは、次の注意点を考慮に入れてください。

- 同じ Portal Server インストールイメージに対応するすべての Portal Server インスタンスが同時にアップグレードされます。
- Portal Server ソフトウェアは、いくつかの異なる役割を実行するサブコンポーネントで構成されますが、すべてが同時にアップグレードされます。
  - Portal-base: 管理 Mbean と付随する管理ソフトウェア、ロギングフレームワーク、および監視関連のソフトウェアが含まれ、これらすべてが 1 つにパッケージ化されています。
  - Portal Server Web アプリケーション : Web コンテナに配備されるいくつかの Web アプリケーションで構成されます。これらの Web アプリケーションの少なくとも一部には Access Manager、さらには Directory Server からのサポートが必要です。
  - Secure Remote Access コア : Portal Server Secure Remote Access をサポートするソフトウェア。jar ファイルに組み込まれている一部のサーブレットとアプレット、および Web コンテナに配備できない一部のサポートファイルで構成されています。

- psupgrade スクリプトは、ホストコンピュータにインストールされている Portal Server サブコンポーネントおよび Web コンテナの依存関係を自動的に検出します。たとえば、このスクリプトはシステムに照会して Portal Server Web アプリケーションの配備先の Application Server または Web Server のバージョンを検出し、検出できた情報に応じて要求する情報を調整します。

アップグレード手順 (Solaris)

次に示す手順は、アップグレードを実行するコンピュータ上の Portal Server に適用されます。

1. root としてログインするか、スーパーユーザーになります。

   ```
   su -
   ```

2. すべての共有コンポーネントの Release 5 への同期をまだ実行していない場合は、実行します。

   手順は 67 ページの第 2 章「Java ES 共有コンポーネントのアップグレード」で説明されています。

   この手順は、322 ページの手順 8 で psupgrade スクリプトを実行するために必要な前提条件です。

3. ローカルで実行されている可能性のある Portal Server Secure Remote Access のゲートウェイ、Rewriter プロキシ、または Netlet プロキシのインスタンスをすべて停止します。

   PortalServer6-base/bin gateway stop
   PortalServer6-base/bin netletd stop
   PortalServer6-base/bin rwproxyd stop

   これらのプロセスが停止したことを確認します。

   ゲートウェイ : netstat -an | grep 443
   Rewriter プロキシ : netstat -an | grep 10443
   Netlet プロキシ : netstat -an | grep 10555

4. Access Manager が Portal Server の配備先とは別の Web コンテナに配備されている場合は、それが実行されていることを確認します。

5. Portal Server がまだ実行されていない場合は、配備先の Web コンテナを起動することによって起動します。

   Web Server 6.*x:*
   WebServer-base/https-*instanceName*/start

   Web Server 7.0*:*
   Admin Server--
   WebServer7Config-base/admin-server/bin/startserv
   Instance Server--
   WebServer7Config-base/https-*configName*/bin/startserv

Application Server *8.x:*
AppServer8-base/bin/asadmin start-domain --user *admin_ID*
--password *password domainName*

6. psupgrade スクリプトで必要な 2 つの環境変数 (ANT_HOME および JAVA_HOME) を設定します。次に例を示します。

```
export ANT_HOME=/usr/sfw
export JAVA_HOME=/usr/jdk/entsys-j2se
```

7. コンピュータ上に十分なスワップ空間があることを確認します。

   ガイドラインとして、スワップ空間を物理 RAM の容量の 2 倍に設定することをお勧めします。

8. Java ES Release 5 ディストリビューションから psupgrade スクリプトを実行します。

```
cd os_arch/Products/portal_svr/Tools/upgrade/bin
./psupgrade
```

   ここで、*os_arch* は、Solaris_sparc などのプラットフォームに対応します。

注 間違った *os_arch* ディレクトリから psupgrade を誤って実行した場合は、次のような手順でバックアウトする必要があります。

a. 正しい *os_arch* ディレクトリに移動します。

b. Portal Server データへの変更を元に戻します。

```
./psupgrade rollback
```

   要求されたパラメータとパスワードを指定します。

c. 再度 psupgrade を実行します。

psupgrade スクリプトは、インストール済みの Portal Server コンポーネントおよびローカライズ版パッケージを検出し、Java ES インストーラを起動して新しいパッケージをインストールしたあと、システムに照会して Portal Server Web アプリケーションの配備先の Web コンテナに関連した場所やポート番号、その他の情報を検出します。Web コンテナのアップグレードシナリオ (311 ページの表 15-4 を参照) によっては、スクリプトから、Portal Server を適切な Web コンテナに配備するために必要な追加情報の入力が要求されます。

317 ページの表 15-6 は、表 15-4 に記載されている Web コンテナの各種のアップグレードシナリオで要求される情報を示しています。

注 psupgrade のパラメータには必ず正しい値を入力してください。あとで戻って変更することはできず、また psupgrade スクリプトによって加えられた変更のロールバックもきわめて困難です。Portal Server データへの変更を元に戻すには、次のコマンド

```
./psupgrade rollback
```

を実行してから、再度 psupgrade を実行してみてください。

9. 必要に応じて、Web コンテナの JVM 設定を復元します。

   JVM 設定で Release 5 の Portal Server がサポートされていることを確認するには、次の手順を実行します。

   a. アップグレードの前に記録した Portal Server に対する Web コンテナの JVM 設定が、アップグレード手順の結果として変更されていないことを確認します。

      318 ページの「Java 仮想マシン (JVM) の設定の記録」を参照してください。

   b. 設定が変更されている場合は、アップグレードの前に記録した値に復元します。

      以前に設定されなかったとしても、次の JVM 設定が含まれている必要があります。

      ```
      <jvm-options>-XX:MaxPermSize=256m</jvm-options>
      <jvm-options>-XX:+CMSPermGenSweepingEnabled</jvm-options>
      <jvm-options>-XX:+CMSClassUnloadingEnabled</jvm-options>
      ```

10. Web コンテナを停止し、再起動します。

    すべての状況で必要なわけではありませんが、Web コンテナを再起動すると、Portal Server がクリーンな状態で起動されるようになります。

    a. Web コンテナを次のように停止します。

       Web Server 6.*x:*
       WebServer-base/https-*instanceName*/stop

       Web Server 7.0*:*
       Admin Server--
       WebServer7Config-base/admin-server/bin/stopserv
       Instance Server--
       WebServer7Config-base/https-*configName*/bin/stopserv

       Application Server *8.x:*
       AppServer8-base/bin/asadmin stop-domain --user *admin_ID*
       --password *password domainName*

b. 321 ページの手順 5 にあるコマンドを使用して、Web コンテナを再起動します。

Release 4 の Portal Server のアップグレード (Linux)

ここでは、Portal Server のアップグレード手順に影響する注意点を示したあと、その手順自体について説明します。

アップグレードの注意点 (Linux)

Linux プラットフォームで Portal Server ソフトウェアを Release 5 にアップグレードするときは、Linux オペレーティングシステムでは Release 5 の Portal Server が Release 4 と同じパスにインストールされる点を除き、Solaris プラットフォームのときと同じ注意点 (320 ページの「アップグレードの注意点 (Solaris)」を参照) を考慮に入れてください。その結果、Release 5 の RPM のインストール時に、`psupgrade` スクリプトによって以前の RPM が削除されます。

アップグレード手順 (Linux)

次に示す手順は、アップグレードを実行するコンピュータ上の Portal Server に適用されます。

注意 Linux での Java ES Release 4 から Release 5 へのアップグレードはロールバックできません。次の手順を実行する前に、必ずシステムをバックアップしてください。

1. root としてログインするか、スーパーユーザーになります。

   ```
   su -
   ```

2. すべての共有コンポーネントの Release 5 への同期をまだ実行していない場合は、実行します。

   手順は 67 ページの第 2 章「Java ES 共有コンポーネントのアップグレード」で説明されています。

   この手順は、325 ページの手順 8 で `psupgrade` スクリプトを実行するために必要な前提条件です。

3. ローカルで実行されている可能性のある Portal Server Secure Remote Access のゲートウェイ、Rewriter プロキシ、または Netlet プロキシのインスタンスをすべて停止します。

   PortalServer6-base`/bin gateway stop`
   PortalServer6-base`/bin netletd stop`
   PortalServer6-base`/bin rwproxyd stop`

   これらのプロセスが停止したことを確認します。

ゲートウェイ : `netstat -an | grep 443`
Rewriter プロキシ : `netstat -an | grep 10443`
Netlet プロキシ : `netstat -an | grep 10555`

4. Access Manager が Portal Server の配備先とは別の Web コンテナに配備されている場合は、それが実行されていることを確認します。
5. Portal Server がまだ実行されていない場合は、配備先の Web コンテナを起動することによって起動します。

   Web Server 6.*x:*
   WebServer-base`/https-`*instanceName*`/start`

   Web Server 7.0*:*
   Admin Server--
   WebServer7Config-base`/admin-server/bin/startserv`
   Instance Server--
   WebServer7Config-base`/https-`*configName*`/bin/startserv`

   Application Server *8.x:*
   AppServer8-base`/bin/asadmin start-domain --user` *admin_ID*
   `--password` *password domainName*
6. `psupgrade` スクリプトで必要な 2 つの環境変数 (`ANT_HOME` および `JAVA_HOME`) を設定します。次に例を示します。

   ```
   export ANT_HOME=/opt/sun
   export JAVA_HOME=/usr/jdk/entsys-j2se
   ```
7. コンピュータ上に十分なスワップ空間があることを確認します。

   ガイドラインとして、スワップ空間を物理 RAM の容量の 2 倍に設定することをお勧めします。
8. Java ES Release 5 ディストリビューションから `psupgrade` スクリプトを実行します。

   `cd` *os_arch*`/Products/portal_svr/Tools/upgrade/bin`
   `./psupgrade`

   ここで、*os_arch* は、`Linux_x86` などのプラットフォームに対応します。

   `psupgrade` スクリプトは、インストール済みの Portal Server コンポーネントおよびローカライズ版パッケージを検出し、Java ES インストーラを起動して新しいパッケージをインストールしたあと、システムに照会して Portal Server Web アプリケーションの配備先の Web コンテナに関連した場所やポート番号、その他の情報を検出します。Web コンテナのアップグレードシナリオ (311 ページの表 15-4 を参照) によっては、スクリプトから、Portal Server を適切な Web コンテナに配備するために必要な追加情報の入力が要求されます。

   317 ページの表 15-6 は、表 15-4 に記載されている Web コンテナの各種のアップグレードシナリオで要求される情報を示しています。

注	psupgrade のパラメータには必ず正しい値を入力してください。あとで戻って変更することはできず、また psupgrade スクリプトによって加えられた変更のロールバックもきわめて困難です。**注意** : psupgrade スクリプトを実行する前に、システムをバックアップしてください。

9. *PortalServer7Config-base*/platform.conf.default 設定ファイルを変更します。

   psupgrade によってバックアップされた次のファイルから、gateway.logging.password を含む行をコピーします。

   PortalServer6Config-base.bak/platform.conf.default

   次に、その行を *PortalServer7Config-base*/platform.conf.default に書き込みます。

10. 必要に応じて、Web コンテナの JVM 設定を復元します。

    JVM 設定で Release 5 の Portal Server がサポートされていることを確認するには、次の手順を実行します。

    a. アップグレードの前に記録した Portal Server に対する Web コンテナの JVM 設定が、アップグレード手順の結果として変更されていないことを確認します。

       318 ページの「Java 仮想マシン (JVM) の設定の記録」を参照してください。

    b. 設定が変更されている場合は、アップグレードの前に記録した値に復元します。

       以前に設定されなかったとしても、次の JVM 設定が含まれている必要があります。

       ```
       <jvm-options>-XX:MaxPermSize=256m</jvm-options>
       <jvm-options>-XX:+CMSPermGenSweepingEnabled</jvm-options>
       <jvm-options>-XX:+CMSClassUnloadingEnabled</jvm-options>
       ```

11. Web コンテナを停止し、再起動します。

    すべての状況で必要なわけではありませんが、Web コンテナを再起動すると、Portal Server がクリーンな状態で起動されるようになります。

a. Web コンテナを次のように停止します。

Web Server 6.*x*:
WebServer-base/https-*instanceName*/stop

Web Server 7.0:
Admin Server--
WebServer7Config-base/admin-server/bin/stopserv
Instance Server--
WebServer7Config-base/https-*configName*/bin/stopserv

Application Server *8.x*:
AppServer8-base/bin/asadmin stop-domain --user *admin_ID*
--password *password domainName*

b. 325 ページの手順 5 にあるコマンドを使用して、Web コンテナを再起動します。

アップグレードの確認

Release 5 パッケージのインストールは、次のコマンドを使用することにより確認できます。

PortalServer7-base/bin/psadmin --version --adminuser *admin_ID*
-f *adminpasswordfile*

出力値については、315 ページの表 15-5 を参照してください。

完全アップグレードを確認するには、ポータルデスクトップが起動し、psadmin 管理ユーティリティーが説明のとおりに機能することを確認します。

また、次のアップグレードログファイルを確認することもできます。

/var/sadm/install/logs/Sun_Java_System_Portal_Server_upagrade.log*
PortalServer7Data-base/logs/admin/
PortalServer7Data-base/logs/config/

Release 4 のアップグレード後の作業

次の状況に対処するために必要なアップグレード後の手順に注意してください。

- web-src カスタムデータの移行
- カスタムポートレットアプリケーションの再配備
- カスタマイズされたポートレットアプリケーションの移行
- ブックマークの正しいリンクとアプリケーションチャネル
- 検索サーバーへのアクセスの修正
- Directory Proxy Server の設定の復元

- web-src カスタムデータの移行
- Portal Server Secure Remote Access コンポーネントの手動での登録
- URLScrapper チャネルの有効化
- ログアウトページの変更

web-src カスタムデータの移行

次のディレクトリに、イメージ、javascript ファイル、`portal.war` 作成用のその他のあらゆるファイルなどのカスタムデータを追加している場合 :

PortalServer6-base`/web-src`

これらの追加のファイルを、Release 5 の Portal Server の対応する次のディレクトリにコピーしてください。

PortalServer7-base`/web-src`

カスタムポートレットアプリケーションの再配備

カスタムポートレットアプリケーションを作成および配備している場合は、Release 5 の Portal Server にアップグレードしたあとに、これらのポートレットを手動で再配備してください。ディスプレイプロファイルエントリが存在してチャネル名が表示される場合でも、カスタムポートレットを再配備するまで内容は表示されません。

次のコマンドを使用して、ポートレットを再配備します。

PortalServer7-base`/bin/psadmin deploy-portlet`

次の場所で対応する `.war` および XML ファイルを探すことによって、再配備を確認できます。

PortalServer7Data-base`/portals/Upgraded/war`

カスタマイズされたポートレットアプリケーションの移行

SJWC (Sun Java Web Console) で提供されるユーザーインタフェースフレームワークに基づいたポートレットアプリケーションを手動で Release 5 に移行し、再配備する必要があります。

特に、この要件は、カスタマイズしてポータルにインストールすることを目的としたサンプルポートレットアプリケーションとして Portal Server とともに配布される 4 つの Web アプリケーション、`filesharing`、`surveys`、`wiki`、`rssportlet` に適用されます。アップグレード中、これらのサンプルポートレットアプリケーションのバグ修正バージョンは、ポートレットアプリケーション領域内のディスクに格納されます。これらのポートレットアプリケーションを独自に使用するためにカスタマイズしている場合は、手動で Release 5 に移行し、再配備する必要があります。Portal Server のアップグレードプロセスでは処理されません。

デフォルトでは、コミュニティーが作成されると、これらのポートレットアプリケーションの一部 (filesharing と surveys) が配備され、ユーザーに使用可能になります。

次の手順を使用すると、SJWC ベースのポートレットアプリケーションをアップグレード、カスタマイズ、および再配備できます。例として、ここでは filesharing ポートレットアプリケーションを使用します。

1. Release 5 の SJWC jar ファイルを展開します。
   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. このうちの 1 つのファイルの名前を変更します。

      ```
      mv WEB-INF/lib/commons-collections-3.1.jar
      WEB-INF/lib/commons-collections.jar
      ```
2. filesharing ポートレットアプリケーションの場所を指定します。

   `cd` *PortalServer7Config-base*`/portals/portal1/portletapps/filesharing`
3. 更新された SJWC ライブラリを挿入します。

   `jar uvf src/filesharing.war.tokenized -C /tmp/lh WEB-INF`
4. filesharing ポートレットアプリケーションをカスタマイズします。

   `ant customize`
5. filesharing ポートレットアプリケーションを再配備します。
   a. *PortalServer7-base*`/bin/psadmin undeploy-portlet -u` *amadmin* `-f passwordfile -p` *portal_id* `-i` *instance_id* `-g filesharing`
   b. `ant deploy`

c. Web コンテナに応じて、次のディレクトリに移動します。

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. sun-web.xml ファイルを開き、最後の行の直前 (つまり、sun-web-app 終了タグの前) に次の行を追加します。

```
<class-loader delegate="false"/>
```

e. filesharing/Web-INF の下にある sun-web.xml ファイルに対して、手順 c と手順 d を繰り返します。

6. surveys や、SJWC フレームワークに基づいたその他の任意のカスタムポートレットアプリケーションに対して、手順 2 ～手順 5 を繰り返します。

7. Web コンテナを再起動します。

ブックマークの正しいリンクとアプリケーションチャネル

Release 4 の Portal Server を Release 5 にアップグレードする場合、ブックマークとアプリケーションチャネルは重複した偽のリンクを持ちます。このリンクを修正するには、次の手順を実行します。

1. PSConsole にログインします。
2. 「共通タスク」タブから、「チャネルおよびコンテナを管理」をクリックします。
3. DN として「DeveloperSample [組織]」を選択し、「了解」をクリックします。
4. 「表示タイプ」で「 JSPTabContainer [デフォルト] 」を選択します。
5. 「MyFrontPageTabPanelContainer」にある「App チャネル」をクリックします。

   右側のフレームに App チャネルプロパティーが表示されます。

   特定のロケールのプロパティーを表示するには、「テーブルの設定変更」をクリックし、ロケール値を入力します (de、fr、es、ja、ko、zh、zh_CN、zh_TW)。
6. userApps プロパティーを編集します。

   a. userApps プロパティーで「値を編集 ...」リンクをクリックします。

      既存のアプリケーションを含むポップアップウィンドウが表示されます。

b. 一覧から次のアプリケーションを削除します。

NetMail Lite
NetMail

c. 次のアプリケーションを一覧に追加します。

NetFile

d. 「保存」をクリックし、「閉じる」をクリックします。

7. targets プロパティーを編集します。

a. targets プロパティーで「値を編集 ...」リンクをクリックします。

既存のターゲットを含むポップアップウィンドウが表示されます。

b. 一覧から次のターゲットを削除します。

```
NetMailLite|
NetMailServlet?nsid=newHTMLSessionNetMailLite|
NetMailServlet?nsid=newHTMLSession
```

```
NetMail|NetMailServlet?nsid=newAppletSession
```

c. 次のターゲットが重複して現れた場合、削除します。

```
Instant Messenger (Java WebStart)|
IMLaunch?provider=IMChannel&launch=jnlp&last=false
```

```
Instant Messenger (Browser)|
IMLaunch?provider=IMChannel&launch=plugin&last=false
```

d. 次のターゲットを一覧に追加します。

```
NetFile|/portal/NetFileApplet?Refer=java2
```

e. 「保存」をクリックし、「閉じる」をクリックします。

検索サーバーへのアクセスの修正

Release 4 の Portal Server を Release 5 にアップグレードすると、Portal Server から検索サーバーが分離されるため、検索サーバーへの URL アクセスが変更されます。

- Release 4 では、この URL は `http://`*hostName*`:`*port*`/portal/search` です。
- Release 5 では、この URL は `http://`*hostName*`:`*port*`/UpgradedSearch/search` です。

そのため、Search、DiscussionLite、Discussions、Instant Messaging Channel などの、SearchProvider または DiscussionProvider インタフェースを実装しているすべてのポータルチャネルに対して、ディスプレイプロファイルを手動で変更してください。特に、Display Profile 内でチャネルがどういう組織レベルまたは機能レベルで発生しても検索サーバーが正しく参照されるように、これらのチャネルの `searchServer` プロパティーを変更してください。`searchServer` 値を次のように変更します。

value="http://*hostName*:*port*/UpgradedSearch/search"

また、Instant Messaging Channel の場合は、Instant Messaging Server の設定プロパティー iim_arch.portal.search を新しい検索サーバー URL に更新する必要があります。

Directory Proxy Server の設定の復元

Portal Server インスタンスが Directory Proxy Server インスタンスを介して Directory Server にアクセスしていた場合は、Directory Proxy Server のホストおよびポート番号の設定を、アップグレード前の元の値に戻してください。アップグレードの準備でこれらのプロパティーの値を変更した手順、320 ページの「Directory Proxy Server の設定の削除」を参照してください。

Portal Server Secure Remote Access コンポーネントの手動での登録

アップグレード手順の最後に、標準出力に記録された null ポインタ例外の問題が発生した場合は、Portal Server Secure Remote Access コンポーネントの登録が失敗したことを示しています。

この状況では、次のコマンドを実行することにより、Portal Server Secure Remote Access コンポーネントを手動で登録する (有効にする) ことができます。

PortalServer7-base/bin/psadmin provision-sra -u *amadminUser* -f *passwordFile* -p *Portal_ID* --gateway-profile *profileName* --enable

URLScrapper チャネルの有効化

Release 4 から Release 5 にアップグレードするときには、URLScrapper チャネルを有効にしてください。次の手順に従ってください。

1. Portal Server コンソールにログインします。

   「ポータル」タブをクリックし、アップグレードされたポータルをクリックします。

2. 「DN を選択」ドロップダウンメニューから、「TopLevel [グローバル]」を選択し、「ディスプレイプロファイルをダウンロード」リンクをクリックします。

   ダウンロードされたファイルを、どこかの一時的な場所に格納します。

3. com.sun.portal.providers.urlscraper.URLScraperProvider の場所に移動します。

4. 始まりが

   ```
   <Provider advanced="false"
   class="com.sun.portal.providers.urlscraper.URLScraperProvider"、
   ```

   終わりが

   ```
   </Provider>
   ```

5. 手順 4 の XML の部分を次のように置き換えます。

```
<Provider advanced="false"
class="com.sun.portal.providers.urlscraper.URLScraperProvider"
container="false" lock="false" merge="fuse" name="URLScraperProvider"
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="UrlScraper Channel"/>

<String advanced="false" lock="false" merge="replace" name="description"
propagate="true" value="This is a test for urlscraper"/>

<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="fontFace1"
propagate="true" value="Sans-serif"/>

<String advanced="false" lock="false" merge="replace" name="productName"
propagate="true" value="Sun JavaTM 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. 変更したファイルを保存し、アップロードします。

ログアウトページの変更

Release 5 の Portal Server のログアウトページは、以前の Access Manager のログアウトページから変更されています。この変更がソフトウェアの欠陥を表しているわけではないことに注意してください。

アップグレードのロールバック (Solaris)

ここでは、Portal Server のアップグレードのロールバック手順に影響する注意点を示したあと、その手順自体について説明します。

ロールバックの注意点 (Solaris)

Release 5 へのアップグレードをロールバックする手順は、*PortalServer6-base* の Release 4 インストールへの復元と、Release 4 の Web アプリケーションの再配備で構成されます。

ロールバック手順 (Solaris)

1. root としてログインするか、スーパーユーザーになります。

   ```
   su -
   ```

2. Directory Server をアップグレード前の状態に復元します。

   Directory Server のバックアップ / 復元のコマンド行および GUI ユーティリティーを使用します。『Sun Java System Directory Server Enterprise Edition 6.0 管理ガイド』(http://docs.sun.com/doc/820-0291?l=ja) の「Directory Server バックアップと復元」の章を参照してください。

3. Web コンテナを停止することにより、Portal Server を停止します。

   Web Server 6.*x:*
   WebServer-base/https-*instanceName*/stop

   Web Server 7.0*:*
   Admin Server--
   WebServer7Config-base/admin-server/bin/stopserv
   Instance Server--
   WebServer7Config-base/https-*configName*/bin/stopserv

   Application Server *8.x:*
   AppServer8-base/bin/asadmin stop-domain --user *admin_ID*
   --password *password domainName*

4. Release 5 の Portal Server パッケージを削除します。

   a. Java ES アンインストーラを起動します。

      ```
      /var/sadm/prod/SUNWentsys5/uninstall
      ```

   b. インストールされているすべての Portal Server コンポーネントを選択します。

c. アンインストールに関して選択した項目を確認します。

d. Java ES アンインストーラを終了します。

5. Web コンテナを再起動することにより、Portal Server を再起動します。

   Web Server 6.*x:*
   WebServer-base/https-*instanceName*/start

   Web Server 7.0*:*
   Admin Server--
   WebServer7Config-base/admin-server/bin/startserv
   Instance Server--
   WebServer7Config-base/https-*configName*/bin/startserv

   Application Server *8.x:*
   AppServer8-base/bin/asadmin start-domain --user *admin_ID*
   --password *password domainName*

6. Java ES Release 5 ディストリビューションから、次のコマンドを使用して Release 4 の Portal Server Web アプリケーションを再配備します。

   cd *os_arch*/Products/portal_svr/Tools/upgrade/bin
   ./psupgrade rollback

   ここで、*os_arch* は、Solaris_sparc などのプラットフォームに対応します。

   psupgrade rollback コマンドは、Release 5 の Portal Server Web アプリケーションの配備を取り消し、Release 4 の Portal Server Web アプリケーションを再配備します。

   このコマンドによって、コンテンツが *PortalServer6-base*/web-src から /var/*PortalServe6-base*/https-*hostName*/*deploy-dir*/web-apps に再配備されます。そのため、Portal Server Web アプリケーションをカスタマイズする場合は、最初に /web-src に対して行い、次に /web-apps に配備するようにしてください。/web-apps の下に変更を加えた場合は、psupgrade rollback コマンドを実行する前に /web-src にレプリケートするようにしてください。そうしないと、これらの変更が上書きされます。

7. Web コンテナを停止し、再起動します。

   すべての状況で必要なわけではありませんが、Web コンテナを再起動すると、Portal Server がクリーンな状態で起動されるようになります。

アップグレードのロールバック (Linux)

Release 5 にアップグレードするには Release 4 バイナリを削除する必要があるため、Linux でアップグレードをロールバックすることはきわめて困難です。

ロールバックするための 1 つの方法として、アップグレードの前に並列システムを作成し、アップグレードしようとする前にそのシステムをテストする方法があります。アップグレードのロールバックが必要になった場合は、その並列システムに戻すことができます。

複数インスタンスのアップグレード

配備のアーキテクチャーによっては、スケーラビリティーを実現し、可用性を向上させるために、Portal Server が複数のコンピュータシステムに配備される場合があります。たとえば、ロードバランサを使用して Portal Server インスタンスを複数のコンピュータで実行することにより、負荷を分散させることができます。

Portal Server のインスタンスが負荷分散されている場合は、以降で説明しているように、サービスを中断することなく連続して Portal Server インスタンスをアップグレードする順次アップグレードを実行できます。この手順では、Release 4 の Portal Server が Release 5 の Portal Server ディレクトリデータでは機能しないという制約を考慮に入れてください。

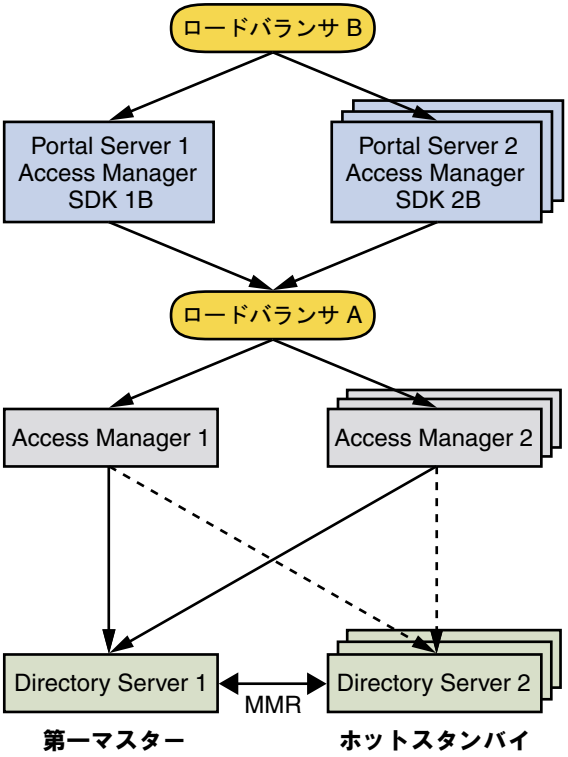
338 ページの図 15-1 に示されている配備アーキテクチャーは、Release 4 の Portal Server インスタンスを Release 5 へ順次アップグレードするための手順を説明するために使用されます。

注 Portal Server Secure Remote Access コンポーネントを含むアーキテクチャーについては、391 ページの「複数インスタンスのアップグレード」を参照してください。

図 15-1 のアーキテクチャーでは、可用性とスケーラビリティを実現するために、複数の Portal Server インスタンスがロードバランサを介してアクセスされます。Portal Server インスタンスは、同様に、ロードバランサを介して Access Manager インスタンスにアクセスします。Access Manager および Access Manager SDK インスタンスは、マルチマスターレプリケーション (MMR) 用に設定されたディレクトリにアクセスします。ほかの Directory Server レプリケーションスキーマも使用できますが、MMR は可用性が高くてスケーラブルな、代表的なディレクトリサービスです。

図 15-1 では、アップグレード手順を説明しやすくするために、Portal Server、Access Manager と Directory Server の複数のインスタンスがグループ化されています。たとえば `Portal Server 2` は、Portal Server の 2 ～ n 番目のインスタンスを表しています。

図 15-1　複数の Portal Server インスタンスに対する配備アーキテクチャーの例



Release 4 の Portal Server の Release 5 への順次アップグレードは、次のように実行されます。

1. Release 4 の Access Manager を Release 5 にアップグレードする場合は、281 ページの「複数インスタンスのアップグレード」の説明に従って順次アップグレードを実行します。Release 4 の Portal Server の Release 5 へのアップグレードでは、Release 4 の Access Manager を Release 5 にアップグレードする必要はないことに注意してください。

2. `Directory Server 1` ではなく `Directory Server 2` をポイントするように、`Portal Server 2` を設定します。

   簡潔にするために、この手順および以降の手順では「`Portal Server 2`」は `Portal Server 2` 〜 `Portal Server n` を意味します。

3. `Portal Server 1` をアップグレードします。

   a. ロードバランサ `B` で `Portal Server 1` を無効にします。

      これで、要求は `Portal Server 1` に経路指定されなくなります。

b. Directory Server MMR を無効にします。

Directory Server 2 は Directory Server 1 と同期されなくなります。

c. Access Manager SDK 1B を Release 5 にアップグレードします。

284 ページの「Release 4 の Access Manager SDK のみのアップグレード」の手順を使用します。

d. Portal Server 1 を Release 5 にアップグレードします。

次の点に注意しながら、314 ページの「Release 4 の Portal Server のアップグレード」の説明に従って Portal Server インスタンスのアップグレードを実行します。

- 319 ページの「ロードバランサの設定の削除」に示されているアップグレード前の作業に特に注意してください。
- アップグレードを実行する前に、ローカルの Access Manager SDK における *AccessManagerConfig-base*/config/AMConfig.properties ファイル内の am.encryption.pwd の値が、それに関連付けられたリモートの Access Manager インスタンスのその値と同じであることを確認してください。
- アップグレードする各 Portal Server インスタンスに対して psupgrade から要求されるポータル インスタンス ID パラメータには、NULL ではない一意の値を必ず指定してください。

Directory Server 1 の Portal Server データが Release 5 に更新されます。

e. ロードバランサ B で Portal Server 1 を有効にします。

要求は、ふたたび Portal Server 1 に経路指定されるようになります。

4. Portal Server 2 をアップグレードします。

a. ロードバランサ B で Portal Server 2 を無効にします。

これで、要求は Portal Server 2 に経路指定されなくなります。

b. Directory Server 1 をポイントするように、Portal Server 2 の設定を復元します。

c. Access Manager SDK 2B を Release 5 にアップグレードします。

339 ページの手順 c と同じ手順を使用します。

d. Portal Server 2 を Release 5 にアップグレードします。

339 ページの手順 d と同じ手順を使用します。

e. ロードバランサ B で Portal Server 2 を有効にします。

要求は、ふたたび Portal Server 2 に経路指定されるようになります。

5. Directory Server MMR を有効にします。

これで、Directory Server 2 の Portal Server データが Directory Server 1 と同期されます。

Java ES Release 3 からの Portal Server のアップグレード

Java ES 2005Q1 (Release 3) Portal Server を Release 5 にアップグレードする手順は、次の例外を除き、Release 4 の Portal Server を Release 5 にアップグレードする手順と同じです。

- Release 3 アップグレード前の作業 : Portal Server の依存関係のアップグレード
- Release 3 の Portal Server のアップグレード
- 複数インスタンスのアップグレード

Release 3 アップグレード前の作業 : Portal Server の依存関係のアップグレード

Portal Server を Release 3 からアップグレードする場合は、Portal Server をアップグレードする前に Access Manager と Web コンテナ (Web Server または Application Server) の両方を Release 4 または Release 5 にアップグレードしてください。いずれかの依存関係を Release 3 のままにすることも、一部の依存関係を Release 4 に、それ以外を Release 5 にアップグレードすることもできません。詳細については、310 ページの「選択的なアップグレードに関する問題」を参照してください。

次の依存関係は、下記に示す順序でアップグレードする必要があります。

1. 共有コンポーネント : Java ES 共有コンポーネントを Release 5 にアップグレードする手順は、67 ページの第 2 章「Java ES 共有コンポーネントのアップグレード」で説明されています。
2. Directory Server: Directory Server を Release 5 にアップグレードする手順は、103 ページの第 5 章「Directory Server」で説明されています。
3. Web コンテナソフトウェア : Web Server または Application Server をアップグレードする手順は、それぞれ、133 ページの第 7 章「Web Server」で説明されています。

注	Weblogic や WebSphere などのサードパーティー Web コンテナをアップグレードすると、Portal Server をサポートするためにこれらのコンテナに加えられたカスタマイズがコンテナのアップグレードによって上書きされるため、Portal Server が動作しなくなる場合があります。 この場合は、アップグレードされた Web コンテナ環境で Portal Server を再インストールおよび再設定する必要があります。

4. Access Manager (Access Manager SDK): Access Manager を Release 5 にアップグレードする手順は、257 ページの第 14 章「Access Manager」で説明されています。

Release 3 の Portal Server のアップグレード

Release 3 Portal Server を Release 5 にアップグレードするには、「Release 4」を「Release 3」と読み替えた上で、313 ページの「Java ES Release 4 からの Portal Server のアップグレード」の手順を使用してください。

Release 3 のアップグレード後の作業

Portal Server を Release 3 から Release 5 にアップグレードする場合は、327 ページの「Release 4 のアップグレード後の作業」で説明されているアップグレード後の手順に加えて、次の状況に対処するために必要なアップグレード後の手順を実行してください。

- ディスカッションへの登録

ディスカッションへの登録

コミュニティー内のディスカッションに登録するには、まずグローバルディスプレイプロファイルの最上位プロパティーを編集して、次の String プロパティーを追加してください。

```
helpURL=en/desktop/usedesk.htm
```

次の手順に従ってください。

1. ディスプレイプロファイル XML snippet ファイル `helpUrl.xml` を作成します。

```
<?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. 次のコマンドを使用して、グローバルディスプレイプロファイルのプロパティーを実行します。

   ```
   ./psadmin modify-dp -u amadminUser -f /tmp/passwordFile -p portal_ID
       -m -g helpUrl.xml
   ```

   ここで、-m オプションは、グローバルディスプレイプロファイル全体を上書きしないために必要です。

複数インスタンスのアップグレード

配備のアーキテクチャーによっては、スケーラビリティーを実現し、可用性を向上させるために、Portal Server が複数のコンピュータシステムに配備される場合があります。たとえば、ロードバランサを使用して Portal Server インスタンスを複数のコンピュータで実行することにより、負荷を分散させることができます。

Portal Server のインスタンスが負荷分散されている場合は、以降で説明しているように、サービスを中断することなく連続して Portal Server インスタンスをアップグレードする順次アップグレードを実行できます。この手順では、Release 3 の Portal Server が Release 5 の Portal Server ディレクトリデータでは機能しないという制約を考慮に入れてください。

Release 3 の Portal Server から Release 5 への順次アップグレードを実行するには、「Release 4」を「Release 3」と読み替えた上で、337 ページの「複数インスタンスのアップグレード」の手順を使用してください。さらに、338 ページの手順 1 の説明に従って、Access Manager をアップグレードしてください。

Java ES Release 2 からの Portal Server のアップグレード

ここでは、Java ES 2004Q2 (Release 2) の Portal Server の Release 5 へのアップグレードについて説明します。このアップグレード手順は、以降の節に説明されているいくつかの変更を除き、Release 4 の Portal Server の Release 5 へのアップグレードとほぼ同じです。

- Release 2 のアップグレード前の作業
- Release 2 の Portal Server のアップグレード
- Release 2 のアップグレード後の作業
- 複数インスタンスのアップグレード

注 Linux プラットフォームで Release 2 の Portal Server からアップグレードする場合は、二重のアップグレード、つまり、Portal Server とオペレーティングシステムの両方をアップグレードする必要があります (Release 5 の Portal Server は RHEL 2.1 でサポートされていない)。詳細については、312 ページの「二重のアップグレード」を参照してください。

Release 2 のアップグレード前の作業

Portal Server を Release 2 からアップグレードするときのアップグレード前の作業は、Portal Server の依存関係のアップグレードを除き、315 ページの「Release 4 のアップグレード前の作業」で説明されている作業と同じです。

Portal Server を Release 2 からアップグレードする場合は、Portal Server をアップグレードする前に Access Manager と Web コンテナ (Web Server または Application Server) の両方を Release 4 または Release 5 にアップグレードしてください。いずれかの依存関係を Release 2 のままにすることも、一部の依存関係を Release 4 に、それ以外を Release 5 にアップグレードすることもできません。詳細については、310 ページの「選択的なアップグレードに関する問題」を参照してください。

特に、Web コンテナソフトウェアを Release 2 からアップグレードしてください。つまり、`psupgrade` スクリプトを実行する場合は、311 ページの表 15-4 のシナリオ 2 とシナリオ 5 だけがサポートされます。

次の依存関係は、下記に示す順序でアップグレードする必要があります。

1. 共有コンポーネント : Java ES 共有コンポーネントを Release 5 にアップグレードする手順は、67 ページの第 2 章「Java ES 共有コンポーネントのアップグレード」で説明されています。

2. Directory Server: Directory Server を Release 5 にアップグレードする手順は、117 ページの「Java ES Release 2 からの Directory Server のアップグレード」で説明されています。
3. Web コンテナソフトウェア : Web Server または Application Server をアップグレードする手順は、それぞれ、133 ページの第 7 章「Web Server」および 203 ページの第 11 章「Application Server」で説明されています。

注	Weblogic や WebSphere などのサードパーティー Web コンテナをアップグレードすると、Portal Server をサポートするためにこれらのコンテナに加えられたカスタマイズがコンテナのアップグレードによって上書きされるため、Portal Server が動作しなくなる場合があります。この場合は、アップグレードされた Web コンテナ環境で Portal Server を再インストールおよび再設定する必要があります。

4. Access Manager (Access Manager SDK): Access Manager を Release 5 にアップグレードする手順は、257 ページの第 14 章「Access Manager」で説明されています。

Release 2 の Portal Server のアップグレード

Portal Server を Release 2 から Release 5 にアップグレードする手順は、以降の節で説明しているように、Portal Server ソフトウェアを配備している Web コンテナによって異なります。

Release 2 の Portal Server のアップグレード : Web Server Web コンテナ

Web Server Web コンテナに配備している場合、Release 2 の Portal Server を Release 5 にアップグレードするには、「Release 4」を「Release 2」と読み替えた上で、313 ページの「Java ES Release 4 からの Portal Server のアップグレード」の手順を使用します。

ただし、Portal Server が Release 5 の Web Server (Web Server 7.0) に配備されている場合は、Release 2 の Web Server をアップグレードする前に、次の手順を実行してください。

1. Web Server 管理コンソールにログインします。
2. 「仮想サーバーを編集」> 「Web アプリケーション」をクリックします。
3. URI に `/portal` または `/portalsamples` のどちらかが含まれている配備済みの Web アプリケーションをすべて削除します。
4. 「保存」をクリックします。

5. 「配備保留中」をクリックします。

Release 2 の Portal Server のアップグレード : Application Server Web コンテナ

Application Server Web コンテナに配備している場合、Release 2 の Portal Server を Release 5 にアップグレードすると、Application Server は Release 2 から Release 5 にアップグレードされます。

Portal Server が最初に配備されたときの Release 2 の Application Server インスタンス (*instanceName*) は、Release 5 にアップグレードされると、Application Server のアップグレード処理によって作成されたノードエージェントの下へ移行されています。このアップグレードされた Application Server インスタンス内の Portal Server のアップグレードには、次の手順が必要です。

1. root としてログインするか、スーパーユーザーになります。

   ```
   su -
   ```

2. すべての共有コンポーネントの Release 5 への同期をまだ実行していない場合は、実行します。

   手順は 67 ページの第 2 章「Java ES 共有コンポーネントのアップグレード」で説明されています。

   この手順は、346 ページの手順 9 で `psupgrade` スクリプトを実行するために必要な前提条件です。

3. ローカルで実行されている可能性のある Portal Server Secure Remote Access のゲートウェイ、Rewriter プロキシ、または Netlet プロキシのインスタンスをすべて停止します。

   PortalServer6-base`/bin gateway stop`
   PortalServer6-base`/bin netletd stop`
   PortalServer6-base`/bin rwproxyd stop`

   これらのプロセスが停止したことを確認します。

   ゲートウェイ : `netstat -an | grep 443`
   Rewriter プロキシ : `netstat -an | grep 10443`
   Netlet プロキシ : `netstat -an | grep 10555`

4. Access Manager が Portal Server の配備先とは別の Web コンテナに配備されている場合は、それが実行されていることを確認します。

5. Portal Server がまだ実行されていない場合は、配備先の Web コンテナを起動することによって起動します。

   a. まだドメイン管理サーバー (DAS) が稼働していない場合は起動します。

      AppServer8-base`/bin/asadmin start-domain --user` *admin_ID*
      `--password` *password domainName*

b. Portal Server が配備されている、アップグレードされた Application Server インスタンス (*instanceName*) がまだ稼働していない場合、起動します。

これは、アップグレードされた Application Server の移行先ノードエージェントを起動することによって行います。

```
AppServer8-base/bin/asadmin start-node-agent --user admin_ID
    --password password nodeagentName
```

上のコマンドおよび以降の手順では、次の規則が使用されています。

- ここで、*nodeagentName* の形式は *hostName_domainName* ですが、デフォルトでは単に *hostName* です。
- デフォルトの *domainName* は `domain1` です。
- デフォルトの *instanceName* は `server1` です。

6. Portal Server コンポーネントの配備を取り消します。

```
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. `psupgrade` スクリプトで必要な 2 つの環境変数 (`ANT_HOME` および `JAVA_HOME`) を設定します。次に例を示します。

Solaris OS

```
export ANT_HOME=/usr/sfw
export JAVA_HOME=/usr/jdk/entsys-j2se
```

Linux OS:

```
export ANT_HOME=/opt/sun
export JAVA_HOME=/usr/jdk/entsys-j2se
```

8. コンピュータ上に十分なスワップ空間があることを確認します。

ガイドラインとして、スワップ空間を物理 RAM の容量の 2 倍に設定することをお勧めします。

9. Java ES Release 5 ディストリビューションから `psupgrade` スクリプトを実行します。

```
cd os_arch/Products/portal_svr/Tools/upgrade/bin
./psupgrade
```

ここで、*os_arch* は、`Solaris_sparc` などのプラットフォームに対応します。

注 間違った *os_arch* ディレクトリから psupgrade を誤って実行した場合は、次のような手順でバックアウトする必要があります。

a. 正しい *os_arch* ディレクトリに移動します。

b. Portal Server データへの変更を元に戻します。

```
./psupgrade rollback
```

要求されたパラメータとパスワードを指定します。

c. 再度 psupgrade を実行します。

psupgrade スクリプトは、Java ES インストーラを起動して新しいパッケージをインストールし、システムに照会して Portal Server Web アプリケーションの配備先の Web コンテナに関連した場所やポート番号、その他の情報を検出します。Web コンテナのアップグレードシナリオ (311 ページの表 15-4 を参照) によっては (この場合は、シナリオ 5)、スクリプトから、Portal Server を適切な Web コンテナに配備するために必要な情報の入力が要求されます。

317 ページの表 15-6 は、Release 2 の Application Server を Release 5 にアップグレードしたとき (シナリオ 5) に要求される情報を示しています。

注 psupgrade のパラメータには必ず正しい値を入力してください。あとで戻って変更することはできず、また psupgrade スクリプトによって加えられた変更のロールバックもきわめて困難です。Portal Server データへの変更を元に戻すには、次のコマンド

```
./psupgrade rollback
```

を実行してから、再度 psupgrade を実行してみてください。

10. 345 ページの手順 5 で起動したドメイン管理サーバー (DAS) とノードエージェントを停止します。

```
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. 手順 10 で停止したドメイン管理サーバー (DAS)、ノードエージェント、およびサーバーインスタンスを再起動します。

注 サーバーインスタンスを起動する前に、下に示すように、`startinstances=false` オプションを使用してノードエージェントを別々に起動する必要があります。

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 のデフォルト値は `4848` です。

Release 2 のアップグレード後の作業

Portal Server を Release 2 から Release 5 にアップグレードする場合は、327 ページの「Release 4 のアップグレード後の作業」で説明されているアップグレード後の手順に加えて、次の状況に対処するために必要なアップグレード後の手順を実行してください。

- シングルサインオン設定
- URLScrapper チャネルの有効化
- ゲートウェイサービスエントリの削除

シングルサインオン設定

Portal Server を Release 2 からアップグレードしたあと、メール、カレンダ、アドレス帳などの、SSO アダプタメタテンプレートを使用してバックエンドサーバーにアクセスするコミュニケーションチャネルは失敗します。

たとえば、Release 2 のメール SSO アダプタメタテンプレートである `SUN-UWC-MAIL` を Messaging Server 固有の設定を使用して変更している場合は、Release 5 へのアップグレード後に、2 つの `SUN-UWC-MAIL` SSO アダプタメタテンプレート (変更されていない独自の Release 2 バージョンと、新しい Release 5 バージョン) が存在します。Portal Server コンソールや `psadmin` コマンド行インタフェースに、どちらも同じ名前を持つ、重複した SSO アダプタメタテンプレートが表示されます。

SSO アダプタメタテンプレートを使用するチャネルは、バックエンドサーバーとの接続を確立したり、データを取得したりすることはできません。

この問題を修正するには、SSO アダプタメタテンプレートのデータを取得し、重複したエントリの名前を変更したあと、変更されたデータを置き換えてください。次の手順に従ってください。

1. SSO アダプタメタテンプレートのデータをエクスポートします。

   次のように、amadmin ユーティリティーを使用して、Access Manager サービスのデータをエクスポートします。

   a. amadmin 要求ファイル /tmp/ssoadapter-template-gets.xml を作成します。

      このファイルは、SSO アダプタメタテンプレートのデータを取得するユーティリティーによって使用されます。

      ```
      <?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. 次の asadmin コマンドを実行します。

      AccessManager-base/bin/amadmin -u *amadminUser* -w *password* -t ssoadapter-templates-get.xml > /tmp/ssoadapter-templates.xml

      このコマンドの出力は、/tmp/ssoadapter-templates.xml に保存されます。

      /tmp/ssoadapter-templates.xml ファイルは、次の形式をしています。

      ```
      sunConfigurationTemplates=
      [<SSO アダプタメタテンプレート 1>, <SSO アダプタメタテンプレート 2>, ...]
      ```

      また、各 <SSO アダプタメタテンプレート> の構文は次のとおりです。

      ```
      default|imap:/?configName=SUN-UWC-MAIL
      &proxyAdminPassword=%5BPROXY-ADMIN_PASSWORD%5D&subType=sun-one
      &enableProxyAuth=false ...
      ```

2. /tmp/ssoadapter-templates.xml ファイルを変更して、重複した SSO アダプタメタテンプレートの名前を変更します。

   a. /tmp/ssoadapter-templates.xml ファイル内の各テンプレートを見つけます。

      default|imap:/?configName= の文字列を探します。

b. 重複した SSO アダプタメタテンプレートの名前を一意の値に置き換えます。

たとえば、SUN-UWC-MAIL SSO アダプタメタテンプレートが 2 つある場合は、そのうちの 1 つの configName 値を SUN-UWC-MAIL2 に置き換えて、2 つのテンプレートの名前が一意になるようにします。

```
default|imap:/?configName=SUN-UWC-MAIL ...
default|imap:/?configName=SUN-UWC-MAIL2 ...
```

3. 変更された SSO アダプタメタテンプレートをインポートして元のデータを上書きする amadmin 要求ファイルを作成します。

a. /tmp/ssoadapter-templates.xml を /tmp/ssoadapter-new-templates.xml にコピーします。

b. /tmp/ssoadapter-new-templates.xml で、次の文字列

```
sunConfigurationTemplates=[
```

を次のように置き換えます。

```
<?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"/>
```

c. すべてのアンパサンド ("&") を "&" に置き換えます。

たとえば、次の行

```
default|imap:/?configName=SUN-UWC-MAIL
&proxyAdminPassword=%5BPROXY-ADMIN_PASSWORD%5D
&subType=sun-one&enableProxyAuth=false...
```

は次のようになります。

```
default|imap:/?configName=SUN-UWC-MAIL
&amp;proxyAdminPassword=%5BPROXY-ADMIN_PASSWORD%5D
&amp;subType=sun-one&amp;enableProxyAuth=false ...
```

d. 各 SSO アダプタメタテンプレートの最後にあるコンマ (",") を削除します。

e. 開始の <Value> タグと終了の </Value> タグを使用して、各 SSO アダプタメタテンプレートをラップします。

次に例を示します。

```
<Value>default|imap:/?configName=SUN-UWC-MAIL
&amp;proxyAdminPassword=%5BPROXY-ADMIN_PASSWORD%5D
&amp;subType=sun-one&amp;enableProxyAuth=false ...</Value>
```

f. 最後の SSO アダプタメタテンプレートから、終了の角括弧 ("]") を削除します。

g. ファイルの最後に次の行を追加します。

```
        </AttributeValuePair>
      </ModifyDefaultValues>
    </SchemaRequests>
  </Requests>
```

上記の手順で使用した 1 つのテンプレートに対して得られる ssoadapter-new-templates.xml ファイルは、次のようになるはずです。

```
<?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. 新しい ssoadapter-new-templates.xml ファイルをインポートします。

AccessManager-base/bin/amadmin -u *amadminUser* -w *password* -v -t ssoadapter-new-templates.xml

この時点で、Portal Server コンソールの SSO アダプタタブにアクセスして、更新された ssoadapter を表示できます。

URLScrapper チャネルの有効化

Release 3 から Release 5 にアップグレードするときには、URLScrapper チャネルを有効にしてください。332 ページの「URLScrapper チャネルの有効化」を参照してください。

ゲートウェイサービスエントリの削除

Portal Server を Release 2 からアップグレードした場合は、`amService-srapGateway` ユーザーエントリを手動で削除してください。そうしないと、アップグレードのあと、Portal Server Secure Remote Access ゲートウェイコンポーネントが使用されている場合は、起動に失敗します。次の手順を実行します。

1. Access Manager コンソールにログインします。
2. 組織 DN 内のすべてのユーザーを表示します。
3. `amService-srapGateway` ユーザーを削除します。

複数インスタンスのアップグレード

Release 2 の Portal Server の Release 5 へのアップグレードでは、複数インスタンスの順次アップグレード (337 ページの「複数インスタンスのアップグレード」を参照) はサポートされていません。

Portal Server の IFR 7.0 からのアップグレード

ここでは、暫定機能リリース (Interim Feature Releases、IFR) 7.0 2005Q4 から Java ES 5 (Release 5) への Portal Server のアップグレードについて説明します。

この節で説明する内容は、次のとおりです。

- Portal Server IFR のアップグレードの概要
- Portal Server IFR 7.0 のアップグレード
- 複数インスタンスのアップグレード

Portal Server IFR のアップグレードの概要

Portal Server IFR 7.0 を Release 5 の Portal Server にアップグレードするときは、アップグレードプロセスに関する次の事項を検討します。

- Portal Server IFR は Application Server 7.*x* でサポートされていないため、311 ページの表 15-4 にあるアップグレードシナリオ 5 は適用されません。
- Portal Server IFR の Release 5 へのアップグレードに使用される `psupgrade` スクリプトは、Release 4 からのアップグレードと異なり、新しいパッケージをインストールしません。代わりに、アップグレード手順では、次のパッチを適用するよう求められます。

表 15-8 Portal Server IFR を Release 5 にアップグレードするためのパッチ[1]

説明	パッチ ID: Solaris 9 および 10	パッチ ID: Linux
Portal Server 7.1	121465-28 (SPARC)	121467-28
	121466-28 (x86)	
Portal Server 7.1 ローカライズ版	123254-02 (SPARC)	123255-02
	124590-02 (x86)	

1. この表に示されているパッチのリビジョン番号は、Java ES Release 5 にアップグレードするための最低要件です。表に記載されているものよりも新しいリビジョンのパッチが使用可能になった場合は、新しいほうのパッチを使用してください。

- `psupgrade` スクリプトは、Release 5 の Web Server に対して、Portal Server IFR の Release 5 へのアップグレードをサポートしていません。この制限には次の 2 つの意味があります。
  - Web Server がすでに Release 5 にアップグレードされている場合は、以前の Web Server コンテナに配備されている Portal Server IFR ソフトウェアをアップグレードすることはできません。そのため、311 ページの表 15-4 の Web コンテナのアップグレードシナリオ 2 はサポートされません。61 ページの「特殊な場合」の警告に注意してください。
  - Web Server がまだ Release 5 にアップグレードされていない場合、後出の「Portal Server IFR 7.0 のアップグレード」に説明されているように、まず以前の Web Server (6.x) コンテナにある Portal Server IFR ソフトウェアを Release 5 にアップグレードし、次に Web Server (および、必要に応じて Access Manager) を Release 5 にアップグレードできます。このアップグレードシナリオの場合は、詳細な手順について 363 ページの「Release 5 の Web Server (7.0) Web コンテナでのアップグレード」を参照してください。

Portal Server IFR 7.0 のアップグレード

この節では、Solaris および Linux プラットフォームで、Portal Server を IFR から Java ES Release 5 にアップグレードする方法について説明します。プラットフォーム固有の手順に依存する項目には、その手順が適用されるオペレーティングシステムが示されています。この節で説明する内容は、次のとおりです。

- IFR 7 のアップグレード前の作業
- Portal Server IFR 7.0 のアップグレード (Solaris)
- IFR 7 の Portal Server のアップグレード (Linux)
- アップグレードの確認
- IFR 7 のアップグレード後の作業
- アップグレードのロールバック (Solaris)
- アップグレードのロールバック (Linux)

IFR 7 のアップグレード前の作業

IFR をアップグレードするときのアップグレード前の作業は、次に挙げる例を除き、Release 4 のアップグレード手順 (315 ページの「Release 4 のアップグレード前の作業」を参照) と同じです。

- 必要な設定情報の入手
- 共通エージェントコンテナの設定

必要な設定情報の入手

317 ページの「必要な設定情報とパスワードの入手」で説明されているように、`psupgrade` スクリプトが要求した情報は Portal Server IFR からのアップグレードに完全には適用されません。Portal Server IFR は Application Server 7.*x* でサポートされていないため、317 ページの表 15-6 にある Web コンテナのアップグレードシナリオ 5 は適用できません。

共通エージェントコンテナの設定

共通エージェントコンテナは、Java ES 監視および管理エージェントのためのコンテナサービスを提供する共有コンポーネントです。Portal Server コンソール、`psadmin` コマンド行インタフェースなどの Portal Server 管理ツールは、共通エージェントコンテナに配備される一連の監視および管理エージェント (ポータル管理サーバーと総称される) を使用します。

Portal Server IFR 7.0 のアップグレードを実行する前に Java ES 共有コンポーネントが Release 5 にアップグレードされている場合、Release 5 の Portal Server コンソールにログインして psadmin コマンド行インタフェースを使用するには次の追加の手順が必要です。Portal ServerIFR 7.0 のアップグレードを実行する前に Java ES 共有コンポーネントが Release 5 にアップグレードされていない場合は、次の追加の手順を無視してください。

1. 共通エージェントコンテナを再設定します。

   PortalServer7-base/bin/psconfig --config
   PortalServer7-base/samples/example2.xml

   example2.xml ファイルによって、再設定の情報が提供されます。psconfig コマンドを実行する前に、まず example2.xml ファイルを編集して、必要なパスワードを入力してください。Portal Server のデフォルト以外の場所を使用している場合は、正しいディレクトリも入力してください。

2. 共通エージェントコンテナを参照するように、Web コンテナのクラスパスを編集します。

   Web コンテナのクラスパスには、以前の共通エージェントコンテナリリースの場所への参照 (*rel4CAC-base-dir*/lib/cacao_cacao.jar) が含まれます。

   この参照を Release 5 の場所 (*rel5CAC-admin-dir*/lib/cacao_cacao.jar) に置き換えます。

3. *PortalServer7Config-base* ディレクトリに、次のプロパティーを含むプロパティーファイル pasconnect.properties が作成されていることを確認します。

   pas.host=

   可能なプロパティー値は、null、localhost、または実際の Portal Server ホスト名です。

4. 共通エージェントコンテナを再起動します。

   rel5CAC-admin-dir/bin/cacaoadm start

Portal Server IFR 7.0 のアップグレード (Solaris)

ここでは、Portal Server IFR のアップグレード手順に影響する注意点を示したあと、その手順自体について説明します。

IFR 7 のアップグレードの注意点 (Solaris)

Portal Server IFR の Java ES Release 5 へのアップグレードでは、Release 4 のアップグレードと同じ注意点 (320 ページの「アップグレードの注意点 (Solaris)」を参照) を考慮に入れてください。

さらに、353 ページの「Portal Server IFR のアップグレードの概要」で説明されている問題も参照してください。

IFR 7 のアップグレード手順 (Solaris)

次に示す手順は、アップグレードを実行するコンピュータ上の Portal Server に適用されます。

1. root としてログインするか、スーパーユーザーになります。

   ```
   su -
   ```

2. ローカルで実行されている可能性のある Portal Server Secure Remote Access のゲートウェイ、Rewriter プロキシ、または Netlet プロキシのインスタンスをすべて停止します。

   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*

   これらのプロセスが停止したことを確認します。

   ゲートウェイ : `netstat -an | grep 443`
   Rewriter プロキシ : `netstat -an | grep 10443`
   Netlet プロキシ : `netstat -an | grep 10555`

3. Access Manager が Portal Server の配備先とは別の Web コンテナに配備されている場合は、それが実行されていることを確認します。

4. Portal Server がまだ実行されていない場合は、配備先の Web コンテナを起動することによって起動します。

   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. 353 ページの表 15-8 に基づいて、必要なパッチを入手します。

   特定のリビジョンを使用するように指示された場合以外は、常に、使用可能な最新のパッチリビジョンを使用してください。

   パッチは http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access から /tmp にダウンロードします。

6. 表 15-8 の適切な Portal Server パッチと、必要に応じてローカライズ版パッチを適用します。

   ```
   patchadd /tmp/patch_ID
   ```

7. パッチのアップグレードが成功したことを確認します。

   ```
   showrev -p | grep patch_ID
   ```

   この出力では、手順 6 で適用したパッチ ID のバージョンが返されるはずです。

8. 手順 6 でローカライズ版パッケージをアップグレードした場合は、Portal Server コンソール JVM のロケールを UTF-8 に設定します。

   ```
   export LC_ALL=ja_JP.UTF-8
   export LANG=ja_JP.UTF-8
   ```

9. psupgrade スクリプトで必要な 2 つの環境変数 (ANT_HOME および JAVA_HOME) を設定します。

   ```
   export ANT_HOME=/usr/sfw
   export JAVA_HOME=/usr/jdk/entsys-j2se
   ```

10. コンピュータ上に十分なスワップ空間があることを確認します。

    ガイドラインとして、スワップ空間を物理 RAM の容量の 2 倍に設定することをお勧めします。

11. psupgrade スクリプトを実行します。

    ```
    cd PortalServer7-base/bin
    ./psupgrade
    ```

    psupgrade スクリプトは Java ES Release 5 ディストリビューションから実行されず、Java ES インストーラも起動しません (パッケージにはすでにパッチが適用済み)。

    このスクリプトは、システムに照会して Portal Server Web アプリケーションの配備先の Web コンテナに関連した場所やポート番号、その他の情報を検出します。Web コンテナのアップグレードシナリオ (311 ページの表 15-4 を参照) によっては、スクリプトから、Portal Server を適切な Web コンテナに配備するために必要な追加情報の入力が要求されます。

    317 ページの表 15-6 は、311 ページの表 15-4 に記載されている Web コンテナの各種のアップグレードシナリオで要求される情報を示しています。

    注 psupgrade のパラメータには必ず正しい値を入力してください。あとで戻って変更することはできず、また psupgrade スクリプトによって加えられた変更のロールバックもきわめて困難です。

12. Web コンテナを停止し、再起動します。

すべての状況で必要なわけではありませんが、Web コンテナを再起動すると、Portal Server がクリーンな状態で起動されるようになります。

IFR 7 の Portal Server のアップグレード (Linux)

ここでは、Portal Server のアップグレード手順に影響する注意点を示したあと、その手順自体について説明します。

IFR 7 のアップグレードの注意点 (Linux)

Linux プラットフォームで Portal Server IFR ソフトウェアを Release 5 にアップグレードするときは、Linux オペレーティングシステムで Release 5 のパッチをインストールすると以前の RPM が削除される点を除き、Solaris プラットフォームのときと同じ注意点 (355 ページの「IFR 7 のアップグレードの注意点 (Solaris)」を参照) を考慮に入れてください。

IFR 7 のアップグレード手順 (Linux)

次に示す手順は、アップグレードを実行するコンピュータ上の Portal Server に適用されます。

注意 Linux での Portal Server IFR から Release 5 へのアップグレードはロールバックできません。次の手順を実行する前に、必ずシステムをバックアップしてください。

1. root としてログインするか、スーパーユーザーになります。

   ```
   su -
   ```

2. ローカルで実行されている可能性のある Portal Server Secure Remote Access のゲートウェイ、Rewriter プロキシ、または Netlet プロキシのインスタンスをすべて停止します。

   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*

   これらのプロセスが停止したことを確認します。

   ゲートウェイ : `netstat -an | grep 443`
   Rewriter プロキシ : `netstat -an | grep 10443`
   Netlet プロキシ : `netstat -an | grep 10555`

3. Access Manager が Portal Server の配備先とは別の Web コンテナに配備されている場合は、それが実行されていることを確認します。
4. Portal Server がまだ実行されていない場合は、配備先の Web コンテナを起動することによって起動します。

   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. 353 ページの表 15-8 のパッチ番号と RPM 名に基づいて、必要なパッチを入手します。

   特定のリビジョンを使用するように指示された場合以外は、常に、使用可能な最新のパッチリビジョンを使用してください。

   パッチは http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access から /tmp にダウンロードします。

6. 353 ページの表 15-8 の Portal Server パッチと、必要に応じて Portal Server のローカライズ版 RPM を、その順序で適用します。

   Portal Server パッチの Readme ファイルを参照すると、スクリプトを使用してパッチの RPM を適用する方法が説明されています。

   ```
   cd /tmp
   ```

   ここで、/tmp は、パッチをダウンロードしたディレクトリです。

   ```
   ./upgradeportalrpm
   ```

   update スクリプトによって RPM がインストールされます。

   ローカライズ版パッチの場合は、次のコマンドを使用して各 RPM をインストールします。

   ```
   rpm -Fvh patchName-version.rpm
   ```

7. パッチのアップグレードが成功したことを確認します。

   ```
   rpm -qa | grep sun-portal
   ```

   RPM のアップグレードリビジョン番号が返されることを確認します。

8. 手順 6 でローカライズ版パッケージをアップグレードした場合は、Portal Server コンソール JVM のロケールを UTF-8 に設定します。

   ```
   export LC_ALL=ja_JP.UTF-8
   export LANG=ja_JP.UTF-8
   ```

9. psupgrade スクリプトで必要な 2 つの環境変数 (ANT_HOME および JAVA_HOME) を設定します。

```
export ANT_HOME=/opt/sun
export JAVA_HOME=/usr/jdk/entsys-j2se
```

10. コンピュータ上に十分なスワップ空間があることを確認します。

    ガイドラインとして、スワップ空間を物理 RAM の容量の 2 倍に設定することをお勧めします。

11. psupgrade スクリプトを実行します。

    ```
    cd PortalServer7-base/bin
    ./psupgrade
    ```

    psupgrade スクリプトは Java ES Release 5 ディストリビューションから実行されず、Java ES インストーラも起動しません (パッケージにはすでにパッチが適用済み)。

    このスクリプトは、システムに照会して Portal Server Web アプリケーションの配備先の Web コンテナに関連した場所やポート番号、その他の情報を検出します。Web コンテナのアップグレードシナリオ (311 ページの表 15-4 を参照) によっては、スクリプトから、Portal Server を適切な Web コンテナに配備するために必要な追加情報の入力が要求されます。

    317 ページの表 15-6 は、311 ページの表 15-4 に記載されている Web コンテナの各種のアップグレードシナリオで要求される情報を示しています。

    注 psupgrade のパラメータには必ず正しい値を入力してください。あとで戻って変更することはできず、また psupgrade スクリプトによって加えられた変更のロールバックもきわめて困難です。

12. Web コンテナを停止し、再起動します。

    すべての状況で必要なわけではありませんが、Web コンテナを再起動すると、Portal Server がクリーンな状態で起動されるようになります。

アップグレードの確認

Portal Server パッケージの Release 5 へのパッチの適用は、次のコマンドを使用することにより確認できます。

```
PortalServer7-base/bin/psadmin --version --adminuser admin_ID
-f adminpasswordfile.
```

出力値については、315 ページの表 15-5 を参照してください。

完全アップグレードを確認するには、ポータルデスクトップが起動し、psadmin 管理ユーティリティーが説明のとおりに機能することを確認します。

また、/var/sadm/install/logs にある次のアップグレードログファイルもチェックできます。

- Sun_Java_System_Portal_Server_upgrade.log
- Sun_Java_System_Portal_Server_upgrade.log_ant_*xxx*.log

  ここで、*xxx* は preupgrade、upgrade、postupgrade のいずれかです。

IFR 7 のアップグレード後の作業

Portal Server を IFR 7 から Release 5 にアップグレードする場合は、次の状況に対処するために必要なアップグレード後の手順を実行してください。

- Java ES Monitoring Framework の有効化
- Application Server Web コンテナでのアップグレード
- Release 5 の Web Server (7.0) Web コンテナでのアップグレード
- カスタムポートレットアプリケーションの再配備
- カスタマイズされたポートレットアプリケーションの移行
- ブックマークの正しいリンクとアプリケーションチャネル
- struts ベースのポートレットの手動での移行

Java ES Monitoring Framework の有効化

Java ES Monitoring Framework (MFWK 共有コンポーネント) を有効にすると、Portal Server コンソール、psadmin コマンド行インタフェースなどの Portal Server 管理ツールが、訪問者の数や頻繁にアクセスされるポータルなどのメトリックスを報告できるようになります。MFWK を有効にするには、次の手順を実行します。

1. 次の 2 つのファイルの場所を確認します。

   MFWK-base/template/jesmf/desktopmfwk.properties
   MFWK-base/template/jesmf/com.sun.cmm.ps.xml

   ここで、*MFWK-base* のパスは次のとおりです。

   /opt/SUNWmfwk　(Solaris)

   /opt/sun/mfwk　(Linux)

2. この 2 つのファイルを次のディレクトリにコピーします。

   PortalServer7Data-base/portals/portal_ID/config/Portal_Instance/

3. desktopmfwk.properties ファイルで、

   com.sun.portal.ProductCollectionId=%PS_DIR%

   を次のように置き換えます。

```
com.sun.portal.ProductCollectionId=Portal_Installed_Location
```

4. 次の 2 つの jar ファイル

   MFWK-base/lib/mfwk_instrum_tk.jar
   MFWK-base/lib/mfwk_agent.jar

   を適切な Web コンテナクラスパスに追加します (Web Server の場合は server.xml ファイル、Application Server の場合は domain.xml ファイル)。

5. 対応する Web コンテナを再起動します。

Application Server Web コンテナでのアップグレード

Portal Server が Application Server Web コンテナに配備されている場合、次の追加の手順を実行して Portal Server を正常に再配備してください。

1. *AppServer8Config-base*/domains/*domainName*/config/server.policy ファイル内で、psconsole 設定を定義しているセクションを見つけます。
2. そのセクションの最後に次の行を追加します。

   ```
   permission java.lang.RuntimePermission "getProtectionDomain"
   ```

3. Application Server インスタンスを再起動します。

   AppServer8-base/bin/asadmin start-domain --user *admin_ID*
   --password *passworddomainName*

   AppServer8-base/bin/asadmin start-node-agent --user *admin_ID*
   --password *password nodeagentName*

   ここで、*nodeagentName* の形式は *hostName_domainName* ですが、デフォルトでは単に *hostName* です。

Application Server Web コンテナでの接続のハングアップ

アップグレードされた Portal Server が Application Server Web コンテナに配備されている場合は、ポータルアプリケーションが Java DB 接続の取得待ちでハングアップすることがあります。この問題に対処するには、次の手順を実行します。

1. *PortalServer7Data-base*/derby/derby.properties 内から、次の 2 つのパラメータの設定を削除します。

   ```
   derby.drda.maxThreads
   derby.drda.timeslice
   ```

2. Java DB を再起動します。

   ```
   ANT_HOME/bin/ant
       -DPS_CONFIG=PortalServer7Config-base/PSConfig.properties
       -buildfile PortalServer7-base/lib/derby.xml
       [stop-instance|start-instance]
   ```

ここで、ANT_HOME は、Solaris では /usr/sfw 、Linux では /opt/sun です。

3. Application Server 用の Java DB 設定を変更します。

   Application Server コンソールを使用して、communitymcPool、FileSharingDBPool、PointBasePool、SurveyDBPool の接続プールリソースの属性値を変更します。

   次の属性値を次のように変更します。

   ```
   Idle Timeout を 300 以上に
   Resource Type を javax.sql.ConnectionPoolDataSource に
   Datasource classname を
       org.apache.derby.jdbc.ClientConnectionPoolDataSource に
   ```

4. Portal Server が配備される Application Server インスタンスを再起動します。

Release 5 の Web Server (7.0) Web コンテナでのアップグレード

psupgrade スクリプトが Release 5 の Web Server 上の Portal Server IFR のアップグレードをサポートしないため (353 ページの「Portal Server IFR のアップグレードの概要」を参照)、まだ Release 5 にアップグレードされていない Web Server にある Portal Server IFR をアップグレードした場合は、次のアップグレード後の追加の手順を実行してください。

1. Web Server (および Access Manager) を Release 5 にアップグレードします。
2. Portal Server コンソールおよび psadmin コマンド行インタフェースで必要な Web Server コンテナ値を再構成します。

   a. LDAP ブラウザを開きます。

      設定値は Directory Server に格納されています。

   b. DN の下で次の行を探します。

      ```
      sunPortalAdminPortalDomainID=defaultDomain
      ->sunPortalAdminPortalDomainPortalID=portal1
      ->sunPortalAdminPortalDomainPortalServerInstanceIn=host-port
      ```

   c. 次のように編集を実行します。

      すべてのエントリが sunPortalAdminPortalDomainPortalServerInstance という文字列で始まっていることに注目します。

      - WebContainerInstanceDir のエントリを削除します。
      - WebContainerDomainName のエントリを追加し、317 ページの表 15-6 にある Web Container Config Name の値を割り当てます。
      - InstallDir、WebContainerType、DocRoot、317 ページの表 15-6 に示されているその他のパラメータなどのエントリを、Release 5 の Web Server (7.0) の値に対応するように編集します。

3. Release 5 の Portal Server インスタンスを作成します。

PortalServer7-base/bin/psadmin create-instance *newInstance_ID*

newInstance_ID の値がすでに存在する場合はエラーがスローされるため、この手順を次の手順 4 の前に実行することをお勧めします。

4. Portal Server IFR インスタンスを削除します。

   PortalServer7-base/bin/psadmin delete-instance *oldInstance_ID*

カスタムポートレットアプリケーションの再配備

カスタムポートレットアプリケーションを作成および配備している場合は、Release 5 の Portal Server にアップグレードしたあとに、これらのポートレットを手動で再配備してください。ディスプレイプロファイルエントリが存在してチャネル名が表示される場合でも、カスタムポートレットを再配備するまで内容は表示されません。

次のコマンドを使用して、ポートレットを再配備します。

PortalServer7-base/bin/psadmin deploy-portlet

次の場所で対応する .war および XML ファイルを探すことによって、再配備を確認できます。

PortalServer7Data-base/portals/Upgraded/war

カスタマイズされたポートレットアプリケーションの移行

SJWC (Sun Java Web Console) で提供されるユーザーインタフェースフレームワークに基づいたポートレットアプリケーションを手動で Release 5 に移行し、再配備する必要があります。

特に、この要件は、カスタマイズしてポータルにインストールすることを目的としたサンプルポートレットアプリケーションとして Portal Server とともに配布される 4 つの Web アプリケーション、filesharing、surveys、wiki、rssportlet に適用されます。アップグレード中、これらのサンプルポートレットアプリケーションのバグ修正バージョンは、ポートレットアプリケーション領域内のディスクに格納されます。これらのポートレットアプリケーションを独自に使用するためにカスタマイズしている場合は、手動で Release 5 に移行し、再配備する必要があります。Portal Server のアップグレードプロセスでは処理されません。

デフォルトでは、コミュニティーが作成されると、これらのポートレットアプリケーションの一部 (filesharing と surveys) が配備され、ユーザーに使用可能になります。

次の手順を使用すると、SJWC ベースのポートレットアプリケーションをアップグレード、カスタマイズ、および再配備できます。例として、ここでは filesharing ポートレットアプリケーションを使用します。

1. Release 5 の SJWC jar ファイルを展開します。
   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

PortalServer7-base/portlet/communityportlets.war が見つからない場合は、*PortalServer7-base*/portlet/core/communityportlets.war を使用します。

d. このうちの 1 つのファイルの名前を変更します。

```
mv WEB-INF/lib/commons-collections-3.1.jar
WEB-INF/lib/commons-collections.jar
```

2. filesharing ポートレットアプリケーションの場所を指定します。

```
cd PortalServer7Config-base/portals/portal1/portletapps/filesharing
```

3. 更新された SJWC ライブラリを挿入します。

```
jar uvf src/filesharing.war.tokenized -C /tmp/lh WEB-INF
```

4. filesharing ポートレットアプリケーションをカスタマイズします。

```
ant customize
```

5. filesharing ポートレットアプリケーションを再配備します。

a. *PortalServer7-base*/bin/psadmin undeploy-portlet -u *amadmin* -f passwordfile -p *portal_id* -i *instance_id* -g filesharing

b. ant deploy

c. Web コンテナに応じて、次のディレクトリに移動します。

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. sun-web.xml ファイルを開き、最後の行の直前 (つまり、sun-web-app 終了タグの前) に次の行を追加します。

```
<class-loader delegate="false"/>
```

e. filesharing/Web-INF の下にある sun-web.xml ファイルに対して、手順 c と手順 d を繰り返します。

6. surveys や、SJWC フレームワークに基づいたその他の任意のカスタムポートレットアプリケーションに対して、手順 2 ～手順 5 を繰り返します。

7. Web コンテナを再起動します。

ブックマークの正しいリンクとアプリケーションチャネル

Release 4 の Portal Server を Release 5 にアップグレードする場合、ブックマークとアプリケーションチャネルは重複した偽のリンクを持ちます。このリンクを修正するには、次の手順を実行します。

1. PSConsole にログインします。
2. 「共通タスク」タブから、「チャネルおよびコンテナを管理」をクリックします。
3. DN として「DeveloperSample [組織]」を選択し、「了解」をクリックします。
4. 「表示タイプ」で「 JSPTabContainer [デフォルト] 」を選択します。
5. 「MyFrontPageTabPanelContainer」にある「App チャネル」をクリックします。

   右側のフレームに App チャネルプロパティーが表示されます。

   特定のロケールのプロパティーを表示するには、「テーブルの設定変更」をクリックし、ロケール値を入力します (de、fr、es、ja、ko、zh、zh_CN、zh_TW)。
6. userApps プロパティーを編集します。
   a. userApps プロパティーで「値を編集 ...」リンクをクリックします。

      既存のアプリケーションを含むポップアップウィンドウが表示されます。

   b. 次のアプリケーションを一覧に追加します。

      NetFile

   c. 「保存」をクリックし、「閉じる」をクリックします。
7. targets プロパティーを編集します。
   a. targets プロパティーで「値を編集 ...」リンクをクリックします。

      既存のターゲットを含むポップアップウィンドウが表示されます。

   b. 次のターゲットを一覧に追加します。

      NetFile|/portal/NetFileApplet?Refer=java2

   c. 「保存」をクリックし、「閉じる」をクリックします。

struts ベースのポートレットの手動での移行

Struts フレームワークに基づいたコードを使用するカスタムポートレットがある場合は、Release 5 の Portal Server に含まれている `struts.jar` ファイルを使用するように、これらのポートレットを手動で更新する必要があります。次の手順に従ってください。

1. struts ベースのポートレットアプリケーションの配備を取り消します。

   PortalServer7-base`/bin psadmin undeploy-portlet`

2. `struts.jar` ファイルの正しいバージョンを使用して、`.war` ファイルを更新します。

   PortalServer7-base`/lib/struts.jar` を *strutsbasedPortlet*`/WEB-INF/lib/struts.jar` にコピーします。

   ここで、*strutsbasedPortlet* は、struts ベースのポートレットファイルが格納されているディレクトリです。

3. *strutsbasedPortlet* ディレクトリの `.war` アーカイブを作成します。

4. ポートレットアプリケーションを再配備します。

   PortalServer7-base`/bin psadmin deploy-portlet`

アップグレードのロールバック (Solaris)

ここでは、Portal Server のアップグレードのロールバック手順に影響する注意点を示したあと、その手順自体について説明します。

ロールバックの注意点 (Solaris)

Release 5 へのアップグレードをロールバックする手順は、*PortalServer7-base* の IFR インストールへの復元と、IFR Web アプリケーションの再配備で構成されます。

ロールバック手順 (Solaris)

1. root としてログインするか、スーパーユーザーになります。

   ```
   su -
   ```

2. Directory Server をアップグレード前の状態に復元します。

   Directory Server のバックアップ / 復元のコマンド行および GUI ユーティリティーを使用します。Sun Java System Directory Server Enterprise Edition 6.0 管理ガイド』(http://docs.sun.com/doc/820-0291?l=ja) の「Directory Server バックアップと復元」の章を参照してください。

3. Release 5 へのアップグレード中に再配備された、Release 5 の Portal Server Web アプリケーションの配備を取り消します。

Web コンテナの管理ユーティリティー (コマンド行またはコンソール) を使用して、次のパッケージの配備を取り消します。

```
portal
psconsole
search1
wsssoportlet
guessnumber
portletsamples
```

4. Web コンテナを停止することにより、Portal Server を停止します。

   Web Server 6.*x:*
   WebServer-base/https-*instanceName*/stop

   Web Server 7.0*:*
   Admin Server--
   WebServer7Config-base/admin-server/bin/stopserv
   Instance Server--
   WebServer7Config-base/https-*configName*/bin/stopserv

   Application Server *8.x:*
   AppServer8-base/bin/asadmin stop-domain --user *admin_ID*
   --password *password domainName*

5. 表 15-8 の Portal Server 7.1 パッチをバックアウトします。

   patchrm *patch_ID*

6. Web コンテナを再起動することにより、Portal Server を再起動します。

   Web Server 6.*x:*
   WebServer-base/https-*instanceName*/start

   Web Server 7.0*:*
   Admin Server--
   WebServer7Config-base/admin-server/bin/startserv
   Instance Server--
   WebServer7Config-base/https-*configName*/bin/startserv

   Application Server *8.x:*
   AppServer8-base/bin/asadmin start-domain --user *admin_ID*
   --password *password domainName*

7. 367 ページの手順 3 で配備を取り消しされた Release 5 の Portal Server Web アプリケーションを配備します。

   Web コンテナの管理ユーティリティー (コマンド行またはコンソール) を使用して、パッケージを配備します。

8. Web コンテナを停止し、再起動します。

すべての状況で必要なわけではありませんが、Web コンテナを再起動すると、Portal Server がクリーンな状態で起動されるようになります。

アップグレードのロールバック (Linux)

アップグレードのロールバックは Linux では実行できません。

ただし、アップグレードおよびアップグレード前のシステムテストを実行する前に、並列システムを作成することができます。アップグレードのロールバックが必要になった場合は、その並列システムに戻すことができます。

複数インスタンスのアップグレード

配備のアーキテクチャーによっては、スケーラビリティーを実現し、可用性を向上させるために、Portal Server が複数のコンピュータシステムに配備される場合があります。たとえば、ロードバランサを使用して Portal Server コンポーネントを複数のコンピュータで実行することにより、負荷を分散させることができます。

Portal Server のインスタンスが負荷分散されている場合は、サービスを中断することなく連続して Portal Server インスタンスをアップグレードする順次アップグレードを実行できます。Portal Server の各インスタンスを、ほかのインスタンスが稼働している間にアップグレードします。IFR から Release 5 への順次アップグレードを実行するには、「Release 4」を「IFR」と読み替えた上で、337 ページの「複数インスタンスのアップグレード」の手順を使用してください。

第 16 章

Portal Server Secure Remote Access

この章では、Portal Server Secure Remote Access を Java ES 5 (Release 5): Sun Java System Portal Server Secure Remote Access 7.1 にアップグレードする方法について説明します。

ここでは、Release 5 でサポートされている各種のアップグレードパスでのアップグレードの注意点の概要について説明します。また、Solaris オペレーティングシステムと Linux オペレーティングシステムの両方でのアップグレードを取り上げます。

- 373 ページの「Portal Server Secure Remote Access のアップグレードの概要」
- 378 ページの「Java ES Release 4 からの Portal Server Secure Remote Access のアップグレード」
- 396 ページの「Java ES Release 3 からの Portal Server Secure Remote Access のアップグレード」
- 398 ページの「Java ES Release 2 からの Portal Server Secure Remote Access のアップグレード」
- 401 ページの「Portal Server Secure Remote Access の IFR 7.0 からのアップグレード」

注 この章では、*PortalServer6-base* と *PortalServer6Config-base* (Portal Server 6.*x* の場合)、および *PortalServer7-base* と *PortalServer7Config-base* (Portal Server 7.*x* の場合) というディレクトリパスを基準に、ファイルの場所が指定されています。これらのパスの少なくとも一部は、Portal Server が最初にインストールされたときにインストールディレクトリとして指定されている可能性があります。指定されていない場合、Java ES インストーラによってデフォルト値が割り当てられています。

次の表に、これらのディレクトリパスのデフォルト値を示します。

表 16-1　Portal Server Secure Remote Access ディレクトリパス

パス名変数	Solaris OS	Linux OS
PortalServer6-*base*	/opt/SUNWps	/opt/sun/portal
PortalServer6Config-*base*	/etc/opt/SUNWps	/etc/opt/sun/portal
PortalServer6Data-*base*	/var/opt/SUNWps	/var/opt/sun/portal
ortalServer7-*base*	/opt/SUNWportal	/opt/sun/portal
PortalServer7Config-*base*	/etc/opt/SUNWportal	/etc/opt/sun/portal
PortalServer7Data-*base*	/var/opt/SUNWportal	/var/opt/sun/portal

Portal Server Secure Remote Access のアップグレードの概要

ここでは、Java ES 5 (Release 5) へのアップグレードに影響する Portal Server Secure Remote Access の一般的な側面について説明します。

- Java ES Release 5 の Portal Server Secure Remote Access について
- Portal Server Secure Remote Access のアップグレードロードマップ
- Portal Server Secure Remote Access のデータ
- Portal Server Secure Remote Access のアップグレード方針

Java ES Release 5 の Portal Server Secure Remote Access について

Portal Server Secure Remote Access (ゲートウェイ、Rewriter プロキシ、Netlet プロキシの各コンポーネントで構成される) は、通常は Portal Server をホストしているコンピュータとは別のコンピュータに配備されますが、Portal Server と緊密に結合されています。Portal Server Secure Remote Access コンポーネントは正式な Portal Server と同じ管理インフラストラクチャーを使用し、Portal Server をホストしているコンピュータ上に存在するサーブレットやアプレットと相互に作用します。

Java ES Release 5 の Portal Server Secure Remote Access は Release 4 に対するメジャーリリースであり、多くの拡張機能や新機能が追加されています。これらの変更の多くは、Release 4 以降の暫定機能リリース (Interim Feature Releases、IFR) で加えられました。Release 5 では、IFR に対する若干の機能変更だけが行われています。IFR の拡張機能および新機能については、『Sun Java System Portal Server 7.1 リリースノート (UNIX 版)』(http://docs.sun.com/app/docs/doc/820-0900/6nck07u85?l=ja&a=view) を参照してください。特に、Release 4 のコマンド行管理インタフェースは `psadmin` コマンドに置き換えられています。

Portal Server Secure Remote Access のアップグレードロードマップ

表 16-2 は、Portal Server Secure Remote Access を Java ES Release 5 へアップグレードする際の、サポートされるアップグレードパスを示しています。この表は Solaris および Linux の両方のオペレーティングシステムに適用されます。

表 16-2　Java ES 5 (Release 5): Portal Server Secure Remote Access 7.1 へのアップグレードパス

Java ES のリリース	Portal Server Secure Remote Access バージョン	一般的な手順	必要な再設定
暫定機能リリース (IFR)	Sun Java System Portal Server Secure Remote Access IFR 7.0 2005Q4	直接アップグレード: パッチを適用してアップグレードスクリプトを使用することにより、実行されます。	なし。
Release 4	Sun Java System Portal Server Secure Remote Access 6.3.1 2005Q4	直接アップグレード: アップグレードスクリプトを使用して行います。	なし。
Release 3	Sun Java System Portal Server Secure Remote Access 6.3.1 2005Q1	直接アップグレード: アップグレードスクリプトを使用して行います。	なし。
Release 2	Sun Java System Portal Server Secure Remote Access 6.3 2004Q2	直接アップグレード: アップグレードスクリプトを使用して行います。	なし。
Release 1	Sun ONE Portal Server Secure Remote Access 6.2 (2003Q4)	直接アップグレード不可: ただし、まず Release 3 にアップグレードし、次に Release 3 から Release 5 にアップグレードすることにより実行できます。	設定データ
それ以前の Java ES リリース		直接アップグレード不可。	

Portal Server Secure Remote Access のデータ

次の表は、Portal Server Secure Remote Access ソフトウェアのアップグレードにより影響を受ける可能性のあるデータの種類を示しています。

表 16-3 Portal Server Secure Remote Access のデータの使用法

データの種類	場所	使用法
設定データ	*PortalServer6Config-base/*	Portal Server Secure Remote Access の設定
ディレクトリスキーマ サービスの設定 ユーザーデータ	Directory Server	Portal Server Secure Remote Access は、ポータルデスクトップなどのサービスの設定、およびディレクトリに格納されているユーザープロファイルデータに依存します。
動的なアプリケーションデータ	なし	Portal Server Secure Remote Access は、セッション状態などのアプリケーションデータを永続的に格納しません。

Portal Server Secure Remote Access のアップグレード方針

Portal Server Secure Remote Access をアップグレードするための方針は一般に、第 1 章「アップグレードの計画」で説明されている多くの考慮事項 (アップグレードパス、Java ES コンポーネント間の依存関係、選択的なアップグレードかすべてのアップグレードか、複数インスタンスの配備など) によって異なります。

ここでは、Portal Server Secure Remote Access のアップグレード計画に影響する可能性のある問題を取り上げ、Portal Server Secure Remote Access の全般的な説明を詳細に記述します。

互換性の問題

Release 5 の Portal Server Secure Remote Access では、ゲートウェイ、Rewriter プロキシ、および Netlet プロキシコンポーネントの起動と停止に使用される `psadmin` コマンドに、公開インタフェースの変更が導入されました。『Sun Java System Portal Server 7.1 Command-Line Reference』(`http://docs.sun.com/doc/819-5030`) を参照してください。

個々の Portal Server Secure Remote Access コンポーネント (ゲートウェイ、Rewriter プロキシ、および Netlet プロキシを含む) は、以前のバージョンとは下位互換性がありません。Portal Server 自体とともに、すべてを Java ES Release 5 で同期する必要があります。この要件は、ローカルおよび配布された Portal Server Secure Remote Access コンポーネントに適用されます。

さらに、Release 5 の Portal Server で使用される Directory Server データ構造と以前のバージョンの Portal Server で使用される Directory Server データ構造の間にも互換性がありません。この非互換性は、同じ Directory Server データを使用した複数の Portal Server インスタンスの順次アップグレードに影響を与えます。

Portal Server Secure Remote Access の依存関係

Portal Server Secure Remote Access は、Portal Server と緊密に結合されていて、Portal Server とパッケージ化されて Portal Server と同じコンピュータ上で動作しているソフトウェアに依存しています。

ただし、Portal Server Secure Remote Access は、ほかの Java ES コンポーネントにも依存します。これらの依存関係は、Portal Server Secure Remote Access ソフトウェアのアップグレードおよび再設定の手順に影響を及ぼす可能性があります。たとえば、Portal Server Secure Remote Access のインタフェースまたは関数が変更された場合は、Portal Server Secure Remote Access が依存するコンポーネントのアップグレードされたバージョンが必要になる場合があります。このようなコンポーネントのアップグレードの必要性は、個々のアップグレードパスによって異なります。

Portal Server Secure Remote Access コンポーネントは、次の Java ES コンポーネントに依存します。

- **共有コンポーネント** : Portal Server Secure Remote Access コンポーネントは、特定の Java ES 共有コンポーネントに依存します (48 ページの表 1-9 を参照)。
- Portal Server: Portal Server Secure Remote Access コンポーネントは、Portal Server Secure Remote Access 機能をサポートするのに必要なローカルコンポーネントを含む Portal Server に必ず依存しています。
- Access Manager (または Access Manager SDK): Portal Server Secure Remote Access コンポーネントは、シングルサインオンを含む、エンドユーザーへの認証および承認サービスを提供するために Access Manager に必ず依存しています。Access Manager をリモートコンピュータ上で動作させる場合、Access Manager SDK がローカルで利用可能である必要があります。
- Directory Server: Portal Server Secure Remote Access は、ユーザーデータを格納する Directory Server に必ず依存しています。そのため、Portal Server Secure Remote Access のアップグレードによって、ディレクトリスキーマの拡張が必要になる場合があります。

選択的なアップグレードに関する問題

Java ES Release 5 は一般に、コンピュータ上のすべてのコンポーネントの選択的なアップグレードをサポートしていますが、Portal Server Secure Remote Access が Portal Server に緊密に結合されているので、Portal Server をアップグレードした場合は Portal Server Secure Remote Access もアップグレードしてください。同様に、Portal Server Secure Remote Access をアップグレードした場合は、Portal Server もアップグレードする必要があります。

その結果、Portal Server Secure Remote Access のアップグレードは、Portal Server と同じ制限 (310 ページの「選択的なアップグレードに関する問題」の Portal Server を参照) に制約されます。つまり、Portal Server Secure Remote Access とその製品コンポーネントのすべての依存関係を Release 5 にアップグレードすることも、Portal Server Secure Remote Access と Portal Server のみを Release 5 にアップグレードして、ほかの製品コンポーネントの依存関係を Release 4 のままにすることもできます。

二重のアップグレード

44 ページの「二重のアップグレード : Java ES ソフトウェアとオペレーティングシステムソフトウェア」で説明されているように、二重のアップグレードでは Portal Server Secure Remote Access とオペレーティングシステムの両方をアップグレードします。次のようなインプレースオペレーティングシステムアップグレードを使用して二重のアップグレードを実行できます。

1. 既存の Portal Server Secure Remote Access データをバックアップします。

   基本的なデータの場所については、375 ページの「Portal Server Secure Remote Access のデータ」を参照してください。

2. オペレーティングシステムをアップグレードします。

   このアップグレードでは、既存のファイルシステムはそのまま残されます。

3. Release 5 の Portal Server Secure Remote Access にアップグレードします。

   アップグレードパスに応じて、この章の該当する節を参照してください。

Java ES Release 4 からの Portal Server Secure Remote Access のアップグレード

ここでは、Portal Server Secure Remote Access の Java ES 2005Q4 (Release 4) から Java ES 5 (Release 5) へのアップグレードについて説明します。

この節で説明する内容は、次のとおりです。

- はじめに
- Release 4 の Portal Server Secure Remote Access のアップグレード
- 複数インスタンスのアップグレード

はじめに

Java ES Release 4 Portal Server Secure Remote Access を Release 5 にアップグレードする場合は、アップグレード処理の次の側面を考慮してください。

- **一般的なアップグレード手順** : このアップグレードは、アップグレードスクリプト `psupgrade` を使用することにより実行されます。このスクリプトは古いパッケージを削除して新しいパッケージをインストールし、さらに必要に応じて設定データを移行します。
- **アップグレードの依存関係** : Portal Server は、いくつかの Java ES 共有コンポーネントに依存します (48 ページの表 1-9 を参照)。Release 5 の Portal Server Secure Remote Access はこれらの共有コンポーネントの Release 4 バージョンと互換性がありますが、Portal Server Secure Remote Access のアップグレードに使用される `psupgrade` スクリプトには ANT 共有コンポーネントの Release 5 バージョンが必要なため、共有コンポーネントのアップグレードはいずれにしても必要です。

  また、376 ページの「Portal Server Secure Remote Access の依存関係」で説明されているように、Release 5 の Portal Server Secure Remote Access は Portal Server、Access Manager、および Directory Server に対して依存関係を持っています。これらの依存関係のアップグレードに対して、次の 2 つの方法がサポートされています (377 ページの「選択的なアップグレードに関する問題」を参照)。
- **アップグレードの依存関係** : Portal Server Secure Remote Access はいくつかの Java ES 共有コンポーネントに依存しますが (48 ページの表 1-9 を参照)、Release 5 の Portal Server Secure Remote Access はこれらのコンポーネントの Release 4 バージョンと互換性があります。したがって、これらの共有コンポーネントのアップグレードは、Portal Server Secure Remote Access の Release 5 へのアップグレードに関しては省略可能です。

ただし、Release 5 の Portal Server Secure Remote Access は、Portal Server に対してのみ強いアップグレード依存関係を持っています。また、376 ページの「Portal Server Secure Remote Access の依存関係」で説明されているように、Release 5 の Portal Server Secure Remote Access は Portal Server、Access Manager、および Directory Server に対して依存関係を持っています。

これらの製品コンポーネントの依存関係のアップグレードに対して、次の 2 つの方法がサポートされています (377 ページの「選択的なアップグレードに関する問題」を参照)。

- すべての依存関係が Release 4 コンポーネントによって満足される (Portal Server を除き、何も Release 5 にアップグレードされない)。
- すべての依存関係が Release 5 コンポーネントによって満足される (すべてが Release 5 にアップグレードされる)。

Portal Server Secure Remote Access に対して採用する方法は、Portal Server で採用する方法と同じにしてください。

- **下位互換性** : Release 5 の Portal Server Secure Remote Access は、Release 4 バージョンと下位互換性があります。
- **アップグレードのロールバック** : Portal Server Secure Remote Access の Release 5 アップグレードの Release 4 へのロールバックは、Release 4 パッケージの復元と Release 4 ディレクトリデータの復元で構成されます。
- **プラットフォームの問題** : Portal Server Secure Remote Access をアップグレードするための一般的な手順は Solaris および Linux の両オペレーティングシステムで共通ですが、Release 5 の Portal Server Secure Remote Access が、Solaris オペレーティングシステムでは新しいパスにインストールされるのに対して、Linux オペレーティングシステムでは Release 4 と同じパスにインストールされます。

Release 4 の Portal Server Secure Remote Access のアップグレード

ここでは、Solaris プラットフォームと Linux プラットフォームの両方で、Portal Server Secure Remote Access の Java ES Release 4 から Java ES Release 5 へのアップグレードを実行する方法について説明します。プラットフォーム固有の手順に依存する項目には、その手順が適用されるオペレーティングシステムが示されています。この節で説明する内容は、次のとおりです。

- Release 4 のアップグレード前の作業
- Release 4 の Portal Server Secure Remote Access のアップグレード (Solaris)
- Release 4 の Portal Server Secure Remote Access のアップグレード (Linux)
- アップグレードの確認

- Release 4 のアップグレード後の作業
- アップグレードのロールバック (Solaris)
- アップグレードのロールバック (Linux)

Release 4 のアップグレード前の作業

Portal Server Secure Remote Access をアップグレードする前に、次の作業を実行してください。

- 現在のバージョン情報の確認
- Portal Server Secure Remote Access が依存するコンポーネントのアップグレード
- Release 4 Portal Server Secure Remote Access の設定情報のバックアップ
- ロードバランサの設定の削除
- Directory Proxy Server の設定の削除
- 必要な設定情報とパスワードの入手

現在のバージョン情報の確認

Portal Server Secure Remote Access の現在のバージョンは、次のコマンドを使用することにより確認できます。

PortalServer6-base/bin/version

表 16-4　Portal Server Secure Remote Access のバージョン確認の出力

Java ES のリリース	Portal Server Secure Remote Access のバージョン番号
Release 2	6.3
Release 3	6.3.1
Release 4	6.3.1[1]
IFR リリース	7.0
Release 5	7.1

1. Release 3 と Release 4 の違いはパッチだけです。Release 4 のパッチは、Solaris の showrev -p | grep *patch_ID* コマンドおよび Linux の rpm -qa sun-portal-core コマンドを使用し、これらのバージョンを Java ES Release 4 の『アップグレードガイド』に示されているバージョンと比較することにより確認できます。

Portal Server Secure Remote Access が依存するコンポーネントのアップグレード

一般的には、コンピュータシステム上 (および、コンピューティング環境内) のすべての Java ES コンポーネントを Java ES Release 5 にアップグレードすることが推奨されます。

Release 5 の Portal Server Secure Remote Access は Java ES 共有コンポーネントの Release 4 バージョンと互換性がありますが、Portal Server Secure Remote Access のアップグレードに使用される `psupgrade` スクリプトには ANT 共有コンポーネントの Release 5 バージョンが必要なため、共有コンポーネントのアップグレードはいずれにしても必要です。

さらに、Portal Server Secure Remote Access には Portal Server のアップグレードも必要です。ただし、依存先のほかの Java ES Release 4 製品コンポーネントをアップグレードする必要はありません。

つまり、依存関係のアップグレード方法は、Portal Server に対して採用する方法と同じです。いずれかの依存関係を Release 5 にアップグレードする場合は、すべてをアップグレードする必要があります (310 ページの「選択的なアップグレードに関する問題」を参照)。ただし、Portal Server Secure Remote Access の Portal Server に対する依存関係のため、Portal Server のアップグレードでは、共有コンポーネントを除き、Portal Server Secure Remote Access の依存関係を考慮に入れてください。

Portal Server Secure Remote Access の依存関係を Release 5 にアップグレードする場合は、Portal Server Secure Remote Access をアップグレードする前に、(すでにアップグレードした可能性があるものを省いて) 依存関係を次の順序でアップグレードするようにしてください。

1. **共有コンポーネント** : Java ES 共有コンポーネントを Release 5 に同期する手順は、67 ページの「Java ES 共有コンポーネントのアップグレード」で説明されています。
2. Portal Server: Portal Server をアップグレードする手順は、305 ページの第 15 章「Portal Server」で説明されています。

Release 4 Portal Server Secure Remote Access の設定情報のバックアップ

Portal Server Secure Remote Access の Release 5 へのアップグレードでは、Portal Server Secure Remote Access ソフトウェアの再設定は必要ありません。ただし、安全策として、設定情報が格納されている次のディレクトリは `psupgrade` スクリプトによってバックアップされます。

PortalServer6Config-base/

ロードバランサの設定の削除

Portal Server Secure Remote Access インスタンスがロードバランサを介してアクセスされる場合は、このようなアクセスの設定に使用される LOAD_BALANCER_URL プロパティーの値によって Portal Server Secure Remote Access のアップグレードが妨げられる場合があります。そのため、Portal Server Secure Remote Access コンポーネントのアップグレードを実行する前に、この設定を変更してください。LOAD_BALANCER_URL プロパティー設定を変更するには、次の手順を実行します。

1. 次のどの設定ファイルがローカルに存在するかを確認してください (これらの設定ファイルの一部は、ローカルにインストールされる可能性のある Portal Server コンポーネントをサポートする)。

   PortalServer6Config-base/PSConfig.properties (Portal Server がローカルの場合)
   PortalServer6Config-base/GWConfig.properties (ゲートウェイがローカルの場合)
   PortalServer6Config-base/RWPConfig.properties (Rewriter プロキシがローカルの場合)
   PortalServer6Config-base/NLPConfig.properties (Netlet プロキシがローカルの場合)

2. これらの設定ファイル内の LOAD_BALANCER_URL プロパティーの現在の値を記録します。

3. アップグレード対象の対応する Portal Server Secure Remote Access インスタンスをポイントするように、LOAD_BALANCER_URL プロパティーの値を変更します。

   LOAD_BALANCER_URL=*hostName*:*port*/portal

4. 下に示すように、次の設定プロパティー (存在する場合) が、ロードバランサではなく、関連する Portal Server Secure Remote Access コンポーネントを参照している必要があります。

   PortalServer6Config-base/platform.conf.default ファイル内 :

   gateway.host=*Gateway_hostName*

   PortalServer6Config-base/GWConfig.properties および
   PortalServer6Config-base/GWConfig-default.properties ファイル内 :

   GW_HOST=*Gateway_hostName*
   GW_IP=*Gateway_hostIP*

   PortalServer6Config-base/RWPConfig.properties および
   PortalServer6Config-base/RWPConfig-default.properties ファイル内 :

   RWP_HOST=*RewriterProxy_hostName*
   RWP_IP=*RewriterProxy_hostIP*

   PortalServer6Config-base/NLPConfig.properties および
   PortalServer6Config-base/NLPConfig-default.properties ファイル内 :

   NLP_HOST=*NetLetProxy_hostName*
   NLP_IP=*NetLetProxy_hostIP*

Directory Proxy Server の設定の削除

Portal Server Secure Remote Access インスタンスが Directory Proxy Server インスタンスを介して Directory Server にアクセスしている場合は、アップグレードを実行する前に Directory Proxy Server のホストおよびポート番号の設定を変更し、アップグレードが完了したあとに元の値に戻す必要があります。

該当する設定を変更するには、次の手順を実行します。

1. 次の Access Manager 設定ファイル内の `DS_HOST` および `DS_PORT` プロパティーの現在の値を記録します。

   AccessManagerConfig-base`/config/AMConfig.properties`

2. 関連する Directory Server インスタンスを直接ポイントするように、`DS_HOST` および `DS_PORT` プロパティーの値を変更します。

必要な設定情報とパスワードの入手

アップグレードシナリオによっては、`psupgrade` スクリプトから、次の管理アカウントに関する情報を入力するように要求されます。

- Directory Server の管理 ID とパスワード
- Access Manager の管理 ID とパスワード
- Directory Server の amldapuser ID とパスワード

Release 4 の Portal Server Secure Remote Access のアップグレード (Solaris)

ここでは、Portal Server Secure Remote Access のアップグレード手順に影響する注意点を示したあと、その手順自体について説明します。

アップグレードの注意点 (Solaris)

Portal Server Secure Remote Access ソフトウェアの Release 5 へのアップグレードでは、次の注意点を考慮に入れてください。

- Portal Server Secure Remote Access ソフトウェアは、いくつかの異なる役割を実行するサブコンポーネントで構成されますが、すべてを同時に Release 5 にアップグレードしてください。
  - Portal-base: 管理 Mbean と付随する管理ソフトウェア、ロギングフレームワーク、および監視関連のソフトウェアが含まれ、これらすべてが SUNWportal-base パッケージにパッケージ化されています。
  - Secure Remote Access アプリケーション : ゲートウェイ、Rewriter プロキシ、および Netlet プロキシが含まれます。これらのアプリケーションは通常、正式な Portal Server をホストしているコンピュータとは別の 1 つ以上のコンピュータに配備されます。Secure Remote Access アプリケーションには、Web コンテナは必要ありません。

- ゲートウェイ、Rewriter プロキシ、および Netlet プロキシが同じコンピュータに配備されていない場合は、ゲートウェイをアップグレードする前に、Rewriter プロキシと Netlet プロキシをアップグレードするようにしてください。
- すべての Portal Server Secure Remote Access サブコンポーネントが同じ Portal Server Secure Remote Access インストールイメージに対応し、アップグレード対象のコンピュータ上に存在する場合は、同時にアップグレードされます。
- psupgrade スクリプトは、ホストコンピュータにインストールされている Portal Server Secure Remote Access サブコンポーネントを自動的に検出し、それらのコンポーネントをアップグレードします。

アップグレード手順 (Solaris)

ここで説明する手順は、アップグレード対象のコンピュータ上の Portal Server Secure Remote Access コンポーネントに適用されます。

1. root としてログインするか、スーパーユーザーになります。

   ```
   su -
   ```

2. すべての共有コンポーネントの Release 5 への同期をまだ実行していない場合は、実行します。

   手順は 67 ページの第 2 章「Java ES 共有コンポーネントのアップグレード」で説明されています。

   この手順は、385 ページの手順 8 で psupgrade スクリプトを実行するために必要な前提条件です。

3. ローカルで実行されているゲートウェイ、Rewriter プロキシ、または Netlet プロキシのインスタンスをすべて停止します。

   PortalServer6-base/bin gateway stop
   PortalServer6-base/bin netletd stop
   PortalServer6-base/bin rwproxyd stop

   これらのプロセスが停止したことを確認します。

   ゲートウェイ : netstat -an | grep 443
   Rewriter プロキシ : netstat -an | grep 10443
   Netlet プロキシ : netstat -an | grep 10555

4. Access Manager が稼働していることを確認します。
5. psupgrade スクリプトで必要な 2 つの環境変数 (ANT_HOME および JAVA_HOME) を設定します。次に例を示します。

   ```
   export ANT_HOME=/usr/sfw
   export JAVA_HOME=/usr/jdk/entsys-j2se
   ```

6. コンピュータ上に十分なスワップ空間があることを確認します。

ガイドラインとして、スワップ空間を物理 RAM の容量の 2 倍に設定することをお勧めします。

7. アップグレード対象の Portal Server Secure Remote Access コンポーネントが Portal Server からリモートにある場合は、dpadmin 実行可能ファイルを、Portal Server をホストしているコンピュータから Portal Server Secure Remote Access コンポーネントをホストしているコンピュータにコピーします。

   dpadmin 実行可能ファイルは、次の場所に格納されています。

   Portal Server がアップグレードされている場合は、
   PortalServer7-base/SUNWps.bak/bin/dpadmin。

   Portal Server がまだアップグレードされていない場合は、
   PortalServer6-base/bin/dpadmin。

8. Java ES Release 5 ディストリビューションから psupgrade スクリプトを実行します。

   ```
   cd os_arch/Products/portal_svr/Tools/upgrade/bin
   ./psupgrade
   ```

   ここで、*os_arch* は、Solaris_sparc などのプラットフォームに対応します。

   psupgrade スクリプトは、Java ES インストーラを起動して新しいパッケージをインストールし、次の情報を要求します。

   - Directory Server の管理 ID とパスワード
   - Access Manager の管理 ID とパスワード
   - Directory Server の amldapuser ID とパスワード

9. 384 ページの手順 3 で停止したゲートウェイ、Rewriter プロキシ、または Netlet プロキシのインスタンスを起動します。

   ```
   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
   ```

   上記のコマンドが失敗する場合は、最初に Portal Server Secure Remote Access コンポーネントを登録して (有効にして) ください。

   ```
   PortalServer7-base/bin/psadmin provision-sra -u amadminUser
       -f passwordFile -p Portal_ID --gateway-profile profileName --enable
   ```

Release 4 の Portal Server Secure Remote Access のアップグレード (Linux)

ここでは、Portal Server Secure Remote Access のアップグレード手順に影響する注意点を示したあと、その手順自体について説明します。

アップグレードの注意点 (Linux)

Linux プラットフォームで Portal Server Secure Remote Access ソフトウェアを Release 5 にアップグレードするときは、Linux オペレーティングシステムでは Release 5 の Portal Server Secure Remote Access が Release 4 と同じパスにインストールされる点を除き、Solaris プラットフォームのときと同じ注意点 (383 ページの「アップグレードの注意点 (Solaris)」を参照) を考慮に入れてください。その結果、Release 5 の RPM のインストール時に、`psupgrade` スクリプトによって以前の RPM が削除されます。

アップグレード手順 (Linux)

次に示す手順は、アップグレードを実行するコンピュータ上の Portal Server Secure Remote Access に適用されます。

注意 Linux での Java ES Release 4 から Release 5 へのアップグレードはロールバックできません。次の手順を実行する前に、必ずシステムをバックアップしてください。

1. root としてログインするか、スーパーユーザーになります。

   ```
   su -
   ```

2. すべての共有コンポーネントの Release 5 への同期をまだ実行していない場合は、実行します。

   手順は 67 ページの第 2 章「Java ES 共有コンポーネントのアップグレード」で説明されています。

   この手順は、387 ページの手順 8 で `psupgrade` スクリプトを実行するために必要な前提条件です。

3. ローカルで実行されているゲートウェイ、Rewriter プロキシ、または Netlet プロキシのインスタンスをすべて停止します。

   PortalServer6-base`/bin gateway stop`
   PortalServer6-base`/bin netletd stop`
   PortalServer6-base`/bin rwproxyd stop`

   これらのプロセスが停止したことを確認します。

ゲートウェイ : `netstat -an | grep 443`
Rewriter プロキシ : `netstat -an | grep 10443`
Netlet プロキシ : `netstat -an | grep 10555`

4. Access Manager が稼働していることを確認します。
5. `psupgrade` スクリプトで必要な 2 つの環境変数 (`ANT_HOME` および `JAVA_HOME`) を設定します。次に例を示します。

   ```
   export ANT_HOME=/opt/sun
   export JAVA_HOME=/usr/jdk/entsys-j2se
   ```

6. コンピュータ上に十分なスワップ空間があることを確認します。

   ガイドラインとして、スワップ空間を物理 RAM の容量の 2 倍に設定することをお勧めします。

7. アップグレード対象の Portal Server Secure Remote Access コンポーネントが Portal Server からリモートにある場合は、`dpadmin` 実行可能ファイルを、Portal Server をホストしているコンピュータから Portal Server Secure Remote Access コンポーネントをホストしているコンピュータにコピーします。

   dpadmin 実行可能ファイルは、次の場所に格納されています。

   Portal Server がアップグレードされている場合は、
   PortalServer7-base`/SUNWps.bak/bin/dpadmin`。

   Portal Server がまだアップグレードされていない場合は、
   PortalServer6-base`/bin/dpadmin`。

8. Java ES Release 5 ディストリビューションから `psupgrade` スクリプトを実行します。

   ```
   cd os_arch/Products/portal_svr/Tools/upgrade/bin
   ./psupgrade
   ```

   ここで、*os_arch* は、`Solaris_sparc` などのプラットフォームに対応します。

   `psupgrade` スクリプトは、Java ES インストーラを起動して新しいパッケージをインストールし、次の情報を要求します。

   - Directory Server の管理 ID とパスワード
   - Access Manager の管理 ID とパスワード
   - Directory Server の amldapuser ID とパスワード

アップグレードの確認

アップグレード対象の Portal Server Secure Remote Access コンポーネントが Portal Server からリモートにある場合は、次のファイル内のバージョン情報をチェックすることにより Release 5 パッケージのインストールを確認できます。

PortalServer7-base/lib/PSversion.properties

ただし、アップグレード対象の Portal Server Secure Remote Access コンポーネントが Portal Server と同じコンピュータ上に存在する場合は、次のコマンドを使用してアップグレードを確認できます。

PortalServer7-base/bin/psadmin --version --adminuser *admin_ID* -f *adminpasswordfile*.

出力値については、380 ページの表 16-4 を参照してください。

また、次の場所にあるアップグレードログファイルを確認することもできます。

/var/sadm/install/logs/Sun_Java_System_Portal_Server_upagrede.log

Release 4 のアップグレード後の作業

Portal Server Secure Remote Access を Release 5 にアップグレードした場合は、次の状況を除き、アップグレード後の作業は必要ありません。

- ロードバランサの設定の復元
- Directory Proxy Server の設定の復元
- Release 4 のローカライズされたプロバイダの削除

ロードバランサの設定の復元

Portal Server Secure Remote Access インスタンスがロードバランサを介してアクセスされている場合は、アップグレードのあと、次の手順を実行してロードバランサ設定を復元する必要があります。

1. *PortalServer7Config-base*/platform.conf.default ファイル内の次のパラメータを設定します。

   gateway.virtualhost=*loadBalancer_hostName loadBalancer_hostIP*
   gateway.external.ip=*loadBalancer_hostIP*
   gateway.dsame.agent=http\://*loadBalancer_hostName*\:
   80/portal/RemoteConfigServlet

2. *PortalServer7Config-base*/GWConfig-default.properties ファイル内の次のパラメータを設定します。

   gateway.ipaddress=*Gateway_hostIP*

3. Rewriter プロキシおよび Netlet プロキシのインスタンスが、Portal Server ホストからリモートにあるコンピュータに配備されている場合は、手順 1 と手順 2 に対応するこれらのプロキシのパラメータを設定します。
4. Portal Server および負荷分散された Portal Server Secure Remote Access インスタンスを再起動します。

Directory Proxy Server の設定の復元

Portal Server Secure Remote Access インスタンスが Directory Proxy Server インスタンスを介して Directory Server にアクセスしていた場合は、Directory Proxy Server のホストおよびポート番号の設定を、アップグレード前の元の値に戻してください。アップグレードの準備でこれらのプロパティーの値を変更した手順、383 ページの「Directory Proxy Server の設定の削除」を参照してください。

Release 4 のローカライズされたプロバイダの削除

ローカライズされたプロキシレットサービスは、Release 4 のローカライズされたプロバイダを次の手順に従って削除するまでロードしません。

1. *PortalServer7Data-base*/portals/Upgraded/desktop ディレクトリに移動します。
2. default_*Locale* から、次のものを除くすべてのディレクトリとファイルを削除します。
   - Portal Server Secure Remote Access には同梱されていない、ユーザーが作成したファイルとディレクトリ
   - message.properties ファイル
   - 次のディレクトリ：

     ```
     AddressBookProvider
     BookmarkProvider
     CalendarProvider
     LoginProvider
     LotusNotesAddressBookProvider
     LotusNotesCalendarProvider
     LotusNotesMailProvider
     MSExchangeAddressBookProvider
     MSExchangeCalendarProvider
     MSExchangeMailProvidervMailProvider
     NotesProvider
     PersonalNoteProvider
     Register
     SampleRSS
     SampleURLScraper
     SampleXML
     TemplateEditContainerProvider
     TemplateTabContainerProvider
     ```

```
URLScraperProvider
UWCAddressBookProvider
UserInfo
UserInfoProvider
XMLProvider
error
```

3. Portal Server が配備される Web コンテナを再起動します。

アップグレードのロールバック (Solaris)

ここでは、Portal Server Secure Remote Access のアップグレードのロールバック手順に影響する注意点を示したあと、その手順自体について説明します。

ロールバックの注意点 (Solaris)

Release 5 へのアップグレードをロールバックする手順は、*PortalServer6-base* の Release 4 インストールへの復元で構成されます。

ロールバック手順 (Solaris)

1. root としてログインするか、スーパーユーザーになります。

   ```
   su -
   ```

2. Directory Server をアップグレード前の状態に復元します。

   Directory Server のバックアップ / 復元のコマンド行および GUI ユーティリティーを使用します。Sun Java System Directory Server Enterprise Edition 6.0 管理ガイド』(http://docs.sun.com/doc/820-0291?l=ja) の「Directory Server バックアップと復元」の章を参照してください。

3. Release 5 の Portal Server Secure Remote Access パッケージを削除します。

   a. Java ES アンインストーラを起動します。

      ```
      /var/sadm/prod/SUNWentsys5/uninstall
      ```

   b. インストールされているすべての Portal Server Secure Remote Access コンポーネントを選択します。

   c. アンインストールに関して選択した項目を確認します。

   d. Java ES アンインストーラを終了します。

4. *PortalServer6-base* および *PortalServer6Config-base* ディレクトリを元の場所に復元します。

   アップグレード中、これらのディレクトリは .bak の拡張子を持つディレクトリに移動されました。

アップグレードのロールバック (Linux)

Release 5 にアップグレードするには Release 4 バイナリを削除する必要があるため、Linux でアップグレードをロールバックすることはきわめて困難です。

ロールバックするための 1 つの方法として、アップグレードの前に並列システムを作成し、アップグレードしようとする前にそのシステムをテストする方法があります。アップグレードのロールバックが必要になった場合は、その並列システムに戻すことができます。

複数インスタンスのアップグレード

配備のアーキテクチャーによっては、セキュリティーとスケーラビリティーを実現し、可用性を向上させるために、ゲートウェイなどの Portal Server Secure Remote Access コンポーネントが複数のコンピュータシステムに配備される場合があります。たとえば、ロードバランサを使用してゲートウェイコンポーネントを複数のコンピュータで実行することにより、負荷を分散させることができます。

ゲートウェイのインスタンスが負荷分散されている場合は、サービスを中断することなく連続してゲートウェイインスタンスをアップグレードする順次アップグレードを実行できます。この手順では、個々の Portal Server Secure Remote Access コンポーネントが以前のバージョンとは下位互換性がなく、Portal Server 自体とともに、すべてを Java ES Release 5 で同期する必要があるという制約を考慮に入れてください。ただし、順次アップグレード中、Portal Server インスタンスがアップグレードされている間も Release 4 の Portal Server Secure Remote Access インスタンスは稼働を継続できます。

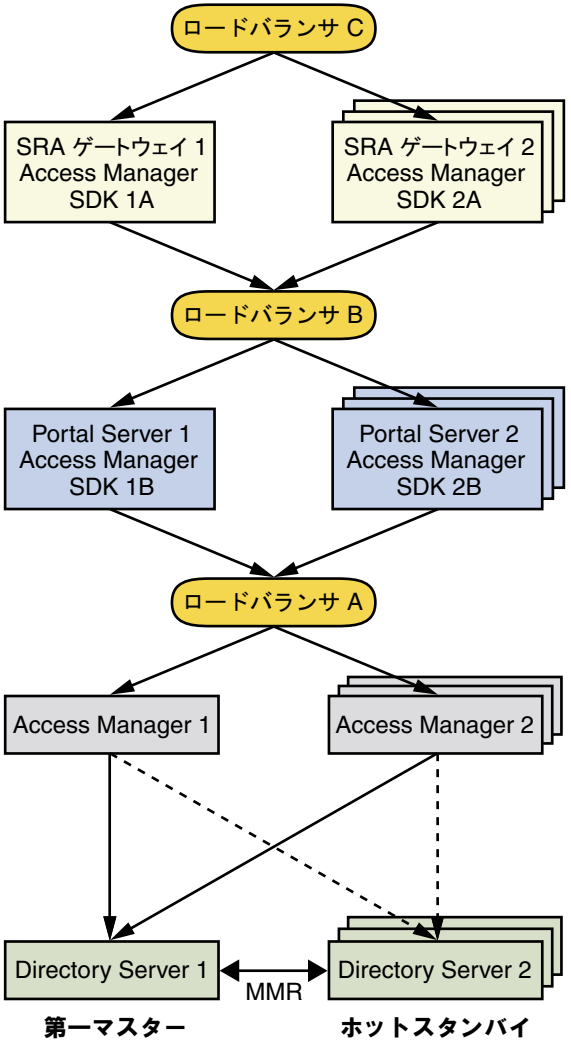
392 ページの図 16-1 に示されている配備アーキテクチャーは、順次アップグレードの手順を説明するために使用されます。

このアーキテクチャーでは、複数の Portal Server インスタンスが Portal Server Secure Remote Access ゲートウェイインスタンスを介してアクセスされます。可用性とスケーラビリティーを実現するために、Portal Server インスタンスとゲートウェイインスタンスの両方は負荷分散されます。

Portal Server インスタンスは、同様に、ロードバランサを介して Access Manager インスタンスにアクセスします。Access Manager および Access Manager SDK インスタンスは、マルチマスターレプリケーション (MMR) 用に設定されたディレクトリにアクセスします。ほかの Directory Server レプリケーションスキーマも使用できますが、MMR は可用性が高くてスケーラブルな、代表的なディレクトリサービスです。

図 16-1 では、アップグレード手順を説明しやすくするために、ゲートウェイ、Portal Server、Access Manager、および Directory Server の複数のインスタンスがグループ化されています。たとえば、`Portal Server 2` は、Portal Server の 2 ～ n 番目のインスタンスを表しています。

図 16-1　複数の Portal Server インスタンスに対する配備アーキテクチャーの例



Release 4 のゲートウェイ (および Portal Server) の Release 5 への順次アップグレードは、次のように実行されます。

1. Release 4 の Access Manager を Release 5 にアップグレードする場合は、281 ページの「複数インスタンスのアップグレード」の説明に従って順次アップグレードを実行します。Release 4 のゲートウェイまたは Release 4 の Portal Server の Release 5 へのアップグレードでは、Release 4 の Access Manager を Release 5 にアップグレードする必要がないことに注意してください。

2. 次のように、Portal Server インスタンスとゲートウェイインスタンスの設定を変更します。

   a. Directory Server 1 ではなく Directory Server 2 をポイントするように、Portal Server 2 を設定します。

      簡潔にするために、この手順および以降の手順では「Portal Server 2」は Portal Server 2 ～ Portal Server n を意味します。

   b. Directory Server 1 ではなく Directory Server 2 をポイントするように、ゲートウェイ 2 を設定します。

      簡潔にするために、この手順および以降の手順では、「ゲートウェイ 2」はゲートウェイ 2 ～ゲートウェイ n を意味します。

3. Portal Server 1 をアップグレードします。

   a. ロードバランサ B で Portal Server 1 を無効にします。

      これで、要求は Portal Server 1 に経路指定されなくなります。

   b. Directory Server MMR を無効にします。

      Directory Server 2 は Directory Server 1 と同期されなくなります。

   c. Access Manager SDK 1B を Release 5 にアップグレードします。

      284 ページの「Release 4 の Access Manager SDK のみのアップグレード」の手順を使用します。

   d. Portal Server 1 を Release 5 にアップグレードします。

      次の点に注意しながら、379 ページの「Release 4 の Portal Server Secure Remote Access のアップグレード」の説明に従って Portal Server インスタンスのアップグレードを実行します。

      - 382 ページの「ロードバランサの設定の削除」に示されているアップグレード前の作業に特に注意してください。
      - アップグレードを実行する前に、ローカルの Access Manager SDK における *AccessManagerConfig-base*/config/AMConfig.properties ファイル内の am.encryption.pwd の値が、それに関連付けられたリモートの Access Manager インスタンスのその値と同じであることを確認してください。
      - アップグレードする各 Portal Server インスタンスに対して psupgrade から要求されるポータル インスタンス ID パラメータには、NULL ではない一意の値を必ず指定してください。

      1 の Portal Server データが Release 5 に更新されます。

4. ゲートウェイ 1 をアップグレードします。

   a. ロードバランサ C でゲートウェイ 1 を無効にします。

      これで、要求はゲートウェイ 1 に経路指定されなくなります。

b. Access Manager SDK 1A を Release 5 にアップグレードします。

284 ページの「Release 4 の Access Manager SDK のみのアップグレード」の手順を使用します。

c. ゲートウェイ 1 を Release 5 にアップグレードします。

次の点に注意しながら、379 ページの「Release 4 の Portal Server Secure Remote Access のアップグレード」の説明に従ってゲートウェイのアップグレードを実行します。

- 382 ページの「ロードバランサの設定の削除」に示されているアップグレード前の作業に特に注意してください。
- アップグレードを実行する前に、ローカルの Access Manager SDK における *AccessManagerConfig-base*/config/AMConfig.properties ファイル内の am.encryption.pwd の値が、それに関連付けられたリモートの Access Manager インスタンスのその値と同じであることを確認してください。

5. 以前に無効にした Portal Server 1 とゲートウェイ 1 を、それぞれのロードバランサで次のように有効にします。

a. ロードバランサ B で Portal Server 1 を有効にします。

要求は、ふたたび Portal Server 1 に経路指定されるようになります。

b. ロードバランサ C でゲートウェイ 1 を有効にします。

要求は、ふたたびゲートウェイ 1 に経路指定されるようになります。

6. Portal Server 2 とゲートウェイ 2 を、それぞれのロードバランサで次のように無効にします。

a. ロードバランサ B で Portal Server 2 を無効にします。

これで、要求は Portal Server 2 に経路指定されなくなります。

b. ロードバランサ C でゲートウェイ 2 を無効にします。

これで、要求はゲートウェイ 2 に経路指定されなくなります。

7. Portal Server 2 をアップグレードします。

a. Directory Server 1 をポイントするように、Portal Server 2 の設定を復元します。

b. Access Manager SDK 2B を Release 5 にアップグレードします。

284 ページの「Release 4 の Access Manager SDK のみのアップグレード」の手順を使用します。

c. Portal Server 2 を Release 5 にアップグレードします。

Portal Server 1 のアップグレードと同じ手順 (393 ページの手順 d) を使用します。

d. ロードバランサ B で Portal Server 2 を有効にします。

要求は、ふたたび Portal Server 2 に経路指定されるようになります。

8. ゲートウェイ 2 をアップグレードします。

a. Directory Server 1 をポイントするように、ゲートウェイ 2 の設定を復元します。

b. Access Manager SDK 2A を Release 5 にアップグレードします。

284 ページの「Release 4 の Access Manager SDK のみのアップグレード」の手順を使用します。

c. Gateway 2 を Release 5 にアップグレードします。

ゲートウェイ 1 のアップグレードと同じ手順 (394 ページの手順 c) を使用します。

d. ロードバランサ C でゲートウェイ 2 を有効にします。

要求は、ふたたびゲートウェイ 2 に経路指定されるようになります。

9. Directory Server MMR を有効にします。

これで、Directory Server 2 の Portal Server データが Directory Server 1 と同期されます。

注 以前のリリースのゲートウェイコンポーネントがアクティブな状態を維持し (上記の手順には該当しない)、かつゲートウェイインスタンスがロードバランサを介してアクセスされる一方で、Portal Server インスタンスが Release 5 にアップグレードされる順次アップグレードシナリオでは、すべてのゲートウェイインスタンスで、*PortalServer6Config-base*/GWConfig.properties ファイルおよび GWConfig-default.properties ファイル内の次の設定プロパティーが、ロードバランサではなくゲートウェイを参照していることを確認するようにしてください。

GW_IP=*Gateway_hostIP*
GW_HOST=*Gateway_hostName*

これらのプロパティーがロードバランサをポイントしている場合、ゲートウェイは、アップグレードされた Portal Server インスタンスにアクセスしなくなります。

Java ES Release 3 からの Portal Server Secure Remote Access のアップグレード

Java ES 2005Q1 (Release 3) Portal Server Secure Remote Access を Release 5 にアップグレードする手順は、次の例外を除き、Release 4 の Portal Server Secure Remote Access を Release 5 にアップグレードする手順と同じです。

- Portal Server Secure Remote Access が依存するコンポーネントのアップグレード
- Release 3 の Portal Server Secure Remote Access のアップグレード
- 複数インスタンスのアップグレード

Portal Server Secure Remote Access が依存するコンポーネントのアップグレード

Portal Server Secure Remote Access を Release 3 からアップグレードする場合は、Portal Server Secure Remote Access をアップグレードする前に Access Manager を Release 4 または Release 5 にアップグレードしてください。ただし、ほかのいずれかの依存関係を Release 3 のままにすることも、一部の依存関係を Release 4 に、それ以外を Release 5 にアップグレードすることもできません。詳細については、377 ページの「選択的なアップグレードに関する問題」を参照してください。

次の依存関係は、下記に示す順序でアップグレードする必要があります。

1. 共有コンポーネント : Java ES 共有コンポーネントを Release 5 にアップグレードする手順は、67 ページの第 2 章「Java ES 共有コンポーネントのアップグレード」で説明されています。
2. Directory Server: Directory Server を Release 5 にアップグレードする手順は、117 ページの「Java ES Release 2 からの Directory Server のアップグレード」で説明されています。
3. Access Manager (Access Manager SDK): Access Manager を Release 5 にアップグレードする手順は、257 ページの第 14 章「Access Manager」で説明されています。
4. Portal Server: Portal Server をアップグレードする手順は、305 ページの第 15 章「Portal Server」で説明されています。

Release 3 の Portal Server Secure Remote Access のアップグレード

Release 3 Portal Server Secure Remote Access を Release 5 にアップグレードするには、「Release 4」を「Release 3」と読み替えた上で、378 ページの「Java ES Release 4 からの Portal Server Secure Remote Access のアップグレード」の手順を使用してください。

複数インスタンスのアップグレード

配備のアーキテクチャーによっては、セキュリティーとスケーラビリティーを実現し、可用性を向上させるために、ゲートウェイなどの Portal Server Secure Remote Access コンポーネントが複数のコンピュータシステムに配備される場合があります。たとえば、ロードバランサを使用してゲートウェイコンポーネントを複数のコンピュータで実行することにより、負荷を分散させることができます。

Release 3 の Portal Server Secure Remote Access からの複数インスタンスのアップグレードを実行するときは、「Release 4」を「Release 3」と読み替えた上で、391 ページの「複数インスタンスのアップグレード」の手順を使用してください。さらに、392 ページの手順 1 の説明に従って、Access Manager をアップグレードしてください。

Java ES Release 2 からの Portal Server Secure Remote Access のアップグレード

ここでは、Java ES 2004Q2 (Release 2) の Portal Server Secure Remote Access の Release 5 へのアップグレードについて説明します。このアップグレード手順は、以降の節に説明されているいくつかの変更を除き、Release 4 の Portal Server Secure Remote Access の Release 5 へのアップグレードとほぼ同じです。

- Release 2 のアップグレード前の作業
- Release 2 の Portal Server Secure Remote Access のアップグレード
- Release 2 のアップグレード後の作業
- 複数インスタンスのアップグレード

注 Linux プラットフォームで Release 2 の Portal Server Secure Remote Access からアップグレードする場合は、二重のアップグレード、つまり、Portal Server Secure Remote Access とオペレーティングシステムの両方をアップグレードする必要があります (Release 5 の Portal Server Secure Remote Access は RHEL 2.1 でサポートされていない)。詳細については、377 ページの「二重のアップグレード」を参照してください。

Release 2 のアップグレード前の作業

Portal Server Secure Remote Access を Release 2 からアップグレードするときのアップグレード前の作業は、次の作業を除き、380 ページの「Release 4 のアップグレード前の作業」で説明されている作業と同じです。

- Portal Server Secure Remote Access が依存するコンポーネントのアップグレード
- ゲートウェイサービスエントリの削除

Portal Server Secure Remote Access が依存するコンポーネントのアップグレード

Portal Server Secure Remote Access を Release 2 からアップグレードする場合は、Portal Server Secure Remote Access をアップグレードする前に Access Manager を Release 4 または Release 5 にアップグレードしてください。なお、ほかのいずれかの依存関係を Release 2 のままにすることも、一部の依存関係を Release 4 に、それ以外を Release 5 にアップグレードすることもできません。詳細については、377 ページの「選択的なアップグレードに関する問題」を参照してください。

次の依存関係は、下記に示す順序でアップグレードする必要があります。

1. 共有コンポーネント : Java ES 共有コンポーネントを Release 5 にアップグレードする手順は、67 ページの第 2 章「Java ES 共有コンポーネントのアップグレード」で説明されています。ただし、共有コンポーネントがまだアップグレードされていない場合は、psupgrade スクリプトによって自動的にアップグレードされます。
2. Directory Server: Directory Server を Release 5 にアップグレードする手順は、117 ページの「Java ES Release 2 からの Directory Server のアップグレード」で説明されています。
3. Access Manager (Access Manager SDK): Access Manager を Release 5 にアップグレードする手順は、257 ページの第 14 章「Access Manager」で説明されています。
4. Portal Server: Portal Server をアップグレードする手順は、305 ページの第 15 章「Portal Server」で説明されています。

Release 2 Portal Server Secure Remote Access を Release 5 にアップグレードするには、「Release 4」を「Release 2」と読み替えた上で、378 ページの「Java ES Release 4 からの Portal Server Secure Remote Access のアップグレード」の手順を使用してください。

ゲートウェイサービスエントリの削除

Portal Server を Release 2 からアップグレードした場合は、amService-srapGateway ユーザーエントリを手動で削除してください。そうしないと、アップグレードのあと、ゲートウェイが起動に失敗します。次の手順を実行します。

1. Access Manager コンソールにログインします。
2. 組織 DN 内のすべてのユーザーを表示します。
3. amService-srapGateway ユーザーを削除します。

Release 2 の Portal Server Secure Remote Access のアップグレード

Java ES 2004Q2 (Release 2) の Portal Server Secure Remote Access を Release 5 にアップグレードする手順は、Release 4 の Portal Server Secure Remote Access を Release 5 にアップグレードする手順と同じです。

Release 2 Portal Server Secure Remote Access を Release 5 にアップグレードするには、「Release 4」を「Release 2」と読み替えた上で、378 ページの「Java ES Release 4 からの Portal Server Secure Remote Access のアップグレード」の手順を使用してください。

Release 2 のアップグレード後の作業

Release 2 からのアップグレードのアップグレード後の作業は、次の作業を除き、388 ページの「Release 4 のアップグレード後の作業」で説明されている作業と同じです。

- プロキシレットサービスのための Portal Server ドメインの設定

プロキシレットサービスのための Portal Server ドメインの設定

Release 2 の Portal Server Secure Remote Access を Release 5 にアップグレードしたあと、正しい Portal Server ドメインの値を設定してください。

1. Portal Server コンソールにログインし、「Secure Remote Access」の下の「プロキシレット」タブに移動します。
2. プロキシレットサービスが見つかった組織の識別名 (DN) を選択します。
3. 「プロキシレットルール」の「ドメイン」フィールドで、`SERVER_DOMAIN` を、Portal Server がインストールされているドメイン名に置き換えます。
4. プロキシレットサービスが見つかったすべての組織に対して、上記の手順を繰り返します。

複数インスタンスのアップグレード

Release 2 の Portal Server Secure Remote Access コンポーネント (または Portal Server) の Release 5 へのアップグレードでは、複数インスタンスの順次アップグレード (391 ページの「複数インスタンスのアップグレード」を参照) はサポートされていません

Portal Server Secure Remote Access の IFR 7.0 からのアップグレード

ここでは、暫定機能リリース (Interim Feature Releases、IFR) 7.0 2005Q4 から Java ES 5 (Release 5) への Portal Server Secure Remote Access のアップグレードについて説明します。

この節で説明する内容は、次のとおりです。

- はじめに
- Portal Server Secure Remote Access IFR 7.0 のアップグレード
- 複数インスタンスのアップグレード

はじめに

Portal Server Secure Remote Access IFR 7.0 を Release 5 にアップグレードするときは、アップグレードプロセスに関する次の事項を考慮します。

Portal Server Secure Remote Access IFR を Release 5 にアップグレードするための `psupgrade` スクリプトは、Release 4 からのアップグレードと異なり、新しいパッケージをインストールしません。代わりに、アップグレード手順では、次のパッチを適用するよう求められます。

表 16-5 Portal Server Secure Remote Access IFR を Release 5 にアップグレードするためのパッチ[1]

説明	パッチ ID: Solaris 9 および 10	パッチ ID: Linux
Portal Server 7.1	121465-10 (SPARC)	121467-10
	121466-10 (x86)	
Portal Server 7.1 ローカライズ版	123254-02 (SPARC)	123255-02
	124590-02 (x86)	

1. この表に示されているパッチのリビジョン番号は、Java ES Release 5 にアップグレードするための最低要件です。表に記載されているものよりも新しいリビジョンのパッチが使用可能になった場合は、新しいほうのパッチを使用してください。

Portal Server Secure Remote Access IFR 7.0 のアップグレード

この節では、Solaris および Linux プラットフォームで、Portal Server Secure Remote Access を IFR から Java ES Release 5 にアップグレードする方法について説明します。プラットフォーム固有の手順に依存する項目には、その手順が適用されるオペレーティングシステムが示されています。この節で説明する内容は、次のとおりです。

- アップグレード前の作業
- Portal Server Secure Remote Access IFR 7.0 のアップグレード (Solaris)
- Portal Server Secure Remote Access IFR 7.0 のアップグレード (Linux)
- アップグレードの確認
- アップグレード後の作業
- アップグレードのロールバック (Solaris)
- アップグレードのロールバック (Linux)

アップグレード前の作業

IFR をアップグレードするときのアップグレード前の作業は、Release 4 のアップグレード手順 (380 ページの「Release 4 のアップグレード前の作業」を参照) と同じです。

Portal Server Secure Remote Access IFR 7.0 のアップグレード (Solaris)

ここでは、Portal Server Secure Remote Access のアップグレード手順に影響する注意点を示したあと、その手順自体について説明します。

IFR 7 のアップグレードの注意点 (Solaris)

Portal Server Secure Remote Access IFR の Release 5 へのアップグレードでは、Release 4 のアップグレードと同じ注意点 (383 ページの「アップグレードの注意点 (Solaris)」を参照) を考慮に入れてください。

IFR 7 のアップグレード手順 (Solaris)

次に示す手順は、アップグレードを実行するコンピュータ上の Portal Server Secure Remote Access に適用されます。

1. root としてログインするか、スーパーユーザーになります。

```
su -
```

2. ローカルで実行されているゲートウェイ、Rewriter プロキシ、または Netlet プロキシのインスタンスをすべて停止します。

   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*

   これらのプロセスが停止したことを確認します。

   ゲートウェイ : netstat -an | grep 443
   Rewriter プロキシ : netstat -an | grep 10443
   Netlet プロキシ : netstat -an | grep 10555

3. Access Manager が稼働していることを確認します。

4. 401 ページの表 16-5 に基づいて、必要なパッチを入手します。

   特定のリビジョンを使用するように指示された場合以外は、常に、使用可能な最新のパッチリビジョンを使用してください。

   パッチは http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access から /tmp にダウンロードします。

5. 表 16-5 の適切な Portal Server パッチと、必要に応じてローカライズ版パッチを適用します。

   patchadd *patch_ID*

6. パッチのアップグレードが成功したことを確認します。

   showrev -p | grep *patch_ID*

   この出力では、手順 5 で適用したパッチ ID のバージョンが返されるはずです。

7. 手順 5 でローカライズ版パッケージをアップグレードした場合は、Portal Server コンソール JVM のロケールを UTF-8 に設定します。

   ```
   export LC_ALL=ja_JP.UTF-8
   export LANG=ja_JP.UTF-8
   ```

8. psupgrade スクリプトで必要な 2 つの環境変数 (ANT_HOME および JAVA_HOME) を設定します。

   ```
   export ANT_HOME=/usr/sfw
   export JAVA_HOME=/usr/jdk/entsys-j2se
   ```

9. コンピュータ上に十分なスワップ空間があることを確認します。

   ガイドラインとして、スワップ空間を物理 RAM の容量の 2 倍に設定することをお勧めします。

10. psupgrade スクリプトを実行します。

    ```
    cd PortalServer7-base/bin
    ./psupgrade
    ```

    psupgrade スクリプトは Java ES Release 5 ディストリビューションから実行されず、Java ES インストーラも起動しません (パッケージにはすでにパッチが適用済み)。

Portal Server Secure Remote Access IFR 7.0 のアップグレード (Linux)

ここでは、Portal Server Secure Remote Access のアップグレード手順に影響する注意点を示したあと、その手順自体について説明します。

IFR 7 のアップグレードの注意点 (Linux)

Linux プラットフォームで Portal Server Secure Remote Access ソフトウェアを Release 5 にアップグレードするときは、Linux オペレーティングシステムでは Release 5 のパッチをインストールすると以前の RPM が削除される点を除き、Solaris プラットフォームのときと同じ注意点 (383 ページの「アップグレードの注意点 (Solaris)」を参照) を考慮に入れてください。

IFR 7 のアップグレード手順 (Linux)

次に示す手順は、アップグレードを実行するコンピュータ上の Portal Server Secure Remote Access に適用されます。

注意 Linux での Portal Server Secure Remote Access IFR から Release 5 へのアップグレードはロールバックできません。次の手順を実行する前に、必ずシステムをバックアップしてください。

1. root としてログインするか、スーパーユーザーになります。

   ```
   su -
   ```

2. ローカルで実行されているゲートウェイ、Rewriter プロキシ、または Netlet プロキシのインスタンスをすべて停止します。

   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*

これらのプロセスが停止したことを確認します。

ゲートウェイ : `netstat -an | grep 443`
Rewriter プロキシ : `netstat -an | grep 10443`
Netlet プロキシ : `netstat -an | grep 10555`

3. Access Manager が稼働していることを確認します。
4. 401 ページの表 16-5 のパッチ番号と RPM 名に基づいて、必要なパッチを入手します。

   特定のリビジョンを使用するように指示された場合以外は、常に、使用可能な最新のパッチリビジョンを使用してください。

   パッチは http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access から /tmp にダウンロードします。
5. 表 16-5 の Portal Server パッチと、必要に応じて Portal Server のローカライズ版 RPM を、その順序で適用します。

   Portal Server パッチの Readme ファイルを参照すると、スクリプトを使用してパッチの RPM を適用する方法が説明されています。

   ```
   cd /tmp
   ```

   ここで、/tmp は、パッチをダウンロードしたディレクトリです。

   ```
   ./update
   ```

   update スクリプトによって RPM がインストールされます。

   ローカライズ版パッチの場合は、次のコマンドを使用して各 RPM をインストールします。

   `rpm -Fvh` *patchName-version*`.rpm`
6. パッチのアップグレードが成功したことを確認します。

   ```
   rpm -qa | grep sun-portal-core
   ```

   RPM のアップグレードリビジョン番号が返されることを確認します。
7. 手順 5 でローカライズ版パッケージをアップグレードした場合は、Portal Server コンソール JVM のロケールを UTF-8 に設定します。

   ```
   export LC_ALL=ja_JP.UTF-8
   export LANG=ja_JP.UTF-8
   ```
8. psupgrade スクリプトで必要な 2 つの環境変数 (ANT_HOME および JAVA_HOME) を設定します。

   ```
   export ANT_HOME=/opt/sun
   export JAVA_HOME=/usr/jdk/entsys-j2se
   ```
9. コンピュータ上に十分なスワップ空間があることを確認します。

ガイドラインとして、スワップ空間を物理 RAM の容量の 2 倍に設定することをお勧めします。

10. psupgrade スクリプトを実行します。

```
cd PortalServer7-base/bin
./psupgrade
```

psupgrade スクリプトは Java ES Release 5 ディストリビューションから実行されず、Java ES インストーラも起動しません (パッケージにはすでにパッチが適用済み)。

アップグレードの確認

Portal Server Secure Remote Access パッケージの Release 5 へのパッチの適用は、次のコマンドを使用することにより確認できます。

```
PortalServer7-base/bin/psadmin --version --adminuser admin_ID
-f adminpasswordfile.
```

出力値については、380 ページの表 16-4 を参照してください。

また、次の場所にあるアップグレードログファイルを確認することもできます。

```
/var/sadm/install/logs/Sun_Java_System_Portal_Server_upagrede.log
```

アップグレード後の作業

Portal Server Secure Remote Access を Release 5 にアップグレードした場合は、アップグレード後の作業は必要ありません。

アップグレードのロールバック (Solaris)

ここでは、Portal Server Secure Remote Access のアップグレードのロールバック手順に影響する注意点を示したあと、その手順自体について説明します。

ロールバックの注意点 (Solaris)

Release 5 へのアップグレードをロールバックする手順は、*PortalServer7-base* の IFR インストールへの復元で構成されます。

ロールバック手順 (Solaris)

1. root としてログインするか、スーパーユーザーになります。

```
su -
```

2. Directory Server をアップグレード前の状態に復元します。

Directory Server のバックアップ / 復元のコマンド行および GUI ユーティリティーを使用します。Sun Java System Directory Server Enterprise Edition 6.0 管理ガイド』(http://docs.sun.com/doc/820-0291?l=ja) の「Directory Server バックアップと復元」の章を参照してください。

3. 401 ページの表 16-5 の Portal Server 7.1 パッチをバックアウトします。

   `patchrm` *patch_ID*

アップグレードのロールバック (Linux)

Linux プラットフォームでは、アップグレードをロールバックする手順はありません。

複数インスタンスのアップグレード

配備のアーキテクチャーによっては、スケーラビリティーを実現し、可用性を向上させるために、Portal Server Secure Remote Access が複数のコンピュータシステムに配備される場合があります。たとえば、ロードバランサを使用して Portal Server Secure Remote Access コンポーネントを複数のコンピュータで実行することにより、負荷を分散させることができます。

Portal Server Secure Remote Access のインスタンスが負荷分散されている場合は、サービスを中断することなく連続して Portal Server Secure Remote Access インスタンスをアップグレードする順次アップグレードを実行できます。Portal Server Secure Remote Access の各インスタンスを、ほかのインスタンスが稼働している間にアップグレードします。402 ページの「Portal Server Secure Remote Access IFR 7.0 のアップグレード」の説明に従って、各インスタンスのアップグレードを実行します。

IFR Portal Server Secure Remote Access からの複数インスタンスのアップグレードを実行する場合は、「Release 4」を「IFR」と読み替えた上で、391 ページの「複数インスタンスのアップグレード」の手順を使用してください。さらに、392 ページの手順 1 の説明に従って、Access Manager をアップグレードしてください。

Java Enterprise System リリースの内容

この付録では、各種の Java Enterprise System リリースの内容について説明します。この章で説明する内容は、次のとおりです。

- 410 ページの「Java ES 2003Q4 (Release 1)」
- 412 ページの「Java ES 2004Q2 (Release 2)」
- 415 ページの「Java ES 2005Q1 (Release 3)」
- 420 ページの「Java ES 2005Q4 (Release 4)」
- 425 ページの「Java ES 5 (Release 5)」

Java ES 2003Q4 (Release 1)

この節では、Java Enterprise System 2003Q4 の内容について説明します。

Release 1 のインストーラで選択可能なコンポーネント

Sun Open Network Environment (Sun ONE) および Sun Cluster コンポーネント製品は、分散型のエンタープライズアプリケーションをサポートするために必要なインフラストラクチャーサービスを提供します。これらのコンポーネント製品は、次のとおりです。

- Sun Cluster 3.1 と Sun ONE 用 Sun Cluster エージェント
- 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

Release 1 の共有コンポーネント

共有コンポーネントは、コンポーネント製品が依存するローカルサービスとテクノロジサポートを提供します。コンポーネント製品をインストールすると、そのコンポーネントに必要な共有コンポーネントがインストールされていない場合に、Java ES インストーラは自動的にこれらのコンポーネントをインストールします。

Java Enterprise System には次の共有コンポーネントが含まれます。

- ANT (Jakarta ANT Java/XML ベースの構築ツール)
- 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 検索エンジン
- 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　また、Perl も Application Server と Directory Server のシステムに必要ですが、Java ES 共有コンポーネントとして自動的にはインストールされません。

Java ES 2004Q2 (Release 2)

この節では、Java Enterprise System 2004Q2 の内容について説明します。

Release 2 のインストーラで選択可能なコンポーネント

コンポーネント製品は、分散型のエンタープライズアプリケーションをサポートするために必要なインフラストラクチャーサービスを提供します。Java Enterprise System を特定のホストにインストールするときは、配備の全体的なアーキテクチャーに基づいて、そのホストにインストールするコンポーネント製品を選択します。

Java Enterprise System 2004Q2 には、次のコンポーネント製品が含まれます。

コミュニケーションサービスとコラボレーションサービス

- 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 サービスとアプリケーションサービス

- Sun Java System Application Server 7.0 Update 3 (Standard および Platform Edition)
- Sun Java System Web Server 6 2004Q2 Update 1 Service Pack 2
- Sun Java System Message Queue 3.5 SP1 (Platform および Enterprise Edition)

ディレクトリサービスとアイデンティティーサービス

- Sun Java System Identity Server 6.2 2004Q2
  (Sun Java System Communications Services 6 2004Q2 ユーザー管理ユーティリティーを含む)
- Sun Java System Directory Server 5 2004Q2
- Sun Java System Directory Proxy Server 5 2004Q2

可用性サービス

- Sun Cluster 3.1 4/04 および Sun Java System 用 Sun Cluster エージェント

管理サービス

- Sun Java System Administration Server 5 2004Q2
- Sun Remote Services Net Connect 3.5

Sun Cluster、Sun Cluster エージェント、および Sun Remote Services Net Connect は Linux オペレーティングシステムでは使用できません。

Release 2 の共有コンポーネント

共有コンポーネントは、コンポーネント製品が依存するローカルサービスとテクノロジサポートを提供します。コンポーネント製品をインストールすると、そのコンポーネントに必要な共有コンポーネントがインストールされていない場合に、Java ES インストーラは自動的にこれらのコンポーネントをインストールします。

Java Enterprise System 2004Q2 には次の共有コンポーネントが含まれます。

- Ant (Jakarta ANT Java/XML ベースの構築ツール)
- 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 検索エンジン
- LDAP C Language SDK
- LDAP Java SDK
- NSPR (Netscape Portable Runtime)
- NSS (Network Security Services)
- Perl LDAP (NSPERL を含む)
- 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 (Release 3)

この節では、Java Enterprise System 2005Q1 の内容について説明します。

Release 3 のインストーラで選択可能なコンポーネント

Java ES インストーラのコンポーネント選択ページでは、選択可能なコンポーネントが提供するサービスごとにグループ化されています。また、次のリストは各コンポーネントとともにインストールされるサブコンポーネントを示します。

コミュニケーションサービスとコラボレーションサービス

- Sun Java System Messaging Server 6 2005Q1
- Sun Java System Calendar Server 6 2005Q1
- Sun Java System Instant Messaging 7 2007Q1
  - Instant Messaging サーバーコア (サーバーおよびマルチプレクサソフトウェアを含む)
  - Instant Messaging リソース
  - Access Manager Instant Messaging サービス
- Sun Java System Calendar Server 6 2005Q1
- Sun Java System Portal Server Secure Remote Access 6 2005Q1
  - Secure Remote Access コア
  - ゲートウェイ
  - Netlet プロキシ
  - Rewriter プロキシ
- Sun Java System Communications Express 2005Q1
- Sun Java System Directory Preparation Tool

Web サービスとアプリケーションサービス

- Sun Java System Application Server Enterprise Edition 8.1 2005Q1
  - ドメイン管理サーバー
  - Application Server ノードエージェント
  - コマンド行管理ツール
  - 負荷分散プラグイン

    設定時の選択により、Web Server または Apache Web Server のいずれかで使用できます。デフォルトは Web Server です。
  - PointBase
  - サンプルアプリケーション
- Sun Java System Web Server 6 2005Q1 Update 1 Service Pack 4
- Sun Java System Message Queue 3 2005Q1

ディレクトリサービスとアイデンティティーサービス

- Sun Java System Access Manager 6.3 2005Q1

  Portal Server および Messaging Server 用の Delegated Administrator プロビジョニングツールは、Access Manager とともに自動的にインストールされます。
  - アイデンティティー管理およびポリシーサービスコア (Delegated Administrator ユーティリティーを含む)
  - Access Manager 管理コンソール
  - 連携管理の共有ドメインサービス
  - Access Manager SDK
- Sun Java System Directory Server 5 2005Q1
- Sun Java System Directory Proxy Server 5 2005Q1

可用性サービス

- Sun Cluster 3.1 9/04
  - Sun Cluster コア
- Sun Java System 用 Sun Cluster エージェント
  - 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 (高可用性セッションストレージで使用)

管理サービス

- Sun Java System Administration Server 5 2005Q1
- Sun[SM] Remote Services Net Connect 3.1.1

注

Sun Cluster、Sun Cluster エージェント、および Sun Remote Services Net Connect は Solaris 10 または Linux オペレーティングシステムでは利用できません。

Sun Remote Services Net Connect は Solaris x86 プラットフォームでは利用できません。

Release 3 の共有コンポーネント

共有コンポーネントは、選択可能なコンポーネントに対してローカルサービスとテクノロジサポートを提供します。Java ES コンポーネントをインストールすると、そのコンポーネントに必要な共有コンポーネントがインストールされていない場合に、インストーラは自動的にこれらのコンポーネントをインストールします。

この Java ES のリリースには、次の共有コンポーネントが含まれています。

- Ant (Jakarta ANT Java/XML ベースの構築ツール)
- Apache SOAP (Simple Object Access Protocol) Runtime
- Berkeley Database
- 共通エージェントコンテナ
- 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 (NSPERL を含む)
- 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 (Solaris のみ)
- Sun Java Monitoring Framework
- Sun Java Web Console
- Tomcat Servlet JSP Container
- XML C Library (`libxml`)
- WSCL (Web services Common Library)

Java ES 2005Q4 (Release 4)

この節では、Java Enterprise System 2005Q4 の内容について説明します。

Release 4 のインストーラで選択可能なコンポーネント

Java ES インストーラのコンポーネント選択ページでは、選択可能なコンポーネントが提供するサービスごとにグループ化されています。また、次のリストは各コンポーネントとともにインストールされるサブコンポーネントを示します。

コミュニケーションサービスとコラボレーションサービス

- 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 サーバーコア (サーバーおよびマルチプレクサソフトウェアを含む)
  - Instant Messaging リソース
  - Access Manager Instant Messaging サービス
- Sun Java System Portal Server 6.3.1 2005Q4
- Sun Java System Portal Server Secure Remote Access 6.3.1 2005Q4
  - Secure Remote Access コア
  - ゲートウェイ
  - Netlet プロキシ
  - Rewriter プロキシ
- 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 コンソールおよびユーティリティー
  - Delegated Administrator Server

Web サービスとアプリケーションサービス

- Sun Java System Application Server Enterprise Edition 8.1 2005Q4
  - ドメイン管理サーバー
  - Application Server ノードエージェント
  - コマンド行管理ツール
  - 負荷分散プラグイン

    設定時の選択により、Web Server または Apache Web Server のいずれかで使用できます。デフォルトは Web Server です。
  - PointBase データベース
  - サンプルアプリケーション
- 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

ディレクトリサービスとアイデンティティーサービス

- Sun Java System Access Manager 7.0 2005Q4
  - アイデンティティー管理およびポリシーサービスコア
  - Access Manager 管理コンソール
  - 連携管理の共有ドメインサービス
  - Access Manager SDK
- Sun Java System Directory Server 5.2 2005Q4
- Sun Java System Directory Proxy Server 5.2 2005Q4

可用性サービス

- Sun Cluster 3.1 8/05
  - Sun Cluster コア
  - Sun Java System 用 Sun Cluster エージェント
    - 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

管理サービス

- Sun Java System Administration Server 5.2 2005Q4

注 Sun Cluster および Sun Cluster エージェントは、Solaris オペレーティングシステムではサポートされていますが、Linux オペレーティングシステムではサポートされていません。

Release 4 の共有コンポーネント

共有コンポーネントは、選択可能なコンポーネントに対してローカルサービスとテクノロジサポートを提供します。Java ES コンポーネントをインストールすると、そのコンポーネントに必要な共有コンポーネントがインストールされていない場合に、インストーラは自動的にこれらのコンポーネントをインストールします。

この Java ES のリリースには、次の共有コンポーネントが含まれています。

- ANT (Jakarta ANT Java/XML ベースの構築ツール)
- ACL (Apache Commons Logging)
- BDB (Berkeley Database)
- CAC (共通エージェントコンテナ)
- 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) コア
- 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、Solaris のみ)
- MFWK (Java ES Monitoring Framework)
- SJWC (Sun Java Web Console)
- WSCL (Web services Common Library)

Java ES 5 (Release 5)

この節では、Java Enterprise System 5 の内容について説明します。

Release 5 のインストーラで選択可能なコンポーネント

Java ES インストーラのコンポーネント選択ページでは、選択可能なコンポーネントが提供するサービスごとにグループ化されています。また、次のリストは各コンポーネントとともにインストールされるサブコンポーネントを示します。コミュニケーションサービスコンポーネントは Java ES インストーラではサポートされなくなったことに注意してください。

コラボレーションサービス

- Sun Java System Portal Server 7.1
- Sun Java System Portal Server Secure Remote Access 7.1
  - ゲートウェイ
  - Netlet プロキシ
  - Rewriter プロキシ

Web サービスとアプリケーションサービス

- Sun Java System Application Server Enterprise Edition 8.2
  - ドメイン管理サーバー
  - Application Server ノードエージェント
  - コマンド行管理ツール
  - 負荷分散プラグイン

    設定時の選択により、Web Server または Apache Web Server のいずれかで使用できます。デフォルトは Web Server です。
  - サンプルアプリケーション
- Sun Java System Web Server 7.0
  - Web Server 7.0 CLI
  - Web Server 7.0 コア
  - Web Server 7.0 サンプル
- Sun Java Web Proxy Server 4.0.4

- Sun Java System Message Queue 3.7UR1
- Service Registry 3.1
  - Service Registry クライアントサポート
  - Service Registry 配備サポート

ディレクトリサービスとアイデンティティーサービス

- Sun Java System Access Manager 7.1
  - アイデンティティー管理およびポリシーサービスコア
  - Access Manager 管理コンソール
  - 連携管理の共有ドメインサービス
  - Access Manager SDK
  - 分散認証
  - クライアント SDK
  - セッションフェイルオーバークライアント
- 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 コマンド行ユーティリティー
  - Sun Java System Directory Proxy Server 6 Core Server

可用性サービス

- Sun Cluster 3.1
  - Sun Cluster コア
- Sun Cluster Geographic Edition 3.1 2006Q4
- Sun Cluster エージェント 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

共有サービス

- すべての共有コンポーネント

  428 ページの「Release 5 の共有コンポーネント」を参照してください。

- Sun Java System Monitoring Console 1.0
- Java DB
  - Java DB クライアント
  - Java DB Server

注 Sun Cluster および Sun Cluster エージェントは、Solaris オペレーティングシステムではサポートされていますが、Linux オペレーティングシステムではサポートされていません。

Release 5 の共有コンポーネント

共有コンポーネントは、選択可能なコンポーネントに対してローカルサービスとテクノロジサポートを提供します。Java ES コンポーネントをインストールすると、そのコンポーネントに必要な共有コンポーネントがインストールされていない場合に、インストーラは自動的にこれらのコンポーネントをインストールします。

この Java ES のリリースには、次の共有コンポーネントが含まれています。

- ANT (Jakarta ANT Java/XML ベースの構築ツール)
- ACL (Apache Commons Logging)
- BDB (Berkeley Database)
- Sun Cluster 専用の CAC (共通エージェントコンテナ)
- CAC (共通エージェントコンテナ)
- 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) コア
- 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、Solaris のみ)
- 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
  アップグレード手順 , 34
  共有コンポーネントに対する依存関係 , 48
  サブコンポーネント , 421, 426
  製品コンポーネントの依存関係 , 53
  略称 , 28, 46
AccessManager-base パス , 257
AccessManagerConfig-base パス , 257
ACL 共有コンポーネント
  依存関係の表 , 48
  フルネーム , 29
amconfig スクリプト (AccessManager), 274
ampre71upgrade スクリプト (Access Manager), 271, 295
amupgrade スクリプト (AccessManager), 278, 302
Ant (Jakarta ANT Java/XML ベースの構築ツール)、「ANT」を参照
ANT 共有コンポーネント
  依存関係の表 , 48
  フルネーム , 29
ANT_HOME 環境変数 , 322, 325, 346, 357, 359, 384, 387, 403, 405
Apache Common Logging、「ACL」を参照
Application Server
  アップグレード手順 , 34
  インストール後の設定 , 222
  共有コンポーネントに対する依存関係 , 48
  サブコンポーネント , 421, 425
  製品コンポーネントの依存関係 , 54
  略称 , 28, 46
AppServer7-base のパス , 204
AppServer7Config-base のパス , 204
AppServer8-base のパス , 203
AppServer8Config-base のパス , 204
AppServer8Install-base のパス , 204

B

BDB 共有コンポーネント
  依存関係の表 , 48
  フルネーム , 29
Berkeley Database、「BDB」を参照

C

CAC 共有コンポーネント
  依存関係の表 , 48
  フルネーム , 29

D

db2bak ユーティリティー , 268
Derby Database 共有コンポーネント , 423

Directory Proxy Server
アップグレード手順 , 34
共有コンポーネントに対する依存関係 , 48
製品コンポーネントの依存関係 , 54
略称 , 28, 46
Directory Server
アップグレード手順 , 34
共有コンポーネントに対する依存関係 , 48
製品コンポーネントの依存関係 , 54
略称 , 28, 46
DirServer-base のパス (Directory Server 6.0), 103, 120

F

FastInfoSet、「FIS」を参照
FIS 共有コンポーネント
依存関係の表 , 48
フルネーム , 29

H

HADB-base のパス , 167
High Availability Session Store (HADB)
アップグレード手順 , 34
共有コンポーネントに対する依存関係 , 48
製品コンポーネントの依存関係 , 54
略称 , 28, 46

I

ICU 共有コンポーネント
依存関係の表 , 48
フルネーム , 29
IM-SDK 共有コンポーネント
依存関係の表 , 48
フルネーム , 29
Instant Messenger SDK、「IM-SDK」を参照
International Components for Unicode、「ICU」を参照

J

JAF 共有コンポーネント
依存関係の表 , 48
フルネーム , 29
JATO 共有コンポーネント
依存関係の表 , 48
フルネーム , 29
Java 2 Platform, Standard Edition、「J2SE」を参照
Java 2 Platform, Standard Edition、「Java SE」を参照
Java API for XML Processing、「JAXP」を参照
Java API for XML Registries、「JAXR」を参照
Java API for XML-based Remote Procedure Call、「JAX-RPC」を参照
Java Architecture for XML Binding、「JAXB」を参照
Java Calendar API、「JCAPI」を参照
Java DB
アップグレード手順 , 34
共有コンポーネントに対する依存関係 , 48
製品コンポーネントの依存関係 , 54
略称 , 28, 46
Java Dynamic Management Kit、「JDMK」を参照
Java ES 2003Q4 (Release 1)
共有コンポーネント , 411
製品コンポーネント , 410
Java ES 2004Q2 (Release 2)
共有コンポーネント , 413
製品コンポーネント , 412
Java ES 2005Q1 (Release 3)
共有コンポーネント , 418
製品コンポーネント , 415
Java ES 2005Q4 (Release 4)
共有コンポーネント , 423
製品コンポーネント , 420
Java ES 5 (Release 5)
共有コンポーネント , 428

製品コンポーネント , 425
Java ES Monitoring Framework、「MFWK」を参照
Java SE 共有コンポーネント
依存関係の表 , 48
フルネーム , 29
Java Security Services、「JSS」を参照
Java Studio Web Application Framework、「JATO」を参照
JAVA_HOME, 346, 357, 359, 384, 387, 403, 405
JAVA_HOME 環境変数 , 276, 299, 322, 325
JavaBeans Activation Framework、「JAF」を参照
JavaHelp 共有コンポーネント
依存関係の表 , 48
フルネーム , 29
JavaMail 共有コンポーネント
依存関係の表 , 48
フルネーム , 30
JAXB 共有コンポーネント
依存関係の表 , 48
フルネーム , 30
JAXP 共有コンポーネント
依存関係の表 , 48
フルネーム , 30
JAXR 共有コンポーネント
依存関係の表 , 48
フルネーム , 30
JAX-RPC 共有コンポーネント
依存関係の表 , 48
フルネーム , 30
JAXWS 共有コンポーネント
依存関係の表 , 49
フルネーム , 30
JCAPI 共有コンポーネント
依存関係の表 , 49
フルネーム , 30
JDMK 共有コンポーネント
依存関係の表 , 49
フルネーム , 30
JSP Standard Library Template、「JSTL」を参照
JSP ファイル、カスタマイズ , 261, 268, 273, 297
JSS 共有コンポーネント
依存関係の表 , 49
フルネーム , 30
JSTL 共有コンポーネント
依存関係の表 , 49
フルネーム , 30

K

KT Search Engine、「KTSE」を参照
KTSE 共有コンポーネント
依存関係の表 , 49
フルネーム , 30

L

LDAP C Language SDK 共有コンポーネント , 428
LDAP C SDK 共有コンポーネント
依存関係の表 , 49
フルネーム , 30
LDAP J SDK 共有コンポーネント , 428
依存関係の表 , 49
フルネーム , 30

M

MA Core 共有コンポーネント
依存関係の表 , 49
フルネーム , 30
Message Queue
アップグレード手順 , 34
共有コンポーネントに対する依存関係 , 48
製品コンポーネントの依存関係 , 55
略称 , 28, 46
MessageQueue-base のパス , 179
MFWK 共有コンポーネント
依存関係の表 , 49

フルネーム , 30
Mobile Access コア、「MA コア」を参照
Monitoring Console
共有コンポーネントに対する依存関係 , 48
製品コンポーネントの依存関係 , 55
略称 , 29
mqupgrade スクリプト (Message Queue), 197, 199

N

Netscape Portable Runtime、「NSPR」を参照
Network Security Services、「NSS」を参照
NSPR 共有コンポーネント
依存関係の表 , 49
フルネーム , 30
NSS 共有コンポーネント
依存関係の表 , 49
フルネーム , 30

P

Portal Server
アップグレード手順 , 35
共有コンポーネントに対する依存関係 , 48
製品コンポーネントの依存関係 , 56
略称 , 29, 46
Portal Server Secure Remote Access
アップグレード手順 , 35
共有コンポーネントに対する依存関係 , 48
製品コンポーネントの依存関係 , 56
略称 , 29, 46
PortalServer6-base パス , 305, 372
PortalServer6Config-base パス , 305, 372
PortalServer7-base パス , 306, 372
PortalServer7Config-base パス , 306, 372
PortalServer7Data-base パス , 306, 372
postInstall スクリプト (ApplicationServer), 223

R

RegistryDomainR4-base パス , 229
RegistryDomainR5-base パス , 229
rel4CAC-base-dir パス , 75
rel5CAC-admin-dir パス , 75

S

SAAJ 共有コンポーネント
依存関係の表 , 49
フルネーム , 30
SAML, 281, 303
SASL 共有コンポーネント
依存関係の表 , 49
フルネーム , 30
Security Assertion Markup Language、「SAML」を参照
SEDC 共有コンポーネント
依存関係の表 , 49
フルネーム , 30
serverRoot のパス (Directory Server 5.x), 103, 119
Service Registry
アップグレード手順 , 35
共有コンポーネントに対する依存関係 , 48
製品コンポーネントの依存関係 , 57
略称 , 29, 46
ServiceRegistryR4-base パス , 157, 229
ServiceRegistryR5-base パス , 229
Simple Authentication and Security Layer、「SASL」を参照
SJWC 共有コンポーネント
依存関係の表 , 49
フルネーム , 30
SML Web Services Security、「XWSS」を参照
SOAP Runtime with Attachments API for Java、「SAAJ」を参照
Sun Cluster
アップグレード手順 , 35
共有コンポーネントに対する依存関係 , 48

製品コンポーネントの依存関係 , 57
略称 , 29, 46
Sun Cluster Geographic Edition
アップグレード手順 , 35
共有コンポーネントに対する依存関係 , 48
製品コンポーネントの依存関係 , 57
略称 , 29, 46
Sun Explorer Data Collector、「SEDC」を参照
Sun Java Web Console、「SJWC」を参照
SunClusterGeo-base パス , 91

W

Web Proxy Server
アップグレード手順 , 35
共有コンポーネントに対する依存関係 , 48
製品コンポーネントの依存関係 , 57
略称 , 29, 46
Web Server
アップグレード手順 , 35
共有コンポーネントに対する依存関係 , 48
製品コンポーネントの依存関係 , 58
略称 , 29, 46
Web services Common Library、「WSCL」を参照
WebProxyServer-base のパス , 241
WebServer6-base パス , 133
WebServer7-base パス , 133
WebServer7Config-base パス , 133
WSCL 共有コンポーネント
依存関係の表 , 49
フルネーム , 30

X

XWSS 共有コンポーネント
依存関係の表 , 49
フルネーム , 30

い

依存関係
共有コンポーネント間 , 51
製品コンポーネント間 , 53
製品コンポーネント、共有コンポーネントに対する , 48

お

オペレーティングシステム :
計画の考慮事項 , 44
サポートマトリックス , 44
二重のアップグレード , 44
パッチ , 44

き

記号の表記規則 , 21
規則
記号 , 21
表記 , 20
共通エージェントコンテナ、「CAC」を参照
共有コンポーネント
Java ES 2003Q4 (Release 1), 411
Java ES 2004Q2 (Release 2), 413
Java ES 2005Q1 (Release 3), 418
Java ES 2005Q4 (Release 4), 423
Java ES 5 (Release 5), 428
依存する製品コンポーネント , 48
相互依存関係 , 51

す

スクリプト
amconfig (Access Manager), 274
ampre71upgrade (Access Manager), 271, 295
amupgrade (Access Manager), 278, 302
mqmigrate (Message Queue), 199
mqupgrade (Message Queue), 197

postInstall (Application Server), 223

せ

製品コンポーネント
Java ES 2003Q4 (Release 1), 410
Java ES 2004Q2 (Release 2), 412
Java ES 2005Q1 (Release 3), 415
Java ES 2005Q4 (Release 4), 420
Java ES 5 (Release 5), 425
共有コンポーネントに対する依存関係 , 48
相互依存関係 , 53

に

二重のアップグレード
コンポーネントサポートの概要 , 46
定義 , 45
手順 , 45

ひ

表記上の規則 , 20