



# BEA WebLogic Server™ およ び WebLogic Express™

リリース ノート

バージョン 7.0 サービス パック 7  
マニュアルの改訂 : 2006 年 3 月

## 著作権

Copyright © 2004, BEA Systems, Inc. All Rights Reserved.

## 限定的権利条項

本ソフトウェアおよびマニュアルは、BEA Systems, Inc. 又は日本ビー・イー・イー・システムズ株式会社（以下、「BEA」といいます）の使用許諾契約に基づいて提供され、その内容に同意する場合にのみ使用することができ、同契約の条項通りにのみ使用またはコピーすることができます。同契約で明示的に許可されている以外の方法で同ソフトウェアをコピーすることは法律に違反します。このマニュアルの一部または全部を、BEA からの書面による事前の同意なしに、複写、複製、翻訳、あるいはいかなる電子媒体または機械可読形式への変換も行うことはできません。

米国政府による使用、複製もしくは開示は、BEA の使用許諾契約、および FAR 52.227-19 の「Commercial Computer Software-Restricted Rights」条項のサブパラグラフ (c)(1)、DFARS 252.227-7013 の「Rights in Technical Data and Computer Software」条項のサブパラグラフ (c)(1)(ii)、NASA FAR 補遺 16-52.227-86 の「Commercial Computer Software--Licensing」条項のサブパラグラフ (d)、もしくはそれらと同等の条項で定める制限の対象となります。

このマニュアルに記載されている内容は予告なく変更されることがあり、また BEA による責務を意味するものではありません。本ソフトウェアおよびマニュアルは「現状のまま」提供され、商品性や特定用途への適合性を始めとする（ただし、これらには限定されない）いかなる種類の保証も与えません。さらに、BEA は、正当性、正確さ、信頼性などについて、本ソフトウェアまたはマニュアルの使用もしくは使用結果に関していかなる確約、保証、あるいは表明も行いません。

## 商標または登録商標

BEA、Jolt、Tuxedo、および WebLogic は BEA Systems, Inc. の登録商標です。BEA Builder、BEA Campaign Manager for WebLogic、BEA eLink、BEA Liquid Data for WebLogic、BEA Manager、BEA WebLogic Commerce Server、BEA WebLogic Enterprise、BEA WebLogic Enterprise Platform、BEA WebLogic Express、BEA WebLogic Integration、BEA WebLogic Personalization Server、BEA WebLogic Platform、BEA WebLogic Portal、BEA WebLogic Server、BEA WebLogic Workshop および How Business Becomes E-Business は、BEA Systems, Inc の商標です。

その他の商標はすべて、関係各社がその権利を有します。

# 目次

## このマニュアルの内容

対象読者 .....	xv
e-docs Web サイト .....	xvi
このマニュアルの印刷方法 .....	xvi
サポート情報 .....	xvi
表記規則 .....	xvii

## 1. WebLogic Server 7.0 の機能と変更点

WebLogic Server 7.0 SP7 での新機能 .....	1-2
Administration Console .....	1-2
プラグイン .....	1-2
WebLogic Server 7.0 SP6 での新機能 .....	1-3
Administration Console .....	1-3
WebLogic Server 7.0 SP5 での新機能 .....	1-4
Oracle 10g JDBC Thin ドライバ .....	1-4
JDBC マルチプールのフェイルオーバー機能の強化 .....	1-5
JDBC 接続プールのテスト機能の強化 .....	1-5
EJB キャッシュ サイズの削除 .....	1-6
ドメイン コンフィグレーション用の Ant タスク .....	1-6
コンフィグレーション可能なログ ローテーションの基準 .....	1-7
強化されたサーバ ログ .....	1-7
クライアントの追跡 .....	1-7
カスタム監査プロバイダ .....	1-7
新しいセキュリティ API .....	1-8

weblogic.Admin および Ant タスク用のユーザ コンフィグレーションおよびユーザ キー ファイル .....	1-8
メッセージ駆動型 Bean の休止と再開 .....	1-8
WebDAV のサポート .....	1-8
WebLogic Server 7.0 SP4 での新機能 .....	1-9
WebLogic Server 7.0 SP3 での新機能 .....	1-9
WebLogic Server 7.0 サービス パック 3 と WebLogic Workshop .....	1-10
プールされた接続から JDBC ベンダ接続へのアクセス .....	1-10
XA JDBC 文キャッシングの機能強化 .....	1-11
JDBC 接続リーク検出の機能強化 .....	1-11
Oracle 仮想プライベート データベースのサポート .....	1-11
JTA トランザクション ログの書き込みポリシー .....	1-12
XA リソースのトランザクション タイムアウトのサポート .....	1-12
パッシベーションされた EJB のディスクからの削除 .....	1-14
Web アプリケーションの context-root のネーミング .....	1-14
Web サービスの WSDL とホーム ページへのアクセスの制御 .....	1-14
セキュリティ レルム オプションの変更 .....	1-15
WebLogic Server 7.0 SP2 での新機能 .....	1-15
アップグレード インストーラに関する警告 .....	1-16
コンフィグレーション ウィザードを使用して作成されたドメインの移行 .....	1-16
WebLogic Server 7.0 から 7.0 サービス パック 2 への更新 .....	1-17
パフォーマンスの強化 .....	1-18
セキュリティの新規および更新マニュアル .....	1-19
Web サービス .....	1-19
JMS トピック サブスクライバのメッセージ セレクタの最適化 .....	1-20
サーバ レベルの追加 EJB コンパイラ オプション .....	1-20
WebLogic Server を Windows サービスとしてインストールする場合の新機能 ..	1-21
Oracle Thin Driver の新しいデフォルト バージョン .....	1-21
WebLogic Server 7.0 SP1 での新機能 .....	1-21

JMS.....	1-22
JMS 仕様 1.0.2b の完全な準拠.....	1-22
JMS ファイル ストアの同期書き込みポリシー.....	1-22
サブレット.....	1-22
Web サービス.....	1-23
WebLogic jDriver for Oracle 9.2.0.....	1-24
EJB の新しいセキュリティ関連要素.....	1-24
WebLogic Server 7.0 Platform Edition と WebLogic Server 7.0 の スタンドアロン版.....	1-25
WebLogic Server 7.0 から 7.0 サービス パック 1 への更新.....	1-25
コンフィグレーション ウィザードを使用して作成されたドメインの移行.....	1-26
WebLogic Server 7.0 での新機能.....	1-31
Web サービス.....	1-32
セキュリティ インフラストラクチャ.....	1-32
開発者用ツール.....	1-33
システム管理の強化.....	1-33
キャッシング.....	1-34
クラスタ化.....	1-35
EJB 2.0.....	1-35
メッセージ駆動型 Bean と非 BEA JMS プロバイダ.....	1-35
ディレクトリ構造.....	1-36
コンフィグレーション ウィザード.....	1-36
インストール.....	1-36
J2EE コネクタ.....	1-37
jCOM.....	1-38
JDBC.....	1-39
JMS.....	1-40
BEA Jolt Client 8.0.1.....	1-42
Java Transaction API (JTA).....	1-42
ネットワーク リソース.....	1-42

RMI-IIOP .....	1-43
Web アプリケーションおよび EJB のセキュリティ コンフィグレーションに 関する更新された手順 .....	1-44
WebLogic Tuxedo Connector .....	1-44
WebLogic XML .....	1-44
ドキュメントとサンプル .....	1-45
プラットフォームのサポート .....	1-47
標準とライブラリに関するバージョンの互換性 .....	1-47
Java Development Kit .....	1-49
WebLogic Javadoc .....	1-50
独自の MBean の作成 .....	1-50
利用可能な他のリソース .....	1-51
短時間で実行可能な手順 .....	1-51
Pet Store 1.3 .....	1-51
サンプル .....	1-51
BEA WebLogic Server 7.0 の紹介 .....	1-51
その他のドキュメント .....	1-52
ニュースグループ .....	1-52
Dev2Dev Online .....	1-52

## 2. 確認済みの問題

WebLogic Server 7.0 サービス パック 7 の確認済みの問題 .....	2-1
コア サーバに関する確認済みの問題 .....	2-2
EJB に関する確認済みの問題 .....	2-2
RMI に関する確認済みの問題 .....	2-3
WebLogic Server 7.0 サービス パック 6 の確認済みの問題 .....	2-3
Administration Console に関する確認済みの問題 .....	2-4
JDBC に関する確認済みの問題 .....	2-4
プラグインに関する確認済みの問題 .....	2-5
WebLogic Server 7.0 サービス パック 5 の確認済みの問題 .....	2-6

コア サーバに関する確認済みの問題 .....	2-6
クラスタに関する確認済みの問題 .....	2-7
ドキュメントに関する確認済みの問題 .....	2-8
EJB に関する確認済みの問題 .....	2-9
デプロイメントに関する確認済みの問題 .....	2-10
JDBC に関する確認済みの問題 .....	2-10
JMS に関する確認済みの問題 .....	2-10
セキュリティに関する確認済みの問題 .....	2-11
WebLogic Tuxedo Connector に関する確認済みの問題 .....	2-12
Web サービスに関する確認済みの問題 .....	2-12
WebLogic Server 7.0 サービス パック 4 の確認済みの問題 .....	2-13
jDriver に関する確認済みの問題 .....	2-13
JSP に関する確認済みの問題 .....	2-14
Web サービスに関する確認済みの問題 .....	2-14
WebLogic Server 7.0 サービス パック 3 の確認済みの問題 .....	2-15
ブラウザに関する確認済みの問題 .....	2-15
EJB に関する確認済みの問題 .....	2-17
ツールに関する確認済みの問題 .....	2-18
Web サービスに関する確認済みの問題 .....	2-19
WebLogic Server 7.0 サービス パック 2 の確認済みの問題 .....	2-20
HPUX 11i ユーザはパッチの適用が必要 .....	2-20
JDK 1.4.1 のサポートと SSL に関する確認済みの問題 .....	2-20
Administration Console に関する確認済みの問題 .....	2-21
コア サーバに関する確認済みの問題 .....	2-22
EJB に関する確認済みの問題 .....	2-29
サンプルと Pet Store に関する確認済みの問題 .....	2-31
インストールに関する確認済みの問題 .....	2-33
相互運用性に関する確認済みの問題 .....	2-34
jDriver に関する確認済みの問題 .....	2-35

JMS に関する確認済みの問題 .....	2-35
JRockit に関する確認済みの問題 .....	2-36
JTA に関する確認済みの問題 .....	2-37
ノード マネージャに関する確認済みの問題 .....	2-38
プラグインに関する確認済みの問題.....	2-40
RMI に関する確認済みの問題 .....	2-40
セキュリティに関する確認済みの問題.....	2-41
サードパーティ製ツールに関する確認済みの問題.....	2-42
ツールに関する確認済みの問題 .....	2-43
WebLogic Tuxedo コネクタに関する確認済みの問題.....	2-46
Web サービスに関する確認済みの問題 .....	2-48
XML に関する確認済みの問題 .....	2-50
WebLogic Server 7.0 サービス パック 1 の確認済みの問題 .....	2-50
Administration Console に関する確認済みの問題 .....	2-50
コア サーバに関する確認済みの問題.....	2-51
EJB に関する確認済みの問題 .....	2-59
サンプルと Pet Store に関する確認済みの問題 .....	2-61
インストールに関する確認済みの問題.....	2-64
JMS に関する確認済みの問題 .....	2-64
ノード マネージャに関する確認済みの問題 .....	2-65
RMI に関する確認済みの問題 .....	2-69
セキュリティに関する確認済みの問題.....	2-70
ツールに関する確認済みの問題 .....	2-78
WebLogic Tuxedo コネクタに関する確認済みの問題.....	2-81
Web サービスに関する確認済みの問題 .....	2-83
XML に関する確認済みの問題.....	2-85
WebLogic Server 7.0 の確認済みの問題.....	2-86
コネクタに関する確認済みの問題 .....	2-86
コア サーバに関する確認済みの問題.....	2-86



EJB に関する確認済みの問題 .....	2-87
JDBC に関する確認済みの問題 .....	2-88
RMI に関する確認済みの問題 .....	2-89
ディレクトリ名のスペース .....	2-90
EJB の値渡しに関して J2EE に準拠していない点 .....	2-90
HTML ページと Netscape .....	2-90
クラスタ、DNS、および NT でのマルチホーム .....	2-91
WebLogic キーストア プロバイダのサポート .....	2-91
マルチサーバ環境の XA 非対応 JDBC ドライバに関するトランザクションの 制限事項 .....	2-91

### 3. サービス パック 7 の解決済みの問題

Administration Console .....	3-2
クラスタ .....	3-4
コア サーバ .....	3-4
デプロイメント .....	3-6
EJB .....	3-6
JDBC .....	3-7
JMS .....	3-8
JSP .....	3-10
JTA .....	3-11
JVM .....	3-12
ノード マネージャ .....	3-12
操作と管理 .....	3-13
プラグイン .....	3-14
RMI .....	3-18
セキュリティ .....	3-19
サブレット .....	3-21
Simple Network Management Protocol (SNMP) .....	3-24
Web サービス .....	3-24

XML.....	3-25
----------	------

## 4. サービス パック 1 から 6 で解決済みの問題

WebLogic Server 7.0 サービス パック 6 のソリューション .....	4-1
Administration Console .....	4-3
クラスタ .....	4-7
コネクタ .....	4-7
コア サーバ.....	4-8
デプロイメント .....	4-13
EJB .....	4-14
インストール.....	4-17
インターナショナルライゼーション .....	4-17
J2EE .....	4-18
JCOM.....	4-18
JDBC.....	4-19
jDriver .....	4-22
JMS .....	4-23
JNDI.....	4-27
JSP.....	4-27
JTA .....	4-29
JVM.....	4-30
ノード マネージャ .....	4-30
操作と管理.....	4-31
プラグイン .....	4-36
RMI.....	4-46
サンプル .....	4-47
Security .....	4-48
サーブレット.....	4-52
Simple Network Management Protocol (SNMP) .....	4-56
Web サービス .....	4-56

WLEC.....	4-58
WebLogic Tuxedo Connector.....	4-59
XML.....	4-61
WebLogic Server 7.0 サービス パック 5 のソリューション.....	4-61
Administration Console.....	4-62
クラスローダ.....	4-63
クラスタ.....	4-65
コネクタ.....	4-68
コア サーバ.....	4-69
デプロイメント.....	4-83
EJB.....	4-86
JCOM.....	4-97
JDBC.....	4-98
jQuery.....	4-104
JMS.....	4-105
JSP.....	4-116
JTA.....	4-120
ノード マネージャ.....	4-121
操作と管理.....	4-122
プラグイン.....	4-127
RMI.....	4-138
セキュリティ.....	4-141
サーブレット.....	4-152
SNMP (Simple Network Management Protocol).....	4-165
Web アプリケーション.....	4-166
WebLogic Tuxedo Connector.....	4-167
Web サービス.....	4-170
WebLogic Server 7.0 サービス パック 4 のソリューション.....	4-175
Administration Console.....	4-176
クラスタ.....	4-177

コネクタ .....	4-177
EJB .....	4-178
JDBC .....	4-179
JMS .....	4-180
JSP とサーブレット .....	4-181
JTA .....	4-181
ノード マネージャ .....	4-182
セキュリティ .....	4-182
ツール .....	4-183
Web サービス .....	4-184
WebLogic Server 7.0 サービス パック 3 のソリューション .....	4-186
Administration Console .....	4-187
クラスタ .....	4-192
コネクタ .....	4-195
コア サーバ.....	4-198
デプロイメント .....	4-208
ドメイン コンフィグレーション ウィザード .....	4-210
EJB .....	4-210
相互運用性.....	4-225
JDBC .....	4-225
jDriver .....	4-232
JMS .....	4-234
JNDI.....	4-245
JSP とサーブレット .....	4-247
JTA .....	4-261
操作と管理.....	4-266
プラグイン.....	4-266
RMI/RMI-IIOP .....	4-269
セキュリティ .....	4-272
システム管理.....	4-280

ツール .....	4-288
Web サービス .....	4-293
WebLogic Tuxedo Connector.....	4-299
XML .....	4-300
WebLogic Server 7.0 サービス パック 2 のソリューション.....	4-303
Administration Console.....	4-304
クラスローダ .....	4-306
クラスタ .....	4-307
コネクタ .....	4-310
コア サーバ.....	4-310
デプロイ .....	4-314
EJB .....	4-315
サンプル.....	4-319
インストーラ .....	4-320
インターナショナルライゼーション .....	4-320
jCOM.....	4-320
JDBC .....	4-321
jDriver .....	4-322
JMS.....	4-323
JNDI .....	4-326
JTA .....	4-327
JTS.....	4-327
その他 .....	4-327
ノード マネージャ .....	4-330
プラグイン .....	4-330
RMI/RMI-IIOP.....	4-332
セキュリティ .....	4-333
サーブレットと JSP .....	4-336
ツール .....	4-342
WebLogic Tuxedo Connector.....	4-344

Web サービス .....	4-345
XML.....	4-346
WebLogic Server 7.0 サービス パック 1 のソリューション .....	4-348
コンソール.....	4-349
コア サーバ.....	4-350
デプロイ .....	4-352
EJB .....	4-353
サンプル .....	4-356
インターナショナルライゼーション .....	4-357
JDBC.....	4-357
JMS .....	4-358
JNDI.....	4-359
JTA .....	4-359
JTS .....	4-359
その他 .....	4-359
RMI/RMI-IIOP .....	4-365
セキュリティ .....	4-367
サーブレットと JSP.....	4-371
ツール .....	4-375
WebLogic Tuxedo Connector .....	4-377
Web サービス .....	4-377

# このマニュアルの内容

このマニュアルでは、BEA WebLogic Server™ の最新リリースを紹介します。新機能、サポートされる仕様、および確認済みの問題についての重要な情報を提供します。

マニュアルの内容は以下のとおりです。

- 第1章「[WebLogic Server 7.0 の機能と変更点](#)」では、WebLogic Server 7.0 と付属のサービスパックに関する重要な情報の概要を示します。
- 第2章「[確認済みの問題](#)」では、WebLogic Server 7.0 およびそのサービスパックにおける確認済みの問題のリストを示します。
- 第3章「[サービスパック 7 の解決済みの問題](#)」では、WebLogic Server 7.0 サービスパック 6 で解決済みのソフトウェアとマニュアルに関する問題のリストを示します。
- 第4章「[サービスパック 1 から 6 で解決済みの問題](#)」では、WebLogic Server 7.0 サービスパック 1 から 6 で解決済みのソフトウェアとマニュアルに関する問題のリストを示します。

## 対象読者

このマニュアルは、WebLogic Server 7.0 を使用するユーザを対象としています。

## e-docs Web サイト

BEA 製品のドキュメントは、BEA の Web サイトで入手できます。BEA のホーム ページで [製品のドキュメント] をクリックします。

## このマニュアルの印刷方法

Web ブラウザの [ファイル | 印刷] オプションを使用すると、Web ブラウザからこのマニュアルを一度に 1 章ずつ印刷できます。

このマニュアルの PDF 版は、WebLogic Server の Web サイトで入手できます。PDF を Adobe Acrobat Reader で開くと、マニュアルの全体（または一部分）を書籍の形式で印刷できます。PDF を表示するには、WebLogic Server ドキュメントのホーム ページを開き、[ドキュメントのダウンロード] をクリックして、印刷するマニュアルを選択します。

Adobe Acrobat Reader は Adobe の Web サイト (<http://www.adobe.co.jp>) で無料で入手できます。

## サポート情報

BEA のドキュメントに関するユーザからのフィードバックは弊社にとって非常に重要です。質問や意見などがあれば、電子メールで [docsupport-jp@beasys.com](mailto:docsupport-jp@beasys.com) までお送りください。寄せられた意見については、ドキュメントを作成および改訂する BEA の専門の担当者が直に目を通します。

電子メールのメッセージには、ご使用のソフトウェアの名前とバージョン、およびドキュメントのタイトルと日付をお書き添えください。本バージョンの BEA WebLogic Server について不明な点がある場合、または BEA WebLogic Server のインストールおよび動作に問題がある場合は、BEA WebSupport ([www.bea.com](http://www.bea.com)) を通じて BEA カスタマ サポートまでお問い合わせください。カスタマ サポートへの連絡方法については、製品パッケージに同梱されているカスタマ サポート カードにも記載されています。



カスタマ サポートでは以下の情報をお尋ねしますので、お問い合わせの際はあらかじめご用意ください。

- お名前、電子メール アドレス、電話番号、ファクス番号
- 会社の名前と住所
- お使いの機種とコード番号
- 製品の名前とバージョン
- 問題の状況と表示されるエラー メッセージの内容

## 表記規則

このマニュアルでは、全体を通して以下の表記規則が使用されています。

表記法	適用
[Ctrl] + [Tab]	複数のキーを同時に押すことを示す。
<i>斜体</i>	強調または書籍のタイトルを示す。
等幅テキスト	コード サンプル、コマンドとそのオプション、データ構造体とそのメンバー、データ型、ディレクトリ、およびファイル名とその拡張子を示す。等幅テキストはキーボードから入力するテキストも示す。 例： <pre>import java.util.Enumeration; chmod u+w * config/examples/applications .java config.xml float</pre>
<i>斜体の等幅テキスト</i>	コード内の変数を示す。 例： <pre>String CustomerName;</pre>

表記法	適用
すべて大文字のテキスト	デバイス名、環境変数、および論理演算子を示す。 例： LPT1 BEA_HOME OR
{ }	構文の中で複数の選択肢を示す。
[ ]	構文の中で任意指定の項目を示す。 例： <pre>java utils.MulticastTest -n name -a address [-p portnumber] [-t timeout] [-s send]</pre>
	構文の中で相互に排他的な選択肢を区切る。 例： <pre>java weblogic.deploy [list deploy undeploy update] password {application} {source}</pre>
...	コマンドラインで以下のいずれかを示す。 <ul style="list-style-type: none"><li>• 引数を複数回繰り返すことができる。</li><li>• 任意指定の引数が省略されている。</li><li>• パラメータや値などの情報を追加入力できる。</li></ul>
.	コード サンプルまたは構文で項目が省略されていることを示す。 . .

# WebLogic Server 7.0 の機能と 変更点

BEA WebLogic Server 7.0 サービス パック 7 へようこそ。先進的な Web アプリケーションサーバである WebLogic Server<sup>TM</sup> は、J2EE 1.3 技術、Web サービス、およびその他の最先端インターネット標準を実装し、可用性、拡張性、安全性に優れたアプリケーションのための信頼できるフレームワークを提供します。WebLogic Server を利用するとさまざまな異種プラットフォームとアプリケーションがシームレスに統合されるため、ネットワークで既存のソフトウェア投資を活用したり、ミッションクリティカルな e-ビジネス アプリケーションの構築に不可欠なエンタープライズクラスのサービスとデータを共有したりすることが可能になります。

WebLogic Server のドキュメント (リリース ノートを含む) は、<http://edocs.beasys.co.jp/e-docs/index.html> で頻繁に更新されます。

非推奨になった機能と API については、『WebLogic Server 7.0 へのアップグレード』の「[非推奨になった API と機能](#)」を参照してください。

以下の節では、WebLogic Server 7.0 の一般リリースとそれに関連するサービス パックで導入された新機能および主要な改善点について説明します。

- [1-2 ページの「WebLogic Server 7.0 SP7 での新機能」](#)
- [1-3 ページの「WebLogic Server 7.0 SP6 での新機能」](#)
- [1-4 ページの「WebLogic Server 7.0 SP5 での新機能」](#)
- [1-9 ページの「WebLogic Server 7.0 SP4 での新機能」](#)
- [1-9 ページの「WebLogic Server 7.0 SP3 での新機能」](#)

- 1-15 ページの「WebLogic Server 7.0 SP2 での新機能」
- 1-21 ページの「WebLogic Server 7.0 SP1 での新機能」
- 1-31 ページの「WebLogic Server 7.0 での新機能」
- 1-47 ページの「プラットフォームのサポート」
- 1-47 ページの「標準とライブラリに関するバージョンの互換性」
- 1-49 ページの「Java Development Kit」
- 1-50 ページの「WebLogic Javadoc」
- 1-50 ページの「独自の MBean の作成」
- 1-51 ページの「利用可能な他のリソース」

## WebLogic Server 7.0 SP7 での新機能

WebLogic Server 7.0 SP7 では、以下の重要な拡張機能が提供されます。

### Administration Console

WebLogic Server を Windows サービスとして実行した場合、リモートのマップされたドライブにあるアプリケーションを Console からデプロイすることはできませんでした。

[ コンフィグレーションするアプリケーションまたはコンポーネントの場所を指定 ] ページに [set new Location] オプションが追加されたため、UNC パスを使用して、Windows のマップされたネットワーク ドライブからアプリケーションをデプロイできるようになりました。

### プラグイン

- NSAPI プラグインで、チャンクにまとめられたリクエストの処理に関するサポートが追加されました。

Sun One Web サーバでは、`magnus.conf` ファイルで `ChunkedRequestBufferSize` プロパティが 0 に設定されている場合にのみ、着信するリクエストのチャンクの分割は行われません。

- IIS プラグインを使用している場合、WebLogic Server にホストされている大きなサイズの静的ファイルをダウンロードすると、速度が大幅に低下していました。20MB より大きなファイルの場合、複数の TCP パケットがクライアントに送信されていたため、パフォーマンスが低下しました。

部分的に処理される TCP パケットの数を減らすために、応答バッファ サイズを増やしました。

## WebLogic Server 7.0 SP6 での新機能

BEA では、WebLogic Server の問題を解決して新しい機能を導入する WebLogic Server 7.0 サービス パックを提供することにより、WebLogic Server 7.0 のユーザを引き続きサポートします。

### Administration Console

サーバとクラスタのコンフィグレーションに [Servlet Extension Case Sensitive] 属性が追加されました。また、セキュリティ ドメイン コンフィグレーションには [Web App Files Case Insensitive] 属性が追加されました。

これらの属性では、win32 以外のすべてのプラットフォームにおいて、Java Server Pages (JSP) のファイルルックアップで大文字 / 小文字を区別するかどうかを指定します。標準の win32 ファイル システムにおけるファイルルックアップでは、常に大文字 / 小文字は区別されません。win 32 以外の大文字 / 小文字を区別しないファイル システム (UNIX の NT Samba マウントや、大文字 / 小文字を区別しないモードでインストールされた Mac OS など) で、大文字 / 小文字を区別しないルックアップを指定すると (これらの属性を false に設定する)、JSP はそのソース コードを返さなくなります。たとえば、JSP が Samba マウントから提供されていて、大文字 / 小文字を区別しないルックアップを指定した場合、WebLogic Server では、すべてのファイル名の拡張子を小文字に変換してから JSP をルックアップします。

詳細については、

<http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA04-67.00.jsp> のセキュリティ勧告を参照してください。

## WebLogic Server 7.0 SP5 での新機能

WebLogic Server 7.0 SP5 では、重要な拡張機能が提供されます。

### Oracle 10g JDBC Thin ドライバ

WebLogic Server 7.0SP5 では、Oracle Thin ドライバの Oracle 10g (10.1.0.2.0) バージョンがリリースに追加されて、Oracle Thin ドライバのデフォルトのバージョンになりました。『WebLogic JDBC プログラマーズ ガイド』の「[WebLogic Server でのサードパーティ ドライバの使い方](#)」を参照してください。

**注意：** Oracle 10g Thin ドライバで確認された問題は次のとおりです。

- 日本語文字を含むマルチバイト文字セットを使用する場合、`setString()` メソッドの呼び出しが失敗します (Oracle TAR 3584729.994、Oracle Bug3437365)。WebLogic Server 7.0 SP5 に付属している Oracle 10g Thin ドライバのバージョンには、この問題に対するパッチが含まれています (WebLogic Server 7.0 SP5 は、キットに付属している、このパッチを含んだ Oracle10g ドライバでの動作が確認されています)。Oracle 10g Thin ドライバの GA バージョンを使用する場合は、Oracle から関連するパッチを入手することを強くお勧めします。
- Oracle 10g Thin ドライバには操作数 1682 というバッチ サイズの制限があります。以前のバージョンの Oracle Thin ドライバには明確なサイズ制限はありませんでした。アプリケーションで非常に大きなバッチ処理を実行する場合、バッチに障害が発生する可能性があります。この問題を回避するには、バッチ サイズを 1682 に制限するか、Oracle Thin ドライバの 9.2.0 バージョンを使用してください。Oracle は 5 から 30 までのバッチ サイズを推奨しています。

## JDBC マルチプールのフェイルオーバー機能の強化

WebLogic Server 7.0SP5 では、JDBC マルチプールで次のような機能が強化されました。

- マルチプール内で無効化されている接続プールに接続を要求しないようにするための、接続要求の転送の改良。
- マルチプール内の失敗した接続プールが回復したときの自動フェイルバック。
- 高可用性アルゴリズムを備えたマルチプール内のビジー状態の接続プールに対するフェイルオーバー。
- 高可用性アルゴリズムを備えたマルチプールのフェイルオーバー コールバック。
- マルチプール (アルゴリズムはどちらでも同じ) のフェイルバック コールバック。

詳細については、『WebLogic JDBC プログラマーズ ガイド』の「[マルチプールのフェイルオーバーの拡張](#)」を参照してください。

## JDBC 接続プールのテスト機能の強化

WebLogic Server 7.0SP5 では、プールされた接続に対するデータベース接続テストの機能を改良し、接続要求処理の遅延を最小限に抑えるために、JDBC 接続プールに次の機能が追加されました。

- `CountOfTestFailuresTillFlush`— 指定する回数のテストが失敗した後で接続プール内のすべての接続を閉じ、それ以上データベース テストを行うことで発生する遅延を最小限に抑えます。
- `CountOfRefreshFailuresTillDisable`— 指定する回数のテストが失敗した後で接続プールを無効にし、データベースの障害発生後に接続要求を処理する際の遅延を最小限に抑えます。

これらの機能を有効にするには、`config.xml` ファイルで `JDBCConnectionPool` オブジェクトに属性を追加します。両方の属性では、`TestConnectionsOnReserve` を `true` に設定し、`TestTableName` の値を指定することも必要です。『WebLogic JDBC プログラマーズ ガイド』の「[JDBC 接続プールのテスト機能の強化](#)」を参照してください。

## EJB キャッシュ サイズの削除

アイドル状態の読み書き対応エンティティ Bean を EJB キャッシュから自動的に削除するようにコンフィグレーションできるようになりました。自動削除は、基準のメモリ占有量が大幅に増えることを防ぎます。

キャッシュ サイズの削除をコンフィグレーションするには、`idle-timeout-seconds` 要素にゼロより大きい値を指定します。何も指定しないか 0 の値を指定する場合、自動キャッシュ サイズ削除は無効になり、アイドル状態の Bean はキャッシュから定期的に削除されません。`weblogic-ejb-jar.xml` デプロイメント記述子の `entity-cache` タグで指定される個別のエンティティ Bean キャッシュの場合は、`concurrency-strategy` が Database、ReadOnly または Optimistic に設定されているエンティティ Bean に対して既存の `idle-timeout-seconds` 要素が有効になります。

この機能は、`entity-cache` タグによる個別のエンティティ キャッシュに適用できるだけでなく、`entity-cache-ref` タグから `idle-timeout-seconds` を設定するアプリケーション キャッシュにも使用できます。

## ドメイン コンフィグレーション用の Ant タスク

WebLogic Server 7.0SP5 では、次の Ant タスクを導入して、管理者に追加の管理機能を提供しています。

- `wlserver`— 新しい単一サーバ ドメインを作成し、管理サーバを起動する
- `wlconfig`— WebLogic Server ドメインをコンフィグレーションする
- `wldeploy`— WebLogic Server ドメインへのアプリケーションを追加する

`wlserver` および `wlconfig` の詳細については、『[管理者ガイド](#)』の「[Ant タスクを使用した WebLogic Server ドメインのコンフィグレーション](#)」を参照してください。`wldeploy` の詳細については、『[WebLogic Server アプリケーションの開発](#)』の「[wldeploy Ant タスク](#)」を参照してください。



## コンフィグレーション可能なログ ローターションの基準

サービス パック 5 で起動し、WebLogic Server を Windows サービスとしてインストールしている場合は、標準出力および標準エラー ログをアーカイブするために、デフォルトの時間間隔を変更したり、他の条件を使用したりできます。『管理者ガイド』の「[ログ ファイル ローターション](#)」を参照してください。

## 強化されたサーバ ログ

管理サーバは、ユーザによってコンフィグレーションが変更されたとき、またはドメイン内のいずれかのリソースで管理操作が実行されたときにログ メッセージを送出するようにコンフィグレーションできます。たとえば、ユーザがドメイン内の管理対象サーバで SSL を無効にすると、管理サーバはログ メッセージを送出します。そのようなメッセージは、ドメインのコンフィグレーションで行われた変更の証跡となります（コンフィグレーション監査）。

## クライアントの追跡

サービス パック 5 から、恒久サブスクライバの IP アドレスのモニタが可能になりました。この情報は `JMSConnectionRuntimeMBean` で使用できます。

## カスタム監査プロバイダ

サービス パック 5 から、WebLogic Server のデフォルト セキュリティ レルムには WebLogic 監査プロバイダが含まれています。WebLogic 監査プロバイダは、WebLogic Server のドメイン コンフィグレーションの変更に関する情報を記録します。この情報には、呼び出された操作だけでなく、変更、編集、または削除された属性値も含まれます。『WebLogic Security サービスの開発』の「[カスタム監査プロバイダを開発する必要があるか](#)」を参照してください。

## 新しいセキュリティ API

これまででは、`weblogic.security.services.Authenticate.login()` を呼び出すことで `PrincipalAuthenticator.authenticate()` メソッドにアクセスできましたが、`PrincipalAuthenticator.assertIdentity()` メソッドにアクセスする方法はありませんでした。このメソッドは、ユーザが HTTP または HTTPS ヘッダ以外 (ファットクライアントの引数など) に含まれるトークンを受け取った場合に便利です。新しい API の `weblogic.security.services.Authentication.assertIdentity()` を使用すると、`PrincipalAuthenticator.assertIdentity()` にアクセスできます。

## weblogic.Admin および Ant タスク用のユーザ コンフィグレーションおよびユーザキー ファイル

WebLogic Server インスタンスに接続する `weblogic.Admin` コマンドでは、ユーザ資格を指定する必要があります。コマンドラインに資格を直接渡したり、暗号化されていない資格をスクリプトに格納したりする代わりに、新しい `STOREUSERCONFIG` コマンドを使用して、ユーザ資格を暗号化できるようになりました。

## メッセージ駆動型 Bean の休止と再開

MDB の休止と再開が可能になりました。

## WebDAV のサポート

WebLogic Server プロキシ プラグインでは、クライアントからサーバに送信できる HTTP コマンドを制限しています。プラグイン コードの検証ルールで、WebDAV 実装に必要な以下の HTTP コマンドが許可されるようになりました。

```
DELETE  
GET
```

HEAD  
OPTIONS  
POST  
PUT  
\*COPY  
LOCK  
MKCOL  
MOVE  
PROPFIND  
PROPPATCH  
SEARCH  
UNLOCK

## WebLogic Server 7.0 SP4 での新機能

WebLogic Server 7.0 サービス パック 4 (SP4) は、すべての WebLogic Platform コンポーネントでサービス パックの内容を最適化する WebLogic Platform 7.0 SP4 と内容が重なっています。WebLogic Integration 7.0、WebLogic Portal 7.0、または完全な Platform 7.0 を使用していて、SP2 より後のサービス パックによる更新を必要としているユーザは、Platform 7.0 SP4 を使用してください。

## WebLogic Server 7.0 SP3 での新機能

WebLogic Server 7.0 SP3 では、重要な拡張機能が提供されます。

## WebLogic Server 7.0 サービス パック 3 と WebLogic Workshop

WebLogic Server 7.0 サービス パック 3 (SP3) には、WebLogic Server と WebLogic Workshop のサービス パックによる更新が含まれています。WebLogic JRockit SP3 を入手して、WebLogic Server 7.0 SP3 と一緒に使用することもできます。

WebLogic Server 7.0 SP3 は、WebLogic Server および WebLogic Workshop のサポート ユーザを対象としています。WebLogic Server 7.0 SP3 は、BEA のサポート サイト (<http://support.bea.com>) からのアップグレード インストーラを通じて利用できます。WebLogic Server 7.0 SP3 アップグレード インストーラを、既存の WebLogic Integration 7.0、WebLogic Portal 7.0、または WebLogic Platform 7.0 全体に適用することはできません。

BEA では、すべての Platform コンポーネント (WebLogic Server、WebLogic Workshop、WebLogic Integration、WebLogic Portal、および WebLogic JRockit) のすべてのサービス パックを合わせた Platform 7.0 サービス パックを提供して、Platform 7.0 のユーザを引き続きサポートします。

## プールされた接続から JDBC ベンダ接続へのアクセス

サービス パック 3 では、WebLogic 接続プールのデータベース接続から、基底の JDBC ベンダ接続にアクセスすることができます。一部のベンダ固有の JDBC 拡張機能では、ベンダ接続へのアクセスが必要です。ベンダ接続が公開されると、プールされた接続は再利用のために接続プールに戻されなくなるため、ベンダ接続を使用するときはパフォーマンスの損失があります。絶対に必要な場合でない限り、ベンダ接続は使用しないことをお勧めします。

『WebLogic JDBC プログラマーズ ガイド』の「[接続プールからの物理的な接続の取得](#)」を参照してください。

## XA JDBC 文キャッシングの機能強化

XA 非対応 JDBC ドライバの代わりに XA (トランザクション対応) JDBC ドライバを使用してデータベース接続を作成する接続プールの場合には、キャッシングが異なる動作をするように、Prepared Statement キャッシュ機能が変更されました。XA Prepared Statement キャッシュの場合、WebLogic Server は最長時間未使用 (LRU) アルゴリズムを使用して、接続プールの各接続のキャッシュに格納する文を決定します。

『管理者ガイド』の「[Prepared Statement キャッシュのパフォーマンスの向上](#)」を参照してください。

## JDBC 接続リーク検出の機能強化

JDBC 接続リーク検出機能が強化されました。アプリケーションが JDBC 接続プールのデータベース接続を正しくクローズしない場合 (接続のリークと呼ばれる)、WebLogic Server は `server_names1cn.tsf` という名前のファイルをドメイン ディレクトリに自動的に作成します。ファイルには、接続がリークした JDBC 接続プール名と、接続がクローズされなかった際のスタック トレースが示されます。以降に接続リークが発生すると、WebLogic Server はそのファイルに情報を追加します。

## Oracle 仮想プライベート データベースのサポート

WebLogic Server は、Oracle 仮想プライベート データベース (VPD) をサポートするようになりました。VPD を使用することで、アプリケーション定義のファイニングレイン アクセス コントロールをサーバで実施し、Oracle 9i データベース サーバ内のアプリケーション コンテキストのセキュリティを確保できます。

詳細については、『WebLogic JDBC プログラマーズ ガイド』の「[Oracle 仮想プライベート データベースによるプログラミング](#)」を参照してください。

## JTA トランザクション ログの書き込みポリシー

WebLogic Server サービス パック 3 では、トランザクション ログにエントリを書き込む方法を選択できるように、トランザクション ログ ファイルの書き込みオプションが追加されました。選択するオプションによって、トランザクション処理中のサーバパフォーマンスに影響を与えることができます。トランザクション ログ ファイルの書き込みオプションは次のとおりです。

- Cache-Flush—(デフォルト) トランザクション ログにエントリが書き込まれるごとに、オペレーティングシステムのキャッシュとオンディスク キャッシュをフラッシュします。コミットレコードが安定ストレージに書き込まれるまで、トランザクションをコミットできません。
- Direct-Write—各書き込みで直にディスクに書き込むようにオペレーティングシステムに通知されます。

『管理者ガイド』の「[トランザクション ログ ファイル書き込みポリシーの設定](#)」を参照してください。

## XA リソースのトランザクション タイムアウトのサポート

リソース マネージャが `javax.transaction.xa.XAResource.setTimeout()` メソッドをサポートする場合、WebLogic Server トランザクション マネージャは、参加する XA リソースに対するトランザクション ブランチのタイムアウト値の設定をサポートするようになりました。XA リソースのデフォルトのタイムアウト値を超える長時間のトランザクションがある場合、トランザクション ブランチのタイムアウトを設定することができます。

WebLogic Server トランザクション マネージャで JDBC XA リソースのトランザクション タイムアウトを設定する場合、`config.xml` ファイルにある JDBC 接続プール タグで、次のプロパティの値を設定します。

- `XASetTransactionTimeout`— ブール値のプロパティ。true に設定する場合、WebLogic Server トランザクション マネージャは `XAResource.start` を呼び出す前に `XAResource.setTimeout()` を呼び出し、`XATransactionTimeout` またはグローバル トランザクション タイムアウト (秒単位) を渡します。false に設定する

と、トランザクション マネージャは `setTransactionTimeout()` を呼び出しません。デフォルト値は `false` です。

- `XATransactionTimeout`—`XAResource.setTransactionTimeout()` メソッドでトランザクション タイムアウト値として渡される秒数。このプロパティを `0` に設定すると、WebLogic Server トランザクション マネージャは、メソッドでグローバル WebLogic Server トランザクション タイムアウト (秒単位) を渡します (Administration Console オンライン ヘルプの「[JTA](#)」を参照)。このパラメータのデフォルト値は、`0` です。設定する場合、この値はグローバル WebLogic Server トランザクション タイムアウトの値以上でなければなりません。

これらのパラメータは、XA JDBC ドライバを使用してデータベース接続を作成する接続プールにのみ適用されます。XA 非対応 JDBC ドライバが使用される場合は無視されます。

これらの値を設定する場合、WebLogic Server トランザクション マネージャは上記のように `XAResource.setTransactionTimeout()` を呼び出します。XA リソース マネージャ (XA JDBC ドライバなど) または XA リソースのメソッドの実装によって、値の使用方法が決まります。たとえば、Oracle の場合、`setTransactionTimeout()` メソッドは、トランザクションの最大アイドル時間として機能するセッション タイムアウト (`SesTm`) を設定します。他の XA リソースでは動作が異なる場合があります。

`XASetTransactionTimeout` および `XATransactionTimeout` プロパティは Administration Console では使用できません。ドメインがアクティブでないときに `config.xml` ファイルに追加する必要があります。次に例を示します。

```
<JDBCConnectionPool
  DriverName="oracle.jdbc.xa.client.OracleXADataSource"
  Name="oraclePool"
  Password="{3DES}8YdvP4FQW3k="
  Properties="user=SCOTT"
  URL="jdbc:oracle:thin:@server:port:sid"
  XASetTransactionTimeout="true"
  XATransactionTimeout="120"/>
```

## パッシベーションされた EJB のディスクからの削除

新しい `weblogic-ejb-jar.xml` デプロイメント記述子要素 (`session-timeout-seconds`) は、EJB コンテナがどのくらい待機してから、アイドル状態のステートフルセッション EJB がディスクから削除されるかを指定します。『WebLogic エンタープライズ JavaBeans プログラマーズ ガイド』の「[session-timeout-seconds](#)」を参照してください。

## Web アプリケーションの context-root のネーミング

エンタープライズ アプリケーション (EAR ファイルまたはディレクトリ) の一部として Web アプリケーションをデプロイする場合、Web アプリケーション コンポーネントの名前は `application.xml` デプロイメント記述子で指定された `context-root` 値に従って付けられます。この値は `weblogic.xml` で指定される `context-root` 値または Web アプリケーションの URI よりも優先されます。

## Web サービスの WSDL とホーム ページへのアクセスの制御

`web-services.xml` デプロイメント記述子ファイルにある `<web-service>` 要素の `exposeWSDL` および `exposeHomepage` 属性を指定して、WebLogic Web サービスの WSDL およびホーム ページへのアクセスを制御できます。

「[WSDL および Web サービスのホーム ページに対してセキュリティを設定する](#)」を参照してください。



## セキュリティ レルム オプションの変更

このリリースの WebLogic Server では、セキュリティ レルム オプションが変更されました。[ デプロイメント記述子内のセキュリティ データを無視 ] オプションは、[ ロールとポリシーのチェック対象 ] および [ デプロイメント記述子のセキュリティ動作 ] オプションで置き換えられました。Administration Console オンライン ヘルプでは、これらの新しいオプションについて説明していません。これらのオプションの説明については、『WebLogic Security の管理』の「[新しいセキュリティ レルムの作成](#)」および『[WebLogic リソースのセキュリティ](#)』の「[URL \(Web\) および EJB リソースを保護する方法](#)」を参照してください。

## WebLogic Server 7.0 SP2 での新機能

以下の節は、BEA WebLogic Server 7.0 SP2 に関する重要な注意です。

- [1-16 ページの「アップグレード インストーラに関する警告」](#)
- [1-16 ページの「コンフィグレーション ウィザードを使用して作成されたドメインの移行」](#)
- [1-17 ページの「WebLogic Server 7.0 から 7.0 サービス パック 2 への更新」](#)
- [1-18 ページの「パフォーマンスの強化」](#)
- [1-19 ページの「セキュリティの新規および更新マニュアル」](#)
- [1-19 ページの「Web サービス」](#)
- [1-20 ページの「JMS トピック サブスクリバのメッセージ セレクタの最適化」](#)
- [1-20 ページの「サーバレベルの追加 EJB コンパイラ オプション」](#)
- [1-21 ページの「WebLogic Server を Windows サービスとしてインストールする場合の新機能」](#)

## アップグレード インストーラに関する警告

WebLogic Server 7.0 SP2 のアップグレード インストーラに関する以下の事項に注意してください。

- 7.0 SP1 のパッチ ファイルは 7.0 SP2 と互換性がないため、7.0 SP1 パッチ ファイルへの参照を CLASSPATH から削除すること。
- アップグレード インストーラは、パッチ ディレクトリまたはユーザが作成したドメインなど、ユーザが作成したファイルは更新しない。

## コンフィグレーション ウィザードを使用して作成されたドメインの移行

コンフィグレーション ウィザードは、新しいドメインを短時間で簡単に作成できる WebLogic Platform のツールです。WebLogic Server 7.0 または 7.0 SP1 のコンフィグレーション ウィザードで作成したドメインがあり、WebLogic Server 7.0 SP2 の環境で引き続き実行する場合、使用する前にドメインを SP2 に移行する必要があります。

ドメインを WebLogic Server 7.0 から WebLogic Server 7.0 SP2 に移行する場合は、[1-26 ページの「コンフィグレーション ウィザードを使用して作成されたドメインの移行」](#)にある 7.0 ドメインから 7.0 SP1 ドメインへの移行手順に従ってください。WebLogic Workshop ドメインがある場合は、[1-17 ページの「WebLogic Workshop ドメインの SP1 から SP2 への移行」](#)の手順にも従う必要があります。

ドメインを WebLogic Server 7.0 SP1 から SP2 に移行する場合は、[1-27 ページの「手順 1：製品 JAR ファイルをアップグレードする」](#)で説明するように、移行スクリプトを実行する必要があります。WebLogic Workshop ドメインがある場合は、[1-17 ページの「WebLogic Workshop ドメインの SP1 から SP2 への移行」](#)の手順にも従う必要があります。

**注意：** SP2 へのドメインの移行に加えて、移行スクリプトでは Sun SDK 1.3.1\_03 が Sun SDK 1.3.1\_06 にアップグレードされます。ご使用の環境で Sun SDK を使用しない場合は、スクリプトを適切に変更してください。

## WebLogic Workshop ドメインの SP1 から SP2 への移行

WebLogic Workshop 7.0 または 7.0 SP1 で作成したドメインがある場合は、startWeblogic スクリプトを Workshop ドメインに合わせて次のように修正する必要があります。

1. 次の行を追加します。

```
call%WL_HOME%\common\bin\commEnv.cmd
```

この行は、set JAVA\_DEBUG 文より前の任意の場所に追加できます。コンフィグレーション ウィザードで WebLogic Workshop テンプレートを使用して作成された新しいドメインの場合、startWeblogic スクリプトには、WL\_HOME、BEAHOME、および JAVA\_HOME 変数を設定する行のすぐ後にこの呼び出しが含まれています。次に例を示します。

```
set WL_HOME=c:\bea\weblogic700
set BEAHOME=c:\bea
set JAVA_HOME=c:\bea\jdk131_06
CALL%WL_HOME%\common\bin\commEnv.cmd
```

2. 次の行を、

```
set JAVA_DEBUG=-hotspot
```

次の行で置き換えます。

```
JAVA_DEBUG=%COMM_CLIENT_VM%
```

## WebLogic Server 7.0 から 7.0 サービス パック 2 への更新

スタンドアロンの WebLogic Server をインストール済みで、WebLogic Platform Edition にアップグレードしていない場合、Smart Update を使用して WebLogic Server 7.0 サービス パック 1 をインストールすることはできません。代わりに、BEA カスタマ サポートの Web サイト (<http://support.bea.com>) からダウンロードできるパッケージ アップグレード インストーラを使用できます。詳細については、『インストール ガイド』の「[ダウンロード可能なインストーラによるサービス パックとローリング パッチのインストール](#)」を参照してください。

どのバージョンの WebLogic Server をインストールしているか確認するには、`BEA_HOME\logs\log.txt` (UNIX の場合は `BEA_HOME/logs/log.txt`) をテキスト エディタで開き、ログ ファイルの最後のエントリを参照します。ログ ファイルの各行は、現在の

BEA ホームのインストール イベント (インストールまたはアンインストール) を表しています。ログ エントリで、インストール済みのバージョンが WebLogic Platform Edition 7.0 SP0 以前であると示されている場合、Smart Update は使用できません。ログ エントリで、インストール済みのバージョンが WebLogic Platform Edition (7.0.0.1) 以降であると示されている場合、Smart Update を使用して WebLogic Server サービス パック 2 をインストールできます。

WebLogic Server と WebLogic Platform Edition は、同じインストール フレームワークを使用します。したがって、インストールのログ エントリでは、「WebLogic Platform」という語を使用して、WebLogic Server または WebLogic Platform Edition のインストールを表します。

## パフォーマンスの強化

WebLogic Server 7.0 SP2 では、パフォーマンス関連の重要な機能が導入されています。

- OOTB (out-of-the-box) パラメータの調整。デフォルトのコンフィグレーションを使用してアプリケーションを実行する場合に、より効率的なパフォーマンスを実現できます。この調整は不要な副作用をもたらしません。
- `PreparedStatementCache` 機能はデフォルトで有効になりました。この機能を使用すると、ネットワークの往復回数とデータベースの準備処理の両方が軽減されます。使用制限などの Prepared Statement キャッシュの詳細については、「[Prepared Statement キャッシュのパフォーマンスの向上](#)」を参照してください。
- MSSQLServer のオプティミスティックな同時実行性によって、従来は Oracle でしか使用できなかった高度な機能を Microsoft SQLServer で使用できるようになります。この機能は一部のベンチマークに必要なもので、パフォーマンスが向上する可能性があります。
- EJB 属性のデータベースへの不要な書き込みを排除するために、値変更チェックを追加。長い値を持つカラム (大きな chars/varchar、および Blob/Clob) に関するパフォーマンスで最も効果が現れます。

# セキュリティの新規および更新マニュアル

このリリースの WebLogic Server では、新しいマニュアルの『[WebLogic リソースのセキュリティ](#)』が用意されています。また、『[WebLogic Security の紹介](#)』、『[WebLogic Security の管理](#)』、および『[WebLogic Security プログラマーズ ガイド](#)』も大幅に更新されています。

## Web サービス

WebLogic Server 7.0 SP2 では、以下の新しい Web サービス機能をサポートしています。

- メモリ使用量の最適化

Web Services Descriptor Language (WSDL) をキャッシュすることで、Web サービスを呼び出すときのメモリ使用量を最適化します。この機能を利用するには、サーバの起動時に新しい `weblogic.webservice.cachewsdl` フラグを使用します。

- ポータブル スタブ

Web サービスのクライアント JAR ファイル (製品の一環として配布されたファイルおよび `clientgen` Ant タスクで生成された Web サービス固有のファイル) を WebLogic Server で動作するアプリケーションの一部として使用する場合、JAR ファイルの Java クラスと WebLogic Server の Java クラスが衝突することがあります。クライアント JAR ファイルがデプロイされている WebLogic Server インスタンスと、そのクライアント JAR ファイルが生成された WebLogic Server のバージョンが異なる場合、この問題はより頻発します。この問題を解決するために、ポータブル スタブを使用します。

**注意:** ポータブル スタブを使用する必要があるのは、クライアントアプリケーションが WebLogic Server にデプロイされ、かつ動作している場合のみです。クライアントアプリケーションがスタンドアロンの場合、ポータブル スタブを使用する必要はありません。

クライアントアプリケーションでポータブル スタブを使用可能にするには、『[WebLogic Web サービス プログラマーズ ガイド](#)』の「[ポータブル スタブの作成と使い方](#)」の指示に従ってください。

- `autotype` および `servicegen` Ant タスクで処理される追加の非組み込みデータ型

`autotype` および `servicegen` Ant タスクでは、以下のように追加の非組み込みデータ型を処理するようになりました。

- 単純なコンテンツを含む `<xsd:complexType>`

- `java.util.Stack`
- JAX-RPC スタイルの列挙値クラス

詳細については、「[servicegen および autotype Ant タスクでサポートされる非組み込みデータ型](#)」を参照してください。

## JMS トピック サブスクライバのメッセージ セレクタの最適化

アプリケーションの一部のクラスでは、WebLogic JMS でトピック サブスクライバのメッセージ セレクタにインデックスを付けることで最適化できます。これらのアプリケーションは、1 つまたは複数のユニークな ID と、それらの ID に基づいてフィルタ処理される何千ものサブスクライバを含むメッセージを持ちます。

一般的な例としては、各サブスクライバが異なるユーザに対応し、各メッセージに 1 つまたは複数の対象ユーザのリストが含まれるインスタント メッセージングアプリケーションがあります。『WebLogic JMS プログラマーズ ガイド』の「[トピック サブスクライバのメッセージ セレクタにインデックスを付けることによるパフォーマンスの最適化](#)」を参照してください。

## サーバ レベルの追加 EJB コンパイラ オプション

追加の EJB コンパイラ オプションは、Administration Console で、EJB モジュール レベル ([EJB | コンフィグレーション | 詳細設定]) に加えてサーバ レベル ([サーバ | コンフィグレーション]) でも指定できるようになりました。

**注意:** EJB レベルの設定が優先されます。EJBComponentMbean の EJB レベル オプションを null に設定している場合、WebLogic Server は ServerMBean のサーバ レベルで指定された値を適用します。

## WebLogic Server を Windows サービスとしてインストールする場合の新機能

WebLogic Server を Windows サービスとして実行するように設定する WebLogic Server ユーティリティが変更されました。

- WebLogic Server Windows サービスを Windows のコントロール パネルから正常に停止できる新しいオプションが用意されています。新しいオプションの `-stopclass` は、『[管理者ガイド](#)』で説明されています。
- `-delay` オプションの動作が変更されました。以前のリリースのこのオプションでは、Windows サービス コントロール マネージャ (SCM) が、別のサービスに依存している Windows サービスを呼び出すまでのミリ秒数を指定しました。このリリースの `-delay` オプションでは、関連する Windows サービスのステータスを `STARTING` から `STARTED` に変更する前に、Windows SCM が待機するミリ秒数を指定します。詳細については、『[管理者ガイド](#)』を参照してください。

## Oracle Thin Driver の新しいデフォルトバージョン

デフォルトの Oracle JDBC Thin Driver はバージョン 9.2.0 に変更されました。サービス パック 2 より前のリリースの WebLogic Server 7.0 では、Oracle Thin Driver のデフォルトのバージョンは 8.1.7 でした。

異なるバージョンの Oracle Thin Driver の使用については、『[WebLogic JDBC プログラマーズ ガイド](#)』の「[Oracle Thin Driver の変更または更新](#)」を参照してください。

## WebLogic Server 7.0 SP1 での新機能

以下の節は、BEA WebLogic Server 7.0 SP1 に関する重要な注意です。

## JMS

サービス パック 1 では新しい JMS の機能が提供されます。

### JMS 仕様 1.0.2b の完全な準拠

WebLogic JMS は Sun Microsystems の JMS 仕様 バージョン 1.0.2b に完全に準拠しています。したがって、メッセージ プロパティ名は、[javax.jms.Message](#) Javadoc で定義されている Java の識別子のメッセージ セレクタ構文仕様に厳密に準拠しなければならなくなりました。

### JMS ファイル ストアの同期書き込みポリシー

WebLogic Server 7.0 サービス パック 1 の WebLogic JMS ファイルストアでは、デフォルトの同期書き込みを使用して、最新メッセージの整合性を保証しています。通常、同期書き込みを無効にすると、ファイルストアのパフォーマンスは大幅に向上します。その代わりに、オペレーティングシステムがクラッシュしたり、ハードウェアの障害が発生したりした場合には、メッセージがトランザクション対応であっても、送信したメッセージが失われたり、同じメッセージを重複して受信したりする可能性があります。

『[管理者ガイド](#)』の「[JMS ファイルストアの同期書き込みポリシーのコンフィグレーション](#)」を参照してください。

## サブレット

WebLogic Server 7.0 サービス パック 1 以降、サブレットの初期化は `weblogic.xml` で指定された `<init-as>` ユーザとして実行されます。`weblogic.xml` で指定されていない場合は、`web.xml` で指定された `<run-as>` ユーザとして実行されます。`web.xml` でも指定されていない場合は、`<anonymous>` ユーザとして実行されます。

この変更の結果として、`<init-as>` 要素と `<destroy-as>` 要素が `weblogic.xml` デプロイメント記述子に追加されました。詳細については、『[Web アプリケーションのアセンブルとコンフィグレーション](#)』の「[weblogic.xml デプロイメント記述子の要素](#)」を参照してください。



# Web サービス

サービス パック 1 の WebLogic Web サービスでは、ドキュメント指向の Web サービス処理の実装をサポートしています。WSDL 1.1 仕様で述べられているように、ドキュメント指向の Web サービス処理では、SOAP メッセージにドキュメントが含まれます。これに対し、RPC 指向の Web サービスでは、SOAP メッセージにパラメータと戻り値が含まれます。詳細については、「[RPC 指向またはドキュメント指向の Web サービス](#)」を参照してください。

サービス パック 1 の WebLogic Web サービスには、以下の新しい Ant タスクも用意されています。

- autotype

WebLogic Web サービスへのパラメータまたは戻り値として使用される非組み込みデータ型に対して、シリアライゼーション クラス、Java 表現、XML スキーマ表現、およびデータ型マッピング情報を生成します。

- source2wsdd

Java クラスとして実装された WebLogic Web サービスの Java ソース ファイルから、`web-services.xml` デプロイメント記述子ファイルを生成します。

- wsdl2Service

WSDL ファイルから WebLogic Web サービスのコンポーネントを生成します。コンポーネントには、Web サービスを実装するための起点として使用できる `web-services.xml` デプロイメント記述子ファイルと Java ソース ファイルが含まれています。

- wspackage

WebLogic Web サービスのコンポーネントを、デプロイ可能な EAR ファイルにパッケージ化します。

また、`servicegen` Ant タスクには、JMS を実装した Web サービスのアセンブルに使用できる以下の新しい属性があります。

- JMSDestination
- JMSDestinationType
- JMSAction
- JMSConnectionFactory
- JMSOperationName

- [JMSMessageType](#)

Ant タスクのリファレンス情報については、「[Web サービス Ant タスクとコマンドラインユーティリティ](#)」を参照してください。

**注意：** `wsdl2Service` Ant タスクは準備段階の実装であり、このリリースでは完全にテストされていません。

## WebLogic jDriver for Oracle 9.2.0

WebLogic Server 7.0 サービス パック 1 では、WebLogic jDriver for Oracle バージョン 9.2.0 と Oracle Thin Driver バージョン 9.2.0 が WebLogic Server 7.0 で動作確認されています。これらのドライバを使用して、Oracle 9.2.0 データベースに接続し、JDBC 接続プールでデータベース接続を作成できます。

詳細については、以下の関連ドキュメントを参照してください。

- 「[WebLogic jDriver for Oracle のコンフィグレーション](#)」
- 「[WebLogic Server でのサードパーティ ドライバの使い方](#)」
- 「[動作確認状況](#)」

## EJB の新しいセキュリティ関連要素

7.0 SP1 では、新しい `weblogic-ejb-jar.xml` 要素の `global-role` が導入されます。この要素は、特定のセキュリティ ロールがセキュリティ レルムで「グローバル」に定義されることを示します。詳細については、『[WebLogic エンタープライズ JavaBeans プログラマーズガイド](#)』を参照してください。

# WebLogic Server 7.0 Platform Edition と WebLogic Server 7.0 のスタンドアロン版

WebLogic Server 7.0 Platform Edition (2002 年 6 月 28 日リリース) には、WebLogic Server 7.0 SP1 に含まれる 2 つの新機能が組み込まれています。それ以前の日 WebLogic Server をダウンロードした場合、それは WebLogic Server 7.0 のスタンドアロン版です。

WebLogic Server Platform Edition と WebLogic Server 7.0 スタンドアロン版は以下のように異なります。

- WebLogic Server 7.0 Platform Edition には、強力な統合開発環境である WebLogic Workshop が追加されました。
- また、Platform Edition の WebLogic Server では、Web サービスを有効にするために個別の JAR ファイルをダウンロードする必要がなくなりました。

WebLogic Server 7.0 SP1 には上記の機能が組み込まれています。

## WebLogic Server 7.0 から 7.0 サービス パック 1 への更新

スタンドアロンの WebLogic Server をインストール済みで、2002 年 6 月 28 日にリリースされた WebLogic Platform Edition にアップグレードしていない場合、Smart Update を使用して WebLogic Server 7.0 サービス パック 1 をインストールすることはできません。代わりに、BEA カスタム サポートの Web サイト (<http://support.bea.com>) からダウンロードできるパッケージ アップグレード インストーラを使用できます。詳細については、『[インストールガイド](#)』の「[ダウンロード可能なインストーラによるサービス パックとローリングパッチのインストール](#)」を参照してください。

どのバージョンの WebLogic Server をインストールしているか確認するには、`BEA_HOME\logs\log.txt` (UNIX の場合は `BEA_HOME/logs/log.txt`) をテキスト エディタで開き、ログ ファイルの最後のエントリを参照します。ログ ファイルの各行は、現在の BEA ホームのインストール イベント (インストールまたはアンインストール) を表しています。ログ エントリで、インストール済みのバージョンが WebLogic Platform Edition 7.0 SP0 以前であると示されている場合、Smart Update は使用できません。ログ エントリで、

インストール済みのバージョンが WebLogic Platform Edition (7.0.0.1) 以降であると示されている場合、Smart Update を使用して WebLogic Server サービス パック 1 をインストールできます。

**注意：** WebLogic Server と WebLogic Platform Edition は、同じインストールフレームワークを使用します。したがって、インストールのログ エントリでは、「WebLogic Platform」という語を使用して、WebLogic Server または WebLogic Platform Edition のインストールを表します。

## コンフィグレーション ウィザードを使用して作成されたドメインの移行

コンフィグレーション ウィザード (WebLogic Platform 7.0 より導入) を使用すると、新しいドメインを短時間で簡単に作成できます。WebLogic Platform 7.0 でコンフィグレーション ウィザードを使用してドメインを作成した場合、そのドメインを WebLogic Platform 7.0 サービス パック 1 で使用するために移行する必要があります。

ほとんどのドメインでは、移行は 3 段階の手順で行われます。

1. ドメイン ディレクトリ内の製品 JAR ファイルをアップグレードします。これを行うための移行スクリプトが用意されています。[1-27 ページの「手順 1: 製品 JAR ファイルをアップグレードする」](#)の指示を参照してください。

**注意：** ドメインを移行前の状態に戻すこともできます。

2. サービス パック 1 の変更点をサポートするために、ドメインを更新します。ドメインの生成に使用したドメイン テンプレートによっては、既存のスクリプトまたはファイルを追加または変更する必要があります。[1-28 ページの「手順 2: サービス パック 1 の変更点をサポートするために、ドメインを更新する」](#)の指示を参照してください。
3. WebLogic Platform 7.0 サービス パック 1 を WebLogic Platform 7.0 インストールとは別の新しいディレクトリにインストールした場合は、新しい `BEA_HOME` ディレクトリの場所を参照するように、ドメイン起動スクリプトとコンフィグレーション ファイルを更新します。[1-31 ページの「手順 3: 新しい BEA\\_HOME ディレクトリの場所を参照するように、起動スクリプトとコンフィグレーション ファイルを更新する \(アップグレード以外の場合のみ\)」](#)の指示を参照してください。

**注意：** 既存の WebLogic Platform 7.0 インストールをアップグレードした場合は、この手順を省略できます。

移行するドメインごとに、この手順を繰り返す必要があります。

**注意：** この節では WebLogic Server に固有のドメインの移行方法を説明します。他の WebLogic Platform ドメインの移行方法については、以下の URL にある『BEA WebLogic Platform リリースノート』の「コンフィグレーション ウィザードを使用して作成されたドメインを移行する」を参照してください。

<http://edocs.beasys.co.jp/e-docs/platform/docs70/relnotes/relnotes.html#migration>

## 手順 1：製品 JAR ファイルをアップグレードする

コンフィグレーション ウィザードを使用して生成したドメインの製品 JAR ファイルを サービス パック 1 にアップグレードするには、`BEA_HOME\weblogic700\server\bin` ディレクトリに移動して、以下のいずれかのコマンドを実行します。

Windows: `migrate.cmd domain mode`

UNIX: `migrate.sh domain mode`

**注意：** 処理を開始するためにいずれかのキーを押すようにメッセージが表示されます。

次の表に、コマンドライン引数を定義します。

**表 1-1 7.0 製品の JAR ファイルを 7.0 SP1 にアップグレードするためのコマンドライン引数**

引数	説明
<code>domain</code>	ドメイン ディレクトリの絶対パス名。
<code>mode</code>	移行モード。モードには以下のいずれかの値を設定できる。 <ul style="list-style-type: none"> <li><code>upgrade</code>：必要に応じて、ドメイン ディレクトリ内の製品 JAR ファイルをアップグレードする。元の製品 JAR ファイルは <code>*.jar.orig</code> として保存される。既存の製品 JAR ファイルのタイムスタンプが、SP1 インストールの対応する製品 JAR ファイルのタイムスタンプよりも新しい場合、そのファイルはスキップされる。これはデフォルト モード。</li> <li><code>revert</code>：生成されたバックアップ ファイル (<code>*.jar.orig</code>) を使用して、移行済みのドメインを元に戻す。<code>*.jar.orig</code> ファイルがない場合、このコマンドは無視される。</li> </ul>

たとえば、以下のコマンドでは、デフォルトのユーザ プロジェクト ディレクトリ (*BEA\_HOME*\user\_projects) にある mydomain というドメインをアップグレードします。

```
Windows : migrate.cmd c:\bea\user_projects\mydomain upgrade
UNIX : migrate.sh c:/bea/user_projects/mydomain upgrade
```

以下のコマンドでは、移行中に mydomain に加えられた変更を元に戻します。

```
Windows : migrate.cmd c:\bea\user_projects\mydomain revert
UNIX : migrate.sh c:/bea/user_projects/mydomain revert
```

## 手順 2 : サービス パック 1 の変更点をサポートするために、ドメインを更新する

ドメインの生成に使用したドメイン テンプレートによっては、WebLogic Platform 7.0 サービス パック 1 の変更点をサポートするために、既存のスクリプトやファイルを追加または変更する必要があります。追加の移行手順については、ドメインの生成に使用したドメイン テンプレートに基づいて、以下の適切な節を参照してください。

- [WebLogic Workshop ドメイン](#)
- [WLS ドメイン](#)
- [WLS Examples](#)
- [WLS Petstore](#)

**注意：** 以下の節の説明に従ってファイルを追加または変更する前に、元のファイルをバックアップしておくことをお勧めします。

## WebLogic Workshop ドメイン

WebLogic Workshop テンプレートに基づいたドメインの場合は、以下の手順を実行します。

1. CLASSPATH で定義された JAR ファイル名に適切な PointBase のバージョン (172 から 183 へ) を反映するように、startWebLogic.cmd (Windows) または startWebLogic.sh (UNIX) コマンドを変更します。どちらのコマンドのファイルも、デフォルトでは以下のディレクトリにあります。

```
BEA_HOME\user_projects\domain
```

startWebLogic.cmd スクリプト (Windows) から抜粋した以下のサンプルでは、更新が必要な箇所を**太字**で表示しています。

## 変更前

```
set PB_CLASSPATH=
%POINTBASEDIR%\eval\pointbase\lib\pbserver42ECF172.jar;
%POINTBASEDIR%\eval\pointbase\lib\pbclient42ECF172.jar
```

## 変更後

```
set PB_CLASSPATH=.;
%POINTBASEDIR%\eval\pointbase\lib\pbserver42ECF183.jar;
%POINTBASEDIR%\eval\pointbase\lib\pbclient42ECF183.jar
```

2. 以下のファイルを *BEA\_HOME*\weblogic700\samples\workshop ディレクトリから WebLogic Workshop ドメインの *BEA\_HOME*\user\_projects\domain ディレクトリにコピーします。これらのファイル名のいずれかを使用して作成されたファイルを上書きしないように注意してください。

```
setWorkshopEnv.cmd
setWorkshopEnv.sh
startPointBaseConsole.cmd
startPointBaseConsole.sh
URLs.dat
```

## WLS ドメイン

WLS ドメイン テンプレートに基づいたドメインの場合は、既存のスクリプトやファイルを追加または変更する必要はありません。

## WLS Examples

WLS Examples ドメイン テンプレートに基づいたドメインの場合は、以下の手順を実行します。

1. `startExamplesServer.cmd` (Windows) または `startExamplesServer.sh` (UNIX) コマンドに含まれる `CLASSPATH` 定義を、以下のように変更します。
  - JAR ファイル名に適切な PointBase のバージョン (172 から 183 へ) を反映します。
  - *BEA\_HOME*\server\lib\webservices.jar ファイルを追加します。

どちらのコマンドのファイルも、デフォルトでは以下のディレクトリにあります。

```
BEA_HOME\user_projects\domain
```

`startExamplesServer.cmd` スクリプト (Windows) から抜粋した以下のサンプルでは、更新が必要な箇所を太字で表示しています。

## 変更前

```
set CLASSPATH=
c:\bea\jdk131_03\lib\tools.jar;%POINTBASE_HOME%\lib\
pbserver42ECF172.jar;%POINTBASE_HOME%\lib\
pbclient42ECF172.jar;%CLIENT_CLASSES%;%SERVER_CLASSES%;
%COMMON_CLASSES%;%CLIENT_CLASSES%\utils_common.jar
```

## 変更後

```
set CLASSPATH=
c:\bea\jdk131_03\lib\tools.jar;%POINTBASE_HOME%\lib\
pbserver42ECF183.jar;%POINTBASE_HOME%\lib\
pbclient42ECF183.jar;%CLIENT_CLASSES%;%SERVER_CLASSES%;
%COMMON_CLASSES%;%CLIENT_CLASSES%\utils_common.jar;
c:\bea\weblogic700\server\lib\webservices.jar
```

### 2. Webservices\_trader.ear ファイルを、

*BEA\_HOME*\samples\server\config\examples\applications ディレクトリから Web アプリケーションの *BEA\_HOME*\user\_projects\WLSEExampleDomain\applications ディレクトリにコピーします。カスタマイズ済みのファイルを上書きしないように注意してください。

## WLS Petstore

WLS Petstore ドメイン テンプレートに基づいたドメインの場合は、CLASSPATH で定義された JAR ファイル名に適切な PointBase のバージョン (172 から 183 へ) を反映するように、startPetStore.cmd (Windows) または startPetStore.sh (UNIX) コマンドを変更します。

どちらのコマンドのファイルも、デフォルトでは以下のディレクトリにあります。

```
BEA_HOME\user_projects\domain
```

startPetStore.cmd スクリプト (Windows) から抜粋した以下のサンプルでは、更新が必要な箇所を太字で表示しています。

## 変更前

```
set
CLASSPATH=%JAVA_HOME%\lib\tools.jar;%POINTBASE_HOME%\lib\pbserver42ECF1
72.jar;%POINTBASE_HOME%\lib\pbclient42ECF172.jar;%SERVER_CLASSES%;%COMM
ON_CLASSES%
```

## 変更後

```
set
CLASSPATH=%JAVA_HOME%\lib\tools.jar;%POINTBASE_HOME%\lib\pbserver42ECF1
```



```
83.jar;%POINTBASE_HOME%\lib\pbclient42ECF183.jar;%SERVER_CLASSES%;%COMMON_CLASSES%
```

### 手順 3 : 新しい BEA\_HOME ディレクトリの場所を参照するように、起動スクリプトとコンフィグレーション ファイルを更新する (アップグレード以外の場合のみ)

**注意:** この手順は、WebLogic Platform 7.0 サービス パック 1 を WebLogic Platform 7.0 インストールとは別の新しいディレクトリにインストールした場合にのみ必要です。既存の WebLogic Platform 7.0 インストールをアップグレードした場合は、この手順を省略できます。

ドメイン起動スクリプト (startWebLogic など) とコンフィグレーション ファイル (config.xml など) では、BEA\_HOME ディレクトリ内のファイルへの絶対パス名が定義されています。新しい BEA\_HOME ディレクトリの場所を参照するように、これらの絶対パス名を検索して更新する必要があります。さらに、BEA\_HOME ディレクトリ内のファイルへの絶対パス名が定義されているカスタム スクリプト (構築スクリプトなど) も、新しい BEA\_HOME ディレクトリの場所を反映するように更新する必要があります。

多くの起動スクリプトでは、BEA\_HOME ディレクトリを参照する変数などの環境変数を現在のシェルで設定しています。スクリプト ファイルで BEA\_HOME の参照を更新したら、新しいシェルを開いて、最新の環境設定が使用されていることを確認してください。

## WebLogic Server 7.0 での新機能

BEA WebLogic Server は 7.0 リリースで大幅に変更されました。この節では、WebLogic Server 7.0 と以前のバージョンとの主な相違点について説明します。後の節では、サービス パック リリースで追加された変更点について説明します。

## Web サービス

新しい機能により、柔軟性が向上し、他の主要な Web サービス ベンダーとの相互運用性が保証されるようになり、J2EE プログラマは J2EE コンポーネントを Web サービスとして簡単に公開できるようになりました。プログラミング モデルは、一層の拡張性を提供しながら、SOAP や WSDL の複雑な面をユーザから隠します。以下のような新機能があります。

- Java API for XML Based RPC (JAX-RPC) バージョン 1.0 の実装
- Web サービスのプログラミング作業の大半を自動化する、強化された Ant タスク
- ユーザ定義データ型のサポート
- SOAP メッセージのリクエストと応答をインターセプトするためのメッセージ ハンドラ
- ポイント ツー ポイントのセキュリティ
- J2ME クライアントのサポート
- UDDI レジストリとクライアント API

WebLogic Web サービスの詳細については、『[WebLogic Web サービス プログラマーズ ガイド](#)』を参照してください。

## セキュリティ インフラストラクチャ

WebLogic のセキュリティ サービスは完全に設計し直されており、認証、認可、監査、ロール マッピング、資格マッピング、およびキーストア (PKI) 管理のための完全に実装された SSPI (Security Service Provider Interfaces) 群を公開するモジュール設計を提供します。サードパーティ セキュリティ ベンダ製のモジュールを WebLogic Server フレームワークにそのまま組み込むことができ、サードパーティ製管理ツールを Administration Console に統合できます。新しいロールベースの認可モジュールをすべての J2EE リソースおよび非 J2EE リソースに適用でき、埋め込まれているセキュリティ ポリシー エンジンにより、動的割り当てロールおよびアクセス特権に対する自由記述形式のルールを簡単に作成できます。[WebLogic セキュリティ](#)のドキュメントを参照してください。

## 開発者用ツール

WebLogic Builder は、アプリケーション サーバに短時間でデプロイするための Java ファイルを準備します。WebLogic Builder は、J2EE アプリケーション モジュールをアセンブルし、デプロイメント記述子を作成および編集し、WebLogic Server にそれをデプロイするための、グラフィカル ツールです。ツールを実行するには、サーバを起動し、シェルの環境を設定して、「java weblogic.marathon.Main」と入力します。詳細については、『[WebLogic Builder Online Help](#)』を参照してください。

EJBGen は EJB 生成ツールです。javadoc のコメントを使って、単一の Bean ソース ファイル、デプロイメント記述子ファイル、および EJB のホーム、ローカル、リモートの各インタフェースから、EJB を生成します。詳細については、『[WebLogic エンタープライズ JavaBeans プログラマーズ ガイド](#)』を参照してください。

weblogic.Deployer ユーティリティは、以前の weblogic.deploy ユーティリティに代わるものです。新しいアプリケーションのデプロイ、アプリケーションの再デプロイ、アプリケーションの一部の再デプロイ、アプリケーションの非アクティブ化、アプリケーションのアンデプロイ、デプロイメント タスクのキャンセル、全デプロイメント タスクのリストなどはすべて、weblogic.Deployer の簡単なコマンドライン インタフェースを使って行うことができます。

バージョン 7.0.0.1 以降の WebLogic Server では、BEA WebLogic Workshop™ も提供されます。WebLogic Workshop は、J2EE のエキスパートだけでなく、すべてのアプリケーション開発者が BEA WebLogic Enterprise Platform™ でエンタープライズクラスの Web サービス アプリケーションを迅速に作成、テスト、およびデプロイできるようにする統合開発フレームワークです。BEA WebLogic Workshop は、J2EE の複雑さを隠蔽し、コンポーネント、データ、およびアプリケーション ビジネス ロジックを容易に作成および接続できるようにする統合的な開発プラットフォームを提供します。これまでで初めて、アプリケーションおよびエンタープライズの開発者が同じプラットフォームで同じ言語を使って共同作業することが可能となり、IT 組織全体の生産性が大幅に向上します。

これらを初めとする各種ツールの詳細については、『[WebLogic Server ツール](#)』を参照してください。

## システム管理の強化

WebLogic Server 7.0 で追加された管理上の改良点および新機能は以下のとおりです。

- サーバのライフサイクル管理機能の向上。サーバが起動してから停止するまでの間にたどるいくつかの動作状態を、リモート環境からモニタおよび制御できます。これには、必要な場合に直ちにオンラインに移行できるようサーバの状態を維持することのできる、ホットスタンバイモードも含まれます。
- 管理対象サーバの独立性。WebLogic Server のインスタンスは、管理サーバに接続できない場合でも起動できるようになりました。
- アプリケーションのデプロイメントの改善。アプリケーションのデプロイメントでは、2 フェーズのデプロイメントモデルが使用されるようになっています。Administration Console に新しいデプロイメント画面が追加されると共に、新しいコマンドラインツールが提供されています。
- 移行可能なサービス。クラスタ内の単一のサーバインスタンスにデプロイされている JMS や JTA などの特定のサービスは、サーバで障害が発生した場合には、クライアントのリクエストを中断しない、他のサーバに移行できます。
- ネットワーク チャネル。ネットワーク チャネルを使うと、1 つ以上の WebLogic Server インスタンスまたはクラスタに、追加ポートをコンフィグレーションできます。複数のネットワーク インタフェース カードの使用も、サポートされています。
- ドメインの状態のモニタ。ドメイン内のサーバおよびアプリケーションの状態をモニタできます。クラスタは、サーバで障害が発生した場合は、クライアントのリクエストを他のクラスタメンバにまわしたり、サーバを再起動したりして、自動的に対処できます。  
また、サーバは、それ自体の状態もモニタできます。アプリケーションは、ノードマネージャを利用してサーバをモニタし、サーバで障害が発生した場合は適切な対応を行うことができます。
- Administration Console のヘルプ機能の改善。Administration Console では、すべての属性や画面についての状況依存ヘルプをオンラインで利用できます。

## キャッシング

WebLogic Server 6.0 では、JSP の Cache タグが導入されて、バックエンドと不必要な通信を行わなくても、メモリ中にある JSP の特定の部分をキャッシングできるようになりました。WebLogic Server 7.0 の CacheFilter を利用すると、ページ全体、URL、およびファイルタイプに対するキャッシングを簡単にコンフィグレーションできます。アプリケーションのコードを変更しなくても、キャッシングを有効にして、パフォーマンスやスケーラビリティの改善を直ちに確認できます。

## クラスタ化

マシンをマルチホームにしなくても、クラスタを設定できるようになりました。以前のリリースの場合、クラスタではサーバごとにユニークなリスン アドレスが必要でしたが、各サーバは同じポート上でリスンしていました。1 台のコンピュータにクラスタを作成する場合は、そのコンピュータに複数の IP アドレスを設定して、マルチホーム環境を設定する必要がありました。現在は、クラスタ内の異なるサーバに異なるリスン ポート番号を割り当てることで、1 つのリスン アドレスを持つ 1 つのマシン上にクラスタ環境を設定できます。詳細については、『[WebLogic Server クラスタ ユーザーズ ガイド](#)』を参照してください。

## EJB 2.0

新しいエンタープライズ JavaBean (EJB) 2.0 機能には、アプリケーション コード内でクエリを実行するための動的クエリ サポートや、WebLogic Server のコンテナ管理による永続性を利用した複数の DBMS テーブル マッピング、さらに、サブクエリ、集約関数、結果セットを返すクエリ、大文字 / 小文字を区別しない検索、および NO WAIT 文を含む SELECT FOR UPDATE に対する新しい EJB WebLogic QL サポート、改善された同時実行性とキャッシング、EJB リンクのサポート、そしてバルク挿入更新のサポートなどが含まれます。EJB の新機能の詳細については、『[WebLogic エンタープライズ JavaBeans プログラマーズ ガイド](#)』を参照してください。

## メッセージ駆動型 Bean と非 BEA JMS プロバイダ

WebLogic Server 7.0 以降、トランザクション MDB のあるアプリケーションは、MDB によって処理されるメッセージに関して非 BEA JMS プロバイダとの間で「かならず 1 回」のセマンティクスを実現できます。また、WebLogic Server は XA を使用して非 BEA JMS プロバイダをトランザクションの中に自動的に取得します。詳細については、「[非 BEA JMS プロバイダのメッセージ駆動型 Bean のコンフィグレーション](#)」を参照してください。

## ディレクトリ構造

WebLogic Server 7.0 で構築するアプリケーションのディレクトリ構造と格納場所が変更されています。ディレクトリ構造が新しくなったことで、柔軟性が高まり、アプリケーション開発のベスト プラクティスが促進されます。詳細については、『インストール ガイド』の「WebLogic Server のインストール準備」の章の「[BEA ホーム ディレクトリ](#)」の節を参照してください。

## コンフィグレーション ウィザード

コンフィグレーション ウィザードを使うと、コンフィグレーション済みのドメインを好きな場所に生成できるので、ドメインとクラスタの管理がとて簡単になりました。ドメインとは、相互関係のある WebLogic Server リソースの集合であり、同じコンフィグレーション ファイルを共有し、WebLogic Server インスタンス、WebLogic Server クラスタ、およびアプリケーションを含む 1 つの単位として管理されます。コンフィグレーション ウィザードは、ユーザの問い合わせに基づき、適切なコンフィグレーション済みのコンポーネントと関係する資産を利用して、ドメイン、サーバ、およびエンタープライズアプリケーションを生成します。カスタム インストールの途中およびそれ以降のいつでも、ドメインコンフィグレーション ウィザードを実行できます。コンフィグレーション ウィザードは、特に、クラスタを容易にセットアップできるよう設計されています。コンフィグレーション ウィザード起動用のスクリプトは、`WL_HOME\common\bin` ディレクトリにあります。

## インストール

WebLogic Server のこのリリースでは、インストールに関して以下の点が新しく拡張されています。詳細については、『[インストール ガイド](#)』を参照してください。

## インストール オプション

WebLogic Server 7.0 のインストール時には、標準インストール オプションまたはカスタムインストール オプションを選択できます。標準インストールには、WebLogic Server を簡単にインストールするための最も一般的なオプションが含まれています。カスタムインストールでは、インストール オプションを選択したり、インストール中にドメインをコンフィグレーションしたりできます。

## Smart Update

WebLogic Server 7.0 に付属する新しいインストーラ プログラムを使うと、サービス パックや製品の更新情報を探してインストールすることもできます。

## J2EE コネクタ

J2EE コネクタでは、WebLogic Server のアプリケーションと既存のアプリケーションを統合する標準的な方法が定義されています。WebLogic Server 7.0 は、コネクタ 1.0 仕様の最終確定版を実装しています。このリリースには、JCA アダプタの接続およびセキュリティ管理に対する拡張機能も含まれています。

WebLogic Server のこのリリースでは、以下の J2EE コネクタ アーキテクチャ拡張機能が追加されました。詳細については、『[WebLogic J2EE コネクタ アーキテクチャ](#)』を参照してください。

## JNDI の実装

このリリースでは、参照可能な JNDI の実装がサポートされています。この機能に関連のある以前の問題が解決されていることを確認するための統合テストが行われています。

## セキュア パスワード資格ストレージ

このリリースでは、リソース アダプタ デプロイヤが、必要に応じて、指定した認証 / 認可メカニズムをセキュア パスワード資格ストレージを通して組み込むための、標準的な方法が提供されています。リソース アダプタ アーカイブ内の `weblogic-ra.xml` デプロイメント記述子で提供されていたセキュリティ プリンシパル マッピング メカニズムは、この WebLogic Server ストレージ メカニズムに変わっています。

この新しいストレージ メカニズムは、開始プリンシパル (WebLogic Server ユーザ名とパスワードの組み合わせなど) をリソース プリンシパル (EIS ユーザ名とパスワードの組み合わせ) にマッピングするために使用します。

## 接続リークの検出

旧リリースにおける接続リーク検出メカニズムは、接続が作成されると開始し、接続の使用時間が経過するとトリガするタイマを利用していました。このリリースでは、接続リークを検出するための新しいメカニズムが採用されています。このメカニズムは、ガベージコレクタを利用し、接続オブジェクトの使用を追跡するアイドル タイマが提供されています。

## プールの動的な変更

このリリースでは、実行時にコンフィグレーション可能な各種接続プール パラメータを、実行時に動的に変更できるようになっています。Administration Console のデプロイメント記述子エディタを使ってこれらのパラメータを変更すると、変更は直ちに有効になります。WebLogic Server を再起動したり、リソース アダプタを再デプロイしたりする必要はありません。

## ra.xml 仕様のセキュリティ ポリシー処理

BEA WebLogic Server の J2EE コネクタ アーキテクチャでは、管理対象の実行時環境におけるリソース アダプタの実行に対して、一群のセキュリティ パーミッションが提供されています。また、WebLogic Server は、システム リソースにアクセスするためのリソース アダプタの明示的パーミッションも許可します。

## jCOM

WebLogic jCOM は、WebLogic Server にデプロイされた Java および J2EE オブジェクトと、Microsoft Office ファミリ製品、VisualBasic や C++ のオブジェクト、およびその他の COM/DCOM 対応環境で利用可能な Microsoft ActiveX コンポーネントとの間の双方向アクセスを可能にする、ソフトウェアブリッジです。

WebLogic jCOM の機能により、Microsoft COM クライアントは WebLogic Server のオブジェクトに COM コンポーネントと同じようにアクセスでき、また WebLogic Server のアプリケーションは COM コンポーネントに Java オブジェクトと同じようにアクセスできます。どちらの場合も、オブジェクトの種類の間にある違いは、WebLogic jCOM によって透過的になります。COM クライアントからは WebLogic Server オブジェクトは COM オブジェクトのように見え、WebLogic Server アプリケーションからは COM コンポーネントは Java オブジェクトのように見えます。



## JDBC

WebLogic JDBC に関する新機能および改善された機能は、以下のとおりです。

### ARRAY、STRUCT、REF のサポート

WebLogic Server 7.0 は、ARRAY、STRUCT、および REF をサポートしています。これには、JDBC の標準メソッドと一部の Oracle 拡張機能が含まれています。ARRAY、STRUCT、または REF を使用するには、Oracle Thin Driver を使う必要があります。詳細については、『WebLogic JDBC プログラマーズ ガイド』の「[Oracle 拡張機能と Oracle Thin Driver の使用](#)」を参照してください。

### JDBC 接続プール管理 MBean

WebLogic Server 管理アーキテクチャ (JMX) の一部として、JDBCConnectionPool 管理 MBean が提供されます。JDBCConnectionPool MBean を使えば、Java アプリケーションの中から接続プールを動的に作成およびコンフィグレーションすることができます。詳細については、『WebLogic JDBC プログラマーズ ガイド』の「[接続プールの動的作成](#)」を参照してください。

### WebLogic Server に付属するサードパーティ製 JDBC ドライバ

WebLogic Server には、Oracle Thin Driver および Sybase jConnect JDBC ドライバの複数のバージョンが付属しており、異なるフォルダに格納されています。使用する JDBC ドライバのバージョンを選択できます。また、DBMS ベンダから提供されるバージョンに簡単に更新できます。詳細については、『WebLogic JDBC プログラマーズ ガイド』の「[サードパーティ JDBC ドライバの概要](#)」を参照してください。

### Oracle 9i のサポート

WebLogic Server は Oracle 9i をサポートしています。Oracle 9i のデータベースに接続するには、WebLogic jDriver for Oracle や Oracle Thin Driver Version 9.0.1 以降のような、Oracle 9i をサポートする JDBC ドライバを使う必要があります。詳細については、『WebLogic jDriver for Oracle のコンフィグレーションと使い方』および『[WebLogic JDBC プログラマーズ ガイド](#)』を参照してください。

## JMS

WebLogic JMS の新機能は以下のとおりです。

### 向上した可用性

WebLogic JMS は、クラスタ環境用の WebLogic Server コアに実装されている移行フレームワークを利用します。これにより、WebLogic JMS は、移行リクエストに適切に応答し、JMS サーバを正しい順序でオンラインまたはオフラインにすることができます。これには、スケジューリングされた移行だけでなく、WebLogic Server の障害への対応としての移行も含まれます。

詳細については、『WebLogic JMS プログラマーズ ガイド』の「[WebLogic JMS の管理](#)」を参照してください。

### WebLogic クラスタ内に分散された送り先

WebLogic JMS の高可用性の実装を利用すれば、複数の物理的な送り先を単一の分散型送り先セットのメンバとしてコンフィグレーションすることができ、クラスタのサーバの 1 つで障害が発生した場合でも、サービスの連続性を一定の水準に保つことができます。特に、特定の送り先（キューまたはトピック）の複数のインスタンスをクラスタ内にコンフィグレーションできます。クラスタ内のインスタンスの 1 つで障害が発生した場合は、同じ分散型送り先の他のインスタンスが、JMS のプロデューサおよびコンシューマにサービスを提供できます。

詳細については、『WebLogic JMS プログラマーズ ガイド』の「[WebLogic JMS アプリケーションの開発](#)」および『[管理者ガイド](#)』の「[JMS の管理](#)」を参照してください。

### フロー制御

フロー制御機能を使うと、過負荷状態になっていると判断した JMS サーバや送り先は、メッセージ プロデューサの処理速度を遅くすることができます。特に、JMS サーバまたは送り先は、指定されているバイトしきい値またはメッセージしきい値を超えたなら、メッセージ フローを制限するようプロデューサに指示できます。

詳細については、『[管理者ガイド](#)』の「[JMS の管理](#)」を参照してください。

## WebLogic メッセージングブリッジ

メッセージングブリッジは、2つのメッセージングプロバイダ間のメッセージ転送を行います。WebLogic メッセージングブリッジ機能を使うと、WebLogic JMS の個別の実装を含む任意の2つのメッセージング製品間に、保存と転送のメカニズムをコンフィグレーションできます。

詳細については、『管理者ガイド』の「[メッセージングブリッジのコンフィグレーション](#)」を参照してください。

## メッセージ ページング

メッセージ ページング機能は、メッセージ負荷が指定されたしきい値に達したなら、仮想メモリから永続ストレージにメッセージをスワップアウトすることで、メッセージ負荷のピーク時に貴重な仮想メモリを解放します。今日のエンタープライズアプリケーションで必要とされる大きなメッセージ空間を備えた WebLogic Server の実装は、パフォーマンスに関してこの機能から大きな恩恵を受けます。

詳細については、『管理者ガイド』の「[JMS の管理](#)」を参照してください。

## 送り先ルックアップでのサーバ アフィニティ

`createTopic()` メソッドおよび `createQueue()` メソッドを使用するときに、「`JMS_Server_Name./Destination_Name`」構文で送り先ルックアップ時のサーバ アフィニティを示すことができるようになりました。送り先が接続ファクトリと同じ JVM にローカルにデプロイされている場合、接続ファクトリからはローカルの送り先と一致する名前のみが返されます。名前がローカル JVM にない場合は、同じ名前が別の JVM にデプロイされていても例外が送出されます。

アプリケーションでは、この規約を利用して `createTopic` メソッドおよび `createQueue` メソッドの使用時にサーバ名のハードコード化を避けることができます。その結果、他の JMS サーバでコードをそのまま再利用できます。詳細については、『WebLogic JMS プログラマーズ ガイド』の「[WebLogic JMS アプリケーションの開発](#)」を参照してください。

## BEA Jolt Client 8.0.1

BEA Jolt Client 8.0.1 は、Tuxedo サーバ上で動作する Jolt サービス リスナ (JSL) を経由して Tuxedo サービスへの要求を管理する Java ベースのクライアント API です。この Jolt API は WebLogic API 内部に組み込まれ、サーブレットなどの BEA WebLogic アプリケーションからアクセスできます。

Jolt Java Client 8.0.1 は、[BEA Product Download Center](#) から入手できます。BEA WebLogic Server 7.0 へのリンクを辿り、[Modules for WebLogic Server] リストから [Jolt Java Client 8.0.1] を選択してください。Jolt のライセンスが必要です。

## Java Transaction API (JTA)

WebLogic Server のこのリリースでは、Java Transaction API に関して以下のような機能強化が新しく行われています。詳細については、『[WebLogic JTA プログラマーズ ガイド](#)』を参照してください。

### トランザクション回復サービスの移動

WebLogic Server 7.0 では、クラスタ内のサーバから同じクラスタ内の別のサーバに、トランザクション回復サービスを移動できます。これにより、可用性の高いトランザクション回復が実現されます。トランザクション回復サービスは、障害の発生したサーバで完了していないトランザクションを完了させます。詳細については、『[WebLogic Server 管理者ガイド](#)』の「[サーバに障害が発生した後のトランザクションの回復](#)」を参照してください。

## ネットワーク リソース

WebLogic Server 7.0 では、2 種類の新しいコンフィグレーション可能なネットワーク リソースが導入されています。どちらも、Administration Console または WebLogic Server MBean を使って管理できます。それらのリソースは、ネットワーク チャネルとネットワーク アクセス ポイント (NAP) です。

ネットワーク チャネルでは、WebLogic Server に対するネットワーク接続の基本属性が定義されています。Administration Console で個別のエンティティとしてチャネルをコンフィグレーションし、ドメイン内のサーバに 1 つ以上のチャネルを追加します。複数のサーバで 1 つのチャネルを使用すると、チャネルのコンフィグレーションを変更することで、そ

のチャンネルを使用するすべてのサーバの接続属性が自動的に変わるので、WebLogic Server ドメインに対するネットワークのコンフィグレーションが簡単になります。また、1 つのサーバに対して複数のチャンネルを作成して割り当てることができます。複数のチャンネルを使うと、プロトコルごと、リスンポートごと、またはその他のチャンネルコンフィグレーションプロパティごとに、ネットワークトラフィックを分けることができます。たとえば、1 つのサーバで2 つのチャンネルを使えば、セキュアなトラフィック用と非セキュアなトラフィック用に、デフォルトの接続プロパティを調整できます。詳細については、「[ネットワークリソースのコンフィグレーション](#)」を参照してください。

ネットワークアクセスポイント (NAP) は、実際のポート番号と、オプションとしてネットワークチャンネルが使用する特定の IP アドレスを表す、コンフィグレーション可能なリソースです。ネットワークチャンネルだけでもドメイン内の共有接続プロパティの管理に役立ちますが、普通は、NAP とチャンネルを併用して、ネットワークトラフィックを複数の NIC およびポート番号に分割します。また、NAP を使えば、特定のチャンネル属性をオーバーライドして、WebLogic Server のネットワークコンフィグレーションをカスタマイズすることもできます。詳細については、「[ネットワークリソースのコンフィグレーション](#)」を参照してください。

## RMI-IIOP

RMI-IIOP の WebLogic 実装では、パフォーマンス向上のためのさまざまな拡張に加えて以下のような機能が追加されました。

- RMI-IIOP にトランザクションサポートを追加する Object-Transaction-Services 1.2 (OTS 1.2) のサポート。外部のアプリケーションサーバでホストされているリモートオブジェクトを呼び出すために使用できます。
- クライアント認証、委託、および特権機能を提供する CSIv2 (Common Secure Interoperability, version 2)。外部アプリケーションサーバとも動作します。
- 完全にクラスタ化可能な WebLogic IIOP クライアント。次のコマンドラインパラメータを使うだけで、WebLogic IIOP クライアントのサポートを有効にすることができます。`-D weblogic.system.iiop.enableClient=true`
- IIOP スタブのホットコード生成。ejbc または rmic で `-iiop` フラグを使う必要はありません。
- CosNaming API の完全なサポート。

# Web アプリケーションおよび EJB のセキュリティ コンフィグレーションに関する更新された手順

このリリースの WebLogic Server では、Web アプリケーションと EJB を保護する手順が改良されました。詳細な説明については、『[WebLogic リソースの保護](#)』を参照してください。

## WebLogic Tuxedo Connector

WebLogic Tuxedo Connector には、以下の機能が追加されました。

- WebLogic Server サービスと完全に統合されるようになっており、個別の XML コンフィグレーション ファイルは使用しなくなった
- WebLogic Server の `weblogic.wtc.encoding` プロパティを使用したコードセットの変換
- バッファ表示機能のサポート
- クラスタ環境で使用できる

## WebLogic XML

WebLogic Server 7.0 には、以下の新しい XML 機能が追加されています。

- XML ストリーミング API: WebLogic XML ストリーミング API は、XML ドキュメントを解析および生成するための簡単でわかりやすい手段を提供します。SAX API と似ていますが、XML ドキュメントに対する手続き的なストリーム ベースの処理が可能であり、SAX イベント ハンドラを作成する必要はありません。SAX イベント ハンドラで複雑な XML ドキュメントを扱うと、ハンドラ自体が複雑になる場合があります。つまり、ストリーミング API を利用すれば、SAX API よりきめ細かく解析を制御できます。
- XML アプリケーション スコープ: アプリケーション スコープとは、WebLogic Server のコンフィグレーション全体に対してではなく、特定のエンタープライズ アプリケーションに対してリソースをコンフィグレーションすることです。XML の場合、このよ

うなリソースとしては、パーサ、トランスフォーマ、外部エンティティ、および外部エンティティのキャッシュ コンフィグレーションがあります。アプリケーション スコープの最大の利点は、特定のアプリケーションに対するリソースがそのアプリケーション専用になることです。アプリケーション スコープを使うと、アプリケーションごとに異なるパーサをコンフィグレーションしたり、アプリケーション用の DTD を EAR ファイルや展開されたエンタープライズ ディレクトリに格納したりすることができます。

これらの新機能の詳細については、『[WebLogic XML プログラマーズ ガイド](#)』を参照してください。

## ドキュメントとサンプル

WebLogic Server 7.0 に付属するドキュメントとサンプルに関しては、以下のような変更がありました。

### ドキュメント

e-docs は、WebLogic Server のための情報源です。このドキュメントでは、WebLogic Server 7.0 の最新の機能と変更点がすべて網羅されています。[Administration Console オンライン ヘルプ](#)も大幅に変更されており、e-docs で利用できます。また、以下の新しいガイドが追加されています。

- 『[WebLogic Server ロギング サービスの使い方](#)』
- 『[BEA WebLogic Server 7.0 へのアップグレード](#)』
- 『[WebLogic Server ドメイン管理](#)』
- 「[セキュリティ](#)」 ページ
- 「[開発者ツール](#)」 ページ
- 「[デプロイメント](#)」 ページ

### サンプル

WebLogic Server 7.0 に付属するサンプルが大幅に変更されました。変更点は以下のとおりです。

- サンプルの一部は「そのまま」実行できます。サンプル サーバを起動すると、ブラウザが開き、WebLogic Server の機能のいくつかを試してみることができます。

- サンプル サーバおよび Pet Store サーバを実行するための起動スクリプトが使いやすいように改善されました。
- 多くのサンプルには、サンプルを簡単に実行できる `ant` コマンドが用意されています。各サンプルに付属する `build.xml` ファイルを見れば、スクリプト化されているコマンドライン構文を確認できます。
- サンプルが使用するデータベース システムが変更されました。WebLogic 7.0 のサンプルは、100% pure Java の RDBMS である PointBase® Server を使用しています。

以下の機能をデモンストレーションする新しいサンプルが追加されています。

- クラスタ化されたサーバによるセッション レプリケーション
- EJB 2.0 に対する BLOB/CLOB、Bands、Embedded Key、および Multi-table
- RMI-IIOP のセキュリティで保護された Tuxedo クライアント
- Javamail
- jCom
- JDBC 動的データ ソース、JDBC Oracle 拡張機能、JDBC 動的プール
- XML でデータを送信するメッセージ駆動型 Bean を使用する JMS
- WebLogic における最新のセキュリティの実装をデモンストレーションする新しいセキュリティ サンプル
- サブレットのフィルタ、ライフサイクル、およびラッパー
- WebLogic XML ストリーミング API の使用
- EJB 2.0 のステートレス セッション Bean やログ機能を使用する Web サービス

### Pet Store 1.3

WebLogic Server 7.0 に付属する Pet Store サンプル アプリケーションが、バージョン 1.3 にアップグレードされました。Pet Store アプリケーションは [ スタート ] メニューから利用でき、ソース コードは `SAMPLES_HOME\server\src\petstore` ディレクトリにあります。`SAMPLES_HOME` は、WebLogic プラットフォームのすべてのサンプルが収められている場所で、デフォルトでは `c:\bea\weblogic700\samples` です。



## WebLogic Server ツアー

WebLogic Server ツアーでは、Pet Store アプリケーションを使って機能のデモンストレーションを行い、WebLogic Server の概要を紹介します。ツアーは、[スタート]メニューから利用できます。

## プラットフォームのサポート

詳しいプラットフォーム サポートの最新情報については、[プラットフォーム サポートのページ](#)を参照してください。

## 標準とライブラリに関するバージョンの互換性

下の表に、WebLogic Server 7.0 に含まれる、または互換性のある標準とライブラリのバージョンを示します。

表 1-2 WebLogic Server 7.0 と互換性のある標準およびライブラリ

標準またはライブラリ	バージョン
org.w3c.dom インタフェース	
org.xml.sax インタフェース	
antlr	2.7.1
EJB	2.0 および 1.1
HTTP	1.1
JAAS	.0 全体
Java RMI	1.0
JavaMail	1.1

表 1-2 WebLogic Server 7.0 と互換性のある標準およびライブラリ

JAXP	1.1
JAX-RPC	1.0
JCA	1.0
JDBC	2.0.x
JMS	1.0.2
JMX	1.0
JMX	1.0
JNDI	1.2
JSP	1.2 および 1.1
JTA	1.0.1
LDAP	2
Oracle	<ul style="list-style-type: none"> <li>● 8.1.7</li> <li>● 9.0.1</li> <li>● 9.2.0</li> </ul>
OTS/JTA	1.2/1.0.1
RMI/IIOP	1.0
サーブレット	2.3 および 2.2
SSL	3
X.509	3
Xerxes DOM および SAX	<ul style="list-style-type: none"> <li>● DOM パーサ: レベル 2.0 コア</li> <li>● SAX パーサ: バージョン 2.0</li> </ul>

表 1-2 WebLogic Server 7.0 と互換性のある標準およびライブラリ

XML	org.apache.xalan Xerces XML パーサのバージョンと互換性のある任意のバージョン
XML パーサ	org.apache.xerces 組み込みのバージョン : 1.4.4 WebLogic Server 7.0.0.1 以前と互換性があるもの : <ul style="list-style-type: none"> <li>● 1.2.2</li> <li>● 1.2.3</li> <li>● 1.3.0</li> <li>● 1.3.1</li> <li>● 1.4.0</li> <li>● 2.0.0</li> <li>● 2.0.1</li> </ul> WebLogic Server 7.0 SP1 以降と互換性があるもの : <ul style="list-style-type: none"> <li>● Xerces の任意のバージョン</li> </ul>

## Java Development Kit

JDK には Java の実行時環境 (Java 仮想マシン : JVM) と、Java アプリケーションのコンパイルおよびデバッグ用ツールが用意されています。WebLogic Server 7.0 には、Sun Microsystems の JDK 1.3.1\_02 が含まれています。

# WebLogic Javadoc

外部に対してサポートされている API のドキュメントの生成には、Javadoc が使われています。Javadoc は Sun から提供されているツールで、パブリック スコープにおいて宣言されているクラスの API をドキュメント化するために使用します。パブリック スコープで宣言されているクラスは、それ自体のパッケージの外部にあるクラスから見ることはできません。BEA では、Javadoc のカスタム バージョンを使用しています。このバージョンは、外部に対してサポートされているパブリック クラスと、内部だけで使用するパブリック クラスを区別するように拡張されています。ただし、Javadoc は、使用方法のページ、継承ツリー、およびその他の補助リストにおいて、内部のクラスまたはメソッドに対する間接的な参照を生成できます。このような間接的な参照は、内部のメソッドまたはクラスのサポートを意味してはいません。BEA は、属しているクラスに対するメインの Javadoc ページで完全に記述されている API だけをサポートします。つまり、左上のフレームでパッケージを選択すると、そのパッケージに含まれていて外部に対してサポートされるすべてのクラスが、左下のフレームに表示されます。

## 独自の MBean の作成

WebLogic Server では、セキュリティ サービスや JMS などの WebLogic サービスと、EJB や Web アプリケーションなどのデプロイ可能な J2EE モジュールをコンフィグレーションおよびモニタするために、多数の MBean を提供しています。アプリケーションまたはサービスをコンフィグレーションするために追加の MBean を使用する場合は、独自の MBean を作成することができます。

セキュリティ プロバイダ用の MBean を作成する場合は、WebLogic Security サービスで用意されている MBean 生成ツールを使用します。このツールでは、セキュリティ プロバイダ MBean の生成のみをサポートしています。セキュリティ プロバイダの詳細については、『[WebLogic Security サービスの開発](#)』を参照してください。

WebLogic Server のその他の管理対象リソース用に MBean を作成する場合は、JMX 仕様 (<http://java.sun.com/products/JavaManagement> からダウンロード可能) を参照してください。WebLogic Server のインスタンス上で作成および登録する MBean は、JMX 仕様で定義されているすべての JMX 機能を利用できます。WebLogic Server における JMX 実装の詳細については、『[WebLogic JMX Service プログラマーズ ガイド](#)』を参照してください。

# 利用可能な他のリソース

ここでは、WebLogic Server 7.0 に関係する役に立つ情報を紹介します。ハイパーリンクを利用するには、インターネット アクセスが必要です。

## 短時間で実行可能な手順

HTML ファイル、JSP、およびサーブレットを短時間でデプロイできる概要レベルの手順については、

[http://edocs.beasys.co.jp/e-docs/wls/docs70/quickstart/quick\\_start.html](http://edocs.beasys.co.jp/e-docs/wls/docs70/quickstart/quick_start.html) を参照してください。

## Pet Store 1.3

Pet Store は、WebLogic Server の機能を紹介するサンプル アプリケーションです。WebLogic Server 7.0 をインストールしたディレクトリにあり、[スタート]メニューから利用できます。

## サンプル

コード サンプルをインストールした場合は、WebLogic Server のインストール先の `SAMPLES_HOME\server\src\examples` ディレクトリに格納されています。`SAMPLES_HOME` は、WebLogic プラットフォームのすべてのサンプルが収められている場所で、デフォルトでは `c:\bea\weblogic700\samples` です。サンプルは、Windows では、[スタート]メニューから利用することもできます。

## BEA WebLogic Server 7.0 の紹介

WebLogic Server の機能と J2EE アプリケーション アーキテクチャの概要については、『[BEA WebLogic Server 7.0 および WebLogic Express の紹介](#)』を参照してください。

## その他のドキュメント

管理、プログラミング、および参照ガイドを含む BEA WebLogic Server の完全なドキュメントセットは、BEA WebLogic Server オンライン ドキュメントの CD-ROM に収められており、BEA の Web サイト (<http://edocs.beasys.co.jp/e-docs/wls/docs70/index.html>) でも提供されています。

## ニュースグループ

BEA WebLogic Server のニュースグループは、コミュニティによる BEA 製品のサポートを提供します。BEA 関連のニュースグループについては、<http://newsgroups.bea.com/> および <news://newsgroups.bea.com> を参照してください。

## Dev2Dev Online

BEA の Dev2Dev Online サイトでは、短期間で簡単に e コマースを開発できるリソースが提供されています。Dev2Dev Online を利用するには、<http://developer.bea.com/> にアクセスしてください。

# 確認済みの問題

以下の節では、WebLogic Server 7.0 とそのサービス パックにおける確認済みの問題、サポートのレベル、注意事項、および一般的な問題について説明します。

- [2-1 ページの「WebLogic Server 7.0 サービス パック 7 の確認済みの問題」](#)
- [2-3 ページの「WebLogic Server 7.0 サービス パック 6 の確認済みの問題」](#)
- [2-6 ページの「WebLogic Server 7.0 サービス パック 5 の確認済みの問題」](#)
- [2-13 ページの「WebLogic Server 7.0 サービス パック 4 の確認済みの問題」](#)
- [2-15 ページの「WebLogic Server 7.0 サービス パック 3 の確認済みの問題」](#)
- [2-20 ページの「WebLogic Server 7.0 サービス パック 2 の確認済みの問題」](#)
- [2-50 ページの「WebLogic Server 7.0 サービス パック 1 の確認済みの問題」](#)
- [2-86 ページの「WebLogic Server 7.0 の確認済みの問題」](#)

## WebLogic Server 7.0 サービス パック 7 の 確認済みの問題

WebLogic Server 7.0 SP7 では、以下の問題が確認されています。

## コア サーバに関する確認済みの問題

変更要求番号	説明	問題が修正されたリリース
CR174779	<p>プライマリ サーバが停止を始めた後でも、既存のセッションからの要求がプライマリ サーバに転送されます。</p> <p><b>回避策:</b> 少し時間をおいてからページを更新してください。</p>	

## EJB に関する確認済みの問題

変更要求番号	説明	問題が修正されたリリース
CR262695	<p>ejb.xml および weblogic-cmp-rdbms-jar.xml デプロイメント記述子ファイルで完全修飾クラス名が生成されません。代わりに、メソッド パラメータの値として単純なクラス名のみが生成されます。</p> <p><b>回避策:</b> EJB でアンロード可能なクラスを使用している場合は、デプロイメント記述子ファイルを手動で編集し、完全修飾名を使用してその EJB を参照するようにしてください。</p>	



## RMI に関する確認済みの問題

変更要求番号	説明	問題が修正されたリリース
CR258722	<p>IIOP プロトコルを介して WebLogic Server 7.0 SP7 のクライアントとして動作する場合、WebLogic Server 6.1 にはいくつかの制限事項がありました。WebLogic Server 6.1 はクラスの変更をサポートしていないので、WebLogic Server 6.1 と WebLogic Server 7.0 SP7 の間では同一のクラスのみを渡すことができます。</p> <p><b>回避策:</b> この問題に留意してください。</p>	
CR261149	<p>IIOP を介したリモート リクエストに中位が大きなサイズの String が含まれている場合、WebLogic Server 7.0 SP7 では 7.0SP6 に比べて若干速度が低下する可能性があります。これは、WebLogic Server 7.0SP7 で使用されている JDK 1.4.2_10 の変更点によるものです。</p> <p><b>回避策:</b> <code>XX: +UseTLAB JVM オプション フラグ</code> を渡します。このフラグはスレッドローカルのメモリ割り当てを有効にします。SPARC 上のサーバ VM の場合にのみ、このフラグはデフォルトで有効になっています。</p>	

## WebLogic Server 7.0 サービス パック 6 の確認済みの問題

WebLogic Server 7.0 SP6 では、以下の問題が確認されています。

## Administration Console に関する確認済みの問題

変更要求番号	説明	問題が修正されたリリース
CR213792	このリリースの WebLogic Server では、EJB に対してセキュリティ ロールとセキュリティ ポリシーを定義するためのヘルプ情報が Administration Console に表示されません。EJB に対するセキュリティ ロールおよびセキュリティ ポリシーの割り当てについては、「 <a href="#">WebLogic リソースの保護</a> 」を参照してください。	
CR215117	WebLogic Server 7.0 SP6 では、Administration Console の [サーバ   モニタ   バージョン] ページに <code>weblogic.management.runtime.JVMRuntimeMBean</code> の属性 <code>JavaVMVendor</code> が追加されました。 Administration Console オンライン ヘルプには、現在この属性に対応するポップアップがありません。そのため、この属性のオンライン ヘルプにアクセスしようとすると 404 エラーが発生します。	

## JDBC に関する確認済みの問題

変更要求番号	説明	問題が修正されたリリース
CR192616	ネット ドライバを経由する API <code>getXAConnection()</code> は、IBM DB2 UDB type 2 JDBC 2.0 ドライバでサポートされていません。 <code>getXAConnection()</code> API 対応の他のドライバを使用することをお勧めします。	

## プラグインに関する確認済みの問題

変更要求番号	説明	問題が修正されたリリース
CR217777	<p>負荷がかかると Apache HTTP Server プラグインにデッドロックが発生します。</p> <p><b>問題:</b> Apache HTTP Server プラグインを使用してクライアントの接続リクエストをクラスタ内の WebLogic Server インスタンスに分散する場合、クライアントがクラスタに接続できずに Apache HTTP Server プラグインのログに次のメッセージが記録されることがあります。</p> <pre>[error] server reached MaxClients setting, consider raising the MaxClients setting</pre> <p>この問題は、負荷のかかった状況下で発生します。</p> <p>プラットフォーム: すべて</p> <p><b>回避策:</b> Apache HTTP Server プラグインをコンフィグレーションする場合は、httpd.conf コンフィグレーション ファイルの MaxClients パラメータに、クラスタでサポートできる必要のあるクライアント接続の数と同等の値を設定します。</p> <p>Apache HTTP Server プラグインのコンフィグレーションの詳細については、次の URL にある『WebLogic Server における Web サーバ プラグインの使い方』の「Apache HTTP Server プラグインのインストールとコンフィグレーション」を参照してください。</p> <p><a href="http://edocs.beasys.co.jp/e-docs/wls/docs81/plugins/apache.html">http://edocs.beasys.co.jp/e-docs/wls/docs81/plugins/apache.html</a></p>	

# WebLogic Server 7.0 サービス パック 5 の 確認済みの問題

WebLogic Server 7.0 サービス パック 5 では、以下の問題が確認されています。

## コア サーバに関する確認済みの問題

変更要求番号	説明	問題が修正されたリリース
CR123714	<p>それぞれ独自のネットワーク インタフェース カードを備えた複数のサーバが単一のクラスタまたはドメイン内で動作する場合、ネットワーク クラスロードが失敗する可能性があります。</p> <p>この障害は、サーバの JVMID で <code>hostAddress</code> が常にデフォルト チャネル アドレスと見なされるために発生します。クライアントは <code>RJVMImpl.getCodeBase</code> を使用して、クラスのダウンロード元 URL を検索し、デフォルト チャネルに接続しようとして失敗します。</p> <p>8.1 では、JVMID の <code>hostAddress</code> はリクエストの発信元のチャネルに基づいています。従って <code>hostAddress</code> はチャネルのアドレスを適切に登録します。</p>	8.1
CR079543、 CR084568	<p>Unix では、WebLogic Server をインストールするディレクトリ (<code>BEAHOME</code>) の名前にスペースが含まれている場合、一部のサーバ起動スクリプトが失敗する可能性があります。</p>	

## クラスタに関する確認済みの問題

変更要求番号	説明	問題が修正されたりリリース
CR168598	<p>WebLogic Server 7.0 で、管理対象サーバの起動に異なるモードを指定している場合でも、管理対象サーバが管理サーバと同じモードで起動します。</p> <p>たとえば、管理サーバを開発モードで起動してから、コマンドラインオプションの <code>-Dweblogic.ProductionModeEnabled=true</code> を指定して管理対象サーバを起動しても、管理対象サーバは開発モードで起動します。</p> <p>管理サーバがプロダクション モードで起動している場合にのみ、管理対象サーバはプロダクション モードで起動します。</p>	
CR135944、 CR100373、 CR130592、 CR135864	<p>管理対象サーバが管理サーバとは独立したファイル システムまたはマシン上にある場合、管理対象サーバ独立モード（つまり、管理サーバがダウンしている状態）の管理対象サーバにステージングされた 1 フェーズ アプリケーションはデプロイされません。管理対象サーバが MSI モードで起動される場合、次のようなデプロイメント エラーが発生します。</p> <pre data-bbox="393 986 1067 1413"> &lt;Feb 5, 2004 5:06:36 PM EST&gt; &lt;Error&gt; &lt;Management&gt; &lt;141006&gt; &lt;Application 474001:Name=DefaultWebApp,Type=Application not found at \home2\mks\bea70sp4\user_projects\474001\application s\DefaultWebApp&gt;&lt;Feb 5, 2004 5:06:36 PM EST&gt; &lt;Error&gt; &lt;Management&gt; &lt;141004&gt; &lt;IOException opening application 474001:Name=DefaultWebApp,Type=Application, loading from path \home2\mks\bea70sp4\user_projects\474001\application s java.io.FileNotFoundException: File does not exist: \home2\mks\bea70sp4\user_projects\474001\application s\DefaultWebApp at weblogic.management.mbeans.custom.Application.adminL oad(Application.java:627) </pre> <p>このエラーは、1 フェーズ アプリケーションに対するステージングモードがなく、管理対象サーバでファイルがキャッシュ（ステージング）されないために発生します。</p>	8.1

## ドキュメントに関する確認済みの問題

変更要求番号	説明	問題が修正されたリリース
CR196210	<p>このドキュメントの以前のバージョンおよび他のサンプルのドキュメントに、Administration Server で MBeanHome インタフェースをルックアップする方法として</p> <pre>weblogic.management.Admin.getInstance().getAdminMBeanHome()</pre> <p>を使用するという誤った説明がありました。</p> <p><code>weblogic.management.Admin</code> クラスはパブリックではありません。非パブリックなこのクラスを使用する代わりに、JNDI を使用して MBeanHome を取得します。</p> <p>『WebLogic JMX Service プログラマーズ ガイド』の「例 : アクティブなドメインとサーバの判別」を参照してください。</p>	

## EJB に関する確認済みの問題

変更要求番号	説明	問題が修正されたリリース
CR112260	<p>EJB の &lt;security-permission&gt; 要素が Administration Console に表示されず、security-permission 属性に対する記述子の検証が失敗しました。Bean がデプロイされる時に、java.vm.version プロパティを変更することができました。</p> <p>weblogic-ejb-jar.xml に次のスタンザが含まれていました。</p> <pre data-bbox="380 696 1042 925"> &lt;security-permission&gt;   &lt;description&gt;Optional explanation goes   here&lt;/description&gt;  &lt;security-permission-spec&gt;     grant { permission   java.util.PropertyPermission "java.vm.vendor",   "read"; } ;   &lt;/security-permission-spec&gt; &lt;/security-permission&gt; &lt;/weblogic-ejb-jar&gt; </pre> <p><b>回避策:</b> Administration Console で &lt;security-permission&gt; の表示と変更ができるようにコードを変更しました。</p> <p>必要なセキュリティ パーミッションを正しくコンフィグレーションするには、次の手順を実行する必要があります。</p> <ol data-bbox="380 1081 1049 1477" style="list-style-type: none"> <li data-bbox="380 1081 1049 1168">1. &lt;security-permission-spec&gt; の付与パーミッションを { permission java.util.PropertyPermission "java.vm.vendor", "read"; } に変更します。</li> <li data-bbox="380 1185 1049 1411">2. %WL_HOME%\server\lib\weblogic.policy ファイルに次の行を追加します。  <pre data-bbox="420 1255 1042 1341"> add grant codeBase "file:/&lt;Your user_projects dir&gt;/YourDomain/lib/-" {permission   java.security.AllPermission; } ; </pre> 動的に生成されるスタブの classpath は lib であるため、この行が必要です。</li> <li data-bbox="380 1428 1049 1477">3. -Djava.security.manager オプションを指定してサーバを起動します。</li> </ol>	

## デプロイメントに関する確認済みの問題

変更要求番号	説明	問題が修正されたリリース
CR099037	クラスタのデフォルト Web アプリケーションをコンフィグレーションし、その Web アプリケーションをクラスタ内の単一のサーバまたはクラスタに割り当てるか再デプロイすると、デプロイメントが失敗します。	

## JDBC に関する確認済みの問題

変更要求番号	説明	問題が修正されたリリース
CR173725	<p>Oracle 10g thin ドライバを使用する場合に、無効なバッチの値の例外が発生しました。分析の結果、バッチ サイズに対する Oracle thin ドライバの制限が原因であることがわかりました。制限は 16383 (short の最大値の半分) のようです。</p> <p><b>回避策:</b> バッチ サイズを制限するか 920 ドライバを使用します。</p> <p>Oracle によるとこのバグは修正済みであり、10.1.0.2 ドライバにバックポートされています。</p>	

## JMS に関する確認済みの問題

変更要求番号	説明	問題が修正されたリリース
CR108664	メッセージが既に列挙値時間の期限を過ぎている場合でも、キュー ブラウザは期限切れのメッセージを返します。	8.1



## セキュリティに関する確認済みの問題

変更要求番号	説明	問題が修正されたりリリース
CR096349、 CR133465	<p>デフォルトでは、あるサイトの MBean ホームを、JNDI を介して匿名ユーザが取得できます。MBean ホームを通じて、多数のコンフィグレーション MBean を取得したり調べたりできます。デフォルトでは、この状況でセキュリティの脆弱性は確認されていませんが、BEA Systems では、コンフィグレーション データを慎重に保護することがセキュリティのベスト プラクティスであると考えています。</p> <p>この問題に対する推奨事項の詳細については、「<a href="#">Security Advisory BEA04-43.01</a>」を参照してください。</p>	8.1
CR125639	<p><code>weblogic/security/SSL/SSLSocketFactory.java</code> ファイルにある注意事項が正しくありません。正しい注意事項は次のとおりです。</p> <p>JSSE によって使用される <code>getDefault()</code> メソッドは、現在サーバで使用されている Certicom SSL 実装ではなく、古い Phaos 実装を使用するように初期化されたインスタンスを返すため、この <code>SSLSocketFactory</code> を JSSE にプラグインすることはできません。Certicom の実装はサーバの SSL 信頼コンフィグレーションに従ってコンフィグレーションされるのに対し、Phaos 実装では信頼性のある証明書をコンフィグレーションできません。JSSE から Certicom 実装にアクセスする必要がある場合は、<code>weblogic.security.SSL.SSLSocketFactory.getDefaultJSSE()</code> メソッドによって返される <code>SocketFactory</code> インスタンスをラップするクラスのインスタンスを返すために、<code>javax.net.SocketFactory</code> クラスを拡張して、その <code>getDefault()</code> メソッドを実装できます。以下のコード例では、<code>javax.net.SocketFactory</code> クラスを拡張して、JSSE から Certicom 実装にアクセスする方法を示します。</p> <pre>public class MySSLSocketFactory extends javax.net.SocketFactory { private static MySSLSocketFactory defFactory; private javax.net.SocketFactory factory; private MySSLSocketFactory() { factory = weblogic.security.SSL.SSLSocketFactory. getDefaultJSSE(); } public static javax.net.SocketFactory getDefault() { if (defFactory == null) defFactory = new MySSLSocketFactory(); return defFactory; } public Socket createSocket(InetAddress host, int port) throws IOException { return factory.createSocket(host, port); } // 他の 3 つの抽象 createSocket // メソッドを同様に実装する ... }</pre>	

## WebLogic Tuxedo Connector に関する確認済みの問題

変更要求番号	説明	問題が修正されたリリース
CR074356	<p>tBridge へのリダイレクトをコンフィグレーションするときに、リダイレクトのコンフィグレーション情報が config.xml ファイルの誤った場所に表示されます。リダイレクトのコンフィグレーションが tBridgeGlobal MBean の子として表示されます。tBridgeGlobal MBean と同じレベルに表示される必要があります。</p> <p><b>回避策:</b> config.xml ファイルを手動で編集してください。</p>	8.1

## Web サービスに関する確認済みの問題

変更要求番号	説明	問題が修正されたリリース
CR091230	<p>Microsoft Windows で、clientgen Ant タスクは、名前の大文字 / 小文字が異なるだけのエンティティを上書きする可能性があります。この問題は、Windows では大文字と小文字が区別されないために発生します。</p>	

# WebLogic Server 7.0 サービス パック 4 の 確認済みの問題

WebLogic Server 7.0 SP4 では、以下の問題が確認されています。

## jDriver に関する確認済みの問題

変更要求番号	説明	問題が修正 されたり リリース
CR094209	<p>分散トランザクションと Oracle Server 9.2.0.x を使用する WebLogic Server アプリケーションでは、Oracle Thin ドライバ 9.2.0.x に関する問題が発生します。</p> <p><b>回避策:</b>パッチ 2717235 を当てて Oracle 9.2.0.3 ドライバを使用します。Oracle の Web サイトを確認して、必要なパッチを適用してください。</p> <p>上記の回避策に加えて、以下の調整方法によって問題が緩和される可能性があります。</p> <ul style="list-style-type: none"><li>WebLogic Server が接続を待たなくても XA リソースでのアクティビティを完了できるように、コンフィグレーションされてる JDBC 接続プールの接続数をデフォルト実行スレッドの数と同じ値に設定する。</li></ul>	

## JSP に関する確認済みの問題

変更要求番号	説明	問題が修正されたりリリース
CR061104	<p>Windows システムで JSP コンパイラを実行した場合、出力ファイルの名前は常に小文字で作成されます。この動作を防止し、クラス名で使用されている文字の大小を維持するには、システムプロパティ <code>weblogic.jsp.windows.caseSensitive</code> を <code>true</code> に設定します。プロパティの設定は、JSP をコンパイルするときに次のコマンドを使用してコマンドラインで行うことができます。</p> <pre>java -Dweblogic.jsp.windows.caseSensitive=true weblogic.jspc *.jsp</pre> <p>あるいは、次のコマンドを WebLogic Server の起動スクリプトに挿入します。</p> <pre>-Dweblogic.jsp.windows.caseSensitive=true</pre>	

## Web サービスに関する確認済みの問題

変更要求番号	説明	問題が修正されたりリリース
CR092268	<p>プロキシのユーザ名とパスワードが送信されないため、ファイアウォールの内側からプロキシ経由で <code>clientgen</code> を使用できない場合があります。</p>	

# WebLogic Server 7.0 サービス パック 3 の 確認済みの問題

WebLogic Server 7.0 SP3 では、以下の問題が確認されています。

## ブラウザに関する確認済みの問題

変更要求番号	説明	問題が修正されたりリリース
CR106615	HP-UX 11.0 および 11i 上で Netscape バージョン 4.79 を使用する 場合、バス エラーが原因でブラウザ セッションが失敗する可能性 があります。	
CR107134	(JRockit)。Linux 上で動作する Oracle サーバ バージョン 8.1.7.3 を 使用し、JVM として JRockit を使用する場合、新しい EJB を作成 すると end-of-file エラーが発生することがあります。Oracle の次 のバージョン 8.1.7.4 にはこの問題はありません。	

## 確認済みの問題

変更要求番号	説明	問題が修正 されたり リリース
CR108206	<p>HP コードの libCsup.2 によって送出される例外は WebLogic Server jDriver のコードで処理されず、Weblogic Server でコアダンプが発生する可能性があります。この問題は、HP コード、ネイティブコード、および Java の 3 層のコードがある場合に発生します。問題は Oracle 9.2.0 の実行中に観察されました。</p> <p><b>回避策</b> : Oracle 9.2.0.3 のクライアント ライブラリを使用します。</p>	
CR094639	<p>EJB の security-constraint にワイルドカード URL (*) を追加すると、セッション タイムアウト後の req.getSession(false) からの戻り値が変わります。security-constraint にワイルドカードではない単一の URL が含まれる場合、req.getSession(false) からの戻り値は null です。security-constraint にワイルドカードではない URL が含まれていて、さらにワイルドカード (*) URL も含まれる場合、タイムアウトが発生しても、req.getSession(false) からの戻り値は有効なセッションになります。問題が修正されるリリース : WebLogic Server 7.0 SP5</p>	

## EJB に関する確認済みの問題

変更要求番号	説明	問題が修正されたリリース
CR082058	EJB がエンタープライズ アプリケーションの一部としてデプロイされる場合、アプリケーション全体を再デプロイしないと EJB を再デプロイできません。	8.1
CR096713	一部の EJBCacheRuntime オブジェクト ID に対して snmpwalk コマンドを実行すると、その RuntimeMBeans は削除されますが、オブジェクト ID は管理情報ベースにそのまま存在するため、結果が返されません (150.x.x 実行時 EJB エンティティ ホーム、458.x.x 実行時 EJB ステートフル ホーム、170.x.x 実行時 EJB ステートレス ホーム、137.x.x 実行時 EJBCache、136.x.x 実行時 EJB キャッシュ モニタ、150.x.x 実行時 EJB エンティティ ホーム、160.x.x 実行時 EJB メッセージ駆動型、161.x.x 実行時 EJB プール、458.x.x 実行時 EJB ステートフル ホーム、170.x.x 実行時 EJB ステートレス ホーム、455.x.x 実行時コネクタ接続プール、および 312.x.x JTA 実行時 JTA 統計)。	

## ツールに関する確認済みの問題

変更要求番号	説明	問題が修正されたリリース
CR092278	weblogic.Deployer ユーティリティを使用して Web アプリケーションを再デプロイするには、コマンドラインで <code>-source</code> および <code>-target</code> オプションの両方を指定する必要があります。 <code>-source</code> および <code>-target</code> オプションを指定しない場合、ツールは成功したことを示しますが Web アプリケーションは再デプロイされません。	



## Web サービスに関する確認済みの問題

変更要求番号	説明	問題が修正されたリリース
CR101160	<p>サービス固有の例外の複合データ型が、次のように、JAX-RPC (Java API for XML-Based RPC) 1.0 の仕様に厳密に準拠して処理されません。</p> <ul style="list-style-type: none"><li>• <b>セクション 4.3.6 WSDL Fault</b>、サービス固有の例外：メッセージ部分が <code>xsd:complexType</code> である <code>wsdl:fault</code> 要素の含まれる WSDL から <code>clientgen Ant</code> タスクによって JAX-RPC スタブが生成される場合、生成されたスタブは <code>xsd:complexType</code> を間違って Java の単純型にマッピングします。つまり、クライアントアプリケーションがそのスタブを使用して Web サービスを呼び出し、サービスから (<code>javax.xml.rpc.SOAPFaultException</code> ではなく) サービス固有の例外が送出されたときに、クライアントアプリケーションはその例外から完全な複合データにアクセスできないということです。 この問題の回避策はありません。</li><li>• <b>セクション 5.5.5 Methods</b>、例外、4 番目の項目：WebLogic Web サービス ランタイムは、サービス固有の Java 例外クラスの複数のフィールドを <code>xsd:complexType</code> ではなく間違って単純 XML データ型にマッピングします。 <b>回避策</b>：WebLogic Web サービスを実装する際に、サービス固有の例外ではなく常に <code>javax.xml.rpc.SOAPFaultException</code> を送出します。</li></ul>	

## WebLogic Server 7.0 サービス パック 2 の 確認済みの問題

WebLogic Server 7.0 SP2 では、以下の問題が確認されています。

### HPUX 11i ユーザはパッチの適用が必要

WebLogic Server 7.0 SP2 には、正しくないバージョンの `hpux11/libweblogicunix1.sl` ライブラリがバンドルされていました。適切なライブラリを入手するために、HPUX 11i ユーザは次の場所からパッチをダウンロードして適用する必要があります。

<http://commerce.bea.com/products/weblogicserver/suppljar.jsp>

パッチを適用するには、次の手順に従います。

1. パッチをダウンロードします。
2. それを `$WLHOME/weblogic700/server/lib` に展開します。

```
jar -xvf CR102314_700sp2.jar
```

これで、`hpux11/libweblogicunix1.sl` が新しいライブラリに置き換えられます。

3. `.jar` ファイルを `CLASSPATH` に追加します。

### JDK 1.4.1 のサポートと SSL に関する確認済みの 問題

JDK 1.4.1 クライアントで SSL を使用するには、`jre\lib` ディレクトリから Java 2 SDK `jsse.jar` を削除する必要があります。

```
ディレクトリ : C:\j2sdk1.4.1\jre\lib:
```

```
08/29/2002 01:10a 462,386 jsse.jar 1 File(s) 462,386 b
```

## Administration Console に関する確認済みの問題

変更要求番号	説明	問題が修正されたリリース
CR110791	Administration Console にカスタム ヘルプ ページを追加する場合は、『 <a href="#">Administration Console の拡張</a> 』で説明されているコンソール拡張を使用します。  代わりの手段で <code>console.war</code> にカスタム ヘルプ ファイルを配置する場合は、必ず <code>console.war</code> を <code>WL_HOME/server/lib/weblogic.jar</code> に追加してください。そうしないと、サーバの起動時に <code>console.war</code> が再生成されたときにカスタム ファイルが消去されます。	なし
CR066112	Administration Console を使用するデプロイメント手順が、6.x のリリースから変更されています。詳細については、「 <a href="#">WebLogic Server デプロイメント</a> 」を参照してください。	なし
CR069828	Web サービスのデプロイメント記述子は、Administration Console を使用して編集できなくなりました。	なし
CR069927	[サーバ   モニタ   プロセス出力] を選択し、いずれかの表示オプションを選択すると、出力は表示されません。ノード マネージャが動作している状態で [ノード マネージャ出力を表示] を選択すると、ノード マネージャがクラッシュします。	7.0 SP3
CR073164	Administration Console を使って、管理サーバの現在のドメイン以外のドメインを編集することはできません。	
CR074339	Netscape 4.7 を使って Console にアクセスすると、コンフィグレーション オブジェクトの作成に対する説明テキストが表示されません。	8.1
CR094174	Internet Explorer のバージョン 5 を使用している場合、Administration Console の左ペインで [JDBC   マルチプール] を選択すると、 <code>RuntimeException</code> が送出されます。	

## コアサーバに関する確認済みの問題

変更要求番号	説明	問題が修正されたりリリース
CR048834	<p>Solaris で JDK 1.3.1 を使用している際に、現地時間を設定しているにも関わらず WebLogic のログ メッセージのタイムスタンプがグリニッジ標準時 (GMT) で記録されている場合には、<code>jre/lib/i18n.jar</code> を更新する必要があります。</p> <p>この JAR ファイルから、JRE がタイムスタンプをローカライズするために使用するリソースが失われている場合があります。</p> <p><b>回避策：</b></p> <ul style="list-style-type: none"> <li>適切な <code>i18n.jar</code> を備えた JRE をインストールし直す</li> <li><code>i18n.jar</code> がプラットフォームでサポートされていない場合は、カスタム ResourceBundle を、ロードされる場所から CLASSPATH に実装し直す</li> <li>インストール後に、タイムゾーンが正しく設定されていること、および必要がある場合には DST がサポートされていることを確認する</li> </ul>	
CR059543	<p>Administration Console を使って XML レジストリを変更して <code>DocumentBuilderFactory</code> テキストフィールドに NULL を設定した後、同じ XML レジストリを表示すると、<code>DocumentBuilderFactory</code> テキストフィールドが空になります。WebLogic Server のデフォルトの <code>DocumentBuilderFactory</code> である <code>weblogic.apache.xerces.jaxp.DocumentBuilderFactoryImpl</code> が設定されていなければなりません。</p>	
CR063799	<p><code>VirtualHostMBean</code> で <code>getLogFileName</code> を呼び出すと、不正なパスが返ります。</p>	8.1
CR065299、 CR067642	<p>Java HotSpot(TM) サーバ VM は、WebLogic Server 7.0 ではまだサポートされていません。動作確認されたこの VM のクライアントバージョンを使用してください。</p>	

変更要求番号	説明	問題が修正されたりリリース
CR066581	<p>javax.management.timer.Timer の WebLogic Server バージョン weblogic.management.timer.Timer は、JMX タイマと互換性がありませんでした。</p> <p>WebLogic のクラスを使ってインスタンス化すると、作成されるタイマは、リスナの実行時に、WebLogic のスレッド プールを使用し、ユーザの ID を受け付けるようになります。</p>	
CR066723	<p>開発モードでは、管理対象サーバの検出はデフォルトでオフになっています。これをオンにするには、WebLogic Server を起動するコマンドライン引数に、 -Dweblogic.manager.discover=false を追加します。</p>	
CR067087	<p>現在、WebLogic Server では、weblogic-applications.xml ファイル内でセキュリティ パーミッションを設定する手段は提供されていません。</p>	
CR067814	<p>LDAP ブラウザを使って WebLogic Server 上の組み込み LDAP サーバを表示する場合は、次の入力を使用してみてください。</p> <pre>host: localhost port: 7003 Anonymousbind LDAP Version 3</pre>	
CR067868	<p>HotSpot JVM を使用した場合、大きなデプロイメントを実行すると、サーバがハングする可能性があります。</p> <p><b>回避策:</b> -XX:MaxPermSize をもっと大きな値に設定してみてください。</p>	

変更要求番号	説明	問題が修正 されたり リリース
CR068186	<p>マルチホームの Windows 2000 システムに WebLogic Server Version 7.0 をインストールすると、コンフィグレーションされたサーバは、誤って、各 IP アドレスをシステム自体に対する共通の DNS 名に解決します。たとえば、MyWin2KMachine という名前で下記の IP アドレスを使用するマルチホームの Windows 2000 マシンについて考えてみます。</p> <p style="padding-left: 40px;">192.168.1.105 192.168.1.106 192.168.1.107</p> <p>システムに複数の WebLogic Server インスタンスをインストールし、サーバのリスン アドレス属性に対して IP アドレス 192.168.105、192.168.1.106、および 192.168.1.107 を指定すると、IP アドレスを解決する際に、すべてのサーバが同じ DNS 名 MyWin2KMachine を生成します。このため、リモートサーバのソースへのアクセス時に、衝突が発生する可能性があります。</p> <p><b>回避策：</b></p> <ul style="list-style-type: none"> <li>• c:/winnt/system32/drivers/hosts ファイルをテキスト エディタで編集し、各 IP アドレスに明示的に名前を設定します。次に例を示します。 <p style="padding-left: 40px;">192.168.1.105 MyWin2KMachine1 192.168.1.106 MyWin2KMachine2 192.168.1.107 MyWin2KMachine3</p> </li> <li>• マシンの IP アドレスに対する名前が DNS サーバに登録されている場合は、サーバのリスン アドレス属性をコンフィグレーションする際に、IP アドレスではなく、外部から取得可能なこの DNS 名を使用します。たとえば、192.168.105、192.168.106、192.168.107 を使うのではなく、MyWin2KMachine1、MyWin2KMachine2、および MyWin2KMachine3 を使ってサーバをコンフィグレーションします。この問題は、Windows NT などの他のソフトウェアプラットフォームには関係ありません。</li> </ul>	

変更要求番号	説明	問題が修正されたりリリース
CR068228	互換性モードを使用していない場合、コマンドライン管理ツールを使用すると、guest に対するパスワードの入力を求められる場合があります。「guest」のデフォルトパスワードは「guest」です。	
CR068247	WebLogic Server 7.0 に付属する JSP は、初めてアクセスされる前にはコンパイルされていません。コンパイル済みの JSP をデプロイすると、WebLogic Server はそれを再コンパイルし、/wlnotdelete/ ディレクトリに格納する場合があります。	
CR068361	バージョン 1.1 の EJB を Bean 管理による永続性からコンテナ管理による永続性に移行する場合、Bean を選択して [Container-managed persistence] を選択すると、詳細オプションと自動キー生成オプションが生成されなければなりません、生成されません。	
CR069278	<p>コマンドラインを使って接続プールを作成するとき、シェルが 1 組の引用符を削除するようなプラットフォームでは、プロパティ文字列でエスケープできるのは二重引用符だけです。</p> <p><b>回避策:</b> この過程で引用符が削除されないプラットフォームにおいては、1 組の二重引用符だけを使用する必要があります。また、接続プロパティごとに property 引数を繰り返して使用すれば、プロパティの文字列を引用符で囲む必要がなくなります。次に例を示します。</p> <pre data-bbox="373 1142 1038 1203">-property Property user=scott -property Password tiger -property URL jdbc:weblogic:oracle:ora817</pre>	
CR069602	WebLogic Server の JTA の実装は、JTA 仕様バージョン 1.0.1a に準拠するように更新されました。これには、Transaction.commit() メソッドと TransactionManager.setTimeout(int) メソッドに対する小さな変更も含まれています。	

## 確認済みの問題

変更要求番号	説明	問題が修正されたりリリース
CR072646	<p>Windows 2000 で Netscape バージョン 4.75 以前を使用する場合、ファイルの関連付けが、Internet Explorer で使用される同様の関連付けから正しく変換されません。このため、Windows の [ スタート ] メニューから [About WebLogic Server 7.0]、[About Examples]、および [About Pet Store] を開くときに、以下のような動作が起きます。</p> <ul style="list-style-type: none"><li>• ( ブラウザにページが表示された場合でも ) 対象の URI が見つからなかったことを示すエラー メッセージが表示される。</li><li>• ブラウザが既に動作している場合、HTML ページがブラウザに表示されない。</li></ul>	
CR072940	<p>config.xml ファイルに &lt;SecurityConfiguration&gt; 要素が設定されていない場合、またはこの要素に資格属性および名前属性がない場合は、サーバがこの属性を作成しようとします。</p> <p>config.xml ファイルが書き込み可能な場合、Java クライアントは、この資格の値をローカルな JVM メモリ空間にキャッシュします。そして、JVM の有効期間の間、JVM クライアントはこの資格情報を使用します。しかし、JVM が再起動すると、資格情報がなくなるので、JVM クライアントはアクセスを拒否されます。</p> <p><b>回避策</b> : 資格が既に設定されている時点で、config.xml ファイルを 1 回保存します。</p>	
CR073395	<p>Oracle データベースを使用し、複数のクライアントで Sun の Pet Store アプリケーションを実行しようとする、問題が発生します。</p>	
CR073647	<p>現時点では、AdministrationPort を使用できるのは Administration ロールのユーザだけです。</p>	8.1



変更要求番号	説明	問題が修正されたりリリース
CR073737	<p>アクティブなドメインに対する管理サーバを再起動すると、管理サーバが管理対象サーバを発見できない場合があります。この現象は、以下のような状況で発生します。</p> <p>Windows オペレーティングシステムを実行するマルチホームのマシンでは、管理サーバと1つ以上のプロセスが、同じリスンポートを使用します(ただし IP アドレスは異なります)。これは、Windows オペレーティングシステムの制約のためです。再起動した管理サーバが管理対象サーバを発見できない場合、管理サーバを通してドメインに発行したコマンドは失敗します。たとえば、管理対象サーバ独立 (MSI) モードで起動した管理対象サーバは、管理サーバが制御を回復した後で、シャットダウン コマンドに応答できない可能性があります。</p>	
CR073793	<p>非対象のモジュールに対する URI が <code>config.xml</code> で設定されていない場合は、非対象のモジュールを含むアプリケーションをデプロイできません。</p>	8.1
CR074296	<p>場合によっては、Administration Console で複数の WebLogic Tuxedo Connector WTCResources MBean を作成できます。特定の WTCServer MBean コンフィグレーションに対して許可される WTCResources MBean は1つだけです。余分なコンフィグレーションを削除し、WebLogic Tuxedo Connector を再起動する必要があります。</p>	8.1
CR074452	<p>WebLogic Server に大きな負荷がかかっているときに JVM からエラーが送出される場合は、Java のメモリ ヒープを増やしてみてください。たとえば、WebLogic Server の起動コマンドを修正し、次の指定を追加します。MEM_ARGS="-XX:MaxPermSize=128m -Xms512m -Xmx512m"</p>	
CR074969	<p>JNDI 名のない EJB のデプロイで問題が発生する場合があります。WebLogic Server の旧リリースで行われていたように、JNDI 名を含む EJB を引き続きデプロイすることをお勧めします。</p>	

## 確認済みの問題

変更要求番号	説明	問題が修正されたりリリース
CR085169	<p>Iplanet 4.1SP9 を使用している場合、<code>socket.getInputStream().readLine()</code> を使用する Java クライアントの <code>read_timeout</code> が原因で、Java クライアントはプラグインから POST データを読み込むことができません。これは <code>netbuf_getbytes()</code> の問題によるものです。</p> <p><b>回避策:</b> WebLogic Server とともに Iplanet 6.0SP2 を使用すると、この問題が解消します。</p>	
CR099314	<p><a href="http://developer.java.sun.com/developer/bugParade/bugs/4755829.html">http://developer.java.sun.com/developer/bugParade/bugs/4755829.html</a> で説明されているとおり、JDK 1.3.1_04 および <code>nohup</code> コマンドには問題があります。</p> <p><code>nohup</code> コマンドは、<code>nohup</code> の呼び出しに関係なく障害を発生します (親シェルが終了すると WebLogic Server プロセスが終了する)。この問題は <code>ksh</code> (Solaris 8 のデフォルトシェル) で確認されましたが、他のシェルでも発生することが考えられます。<code>bash</code> シェルでは、この問題は起こりません。</p> <p><b>回避策:</b></p> <ul style="list-style-type: none"><li>• <a href="http://askbea-int.beasys.com/askbea/wls/S-15924.html">http://askbea-int.beasys.com/askbea/wls/S-15924.html</a> で回避策 (<code>-Xrs</code> フラグ) が提供されているが、その代償としてスレッドダンプを失うことになる。</li><li>• "<code>nohup startWeblogic &amp;</code>" の代わりに "<code>(startWeblogic &gt;output 2&gt;&amp;1 &amp;)</code>" を使用する。</li></ul>	
CR110028、 CR120181	<p>WebLogic Server 7.0 のスタック スレッド検出は、デフォルト実行キューのスレッドに制限されます。</p> <p>スタック スレッド検出は <code>weblogic.t3.srvr.CoreHealthService</code> の <code>checkStuckThreads</code> メソッドによって実行され、プライベート変数 <code>policyName</code> (スタック スレッドをチェックするキュー名) は静的に「default」に設定されます。</p>	

## EJB に関する確認済みの問題

変更要求番号	説明	問題が修正されたリリース
CR089847	<p>WebLogic Server の EJB デプロイメント プロパティの多くには、パフォーマンスのために最適化されたデフォルト値があります。これらのデフォルト値には、EJB 仕様に準拠していないものもあります。</p> <p><b>回避策</b>：WebLogic Server を EJB 仕様に準拠させるには、以下のプロパティを設定します。</p> <p>weblogic-ejb-jar.xml で、<code>enable-call-by-reference</code> を <code>False</code> に設定する。</p> <p>weblogic-cmp-jar.xml で、<code>include-updates</code> を <code>True</code> に設定する。</p> <p>weblogic-cmp-jar.xml で、<code>check-exists-on-method</code> を <code>True</code> に設定する。</p>	なし

## 確認済みの問題

変更要求番号	説明	問題が修正されたリリース
CR089630	<p>OracleThinDriver では、表に BLOB および LONG RAW の両方のカラムがある場合に、SELECT 句で LONG RAW カラムより前に BLOB カラムがリストされていても、LONG RAW カラムが先にストリームから読み込まれる必要があります。</p> <p>現在 EJB コンテナではこのようなケースは処理されず、<code>java.sql.SQLException "Stream already closed"</code> という例外が送出されます。</p> <p><b>回避策:</b> JDriver を使用してください。</p>	
CR097323	<p><code>.jar</code> ファイルに未コンパイルのクラスおよびインタフェースが含まれる場合に、クラスタ内の 1 つのサーバインスタンスに EJB をデプロイまたは再デプロイすると、固定デプロイメントと呼ばれる問題が生じることが知られています。</p> <p>未コンパイルの EJB は、デプロイ中にクラスタ内の各サーバインスタンスにコピーされますが、この EJB はデプロイ先のサーバインスタンス上でしかコンパイルされません。その結果、EJB の対象とされていないクラスタのサーバインスタンスには、コンパイルの過程で生成される、EJB の呼び出しに必要なクラスが存在しません。別のサーバインスタンスのクライアントが固定 EJB を呼び出そうとしても失敗し、RMI レイヤでアサーションエラーが送出されます。</p> <p><b>回避策:</b> クラスタ内の 1 つのサーバインスタンスに EJB をデプロイまたは再デプロイする場合は、生成されたクラスがすべてのサーバインスタンスに必ずコピーされ、クラスタ内のすべてのノードから利用可能になるように、デプロイ前にその EJB をコンパイルしてください。</p>	なし

## サンプルと Pet Store に関する確認済みの問題

変更要求番号	説明	問題が修正されたリリース
CR055626	<p>webapp セキュリティ サンプルの login.jsp ファイルでは、ユーザ ログイン時の URL エンコーディングが無効になっています。login.jsp ファイルは次の場所にあります。</p> <p>SAMPLES_HOME\server\src\examples\webapp\security login.jsp ファイルには次のコードがあります。</p> <pre>&lt;form method="POST" action="j_security_check"&gt; URL エンコーディングを許可するために使用するコードの例を次に示します。  &lt;FORM ACTION="&lt;%= response.encodeURL("j_security_check") %&gt;" METHOD=POST&gt;</pre>	7.0 SP2
CR068009	<p>Microsoft VB スクリプトのバージョンのため、ZeroClient jCom サンプルを実行するには Excel 2000 以降が必要です。</p>	
CR071260	<p>サンプルおよび Pet Store に対するパスワードとユーザ名が、WebLogic Server 7.0 ベータ版から変更されました。</p> <p>installadministrator/installadministrator は使用できません。代わりに、weblogic および weblogic を使用してください。</p>	なし
CR073395	<p>Pet Store では、複数のクライアントが同じリソースを使用する際に問題があります。この問題に対する対処は Sun において行われています。</p>	

## 確認済みの問題

変更要求番号	説明	問題が修正されたリリース
CR074004	<p>SAMPLES_HOME\server\src\examples\security ディレクトリにある package-summary.html ファイルでは、Security サンプルの概要の表の中で audit サンプルと cert サンプルが参照されています。これらのサンプルは、WebLogic Server 7.0 で非推奨になった API を使用しているため、WebLogic Server の配布キットから削除されています。</p> <p>また、SSLClient、JAAS、および Net の各サンプルの正確な説明は、以下のとおりです。</p> <p><b>SSLClient</b> : クライアントとして機能する WebLogic Server から別の WebLogic Server またはアプリケーション サーバに、発信 SSL 接続または相互 SSL 接続を確立する方法を示します。</p> <p><b>JAAS</b> : ユーザ名とパスワードを検証し、有効な場合は Trader ステートレス セッション EJB を実行することで、Java Authentication and Authorization Service (JAAS) の使い方を示します。</p> <p><b>Net</b>: weblogic.security.net.ConnectionFilter インタフェースと ConnectionFilterRulesListener インタフェースを実装する接続フィルタの作り方を示します。接続フィルタを利用すれば、クライアントの IP アドレスとプロトコルおよびローカルとリモートのポート番号を基にして、WebLogic Server 環境へのクライアント接続を拒否したり許可したりすることができます。</p>	なし
CR080288	<p>ant run を使用してクライアントを実行するときに、RMI クラスタ サンプルに問題があります。クライアントに対する java 呼び出しに間違った引用符が含まれています。</p>	

## インストールに関する確認済みの問題

変更要求番号	説明	問題が修正されたリリース
CR085531	<p>SmartUpdate を介して WebLogic Server のアップグレードおよびダウングレードを正常に行った後、次にアップグレードを行うと以下のエラーが発生します。</p> <p>Installer is not compatible with the installed product. Your product directory contains installed components that are not available in the current installer. Please check your installer to ensure it matches the product distribution on your system.</p> <p><b>回避策:</b> カスタマ サポートの担当者にお問い合わせください。</p>	

## 相互運用性に関する確認済みの問題

変更要求番号	説明	問題が修正されたリリース
CR093343 および CR093347	<p>Java 2 SDK 1.4.1 上で動作する Java クライアントが Java 2 SDK 1.3.1 上で動作するサーバにアクセスすると、以下のようなエラーが発生します。</p> <ul style="list-style-type: none"> <li>• getDate() メソッドと setDate() メソッドが矛盾した結果を返す。</li> <li>• クライアントから getTimeStamp() および setTimeStamp() を呼び出すと、クライアントは「class not found」エラーを受け取る。</li> </ul> <p>これらは Sun Java 2 SDK (Ticket #808980) で確認された問題です。Sun では Java 2 SDK 1.5 で問題を修正する予定です。</p> <p><b>回避策:</b> クライアント アプリケーションを SDK 1.3.1 上で実行してください。</p>	
CR092507、CR095479	<p>7.0 SP1 サーバと 7.0 SP2 サーバを相互運用する場合、MarshalException/UnmarshalException を送出するリモート メソッドでアンマーシャリングの問題が発生する可能性があります。この問題は、MarshalException および UnmarshalException が SP2 では正しい完了ステータスにマップされているものの、SP1 では誤った完了ステータスにマップされていることが原因で発生します。</p>	
CR095804	<p>WebLogic Server 7.0 SP2 クライアントが WebLogic 8.1 サーバインスタンスから Coordinator リモート オブジェクトを取得しようとする、リモート スタブを CoordinatorOneway インタフェースにキャストするときに ClassCastException が送出されません。このため、8.1 調整サーバへコミットを渡すことができません。</p>	7.0 SP3



## jDriver に関する確認済みの問題

変更要求番号	説明	問題が修正されたリ リース
CR094209	<p>分散トランザクションと Oracle Server 9.2.0.x を使用する WebLogic Server アプリケーションでは、Oracle Thin ドライバ 9.2.0.x に関する問題が発生します。</p> <p><b>回避策:</b> パッチ 2717235 を当てて Oracle 9.2.0.3 ドライバを使用します。Oracle の Web サイトを確認して、必要なパッチを適用してください。</p> <p>上記の回避策に加えて、以下の調整方法によって問題が緩和される可能性があります。</p> <ul style="list-style-type: none"> <li>WebLogic Server が接続を待たなくても XA リソースでのアクティビティを完了できるように、コンフィグレーションされてる JDBC 接続プールの接続数をデフォルト実行スレッドの数と同じ値に設定する。</li> </ul>	

## JMS に関する確認済みの問題

変更要求番号	説明	問題が修正された リリース
CR075438	<p>分散トピックにデプロイされ、その分散トピックの 2 つのメンバーをホストするサーバインスタンスをターゲットとしたメッセージ駆動型 Bean (MDB) は、分散トピックのすべてのメンバーにもデプロイされます。</p> <p>MDB がメンバーの送り先名に固定されるので、これは当然予期される動作です。したがって、JMS サーバに 2 つのメンバーがある場合、メンバーごとに 1 つ、2 つの MDB がデプロイされます。</p>	

## JRokit に関する確認済みの問題

変更要求番号	説明	問題が修正されたリリース
CR093270	<p>WebLogic Server 7.0 SP2 が JRokit JVM で動作している場合に、UDDI でプライベート レジストリに対して新しく Web サービスをパブリッシュしようとする、次のようなエラーメッセージが表示されてパブリッシュは失敗します。"An error has occurred E_fatalError(10500): a serious technical error has occurred while processing the request."</p> <p>これは UDDI 実装 (AUDDI 1.0) のバグによるものです。</p>	

## JTA に関する確認済みの問題

変更要求番号	説明	問題が修正されたリリース
CR089804	<p>トランザクションで、同じ接続プールを指している 2 つの Tx データソースから接続を使用することはできません。2 番目の Tx データソースから接続を取得しようとする、XA_PROTO エラーを受け取ります。</p> <p>WebLogic Server は内部的に、データソース名ではなく接続プール名を使用してリソースを特定します。異なる Tx データソースを使用して同じ接続プールから 2 つの別々の接続を取得する場合、その接続は同じリソースとして扱われます。トランザクションでは、1 番目の Tx データソースに対して start/enlist が呼び出され、その後 2 番目の Tx データソースに対して start/enlist が試行されます。サーバはリソースを同じものと見なすため、新しいトランザクションを開始する前に end/delist が呼び出されないことが原因で XA_PROTO エラーが送出されます。</p> <p><b>回避策：</b></p> <ul style="list-style-type: none"><li>トランザクション中に 1 つの Tx データソースの接続のみを使用する。</li><li>異なる接続プールを指している 2 つの Tx データソースを使用する。</li></ul> <p>いずれの回避策でも、パフォーマンスが低下することはありません。</p>	

## ノード マネージャに関する確認済みの問題

変更要求番号	説明	問題が修正されたリリース
CR074230	<p>WebLogic Server にデプロイされたアプリケーションは、<code>ServerMBean.getRootDirectory()</code> メソッドで取得したルート ディレクトリを基準にファイルのルックアップを行う必要があります ( デフォルトは「.」ディレクトリ)。たとえば、ファイルのルックアップを実行するには、次のようなコードを使用します。</p> <pre>String rootDir = ServerMBean.getRootDirectory(); // アプリケーションのルート ディレクトリ File f = new File(rootDir + File.separator + "foo.in"); 次のようなシンプルなコードは使用しません。 File f = new File("foo.in"); ノード マネージャを使用して起動されるサーバにアプリケーションがデプロイされる場合は、代わりに次のメソッド呼び出しを使用します。 String rootDir // アプリケーションのルート ディレクトリ if ((rootDir = ServerMBean.getRootDirectory()) == null) rootDir = ServerStartMBean.getRootDirectory(); File f = new File(rootDir + File.separator + "foo.in"); ServerStartMBean.getRootDirectory() メソッドは、ノード マネージャを使用して起動するようにサーバをコンフィグレーションする際に指定した Root Directory 値を取得します。これは、Administration Console の [ コンフィグレーション   リモートスタート ] ページで指定された Root Directory 属性に対応します。 ノード マネージャによってリモートから起動された管理対象サーバは、[ リモートスタート ] タブで設定された <code>weblogic.RootDirectory</code> を認識しません。この問題が起こらないようにするには、[ ドメイン   サーバ   コンフィグレーション   デプロイメント ] タブでステージングディレクトリの絶対パスを入力します。相対パスを入力しないでください。WebLogic Server のルート ディレクトリを含めた絶対パスを入力します。</pre>	
CR073078	<p>管理サーバは、ノード マネージャに対する SSL 接続のセットアップを試みる際、管理サーバのルート ディレクトリで証明書ファイルの自動検索は行いません。管理サーバのルート ディレクトリに SSL 証明書ファイルを格納してある場合は、Administration Console の [ 接続   SSL ] タブを使って、証明書ファイルへの絶対パスをコンフィグレーションする必要があります。</p>	8.1

変更要求番号	説明	問題が修正されたリリース
CR075142	<p>管理サーバとノード マネージャが同時に TCP ソケットのクローズを試みる場合があり、以下のエラー メッセージがノード マネージャのログ ファイルに記録されます。</p> <pre data-bbox="364 534 1049 649">&lt;Apr 18, 2002 5:32:55 PM EDT&gt; &lt;Error&gt; &lt;NodeManager@localhost:5555&gt; &lt;OutputHandler: done2 socket close failed, reason: java.net.SocketException: Connection reset by peer.&gt;</pre> <p>この問題は、ノード マネージャまたは管理サーバの動作には影響を与えません。</p>	
CR094722	<p>クラスタ内の管理対象サーバを起動すると、管理サーバで socketWrite エラーが送出されます。</p> <p><b>回避策:</b> 問題を無視するか、管理対象ノードを再起動してください。</p>	

## プラグインに関する確認済みの問題

変更要求番号	説明	問題が修正されたりリリース
CR095161	<p>負荷がかかると、Apache プラグインがバックエンドの WebLogic Server インスタンスから CONNECTION REFUSED エラーを受け取る可能性があります。このエラーは、Apache Server バージョン 1.3.22 と Apache プラグイン、および Solaris 8 上で動作する WebLogic Server で確認されました。</p>	

## RMI に関する確認済みの問題

変更要求番号	説明	問題が修正されたりリリース
CR099693	<p>クライアントが t3 プロトコルを使用してファイアウォールから WebLogic Server にアクセスすると、サーバは初期化されていない接続上でメッセージをルーティングしようとしたため、エラーが発生しました。</p> <p>サーバの ExternalDNSName が指定されました。初期接続時に、クライアントはサーバインスタンスの ExternalDNSName へリクエストを送信しました。サーバインスタンスからの応答にある JVMID は、サーバの内部 IP アドレスと内部 DNS 名を必要としていました。クライアントからの以降のリクエストは認識されませんでした。</p> <p>クライアントがファイアウォールを介して t3 で接続する場合に問題が発生します。回避策は、サーバインスタンスの内部 DNS 名を外部 IP アドレスにマップするエントリを外部 DNS サーバに追加することです。</p>	7.0 SP03

## セキュリティに関する確認済みの問題

BEA のセキュリティ勧告については、dev2dev Web サイトにある BEA の Advisories & Notifications ページを参照してください。このページでは、セキュリティ関連のパッチをダウンロードしたり、新しいセキュリティ勧告の通知を受けるよう登録したりできます。Advisories & Notifications ページには、<http://dev2dev.bea.com/advisories> でアクセスできます。

BEA では、BEA 製品に関するセキュリティ問題をご報告いただくための電子メール アドレス (security-report@bea.com) も用意しています。

以下の表に、このサービス パック リリースで確認されたセキュリティ関連の問題を示します。

変更要求番号	説明	問題が修正されたりリリース
CR086158	<a href="http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA02-22.jsp">http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA02-22.jsp</a> のセキュリティ勧告に関する情報を確認してください。	
CR092546	以下の仕様で定義されているように、ロール名の正しい構文は XML_NMTOKEN でなければなりません。 <a href="http://www.w3.org/TR/REC-xml#NT-Nmtoken">http://www.w3.org/TR/REC-xml#NT-Nmtoken</a>	
CR093652	JDK 1.4.1 で構築されたクライアントが SSL を使用するには、Java 2 SDK の jsse.jar ファイルを JAVA_HOME\jre\lib ディレクトリから削除する必要があります。	
CR094566	このリリースの WebLogic Server で nCipher JCE プロバイダを使用するには、JDK 1.3.1 を使用してクライアントを構築する必要があります。	
CR096589	weblogic.servlet.security.ServletAuthentication.s trong() は壊れており、クラス キャスト例外により失敗します。 <b>回避策:</b> パッチ CR096589_70sp2.jar をインストールしてください。	7.0 SP3

## サードパーティ製ツールに関する確認済みの問題

変更要求番号	説明	問題が修正されたりリリース
CR071527	Resonate (WebLogic Server をバックエンドサーバとして使用するサードパーティ製ロード バランシング ツール) でセッションの永続化に関する問題があります。Resonate では大文字と小文字を区別するため、URLEncoding がクッキーを JSESSIONID ではなく jsessionid として送信した場合、Resonate はクッキーを検出できません。	



## ツールに関する確認済みの問題

変更要求番号	説明	問題が修正されたリリース
CR064893	<p>以下のデプロイメント記述子要素は、WebLogic Builder では変更できません。</p> <p>ejb-jar.xml の要素</p> <ul style="list-style-type: none"> <li>• cascade-delete (EB/Advanced)</li> <li>• weblogic ejb xml</li> <li>• transaction-isolation</li> <li>• description、display-name、large-icon、small-icon</li> <li>• exclude-list</li> <li>• unchecked</li> <li>• use-caller-identity</li> </ul> <p>weblogic-cmp.xml の要素</p> <ul style="list-style-type: none"> <li>• enable-dynamic-queries (EB/Advanced)</li> <li>• isolation-level (Methods/Transactions)</li> <li>• multiple table mapping</li> <li>• db-cascade-delete</li> <li>• field-group, group-name (*)</li> <li>• relationship-caching、caching-name、caching-element</li> <li>• validate-db-schema-with</li> <li>• database-type</li> </ul>	
CR065250	<p>データベースに対するドライバクラスがクライアントにないと、WebLogic Builder はサーバ上のデータベースへのアクセスに失敗する場合があります。</p>	
CR066466	<p>WebLogic Builder では、操作をキャンセルすることはできません。</p>	
CR066668	<p>Linux の場合、WebLogic Builder のファイル ブラウザはカレントディレクトリではなくルートで開きます。</p>	
CR068152	<p>WebLogic Builder の EJB 要素では、Add、Edit、および Delete の各動作が遅くなる場合があります。</p>	8.1

## 確認済みの問題

変更要求番号	説明	問題が修正されたリリース
CR068165	WebLogic Builder において、[Resource Links] の [Add] ダイアログ、[WebLogic 設定]、および [J2EE Links] では、アプリケーション スコープのデータ リソースの選択が正しく表示されません。	
CR068620	WebLogic Builder では、表のカラムのサイズを正しく変更できません。	8.1
CR069568	WebLogic Builder では、EJB メソッド パーミッションの 1 つのメソッドに複数のロールをマップできません。	
CR071833、 CR072541	WebLogic Builder の RAR サポートには、デプロイメント記述子の生成が含まれていません。	
CR072700、 CR074591	WebLogic Builder のデプロイメント ダイアログ ボックスを使用してアプリケーションをアンデプロイする場合に、そのダイアログでアンデプロイメントが示されません。このダイアログでは、デプロイメントの失敗したアプリケーションがデプロイされたと報告される場合もあります。	7.0 SP3
CR073975	WebLogic Builder では、メソッドまたはトランザクション属性を指定しなくてもトランザクションが追加されてしまいます。	7.0 SP3
CR074210	WebLogic Builder ではオプティミスティックな同時方式が機能しません。	8.1
CR074322、 CR074856	WebLogic Builder では、編集処理が取り消せない場合があります。	8.1
CR074566	(include-update のように) WebLogic Builder の BeanGrid/Edit を通じて修正されたファインダまたはコンテナ管理の永続性の値は、個別のパネルで表されない場合があります。	8.1
CR074622	WebLogic Builder のポップアップ ダイアログを使用してコンテナ管理の永続性フィールドを編集する際に、カラムの型を OracleCLOB に設定して、[CMP] ツリー ノードをクリックした場合に、そのカラムの型の変更が [CMP] パネルに表示されません。	
CR074728	WebLogic Builder では、メソッドをローカル メソッドに変更することができません。	

変更要求番号	説明	問題が修正されたリリース
CR074754	一部のトランザクション属性を編集すると、WebLogic Builder で記述子が重複して作成される場合があります。	
CR074961	WebLogic Builder を使用して <code>web.xml</code> 記述子を生成すると、7.0 より前のセキュリティ プロトコルに関連するセキュリティ ロール「 <code>guest</code> 」が作成されます。	
CR075129	ステートレス セッション Bean をステートフルに切り替えた場合に、WebLogic Builder ではそのモジュールが変更されていることが示されない場合があります。	
CR075228	トランザクションまたはパーミッション メソッドを削除すると、WebLogic Builder ではアサーション エラーが送出される場合があります。	8.1
CR075586	WebLogic Server サンプル サーバまたはサンプルで WebLogic Builder を使用する場合は、WebLogic Builder を起動する前に <code>setExamplesEnv (%SAMPLES_HOME%\server\config\examples</code> にあります) を実行し、PointBase データベース サーバを実行するために必要なファイルをクラスパスに追加する必要があります。これを行わないと、データベースのエラー メッセージが表示されます。	
CR075753	記述子の一部の要素は WebLogic Builder の検証をパスしても、 <code>ejbc</code> によってエラーと判定されることがあります。	
CR075924	WebLogic Builder では、EAR でネストされた WAR ファイルの記述子ファイルが正しく生成されない場合があります。	8.1
CR075936	WebLogic Builder では、 <code>estimated-bean-size</code> 要素で無効な値が生成される場合があります。	8.1

## WebLogic Tuxedo コネクタに関する確認済みの問題

変更要求番号	説明	問題が修正されたリリース
CR067275	RMI over IIOP を使用するクラスタの WebLogic Tuxedo Connector では、ロード バランシングとフェイルオーバーはサポートされていません。	

変更要求番号	説明	問題が修正されたリリース
CR079630	<p>関連 ID を設定する場合、<code>setJMSCorrelationID(String)</code> メソッドの使用を考えるユーザが多いと思われます。このメソッドでは 32 文字の文字列を使用し、それを 64 バイトの配列に格納します。JMS では文字列は UTF-16BE として格納されます。tBridge が Tuxedo から受け取る関連 ID も 32 文字ですが、こちらは 32 バイトです。tBridge では <code>setJMSCorrelationIDAsBytes(byte[])</code> を使用して、受信側の JMS キュー用のメッセージが作成されます。この 2 つの文字列は ASCII では同じ文字列になることもありますが、長さが異なるために同じ文字列とは見なされません。</p> <p>関連 ID を Tuxedo から JMS キュー内に返された ID と比較する必要がある場合には、バイト配列を作成して関連 ID の 16 進数値を格納します。その後 <code>setJMSCorrelationIDAsBytes()</code> および <code>getJMSCorrelationIDAsBytes()</code> メソッドを使用して ID を作成し、Tuxedo から JMS キュー内に返された ID と比較します。</p> <p>たとえば関連 ID の文字列が "1234567890ABCDEFGHIJKLMNQRSTUUV" の場合、次のようになります。</p> <pre>private byte[] coridbyte={0x31,0x32,0x33,0x34,0x35,0x36,0x37,                           0x38,0x39,0x30,0x41,0x42,0x43,0x44,                           0x45,0x46,0x47,0x48,0x49,0x4a,0x4b,                           0x4c,0x4d,0x4e,0x4f,0x50,0x51,0x52,                           0x53,0x54,0x55,0x56}; msg.setJMSCorrelationIDAsBytes(coridbyte); corIDAsBytes = msg.getJMSCorrelationIDAsBytes();</pre> <p><code>corIDAsBytes</code> に Tuxedo から返された関連 ID と正しく比較できる値が格納されます。</p>	

## Web サービスに関する確認済みの問題

変更要求番号	説明	問題が修正されたリリース
CR067190	WebLogic Web サービスでは、complexTypes および同じ名前の複数の要素がある XML スキーマはサポートされていません。	
CR072890	<p>コマンドライン バージョンの servicegen Ant タスクでは、Ant タスク属性に対応する以下のオプションがサポートされていません。</p> <ul style="list-style-type: none"> <li>-overwrite</li> <li>-contextURI</li> <li>-includeEJBs</li> <li>-excludeEJBs</li> <li>-generateTypes</li> </ul> <p>コマンドライン バージョンの clientgen Ant タスクでは、Ant タスク属性に対応する以下のオプションがサポートされていません。</p> <ul style="list-style-type: none"> <li>-autotype</li> <li>-overwrite</li> </ul>	8.1
CR073661	無効な web-services.xml ファイルを使用して不正確に記述された WebLogic Web サービスは WebLogic Server にデプロイはされませんが、そのサービスを呼び出そうとするクライアント アプリケーションからアクセスできません。	
CR083493	Web サービス サンプルのパッケージに含まれる messages サンプルと multicomponent サンプルには、ダウンロード可能な client.jar ファイルを生成するために必要なコードがありません。これは、用意されているクライアントで動作するサンプルの機能には影響しません。	
CR094688	<p>7.0 Web サービスのポータブル スタブ クライアント .jar を使用すると、java.lang.NoClassDefFoundError が発生する可能性があります。</p> <p><b>回避策</b>：完全な CLASSPATH を使用してクライアントを実行し、VersionMaker ユーティリティにクラス間の相違を指定してください。</p>	

変更要求番号	説明	問題が修正されたリリース
CR095629	<p>メモリ使用量を最適化するために新しい <code>weblogic.webservice.cachewsdl</code> フラグを使用する場合、サーバ ログに以下のような誤った警告メッセージが表示されます。</p> <pre data-bbox="364 522 999 574">&lt;Warning&gt; &lt;Management&gt; &lt;141087&gt; &lt;Unrecognized property: webservice.cachewsdl&gt;</pre> <p>この警告は正しくありません。このメッセージがあってもフラグは有効になります。</p>	
CR101160	<p>サービス固有の例外の複合データ型が、次のように、JAX-RPC (Java API for XML-Based RPC) 1.0 の仕様に厳密に準拠して処理されません。</p> <ul data-bbox="364 782 1083 1425" style="list-style-type: none"><li data-bbox="364 782 1083 1173">• <b>セクション 4.3.6 WSDL Fault</b>、サービス固有の例外：メッセージ部分が <code>xsd:complexType</code> である <code>wsdl:fault</code> 要素の含まれる WSDL から <code>clientgen Ant</code> タスクによって JAX-RPC スタブが生成される場合、生成されたスタブは <code>xsd:complexType</code> を間違って Java の単純型にマッピングします。つまり、クライアントアプリケーションがそのスタブを使用して Web サービスを呼び出し、サービスから (<code>javax.xml.rpc.SOAPFaultException</code> ではなく) サービス固有の例外が送出されたときに、クライアントアプリケーションはその例外から完全な複合データにアクセスできないということです。 この問題の回避策はありません。</li><li data-bbox="364 1194 1083 1425">• <b>セクション 5.5.5 Methods</b>、例外、4 番目の項目：WebLogic Web サービス ランタイムは、サービス固有の Java 例外クラスの複数のフィールドを <code>xsd:complexType</code> ではなく間違って単純 XML データ型にマッピングします。 この問題を回避するには、WebLogic Web サービスを実装する際に、サービス固有の例外ではなく常に <code>javax.xml.rpc.SOAPFaultException</code> を送出します。</li></ul>	

## XML に関する確認済みの問題

変更要求番号	説明	問題が修正されたリリース
CR070547	Solaris で XML ストリーミング API を使用する場合、一部の文字参照が NT の場合と同じ文字に解決されません。	

## WebLogic Server 7.0 サービス パック 1 の確認済みの問題

この節では、WebLogic Server サービス パック 1 で確認済みの問題を示します。

### Administration Console に関する確認済みの問題

変更要求番号	説明	問題が修正されたリリース
CR066112	Administration Console を使用するデプロイメント手順が、6.x のリリースから変更されています。詳細については、「 <a href="#">WebLogic Server デプロイメント</a> 」を参照してください。	なし
CR069828	Web サービスのデプロイメント記述子は、Administration Console を使用して編集できなくなりました。	なし
CR069927	[サーバ   モニタ   プロセス出力] を選択し、いずれかの表示オプションを選択すると、出力は表示されません。ノード マネージャが動作している状態で [ノード マネージャ出力を表示] を選択すると、ノード マネージャがクラッシュします。	



変更要求番号	説明	問題が修正されたリリース
CR073164	Administration Console を使って、管理サーバの現在のドメイン以外のドメインを編集することはできません。	
CR074339	Netscape 4.7 を使って Console にアクセスすると、コンフィグレーション オブジェクトの作成に対する説明テキストが表示されません。	8.1
CR074677	WebLogic Server のバージョン 6.x で作成されたドメインをアップグレードすると、ユーザ system は、Administration Console へのアクセスを明示的に許可しなくなります。Console にアクセスするには、Administrators グループのメンバになる必要があります。	7.0 SP2

## コア サーバに関する確認済みの問題

変更要求番号	説明	問題が修正されたリリース
CR110058	<p>動的なグループ属性 memberURL で、LDAP URL のフィルタ部分に、フォワード スラッシュ (「/」) などの予約文字がある文字列値が含まれている場合、MalformedURLException が発生する可能性があります。フィルタ部分が、LDAP ディレクトリ サーバ内のフォワード スラッシュ文字を含むオブジェクトに解決される場合、このような URL になる可能性があります。</p> <p><b>回避策:</b> ディレクトリで動的グループを使用する場合は、LDAP ディレクトリのオブジェクト名に予約文字 (「:」 「?」 「/」) を使用しないでください。netscape.ldap.LDAPUrl ユーティリティ クラス (オープン ソース) はこれらの文字を処理できません。</p>	
CR030981	JDBC T3 ドライバは WebLogic Server 6.0 で非推奨になっており、このリリースから削除されました。	なし

## 確認済みの問題

変更要求番号	説明	問題が修正されたリリース
CR048834	<p>Solaris で JDK 1.3.1 を使用している際に、現地時間を設定しているにも関わらず WebLogic のログ メッセージのタイムスタンプが GMT で記録されている場合には、<code>jre/lib/i18n.jar</code> を更新する必要があります。</p> <p>この JAR ファイルから、JRE がタイムスタンプをローカライズするために使用するリソースが失われている場合があります。</p> <p><b>回避策</b>：その場合、以下の方法で修正します。</p> <ul style="list-style-type: none"><li>適切な <code>i18n.jar</code> を備えた JRE をインストールし直す</li><li><code>i18n.jar</code> がプラットフォームでサポートされていない場合は、カスタム <code>ResourceBundle</code> を、ロードされる場所から <code>CLASSPATH</code> に実装し直す</li><li>インストール後に、タイムゾーンが正しく設定されていること、および必要がある場合には DST がサポートされていることを確認する</li></ul>	
CR059543	<p>Administration Console を使って XML レジストリを変更して <code>DocumentBuilderFactory</code> テキスト フィールドに <code>NULL</code> を設定した後、同じ XML レジストリを表示すると、<code>DocumentBuilderFactory</code> テキスト フィールドが空になっています。WebLogic Server のデフォルトの <code>DocumentBuilderFactory</code> である <code>weblogic.apache.xerces.jaxp.DocumentBuilderFactoryImpl</code> が設定されていなければなりません。</p>	
CR063799	<p><code>VirtualHostMBean</code> で <code>getLogFileName</code> を呼び出すと、不正なパスが返ります。</p>	8.1
CR065299、 CR067642	<p>Java HotSpot(TM) サーバ VM は、WebLogic Server 7.0 ではまだサポートされていません。動作確認されたこの VM のクライアントバージョンを使用してください。</p>	

変更要求番号	説明	問題が修正されたリリース
CR066581	WebLogic Server には、 <code>javax.management.timer.Timer</code> の独自バージョンである <code>weblogic.management.timer.Timer</code> があります。WebLogic のクラスを使ってインスタンス化すると、作成されるタイマは、リスナの実行時に、WebLogic のスレッド プールを使用し、ユーザの ID を受け付けるようになります。	
CR066723	開発モードでは、管理対象サーバの検出はデフォルトでオフになっています。これをオンにするには、WebLogic Server を起動するコマンドライン引数に、 <code>-Dweblogic.manager.discover=false</code> を追加します。	
CR067087	現在、WebLogic Server では、 <code>weblogic-applications.xml</code> ファイル内でセキュリティ パーミッションを設定する手段は提供されていません。	
CR067814	LDAP ブラウザを使って WebLogic Server 上の組み込み LDAP サーバを表示する場合は、次の入力を使用してみてください。 <code>host: localhost</code> <code>port: 7003</code> <code>Anonymousbind</code> <code>LDAP Version 3</code>	
CR067868	HotSpot JVM を使用した場合、大きなデプロイメントを実行すると、サーバがハングする可能性があります。 <b>回避策:</b> <code>-XX:MaxPermSize</code> をもっと大きな値に設定してみてください。	

変更要求番号	説明	問題が修正されたリリース
CR068186	<p>マルチホームの Windows 2000 システムに WebLogic Server Version 7.0 をインストールすると、コンフィグレーションされたサーバは、誤って、各 IP アドレスをシステム自体に対する共通の DNS 名に解決します。たとえば、MyWin2KMachine という名前で下記の IP アドレスを使用するマルチホームの Windows 2000 マシンについて考えてみます。</p> <pre> 192.168.1.105 192.168.1.106 192.168.1.107 </pre> <p>システムに複数の WebLogic Server インスタンスをインストールし、サーバのリスン アドレス属性に対して IP アドレス 192.168.105、192.168.1.106、および 192.168.1.107 を指定すると、IP アドレスを解決する際に、すべてのサーバが同じ DNS 名 MyWin2KMachine を生成します。このため、リモートサーバのリソースへのアクセス時に、衝突が発生する可能性があります。</p> <p><b>回避策：</b>以下のいずれかを実行します。</p> <ul style="list-style-type: none"> <li> <p>c:/winnt/system32/drivers/hosts ファイルをテキストエディタで編集し、各 IP アドレスに明示的に名前を設定します。次に例を示します。</p> <pre> 192.168.1.105 MyWin2KMachine1 192.168.1.106 MyWin2KMachine2 192.168.1.107 MyWin2KMachine3 </pre> </li> <li> <p>マシンの IP アドレスに対する名前が DNS サーバに登録されている場合は、サーバのリスン アドレス属性をコンフィグレーションする際に、IP アドレスではなく、外部から取得可能なこの DNS 名を使用します。たとえば、192.168.105、192.168.106、192.168.107 を使うのではなく、MyWin2KMachine1、MyWin2KMachine2、および MyWin2KMachine3 を使ってサーバをコンフィグレーションします。この問題は、Windows NT などの他のソフトウェア プラットフォームには関係ありません。</p> </li> </ul>	

変更要求番号	説明	問題が修正されたリリース
CR068228	互換性モードを使用していない場合、コマンドライン管理ツールを使用すると、guest に対するパスワードの入力を求められる場合があります。「guest」のデフォルトパスワードは「guest」です。	
CR068247	WebLogic Server 7.0 に付属する JSP は、初めてアクセスされる前にはコンパイルされていません。コンパイル済みの JSP をデプロイすると、WebLogic Server はそれを再コンパイルし、/wlnotdelete/ディレクトリに格納する場合があります。	
CR068361	バージョン 1.1 の EJB を Bean 管理による永続性からコンテナ管理による永続性に移行する場合、Bean を選択して [Container-managed persistence] を選択すると、詳細オプションと自動キー生成オプションが生成されなければなりません、生成されません。	
CR069278	<p>コマンドラインを使って接続プールを作成するとき、シェルが 1 組の引用符を削除するようなプラットフォームでは、プロパティ文字列でエスケープできるのは二重引用符だけです。この過程で引用符が削除されないプラットフォームにおいては、1 組の二重引用符だけを使用する必要があります。また、接続プロパティごとに property 引数を繰り返して使用すれば、プロパティの文字列を引用符で囲む必要がなくなります。次に例を示します。</p> <pre data-bbox="364 1078 1029 1130">-property Property user=scott -property Password tiger -property URL jdbc:weblogic:oracle:ora817</pre>	
CR069602	<p>WebLogic Server の JTA の実装は、JTA 仕様バージョン 1.0.1a に準拠するように更新されました。これには、Transaction.commit() メソッドと TransactionManager.setTimeout(int) メソッドに対する小さな変更も含まれています。</p>	
CR070429	現時点では、前回の正常な COMMO コンフィグレーションを使って再起動する手段はありません。	7.0 SP2

## 確認済みの問題

変更要求番号	説明	問題が修正されたリリース
CR071954	<p>バージョン 6.x の <code>weblogic.management.Admin</code> クラスには、静的メソッド <code>getActiveDomain()</code> がありました。WebLogic Server 7.0 では、<code>getActiveDomain()</code> は静的ではなくなっています。その代わりとして、<code>Admin.getInstance().getAdminMBeanHome().getActiveDomain();</code> を使用します。</p> <p>この代替 API の使い方の例については、『WebLogic JMX Service プログラマーズ ガイド』の「<a href="#">アクティブなドメインとサーバの判別</a>」を参照してください。</p>	
CR072646	<p>Windows 2000 で Netscape バージョン 4.75 以前を使用する場合、ファイルの関連付けが、Internet Explorer で使用される同様の関連付けから正しく変換されません。このため、Windows の [ スタート ] メニューから [About WebLogic Server 7.0]、[About Examples]、および [About Pet Store] を開くときに、以下のような動作が起きます。</p> <ul style="list-style-type: none"><li>• (ブラウザにページが表示された場合でも) 対象の URI が見つからなかったことを示すエラー メッセージが表示される。</li><li>• ブラウザが既に動作している場合、HTML ページがブラウザに表示されない。</li></ul>	
CR072940	<p><code>config.xml</code> ファイルに <code>&lt;SecurityConfiguration&gt;</code> 要素が設定されていない場合、またはこの要素に資格属性および名前属性がない場合は、サーバがこの属性を作成しようとします。<code>config.xml</code> ファイルが書き込み可能な場合、Java クライアントは、この資格の値をローカルな JVM メモリ空間にキャッシュします。そして、JVM の有効期間の間、JVM クライアントはこの資格情報を使用します。しかし、JVM が再起動すると、資格情報がなくなるので、JVM クライアントはアクセスを拒否されます。</p> <p><b>回避策:</b> 資格が既に設定されている時点で、<code>config.xml</code> ファイルを 1 回保存します。</p>	
CR073395	<p>Oracle データベースを使用し、複数のクライアントで Sun の Pet Store アプリケーションを実行しようとすると、問題が発生します。</p>	

変更要求番号	説明	問題が修正されたリリース
CR073647	現時点では、AdministrationPort を使用できるのは Administration ロールのユーザだけです。	8.1
CR073737	<p>アクティブなドメインに対する管理サーバを再起動すると、管理サーバが管理対象サーバを発見できない場合があります。この現象は、以下のような状況で発生します。</p> <p>Windows オペレーティングシステムを実行するマルチホームのマシンでは、管理サーバと1つ以上のプロセスが、同じリスポートを使用します(ただし IP アドレスは異なります)。これは、Windows オペレーティングシステムの制約のためです。再起動した管理サーバが管理対象サーバを発見できない場合、管理サーバを通してドメインに発行したコマンドは失敗します。たとえば、管理対象サーバ独立 (MSI) モードで起動した管理対象サーバは、管理サーバが制御を回復した後で、シャットダウン コマンドに応答できない可能性があります。</p>	8.1
CR073793	非対象のモジュールに対する URI が config.xml で設定されていない場合は、非対象のモジュールを含むアプリケーションをデプロイできません。	8.1
CR074296	場合によっては、Administration Console で複数の WebLogic Tuxedo Connector WTCResources MBean を作成できます。特定の WTCServer MBean コンフィグレーションに対して許可される WTCResources MBean は1つだけです。余分なコンフィグレーションを削除し、WebLogic Tuxedo Connector を再起動する必要があります。	8.1
CR074452	<p>WebLogic Server に大きな負荷がかかっているときに JVM からエラーが送出される場合は、Java のメモリ ヒープを増やしてみてください。たとえば、WebLogic の起動コマンドを修正し、次の指定を追加します。MEM_ARGS="-XX:MaxPermSize=128m -Xms512m -Xmx512m"</p>	8.1
CR074969	JNDI 名のない EJB のデプロイで問題が発生する場合があります。WebLogic Server の旧リリースで行われていたように、JNDI 名を含む EJB を引き続きデプロイすることをお勧めします。	8.1

## 確認済みの問題

変更要求番号	説明	問題が修正されたリリース
CR085169	<p>Iplanet 4.1SP9 を使用している場合、<code>socket.getInputStream().readLine()</code> を使用する Java クライアントの <code>read_timeout</code> が原因で、Java クライアントはプラグインから POST データを読み込むことができません。これは <code>netbuf_getbytes()</code> の問題によるものです。</p> <p><b>回避策:</b> WebLogic Server とともに Iplanet 6.0SP2 を使用します。</p>	
CR086784	<p>Sun Microsystem の JVM 1.3.1_03 を使用して Windows 2000 上で WebLogic Server 7.0 SP1 を起動すると、デッドロックの問題が発生します。この問題は JVM の過度の最適化により発生します。</p> <p><b>回避策:</b> JVM の <code>-XX:-UseCompilerSafepoints</code> オプションを使用して、過度の最適化を無効にします。</p>	



## EJB に関する確認済みの問題

変更要求番号	説明	問題が修正されたリリース
CR124381	<p>EAR ファイルは、デプロイメント時にメモリ不足エラーを発生します。例外は次のとおりです。</p> <pre>&lt;Sep 30, 2003 8:28:41 AM MDT&gt; &lt;Info&gt; &lt;Deployer&gt; &lt;BEA-149060&gt; &lt;Module ttri_depl.jar of application OSS_TTRI successfully transitioned from prepared to unprepared on server myserver.&gt;</pre> <pre>&lt;Sep 30, 2003 8:28:41 AM MDT&gt; &lt;Info&gt; &lt;Deployer&gt; &lt;BEA-149033&gt; &lt;failed application OSS_TTRI on myserver&gt;</pre> <pre>&lt;Sep 30, 2003 8:28:41 AM MDT&gt; &lt;Info&gt; &lt;Management&gt; &lt;BEA-140009&gt; &lt;Configuration changes for the domain have been saved to the repository.&gt;</pre> <p><b>回避策:</b> weblogic-ejb-jar.xml で max-beans-in-free-pool の量設定を減らします。たとえば、 &lt;max-beans-in-free-pool&gt;2147483647&lt;/max-beans-in-free-pool&gt; この行を次のように変更します。 &lt;max-beans-in-free-pool&gt;100&lt;/max-beans-in-free-pool&gt;</p>	
CR089847	<p>WebLogic Server の EJB デプロイメント プロパティの多くには、パフォーマンスのために最適化されたデフォルト値があります。これらのデフォルト値には、EJB 仕様に準拠していないものもあります。</p> <p>WebLogic Server を EJB 仕様に準拠させるには、以下のプロパティを設定する必要があります。</p> <p>weblogic-ejb-jar.xml で、enable-call-by-reference を False に設定する。</p> <p>weblogic-cmp-jar.xml で、include-updates を True に設定する。</p> <p>weblogic-cmp-jar.xml で、check-exists-on-method を True に設定する。</p>	なし

## 確認済みの問題

変更要求番号	説明	問題が修正されたリリース
CR089630	<p>OracleThinDriver では、表に BLOB および LONG RAW の両方のカラムがある場合に、SELECT 句で LONG RAW カラムより前に BLOB カラムがリストされていても、LONG RAW カラムが先にストリームから読み込まれる必要があります。</p> <p>現在 EJB コンテナではこのようなケースは処理されず、<code>java.sql.SQLException "Stream already closed"</code> という例外が送出されます。</p> <p><b>回避策:</b> JDriver を使用してください。</p>	
CR097323	<p><code>.jar</code> ファイルに未コンパイルのクラスおよびインタフェースが含まれる場合に、クラスタ内の 1 つのサーバインスタンスに EJB をデプロイまたは再デプロイすると、固定デプロイメントと呼ばれる問題が生じることが知られています。</p> <p>デプロイメント中、未コンパイルの EJB はクラスタ内の各サーバインスタンスにコピーされますが、コンパイルはデプロイされているサーバインスタンス上でのみ行われます。その結果、クラスタ内で EJB の対象となっていないサーバインスタンスでは、EJB の呼び出しに必要な、コンパイル中に生成されるクラスが不足します。別のサーバインスタンスのクライアントが固定 EJB を呼び出そうとしても失敗し、RMI レイヤでアサーション エラーが送出されます。</p> <p><b>回避策:</b> クラスタ内の 1 つのサーバインスタンスに EJB をデプロイまたは再デプロイする場合は、生成されたクラスが、クラスタ内のすべてのノードに利用可能なすべてのサーバインスタンスに必ずコピーされるように、デプロイ前にその EJB を <code>ejbc</code> でコンパイルしてください。</p>	

## サンプルと Pet Store に関する確認済みの問題

変更要求番号	説明	問題が修正されたリリース
CR055626	<p>webapp security サンプルの login.jsp ファイルでは、ユーザログイン時の URL エンコーディングが無効になっています。login.jsp ファイルは次の場所にあります。</p> <p>SAMPLES_HOME\server\src\examples\webapp\security login.jsp ファイルには次のコードがあります。</p> <pre>&lt;form method="POST" action="j_security_check"&gt; URL エンコーディングを許可するために使用するコードの例を次に示します。  &lt;FORM ACTION="&lt;%= response.encodeURL("j_security_check") %&gt;" METHOD=POST&gt;</pre>	7.0 SP2
CR068009	<p>Microsoft VB スクリプトのバージョンのため、ZeroClient jCom サンプルを実行するには Excel 2000 以降が必要です。</p>	
CR071260	<p>サンプルおよび Pet Store に対するパスワードとユーザ名が、WebLogic Server 7.0 ベータ版から変更されました。installadministrator および installadministrator は使用できません。代わりに、weblogic および weblogic を使用してください。</p>	なし
CR073395	<p>Pet Store では、複数のクライアントが同じリソースを使用する際に問題があります。この問題に対する対処は Sun において行われています。</p>	

## 確認済みの問題

変更要求番号	説明	問題が修正されたリリース
CR074004	<p>SAMPLES_HOME\server\src\examples\security ディレクトリにある package-summary.html ファイルでは、Security サンプルの概要の表の中で audit サンプルと cert サンプルが参照されています。これらのサンプルは、WebLogic Server 7.0 で非推奨になった API を使用しているため、WebLogic Server の配布キットから削除されています。</p> <p>また、SSLClient、JAAS、および Net の各サンプルの正確な説明は、以下のとおりです。</p> <p><b>SSLClient</b> : クライアントとして機能する WebLogic Server から別の WebLogic Server またはアプリケーション サーバに、発信 SSL 接続または相互 SSL 接続を確立する方法を示します。</p> <p><b>JAAS</b> : ユーザ名とパスワードを検証し、有効な場合は Trader ステートレス セッション EJB を実行することで、Java Authentication and Authorization Service (JAAS) の使い方を示します。</p> <p><b>Net</b>: weblogic.security.net.ConnectionFilter インタフェースと ConnectionFilterRulesListener インタフェースを実装する接続フィルタの作り方を示します。接続フィルタを利用すれば、クライアントの IP アドレスとプロトコルおよびローカルとリモートのポート番号を基にして、WebLogic Server 環境へのクライアント接続を拒否したり許可したりすることができます。</p>	なし

変更要求番号	説明	問題が修正されたリリース
CR080288	<p>ant run を使用してクライアントを実行するときに、RMI クラスタ サンプルに問題があります。クライアントに対する java 呼び出しに間違った引用符が含まれています。</p>	
CR085521	<p>r3client Web サービス サンプルの package-summary.html ファイルは不完全です。このファイルは <code>SAMPLES_HOME\server\src\examples\webservices\r3client</code> ディレクトリにあります。</p> <p>このサンプルは WebLogic Web サービス クライアントを実行して、SOAPBuilders Interoperability Lab Round 3 テストスイートの実施方法を示すものです。</p> <p>サンプルをビルドするには、次の手順に従います。</p> <ol style="list-style-type: none"> <li>1. 「クイック スタート」で説明されているように、開発用シェルを設定します。</li> <li>2. <code>SAMPLES_HOME\server\src\examples\webservices\r3client</code> ディレクトリに移動します。SAMPLES_HOME は、サンプルの WebLogic Server ドメイン ディレクトリを表します。</li> <li>3. コマンドラインで以下の Java ant ユーティリティを実行して、サンプルをアセンブルおよびコンパイルします。 <pre>prompt&gt; ant</pre> </li> <li>4. コマンドラインで以下の Ant タスクを実行して、結果を HTML テーブルにコンパイルします。 <pre>prompt&gt; ant compileResults</pre> </li> </ol> <p><b>注意：</b> アプリケーションは WebLogic Server 上で動作する Web サービスではなく、SOAPBuilders Web サイトの Web サービスを呼び出すので、WebLogic Server を起動する必要はありません。</p> <p>出力をチェックするには、ブラウザで以下のファイルを開いて、HTML テーブルのテスト結果を参照します。</p> <p><code>SAMPLES_HOME\server\src\examples\webservices\r3client\results\summary.html</code></p>	7.0 SP2

## インストールに関する確認済みの問題

変更要求番号	説明	問題が修正されたリリース
CR085531	<p>SmartUpdate を介して WebLogic Server のアップグレードおよびダウングレードを正常に行った後、次にアップグレードを行うと以下のエラーが発生します。</p> <p>Installer is not compatible with the installed product. Your product directory contains installed components that are not available in the current installer. Please check your installer to ensure it matches the product distribution on your system.</p> <p><b>回避策:</b> カスタマ サポートの担当者にお問い合わせください。</p>	

## JMS に関する確認済みの問題

変更要求番号	説明	問題が修正されたリリース
CR066079	<p>WebLogic JMS は Sun Microsystems の JMS 仕様バージョン 1.0.b に完全に準拠しています。したがって、メッセージプロパティ名にスペースを含めることはできません。</p>	
CR075438	<p>分散トピックにデプロイされ、その分散トピックの 2 つのメンバーをホストするサーバインスタンスをターゲットとしたメッセージ駆動型 Bean (MDB) は、分散トピックのすべてのメンバーにもデプロイされます。</p> <p>MDB がメンバーの送り先名に固定されるので、これは当然予想される動作です。したがって、JMS サーバに 2 つのメンバーがある場合、メンバーごとに 1 つ、2 つの MDB がデプロイされます。</p>	

## ノード マネージャに関する確認済みの問題

変更要求番号	説明	問題が修正されたリリース
CR071591	<p>ノード マネージャは、HTTP 接続の開始を試みることで、管理対象サーバの障害を検出します。ノード マネージャが接続障害を受信した場合、ノード マネージャの出力で接続例外が示されることがあります。</p> <pre>&lt;Mar 19, 2002 2:30:04 PM EST&gt; &lt;Warning&gt; &lt;net&gt; &lt;000900&gt; &lt;Could not open connection java.net.ConnectException: Connection refused: connect</pre>	7.0 SP2
CR073078	<p>管理サーバは、ノード マネージャに対する SSL 接続のセットアップを試みる際、管理サーバのルート ディレクトリで証明書ファイルの自動検索は行いません。</p> <p><b>回避策:</b> 管理サーバのルート ディレクトリに SSL 証明書ファイルを格納してある場合は、Administration Console の [ 接続   SSL ] タブを使って、証明書ファイルへの絶対パスをコンフィグレーションする必要があります。</p>	8.1

変更要求番号	説明	問題が修正されたリリース
CR099693	<p>7.0 (G.A., SP01、および SP02) では、クライアントが t3 プロトコルを使用してファイアウォールから WebLogic Server にアクセスすると、サーバで次のエラーが発生しました。</p> <pre data-bbox="299 522 1018 921"> #####&lt;Feb 17, 2003 10:15:59 AM EST&gt; &lt;Error&gt; &lt;RJVM&gt; &lt;ewibeasbx01.merck.com&gt; &lt;edmServer&gt; &lt;ExecuteThread: '14' for queue: 'default'&gt; &lt;kernel identity&gt; &lt;&gt; &lt;000506&gt; &lt;Closing: weblogic.rjvm.t3.T3JVMConnection@23730e because of Server expected to route a message received over an uninitialized connection: 'JVMMessage from: '3684268472702687546S:172.16.2.2:[7001,7001,7002,7002,7001,7002,-1]:edmServer' to: 'OS:172.16.1.250:[7000,-1,-1,-1,-1,-1,-1]' cmd: 'CMD_IDENTIFY_REQUEST', QOS: '101', responseId: '0', invokableId: '0',flags: 'JVMIDs Sent, TX Context Not Sent', abbrev offset: '594''&gt; </pre> <p>サーバの ExternalDNSName が指定されました。初期接続時に、クライアントはサーバインスタンスの ExternalDNSName ヘリクエストを送信しました。サーバインスタンスからの応答にある JVMID は、サーバの内部 IP アドレスと内部 DNS 名を必要としていました。クライアントからの以降のリクエストは認識されませんでした。</p> <p>これは SP03 で解決された WebLogic Server 7.0 の既知の問題です。クライアントがファイアウォールを介して t3 で接続する場合に問題が発生します。</p> <p><b>回避策:</b> サーバインスタンスの内部 DNS 名を外部 IP アドレスにマップするエントリを外部 DNS サーバに追加します。</p>	7.0 SP03



変更要求番号	説明	問題が修正されたリリース
CR074230	<p>WebLogic Server にデプロイされたアプリケーションは、<code>ServerMBean.getRootDirectory()</code> メソッドで取得したルートディレクトリを基準にファイルのルックアップを行う必要があります (デフォルトは「.」ディレクトリ)。たとえば、ファイルのルックアップを実行するには、次のようなコードを使用します。</p> <pre>String rootDir = ServerMBean.getRootDirectory(); // アプリケーションのルート ディレクトリ File f = new File(rootDir + File.separator + "foo.in");</pre> <p>次のようなシンプルなコードは使用しません。</p> <pre>File f = new File("foo.in");</pre> <p>ノード マネージャを使用して起動されるサーバにアプリケーションがデプロイされる場合は、代わりに次のメソッド呼び出しを使用します。</p> <pre>String rootDir // アプリケーションのルート ディレクトリ if ((rootDir = ServerMBean.getRootDirectory()) == null) rootDir = ServerStartMBean.getRootDirectory(); File f = new File(rootDir + File.separator + "foo.in");</pre> <p><code>ServerStartMBean.getRootDirectory()</code> メソッドは、ノード マネージャを使用して起動するようにサーバをコンフィグレーションする際に指定した <code>Root Directory</code> 値を取得します。これは、Administration Console の [ コンフィグレーション   リモートスタート ] ページで指定された <code>Root Directory</code> 属性に対応します。</p> <p>ノード マネージャによってリモートから起動された管理対象サーバは、[ リモート スタート ] タブで設定された <code>weblogic.RootDirectory</code> を認識しません。</p> <p>この問題が起こらないようにするには、[ ドメイン   サーバ   コンフィグレーション   デプロイメント ] タブでステージングディレクトリの絶対パスを入力します。相対パスを入力しないでください。WebLogic Server のルートディレクトリを含めた絶対パスを入力します。</p>	

## 確認済みの問題

---

変更要求番号	説明	問題が修正されたリリース
CR075142	<p>管理サーバとノード マネージャが同時に TCP ソケットのクローズを試みる場合があり、以下のエラー メッセージがノード マネージャのログ ファイルに記録されます。</p> <pre data-bbox="299 534 989 649">&lt;Apr 18, 2002 5:32:55 PM EDT&gt; &lt;Error&gt; &lt;NodeManager@localhost:5555&gt; &lt;OutputHandler: done2 socket close failed, reason: java.net.SocketException: Connection reset by peer.&gt;</pre> <p>この問題は、ノード マネージャまたは管理サーバの動作には影響を与えません。</p>	

---

## RMI に関する確認済みの問題

変更要求番号	説明	問題が修正されたリリース
CR099693	<p>7.0 (G.A.、SP01、および SP02) では、クライアントが t3 プロトコルを使用してファイアウォールから WebLogic Server にアクセスすると、サーバで次のエラーが発生しました。</p> <pre> #####&lt;Feb 17, 2003 10:15:59 AM EST&gt; &lt;Error&gt; &lt;RJVM&gt; &lt;ewibeasbx01.merck.com&gt; &lt;edmServer&gt; &lt;ExecuteThread: '14' for queue: 'default'&gt; &lt;kernel identity&gt; &lt;&gt; &lt;000506&gt; &lt;Closing: weblogic.rjvm.t3.T3JVMConnection@23730e because of Server expected to route a message received over an uninitialized connection: 'JVMMMessage from: '3684268472702687546S:172.16.2.2:[7001,7001,7002,700 2,7001,7002,-1]:edm:edmServer' to: 'OS:172.16.1.250:[7000,-1,-1,-1,-1,-1,-1]' cmd: 'CMD_IDENTIFY_REQUEST', QOS: '101', responseId: '0', invokableId: '0', flags: 'JVMIDs Sent, TX Context Not Sent', abbrev offset: '594''&gt; </pre> <p>サーバの ExternalDNSName が指定されました。初期接続時に、クライアントはサーバインスタンスの ExternalDNSName へリクエストを送信しました。サーバインスタンスからの応答にある JVMID は、サーバの内部 IP アドレスと内部 DNS 名でした。クライアントからの以降のリクエストは認識されませんでした。</p> <p>これは SP03 で解決された WebLogic Server 7.0 の既知の問題です。クライアントがファイアウォールを介して t3 で接続する場合に問題が発生します。回避策は、サーバインスタンスの内部 DNS 名を外部 IP アドレスにマップするエントリを外部 DNS サーバに追加することです。</p>	7.0 SP03

## セキュリティに関する確認済みの問題

BEA のセキュリティ勧告については、dev2dev Web サイトにある BEA の Advisories & Notifications ページを参照してください。このページでは、セキュリティ関連のパッチをダウンロードしたり、新しいセキュリティ勧告の通知を受けるよう登録したりできます。Advisories & Notifications ページには、<http://dev2dev.bea.com/advisories> でアクセスできます。

BEA では、BEA 製品に関するセキュリティ問題をご報告いただくための電子メール アドレス ([security-report@bea.com](mailto:security-report@bea.com)) も用意しています。

以下の表に、このサービス パック リリースで確認されたセキュリティ関連の問題を示します。

変更要求番号	説明	問題が修正されたリリース
CR135489	カスタム セキュリティ プロバイダをコンフィグレーションする場合、プロバイダ用の MBean JAR ファイルを <b>WL_HOME\lib\mbeantypes</b> ディレクトリに置きます。このディレクトリに他のファイルを配置することはお勧めできません。WebLogic Server はこのディレクトリ内のすべてのファイルをロードしようとし、サーバの起動エラーを引き起こす可能性があります。	
CR051215	Solaris でのパスワードのプロンプトに問題があります。パスワードのプロンプトに対して [Ctrl] + [C] を押すと、画面への出力がすべて非表示テキストで行われるため、シェル ウィンドウを使用できなくなります。	
CR061030	WebLogic では EJB、サーブレット、およびリソース アダプタのデプロイメント記述子にある grant 文を解析しますが、システム変数を展開しません。 <b>回避策：</b> EJB、サーブレット、およびリソース アダプタのデプロイメント記述子にある <security-permission> タグでは変数を使用しないでください。	

変更要求番号	説明	問題が修正されたリリース
CR064593	<p>SSL の存続期間とキャッシュ サイズの制御に使用される以下のコマンドライン引数は、WebLogic Server 7.0 では無視されます。</p> <pre>-Dweblogic.security.SSL.sessionCache.size= sessioncachesize -Dweblogic.security.SSL.sessionCache.ttl= sessioncachetimetolive</pre>	7.0 サービス スバック 5
CR064837	<p>SSL を使用するクライアントは有効なライセンスが必須になりました。ライセンスの位置を指定する手順については、WebLogic Server のセキュリティに関するドキュメントを参照してください。</p> <p>また、SSL を使用するクライアントは、デフォルトで、接続先のサーバの証明書に対する信頼チェックを行う必要があります。信頼性のある認証局を指定する方法については、ドキュメントを参照してください。</p>	

変更要求番号	説明	問題が修正されたリリース
CR066782	<p>JCE プロバイダを静的にロードすると、または Sun による JSSE の実装を使用すると、SSL と WebLogic Server の間の対話で問題が発生する可能性があります。次のようなエラーが表示されません。</p> <pre data-bbox="323 545 919 597">&lt;&lt;java.security.NoSuchAlgorithm&gt; Exception: Algorithm XXXXXX not available&gt;&gt;</pre> <p>または</p> <pre data-bbox="323 647 1020 868">&lt;Unable to initialize the server: Fatal initialization exception Throwable: java.lang.ExceptionInInitializerError java.lang.ExceptionInInitializerError: java.lang.SecurityException: class "com.rsa.asn1.b"'s signer information does not match signer information of other classes in the same package</pre> <p>Sun による JSSE の実装を使用している場合は、WebLogic による SSL の実装との間で衝突が発生するため、このようなエラーが発生します。</p> <p><b>回避策:</b> 衝突を解決するには、JDK のディレクトリから <code>jsse.jar</code> (Sun の JDK 1.4 がデフォルトで提供するもの) を削除する必要があります。</p> <p>JCE プロバイダを使用している場合も、このエラーが発生することがあります。 <code>java.security</code> ファイルで静的にプロバイダを定義するのではなく動的にロードすると、問題が解決することがあります。次に、 <code>java.security</code> ポリシーの例を示します。</p> <pre data-bbox="323 1225 948 1277">security.provider.1=sun.security.provider.Sun security.provider.2=com.sun.rsajca.Provider</pre> <p><code>sunJCE</code> プロバイダは、それが静的に登録されていることがわかった場合は WebLogic Server によってアンロードされます。<code>sunJCE</code> プロバイダが必要な場合は、動的にロードしてください。</p>	
CR067087	<p>現時点では、 <code>security-permission-spec</code> 属性を <code>weblogic-application.xml</code> に追加できません。この属性を使用できるのは、 <code>weblogic-ejb-jar.xml</code> および <code>weblogic.xml</code> の中だけです。</p>	

変更要求番号	説明	問題が修正されたリリース
CR067621	<p>TLS_NULL_WITH_NULL_NULL 暗号スイートはサポートされておらず、使用しても無視されます。暗号スイートが使用されていないことを示す警告メッセージが、ログに記録されます。</p>	
CR067814	<p>LDAP ブラウザを使って WebLogic Server の組み込み LDAP サーバを表示する場合は、以下の情報を使用してください。</p> <pre>host:localhost</pre> <p>port: <i>WebLogic Server</i> のリスン ポート</p> <p>User DN:cn=Admin</p> <p>Password: [ドメイン   セキュリティ   組み込み LDAP] タブで、組み込み LDAP サーバに対して定義されたパスワード</p>	
CR072154	<p>MyListener クラスは、SSLSocketClient.java で実装されています。SSLClient の Package-Summary.html ファイルの「クラスの概要」の表では、このクラスは SSLSocketClient.MyListener と示されています。</p> <p>ファイルの場所は以下のとおりです。</p> <pre>SAMPLES_HOME\server\src\examples\security\sslclient</pre>	7.0 SP2

変更要求番号	説明	問題が修正されたリリース
CR073187	<p>TLS_RSA_WITH_DES_CBC_SHA 暗号スイートでは、SSL ライセンスでエクスポート レベルの暗号化だけが許可されている場合の問題が確認されています。この状況のため、このスイートを使用するようにコンフィグレーションされているサーバは、サーバの起動時に SSL の開始に失敗します。この障害の際に表示されるエラーの例を次に示します。</p> <pre data-bbox="323 609 1029 973"> &lt;Apr 2, 2002 7:32:24 AM PST&gt; &lt;Emergency&gt; &lt;WebLogicServer&gt; &lt;000209&gt;&lt;ListenThread.run() failed: java.lang.IllegalArgumentException: Export Restriction on suite: TLS_RSA_WITH_DES_CBC_SHA java.lang.IllegalArgumentException: Export Restriction on suite:TLS_RSA_WITH_DES_CBC_SHA at com.certicom.tls.TLSSystem.setEnabledCipherSuites ( Unknown Source) at javax.net.ssl.impl.SSLServerSocketImpl.setEnabledC ipherSuites(Unknown Source) at weblogic.t3.srvr.SSLListenThread.newServerSocket (S SLListenThread.java:667) at weblogic.t3.srvr.ListenThread.run </pre> <p><b>回避策:</b> 特定の暗号スイートを使用するようサーバがコンフィグレーションされている場合、WebLogic Server を正常に起動するには、暗号スイートのコンフィグレーション設定から次の項目を削除してください。</p> <pre data-bbox="323 1117 659 1168"> TLS_RSA_WITH_DES_CBC_SHA SSL_RSA_WITH_DES_CBC_SHA </pre>	
CR074716	<p>Administrator、Operator、Deployer、および Monitor の各ロールは、このリリースでは部分的にしか機能しません。これらのロールに追加されたユーザは、適切な管理 MBean の動作にアクセスできますが、Administration Console にはアクセスできません。</p> <p><b>回避策:</b> MBean の動作とコンソールの両方にユーザがアクセスできるようにするには、ユーザを Administrators、Operators、Deployers、および Monitors のグループに追加する必要があります。さらに、互換性モードで実行している場合は、システムユーザが Administrators グループに入っていることを確認する必要があります。</p>	



変更要求番号	説明	問題が修正されたリリース
CR075658	<p>互換性モードで動作している場合、またはファイル レルムをセキュリティ レルムとして使用している場合に、JDBC 接続プールのアクセス制御リスト (ACL) に関して確認済みの問題があります。fileRealm.properties ファイルに接続プールで any パーミッションを付与する ACL が含まれている場合は、shrink および reset パーミッションを明示的に付与する ACL があっても、その接続プールでは shrink アクションと reset アクションが許可されません。</p>	
CR081250	<p>LDAP Active Directory 認証プロバイダで無効なユーザアカウントが WebLogic Server に対して認証されます。</p> <p><b>回避策：</b></p> <p>UF_ACCOUNTDISABLE ビットセットを持たないアカウントだけを返すように、ユーザ フィルタを変更します。次に例を示します。</p> <p>元のフィルタ microsoft.user.filter=(&amp;(sAMAccountName=%u)(object class=user))</p> <p>変更後のテキスト microsoft.user.filter=(&amp;(sAMAccountName=%u)(object class=user)(!userAccountControl:1.2.840.113556.1.4.803:=2)))</p>	
CR082106	<p>ユーザ アカウントがロックされたためそのユーザ アカウントを削除して、同じ名前とパスワードを持つ別のユーザ アカウントを追加しても、UserLockout 属性がリセットされません。</p>	

## 確認済みの問題

変更要求番号	説明	問題が修正されたリリース
CR082926	<p>セキュリティ ネットワーク接続フィルタのサンプルにある SimpleConnectionFilter2 サンプルでは、WebLogic Server が起動しない可能性があります。この問題は Solaris プラットフォームでのみ発生が確認されていますが、他のプラットフォームでも起きる可能性があります。</p> <p>サンプル ファイルは次の場所にあります。</p> <pre>SAMPLES_HOME\server\src\examples\security\net</pre> <p>たとえば、ノード sunbird で SimpleConnectionFilter2 サンプルを実行しているときに、以下のフィルタ ルールが原因でサーバが起動しない可能性があります。</p> <pre>template2 10.61.4.13 7001 deny</pre> <p><b>回避策:</b> 以下のフィルタ ルールを使用すると、サーバは正常に起動できます。</p> <pre>sunbird 172.18.130.72 7001 deny</pre>	
CR083155	<p>weblogic.security.service.ResourceBase クラスの <code>getValues()</code> メソッドから返される文字列配列は、文字列配列の値型をサポートしていません。特に、JNDIResource の「path」キーと EJBResource の「signature」キーは常に null を返します。WebLogic 7.0 ( サービス パック 1 未適用 ) を使用している場合、配列の中に null 以外の値がない場合、NullPointerException が取得されます。</p>	
CR083242	<p>セキュリティ レルムでレルム アダプタ認証プロバイダをコンフィグレーションした後、WebLogic Server が起動できなくなります。</p> <p><b>回避策:</b> レルム アダプタ認証プロバイダをコンフィグレーションする前に、WebLogic Server を起動するディレクトリに空の <code>filerealm.properties</code> ファイルを作成します。レルム アダプタ認証プロバイダをコンフィグレーションするときに、[ 制御フラグ ] を Optional に設定します。WebLogic Server を再起動します。</p>	

変更要求番号	説明	問題が修正されたリリース
CR083273	<p>ドメインのユーザ名とパスワードの組み合わせが <code>weblogic/weblogic</code> と指定されている場合、Administration Console でユーザ名とパスワードをあらかじめ取得する際にセキュリティ リスクが生じます。</p> <p><b>回避策:</b> プロダクション環境では、<code>weblogic/weblogic</code> というユーザ名とパスワードの組み合わせを使用しないでください。</p>	
CR085089	<p>カスタム認可プロバイダを作成する場合、<code>ContextElement</code> の <code>getname()</code> メソッドで、同じ型の複数の EJB パラメータを指定すると、エラーが発生します。その型の最初のパラメータだけが取得されます。この確認済みの問題は WebLogic Server 7.0 SP02 で修正されます。</p>	

## ツールに関する確認済みの問題

変更要求番号	説明	問題が修正されたリリース
CR064893	<p>以下のデプロイメント記述子要素は、WebLogic Builder では変更できません。</p> <p>ejb-jar.xml の要素</p> <ul style="list-style-type: none"><li>• cascade-delete (EB/Advanced)</li><li>• weblogic ejb xml</li><li>• transaction-isolation</li><li>• description、display-name、large-icon、small-icon</li><li>• exclude-list</li><li>• unchecked</li><li>• use-caller-identity</li></ul> <p>weblogic-cmp.xml の要素</p> <ul style="list-style-type: none"><li>• enable-dynamic-queries (EB/Advanced)</li><li>• isolation-level (Methods/Transactions)</li><li>• multiple table mapping</li><li>• db-cascade-delete</li><li>• field-group、group-name (*)</li><li>• relationship-caching、caching-name、caching-element</li><li>• validate-db-schema-with</li><li>• database-type</li></ul>	
CR065250	<p>データベースに対するドライバクラスがクライアントにないと、WebLogic Builder はサーバ上のデータベースへのアクセスに失敗する場合があります。</p>	
CR066466	<p>WebLogic Builder では、操作をキャンセルすることはできません。</p>	
CR066668	<p>Linux の場合、WebLogic Builder のファイル ブラウザはカレントディレクトリではなくルートで開きます。</p>	
CR068152	<p>WebLogic Builder の EJB 要素では、Add、Edit、および Delete の各動作が遅くなる場合があります。</p>	8.1

変更要求番号	説明	問題が修正されたリリース
CR068165	WebLogic Builder において、[Resource Links] の [Add] ダイアログ、[WebLogic 設定]、および [J2EE Links] では、アプリケーション スコープのデータ リソースの選択が正しく表示されません。	
CR068620	WebLogic Builder では、表のカラムのサイズを正しく変更できません。	8.1
CR069568	WebLogic Builder では、EJB メソッド パーミッションの 1 つのメソッドに複数のロールをマップできません。	
CR071833、 CR072541	WebLogic Builder の RAR サポートには、デプロイメント記述子の生成が含まれていません。	
CR072700、 CR074591	WebLogic Builder のデプロイメント ダイアログ ボックスを使用してアプリケーションをアンデプロイする場合に、そのダイアログでアンデプロイメントが示されません。このダイアログでは、デプロイメントの失敗したアプリケーションがデプロイされたと報告される場合もあります。	
CR073715	WebLogic Builder では、複数の RDBMS XML ファイルを表示できないことがあります。	7.0 SP2
CR073975	WebLogic Builder では、メソッドまたはトランザクション属性を指定しなくてもトランザクションが追加されてしまいます。	7.0 SP3
CR074210	WebLogic Builder ではオプティミスティックな同時方式が機能しません。	8.1
CR074247	WebLogic Builder では、メソッド名が重複しているファインダを追加すると、新しいファインダが追加される代わりにオリジナルのファインダが更新される場合があります。	7.0 SP2
CR074322、 CR074856	WebLogic Builder では、編集処理が取り消せない場合があります。	8.1
CR074566	(include-update のように) WebLogic Builder の BeanGrid/Edit を通じて修正されたファインダまたは CMP 値は、個別のパネルで表されない場合があります。	8.1

## 確認済みの問題

変更要求番号	説明	問題が修正されたリリース
CR074622	WebLogic Builder のポップアップ ダイアログを使用して CMP フィールドを編集する際に、カラムの型を OracleCLOB に設定して、[CMP] ツリー ノードをクリックした場合に、そのカラムの型の変更が [CMP] パネルに表示されません。	
CR074634、 CR074955、 CR074972	WebLogic Builder の関連ウィザードを使用して既に複数の関連がコンフィグレーションされている CMP の関連を編集すると例外が送出されます。	7.0 SP2
CR074639	WebLogic Builder の編集パネルを使用してファインダ メソッドの名前を編集した場合に、その変更は XML に伝播されません。[ファインダ] ポップアップ ダイアログを使用して編集した場合、その変更は XML にも編集パネルにも伝播されません。	7.0 SP2
CR074728	WebLogic Builder では、メソッドをローカルメソッドに変更することができません。	
CR074754	一部のトランザクション属性を編集すると、WebLogic Builder で記述子が重複して作成される場合があります。	
CR074775	WebLogic Builder では enable-call-by-reference 要素を明示的に設定することをお勧めします。	
CR074961	WebLogic Builder を使用して web.xml 記述子を生成すると、7.0 より前のセキュリティ プロトコルに関連するセキュリティ ロール「guest」が作成されます。	
CR075005	WebLogic Builder でコンポーネントを開くときに無効なドライブを指定すると (テキストボックスで無効なドライブを入力し、[Enter] を押して [開く] を選択する)、NullPointerException が発生します。	7.0SP2
CR075006	場合によっては、WebLogic Builder で 1 対多の関連が正しく作成されません。	7.0 SP2
CR075129	ステートレス セッション Bean をステートフルに切り替えた場合に、WebLogic Builder ではそのモジュールが変更されていることが示されない場合があります。	

変更要求番号	説明	問題が修正されたリリース
CR075228	トランザクションまたはパーミッション メソッドを削除すると、WebLogic Builder ではアサーション エラーが送出される場合があります。	8.1
CR075586	WebLogic Server サンプル サーバまたはサンプルで WebLogic Builder を使用する場合は、WebLogic Builder を起動する前に setExamplesEnv (%SAMPLES_HOME%\server\config)\examples にあります) を実行し、PointBase データベース サーバを実行するために必要なファイルをクラスパスに追加する必要があります。これを行わないと、データベースのエラー メッセージが表示されます。	
CR075753	記述子の一部の要素は WebLogic Builder の検証をパスしても、ejbc によってエラーと判定されることがあります。	
CR075924	WebLogic Builder では、EAR でネストされた WAR ファイルの記述子ファイルが正しく生成されない場合があります。	8.1
CR075936	WebLogic Builder では、<estimated-bean-size> 要素で無効な値が生成される場合があります。	8.1

## WebLogic Tuxedo コネクタに関する確認済みの問題

変更要求番号	説明	問題が修正されたリリース
CR067275	RMI over IIOP を使用するクラスタの WebLogic Tuxedo Connector では、ロード バランシングとフェイルオーバーはサポートされていません。	

変更要求番号	説明	問題が修正されたリリース
CR079630	<p>           関連 ID を設定する場合、<code>setJMSCorrelationID(String)</code> メソッドの使用を考えるユーザが多いと思われます。このメソッドでは 32 文字の文字列を使用し、それを 64 バイトの配列に格納します。JMS では文字列は UTF-16BE として格納されます。tBridge が Tuxedo から受け取る関連 ID も 32 文字ですが、こちらは 32 バイトです。tBridge では <code>setJMSCorrelationIDAsBytes(byte[])</code> を使用して、受信側の JMS キュー用のメッセージが作成されます。この 2 つの文字列は ASCII では同じ文字列になることもありますが、長さが異なるために同じ文字列とは見なされません。         </p> <p> <b>回避策:</b> 関連 ID を Tuxedo から JMS キュー内に返された ID と比較する必要がある場合には、バイト配列を作成して関連 ID の 16 進数値を格納します。その後 <code>setJMSCorrelationIDAsBytes()</code> および <code>getJMSCorrelationIDAsBytes()</code> メソッドを使用して ID を作成し、Tuxedo から JMS キュー内に返された ID と比較します。たとえば関連 ID の文字列が "1234567890ABCDEFGHIJKLMNQRSTU" の場合、次のようになります。         </p> <pre> private byte[] coridbyte={0x31,0x32,0x33,0x34,0x35,0x36,0x37,                           0x38,0x39,0x30,0x41,0x42,0x43,0x44,                           0x45,0x46,0x47,0x48,0x49,0x4a,0x4b,                           0x4c,0x4d,0x4e,0x4f,0x50,0x51,0x52,                           0x53,0x54,0x55,0x56}; msg.setJMSCorrelationIDAsBytes(coridbyte); corIDasBytes = msg.getJMSCorrelationIDAsBytes();  corIDasBytes に Tuxedo から返された関連 ID と正しく比較できる値が格納されます。 </pre>	



## Web サービスに関する確認済みの問題

変更要求番号	説明	問題が修正されたリリース
CR067190	WebLogic Web サービスでは、complexTypes および同じ名前の複数の要素がある XML スキーマはサポートされていません。	
CR072890	コマンドライン パージョンの servicegen Ant タスクでは、Ant タスク属性に対応する以下のオプションがサポートされていません。  -overwrite -contextURI -includeEJBs -excludeEJBs -generateTypes  コマンドライン パージョンの clientgen Ant タスクでは、Ant タスク属性に対応する以下のオプションがサポートされていません。  -autotype -overwrite	8.1
CR073661	無効な web-services.xml ファイルを使用して不正確に記述された WebLogic Web サービスは WebLogic Server にデプロイはされませんが、そのサービスを呼び出そうとするクライアント アプリケーションからアクセスできません。	
CR073899	WebLogic Web サービスは、SOAP 障害の生成時に余計な非 CDATA 文字を追加します。	7.0 SP2

変更要求番号	説明	問題が修正されたリリース
CR083493	<p>Web サービス サンプルのパッケージに含まれる messages サンプルと multicomponent サンプルには、ダウンロード可能な client.jar ファイルを生成するために必要なコードがありません。これは、用意されているクライアントで動作するサンプルの機能には影響しません。</p>	
CR101160	<p>サービス固有の例外の複合データ型が、次のように、JAX-RPC (Java API for XML-Based RPC) 1.0 の仕様に厳密に準拠して処理されません。</p> <ul style="list-style-type: none"> <li> <p>• <b>セクション 4.3.6 WSDL Fault</b>、サービス固有の例外：メッセージ部分が <code>xsd:complexType</code> である <code>wsdl:fault</code> 要素の含まれる WSDL から <code>clientgen Ant</code> タスクによって JAX-RPC スタブが生成される場合、生成されたスタブは <code>xsd:complexType</code> を間違って Java の単純型にマッピングします。つまり、クライアントアプリケーションがそのスタブを使用して Web サービスを呼び出し、サービスから (<code>javax.xml.rpc.SOAPFaultException</code> ではなく) サービス固有の例外が送出されたときに、クライアントアプリケーションはその例外から完全な複合データにアクセスできないということです。</p> <p>この問題の回避策はありません。</p> </li> <li> <p>• <b>セクション 5.5.5 Methods</b>、例外、4 番目の項目：WebLogic Web サービス ランタイムは、サービス固有の Java 例外クラスの複数のフィールドを <code>xsd:complexType</code> ではなく間違って単純 XML データ型にマッピングします。</p> <p><b>回避策：</b> WebLogic Web サービスを実装する際に、サービス固有の例外ではなく常に <code>javax.xml.rpc.SOAPFaultException</code> を送出します。</p> </li> </ul>	

## XML に関する確認済みの問題

変更要求番号	説明	問題が修正された リリース
CR070547	Solaris で XML ストリーミング API を使用する場合、一部の文字参照が NT の場合と同じ文字に解決されません。	

## WebLogic Server 7.0 の確認済みの問題

この節では、特定のサービス パックと関連付けられていない WebLogic Server 7.0 の問題を示します。

### コネクタに関する確認済みの問題

変更要求番号	説明	問題が修正されたりリリース
CR098342	<p>WebLogic Server 7.0 の JCA 実装は、IBM CICS リソース アダプタと互換性がありません。</p> <p><code>javax.resource.spi.ConnectionManager</code> インタフェースの BEA 実装は、<code>javax.resource.spi.ManagedConnection</code> インタフェースのリソース アダプタ実装の <code>getConnection()</code> メソッドによって返されるオブジェクトを直接には返しません。</p> <p>代わりに、<code>java.lang.reflect.Proxy</code> クラスで動的に生成されるクラスのインスタンスが返されます。</p>	

### コア サーバに関する確認済みの問題

変更要求番号	説明	問題が修正されたりリリース
CR106616	<p>Linux AS2.1 および HP-UX で動作する Netscape パージョン 4.79 には問題があります。</p> <p>WebLogic Server Administration Console でのいくつかの操作の後、ブラウザはバス エラー (コア ダンプの場合もある) で停止します。</p>	

## EJB に関する確認済みの問題

変更要求番号	説明	問題が修正されたりリリース
CR061938	<p>EJB QL クエリは、SQL クロス積の結果として誤った重複を返す可能性があります。これは次のような状況で発生する可能性があります。</p> <ul style="list-style-type: none"> <li>• EJB-QL クエリに関係をナビゲートするパス式が含まれる場合。生成される SQL SELECT 句には複数のテーブルが含まれます。</li> <li>• WHERE 句に関係をナビゲートする OR オペランドが含まれるが、OR オペランド内のすべてのパス式が、クエリで生成される SQL SELECT 句のすべてのテーブルにマップされるわけではない場合。この場合、その OR オペランドの結果内にクロス積が現れます。</li> </ul> <p>この問題を次の例に示します。 EJB QL:</p> <pre> SELECT OBJECT(c) FROM CustomerBean AS c, IN(c.accounts)accts WHERE c.name = '100' OR c.accts.bal = 300 DATA: customer '100' exists but has no accounts EXPECTED RESULT: customer '100' from clause #1 ACTUAL RESULT: customer '100' X number of accts SQL: DROP TABLE thorick_customers; CREATE TABLE thorick_customers (   cust_name VARCHAR(10),   cust_interests VARCHAR(10),   cust_rating INTEGER, acct_id INTEGER,   PRIMARY KEY (cust_name)); DROP TABLE thorick_accounts; CREATE TABLE thorick_accounts (acct_id INTEGER, bal FLOAT, PRIMARY KEY (acct_id)); INSERT INTO thorick_accounts VALUES (100, 100.0); INSERT INTO thorick_accounts VALUES (200, 200.0); INSERT INTO thorick_accounts VALUES (300, 300.0); INSERT INTO thorick_accounts VALUES (400, 400.0); INSERT INTO thorick_accounts VALUES (500, 500.0); INSERT INTO thorick_customers VALUES('100', 'jazz', 2, null); INSERT INTO thorick_customers VALUES('900', 'punk', 3, 400); SELECT WL0.cust_name, WL0.cust_interests, WL0.acct_id FROM thorick_Customers WL0, thorick_Accounts WL1 WHERE WL0.cust_name = '100' OR ( w11.bal = 300 AND w10.acct_id=w11.acct_id ); CUST_NAME    CUST_INTER    ACCT_ID ----- 100           jazz 100           jazz 100           jazz 100           jazz 100           jazz </pre>	

## JDBC に関する確認済みの問題

変更要求番号	説明	問題が修正 されたリ リース
CR101419	クラスタの DataSources および TxDataSources のフェイルオーバがこのリリースでは機能しません。	

## RMI に関する確認済みの問題

変更要求番号	説明	問題が修正されたリリース
CR099693	<p>7.0 (G.A.、SP01、および SP02) では、クライアントが t3 プロトコルを使用してファイアウォールから WebLogic Server にアクセスすると、サーバで次のエラーが発生します。</p> <pre>####&lt;Feb 17, 2003 10:15:59 AM EST&gt; &lt;Error&gt; &lt;RJVM&gt; &lt;ewibeasbx01.merck.com&gt; &lt;edmServer&gt; &lt;ExecuteThread: '14' for queue: 'default'&gt; &lt;kernel identity&gt; &lt;&gt; &lt;000506&gt; &lt;Closing: weblogic.rjvm.t3.T3JVMConnection@23730e because of Server expected to route a message received over an uninitialized connection: 'JVMMMessage from: '3684268472702687546S:172.16.2.2:[7001,7001,7002,700 2,7001,7002,-1]:edm:edmServer' to: 'OS:172.16.1.250:[7000,-1,-1,-1,-1,-1,-1]' cmd: 'CMD_IDENTIFY_REQUEST', QOS: '101', responseId: '0', invokableId: '0', flags: 'JVMIDs Sent, TX Context Not Sent', abbrev offset: '594''&gt;</pre> <p>サーバの ExternalDNSName が指定されました。初期接続時に、クライアントはサーバインスタンスの ExternalDNSName へリクエストを送信しました。サーバインスタンスからの応答にある JVMID は、サーバの内部 IP アドレスと内部 DNS 名を必要としていました。クライアントからの以降のリクエストは認識されませんでした。</p> <p>これは SP03 で解決された WebLogic Server 7.0 の既知の問題です。クライアントがファイアウォールを介して t3 で接続する場合に問題が発生します。</p> <p><b>回避策:</b> サーバインスタンスの内部 DNS 名を外部 IP アドレスにマップするエントリを外部 DNS サーバに追加します。</p>	7.0 SP03

## ディレクトリ名のスペース

WebLogic Server は、名前またはアドレスにスペースが含まれるディレクトリからは起動しません。たとえば、次のディレクトリ名の場合、起動時に失敗します。

```
D:\bea new\weblogic700 space\
```

```
D:\bea new\
```

WebLogic Server 7.0 ではディレクトリ名にスペースを含めないでください。この問題は変更要求 CR084568 で示されています。

## EJB の値渡しに関して J2EE に準拠していない点

WebLogic Server の EJB デプロイメント記述子の多くには、パフォーマンスのために最適化されたデフォルト値があります。これらのデフォルト値には、EJB 仕様に準拠していないものもあります。WebLogic Server を EJB 仕様に準拠させる場合は、以下のプロパティを設定する必要があります。

weblogic-ejb-jar.xml で、enable-call-by-reference を False に設定する。

weblogic-cmp-jar.xml で、include-updates を True に設定する。

weblogic-cmp-jar.xml で、check-exists-on-method を True に設定する。

## HTML ページと Netscape

WebLogic Server の起動時に Netscape ブラウザが既に動作している場合は、一部の HTML ページが表示されません。たとえば、Netscape が動作しており、Microsoft Internet Explorer がない状態で、[スタート]メニューから [About WebLogic Server] ページを選択した場合、画面は点滅しますが、ページは表示されません。



## クラスタ、DNS、および NT でのマルチホーム

クラスタ環境の Windows NT でマルチホームを使用する場合は、名前の問題に注意する必要があります。たとえば、マルチホームの Windows NT マシンでクラスタが動作しており、クラスタ内の 1 つのサーバがマシン名と同じ DNS 名にバインドされている場合は名前の衝突が発生する場合があります。URL でその DNS 名を使用してサーバにアクセスしようとする、Windows NT ではその DNS 名がマルチホーム Windows NT マシンのいずれかの IP アドレスに変換される場合があります。この場合、要求が間違ったアドレスに送られる場合があります。DNS 名は、マシン名と一致しないようにしてください。

## WebLogic キーストア プロバイダのサポート

WebLogic キーストア プロバイダでは Java 2 Enterprise Edition (JKS) キーストアのみを使用できます。パスワードで保護される JKS キーストアを、WebLogic キーストア プロバイダで使用できるようになりました。

**注意：** WebLogic キーストア プロバイダはこのリリースの WebLogic Server 7.0 で非推奨となりました。

## マルチサーバ環境の XA 非対応 JDBC ドライバに関するトランザクションの制限事項

TxDataSource の [ 非 XA ドライバ用に 2 フェーズ コミットをエミュレート ] (または `EnableTwoPhaseCommit=true`) オプションは、単一の XA 非対応リソースが分散トランザクションに参加できるようにするために設計されています。このような TxDataSource は 1 つの分散トランザクションで 1 つだけ使用できます。

マルチサーバ環境 (クラスタを含む) では、複数のサーバ上の XA 非対応リソースで TxDataSource をコンフィグレーションする場合、アプリケーションはトランザクションコーディネータとして動作するサーバ上のリソースにアクセスする必要があります。そうしない場合、コミット処理中に、ヒューリスティックな終了を引き起こすエラーが発生します。

クラスタ化された環境では、複数サーバ間のロード バランシングが原因でこのエラーが発生する大きな危険性があります。

## 確認済みの問題

### 回避策：

分散トランザクションで JDBC リソースにアクセスするには、XA をサポートする JDBC ドライバを使用します。

XA 非対応 JDBC ドライバで TxDataSource をコンフィグレーションする方法については、『管理者ガイド』の「[分散トランザクション用の XA 非対応 JDBC ドライバのコンフィグレーション](#)」を参照してください。

# サービス パック 7 の解決済みの問題

サービス パックは累積的なものです。サービス パック 7 には、それ以前にリリースされた WebLogic Server 7.0 のサービス パックで行われたすべての修正が含まれています。以前のサービス パックで行われた修正の説明については、[第 4 章「サービス パック 1 から 6 で解決済みの問題」](#)を参照してください。

以下の節では、WebLogic Server 7.0 サービス パック 7 で解決された問題について説明します。

- [3-2 ページの「Administration Console」](#)
- [3-4 ページの「クラスタ」](#)
- [3-4 ページの「コア サーバ」](#)
- [3-6 ページの「デプロイメント」](#)
- [3-6 ページの「EJB」](#)
- [3-7 ページの「JDBC」](#)
- [3-8 ページの「JMS」](#)
- [3-10 ページの「JSP」](#)
- [3-11 ページの「JTA」](#)
- [3-12 ページの「JVM」](#)
- [3-12 ページの「ノード マネージャ」](#)

## サービス パック 7 の解決済みの問題

- [3-13 ページの「操作と管理」](#)
- [3-14 ページの「プラグイン」](#)
- [3-18 ページの「RMI」](#)
- [3-19 ページの「セキュリティ」](#)
- [3-21 ページの「サーブレット」](#)
- [3-24 ページの「Simple Network Management Protocol \(SNMP\)」](#)
- [3-24 ページの「Web サービス」](#)
- [3-25 ページの「XML」](#)

# Administration Console

変更要求番号	説明
CR082722	デプロイメント記述子エディタが <code>ejb-jar.xml</code> ファイルの <code>&lt;unchecked/&gt;</code> メソッドパーミッションを適切に処理していませんでした。 この問題は解決済みです。
CR108255	英語版の WebLogic Server Administration Console ([グループ] タブ) で日本語のユーザ名を作成しようとすると、 <code>weblogic.management.utils.NotFoundException</code> が送出されました。 この問題は解決されています。
CR109110	このリリースより前では、WebLogic Server Administration Console を使用して既に削除済みの WTC オブジェクトを削除すると、 <code>InstanceNotFoundException</code> が送出されて、Console にスタックトレースが表示されました。 スタックトレースの代わりに、意味のあるエラーメッセージが表示されるようになりました。
CR122389	WebLogic Server はノードの名前に関して、大文字 / 小文字の区別を無視していました。その結果、同じ名前で大文字 / 小文字が異なる複数のノードがあっても、WebLogic Server は JNDI ツリーにノードを 1 つだけ表示していました。 この問題は解決済みです。

変更要求番号	説明
CR209949	<p><code>alternateTypesDirectory</code> プロパティの値としてカンマで区切られた複数のディレクトリを指定すると、WebLogic Server はこれらのディレクトリに存在する有効なアーカイブをロードしませんでした。プロパティに1つのディレクトリが指定された場合にのみ、アーカイブがロードされました。</p> <p>この問題は解決済みです。</p>
CR216670	<p>認証の試行回数を超えると、ロックアウトされたことを示すメッセージがドメインでログに記録されました。</p> <p>現在は、Administration Console にもロックアウト メッセージが表示されます。</p>
CR216851	<p>[ ポリシーの定義 ] および [ ロールを定義 ] ポップアップ メニュー オプションは JMS ブリッジ送り先では使用できなくなりました。</p>
CR238054	<p>EJB に関連するモニタ統計が WebLogic Server Administration Console で表示されませんでした。</p> <p>この問題は解決済みです。</p>
CR239018	<p>JMS 送り先に対してセキュリティ ポリシーをコンフィグレーションすると <code>CreateException</code> が発生しました。</p> <p>この問題は解決済みです。</p>
CR241554	<p>管理対象サーバ独立 (MSI) モードで管理対象サーバを起動するために [ 管理対象サーバ独立のファイル レプリケーションを有効化 ] チェック ボックスを選択した場合に、管理対象サーバのルート ディレクトリで、コンフィグレーション ファイル (<code>config.xml</code>) が、<code>config.xml</code> ではなく、誤って <code>msi-config.xml</code> としてコピーされました。サーバ起動時に <code>config.xml</code> ファイルが見つからないため、管理対象サーバは MSI モードで起動されませんでした。</p> <p>この問題は解決済みです。</p>
CR241861	<p>Web アプリケーションに関連するモニタ統計が WebLogic Server Administration Console で表示されませんでした。</p> <p>この問題は解決済みです。</p>

## サービス パック 7 の解決済みの問題

変更要求番号	説明
CR253611	ユーザ パスワードを変更した後で新しいパスワードをコンフィグレーションしようとする、WebLogic Server Administration Console で <code>NullPointerException</code> が送出されました。 この問題は解決済みです。
CR255254	WebServer MBean に関連付けられた <code>AuthCookieEnabled</code> 属性を WebLogic Server Administration Console で設定できませんでした。 この問題は解決済みです。

## クラスタ

変更要求番号	説明
CR241267	クラスタ内のあるサーバが使用できなくなった場合、無効なサーバルックアップが行われる可能性があります。これは、内部ハッシュ テーブルが適切に更新されないことが原因でした。 この問題は解決済みです。

## コア サーバ

変更要求番号	説明
CR132806、CR126374	現在のスレッドの優先順位を調べるための <code>Thread.currentThread().getPriority()</code> メソッドが、コンフィグレーション時に設定された実際の優先順位ではなく、デフォルト値を返しました。 この問題は解決済みです。
CR196369	以下の URL のセキュリティ勧告を参照してください。 <a href="http://dev2dev.bea.com/pub/advisory/138">http://dev2dev.bea.com/pub/advisory/138</a>

変更要求番号	説明
CR200568、 CR128417、 CR134918、CR172274	<p>サーバが TCP_NODELAY オプションを設定できるようにクライアントがソケットのリセットを行ったときに、サーバ側のソケットが IDLE 状態のままであることがありました。このため、それらの IDLE ソケットは適切にクリーンアップされていませんでした。</p> <p>現在は、ソケットの TCP_NODELAY オプションの設定時に例外が発生すると、サーバ側のソケットはクリーンアップされます。</p>
CR210488	<p>AIX プラットフォームで IIOP を使用する EJB をコンパイルする際に例外が発生しました。</p> <p>この問題は解決済みです。</p>
CR211299	<p>以下の URL のセキュリティ勧告を参照してください。</p> <p><a href="http://dev2dev.bea.com/pub/advisory/156">http://dev2dev.bea.com/pub/advisory/156</a></p>
CR218491、CR201647	<p>BasicReplicaList クラスの同期化の問題が原因で、EJB 層の再起動後に OptionalDataException をネストした UnmarshalException が送出されました。</p> <p>この問題は解決済みです。</p>
CR236296	<p>call-by-reference-enabled プロパティが False に設定されていて、JSP が EJB を呼び出した場合に、メモリ リークが発生しました。</p> <p>この問題は解決済みです。</p>
CR239324、CR239819	<p>ソケットを登録解除してから登録した場合に、この 2 つの呼び出し間にタイムアウトのタイマーが実行されなかった場合、データ構造は同期化されず、新しく登録されたソケットはタイムアウトされなくなりました。</p> <p>この問題は解決済みです。</p>
CR243895	<p>リモート EJB の呼び出し中に、HashMap が 2 つのスレッドによって同時に変更されたため、ConcurrentModificationException が発生しました。</p> <p>HashMap へのアクセスは同期化されるようになりました。</p>
CR248250	<p>リモートオブジェクトがリモートからもローカル(サーバ内)からもアクセスされる場合に、ローカルアクセスが原因でデータの破損が起き、その結果いずれの場合もロード バランシングが失敗しました。</p> <p>この問題は解決済みです。</p>

## デプロイメント

変更要求番号	説明
CR215116	<p>デフォルトでは、自動デプロイメントが有効になっている場合、サーバのアプリケーション ディレクトリから削除されるときに、アプリケーションは自動的にアンデプロイされます。ただし、(削除操作の後で)このディレクトリにアプリケーションをコピーしようとする、<code>UndeclaredThrowableException</code> が送出されました。</p> <p>この問題は解決済みです。</p>

## EJB

変更要求番号	説明
CR122378、 CR200204	<p>ローカル インタフェースとリモート インタフェースの両方を持つ EJB がデプロイされていて、同じアプリケーション (EAR) にデプロイされている Web クライアントからアクセスされる場合、<code>LocalHome</code> の <code>create</code> が呼び出されると <code>ClassCastException</code> が送出されました。</p> <p>この問題は解決済みです。</p>
CR187691	<p>エンティティ Bean で <code>concurrency-strategy</code> プロパティが <code>Exclusive</code> に、<code>cache-between-transactions</code> プロパティが <code>True</code> に設定されている場合、<code>findByPrimaryKey</code> メソッドが呼び出されると <code>CacheFullException</code> が発生する可能性があります。</p> <p>この問題は解決済みです。</p>
CR221686	<p>EAR ファイル内の JSP と 100 以上の EJB にアクセスする場合、(分岐プロセスに関する) コマンドの長さの制限があるため、コンテナによるコンパイルが失敗しました。</p> <p>インラインのコンパイルにはこの制限がないため、現在は、デフォルトで、JSP と EJB はインラインでコンパイルされます。</p>
CR229290	<p>EJB のコンパイル中に、WebLogic Server は <code>config.xml</code> ファイルの <code>extra-rmic-options</code> 要素で指定されたオプションを使用しませんでした。</p> <p>この問題は解決済みです。</p>



変更要求番号	説明
CR240649、 CR174593	ejbStore メソッドでエンティティ Bean のファインダが呼び出されて、このメソッドのトランザクションに新しい Bean が登録された場合、または、同じタイプのエンティティ Bean がこのメソッドで変更された場合、 ConcurrentModificationException が送出されました。 この問題は解決済みです。
CR258768	十分な権限を持たないユーザがステートフル セッション Bean の remove () メソッドを呼び出した場合に、AssertionError が発生しました。 正しいメッセージの AccessException が送出されるようになりました。
CR261092	JMS プロバイダとして MQSeries を使用する場合に、WebLogic Server は XAER_PROTO を送出しなくなりました。

## JDBC

変更要求番号	説明
CR108103	負荷がかかっている場合に、JDBC 接続プールの待機数に関する統計で、正しい値が表示されないことがありました。 この問題は解決済みです。
CR214723	文字セットが JIS16EUC に設定されている Oracle データベースと 10g ドライバを使用する Portal アプリケーションがデプロイされている場合に、 SQLException が発生しました。 この問題は解決済みです。
CR241211	TxDataSource よりも前に作成された XAConnection が Connection.reserve によって呼び出されたときに、その XAConnection に null でないデータ ソースが割り当てられていなかったため、起動時に NullPointerException が発生しました。 この問題は解決済みです。
CR242502	回復が原因で送出された XAER_NOTA 例外は、JDBC デバッグ フラグが有効になっている場合にのみ、サーバ ログに出力されるようになりました。

## サービス パック 7 の解決済みの問題

変更要求番号	説明
CR249009、 CR243824	登録に失敗したとき、接続がプールに解放される前に、トランザクションと接続の関連付けを解除する <code>XA.end</code> が呼び出されませんでした。 この問題は解決済みです。
CR256316	サスペンドされたトランザクションがタイムアウトして再開された場合に、 <code>InvalidTransactionException</code> が発生しました。 この問題は解決済みです。
CR263367	製品パッケージ内の Sybase JConnect ドライバは JConnect 6.05 に更新されました。

## JMS

変更要求番号	説明
CR172511、 CR174691	最初にすべての <code>JMSConsumer</code> クライアントを閉じる前に <code>JMSConnection</code> セッションを閉じると、メモリ リークが発生する可能性があります。 この問題は解決済みです。
CR176366	<code>getObject()</code> メソッドが <code>onMessage()</code> API の <code>Boolean.TYPE</code> で設定された場合に、次の <code>JMSException</code> が送出されていました。 <pre>javax.ejb.EJBException: nested exception is:weblogic.jms.common.JMSException: Error deserializing object weblogic.jms.common.JMSException: Error deserializing object</pre> この問題は解決済みです。
CR177558	<code>NO_ACKNOWLEDGE</code> 配信モードの場合に、サーバサイドのメモリ リークは発生しなくなりました。

変更要求番号	説明
CR178775	<p>JMS クライアントが WebLogic Server に対して JMS 接続を繰り返し開いて閉じた場合、JMSConnection と DispatcherWrapper などの関連するオブジェクトはクライアントサイドとサーバサイドの両方で解放されませんでした。このため、OutOfMemoryError が発生しました。このエラーは、クライアントが JMS 接続を 1 つ開いて保持している間に、他の JMS 接続を開いて閉じると発生します。</p> <p>この問題は解決済みです。</p>
CR187610	<p>JMS ブリッジは、ブリッジのディスパッチ リクエストに対して、メッセージングブリッジのスレッド プールを使用していませんでした。</p> <p>現在は、メッセージングブリッジのスレッド プールがコンフィグレーションされている場合、JMS ブリッジはそれを使用します。メッセージングブリッジのスレッド プールにコンフィグレーションされている使用可能なスレッド数が十分でない場合、ブリッジはデフォルトの実行スレッド プールを使用します。</p> <p>ブリッジは、コンフィグレーションされているメッセージングブリッジのスレッド プール サイズが不十分な場合、警告メッセージをログに記録し、代わりにデフォルト実行プールからスレッドを取得します。</p>
CR216063、 CR256579、 CR234758	<p>MessagingBridgeRuntime mbean getState() は状態の情報を適切に更新するようになりました。</p>
CR226919	<p>一時的な送り先の delete() の失敗によって発生した JMS メモリ リークは、発生しなくなりました。</p>
CR229495	<p>バックエンドの JMS サーバが停止した場合、確認応答の完了の待機中にクライアントがハングすることはなくなりました。</p>
CR235404	<p>クライアント接続を閉じる前にプロデューサが明示的に閉じられていない場合、メモリ リークが発生しました。</p> <p>この問題は修正されています。</p>
CR239642	<p>BESession.acknowledge() で NullPointerException は発生しなくなりました。</p>

## JSP

変更要求番号	説明
CR217352	<p>無効な URL を使用して JSP にアクセスすると、不適切な JSP マッピングが原因で 404 エラーが送出されました。</p> <p>コード内で <code>exactMapping</code> のデフォルト値を <code>true</code> に変更することにより、この JSP マッピングの問題は解決されました。必要な場合は、<code>weblogic.xml</code> ファイルでこの値を <code>false</code> に設定できます。</p>
CR219805	<p>以下のように、Java のコメントが複数のスクリプトレットにまたがる場合に、JSP のコンパイルは失敗なくなりました。</p> <pre>&lt;%/ *&gt; [HTML code] &lt;%/ */ {   /*comment*/ } %&gt;</pre>
CR242858	<p><code>Weblogic-vtags.jar</code> で使用可能な <code>wl:summary</code> タグが JSP によって使用された場合に、クエリ文字列が次のページに渡されませんでした。</p> <p>省略可能なブール型の <code>AddQueryParams</code> フィールドを <code>wl:summary</code> タグに追加することで、この問題を解決しました。<code>AddQueryParams</code> フィールドを <code>True</code> に設定した場合、クエリ文字列は URI に付加されます。</p>
CR254360	<p><code>weblogic.jspc</code> が使用されたとき、このアプリケーション コードでコンテキスト クラスローダを使用した場合に、<code>WEB-INF\lib</code> ディレクトリのクラスに対して <code>ClassNotFoundException</code> が送出されました。</p> <p>このような状況で <code>ClassNotFoundException</code> は送出されなくなりました。</p>

# JTA

変更要求番号	説明
CR048979	<p>JTS 接続の複数の登録において、リソースの初期化が同時に試行される場合に、競合状況は発生しなくなりました。</p> <p>その結果、JST 接続を取得するときに、WebLogic Server は <code>javax.management.InstanceAlreadyExistsException</code> を送出しなくなりました。</p>
CR213034	<p>トランザクション メッセージの移行後に、JMS メッセージの再配信が行われませんでした。</p> <p>この問題は解決済みです。</p>
CR215559	<p>グローバル トランザクションに関与するリソースがヒューリスティック障害の例外を送出した場合に、すべてのサブコーディネータとリソースがコミットに応答しなかったにも関わらず、トランザクションは完了と宣言されました。</p> <p>理想としては、コミット処理中にいずれかのサブコーディネータ サーバが停止した場合は、すべてのリソースとサブコーディネータから応答があるまで、コーディネータはコミットすべきではありません。WebLogic Server ではすべてのサブコーディネータがコミット呼び出しを完了できるようになっていなかったため、トランザクションがコミットと宣言されていても、そのトランザクションは保留中のままでした。</p> <p>この問題は解決済みです。</p>
CR249090	<p>トランザクションが準備フェーズに入り、タイムアウトに近づくと、<code>Object.wait()</code> メソッドに負の値が渡されなくなる可能性があります。その結果 <code>wait</code> メソッドによって <code>IllegalArgumentException</code> が送出され、これが <code>SystemException</code> 例外にラップされてコミットの呼び出し側に送出されます。</p> <p>この問題は解決済みです。</p>

## JVM

---

変更要求番号	説明
CR218538、 CR241553	RJVM が停止されるときに JAVA LEVEL DEADLOCK は発生しなくなりました。
CR252304	MaxMessageSizeExceededException の発生後にリモート サーバの RMI サービスがサーブレットから呼び出されると、一部のスレッドは RMI 応答を待機している状態で無期限に停止しました。 この問題は解決済みです。

---

## ノード マネージャ

---

変更要求番号	説明
CR111339、 CR237367	<p>低速のハードウェアでノード マネージャを使用して複数の WebLogic Server インスタンスを起動しようとする、次のエラーが生じていました。</p> <pre>&lt;Jun 25, 2003 5:16:15 PM CDT&gt; &lt;Error&gt; &lt;NodeManager@192.168.190.4:5555&gt; &lt;_COMMAND_EXCEPTION_Request: failed to execute command 'online' on server 'ifgw1catapp4 - reason: '[JavaProcessControlOnline: Could not get valid pid for WebLogic process.]'&gt;</pre> <p>この問題は、ファイルに書き込む機会がないうちにノード マネージャが PID ファイルを検索していたことが原因でした。</p> <p>新しいプロパティを使用すると、ユーザはノード マネージャがファイルから PID を読み込む再試行回数を指定できます。各再試行は 2 秒間待機します。導入されたプロパティは PIDFileReadRetryCount であり、システム プロパティとして、または nodemanager.properties ファイルで指定できます。</p> <p>PIDFileReadRetryCount の値がゼロより大きい場合、ノード マネージャはファイルから PID を読み込むのをあきらめるまで PIDFileReadRetryCount の回数だけ試行します。PIDFileReadRetryCount のデフォルト値はゼロなので、デフォルト動作に変更はありません。</p>

---

## 操作と管理

変更要求番号	説明
CR179126	JMS サーバをホストする管理サーバで <code>weblogic.Admin</code> ユーティリティを使用して <code>addTarget</code> メソッドが呼び出されたときに、JMS サーバがまだ対象指定されていない場合、このユーティリティは <code>InvalidAttributeValueException</code> を送出しました。 この問題は解決済みです。
CR189763	ステージングされたアプリケーションが MSI モードで実行中のサーバにデプロイされた場合に、サーバは誤ってステージングディレクトリを作成しました。現在は、ステージングされたアプリケーションが MSI モードで実行中のサーバにデプロイされると、WebLogic Server は既存のステージングディレクトリを使用するか、新しいステージングディレクトリを作成するかを判断します。その結果、MSI モードで実行中のサーバ(サーバプロセスはディレクトリ構造へのアクセス権を持たない)にデプロイされるときに <code>InvalidAttributeValueException</code> は発生しなくなりました。
CR196774、 CR135780	WebLogic Server の起動中に WebLogic コンフィグレーション プロパティが使用され、そのプロパティが WebLogic Server 管理サブシステムに認識されない場合、重大度 WARNING のメッセージがログに記録されました。 メッセージの重大度を INFO に下げて、メッセージの性質を正確に反映するようにメッセージのテキストを変更しました。
CR224350	<code>weblogic.management.timer.Timer</code> の <code>handleNotification()</code> メソッドで未処理の例外がある場合、タイマー通知の配信が行われませんでした。
CR235443	管理 MBean サーバスタブで初期化される <code>comm</code> MBean プロキシスタブは、管理サーバの再起動時に更新されるようになりました。

## サービス パック 7 の解決済みの問題

変更要求番号	説明
CR259326	<p>JDBCDataSourceFactory の Properties 属性のパスワードは暗号化されていませんでした。</p> <p>現在は、JDBCDataSourceFactoryMBean に Password 属性が追加されています。この属性で指定された値は暗号化され、パスワードへのアクセスは管理者に制限されます。また、JDBCDataSourceFactoryMBean.Properties 属性にはパスワードを指定できなくなりました。</p>
CR262604	<p>匿名ユーザは weblogic.Admin PING を使用できなくなりました。この動作に問題はありますが、不適切なエラー メッセージが表示されていました。</p> <p>匿名の JNDI ルックアップが許可されていないことを示すように、エラー メッセージが変更されました。</p>

## プラグイン

変更要求番号	説明
CR201397	<p>Idempotent=OFF の場合、WLIOTimeoutSecs の経過後に READ_TIMEOUT 例外が発生すると、IIS プロキシ プラグインは再試行しません。</p>
CR205132	<p>HP-UX で、Apache 2.0.49 (with mod_wl_20.so) の子プロセスが常に終了するようになりました。</p>
CR206043	<p>4096 バイトより大きな HTTP GET リクエストを実行する場合、wlproxy ログ ファイルに以下のエラーが表示されます。</p> <pre>request too long: XXXX, max is 4096</pre> <p>この問題は解決されて、エラーは送出されなくなりました。</p>
CR206340	<p>WebLogic Server を停止するときのフェイルオーバー ロジックが修正されました。</p> <p>SHUTDOWN シーケンスの開始後に、WebLogic Server は HTTP 404 を返さなくなりました。</p>



変更要求番号	説明
CR210417	<p>WebLogic Server のクラスタ化環境でプラグインが設定されている場合、フェイルオーバー時に HTTP リクエストは常にセカンダリにルーティングされていました。その結果、セカンダリはすべてのセッションをホストし、サーバの能力の 50% 以上を使用していました。</p> <p>現在、プラグインはデフォルトではセカンダリにフェイルオーバーしません。代わりに、使用可能な次のサーバにリクエストをランダムにルーティングします。</p>
CR210497	<p>インストーラの <code>build-native.xml</code> ファイルでバイナリのパスが間違っていたため、<code>iisproxy128.dll</code> のビットをインストーラで使用できませんでした。</p> <p>パスを修正したので <code>iisproxy128.dll</code> のビットをインストーラで使用できるようになりました。</p>
CR210899	<p><code>PathTrim</code> および <code>PathPrepend</code> プロパティと一緒に <code>DefaultFileName</code> プロパティが設定されている場合に、最終的な URI が誤った形式になることはなくなりました。</p>
CR213809	<p>HP-UX プラットフォームで、以前の Apache プラグインが未定義のシンボル <code>AP_NEED_SET_MUTEX_PERMS</code> を使用していました。</p> <p>この問題は解決済みです。最新のパッチについては、BEA カスタマ サポートにお問い合わせください。</p>
CR216445	<p>応答ヘッダのサイズ制限は削除されました。NSAPI プラグインでは 16k より大きな応答ヘッダを使用できます。</p>
CR216621	<p><code>mod_rewrite</code> によって行われた URI の変更は <code>request_rec-&gt;uri</code> にのみ反映されます。そのため、Apache プラグインはデフォルトで <code>request_rec-&gt;uri</code> を使用するようになりました。</p> <p>また、新しいプロパティ <code>WLForwardUriUnparsed</code> が追加されました。このプロパティを ON に設定した場合、Apache プラグインは <code>request_rec-&gt;uri</code> の代わりに <code>request_rec-&gt;unparsed_uri</code> を使用します。</p> <p><b>注意:</b> Apache プラグインで <code>WLForwardUriUnparsed</code> を ON に設定した場合、<code>mod_rewrite</code> を適切に使用できません。</p>
CR216841	<p>未解析の URI を使用する場合、<code>pathPrepend</code> と <code>pathTrim</code> に関する処理を行う前に、絶対 URL が相対 URL に変換されていませんでした。</p> <p>現在、Apache プラグインは絶対 URL を検出すると、その URL を相対 URI に変換してから、<code>pathTrim</code> および <code>pathPrepend</code> プロパティを適用します。</p>

## サービス パック 7 の解決済みの問題

変更要求番号	説明
CR218494	Linux および HP_UX 上でメモリベースのロック メカニズムを使用して正常に停止するときに、WebLogic Server はエラーを送出しなくなりました。
CR220685	以下の条件に該当する場合に、WebLogic Server は新しい接続を使用して再試行しなくなりました。 <ul style="list-style-type: none"><li>• Idempotent が OFF に設定されている。</li><li>• 再利用された接続が使用されている。</li><li>• バックエンド サーバの応答がない。</li></ul>
CR222702	プラグインで APLOG_NOTICE ログ レベルが使用されている場合、Apache Web サーバは負荷のかかった状況で巨大なログ ファイルを生成しなくなりました。
CR222855	SunOne 6.1 の最新のプロキシ プラグインでは、WLIOTimeoutSecs より前に ErrorPage を提供するようになりました。
CR223636	複数のクラスタがある環境では、プラグインのログ ファイルで特定のリクエストに関するすべてのログ メッセージを追跡するのは困難でした。 現在は、各ログ メッセージにユニークなリクエスト ID が関連付けられています。
CR224513	NSAPI プラグインは、WLExcludePathOrMimeType プロパティの値として指定されたディレクトリ名に一致するリクエストをブロックするようになりました。
CR229338	プラグインのインフラストラクチャでは、デフォルトのログ ファイルを開けない場合とホスト名を解決できない場合に、エラーをログに記録するようになりました。
CR229990	ISAPI プラグインから応答を受信するときの 200 ミリ秒の遅延を避けるため、Nagle アルゴリズムを無効にしました。
CR231085、 CR205852	応答から transfer-encoding ヘッダを削除することでこの問題を解決しました。 HTTPClusterServlet がデプロイされている Web コンテナでは、transfer-encoding ヘッダをクライアントに送信すべきかどうかを判断するようになりました。

変更要求番号	説明
CR232588	WebLogic Apache モジュールの <code>CookieName</code> パラメータは Apache <code>mod_usertrack</code> モジュールの <code>CookieName</code> パラメータをオーバーライドします。 <code>CookieName</code> パラメータを WebLogic Apache モジュールの <code>WLCookieName</code> で置き換えてください。
CR233924	現在、 <code>HTTPClusterServlet</code> は最初に優先プライマリ サーバ、次にセカンダリサーバを試行します。いずれのサーバも使用できない場合、リクエストは使用可能な次のサーバにリダイレクトされます。
CR235434	NSAPI プラグインと Sun One 6.1 を使用している場合に、HTTP 応答ヘッダは 2 つの「Date」フィールドを記録することはなくなりました。
CR237058	<code>iisproxy.dll</code> を実行する W3SVC IIS アプリケーション プールがクラッシュすることがありました。 この問題は解決済みです。
CR237675	IIS にホストされているリソースが直接要求を受けた場合、応答は 1 パケットで送信されました。しかし、WebLogic Server にホストされているリソースが IIS プラグインを介して要求を受けた場合、サーバが生成した応答は 2 パケットで送信されました (ただし、サーバから IIS へは 1 パケットで送信されていました)。 1 つの TCP パケットにヘッダと本文が含まれている場合 (WebLogic Server から IIS に送信された場合)、IIS プロキシ プラグインは応答を送信するようになりました。
CR238106	クラスタ化環境において、WebDAV 実装を備えたクライアントが <code>HTTPClusterServlet</code> を介して管理対象サーバにアクセスした場合に、WebLogic Server は <code>java.net.SocketTimeoutException</code> を送出しなくなりました。
CR240593	クラスタ ノードの正常な停止によって進行中のセッションが失敗することなくなりました。
CR243059	バックエンドの WebLogic クラスタが負荷の高い状況にある場合、NSAPI プラグインが原因で Web サーバ プロセスがクラッシュしました。 この問題は解決済みです。

## サービス パック 7 の解決済みの問題

変更要求番号	説明
CR245461	Web アプリケーションのウェルカム ページが、PathPrepend および PathTrim プロパティが設定された Apache プラグインからアクセスされ場合、HTTP リクエストの無限ループがブラウザからプラグイン経由で WebLogic Server に送信されました。 この問題は解決済みです。
CR251562	クライアントが HTTP チャンク転送として大きな POST データを送信する場合に、Apache プラグインは失敗しなくなりました。
CR253814	HTTP 転送エンコーディングが「chunked」である場合に、Apache プラグインは失敗しなくなりました。
CR255498	Apache (2.0.48 より前のバージョン) 用の WebLogic プラグインは、MPM がマルチスレッドの場合にのみミューテックスを作成するように再設計されました。
CR259649	値がディレクトリに設定されている場合、WLExcludePathOrMimeType プロパティは適切に機能しませんでした。 この問題は解決済みです。

## RMI

変更要求番号	説明
CR215390	サーバがクライアントにリクエストを送信するときに RMI <code>ClassCastException</code> は発生しなくなりました。
CR249240	2 つのコード パスが異なる順序でロックを取得しようとした場合に、IOP <code>ConnectionManager</code> で Java レベルのデッドロックは発生しなくなりました。

# セキュリティ

変更要求番号	説明
CR136729	<p>このリリースより前では、コンフィグレーション済みの監査重大度レベル以上の重大度レベルのイベントがログに記録されていました。</p> <p>現在、デフォルト監査プロバイダのコンフィグレーションには <code>CUSTOM</code> 重大度オプションがあります。レベルを <code>CUSTOM</code> に設定し、1 つまたは複数の重大度レベルを指定した場合は、指定されたレベルのイベントのみが監査されます。ただし、このオプションを設定しない場合は、コンフィグレーション済みの監査重大度レベル以上の重大度レベルを持つすべてのイベントがログに記録されます。</p> <p>重大度レベルを示すシステム プロパティは <code>weblogic.security.auditInformation</code>、<code>weblogic.security.auditWarning</code>、<code>weblogic.security.auditError</code>、<code>weblogic.security.auditSuccess</code>、および <code>weblogic.security.auditFailure</code> です。</p> <p>たとえば、<code>INFORMATION</code> と <code>FAILURE</code> のイベントのみをログに記録するには、プロバイダで <code>CUSTOM</code> と設定し、これらのシステム プロパティを <code>WebLogic Server</code> に対して設定します。</p> <pre>-Dweblogic.security.auditInformation=true -Dweblogic.security.auditFailure=true</pre>
CR182006	<p>x509 証明書の <code>SubjectAlternativeName</code> 拡張が重大 (クリティカル) としてマークされていたことが原因で、証明書検証が失敗しました。</p> <p>x509 証明書でこの拡張が重大 (クリティカル) としてマークされていても、SSL ハンドシェイク中にその証明書が検証されるようになりました。</p>
CR210310、 CR231457	<p>基本制約拡張でパスの長さの制約に関するフィールドが省略されている CA 証明書が CA 証明書として認識されませんでした。その結果、この証明書を含む証明書チェーンは、SSL 接続中に証明書検証に失敗しました。現在、このような証明書はパスの長さの制約を持たない CA 証明書として正しく解釈されます。</p>
CR212348	<p>いずれかのグループ メンバーの識別名が無効な場合、<code>listGroupMembers()</code> メソッドの SSPI 呼び出しが例外により中断されました。</p> <p><code>listgroupmembers()</code> メソッドは、有効なグループを表示し、識別名を検証できないメンバーを無視するようになりました。</p>

## サービス パック 7 の解決済みの問題

変更要求番号	説明
CR215515	WebLogic Server が長さ 0 のパケットを送信する場合、一部のバージョンの Internet Explorer がハングします。 現在は、パブリック プロパティ <code>-Dweblogic.security.SSL.DisableSSLEmptyRecords=true</code> を追加することで、長さ 0 のパケットの送信を無効にすることができます。このプロパティを有効にすると、Internet Explorer がこの原因によってハングするのを防ぐことができます。このプロパティのデフォルト値は <code>false</code> です。
CR215960	JSAFE ネイティブライブラリの追加がサポートされました。WebLogic Server 7.0 SP7 には <code>jsafe.jar</code> (バージョン 3.4.3) のみが付属しているため、他のネイティブライブラリについては、ベンダのサポート サービスにお問い合わせください。
CR218240、 CR231517、 CR230119、 CR230439、 CR233424	<a href="http://dev2dev.bea.com/pub/advisory/122">http://dev2dev.bea.com/pub/advisory/122</a> のセキュリティ勧告を参照してください。
CR234477	関連付けられたデプロイメント記述子に、ワイルドカード文字またはエスケープされる引用符を使用したパーミッション文が含まれていた場合、リソースアダプタのデプロイ中に <code>InvalidParameterException</code> が送出されました。 この問題は解決済みです。
CR242606	(ドメイン側のセキュリティ コンフィグレーションの一部として) <code>MasterFirst</code> フラグが <code>True</code> に設定されていて、組み込み LDAP のデフォルトのアクセス制御リストが使用されている場合に、管理対象サーバは起動時に失敗しなくなりました。
CR243242	認証プロバイダでユーザ名の大きい文字 / 小さい文字を区別せず、 <code>caseSensitiveUserNames</code> が <code>false</code> (デフォルト値) に設定されている場合、または、認証プロバイダでユーザ名の大きい文字 / 小さい文字を区別し、 <code>caseSensitiveUserNames</code> が <code>true</code> に設定されている場合、ユーザのロック、アンロック、および誤ったパスワードの数の動作が正しくなりました。
CR250189、 CR240904	動的グループの <code>memberURL</code> 属性は、そのグループに属するユーザを正しく取得するようになりました。

変更要求番号	説明
CR250264	WebLogic Server に用意されているデフォルトのアクセス制御リストが原因で、ユーザは関連付けられたパスワードを変更できませんでした。 この問題は解決済みです。
CR252670	javax.crypto.Cipher に対する呼び出しが行われるたびに weblogic.jar が検証されていたため、発信 SSL 接続の速度が大幅に低下していました。 この問題は解決済みです。
CR261064	WebLogic キーストア プロバイダを JCA 仕様に従って指定し、管理者がコンフィグレーションできるようになりました。

## サーブレット

変更要求番号	説明
CR106364	相対パス (/../../test.jsp など) を含む URL が RequestDispatcher.forward () メソッドに渡される場合、404 エラーは発生しなくなりました。
CR173426	weblogic700-web-jar.dtd ファイルに servlet-name 要素定義が追加されました。

## サービス パック 7 の解決済みの問題

変更要求番号	説明
CR185454、 CR110293	<p>セッション数のカウントは概算ですが、サーバの再起動時に 0 に初期化されていませんでした。また、インクリメントとデクリメントが同期していませんでした。このため、競合状態が発生して正確にカウントされませんでした。さらに、レプリケートされたセッションは、becomePrimary、becomeSecondary、および becomeUnregistered コールバックではインクリメントおよびデクリメントされていませんでした。</p> <p>現在は、メモリおよびレプリケートされたセッションのカウントがマップから直接取得されるようになりました。ファイルベースのセッションの場合はファイルシステムから、JDBC の場合はデータベース クエリ経由でカウントが取得されます。この修正の結果、セッション数が正確にカウントされるようになりました。ただし、実行時 MBean が返すカウントが 2 つ増えたことに注意してください。</p> <ol style="list-style-type: none"><li>1. これまでにサーバが開いたセッションの総数</li><li>2. メモリ内のオープン セッションの最大数</li></ol> <p>この変更は、これら 2 つのカウントには影響しません。インクリメントとデクリメントの方法はこれまでどおりです。これらのカウントは概算で、サーバの再起動以降の数を表します。これらは、サーバの再起動後に 0 にリセットされません。</p>
CR204553、 CR136735、 CR256088	<p>カスタム ロガーを使用し <code>getRemoteUser</code> を呼び出すと、<code>SessionData</code> のメモリ リークがサーブレット コンテナで発生しました。</p> <p><code>SessionData</code> の参照回数は、リクエストをロギングしている間にカスタム ロガーが <code>HttpAccountingInfo.getRemoteUser()</code>、<code>getRemoteUse()</code>、<code>getRequestedSessionId()</code>、<code>getUserPrincipal()</code>、または <code>isRequestedSessionIdValid()</code> を呼び出すと増加する場合があります。これは、これらのメソッドが <code>SessionInternal</code> オブジェクトを取得しようとするためです。しかし参照回数は減少しません。</p> <p>この問題は解決済みです。変更によって、<code>ServletRequestImpl</code> の <code>getRemoteUse()</code>、<code>getRequestedSessionId()</code>、<code>getUserPrincipal()</code>、および <code>isRequestedSessionIdValid()</code> から返される値は、リクエストのロギング前に <code>HttpAccountingInfoImpl</code> オブジェクトにキャッシュされ、カスタム ロガーが <code>HttpAccountingInfo</code> のメソッドを呼び出すとキャッシュされた値が使用されるようになりました。これにより、参照回数が増加しなくなります。</p>



変更要求番号	説明
CR208696	<p>war アーカイブをデプロイするために <code>external_stage</code> モードを使用すると、<code>.wlnotdelete</code> ディレクトリの既存の war アーカイブは停止時に削除されて、起動時に再びコピーされていました。この問題によってパフォーマンスが低下しました。</p> <p>現在、既存の war アーカイブは停止時に削除されなくなったため、サーバの起動時間は大幅に短縮されました。</p>
CR214104、 CR222683、 CR228624	<p>最初の 4 文字に予約されたキーワードを指定した SessionID が生成されると、パフォーマンスが影響を受けていました。この問題を修正するために SessionID の生成ロジックを変更しました。</p>
CR214457、 CR202495、 CR236810、 CR239392、 CR206002	<p>以下の URL のセキュリティ勧告を参照してください。  <a href="http://dev2dev.bea.com/pub/advisory/130">http://dev2dev.bea.com/pub/advisory/130</a></p>
CR223609	<p>Web アプリケーション <code>ClassLoader</code> は、アプリケーションの Web アーカイブ内のマニフェスト クラスパス エントリを無視しました。そのため、この Web アプリケーションを実行しようとする、クラスが見つからないことを示す例外が発生しました。</p> <p>この問題は解決済みです。</p>
CR229577	<p><code>FormBasedAuthentication</code> で、ユーザがフォームベースのログイン ページを使用して認証を受ける前と後に、要求された URL が適切に格納されませんでした。</p> <p>元のリクエストの完全な URL (<code>http</code> または <code>https</code> プロトコル方式を含む) を格納するために、<code>weblogic.xml</code> ファイルに次のパラメータが追加されました。</p> <pre data-bbox="370 1256 1119 1350">&lt;container-descriptor&gt;   &lt;retain-original-url&gt; true &lt;/retain-original-url&gt; &lt;/container-descriptor&gt;</pre>
CR236526	<p>JDBC 永続性の場合、サーバ間のフェイルオーバーの際に、古くなったセッションを含むリクエストは提供されなくなりました。</p>
CR239831	<p><code>autoFlush=false</code> の JSP から大きなサイズのファイルが転送されると、<code>ChunkOutput.writeStream()</code> で無限ループが発生しました。</p> <p>この問題は解決済みです。</p>

## Simple Network Management Protocol (SNMP)

変更要求番号	説明
CR222114	SNMP coldStart トラップのエンタープライズ OID が誤っていました。「.1.3.6.1.4.1.140.625」に変更されました。

## Web サービス

変更要求番号	説明
CR214959	SOAP メッセージ内に要素の配列が格納された場合、メソッドで渡される実際の値に関係なく、以降の要素が常に null 値として Web サービス メソッド変数に渡されました。
CR257606	SSL を使用して Web サービスの呼び出しを行ったとき、WSDL の取得に使用されたソケットが閉じられたため <code>java.net.SocketException</code> が発生しました。このエラーは、開かれているファイル数が多過ぎたために発生しました。 この問題は解決済みです。
CR258325	発信 SSL 呼び出しを行う Web サービス クライアントでメモリ リークは発生しなくなりました。
CR259227	Red Hat Enterprise Linux 4.0 AS で、 <code>generateClientJar</code> (clientgen API) が <code>ClientGenException</code> により失敗しました。 この問題は解決済みです。
CR262186	WSDL メタデータにアクセスできないサービス オブジェクトがポート情報を判別できない場合に、 <code>ServiceException</code> が送出されるようになりました。

# XML

変更要求番号	説明
CR215016	<p>中央の Web サーバから HTTP を介して DTD を検索する場合、その URL でエラーが発生すると（つまり、Web サーバが一時的に停止すると）、XML レジストリはキャッシュ内の DTD を更新しないため、その DTD が使用されるたびにエラーが発生しました。</p> <p>現在、XML レジストリはキャッシュ内の DTD を更新します。それによって、問題は解決されました。</p>

## サービス パック 7 の解決済みの問題

# サービス パック 1 から 6 で解決済みの問題

以下の節では、WebLogic Server 7.0 の以前のサービス パックで修正されている問題を説明します。サービス パックは累積的であり、サービス パック 6 にはそれ以前にリリースされた WebLogic Server 7.0 用のサービス パックで行われたすべての修正が含まれます。最新のサービス パックで解決された問題の説明については、[第 3 章「サービス パック 7 の解決済みの問題」](#)を参照してください。

- [4-1 ページの「WebLogic Server 7.0 サービス パック 6 のソリューション」](#)
- [4-61 ページの「WebLogic Server 7.0 サービス パック 5 のソリューション」](#)
- [4-175 ページの「WebLogic Server 7.0 サービス パック 4 のソリューション」](#)
- [4-186 ページの「WebLogic Server 7.0 サービス パック 3 のソリューション」](#)
- [4-303 ページの「WebLogic Server 7.0 サービス パック 2 のソリューション」](#)
- [4-348 ページの「WebLogic Server 7.0 サービス パック 1 のソリューション」](#)

## WebLogic Server 7.0 サービス パック 6 のソリューション

以下の節では、WebLogic Server 7.0 サービス パック 6 で解決された問題について説明します。

## サービス パック 1 から 6 で解決済みの問題

- 4-62 ページの「Administration Console」
- 4-65 ページの「クラスタ」
- 4-68 ページの「コネクタ」
- 4-69 ページの「コア サーバ」
- 4-83 ページの「デプロイメント」
- 4-86 ページの「EJB」
- 4-17 ページの「インストール」
- 4-17 ページの「インターナショナルライゼーション」
- 4-18 ページの「J2EE」
- 4-97 ページの「JCOM」
- 4-98 ページの「JDBC」
- 4-104 ページの「jDriver」
- 4-105 ページの「JMS」
- 4-27 ページの「JNDI」
- 4-116 ページの「JSP」
- 4-120 ページの「JTA」
- 4-30 ページの「JVM」
- 4-121 ページの「ノード マネージャ」
- 4-122 ページの「操作と管理」
- 4-127 ページの「プラグイン」
- 4-138 ページの「RMI」
- 4-47 ページの「サンプル」
- 4-141 ページの「セキュリティ」
- 4-152 ページの「サブレット」
- 4-165 ページの「SNMP (Simple Network Management Protocol)」

- [4-56 ページの「Web サービス」](#)
- [4-58 ページの「WLEC」](#)
- [4-61 ページの「XML」](#)

## Administration Console

変更要求番号	説明
CR097378	<p>Web アプリケーション デプロイメント記述子の編集に、Administration Console で NullPointerException 例外が発生することがありました。</p> <p>WebLogic Server は記述子が null かどうかをチェックしていませんでした。null の場合は記述子を解析できなくなりました。そのため NullPointerException が送出されました。</p> <p>記述子が null の場合は、適切なエラー アクションが呼び出されて、正しいメッセージが表示されるようになりました。</p>
CR106965	<p>Administration Console では、[すべてのエンティティ EJB ランタイムのモニタ ...] または [すべてのメッセージ駆動型 EJB ランタイムのモニタ ...] で、同様の名前を持つ複数の EJB JAR ファイルの情報が誤って含まれていました。</p> <p>現在の Administration Console では、選択したスコープに関連付けられた MBean のみが表示されます。</p>
CR132429	<p>Administration Console には、最後に入力したユーザを記憶するように、ユーザから指示するオプションはありませんでした。デフォルトでは、最後に入力したユーザは 1 週間有効なクッキーとして保存されていました。</p> <p>現在、Administration Console のログイン フォームには [このコンピュータに ID を記憶] というチェックボックスがあります。このチェックボックスはデフォルトでチェックされています。このチェックボックスをチェックしない場合、WebLogic Server は現在のセッションのユーザ ID のみを記憶します。</p>
CR130110	<p>相対パスではなく絶対パスが指定されている場合にアップロードディレクトリの問題が発生しました。</p> <p>WebLogic Server はアップロードの際に相対パスか絶対パスかをチェックするようになりました。</p>

変更要求番号	説明
CR079771	<p>WebLogic Server はクラスタの起動やドメインの起動 / 強制停止の操作結果を報告するために、ブラウザにファイル リンクを送信していました。そのリンクは、ブラウザが管理サーバと同じマシンで実行される場合のみ機能しました。これはすべてのプラットフォームで同様でした。</p> <p>WebLogic Server は、ファイルを処理してブラウザに HTML 形式で出力を送信するアクション リンクを送信するようになりました。</p>
CR172233	<p>プロバイダのユーザ / グループ リストにアクセスしているときに、あるプロバイダが停止していると、Administration Console ではリストが空白になりました。現在の Administration Console では、アクセス可能なプロバイダのユーザ / グループ リストが表示され、停止しているプロバイダがある場合はエラーが表示されます。</p>
CR174734	<p>削除したレルムを再作成すると、InstanceAlreadyExistException が発生していました。</p> <p>現在は、削除したレルムを再作成しても InstanceAlreadyExistException が送出不されるように、レルムは完全に削除されます。</p>
CR180635	<p>Administration Console のブラウザからアーカイブ ファイルをアップロードしているときに、指定したパスとは関係のないドメイン ディレクトリにアーカイブ ファイルがアップロードされました。</p> <p>現在の WebLogic Server では、ファイルが正しいディレクトリにアップロードされるように、ファイルをサーバにアップロードするときにアップロード ディレクトリ プロパティを使用します。</p>
CR181865	<p>Administration Console では、JMS サーバが対象になっていない場合でも、JMS 分散送り先にメンバーを追加することができました。</p> <p>現在は、そのメンバーの JMS サーバが対象として指定されていない場合、JMS 分散送り先にメンバーを追加することはできません。</p>
CR096998	<p>Administration Console を使用して CMP EJB デプロイメント記述子を編集するときに、WebLogic Server は NoSuchMethodException を送出しなくなりました。</p>
CR106399	<p>examples ドメインで Administration Console からメソッド パーミッションを作成してサーバを再起動した場合、WebLogic Server は「page not found」例外を送出していました。</p> <p>ejb20 用の MethodPermission.jsp を追加して問題を解決しました。</p>



変更要求番号	説明
CR188452	<p>Administration Console はサーバ ログを表示できませんでした。代わりに、例外が送出されました。</p> <p>コードの修正で、この問題は解消されました。現在は、サーバ ログが Administration Console に表示されます。</p>
CR081329	<p>Administration Console の更新機能が修正されました。[更新] ボタンをクリックしても以前のようにページが開かれることはありません。代わりに、現在開いているページが更新されます。</p>
CR206741	<p>WebLogic Server は、特に対象がクラスタまたは仮想ホストである場合に、その対象上のコンポーネントのデプロイメント ステータスを検索するために、非常に多数の RMI 呼び出しを行っていました。また、対象のコンポーネント ステータスは、その対象の正確なデプロイメント ステータスを表していませんでした。</p> <p>コンポーネントの可用性ステータスは、そのコンポーネントがサーバ上で動作しているかどうかを示すものです。対象のクラスタまたは仮想ホストのすべてのサーバ上にある、特定のコンポーネントの現在の可用性も示します。可用性ステータスは、対象のサーバが正常停止または強制停止するときに更新されます。</p> <p>デプロイメント ステータスと可用性ステータスを集約して、コンポーネントのデプロイメント ステータスが表示されるようになりました。コンポーネントの集約されたデプロイメント ステータスは、コンポーネントが1つのサーバにデプロイされている場合は [使用可能] または [使用不可能]、クラスタまたは仮想ホストにデプロイされている場合は [使用可能]、[使用不可能]、または [一部使用可能] になります。[一部使用可能] デプロイメント ステータスは、そのコンポーネントがクラスタまたは仮想ホストの一部のサーバでのみ使用可能であることを表しています。</p> <p>この変更の結果、クラスタ デプロイメントのステータス レポートを取得するのに必要な RMI 呼び出しの数が最小限に抑えられ、パフォーマンスが向上しました。</p>

変更要求番号	説明
CR214994	Administration Console の [ サーバ   モニタ   バージョン ] ページでは、システム プロパティ <code>java.vm.vendor</code> が正しく表示されるようになりました。
CR178658	<p>サーバとクラスタのコンフィグレーションに [Servlet Extension Case Sensitive] 属性が追加されました。また、セキュリティドメイン コンフィグレーションには [Web App Files Case Insensitive] 属性が追加されました。</p> <p>これらの属性では、win32 以外のすべてのプラットフォームにおいて、Java Server Pages (JSP) のファイル ルックアップで大文字 / 小文字を区別するかどうかを指定します。標準の win32 ファイル システムにおけるファイル ルックアップでは、常に大文字 / 小文字は区別されません。win 32 以外の大文字 / 小文字を区別しないファイル システム (UNIX の NT Samba マウントや、大文字 / 小文字を区別しないモードでインストールされた Mac OS など) で、大文字 / 小文字を区別しないルックアップを指定すると (これらの属性を false に設定する)、JSP はそのソース コードを返さなくなります。たとえば、JSP が Samba マウントから提供されていて、大文字 / 小文字を区別しないルックアップを指定した場合、WebLogic Server はすべてのファイル名拡張子を小文字に変換してから JSP をルックアップします。</p> <p>使用できる値と用法 :</p> <ol style="list-style-type: none"> <li>OS ( デフォルト値 ): WebLogic Server はパターン マッチングに関してオペレーティング システムに依存する。</li> <li>True: WebLogic Server はオペレーティング システムに関係なく大文字 / 小文字を区別しない。</li> <li>False: WebLogic Server はオペレーティング システムに関係なく大文字 / 小文字を区別する。</li> </ol>

## クラスタ

変更要求番号	説明
CR177776	<p>WebLogic Server では、レプリケートされたセッション http リクエストが、そのリクエストのプライマリでもセカンダリでもないサーバに届いたときに、クラスタで分散デッドロックが発生しました。このようなリクエストがクラスタ内のサーバにいくつも送られると、それが原因でデフォルト スレッド プールのすべてのスレッドが消費されて、ある時点になると、デフォルト スレッド プールには応答を受信するために使用できるスレッドがなくなります。その結果分散デッドロックが発生します。</p> <p>WebLogic Server では、このような状況でデッドロックが発生しなくなりました。</p>
CR206375	<p>クラスタ内で http リクエストをロード バランシングすると、以下のような誤ったエラー メッセージが報告されることがありました。</p> <pre>Error while primary becoming secondary, [No old secondary found for roid:&lt;roid&gt;] or [No new primary found for roid:&lt;roid&gt;]</pre> <p>このエラー メッセージは、プラグインまたはロード バランサで、セッション維持のコンフィグレーションに問題があることを示しています。</p> <p>WebLogic Server は、問題を説明し、ロード バランサまたはプラグインでセッション維持のコンフィグレーションをチェックするように指示する、正しいメッセージを表示するようになりました。</p>

## コネクタ

変更要求番号	説明
CR184893	RAR からのクラスローディングの速度が低下しなくなりました。

## コア サーバ

変更要求番号	説明
CR108791	<p>サーバ状態が SHUTDOWN_IN_PROCESS のときに Runtime.getState が呼び出されると、WebLogic Server は誤った文字列を返していました。</p> <p>現在の WebLogic Server は、サーバの状態を表す正しい文字列を返します。</p> <p>アプリケーションが以前の誤った文字列定数に依存していた場合は、文字列定数を変更する必要があります。</p>
CR125000 CR129560	<p>クラスタ サービスでは、マルチキャスト ポートへのバインド後にプロセスの実際の uid/gid を設定していました。その結果、WebLogic Server がリスン ポートへのバインド後にユーザを切り替えようとしたときに、エラーが発生しました。</p> <p>現在、WebLogic Server は、ポートへバインドする場合を除いて、サーバの起動時にバインド後の uid/gid を使用し、すべてのポートでリスンした後でのみ実際の uid/gid に切り替えます。</p>
CR121483	<p>モニタ サブシステムが abbrev テーブルの値を読み込もうとしていて、同時に abbrev テーブルが rjvm レイヤによって変更されていたときに、ConcurrentModificationException が送出されました。</p> <p>HashMap が 2 つのスレッドから同時に変更されないように、モニタ サブシステムにデータを送信する前にキー セットを複製することで、ConcurrentModificationException を回避しました。</p>
CR130376	<p>ドキュメントの「サーバの起動と停止」の節では、CLASSPATH が長すぎる場合でも 1 行としてファイルに追加され、-classpath @filename としてアクセスできると説明されていました。しかし、beasvc が CLASSPATH ファイルの内容をロードする際に最後の文字を削除してしまうことがあったため、上記の説明どおりには機能していませんでした。この問題は、ファイルが改行で終わっていない場合にのみ発生していました。</p> <p>この問題は解決済みです。</p>

変更要求番号	説明
CR133631	<p>beasvc.exe でコンフィグレーションされている Windows サービスを停止する場合、停止クラスを指定するときにタイムアウトが発生することがありました。beasvc.exe の停止スレッドとメイン スレッドの間で競合状態が発生したため、サービスがタイムアウトしていました。</p> <p>競合状態を修正したため、Windows サービスはタイムアウトしなくなりました。</p>
CR135225	<p>RJVM と NTSocketMuxer の間でデッドロックが発生していました。</p> <p>WebLogic Server がディスパッチ中に IORecord でロックを保持しないようにするコードを追加しました。そのためデッドロックは発生しなくなりました。</p>
CR105444	<p>-Dweblogic.Stdout および -Dweblogic.Stderr コマンドライン オプションを使用してログ ファイルのパスを指定したものの、そのディレクトリが存在しない場合に、サーバは FileNotFoundException Exception を送出していました。</p> <p>コマンドライン オプションで指定されたディレクトリ構造が存在しない場合は、そのディレクトリを作成するようにコードを改良して、この問題を解決しました。</p>
CR099356、 CR099005、CR186191	<p>JVMWebSocketManager で競合状況が発生すると、MuxableSocketDiscriminator.isMessageComplete が NullPointerException を送出し、その結果 JVMWebSocketManager が無限ループに入っていました。</p> <p>現在は、JVMWebSocketReader が静的イニシャライズで必要な情報を初期化するので、初期化は 1 回だけ行われて、無限ループは発生しません。</p>
CR130409	<p>カスタム キュー上の「リソースを大量に消費する動作の遅い」リクエストを抑制するために -Dweblogic.kernel.allowQueueThrottling フラグを使用して実行キューの最大長を設定しても、クライアントは 503 応答を受け取ることなくタイムアウトを待機していました。</p> <p>この問題は、呼び出し側のディスパッチの戻り値をチェックし、sendError() API を使用して 503 応答を返すようにコードを修正することで解決しました。</p>

変更要求番号	説明
CR174955、CR105257	<p>IOP 接続プールが初期化される前に WLECSERVICE.getConnectionPoolCount を呼び出すと、NullPointerException が発生していました。</p> <p>現在は、IOP 接続プールが null の場合には 0 を返します。その結果、WLECSERVICE.getConnectionPoolCount を呼び出したときに NullPointerException が送出されなくなりました。</p>
CR180426	<p>WebLogic Server がクラスタ内で動作しているときに、クラスタの DNS 名が設定されていない場合、警告メッセージが表示されました。この警告メッセージによってクラスタのパフォーマンスが低下していました。</p> <p>現在は、クラスタアドレスがネットワーク チャネルで設定されていない場合、クラスタアドレスは常にクラスタ MBean から読み込まれます。その結果、警告メッセージによるクラスタのパフォーマンスの問題は起きなくなりました。</p>
CR180532	<p>ドメイン ログとサーバログが同時にローテーションするときに、デッドロックが発生することがありました。</p> <p>現在、WebLogic Server はログをローテーションする間はメッセージをログに記録しません。その結果、ログのローテーション時にデッドロックが発生しなくなりました。</p>
CR173958、CR185090	<p>フロントエンドとバックエンドの間のプロトコルを「セキュア」から「非セキュア」に変更した場合に、t3 を使用した 6.1 SP4 ドメインと 7.0 SP5 ドメインの相互運用性が正しく機能しませんでした。認証呼び出しの際に、フロントエンド QOS がバックエンドに伝播され、呼び出しに失敗していました。</p> <p>この問題は、ブートストラップ認証呼び出しに「匿名」を使用することで解決しました。</p>
CR175607	<p>WebLogic Server をアンインストールした直後に Windows サービスとしてインストールしなおすと、誤ったレジストリ キーが作成されることがあったため、正常に起動できなくなる問題が発生していました。</p> <p>コードを修正したことで、レジストリ キーがフラッシュされて正しく閉じられるようになりました。</p>

変更要求番号	説明
CR179262	<p>起動時に、weblogic.jms.common.DistributedDestinationManager と weblogic.cluster.MemberManager の間でデッドロックが発生しました。</p> <p>MemberManager での同期化を減らしたため、デッドロックは起きなくなりました。</p>
CR182838	<p>LocalServerRef が hashCode メソッドを実装していなかったため、異なる PK を持つ複数のエンティティ Bean が同じスタブを保持していました。</p> <p>現在、LocalServerRef は hashCode メソッドを正しく実装しています。</p>
CR185841	<p>多数の管理対象サーバを同時に起動する場合、管理サーバの CPU 使用率が高くなりました。</p> <p>現在の WebLogic Server では、ServerMbean.getName() に対するリモート呼び出しの回数が減っています。その結果、大きなクラスタで管理対象サーバの起動時間が速くなり、管理サーバの CPU 使用率が下がりました。</p>
CR172366	<p>beasvc.exe からイベント ログに出力されるメッセージが、日本語環境では正常に表示されませんでした。</p> <p>コードを修正したことで、英語以外のすべての言語環境でも、英語のメッセージが出力されるようになりました。</p>
CR181986	<p>サービスとして実行される WebLogic Server が多数のスレッドを使用すると、メモリ不足になることがありました。</p> <p>beasvc.exe と beasvc64.exe で使用される予約スタック サイズを 1MB から 256KB に減らすことで、メモリの問題を解消しました。</p>
CR182684	<p>beasvc サービスハンドラが呼び出されるたびに、beasvc ログは呼び出されたことを示す行を追加していました。</p> <p>次に例を示します。</p> <pre>Tue Apr 27 11:52:17 2004] [I] [service_ctrl] 4</pre> <p>そのため、サーバで実際のアクティビティがないときにログが一杯になりました。</p> <p>デバッグ文を削除して、ログの問題を解消しました。</p>

変更要求番号	説明
CR188371	<p>アプリケーションがステートレス セッション Bean のリモート スタブをキャッシュしたときに、クラスタ内のすべてのサーバを再起動すると、スタブでは、リモート オブジェクトが使用できるサーバ ノードのリストを更新できなくなり、フェイルオーバが失敗しました。スタブにはクラスタ ノードで初期コンテキストを再確立するのに必要な情報がなく、リモート メソッドの呼び出しが失敗したためにこの状況が発生しました。</p> <p>WebLogic Server のコードは、WebLogic Server 6.1 ではステートレス セッション Bean スタブのスレッド環境を伝播していませんでしたが、WebLogic Server 7.0 以降ではアンマーシャリングの際に、フェイルオーバが動作する環境を設定するために必要です。</p> <p>現在、クラスタ対応のステートレス セッション Bean の実行時記述子では、propagate-environment 属性がデフォルトで true に設定されています。アンマーシャリングの際に環境が渡されるように、記述子が読み込まれて propagateEnvironment が設定されます。これにより、フェイルオーバロジックはクラスタ ノードを再接続して、リモート オブジェクトが使用できるサーバ ノードの新しいリストを取得し、フェイルオーバを適切に行えるようになりました。</p>
CR190417	<p>リクエストがプライマリでもセカンダリでもないサーバに届いた場合、リクエストはセッション ID があるホストとポートの情報を使用してセカンダリ サーバ上でセッションをルックアップして、既存のセカンダリからセッションを取得していました。リクエストは既存のセカンダリからセッションを正常に取得した後で、既存のセッションを削除して新しいセッションを作成しようとしていました。</p> <p>2 つのリクエストが同時に発生した場合、分散デッドロックが発生する可能性があります。これらのリクエストは「レプリケーション」スレッドキューに届きますが、「レプリケーション」スレッドキューには 2 つのスレッドしかなく、応答を読み取るために使用できる他のスレッドがないためです。</p> <p>問題を修正するコードを WebLogic Server に追加したため、デッドロックは発生しなくなりました。</p>
CR190507	<p>ReplicationManager の一方向の呼び出しに関する問題を修正しました。</p>



変更要求番号	説明
CR203523	<p>以前のリリースの WebLogic Server で表示されていた以下のログメッセージが復活しました。</p> <pre>&lt;Warning&gt; &lt;WebLogicServer&gt; &lt;000333&gt; &lt;Queue usage is greater than QueueLengthThresholdPercent "3%" of the maximum queue size. We'll try to allocate ThreadsIncrease "5" thread(s) to help.&gt;</pre>
CR206459	<p>クライアントサイドで作成できる最大スレッド数の制限 (50) を修正しました。</p> <p>現在、クライアントサイドで作成できる最大スレッド数の制限は 65536 に設定されています。</p>
CR183620	<p>ドメイン コンフィグレーション ウィザードから生成される管理対象サーバの起動スクリプト (startManagedWebLogic.sh または startManagedWebLogic.cmd) には、次のような誤った行が含まれていました。</p> <pre>cd @USERDOMAIN_HOME</pre> <p>この誤った行が原因で次のエラーが発生しました。</p> <pre>C:\bea70sp5\user_projects\mydomain&gt;echo off The system cannot find the path specified.</pre> <p>この問題を修正するために、現在のドメイン コンフィグレーション ウィザードは文字列を置換しています。</p>

## デプロイメント

変更要求番号	説明
CR179645	WebLogic Server では、大きな JAR ファイルを復元する速度が向上しました。
CR183537、 CR187458	<p>アプリケーションの再デプロイ時に対象が指定されない場合、weblogic.Deployer はデフォルトの対象として管理サーバを追加していました。</p> <p>現在は、対象が指定されなかった場合、デプロイされていたアプリケーションは現在のすべての対象に適切に再デプロイされます。新しいアプリケーションはデフォルトで管理サーバにデプロイされます。</p>

---

変更要求番号	説明
CR192196	<p>WebLogic Server 8.1 クライアントが WebLogic 7.0 管理サーバに対して <code>DeployerRuntime.getDeployerRuntime(MBeanHome)</code> を呼び出した場合、<code>InstanceNotFoundException</code> が発生していました。</p> <p><code>DeployerRuntime.getDeployerRuntime(MBeanHome)</code> の実装が変更されました。その結果、WebLogic Server 8.1 クライアントが WebLogic 7.0 管理サーバから (または WebLogic Server 7.0 クライアントが WebLogic 8.1 管理サーバから) <code>DeployerRuntime MBean</code> を取得しても <code>InstanceNotFoundException</code> は発生しなくなりました。</p>
CR124450	<p>サーバが開発モードで動作している場合に、Microsoft Windows エクスプローラを使用して、サーバ上にデプロイされている EAR ファイルをアプリケーションディレクトリから削除しようとした場合、以下のエラーは発生しなくなりました。</p> <p>Cannot delete abc.ear:there has been a sharing violation, The source or destination file may be in use.</p>

---

## EJB

---

変更要求番号	説明
CR055396	<p>EJB QL の構文エラーが発生したときに、WebLogic Server は誤った XML ファイルの参照を含むエラー メッセージを生成しました。</p> <p>EJB QL に構文エラーがある場合、WebLogic Server は次のようなメッセージを生成するようになりました。</p>
CR060229	<p>Administration Console はステートフル EJB とエンティティ EJB の [コミットしたトランザクション数]、[ロールバックしたトランザクション数] または [トランザクションのタイムアウト総数] を表示していませんでした。</p> <p>Administration Console でこれらの項目をモニタできるようになりました。</p>
CR087261	<p>EJBDeployer はメッセージ駆動型 Bean のログに誤ったデプロイメントメッセージを書き込んでいました。</p> <p>メッセージ駆動型 Bean がデプロイされるときに正しいメッセージがログに記録されるようになりました。</p>

---

変更要求番号	説明
CR127369	<p>複数の Bean が同じ Java クラスに基づいている場合、AssertionError が送出されることがありました。</p> <p>このエラーは以下の条件が満たされると発生しました。</p> <ol style="list-style-type: none"><li>1. Bean A が Bean X と多対 1 の関係にある（一方向の関係）。</li><li>2. Bean B が Bean X と多対 1 の関係にある（一方向の関係）。</li><li>3. Bean A と B は同じ Java クラスに基づいた別々のデプロイメントである。</li></ol> <p>Bean の関係を処理する場合、WebLogic Server は cmr-field 名のリストを保持します。cmr-field 名が宣言されていない場合は、Bean のクラス名に基づいて cmr-field 名を作成します。上記の場合では、Bean X の関係を処理するとき、Bean A との関係の cmr-field 名と Bean B との関係の cmr-field 名が作成されます。これらのクラス名は同じであるため、作成される cmr-field 名も同じになります。そのため AssertionError が発生します。</p> <p>cmr-field 名をユニークにするためのコードを追加して、名前が衝突する可能性を排除しました。</p>
CR108169	<p>Administration Console を使用して EJB デプロイメント記述子を編集する際、誤ったメソッド名が表示されていました。</p> <p>正しいメソッド名が表示されるようになりました。</p>
CR135410	<p>Bean をロックアップして初めてビジネス メソッドを呼び出す場合、ejbStore からの EntityContext.getCallerPrincipal() への呼び出しではユーザが「anonymous」として返されていました。その後、ejbStore を再度呼び出すと、正しいユーザ名が返されました。これは.ejbStore 固有の問題であり、他のメソッドでは発生しません。</p> <p>この問題は解決済みです。</p>
CR187121	<p>デプロイメント記述子で idleTimeoutSecs に 60000000 のような大きな値が設定されている場合、1000 を掛けてミリ秒に変換すると、負の値にオーバーフローしていました。その結果、パッシベーションされた Bean をディスクから消去するトリガが頻繁に発生して、CPU の使用率が高くなりました。</p> <p>タイムアウト値をミリ秒単位で保持する EJB コンテナの変数 (idleTimeoutMS、sessionTimeoutMS、readTimeoutMS など) が int 型から long 型に変更されました。これにより数値のオーバーフローは発生しなくなりました。</p>

変更要求番号	説明
CR189847	<p>ステートフル セッション Bean (SFSB) のインメモリ レプリケーションが原因でメモリ リークのような状況が発生しました。</p> <p>メモリ リークは発生していませんでしたが、ヒープメモリが効率的に使用されていませんでした。現在は、SFSB のインメモリ レプリケーションでヒープメモリが効率的に使用されます。</p>
CR196593	<p>NRUCache でのロックの競合が原因で、ステートフル セッション Bean (SFSB) のレプリケーション時にサーバがハングしていました。</p> <p>SFSB のレプリケーション時にサーバがハングしないように、NRUCache でのロックの競合を削減しました。</p>
CR197817	<p>WebLogic Server がエンティティ Bean のファインダ メソッドを実行すると、NullPointerException が送出されていました。</p> <p>NullPointerException を解消するため、現在の生成コードでは Bean が null の場合も予期しており、Bean は適切に処理されるようになりました。</p>
CR132510	<p>Optimistic 同時方式を使用し、cache-between-transaction が true に設定されたエンティティ Bean で、リレーションシップ キャッシングを有効にした場合、WebLogic Server は IllegalStateException を送出しなくなりました。</p>
CR205974	<p>コレクションの値を持つ CMR フィールドが、作成元のトランザクション以外のトランザクションでアクセスされた場合に、WebLogic Server は IllegalStateException を送出しなくなりました。</p> <p>その結果、WebLogic Server は、キャッシング設定の有効 / 無効に関係なく、ファインダ クエリから正しい結果セットを配信します。</p>
CR203644	<p>インメモリ レプリケート セッションによりレプリケートされたクラスタ内のステートフル セッション Bean は、インスタンスがバッチセッションされた後、またはクラスタ インスタンスが停止した後に NoSuchObjectException または LeasedRemoteRef エラーを送出しなくなりました。</p>

## インストール

---

変更要求番号	説明
CR121040	<p>サービスをアンインストールして再インストールしないと、WebLogic Server Windows サービスのパスワードを変更することはできませんでした。</p> <p>現在は、サービスを再インストールしないでもパラメータを編集できるように、edit オプションが用意されています。</p> <p>個別のまたは複数のオプションを次のように同時に編集できます。</p> <pre>beasvc -edit -svcname:wls_service -javahome:"C:\java\java142_05" -password:"blah" (このコマンドでは javahome および password パラメータの値が変わります)</pre> <p>このオプションはサービスの再起動時に有効になります。</p>

---

## インターナショナルライゼーション

---

変更要求番号	説明
CR079432	<p>MessageLocalizer は、110ngen ユーティリティを使用して、ローカライズされたカタログ ファイルに 110n_package 属性を設定していませんでした。</p> <p>MessageLocalizer はローカライズされたカタログ ファイルに 110n_package 属性を正しく設定するようになりました。</p>

---

## J2EE

変更要求番号	説明
CR132360	<p>weblogic.j2eeclient.Main が WebStart で動作しませんでした。このクライアント jar ファイルは、WebStart クライアントによってロードされる引数であり、クラスパスに設定されていました。しかし、現在のディレクトリで使用できなかったため、エラーが発生しました。</p> <p>システム クラスパスからクライアント jar ファイルをロードするために、WebLogic Server にシステム プロパティ (weblogic.j2ee.client.isWebStart) が追加されました。weblogic.j2eeclient.Main は WebStart で動作するようになりました。</p>
CR103309	<p><a href="http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA04-70.00.jsp">http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA04-70.00.jsp</a> のセキュリティ 勧告を参照してください。</p>
CR186016	<p>アプリケーションの再デプロイ時に、WebLogic Server 7.0 サービス パック 5 以前と同様、J2EEApplicationContainer によって JDBCModule のような内部モジュールが再デプロイされるようになりました。</p>
CR097079	<p>アプリケーションが管理対象サーバから削除される時、サーバのステージングディレクトリから補助的な jar ファイルも削除されるようになりました。</p>
CR103506	<p>名前にピリオド「.」が入っているディレクトリが含まれた展開形式の .ear ディレクトリをデプロイするときに、NullPointerException は送出されなくなりました。</p>

## JCOM

変更要求番号	説明
CR185934	<p>Microsoft Security Bulletin MS04-011 が適用されたときに、COM to Java クライアントで障害が起きていました。</p> <p>この問題は解決されています。</p>
CR199844	<p>例外が発生すると COM ソケットがリークしていました。</p> <p>例外が発生した場合、COM ソケットは適切に閉じられるようになりました。</p>

# JDBC

変更要求番号	説明
CR136746	JDBC ドライバの明示的なネーミングはクラス <code>VendorId</code> で使用できませんでした。 この問題は解決済みです。
CR127720	新しいバージョンの JDBC ドライバは、接続のトランザクションの状態を追跡します。接続でローカル トランザクションがアクティブな場合、その接続で XA の操作は実行できず、 <code>xa_start()</code> が呼び出されたときに <code>XAER_PROTO</code> または <code>XAER_RMERR</code> が生じていました。結果として、アプリケーションでは、ローカル トランザクションを開始したが終了していないコード中の場所を絞り込む冗長なプロセスを実行しなければなりませんでした。 この問題は、特殊な XA 接続がプールに 2 度解放されないように回復メソッドのコードを修正して解決しました。
CR174126	複数の異なる Prepared Statement を 1 つのトランザクションで使用すると、XA 対応の Statement キャッシュでバグが発生し、閉じられていない JDBC 文のハンドルが失われていました。このことが原因で、Oracle のすべての利用可能なカーソルがすぐに消費およびリークされていました。 XA 対応の Statement キャッシュを実装するために使用されるコレクション オブジェクトが修正されました。
CR177220	マルチサーバドメインで、グローバル トランザクションに参加しているクライアントが次の例外を受け取っていました。 <code>weblogic.transaction.RollbackException: delist() failed on resource jdbcXAPool</code> データソースが null の場合はリソース マネージャが設定されないように、コードが変更されました。データソースが初期化されると、リソース マネージャは <code>VendorXAResource</code> によってラップされます。
CR180097	JDBC 接続プールの「 <code>CountOfTestFailuresTillFlush</code> 」プロパティで、文のキャッシュがクリアされていませんでした。 この問題は、コードを修正することで解決しました。 「フラッシュ プール」機能の使い方の詳細については、『WebLogic JDBC プログラマーズ ガイド』の「 <a href="#">JDBC 接続プールのテスト機能の拡張</a> 」を参照してください。

変更要求番号	説明
CR184819	<p>WebLogic Server マルチプールは、RAC の複数のインスタンスがコンフィグレーションされていて、最初の RAC インスタンスが停止した場合に、フェイルオーバーしていませんでした。</p> <p>現在、マルチプールは正しくフェイルオーバーします。</p>
CR182410	<p>負荷の高い状況で、WebLogic Server の速度が大幅に低下しました。</p> <p><code>weblogic.jdbc.common.internal.MultiPool.searchLoadBalance()</code> メソッドから同期化を削除して、負荷の高い状況での速度低下を解消しました。</p>
CR189978	<p>プールのコンフィグレーションが誤っているか、データベースが応答しない場合に、アクティブ化されないプールが作成される可能性があります。プールがアクティブ化されない場合、WebLogic Server は <code>ResourceException</code> を送出しました。この例外が送出されたとき、JDBC マルチプールはアクティブな次のプールにフェイルオーバーしませんでした。</p> <p>現在、WebLogic Server はプールがアクティブ化されないというメッセージと共に <code>ConnectDisabledException</code> を送出します。JDBC マルチプールはこの例外を検出して対応し、使用可能な別のプールがあれば再試行します。その結果、JDBC マルチプールは、リスト内にアクティブ化されないプールがあった場合、アクティブなプールにフェイルオーバーするようになりました。</p>
CR187604	<p>JTS Statement ラッパーの <code>getMoreResults</code> および <code>getResultSet</code> を呼び出した場合、既に閉じられた同じ結果セットが返されることはなくなりました。</p>
CR196738	<p>MSSQLServer4 ドライバの <code>Statement.cancel()</code> メソッドによって JDBC 接続にエラーが起きていました。接続が閉じられた後に JDBC 文がトランザクション内で閉じられていませんでした。</p> <p>MSSQLServer4 ドライバの <code>Statement.cancel()</code> メソッドが変更されて、ノーオペレーション命令となったため、接続でエラーが起きなくなりました。この変更はトランザクションのロールバックには影響しません。</p>



変更要求番号	説明
CR197163	<p>JDBC 接続プールの接続のテストでは、DBMS に毎回 SQL の解析とテストを要求する単純な文がテストのたびに新しく作成されていたため、大量のデータベース リソースを消費していました。</p> <p>プールは接続のテストで 1 つの Prepared Statement を再利用するようになったため、パフォーマンスが向上しました。ただし、アプリケーションの DBMS テーブルやプロシージャがテスト SQL で参照されていて、実行時に構造的に変更される場合（インデックスの追加など）、テスト用 PreparedStatement のクエリの実施が無効になる可能性があります。その結果、以降のテストが失敗して、接続とテスト文は置き換えられます。通常、Administration Console で指示されるテスト SQL には、構造的なテーブルの変更は含まれないので、接続の不必要な再利用の問題は最小限に抑えられています。</p>
CR203460	<p>XAConnectionFactory によってプールの接続が作成されるときに、その seconds-to-trust 値が設定されていない場合はデフォルトのままでした。</p> <p>現在は seconds-to-trust 値が設定されます。</p>
CR193357	<p>JDBC のリフレッシュ機能では、予約されていない接続の数を誤って数えていました。</p> <p>JDBC のリフレッシュ機能は予約されていない接続の数を正しく数えて、不要な処理をしなくなりました。</p>
CR087241	<p>XA 呼び出しと Oracle ドライバを使用する JDBC アプリケーションで、トランザクションがサスペンドされて再開されるときに行を取得する際に、「ORA-01002: フェッチ順序が無効です」エラーが発生していました。</p> <p>XA 呼び出しで Oracle 10g thin ドライバを使用する場合のこの問題を修正するために、AddOracleFlagToXAResource フラグが追加されました。XA 呼び出しを使用する場合に「ORA-01002: フェッチ順序が無効です」エラーを回避するには、Oracle 10g thin ドライバを使用して JDBCConnectionPool フラグを有効にする必要があります。</p>

# jDriver

変更要求番号	説明
CR142730、 CR190124、 CR190125	<p>長いデータベースの停止の後、TestConnectionsOnReserve が true に設定されている XA jDriver for Oracle を使用する JDBC 接続プールが、データベースとの接続を回復および再作成できていませんでした。以下のエラー メッセージが送出されていました。</p> <pre>&lt;Warning&gt; &lt;JDBC&gt; &lt;001096&gt; &lt;Refreshing this bad pool connection failed weblogic.common.ResourceException: java.sql.SQLException: open failed for XAResource 'oracleXAPool' with error XAER_RMERR: A resource manager error has occurred in the transaction branch.</pre> <p>および</p> <pre>&lt;Warning&gt; &lt;JDBC&gt; &lt;001096&gt; &lt;Refreshing this bad pool connection failed weblogic.common.ResourceException: java.sql.SQLException: LDA pool exhausted - make sure you call Connection.close()</pre> <p>その停止の間に、JDBC 接続プールが接続を再作成しようとしたが、失敗したときに、その接続の試行がクリーンアップされず、(LDA において) Oracle クライアントのリソースが枯渇していたことが問題でした。</p> <p>jDriver は、失敗した接続作成の試行をクリーンアップするようになりました。</p>
CR136168	<p>WebLogic jDriver は、負荷が大きい状態で Long raw 型を使用しているときに、Oracle エラー メッセージ ORA-02392 でサーバをクラッシュさせることがあります。</p> <p>修正コードを実装して、この問題を解決しました。</p>
CR172462	<p>AL32UTF8 文字セットが使用されているときに、WebLogic Server jDriver が Oracle 9.2 で正しく機能していませんでした。</p> <p>この問題は、コードを修正することで解決しました。</p>

# JMS

変更要求番号	説明
CR099554、 CR195519	環境によっては、JMS メッセージが保留状態にあるサーバが停止されるか、クラッシュした場合に、保留中のメッセージがサーバの再起動時に回復されませんでした。 保留中メッセージがサーバの再起動時に回復されるようになりました。
CR135578	非恒久トピックで、未処理の非恒久トピック メッセージが再配信遅延のためにまだ保留中のときに、コンシューマが終了した場合、このトピックの保留中バイト数の数字が正しく更新されていませんでした。 現在は、JMS の統計で、再配信遅延パラメータがコンフィグレーションされている場合にコンシューマが終了したとき、非恒久トピック メッセージの保留中バイト数は適切に調整されます。コンシューマが終了した場合、確認応答されていない非恒久トピック メッセージに対して再配信遅延は実施されません。
CR134155	トピック セッションからの JMS メッセージが、受信側がトランザクション セッションを使用して作成され、AUTO-ACKNOWLEDGE が使用され、メッセージがロールバックされた場合にメモリ リークを生じさせていました。 コードを修正してこの問題は解決されました。
CR178775	JMS クライアントが WebLogic Server に対して JMS 接続を繰り返し開いて閉じた場合、JMSConnection と DispatcherWrapper などの関連するオブジェクトはクライアントサイドとサーバサイドの両方で解放されませんでした。このため、OutOfMemoryError が発生しました。このエラーは、クライアントが JMS 接続を 1 つ開いて保持している間に、他の JMS 接続を開いて閉じると発生します。 コードの変更により、この問題は修正されました。
CR179737	競合状況が原因で、JMS が失われた接続を検出してクリーンアップできなくなる場合があります。 このような競合状況の発生を防ぐために再試行ロジックが強化されました。

変更要求番号	説明
CR173565	<p>XA トランザクションが失敗したときに、JMS メッセージが 2 度配信されていました。</p> <p>失敗したトランザクションを処理し、曖昧な状態のトランザクションがあることをユーザに通知するために適切なエラー メッセージをログに記録するように、コードが修正されました。</p> <p>この修正の前は、曖昧な状態にあるトランザクションのメッセージはすべてロールバックされ、メッセージがメモリにあるために再配信されていました。しかし、ストア内のレコード(ある場合)はそのレコードのストアのハンドルが無効であるために正しく更新されず、レコード(ある場合)はストアに残されていました。したがって、JMS サーバが再起動されたときに、メッセージは再配信されてメッセージの重複が生じていました。</p> <p>現在は、曖昧な状態にあるトランザクションのメッセージは、JMS サーバが再起動しないと回復できないようになりました。JMS サーバの再起動なしにメッセージを回復しようとすると、RMERR が生じます。</p>
CR183572	<p>JMS IOThreadPool 名が誤っていたため、WebLogic Server のすべての JMS ストアが同じ 2 つのスレッドを共有していました。その結果、3 つ以上の JMS ストアが同じ WebLogic Server 上で動作している場合、パフォーマンスが低下する可能性があります。</p> <p>JMS IOThreadPool 名が正しくなったため、パフォーマンスが低下する可能性はなくなりました。</p>
CR178405	<p>期限切れメッセージの調査時に、<code>java.util.ConcurrentModificationException</code> が送出されていました。</p> <p>この問題は、コードを修正することで解決しました。</p>
CR182338	<p>ConnectionFactory で RedeliveryLimit がコンフィグレーションされている場合、RedeliveryLimit の到達回数が MessagesMaximum の設定よりも大きくなると、レシーバはメッセージの処理を停止します。</p> <p>例 : RedeliveryLimit=0, MessagesMaximum=10。レシーバは 1 つのメッセージを受け取ると <code>session.recover()</code> を 10 回呼び出します。レシーバがメッセージの処理を停止すると、コンソールには保留中メッセージ数として 10 と表示されません。</p> <p>コードを追加して、RedeliveryLimit に達したためにキューからメッセージが削除される場合のウィンドウ カウンタを調整するようにしました。</p>

変更要求番号	説明
CR188040	<p>WebLogic JMS は ServerDebugMBean() の JMS 属性が変更されたときに通知を受け取っていませんでした。</p> <p>現在は、ServerDebugMBean() を使用して、ServerDebugMBean の「DebugJMS」属性を動的に有効または無効にできます。これによって、ユーザは JMS のデバッグ フラグを動的に有効または無効にできます。</p>
CR187945	<p>JMS DistributedDestination ロード バランサは、メンバーが動的に追加されたときに、既存のコンシューマを認識しませんでした。</p> <p>現在の WebLogic Server は、ロード バランサが適切にロード バランシングするために必要な情報を持てるように、分散送り先にメンバーが追加されるときに、メンバーがコンシューマを持つかどうかをチェックします。</p>
CR190438	<p>weblogic.jms.backend.BEQueue オブジェクトと weblogic.jms.backend.BEConsumer オブジェクトの間で、Java レベルのデッドロックが見つかりました。</p> <p>送り先が最初にロックダウンされるようにロックの階層構造を更新することで、この問題を解決しました。</p>

変更要求番号	説明
CR197857	<p>JMS は最小フロー率を保持していました。実際の最小フロー率はコンフィグレーションされた値の約 2 倍でした。接続ファクトリのデフォルトのフロー制御設定 (FlowMinimum=50) を使用する場合、JMS は 1 秒間に 50 メッセージではなく、約 100 メッセージの最小フローを保持していました。</p> <p>現在は、JMS 送り先で指定されたバイト数またはメッセージしきい値を超えた場合に、プロデューサに対するフロー制御が開始されると、フローは FlowMinimum (デフォルトは 50) にまで抑制され、しきい値を超えている間は、その最小フロー率が維持されます。</p> <p>この変更によって、JMS は最小フロー率を正しく保持するようになったため、以前よりもフローが大幅に抑えられました。</p> <p>以前のフロー制御の動作 (8.1 sp4 リリースよりも前の動作) を維持するには、FlowMinimum のコンフィグレーションを 2 倍にして ConnectionFactories に FlowMinimum=100 を追加します。この値は以前のデフォルト値を想定したものです。</p>
CR187610	<p>ブリッジは、ブリッジのディスパッチ リクエストに対して、メッセージングブリッジのスレッド プールを使用していませんでした。</p> <p>現在は、メッセージングブリッジのスレッド プールがコンフィグレーションされている場合、ブリッジはそれを使用します。メッセージングブリッジのスレッド プールにコンフィグレーションされている使用可能なスレッド数が十分でない場合、ブリッジはデフォルトの実行スレッド プールを使用します。</p> <p>ブリッジは、コンフィグレーションされているメッセージングブリッジのスレッド プール サイズが不十分な場合、警告メッセージをログに記録し、代わりにデフォルト実行プールからスレッドを取得します。</p>

## JNDI

変更要求番号	説明
CR197207	<p>EJB 呼び出しの応答で、JNDI ルックアップが静的ブロックから実行された場合に、Java クライアントがハングしました。</p> <p>問題を修正するため、新しいシステム プロパティ (-Dweblogic.rjvm.t3.dispatchOnCompleteMessage) が追加されました。このプロパティはクライアントサイドで有効にする必要があります。このプロパティを有効にすると、t3 リクエストはソケットリーダー スレッド以外のスレッドで処理されます。</p>

## JSP

変更要求番号	説明
CR087857	<p>JSP ページまたはサーブレットによって文字セット エンコーディングが設定された場合に、ServletOutputStream.write(int) メソッド (引数として int 型を取る) は指定された文字セット エンコーダを使用してエンコードされたデータを受け取りました。</p> <p>WebLogic Server は ServletOutputStream.write(int) が呼び出され場合はバイナリデータをエンコードしなくなりました。</p>
CR101992	<p>セッション内にローカルの EJBObject 参照を持つ Web アプリケーションで、再デプロイ後に javax.ejb.EJBException が発生することがありました。</p> <p>例外を捕捉してメッセージをログに記録するコードを追加しました。現在、このような状況では、シリアライゼーション エラーを示すエラー メッセージがログに記録されて、HttpSession、ServletRequest または ServletContext の getAttribute() が null を返します。</p>
CR174837	<p>WebLogic Server ではカスタムの PageContext 実装を許可していませんでした。</p> <p>生成された JSP に含まれていた weblogic.servlet.jsp.PageContextImpl への不要なキャストは削除されました。その結果、カスタムの PageContext 実装が使用できるようになりました。</p>

変更要求番号	説明
CR172380、 CR198902	<p>struts ベースのアプリケーションが EAR ファイルのプリコンパイルに失敗し、次の例外を送出していました。</p> <pre data-bbox="314 331 1166 496">&lt;Error&gt; &lt;Deployer&gt; &lt;149205&gt; &lt;The Slave Deployer failed to initialize the application dos due to error weblogic.management.ManagementException: 149233 - with nested exception: [java.lang.ExceptionInInitializerError]. java.lang.ExceptionInInitializerError: java.lang.NullPointerException</pre> <p>コードの修正によりこの問題は解決され、すべての JSP をプリコンパイルする前にスレッド クラス ローダが <code>ServletClassLoader()</code> メソッドに設定されるようになりました。これで、struts クラスと、後で struts ライブラリがロードしようとする他のクラスのロードで同じクラスローダが使用されるようになります。</p>
CR085091	<p>リクエストが保護されたリソースに対する条件付きの GET (Is-Modified-Since ヘッダを設定) である場合、カスタム エラー ページを提供できなくなる問題がありました。</p> <p>定義されている場合はカスタム エラー ページが提供されるようにロジックを修正しました。</p>
CR180425	<p>WebLogic Server はリクエストがプロキシから来ていない場合でも <code>wlsproxy</code> 固有のヘッダを送信していました。</p> <p>コードを追加して、リクエストがプロキシから来たかどうかをチェックして適切なヘッダを送信するようにしました。</p>



# JTA

変更要求番号	説明
CR130073	<p>アプリケーションが WTC を介して Tuxedo 7.1 から WebLogic Server に複数のロールバックを発行すると、一部の Oracle XA 接続がリークし、プールに戻されていませんでした。XA Debug では、XA End が呼び出されないために接続がクリーンアップされないことが示されていました。</p> <p>XA End を呼び出すようにすることで、接続がプールに戻されるようになり、メモリ不足エラーは発生しなくなりました。</p>
CR125476	<p>トランザクションが準備フェーズにあり、XAER_RMERR または XAER_RMFAIL が送出された場合に、トランザクションがロールバックされずに保留されたままになりました。</p> <p>このような状況でトランザクションがロールバックされるようになりました。</p>
CR184941	<p>サーバをバックアップするため手動で JTA を移行した後に、クラッシュした WebLogic Server を起動しようとしても起動できませんでした。</p> <p>サーバをバックアップするため手動で JTA を移行した後も、クラッシュした WebLogic Server を再起動すると起動できるようになりました。</p>
CR202386	<p>トランザクション マネージャは、新しいリソース マネージャに関連付けられたオブジェクトの <code>xa_start()</code> の呼び出しに失敗していました。</p> <p>この問題を修正するために、デフォルトの登録が標準化されました。アプリケーションが <code>registerResource()</code> を呼び出さない場合、リソースは標準のリソースとしてトランザクション マネージャに登録されます。</p>
CR188558	<p>タイマー スレッドがオフになる前に WebLogic Server を停止すると、一部のリソースのチェックポイントが TLOG に記録されていませんでした。WebLogic Server を再起動しても、これらのリソースの回復処理は実行されず、保留になっていたトランザクションは保留のままになっていました。</p> <p>リソースがローカルで調整され、かつリソースのチェックポイントを記録する必要がある場合は、トランザクションをリソースに登録するとフラグが立てられ、トランザクションが 2 フェーズであれば準備時間にリソースのチェックポイントが TLOG に記録されるようになりました。</p>

---

変更要求番号	説明
CR173464	キャッシュされているコーディネータを取得するときに、スレッドは早期通知状態に入らなくなりました。以前はこのためにサーバがハングしていました。
CR211194	MQSeries リソースの場合は、XA.end (TMSUSPEND)、XA.end (TMFAIL)、XA.rollback という順序の XA 呼び出しシーケンスを呼び出して、MQSeries でメモリ リークが発生しないようにしています。

---

## JVM

---

変更要求番号	説明
CR132228、 CR188670	問題のない IOExceptions が日本語ロケールで抑制されていませんでした。 現在、WebLogic Server はネイティブ IO を有効にする場合、デフォルトではロケールを C に設定します。ロケールを C に設定しない場合は、以下のシステムプロパティを使用できます。 <code>-Dweblogic.nativeIO.useDefaultLocale=true</code> 英語以外のロケールでは、問題のない例外が抑制されます。

---

## ノード マネージャ

---

変更要求番号	説明
CR135917	stderr に対する fdopen が失敗した場合に、NodeManager のネイティブ コードが原因で、セグメンテーション違反が起きることがありました。この問題は修正されています。

---

変更要求番号	説明
CR076968	オフライン メソッドが呼び出された場合、プロセス終了信号として SIGTERM ではなく SIGKILL を連続して送信するように、NodeManager を更新しました。
CR128517	Administration Console を使用してマシンをコンフィグレーションし、NodeManager をコンフィグレーションしない場合に、サーバの状態を問い合わせると、デフォルトでは NodeManager に対して問い合わせが行われます。 NodeManager が必要ない場合は、マシン コンフィグレーションで ListenPort を 0 に設定してください。これにより、NodeManager がコンフィグレーションされていないことがサーバに伝わり、サーバは NodeManager への問い合わせを行わなくなります。

## 操作と管理

変更要求番号	説明
CR132947	ドメイン内の管理対象サーバの停止後にそのドメインの管理サーバが停止すると、管理サーバから常に java.rmi.ConnectException が送出されていました。 この問題は修正されています。
CR187170	配列型の属性が変更されたとき（配列の要素が追加または削除されたとき）に、AttributeAdd/Remove 通知が送信されませんでした。そのため特定のデプロイメントが適切に実行されていませんでした。 現在 WebLogic Server では AttributeChange 通知に加えて、配列型の属性値が変更されたときに AttributeAdd/Remove 通知が送信されるようになっています。
CR108496	コンソールで EJB デプロイメント記述子を編集すると、デプロイされた EJB jar ファイルで <global/> タグが <global>true</global> に変更されていました。 getGlobalRole メソッドの空のタグが追加され、永続化後も検証が失敗しなくなりました。

変更要求番号	説明
CR126191	<p>weblogic.Admin ツールで DESTROY_POOL コマンドを使用した場合、単にすべてのユーザの接続が切断されてプールがサスペンド状態になるだけでした。ドメイン コンフィグレーションからプールが削除されるはずでしたが、実際にはそうではありませんでした。</p> <p>DESTROY_POOL により、プールのコンフィグレーションが削除されるようになりました。プールをサスペンドするには、DISABLE_POOL または SUSPEND を使用してください。</p>
CR134167	<p>EXISTS_POOL が予期されたとおりに機能していませんでした。このコマンドの検索対象は実行時 MBean でしたが、これは、プールが存在している場合であっても、存在したりしなかったりする MBean でした。文書化されていないコマンド DELETE_POOL は、接続プールを削除するための内部実装ですが、この機能が外部的には DESTROY_POOL として文書化されていました。</p> <p>またヘルプ メニューにも、いくつか文書化されていないコマンドについての表示がありました。</p> <p>EXISTS_POOL コマンドの実装は、プールの存在確認の検索対象としてコンフィグレーション MBean を使用するようになりました。</p> <p>DESTROY_POOL コマンドは、基底の DELETE_POOL 実装を利用するようになりました。</p> <p>weblogic.Admin のヘルプ メニューに表示されていたコマンドで、文書化されていないものは、現在すべて無効化されています。無効化されたコマンドは、TEST_POOL、REMOVE_POOL、SUSPEND_POOL、SHUTDOWN_POOL、RESUME_POOL、DELETE_POOL などです。これらのコマンドは今後サポートされなくなり、7.0 SP5 では機能しなくなる予定です。</p> <p>ヘルプ メニューにも表示されなくなっています。</p>
CR171906、 CR179030、 CR208205	<p>Operator ロールに属するユーザによって管理対象サーバが実行されている場合に、管理サーバが起動しませんでした。</p> <p>Operator ロールのユーザに対し、管理サーバの再起動および管理対象サーバへの再接続の操作が許可されました。その結果、現在では Operator ロールに属するユーザによって管理対象サーバが実行されていても、管理サーバを起動できるようになっています。</p>

変更要求番号	説明
CR179524、 R203851、 CR213673	<p>サーバが実行されているにも関わらずサーバ ログをエディタまたはビューアで開いていると、ログがフリーズしました。これは、サーバがログ ローテーションの最中またはその後、ログの記録のためにログ ファイルを再度開くことができなかったためでした。</p> <p>ロガーに再試行ロジックを実装し、最初の試行が失敗した場合には代替りのログ ファイルへのローテーションが試行されるようにしたため、すべての試行が失敗してもログの記録が続行されるようになりました。</p>
CR177510 CR135356	<p>大きな EAR ファイル (600MB) の一部であるモジュールの追加または再デプロイには、非常に長い時間がかかっていました。</p> <p>Web アプリケーション モジュールのコードを変更した結果、パフォーマンスが向上しました。</p> <ul style="list-style-type: none"> <li>現在、Web アプリケーション モジュールでは、モジュール全体を変更するような差分のある場合にモジュールの更新が考慮されるようになっていません。</li> <li>大きなアプリケーションをデプロイしたときの起動時間を短縮するため、コンフィグレーションされているアプリケーションのステージングされた Targets 属性を保持するように、デプロイメント方法が変更されました。</li> </ul> <p>アプリケーションのデプロイメント中に発生する、管理サーバおよび管理対象サーバ間のリモート呼び出しのトラフィックが低減されました。</p>
CR182686	<p>起動クラスまたはサーブレットでシステム プロパティを設定しているときに、<code>ConcurrentModificationException</code> が送出されることがありました。</p> <p>プロパティの同期をとるように変更した結果、システム プロパティの設定時に <code>ConcurrentModificationException</code> が送出されなくなりました。</p>
CR184787	<p>Administration Console において、デプロイメント記述子のセッション パラメータに関するページで、<code>PersistentStoreType</code> の「<code>replicated_if_clustered</code>」値が表示されませんでした。結果として記述子が永続化されたときに、その記述子に違う値が設定されていました。</p> <p>Administration Console でこの値が表示され、正しく記述子に追加されるようになりました。</p>

変更要求番号	説明
CR187170	<p>配列型の属性が変更されたとき（配列の要素が追加または削除されたとき）に、AttributeAdd/Remove 通知が送信されませんでした。そのため特定のデプロイメントが適切に実行されていませんでした。</p> <p>現在 WebLogic Server では AttributeChange 通知に加えて、配列型の属性値が変更されたときに AttributeAdd/Remove 通知が送信されるようになっています。</p>
CR190237	<p>WebLogic Portal 環境で実行する場合に WebLogic Server MbeanHome の getAllMBeans() メソッドが例外を送出していました。</p> <p>WebLogic Server MBeanHome Helper では WebLogic Portal MBean クラスの場所を示す追加情報が必要であり、既存のロジックを修正する必要がありました。</p> <p>WebLogic Portal 環境で ClassNotFoundException 例外が送出されなくなりました。</p>
CR188701	<p>管理サーバが停止しているときに管理対象サーバを再起動すると、デプロイメントに失敗していました。サーバによりアプリケーションがステージングされる際に、そのデプロイ対象であるアプリケーションにステージングモードのセットがない場合、誤ったステージングモードが使用されていました。</p> <p>ステージングモードのセットを持たないが、ステージングモードセットを使用してデプロイされるアプリケーションについても、正しいステージングモードで適切にデプロイメントされるようになりました。</p>
CR190822	<p>JDBC 接続プールのクローン作成処理が機能していませんでした。</p> <p>サーバが対象とされている場合にも、JDBC 接続プールのクローン作成処理が機能するようになりました。</p>
CR200310	<p>MSI モードで実行すると、アプリケーション マネージャが 2 度起動されていました。</p> <p>MSI モードで実行しても、アプリケーション マネージャが 1 度だけ起動するようになりました。</p>
CR135521	<p>Commo MBean を作成または削除する際に、IllegalArgumentException が送出されなくなりました。</p> <p>現在では、タイプが null の場合には InstanceNotFoundException が送出されます。wlconfig などのクライアントでこの例外が捕捉され、JMX MBean Home を使用して MBean の属性値が取得されます。</p>

変更要求番号	説明
CR182072	<p>config.xml ファイルの ConfigurationVersion 属性の値が、サービス パックの更新後に変更されるようになりました。</p> <p>現在、この属性は、サーバが起動されるたびに管理サーバと同じバージョンに設定されます。</p>
CR100218	<p>ExecuteQueueMBean.setThreadPriority() の値はユーザからコンフィグレーションできないため、パブリック javadoc から削除されました。</p>
CR109591	<p>Administration Console で</p> <p>'myserver_ %yyyy%_%MM%_%dd%_%hh%_%mm%.log というログの場所を見つけられませんでした。</p> <p>この問題を修正するため、現在では次のような形式をもとにファイル名の検索を試行するようになっています。</p> <pre>myserver_YYYY_MM_dd_hh_mm.log</pre> <p>次に示す形式は使用されません。</p> <pre>myserver_%YYYY%_%MM%_%dd%_%hh%_%mm%.log</pre> <p>myserver_YYYY_MM_dd_hh_mm.log 形式を使用してもファイル名を検索できない場合には、FileNotFoundException が送出されます。</p>
CR124169	<p>JMX を使用して MessagingBridge を停止またはアンデプロイするときに、アサーション エラーが送出されることはなくなりました。</p>
CR202449	<p>管理サーバ上にサーバインスタンスが作成される際に、そのデフォルト キューが管理サーバの再初期化時まで初期化されていませんでした。</p> <p>サーバインスタンスの作成時またはクローン作成時に、デフォルト キューが初期化されるようになりました。</p>

変更要求番号	説明
CR208697	<p>removeNotification() を呼び出して複数の中間要素を削除するときに、WebLogic Server では notificationID 変数がデクリメントされていました。その後、allNotification[] の最後の要素に対してアクセスが試行されると、InstanceNotFoundException が送出されていました。</p> <p>notificationID 変数の値が allNotification[] のサイズよりも小さくなっていたため、結果として最後の要素に対するアクセスが不可能でした。</p> <p>コードの変更によりこの問題は修正され、removeNotification() が呼び出されたときに ArrayOutOfBoundsException が送出されなくなりました。さらに、InstanceNotFoundException は通知インスタンスが利用不可能な場合にのみ送出されるようになりました。したがって、allNotification[] の最後の要素に対してアクセスが試行された場合に、InstanceNotFoundException が送出されることはなくなりました。</p>
CR211194	<p>XA 呼び出し手順 XA.end (TMSUSPEND) に続いて XA.rollback が呼び出された場合に、MQSeries でメモリ リークが発生することはなくなりました。</p>
CR211815	<p>MBean の登録解除プロセスで使用されるロックのシーケンスは、デッドロックを回避するために修正されるようになりました。</p>

## プラグイン

変更要求番号	説明
CR095984	<p>NSAPI プラグインの使用時にファイル サイズが 30KB を超えたとき、413 メッセージではなく、IE 6.0 の DNS エラー メッセージが受信されていました。</p> <p>MaxPostSize コンフィグレーションの動作が、プラグインを使用しているかどうかに関わらず同じになりました。</p>
CR100070	<p>プラグインが処理するすべてのリクエストに対してデバッグの有効化または無効化を決定するグローバル デバッグ フラグが 1 つしかなかったため、Apache-WLS プラグイン内の各仮想ホストの WLLogFile を管理するのは困難でした。</p> <p>デバッグ オプションを一番上のレベルで指定し、各仮想ホストまたは location タグの内部で上書きできるようになりました。</p>



変更要求番号	説明
CR136968	WebLogic Server では、response.addHeader メソッドが使用される場合に複数のヘッダを受け付けていませんでした。 プラグインで WWW 認証の際に複数の値が許可されるようになりました。
CR171978	iisforward.ini ファイルで FilterPriorityLevel が設定されている場合に、転送パスが途切れていました。 修正コードの実装により、iisforward.ini ファイルで仮想ホストが定義されていない場合は、iisforward.dll ファイルのロード先と同じ場所にある iisproxy.ini ファイルが使用されるようになりました。
CR172072	WlExcludePathOrMimeType パラメータの機能が拡張され、Location タグレベルでの使用が可能になりました。 現在 WlExcludePathOrMimeType パラメータは、グローバルで定義するだけでなく、Location タグレベルでローカルにも定義できます。このプロパティをローカルに定義しても、グローバル プロパティはオーバーライドされませんが、2つのパラメータの結合が定義されることとなります。
CR173581、 CR173878	プラグインにより、クライアントが接続を閉じている場合にも、「error page is unavailable」という紛らわしいログメッセージが apache error.log に記録されていました。 この誤ったログメッセージをコメントアウトするようにコードを変更することで、この問題は解決しました。
CR174431	Iplanet プラグインが EINTR OS エラーを正常に処理するようになりました。
CR175672	Apache サーバは、WebLogic プラグインが wlproxy ログ ファイルを開こうとすると、Debug が OFF に設定されている場合にもハングしていました。 デバッグが無効になっている場合はログ ファイルが設定されないように、コードが修正されました。
CR175989	prefork オプション (マルチプロセスのみ) ではなく worker オプション (マルチスレッド) を使用している場合に、Apache サーバではコアダンプが生成されていました。 ロックおよびロック解除のロジックを修正することで、この問題は解決されました。

変更要求番号	説明
CR177707	<p>リリース 7.0 SP02 のプラグインとクライアント証明書を一緒に使用した場合、WebLogic Server は正常に動作していました。しかし、リリース 7.0 SP06 へのアップグレード後、サーバ ログに以下のエラーが報告されました。</p> <pre data-bbox="314 366 1089 475">Failed to parse the client certificate in header: WL-Proxy-Client-Cert. Ignoring this certificate. java.security.cert.CertificateException: Could not parse certificate: java.io.EOFException</pre> <p>このエラーは、7.0 SP06 プラグインがサーバ インスタンスへの送信時に WL-Proxy-Client-Cert ヘッダを切り詰めていたために発生していました。WebLogic Server に送信されるリクエストに WL-Proxy-Client-Cert が切り詰められていない状態で追加されるように、コードを変更しました。</p>
CR179537	<p>Microsoft Windows プラットフォームで、IIS プロキシ プラグインによりヒープの破損が発生していました。</p> <p>この問題は、内部コードを修正することで解決しました。</p>
CR180236	<p>7.0 の以前のリリースのプラグインとクライアント証明書を一緒に使用した場合、以下のエラーが報告されていました。</p> <pre data-bbox="314 843 1089 953">Failed to parse the client certificate in header: WL-Proxy-Client-Cert. Ignoring this certificate. java.security.cert.CertificateException: Could not parse certificate: java.io.EOFException.</pre> <p>このエラーは、プラグインが WebLogic Server インスタンスへの送信時に WL-Proxy-Client-Cert ヘッダを切り詰めていたために発生していました。</p> <p>この問題は、コードを修正することで解決しました。</p>
CR178792	<p>HTTP リクエストには Content-Length または Transfer-Encoding のいずれかのヘッダを指定できます。</p> <p>Transfer-Encoding ヘッダを「chunked」に設定したリクエストが IO エラーで失敗していました。</p> <p>Transfer-Encoding ヘッダを「chunked」に設定したリクエストをサポートするコードが追加されました。</p>

変更要求番号	説明
CR180560	WebLogic Server に対する新しい接続の作成時に、プラグインではソケットの情報 (localhost:localport remotehost:remoteport) をログ ファイルに出力していませんでした。 プラグインによって WebLogic Server に対する新しい接続が作成されるときに、ローカル ホスト名とローカル ポート番号を伴ったログ メッセージが追加されるようになりました。
CR180417	POST データにクッキーが含まれている場合、クッキーの抽出中にプラグインによって POST データが破損されていました。 POST データからのクッキー抽出を修正するコードを追加しました。
CR180724	最初のクッキーが Web サーバ 1 から作成されてクラスタ 1 に送信されたとします。そのクッキーが再びアプリケーションにヒットした場合、クッキーが Web サーバ 2 に送られ、クラスタ 1 ではなくクラスタ 2 に送信されて、新しいセッションが作成されていました。 現在の WebLogic Server では WLCrossOverEnabled 機能が正しく動作するようになっています。
CR182434	rq->srvhdrs に渡されたヘッダが、完全に小文字になっておらず、大文字と小文字が混在した状態でした。 Content-type、content-length および transfer-encoding ヘッダは現在、完全に小文字で NSAPI に渡されるようになっています。
CR182971	DNSRefreshInterval ごとに ServerList が削除されていたため、コア ダンプが発生していました。 リスト内のすべてのサーバについて DNS ルックアップが行われ、最後にチェックした後に変更されたサーバがあった場合に、ServerInfo の構造が更新されるようになりました。
CR183188	ISAPI プラグインでは Transfer-Encoding ヘッダが chunked に設定されているリクエストを処理できませんでした。 機能の追加により、ISAPI でそのようなリクエストを処理できるようになりました。

変更要求番号	説明
CR183311	<p>シングルスレッド マルチプロセス モジュールを使用しているときに Apache が停止する場合、最初にタイマー スレッドの停止が試行されていました。このタイマー スレッドは存在していないため、コア ダンプが発生していました。</p> <p>Apache がシングルスレッド マルチプロセス モジュールと共に使用されている場合に、タイマー スレッドが作成されなくなりました。</p>
CR183390	<p>WebLogic Server では例外を catch ブロックの中から送出しており、そのために iPlanet が失敗することがありました。</p> <p>例外は catch ブロックから送出されなくなりました。</p>
CR185089	<p>IIS プラグインでは、クライアントが接続を閉じたために WRITE_ERROR_TO_CLIENT 例外が発生すると、HTTP ステータス コード 500 (内部サーバエラー) を送信していました。</p> <p>WRITE_ERROR_TO_CLIENT 例外が捕捉された場合にも、IIS プラグインから HTTP ステータス コード 500 が送信されなくなりました。</p>
CR185668	<p>Apache プラグインを使用し、MatchExpression を使用して複数のクラスタにプロキシする場合、リクエストの転送に使用される URL のセグメントが PathTrim 属性によって削除されていませんでした。</p> <p>strtok API を使用しないで MatchExpression 解析を実装しなおすことで、この問題は修正されました。</p>
CR186148	<p>Apache プラグインではページが見つからなかったときに、適切に 503 エラーを返すのではなく、500 エラーを返していました。</p> <p>プラグインから適切なエラーが返されるようになりました。</p>
CR186470	<p>IIS プラグインを使用しているときに、ファイアウォールを介して多数の新しい接続を作成すると、HTTP ステータス 302 が発生し、接続が閉じていました。</p> <p>HTTP ステータス コードが 302 の場合に、接続が再利用されるようになりました。</p>
CR187282	<p>HTTP 1.1 仕様では、リクエストまたは応答に Content-Length ヘッダと Transfer-Encoding: Chunked ヘッダの両方が含まれている場合、Content-Length ヘッダを無視する必要があると規定されています。プラグインがこの規定にしたがっていなかったため、プールの接続の再利用を伴う特定のシナリオで、エラーが発生することがありました。</p> <p>CTE が有効になっている場合、contentLength が -1 として返されるようになりました。</p>

変更要求番号	説明
CR187577	<p>VirtualHost タグで複数の Location タグが使用されている場合、リクエストに一致する場所が複数あると、Apache プラグインでは誤った URL が生成されていました。</p> <p>指定されない限り、Apache プラグインで正規表現マッチングが使用されなくなりました。</p>
CR187578	<p>httpd.conf で KeepAliveEnabled をオンにコンフィグレーションすると、KeepAliveSecs はデフォルトで 20 に設定され、このデフォルト設定を変更することはできませんでした。</p> <p>コードの追加により、KeepAliveSecs がコンフィグレーション可能になりました。</p>
CR188811	<p>リクエストにクエリ文字列がある場合、WLExcludePathOrMimeType が正しく機能しませんでした。</p> <p>http://webserver:port/weblogic/something.jsp?value=123 のようなリクエストが除外されず、http://webserver:port/weblogic/something.do?name=test.jsp のようなリクエストが転送されませんでした。</p> <p>プラグインで、除外のチェックをするときにクエリ文字列が無視されるようになりました。</p>
CR189251	<p>負荷がかかっている状況で、仮想ホストのプラグイン プロパティを取得しているときに、セグメンテーション エラーが発生していました。</p> <p>strtok API はスレッドセーフではないため、strtok API を strchr で置き換えることでエラーを解消しました。</p>
CR190562	<p>Solaris 上で、WebLogic Server に post データを送信しているときに、プラグインで破損パイプ エラーが発生すると、リクエストが再試行されませんでした。</p> <p>Solaris 上で sendPostData によりパイプの破損が報告された場合、HALF_OPEN_SOCKET_RETRY 例外が送出されるようになりました。</p>
CR189933	<p>WebLogic Server プラグインはスレッド セーフではありませんでした。SvrInfo 配列のメモリ アドレスとそのサイズが他のスレッドにも渡されており、他のスレッドで修正することが可能でした。他のスレッドで修正された場合、最初のスレッドは無効なメモリ位置にアクセスすることになり、セグメント エラーが発生するおそれがありました。</p> <p>WebLogic Server プラグインがスレッド セーフになり、さらに WebLogic Server によって、他のスレッドに渡す前に SvrInfo 配列の読み込み専用コピーが作成されるようになりました。</p>

変更要求番号	説明
CR191552	<p>非推奨になったプロパティ <code>MaxSkips</code> が <code>MaxSkipTime</code> に置き換えられていました。しかし、この新しいプロパティはコード全体で使用されていなかったため、<code>MaxSkipTime</code> パラメータのデフォルト値 10 を変更できませんでした。</p> <p><code>MaxSkipTime</code> がコード全体で使用されるようになり、この値を変更できるようになりました。</p>
CR192875	<p><code>readPostDataIntoFile()</code> の <code>try/catch</code> ブロックから新しい例外が送出された場合に、<code>iPlanet</code> サーバがクラッシュしていました。</p> <p><code>readPostDataIntoFile()</code> が、一時ファイルへの <code>post</code> データの書き込み中にエラーに遭遇した場合、例外を送出するのではなく、例外を返すようになったため、この問題は発生しなくなりました。</p>
CR193447	<p>クッキーに、提示されたプライマリとセカンダリの情報を分ける区切り記号「 」が含まれていませんでした。その結果、<code>parseJVMID()</code> で常にプライマリ サーバの情報が返され、セカンダリ サーバの情報が無視されていました。</p> <p>プライマリとセカンダリの情報を分けるためにクッキーがトークン化され、抽出された両方の値に対して <code>parseJVMID()</code> を呼び出すように修正されました。これにより、<code>parseJVMID()</code> でプライマリ サーバとセカンダリ サーバの両方の情報が返されるようになりました。</p>
CR193985	<p>「creating timer thread in child」メッセージが、ログ レベル <code>[warn]</code> で記録されていました。</p> <p>これらのメッセージがログ レベル <code>[info]</code> で記録されるように修正され、メッセージの本質をよりの確に反映するようになりました。</p>
CR194141	<p><code>getRequestURI</code> が戻り値をデコードすべきでないことを示している場合にも、<code>Apache</code> プラグインが <code>WebLogic Server</code> にリクエストを渡す前に <code>URI</code> をデコードしていました。</p> <p><code>Apache</code> プラグインが、<code>URI</code> をはじめにデコードせずに <code>WebLogic Server</code> に渡すように修正されました。</p>
CR194464	<p>内部の <code>initJVMID()</code> の呼び出しでは、接続が拒否された場合にもサーバの状態が <code>GOOD</code> から <code>BAD</code> に更新されていませんでした。その結果、すでに停止しているサーバにアクセスしようとして時間が浪費されていました。</p> <p>サーバがすでに <code>BAD</code> とマークされている場合には、<code>initJVMID()</code> がそのサーバをスキップして次のサーバに移るよう修正されました。また、接続が拒否された場合に、<code>initJVMID()</code> がサーバの状態を更新するようになりました。</p>

変更要求番号	説明
CR199045	<p>SSL および非 SSL コンフィグレーションを使用してさまざまなパス値が複数のオブジェクト タグでコンフィグレーションされていると、プラグインが SSL リクエストと非 SSL リクエストを正しく切り替えられませんでした。</p> <p>グローバルフラグ (convertDNSToIP) をリクエストごとのフラグ (ConfigInfo 構造内に移動) で置き換えたことで、プラグインが SSL リクエストと非 SSL リクエストを正しく切り替えられるようになりました。</p>
CR201736	<p>ISAPI プラグインを使用している場合に、WL-Proxy-SSL ヘッダが欠落しないようになりました。</p>
CR199446	<p>Apache プラグインでは、Location タグと LocationMatch タグで正規表現がサポートされていませんでした。</p> <p>Apache プラグインで POSIX 正規表現がサポートされるようになりました。</p> <p>次のような Location タグは、今後は無効です。</p> <pre data-bbox="377 725 579 821">&lt;Location */app/*&gt; ... &lt;/Location&gt;</pre> <p>このタグは、次のように変更する必要があります。</p> <pre data-bbox="377 873 552 977">&lt;Location /app/&gt; ... &lt;/Location&gt;</pre> <p>以下に、Location タグで一般的に使用するエン트리について示します。</p> <p>リクエストが特定のパスに一致する場合：</p> <pre data-bbox="377 1072 440 1098">/app/*</pre> <p>これは、次のように変更する必要があります。</p> <pre data-bbox="377 1150 431 1177">/app/</pre> <p>リクエストが特定のパスで始まる場合：</p> <pre data-bbox="377 1229 440 1255">/app/*</pre> <p>これは、次のように変更する必要があります。</p> <pre data-bbox="377 1307 440 1333">^/app/</pre> <p>リクエストが特定の MIME タイプに一致する場合：</p> <pre data-bbox="377 1385 444 1411">*/*.jsp</pre> <p>これは、次のように変更する必要があります。</p> <pre data-bbox="377 1463 417 1489">\.jsp</pre>

変更要求番号	説明
CR199668	ホスト名のマッチングで大文字と小文字が区別されていたため、ホスト検索時に「HostName」と「hostname」がマッチングしませんでした。  iisforward.ini を使用したプロキシのホスト検索で大文字と小文字が区別されなくなりました。これにより、ホスト名のマッチングでも、大文字と小文字が区別されなくなりました。
CR202898	Apache プラグインに HEAD リクエストが送信された場合に、HttpResponse のヘッダから Content-Length ヘッダが欠落しなくなり、Content-Type ヘッダが壊れることもなくなりました。
CR205009	ルート ユーザが Apache プラグイン ( リビジョン 158 ) を起動すると、コア ダンプが発生する問題が修正されました。
CR209778	ConnectRetrySecs で、カスタム値がコンフィグレーション ファイルで指定したとおり使用されるようにコードが修正されました。  現在 ConnectRetrySecs は、Apache と正しく連携するようになっています。
CR209383	FileCaching が OFF に設定されているにもかかわらず、POST データがチャンク化される問題が解決しました。
CR207694	リクエスト中に「Expect: continue-100」が見つかった場合、「HTTP/1.1 100 Continue」が NSAPI による応答の本文ではなくヘッダに格納されるようになりました。
CR205760	リクエストの読み込み中に INSUFFICIENT_BUFFER エラーが送出された場合、新しいメモリが割り当てられましたが、メソッド内でこのメモリが解放された後に、このメモリに対するアクセスが行われていました。  メモリの解放がルーチンの最後に行われるように修正されました。
CR199080	プラグインが WebLogic Server への接続に失敗した場合、GETLASTERROR がログに記録されるようになりました。  また、プラグインが WebLogic Server に接続している間に発生したすべてのエラーについても、ログ メッセージとして記録されます。



変更要求番号	説明
CR128730	<p>HttpClusterServlet が再利用された接続を使用してリクエストをプライマリに送信する場合、HttpClusterServlet は接続にエラーがあるかどうかについて、再接続が可能な時間内に判断できませんでした。その結果、HttpClusterServlet はセカンダリへのフェイル オーバーを行っていませんでした。</p> <p>HttpClusterServlet で、リクエストが正常にプライマリに送信されたかどうかのチェックが行われるようになりました。さらに、プライマリへのリクエストが正常に送信されず、かつ使用した接続が再利用されたものだった場合、HttpClusterServlet はプライマリに対して新しい接続の作成を試行するようになりました。後続のリクエストも失敗して例外が送出された場合には、HttpClusterServlet によってセカンダリへのフェイル オーバーが行われます。</p>
CR208303	<p>AIX では、errno 変数がスレッド セーフではありませんでした。errno 変数をスレッド セーフにするため、Makefile.aix に <code>-D_THREAD_SAFE</code> が追加されました。</p>
CR218680	<p>リクエストごとに、HTTP ステータス コードが正しく設定されるようになりました。</p>

# RMI

変更要求番号	説明
CR177353	<p>アプリケーション クライアントのコードで、リモート スタブをキャッシュし、クラスタにデプロイされているステートレス セッション Bean のリモート メソッドを呼び出す場合、リモート オブジェクトが使用可能であるクラスタ ノードを追跡したリストが、呼び出しのたびにリフレッシュされていました。このリストは、いずれかのノードが回復可能な例外で失敗した場合に、呼び出しをフェイルオーバーするために使用されていました。ここで問題になっていたのは、アプリケーション クライアントが以前の呼び出しでスタブをキャッシュしていて、クラスタ全体が再起動された場合に、フェイルオーバーが動作しないということでした。</p> <p>フェイルオーバーの再試行ロジックが適切ではありませんでした。元のロジックでは <math>n</math> 個のノードを持つクラスタで <math>n-1</math> の再試行回数を許可していましたが、クラスタ全体を再起動したときキャッシュ済みのスタブは古いリストを保持している状態で、古いリストを調べているうちに、すべての再試行回数が消費されていました。そしてその上で、ロジックの最後の試行でリストのリフレッシュが行われる場合があります。たとえクライアント サイド スタブにリストの新しいコピーがあっても、既に <math>n-1</math> 回の試行制限に達しているため、フェイルオーバーが行われることはありませんでした。</p> <p>アプリケーション クライアントにキャッシュされたりリモート スタブが、既存のリストにあるすべてのノードでリモート メソッド呼び出しが失敗した場合にのみ、リストをリフレッシュするようになりました。また、リストがリフレッシュされた場合、スタブには最後に 1 度フェイルオーバーの機会が与えられ、これに失敗した場合にはスタブからアプリケーションに例外が送出され、失敗しなかった場合にはフェイルオーバーが透過的に動作し続けるように修正されています。</p>
CR188748	<p>ゲッターおよびセッター操作でリポジトリ ID を生成するときに、ゲッターまたはセッターに予約されたキーワードが含まれていると余分な「_」が追加されることがあり、それによってリモート呼び出しが失敗することがありました。</p> <p>余分なアンダースコアを削除するように修正されました。</p>

変更要求番号	説明
CR197396	サーバ インスタンスがクライアントへの IIOP ( 発信 ) 応答のために addIndirection を呼び出す間に、マシンに搭載されている CPU の使用率が 100 % に達することがなくなりました。 その結果、パフォーマンスが向上しています。
CR197824	WebSphere 5.1 から WebLogic server に対して EJB 呼び出しが試行されるときにデータとして HashMap が使用されていると、WebLogic server から UNMARSHAL 例外が送出されていました。 この問題を解決するため、WebLogic Server で外部サービス コンテキストが適切に処理されるようになりました。

## サンプル

変更要求番号	説明
CR132509	MedRec アプリケーションに含まれている .NET C# サンプル (%WEBLOGIC_HOME%\samples\server\medrec\src\clients\CSharpClient\bin\ReleaseCSharpClient.exe) を使用した場合、クライアントで患者のレコードは正常に取得されていましたが、クライアントアプリケーションから「Save Changes」を行おうとすると、日付フィールド エラーが示されました。 コードの変更により日付の検証が修正されました。
CR133641	iPlanet を使用している場合に、ホスト名検証で問題が発生して、以下のメッセージが受信されていました。 INFO: Host () doesn't match (), validation failed ERROR: SSLWrite failed コードを修正してこの問題は解決されました。

## Security

変更要求番号	説明
CR121646	<p>クライアント証明書が Extended Key Usage を critical に設定した状態で WebLogic Server に送信されると、BAD_CERTIFICATE エラーが受信され、SSL 接続が終了していました。</p> <p>EnhancedKeyUsage を追加サポートすることで、この問題は解決されました。現在 WebLogic Server では Enhanced Keyusage を critical に設定した状態のクライアント証明書が受け付けられるようになっています。</p>
CR126837	<p>iPlanetAuthenticator MBean で使用される listGroupMembers 実装を介して、ユーザのリストを取得する場合に問題がありました。具体的には、listGroupMembers() メソッドが空のカーソルを返していました。</p> <p>StaticMemberDNAttribute のデフォルト値を「uniquemember」に変更することで、この問題は解決しました。</p>
CR137523	<p>ファイル レルムが refresh() メソッドを呼び出しているときにユーザを定義すると、同期化の問題が発生してユーザ テーブルが壊れていました。</p> <p>ファイル レルムがリフレッシュを実行している間にユーザを定義した場合にも例外が発生されることのないように、ファイル レルムの機能が向上しました。</p>
CR180972	<p>WebLogic Server では送信リクエストごとに新しい SSLSocketFactory を作成していたため、発信 SSL 接続のパフォーマンスが JSSE と比べて低速化していました。</p> <p>リクエストに証明書がない場合には、SSLSocketFactory がキャッシュされるようになりました。</p>
CR182860、 CR187047、 CR194416、 CR214190、 CR214191、 CR214192	<p>Web ブラウザからセキュア ポートを介してプレーン テキストを送信した場合には、SSL がメッセージ ヘッダを誤って解釈し、メッセージの終了前にさらにデータがあるとして待機していました。この遅延により、SSL ハンドシェイク機能に問題が発生していました。SSL ハンドシェイクを完了しようとしてサーバで CPU が使用され、その使用率が 100 % に達していました。この問題の原因は、読み込まれたバイト数についての演算エラーでした。</p> <p>この演算上の問題は修正されました。現在では、セキュア ポートに対してプレーン テキストのメッセージを送信しようとした場合、クライアントはより早い段階でエラーを受け取るようになり、CPU 使用率が 100 % に達することによるサーバのハングは発生しなくなりました。</p>

変更要求番号	説明
CR185438	<p>WebLogic Server と外部 LDAP サーバとの間に、ロード バランサまたはファイアウォールがコンフィグレーションされている場合、外部 LDAP サーバへの接続が切断されることがありました。</p> <p>接続が切断された場合に WebLogic Server によって自動的に接続がリフレッシュされるように修正され、ユーザ リクエストの処理中に接続が切断した場合にも、そのリクエストが完了するようになりました。</p>
CR186916	<p>SubjectUtils.isUserInGroup には 2 つの点で問題がありました。</p> <ul style="list-style-type: none"><li>• 特定のクラスのプリンシパルのみを取得するには、各プリンシパルを評価しながらプリンシパルのセットを作成した後、そのプリンシパルのセットを検索して、あるプリンシパルが検索対象のグループに属しているかどうかを調べる必要がありました。ここでは事実上、プリンシパルに対して 2 つの検索が行われていました。1 つはサブジェクトのすべてのプリンシパルを取得するための検索、もう 1 つはグループ名に関する検索です。</li><li>• javax.security.auth.Subject インタフェースから AuthenticatedSubject を取得して、サブジェクトがグループ内にあるかどうかを検索しようとしても、そのサブジェクトが認証されていない限りできませんでした。つまり、サブジェクトを認証してから、サブジェクトが属しているグループを判別するという追加のサイクルが必要でした。</li></ul> <p>解決策：</p> <p>現在は、すべてのプリンシパルをサブジェクトから取得し、指定された WLSGroup を調べるために 1 回だけ検索をするようになりました。</p> <p>さらに、第 1 パラメータとして AuthenticatedSubject を取る 2 つ目のメソッドが追加され、それら両方のパブリックな isUserInGroup () メソッドが、サブジェクトではなくプリンシパルのセットを使用する内部的な isUserInGroup () メソッドを呼び出すように変更されています。このメソッドを使用することで、サブジェクトから AuthenticatedSubject に変換する必要がなくなります。</p>
CR136414	<p>WebLogic Server では、VeriSign の証明書を使用しているサイトへの HTTPS URL 接続との間に SSL ハンドシェイクを成立できませんでした。</p> <p>この問題は修正されています。</p>
CR161839	<p>TTL キャッシュ内での同期が適切に処理されなかったために、ArrayIndexOutOfBounds 例外が発生していました。</p> <p>TTL キャッシュに同期化処理を追加したことで、この問題は解決されました。</p>

変更要求番号	説明
CR177647	特定のケースで、双方向の SSL 通信を確立するために、サーバの証明書がクライアントの証明書として使用されていました。 コードの変更により、この問題は修正されました。
CR186439	単一のクラスタを含むドメインを Administration Console を介して起動した場合に、LDAP サーバのロック メカニズムが正しく機能するようになりました。その結果、管理対象サーバが起動中にハングすることはなくなりました。
CR189238	認証のために ServletAuthentication.weak を使用している場合、ユーザ名とパスワードを入力した時点で NoSuchElement 例外が送出されました。 この問題を解決するために、要素を返す前にサイズをチェックするように InvalidLogin を変更しました。
CR194315	ファイル レルムのリフレッシュにより NullPointerException 例外が発生する可能性がありました。 コードの変更により、この問題は解決されました。今後は、ファイル レルムがリフレッシュされても NullPointerException 例外が発生することはありません。
CR178854	WebLogic Server 7.0 の以前のサービス パックに同梱されていたデモ用証明書および信頼性のある CA 証明書の中に、2004 年 5 月 14 日で期限の切れているものや Basic Constraints 機能と連携しないものがありました。WebLogic Server 7.0 サービス パック 2 には、Basic Constraints 機能と連携する、更新済みの信頼性のある CA 証明書が含まれており、それらの証明書 (democert.pem、democert1024.pem ca.pem、ca1024.pem、trusted.crt、demo.crt) はファイルとして提供されています。 WebLogic Server 7.0 の以前のサービス パックを使用していて、信頼性のある CA 証明書が期限切れだったり、Basic Constraints 機能と連携しないデモ用証明書が含まれていたりする場合には、 <a href="http://dev2dev.bea.com/products/wlserver81/wls_demo_cas.jsp">http://dev2dev.bea.com/products/wlserver81/wls_demo_cas.jsp</a> で詳細について参照してください。

変更要求番号	説明
CR189285	<p>iPlanet LDAP 認証プロバイダを使用する場合、WebLogic Server には iPlanet LDAP サーバへの接続のプールが保持されます。プールされている接続は、実際には Big IP への接続であり、Big IP では定期的にアイドル接続を閉じるようにコンフィグレーションされています。</p> <p>そのため、WebLogic Server から iPlanet LDAP サーバに対してクエリまたはコマンドを発行した場合に、その iPlanet LDAP サーバへの接続が Big IP によって閉じられていることがありました。その場合にも、この問題を説明する例外が送出されていませんでした。</p> <p>iPlanet LDAP サーバへの接続に問題がある場合、その LDAP サーバから例外が送出されて Web 層に伝播されるように修正されました。その結果、WebLogic Server からも適切な情報を含む例外メッセージが送出されるようになっていません。</p>
CR205373	<p>LDAP 接続プール内に無効な接続がある場合、ユーザの認証が失敗して LoginException が返されていました。</p> <p>接続プールの既存の LDAP 接続を使用したログインが失敗した場合、プール内に新しい接続が作成されて、その接続が使用されるようになりました。その結果、LDAP 接続のタイムアウトによってユーザのリクエストが失敗することはありませんになりました。</p>
CR210229	<p>ServletAuthentication.weak() メソッドの呼び出しで、EmptyStack 例外が発生することはなくなりました。</p>
CR194073	<p>現在、WebLogic Server は critical とマークされた場合にのみ extendedkeyusage 拡張を処理します。その結果、双方向 SSL 通信中にサーバサイドで BAD_CERTIFICATE 例外は送出されなくなりました。</p>
CR045135	<p><a href="http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA04-65.00.jsp">http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA04-65.00.jsp</a> のセキュリティ勧告を参照してください。</p>
CR125592 CR196597	<p><a href="http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA04-66.00.jsp">http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA04-66.00.jsp</a> のセキュリティ勧告を参照してください。</p>
CR127930 CR107359	<p><a href="http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA04_51.00.jsp">http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA04_51.00.jsp</a> のセキュリティ勧告を参照してください。</p>
CR128940	<p><a href="http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA04-67.00.jsp">http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA04-67.00.jsp</a> のセキュリティ勧告を参照してください。</p>

変更要求番号	説明
CR157157	<a href="http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA04-71.00.jsp">http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA04-71.00.jsp</a> のセキュリティ勧告を参照してください。
CR171885	<a href="http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA04_59.00.jsp">http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA04_59.00.jsp</a> のセキュリティ勧告を参照してください。
CR172187	<a href="http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA04-69.00.jsp">http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA04-69.00.jsp</a> のセキュリティ勧告を参照してください。
CR175310	<a href="http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA04_64.00.jsp">http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA04_64.00.jsp</a> のセキュリティ勧告を参照してください。
CR175966	<a href="http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA04-72.00.jsp">http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA04-72.00.jsp</a> のセキュリティ勧告を参照してください。
CR128940	<a href="http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA04-67.00.jsp">http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA04-67.00.jsp</a> のセキュリティ勧告を参照してください。

## サーブレット

変更要求番号	説明
CR120545	weblogic-applicaiton.xml にカスタム xml パーサをコンフィグレーションしているときに、Null Pointer 例外が送出されることはなくなりました。
CR121297	エラー ページが適切なユーザに転送されるように修正されました。
CR112793	クライアントで接続が切断された場合、WebLogic Server ではその後、リクエストが処理されていませんでした。 WebLogic Server にシステム プロパティ ProxyForConnectionResets が追加されました。このプロパティはデフォルトで false に設定されています。 ProxyForConnectionResets を true に設定すると、処理中にクライアントによって接続が切断された場合にも、HttpClusterServlet によるリクエストの処理が続行されます。 結果として、クライアントで接続が切断されてもリクエストが処理されるようになりました。



変更要求番号	説明
CR106348、 CR125989	<p>リスナの障害により、Web アプリケーションがアンデプロイされずに不安定な (デプロイされた) 状態のままになっていました。</p> <p>リスナに障害があった場合 (デプロイメントが完全に成功しなかった場合)、Web アプリケーションのデプロイメントがロールバックされるようになりました。</p>
CR134351	<p>WebLogic Server の HttpURLConnection にデフォルトの読み取りタイムアウト値を設定するために、weblogic.http.client.defaultReadTimeout フラグが導入されました。</p> <p>このフラグが設定されている場合、JVM 上に新しく作成される HttpURLConnection はすべてデフォルトでこの値を保持することになります。この値は、特定の接続で setTimeout または setReadTimeout を呼び出すことでオーバーライド可能です。このフラグを使用しない場合、デフォルトの値は -1 になります。値はミリ秒単位で指定する必要があります。</p>
CR132447	<p>部分文字列として SESSIONID 名を含むヘッダ内にクッキーが存在した場合、SESSIONID に予期しない変更が見られました。</p> <p>クッキーのあるヘッダから SESSIONID を解析するロジックを修正した結果、この問題は解決しました。</p> <p>今回の変更により、既存の SESSIONID または既存のセッションがある場合には、新しい SESSIONID が作成されなくなっています。</p>
CR143448	<p>サーブレット コンテナの内部呼び出しで</p> <pre>java.lang.IllegalStateException: HttpSession is invalid</pre> <p>例外が発生します。同じセッション ID を使用する他のスレッドが ServletRequestImpl.syncSession() の処理中にセッション オブジェクトを無効化すると、SessionData.putValue() または SessionData.isNew() の呼び出し時に IllegalStateException が発生する場合があります。</p> <p>セッションが他のスレッドによって無効化されている場合には、IllegalStateException を無視するようにしてください。</p>
CR128519、 CR132321、 CR130021	<p>HttpProxyServlet の使用時にサーブレット フィルタからのラップされた応答が関与すると、ClassCastException が発生していました。</p> <p>コードの修正により、応答ラッパーが使用されている場合に元の応答が取得されるようになりました。</p>

変更要求番号	説明
CR175651	<p>Struts ベースのアプリケーション内で <code>RequestProcessor.getServletContext()</code> メソッドを呼び出したときに、<code>NullPointerException</code> が送出されていました。</p> <p>この問題を修正するために、WebLogic Server の起動スクリプト ファイルにシステム プロパティ <code>weblogic.http.requestCompletionTimeoutSecs</code> が追加されました。このフラグに指定する値は、コンテナが処理中の作業をすべて終了する前に待機する秒数を示します。デフォルトの値は 0 秒で、このフラグに設定をしない場合、コンテナは待機を行いません。</p> <p>サーブレット コンテナは、どのような Web アプリケーションをデプロイまたはアンデプロイする前にも、このフラグに指定された秒数だけ待機します。</p>
CR184726	<p>転送されたリクエストをポストするクライアントで、<code>weblogic.httpd.inputCharset</code> が設定されている場合にパラメータが失われることはなくなりました。</p>
CR176941	<p><code>GenericProxyServlet.readStatus</code> が断続的に失敗します。この問題は、バックエンド サーバによってすでに閉じられた接続を <code>GenericProxyServlet</code> が再利用した場合に発生します。</p> <p>オープン途中ソケット例外が発生して接続のヘッダがフィルタにより除去された場合に、同じリクエストが再試行されるようにコードを修正したことで、この問題は解決しました。</p>
CR172672	<p>WebLogic Server によってすでに閉じられている接続を <code>HttpClusterServlet</code> が再利用しようとする、<code>SocketException</code> が捕捉され、WebLogic Server が BAD とマークされていました。</p> <p>再利用された接続の使用時に、<code>HttpClusterServlet</code> が <code>SocketException</code> を捕捉した場合、新しい接続で改めて同じ WebLogic Server への接続が試行されるようになりました。</p>
CR195698	<p><code>external_stage</code> 領域で古い WAR モジュールを新しいものに置き換えると、404 エラーが発生していました。</p> <p>実際のアプリケーションを一時フォルダにステージングすることにより、この問題は解決されました。</p>

変更要求番号	説明
CR200450	<p>ブラウザから渡されるロケールのバリエーションが原因で、Weblogic Server の検証ルールに予期しない結果が発生していました。ロケールのヘッダを解析するロジックで、そのバリエーション部分が考慮されていませんでした。</p> <p>言語ヘッダの解析中にロケールのバリエーションが考慮されるようになりました。言語ヘッダは次のような形式になります。</p> <pre>lang-country-variant;q=weight</pre>
CR207481	<p>受信 JSP リクエストの URI がフィルタ内で変換された場合に、Web コンテナで同じ変換済み URL に対して JSPStub が重複して生成され、結果としてメモリリークが発生していました。コードの変更によりこの問題は修正され、Web コンテナで重複する JSPStub が作成されることはなくなりました。</p>
CR188237	<p>HttpClusterServlet の使用時に HttpServletResponse オブジェクトでカスタム出力ストリームを使用する場合に発生していた問題が、解決されました。現在、クライアントでは HttpServletResponse オブジェクト内でカスタム出力ストリームを使用できるようになっています。</p>
CR172214	<p>デフォルトの HttpServer から Web アプリケーションをアンデプロイする処理が、デフォルトの HttpServer にはデプロイされていない場合にも行われていました。Web アプリケーションがデフォルトの HttpServer にデプロイされている場合のみ、デフォルトの HttpServer からのアンデプロイ処理が行われるようになりました。</p>
CR209387	<p>Java クライアントで setTimeout() メソッドが使用されていて、Java クライアントから呼び出しを実行した場合、HTTPURLConnection が HTTP 呼び出しを (2 度ではなく) 1 度だけ実行するようになりました。</p>
CR211410	<p>ChunkUtils.is2chunk が無限ループでスタック状態に陥ることがなくなりました。その結果、WebLogic Server のパフォーマンスが向上しました。</p>
CR188953	<p>weblogic.cache.filter.CacheFilter を使用してキャッシュされたページに対して複数のユーザがアクセスした場合に、JSP ページがハングすることがなくなりました。</p>
CR129064	<p>HTTP リクエストが iPlanet プラグイン経由で送信される場合に、誤って動的サーバリストが WebLogic クラスター 2 に設定されることはなくなりました。</p>

# Simple Network Management Protocol (SNMP)

---

変更要求番号	説明
CR109689、CR103678	<p>サードパーティのコレクタ タスクを使用して SNMP 情報を収集した場合、以下のメッセージがログに記録されていました。</p> <pre>&lt;Error&gt; &lt;SNMP Agent&gt; &lt;000000&gt; &lt; Unable to set Entry Field Value ...&gt;</pre> <p>修正コードを実装して、この問題を解決しました。</p>

---

## Web サービス

---

変更要求番号	説明
CR127391	<p>SOAP HTTP バインディングでは、Web サービス内部で例外が発生した場合、サーバは HTTP 500 「Internal Server Error」と、その例外を示す SOAP 障害応答を送出する必要があると規定されています。</p> <p>多くの Web サービス クライアントでは Web サービスへの HttpConnection を作成するために URLConnection が使用されますが、Weblogic Server では java.net.HttpURLConnection を異なるバージョンでオーバーライドしていました。そして、ステータス コードが 400 以上の場合、WebLogic のバージョンで getInputStream メソッドから例外が送出されていました。</p> <p>この例外が送出された後は、Web サービス クライアントが入力ストリームを取得する手段がなく、これは SOAP HTTP バインディングの仕様に違反していました。</p> <p>500 エラー発生時に入力ストリームが取得されるように修正されました。</p>
CR181695	<p>autotype で継承が適切に処理されていませんでした。types.xml ファイル内で、派生型が基底型を拡張していませんでした。</p> <p>autotype に継承機能が追加され、継承が適切に処理されるようになりました。</p>

---

変更要求番号	説明
CR182069	<p>autotype に typeMappingFile を設定した場合、 予期されるとおりに機能しませんでした。</p> <p>この問題は修正され、autotype で typeMappingFile から既存の型が取得されるようになりました。</p>
CR175471 CR175536	<p>外部 Web サービスに発信プロキシ サーバ (iPlanet Web プロキシ サーバ) を介してアクセスすると、SOAP 障害が返されていました。この問題は、プロキシ認証が有効なプロキシ サーバを介したりリモート Web サービスの呼び出しに、http および https を使用することが許可されていないために発生していました。</p> <p>http および https がプロキシ サーバをトンネリングして Web サービス クライアントに到達できるように、コードが変更されました。</p>
CR177611	<p>autotype によって生成された型マッピング ファイルを servicegen で使用すると、web-service.xml 内でネームスペースの問題が発生していました。</p> <p>servicegen で、autotype により生成された型マッピング ファイル内のネームスペースが認識されるようになりました。</p>
CR183555	<p>JAX-RPC クライアントで、SESSION_MAINTAIN_PROPERTY を使用してセッションを維持することができませんでした。</p> <p>SESSION_MAINTAIN_PROPERTY が次のように設定されている場合に、セッションの維持がサポートされるようになりました。</p> <p>呼び出し側オブジェクトの設定 :</p> <pre>call.setProperty(Call.SESSION_MAINTAIN_PROPERTY, "true"); または call.setProperty(Call.SESSION_MAINTAIN_PROPERTY, new Boolean("true"));</pre> <p>スタブ側オブジェクトの設定 :</p> <pre>(Stub)port)._setProperty(Stub.SESSION_MAINTAIN_PROPERTY, "true"); または (Stub)port)._setProperty(Stub.SESSION_MAINTAIN_PROPERTY, new Boolean("true"));</pre>
CR186319	<p>xml ファイル内に 32 個を超えて連続するスペースがある場合、ファイルの解析に長い時間がかかっていました。</p> <p>バッファのサイズが拡大され、より効率的な xml ファイルの解析が可能になりました。</p>

変更要求番号	説明
CR197698	クライアントサイドのパッチが原因で、Web サービスの HTTPS クライアントが <code>java.net.UnknownHostException: null</code> で失敗することはなくなりました。
CR185173	WebLogic Server 6.1.4 の動的クライアントから WebLogic Server 7.0.2 の Web サービスを呼び出して、VersionMaker ユーティリティを介して <code>webserviceclient+ssl.jar</code> ファイルを実行してから、変更された jar ファイルを WebLogic Server 6.1.x 環境で使用した場合に、 <code>java.lang.NoClassDefFoundError</code> が送出されなくなりました。

## WLEC

変更要求番号	説明
CR121105	<p>WLEC クライアントでは、WLEC 接続プールに関する以下の問題が発生していました。</p> <ul style="list-style-type: none"><li>Administration Console で ConnectionPool が適切に表示されない。</li><li>登録解除されたエンドポイントにより COMM_FAILURES が発生する。</li></ul> <p>これらの問題を解決するコード修正が実装され、WLEC 接続プールの全体的なパフォーマンスが向上しました。</p>

# WebLogic Tuxedo Connector

変更要求番号	説明
CR136608	<p>WTC で Tuxedo に応答を送信する前には、XA.End を呼び出す必要があります。しかし WTC では TP/TPA 呼び出しにおいて、XA.End を呼び出す前に、Tuxedo に応答を送信していました。結果として競合状況が発生し、XA.End が呼び出される前に元のスレッドで XA.Start が開始される場合があります。</p> <p>成功した応答と失敗した応答のすべてのパスをクリーン アップし、Tuxedo へのどのような応答の前にも XA.end が呼び出されるようになりました。</p>
CR172581	<p>WebLogic Server 7.0 の次に示す WTC FML メソッドでは、WebLogic Server 8.1 で同じメソッドを実行した場合に比べてパフォーマンスが低下していました。次のメソッドで低下が見られました。</p> <ul style="list-style-type: none"><li>● TypedFML32._tmpostrecv()</li><li>● Fget()</li><li>● Foccurs()</li><li>● Fadd()</li></ul> <p>これらのメソッドについて、WebLogic Server 7.0 sp6 の時点で WebLogic Server 8.1 からの移植が実施されました。</p>

変更要求番号	説明
CR175113	<p>スレッド不足により、スループットが低下していました。WTC では受信するリクエストに対して、InboundEJBRequest 型のオブジェクトを生成する移植スレッドを生成し、さらにそのオブジェクトを実行するために別のスレッドを生成します。そしてスレッドの実行は「デフォルト」実行キューから行われます。InboundEJBRequest によって実行される EJB アプリケーションの処理速度が、入力リクエストが ( リモート Tuxedo ドメインから ) 到着する速度と比べて遅かった場合に、スレッド不足が発生することがあり、結果として、まずは「デフォルト」実行キューでバックログが発生し、続いて入力リクエストが失われていました。</p> <p>処理速度の遅い EJB アプリケーションを、専用のスレッド プールで実行できるようになりました。WTCService では、「weblogic.wtc.applicationQueue」という名前の実行キューが WTC アプリケーション実行のための専用キューとして認識されます。このキューが事前にコンフィグレーションされていない場合、システム プロパティ <code>weblogic.wtc.applicationQueueSize</code> が設定されていることが分かれば、このキューは ( WTC の起動時に ) 明示的に作成されます。</p>
CR177212	<p>WTC リクエストに対して、<code>NullPointerException</code> が送出されていました。</p> <p>WTC で <code>StringBuffer</code> を初期化する前に、<code>Utilities.xdr_decode_string</code> の戻り値がチェックされるようになりました。</p>
CR179410	<p>WTC で再接続の後に古い接続がクリーン アップされていませんでした。Tuxedo ドメインと WTC は <code>ON_STARTUP</code> 接続ポリシーを使用して接続されます。Tuxedo ドメインの実行されるマシンが停止していた場合、マシンの回復時に WTC で古い接続のクリーン アップが行われていませんでした。</p> <p>同じリモート ドメインで新しい接続が受け取られた場合に、古い接続が切断されるようになりました。</p>



## XML

変更要求番号	説明
CR120545	weblogic-applicaiton.xml にカスタム xml パーサをコンフィグレーションしているときに、NullPointerException が送出されることがなくなりました。
CR135726	<p>特定のスキーマ作成が処理されない場合は、SOAPElement にマップされます。</p> <p>この問題の発生時には、サポートされていないスキーマ作成に遭遇したことが autotype では示されず、生成されたクラスを調べた場合にのみ何らかの処理に誤りがあることが示されていました。</p>

## WebLogic Server 7.0 サービス パック 5 のソリューション

以下の節では、WebLogic Server 7.0 サービス パック 5 (SP5) で解決された問題を説明します。ここで示す解決済みの問題のリストは、定期的に更新されます。

- [4-62 ページの「Administration Console」](#)
- [4-63 ページの「クラスローダ」](#)
- [4-65 ページの「クラスタ」](#)
- [4-68 ページの「コネクタ」](#)
- [4-69 ページの「コア サーバ」](#)
- [4-83 ページの「デプロイメント」](#)
- [4-86 ページの「EJB」](#)
- [4-97 ページの「JCOM」](#)
- [4-98 ページの「JDBC」](#)
- [4-104 ページの「jDriver」](#)

- [4-105 ページの「JMS」](#)
- [4-116 ページの「JSP」](#)
- [4-120 ページの「JTA」](#)
- [4-121 ページの「ノード マネージャ」](#)
- [4-122 ページの「操作と管理」](#)
- [4-127 ページの「プラグイン」](#)
- [4-138 ページの「RMI」](#)
- [4-141 ページの「セキュリティ」](#)
- [4-152 ページの「サブレット」](#)
- [4-165 ページの「SNMP \(Simple Network Management Protocol\)」](#)
- [4-167 ページの「WebLogic Tuxedo Connector」](#)
- [4-166 ページの「Web アプリケーション」](#)
- [4-170 ページの「Web サービス」](#)

## Administration Console

変更要求番号	説明
CR091775	Administration Console からディスパッチ ポリシーをコンフィグレーションできるようになりました。この属性をコンフィグレーションするには、[Web アプリケーション記述子の編集] を選択して、新しいウィンドウで [weblogic-web-app] を選択します。右側に [Dispatch Policy] 属性が表示されます。
CR126506	ユーザはコンソール拡張をコンフィグレーションして、ポリシー定義ページを置き換えることができませんでした。 転送前に拡張を検索するための <code>CreateResourcesAction</code> を有効にするコードが追加されました。

---

変更要求番号	説明
CR132700、 CR135127、 CR174568	<p>GraphApplet の <code>getParameter()</code> メソッドは、最小または最大ヒープサイズの値を取得するために使用されます。WebLogic Server は <code>Integer.parseInt</code> を使用して文字列値を変換していましたが、数が 2G を超える場合はこの解析が失敗して、[ モニタ   パフォーマンス ] タブの <code>applet-weblogic.management.console.applets.GraphApplet</code> に誤った値が表示されました。</p> <p>WebLogic Server は <code>Long.parseLong</code> を使用して、2G を超える場合でもヒープの正しい値を取得するようになりました。</p>
CR100745	<p>Web アプリケーションのデプロイ タブで、アプリケーションが仮想ホストにデプロイされていることが表示される場合は正しく、仮想ホストの対象にデプロイされていることが表示される場合は正しくありませんでした。コードを変更してこの問題を解決しました。</p>

---

## クラスローダ

---

変更要求番号	説明
CR128510	<p><code>MsgAbbrevInputStream</code> の <code>readClassDescriptor()</code> はクラスを解決しようとしたが、不明なクラスに関する <code>ClassNotFoundException</code> が発生しました。対応するデータが読み込まれていない場合、Java シリアライゼーションはこの <code>ClassNotFoundException</code> をスキップしました。</p> <p>現在、<code>MsgAbbrevInputStream</code> <code>readClassDescriptor()</code> はクラスを解決しなくなり、<code>MsgAbbrevInputStream</code> は <code>resolveClass()</code> を実装しています。</p>

---

---

変更要求番号	説明
CR124348	<p>クライアント プログラムは、プロキシ クラスがシステム クラスパスに追加されていないと、<code>java.lang.reflect.Proxy</code> を使用して、WebLogic Server にデプロイされたプロキシ オブジェクトにアクセスできませんでした。オブジェクトがシステム クラスパスにない場合、クライアントは <code>ClassNotFoundException</code> を受け取りました。</p> <p>システム クラスローダだけでなく、アプリケーション固有のクラスローダからインタフェースをロードするために、<code>resolveProxyClass()</code> メソッドが実装されました。</p>
CR111924	<p>リモートサーバ上の EJB からローカルサーバ上の EJB に送出されたユーザ定義の例外を処理する場合、WebLogic Server は <code>ClassNotFoundException</code> を送出しました。この状況は、ユーザ定義の例外がローカル EJB のクラスローダに含まれている場合でも発生しました。例外を処理するソケットリーダーが、アプリケーション クラスローダではなくシステム クラスローダを使用したために問題が発生しました。</p> <p>この問題は、コードを修正して解決されました。</p>

---

## クラスタ

変更要求番号	説明
CR107471、 CR128979、 CR136731	<p>クラスタで HTTP トンネリングを使用する場合に、クライアントが次のメッセージを受け取りました。</p> <pre>java weblogic.Admin -url http://colma:17683 -username system -password password PING 10 &lt;May 29, 2003 10:14:18 AM PDT&gt; &lt;Error&gt; &lt;RJVM&gt; &lt;000515&gt; &lt;execute failed java.net.ProtocolException: Tunneling result not OK, result: 'DEAD', id: '0'java.net.ProtocolException: Tunneling result not OK, result: 'DEAD', id: '0' at weblogic.rjvm.http.HTTPClientJVMConnection.receiveAndDispatch(HTTPC lientJVMConnection.java:422) at weblogic.rjvm.http.HTTPClientJVMConnection.execute(HTTPClientJVMCon nection.java:305) at weblogic.kernel.ExecuteThread.execute(ExecuteThread.java:213 at weblogic.kernel.ExecuteThread.run(ExecuteThread.java:189)&gt; Failed to connect to &lt;urldeleted&gt; due to: &lt;urldeleted&gt;:Bootstrap to &lt;urldeleted&gt; failed. It is likely that the remote side declared peer gone on this JVM]</pre> <p>この問題は、プラグインが各トンネル リクエストを次のサーバへラウンドロビンしようとしたために発生しました。リクエストは同じサーバに固定されていませんでした。</p> <p>この問題は、jsessionid クッキーを設定し、それをプラグインに送り返すことで状態がクライアントで管理されるようにコードを修正して解決されました。</p>
CR112874	<p>WebLogic Server 6.1 SP03 で、ステートフル セッション Bean のクライアントが、クラスタにデプロイされている Bean にアクセスし、そのクラスタでは重みベースのロード バランシングがコンフィグレーションされている場合、重みが 1 番目と 2 番目の管理対象サーバがその順番で強制停止されると、クライアントは次のメッセージを送信しました。</p> <pre>&lt;Jul 22, 2003 1:32:56 AM IST&gt; &lt;Warning&gt; &lt;Kernel&gt; &lt;No replica in list has the expected weight. Reverting to round-robin&gt;</pre> <p>管理対象サーバを再起動すると、ロード バランシング アルゴリズムはラウンドロビンに切り替わりました。分析の結果、管理対象サーバがダウンしたときにレプリカ リストが更新されましたが、競合状態が原因で、RichReplicaList の最大の重みが正しくリセットされなかったことがわかりました。</p> <p>レプリカ リストのサイズが変わったら最大の重みを再計算するようにコードを変更して、問題を解決しました。</p>

変更要求番号	説明
CR111029	<p>クラスタメンバーは、デフォルトの 30 秒間のタイムアウト期間内にハートビートを受信しないと、タイムアウトしました。ハートビートは 10 秒ごとに送信され、サーバはハートビートの取得を 3 期間 (合計待機時間は 30 秒) 待機してから、メンバーをタイムアウトしてクラスタメンバーが使用不可能であると宣言しました。たとえば、セッションレプリケーション中に、セカンダリサーバが TCP レベルで使用できなくなった場合、非常にビジーな Web サイトでは 30 秒の期間が長すぎることもありました。セカンダリがクラスタビューから削除される前に、プライマリは多数のセッションをセカンダリにレプリケートしようとし、その結果サーバがハングしたり速度が低下したりしました。</p> <p>現在は「IdlePeriodsUntilTimeout」の値を調整できるようになっています。config.xml ファイルの &lt;Server IdlePeriodsUntilTimeout="3"&gt; タグで設定します。通常、この値は調整しないでデフォルト (3) のままにしてください。ただし、特定の状況では、アーキテクチャや特定のアプリケーションの問題、または特定の製品のシナリオにおける負荷や利用可能な冗長性に応じて、この値を慎重に調整すると、根本的な原因を特定して修正するまでの間、一時的に問題が軽減される可能性があります。</p> <p>この値を変更する場合は十分に注意し、最大負荷のシナリオでアプリケーションを十分にテストして予期される動作を確認することをお勧めします。すべてのシナリオに対応する推奨事項はないため、この値を変更する必要がある場合は、負荷テストが絶対に必要です。</p>
CR125966	<p>サーバインスタンスが停止される途中でレプリケーションリクエストを処理し、syncSession 中に ClassCastException が発生しました。Web コンテナが ReplicatedSessionContext から MemorySessionContext への型キャストを行おうとしたためです。</p> <p>コードの変更後、サーバインスタンスは停止される途中でレプリケーションリクエストを処理しなくなりました。</p>
CR129234	<p>ReplicatedSessionContext に 2 つのハッシュテーブルがあり、1 つはサーバがプライマリとなっているセッションを格納して、もう 1 つはサーバがセカンダリとなっているセッションを格納しています。セカンダリセッションのハッシュテーブルには sessionId というキーと ROID という値があります。セッションが無効化されて、セッションを削除する要求がセカンダリサーバに届くと、WebLogic Server は ROID を ReplicatedSessionContext に渡し、ROID と値を比較するためにハッシュテーブル内の値を検索した後、エントリを削除しました。</p> <p>WebLogic Server は ROID ではなく sessionId を渡すことで、この動作を回避するようになりました。</p>

変更要求番号	説明
CR127849	<p>クラスタ ノードで均一でないアンバインドとリバインドを行うと、サーバが初期化されていないリモート参照をクライアントに返して、<code>AssertionError</code> が発生する場合があります。</p> <p>この問題は、アンバインド時にプライマリの代理が <code>null</code> に設定されるようにコードを変更して解決されました。</p>
CR112326	<p>レプリケート可能オブジェクトの JNDI リバインド中に <code>weblogic.cluster.BasicServiceOffer</code> でメモリ リークが発生しました。この問題は、コードを修正して解決されました。</p>
CR116954	<p>デフォルト Web アプリケーションがデプロイされていない場合に、応答ヘッダ内のクラスタ サーバリストを取得するために、<code>HttpClusterServlet</code> がダミーのリクエスト「<code>/WLDummyInitJVMIDs</code>」を送信したとき、WebLogic Server は次のようなデバッグメッセージをログに記録しました。</p> <pre data-bbox="377 713 1228 878">#####&lt;Sep 6, 2002 4:56:43 PM EDT&gt; &lt;Debug&gt; &lt;HTTP&gt; &lt;foo&gt; &lt;ms1foo&gt; &lt;ExecuteThread: '13' for queue: 'default'&gt; &lt;kernel identity&gt; &lt;&gt; &lt;101147&gt; &lt;HttpServer(887891,null default ctx,ms1foo) found no context for "/WLDummyInitJVMIDs". This request does not match the context path for any installed web applications and there is no default web application configured.&gt;</pre> <p>コードを変更して問題を解決しました。</p>

変更要求番号	説明
CR121113	<p><code>HttpClusterServlet</code> はエラー メッセージにスレッド名を出力しなかったため、エラーが発生したときに問題の診断が困難でした。</p> <p>トラブルシューティングに役立つように、ログに記録されるエラー メッセージにスレッド名を含めるように <code>GenericProxyServlet.trace()</code> のコードを変更しました。</p>
CR127643、 CR129319	<p>Web アプリケーション内で宣言されたインタフェースを実装した動的プロキシが <code>HttpSession</code> に入り、そのセッションがレプリケート可能であった場合に、WebLogic Server ではセカンダリ サーバでそのインタフェースのクラスをロードすることができませんでした。</p> <p>動的プロキシを正しく解決するために、セカンダリ サーバではインタフェースのあるアプリケーションの名前が必要です。ストリームで <code>applicationName</code> を書き込むために、<code>MsgAbbrevOutputStream</code> で <code>annotateProxyClass</code> が実装されました。受信側で、<code>resolveProxyClass</code> はこのアプリケーション名を使用してアプリケーションからインタフェースのクラスをロードします。</p> <p>この変更の結果、(アプリケーション アーカイブに格納されたインタフェースを実装する) 動的プロキシは <code>HttpSession</code> に入れて、正しくレプリケートすることができます。副作用はないはずです。</p>

## コネクタ

変更要求番号	説明
CR131745	<p><code>LocalTransaction</code> のトランザクション サポートを指定したリソース アダプタでは、使用可能な接続プールに接続が解放されるまで、接続のトランザクション状態はリセットされませんでした。別のスレッドが直ちにそのスレッドを選択して使用しようとする、接続は登録されず、適切に共有されない可能性があります。</p> <p>コードの変更後、トランザクション状態は、トランザクションが完了したら直ちにリセットされるようになりました。これは、利用可能なプールに接続が返される前に行われます。</p>



# コア サーバ

変更要求番号	説明
CR131692	<p>7.0 の停止プロセスでは、アプリケーション レベルのスレッドと WebLogic Server インフラストラクチャの処理を実行しているスレッドが区別されていませんでした。新しい接続のリクエストを拒否すると、WebLogic Server が不適切な動作をしました。</p> <p>たとえば、ノード マネージャが停止途中のサーバを再起動しようとしたり、現在のセッションのセッション レプリケーションが失敗したり、サーバが不明な状態にあるとコンソールが報告したり、といったことです。</p> <p>サーバの正常な停止を支援するコードが追加されました。</p>
CR125245	<p>PreferWebInfClasses が有効になっている状態で、複数のスレッドが Web アプリケーションの同じクラスをロードしようすると、<code>java.lang.LinkageError: duplicate class definition</code>: エラーが発生するおそれがありました。この問題は、クラスが最初にロードされる時に発生しました。</p> <p>ChangeAwareClassLoader の <code>loadClass</code> メソッドを同期化するコード変更で、この問題は解決しました。</p>
CR103999	<p>WebLogic Server では、「Unrecognized property: <code>webservice.client.ssl.strictcertchecking</code>」というエラーが表示されることがありました。このエラーは、渡されたシステム プロパティをプロパティのリストで見つけられないことが原因でした。このエラーは、プロパティのリストでシステム プロパティが渡されていないことが原因です。このエラー メッセージはプロパティが無効になることを示すのではなく、プロパティがプロパティ リストに追加されたら、このメッセージは表示されないはずです。</p>
CR103032	<p>ステートフル セッション Bean の [ トランザクション中の削除を許可 ] 属性をユーザが編集できるように、Administration Console が更新されました。</p>

変更要求番号	説明
CR100705	<p>管理サーバがダウンしている場合に <code>SecurityConfigurationMBean.findDefaultRealm()</code> を失敗させてしまうような、管理サーバへの依存が <code>SecurityConfigurationMBean</code> にありました。</p> <p>管理サーバがダウンしている場合はローカル MBean サーバで処理を実行するメカニズムを提供することで、この問題は解決しました。</p>
CR100373、CR130592	<p>J2EE モジュールのタイプを判別するロジックで、管理対象サーバ用のアプリケーションのローカル コピーまたはステージングされたコピーが使用されていませんでした。この問題は、J2EE モジュールのタイプをサーバが判別できない結果として MSI モードで表面化していました。</p> <p>オリジナルのパスが管理対象サーバの <code>config.xml</code> ファイルにない場合は、<code>stagingLocation</code> を使用して J2EE モジュールのタイプが判別されます。</p>
CR122939	<p>プライマリ サーバとのセッションを維持できないことが原因で、分散デッドロックが発生していました。プライマリでもセカンダリでもないサーバにリクエストが到着すると、セカンダリだけでなくプライマリでも既存のセッションの削除が試行され、現在のサーバで新しいセッションが作成されて、新しいセカンダリが登録されます。その過程で、このサーバは非ブロッキング キュー上のセッションを削除するリモート要求をプライマリに対して行い、プライマリ サーバは非ブロッキング キュー上のそれ自体を削除する削除呼び出しをセカンダリ サーバに対して行います。そのようなリクエストが複数ある場合、非ブロッキング キューにはスレッドが 2 つしかないので現在のサーバで応答を受信する他のスレッドは存在せず、そのために分散デッドロックが発生します。</p> <p>WebLogic Server は、非ブロッキング キューに入ってくるリクエストを処理している間は新しいリモート要求を行わなくなりました。これで、この問題は解決されます。非ブロッキング キューに、応答を受信するスレッドが常に存在するようになるからです。</p>

変更要求番号	説明
CR130352	<p>WebLogic Server 8.1 からサービス パック 4 の 7.0 にバックポートされた WLServer ant タスクは、指定されたプロパティのない config.xml ファイルを作成することがありました。</p> <p>分析の結果、WLServer が明示的には saveDomain (config.xml に MBean の変更を書き込む) を呼び出していないことが判明しました。その代わりに、WLServer は saveDomain を呼び出すトリガに依存していました。問題は、WLServer があまりにも速くサーバを起動および停止するために、トリガが時間的に間に合わないことがあったことでした。</p> <p>saveDomain の呼び出しを WLServer ant タスクに配置し、サーバが停止する前に config.xml が強制的に書き込まれるようにすることで、この問題は解決しました。</p>
CR129002	<p>ユーザが分散送り先のメンバーを一切コンフィグレーションしない場合でも、WebLogic Server は Administration Console からの JMS 分散送り先のコンフィグレーションを許可します。ただし、そのコンフィグレーションが無効な場合はサーバが再起動に失敗していました。</p> <p>JMSLegalHelper のリーガル チェックが緩和された後は、分散送り先がコンフィグレーションされているがメンバーは割り当てられていない場合でも、サーバが起動できなくなることはなくなりました。</p>
CR128648	<p>weblogic.deployer WLConfig タスクがすべての IllegalArgumentException をそのまま受け入れ、それが原因で暗号化されたシステム パスワードの機能に関連するビルド エラーが起きていました。</p> <p>この問題は、コードを修正して解決されました。</p>
CR128445	<p>weblogic.Admin が、サーバ インスタンスで使用されるたびに ServerStartMbean のユーザ名とパスワードを設定していました。ユーザが同じことをしようとした場合は、WRITE 属性へのアクセス権がなく失敗していました。</p> <p>この問題は、ServerStart で WRITE 属性が設定されないようにするコード変更で解決しました。</p>
CR126280	<p>管理サーバを再起動すると、ドメイン管理ポート例外が生じていました。MSI モードをチェックするコードが RMI 起動サービスに追加されました。この修正で、例外が発生しなくなりました。</p>

変更要求番号	説明
CR125309	<p>weblogic.Admin の SHUTDOWN または FORCESHUTDOWN から、不正確な戻りコード ステータスが返されていました。</p> <p>weblogic.Admin の SHUTDOWN または FORCESHUTDOWN からの戻りステータスを修正するコードが追加されました。</p> <p><b>注意:</b> 不正確なステータス コードに依存しているアプリケーションは、記述し直す必要があります。</p>
CR125285	<p>FileDistributionServlet は、sendError() メソッドの呼び出し後に復帰しなければなりません。</p> <p>sendError() の後に (それがメソッドの中で行う最後のことでない場合に) FileDistributionServlet のすべてのメソッドから復帰するためのコードが追加されました。</p>
CR125018	<p>よくある特定のプロキシ処理が、それらが実行されるべきローカル サーバで実行されていませんでした。</p> <p>ローカル呼び出しのマップを実装して userLockoutManager の呼び出しがローカルで行われるようにするコードが追加されました。</p> <p>UserLockoutManager は、管理サーバではなくローカル サーバの情報を返すようになりました。</p>
CR093109	<p>WebLogic Server ドメイン全体を強制停止すると、管理対象サーバだけが強制停止されるのではなく、管理サーバが強制停止されていました。この問題は、コードを修正して解決されました。</p>
CR094219	<p>サーバとドメインのロギングが、次のようにして期待通りに機能していませんでした。</p> <p>ドメイン ログ ファイルが、時間でローテーションするようにコンフィグレーションされているにもかかわらずローテーションできていませんでした。</p> <p>[Limit Log Files] オプションが、SimpleDateNamed ログ ファイルで正しく機能していませんでした。</p> <p>コードの変更で以上の問題は修正され、SimpleDateFormat を使用してログ ファイルに名前を付ける場合にログの fileCount 制限が強制されるようになりました。</p>

変更要求番号	説明
CR094410	<p>DebugHttp が ServerDebugMBean でアクティブ化された場合に、WebLogic Server は SocketResetExceptions を単純なデバッグメッセージではなくエラーとして報告していました。この問題に対処するために、このデバッグメッセージを報告するための logMuxableSocketResetException() がメッセージカタログに追加されました。</p>
CR098578	<p>Bootstrap サブレットが、管理サーバが実際には完全に起動していなかったり、RUNNING 状態にないときに SUCCESS ステータスを返していました。</p> <p>コードの変更の後、Bootstrap サブレットは管理サーバの ServerRuntime MBean の取得を試行し、そのステータスをチェックして成功のステータスを返す前にサーバが RUNNING モードにあることを確認するようになりました。</p> <p>管理サーバの起動と接続を試行する管理対象サーバは現在、管理対象サーバが起動しているときに管理サーバが RUNNING 状態にない場合は「Booted in MSI mode」を返します。</p>
CR098607、CR136879	<p>アプリケーション A がアプリケーション B を使用している状態で、アプリケーション B をルックアップしているときに、アプリケーション A の J2EE コンテナがアプリケーション B の ClassFinder をアタッチすることで DependencyClassLoader を作成します。アプリケーション B の再デプロイメント時に、アプリケーション B は新しい ClassFinder に遭遇し、その古い ClassFinder はその時点で有効ではなくなっています。クライアント リクエストで、サーバは古い DependencyClassLoader を使用しようとし、例外を送出します。</p> <p>例外を生じさせた動作は、変更されました。</p> <p>ReplicaAwareRemoteObject.getPrimaryRepresentative() から impl クラスをロードする目的で DependencyClassLoader が使用されることはなくなりました。</p>
CR104269	<p>Administration Console で新しいカスタム認証プロバイダをコンフィグレーションし、その制御フラグとデフォルト認証プロバイダの制御フラグを両方とも SUFFICIENT に設定し、サーバを再起動して、set attribute コマンドを使用し、サーバを再び再起動した場合に、セキュリティ エラー メッセージが受信され、サーバを起動することができませんでした。</p> <p>この問題は、コードを修正して解決されました。</p>

変更要求番号	説明
CR105338	<p>ファイルの最大数に達したときに、WebLogic Server は管理サーバと管理対象サーバの両方でロギングを停止していました。</p> <p>次のエラー メッセージがドメイン ログ ファイルに出力された後、サーバのロギング サービスが停止していました。</p> <pre> #####&lt;Apr 21, 2003 3:17:39 PM GMT+09:00&gt; &lt;Alert&gt; &lt;Log Management&gt; &lt;KESATO01&gt; &lt;myserver&gt; &lt;ExecuteThread: '10' for queue: 'weblogic.kernel.Default'&gt; &lt;&lt;anonymous&gt;&gt; &lt;&gt; &lt;BEA-170017&gt; &lt;The log file .\myserver\myserver.log will be rotated. Reopen the log file if tailing has stopped. This can happen on some platforms like Windows.&gt; #####&lt;Apr 21, 2003 3:17:40 PM GMT+09:00&gt; &lt;Critical&gt; &lt;Logging&gt; &lt;KESATO01&gt; &lt;myserver&gt; &lt;ExecuteThread: '10' for queue: 'weblogic.kernel.Default'&gt; &lt;&lt;anonymous&gt;&gt; &lt;&gt; &lt;000000&gt; &lt;Handler: 'C:\bea81ga\mydomain\myserver\myserver.log' raised several exceptions. Shutting it down&gt; #####&lt;Apr 21, 2003 3:17:40 PM GMT+09:00&gt; &lt;Error&gt; &lt;Logging&gt; &lt;KESATO01&gt; &lt;myserver&gt; &lt;ExecuteThread: '10' for queue: 'weblogic.kernel.Default'&gt; &lt;&lt;anonymous&gt;&gt; &lt;&gt; &lt;000000&gt; &lt;Handler: 'C:\bea81ga\mydomain\myserver\myserver.log' raised exception when opening. Exception weblogic.logging.LogRotationException null&gt; #####&lt;Apr 21, 2003 3:17:40 PM GMT+09:00&gt; &lt;Error&gt; &lt;Logging&gt; &lt;KESATO01&gt; &lt;myserver&gt; &lt;ExecuteThread: '10' for queue: 'weblogic.kernel.Default'&gt; &lt;&lt;anonymous&gt;&gt; &lt;&gt; &lt;000000&gt; &lt;Handler: 'C:\bea81ga\mydomain\myserver\myserver.log' raised exception when opening. Exception weblogic.logging.LogRotationException null&gt; </pre> <p>この問題が発生すると、すべてのログ メッセージがサーバ ログ ファイルに出力されませんでした。</p> <p>分析の結果、WebLogic Server はローテーション時にログ ファイルを閉じて、ログ ローテーションが失敗した場合に、ログ ファイルが閉じられたままになってしまふということが判明しました。</p> <p>新しいログ ファイルで間違ったインデックスが生成されたときに、ログ ローテーションは失敗していました。最新のタイムスタンプを持つファイルが最後のインデックスであると当然のごとく想定されていたので、そのインデックス名を持つファイルが既に存在しているかどうかチェックされていませんでした。</p> <p>WebLogic Server はログ ファイルが既に存在する場合にログ ファイルを削除しようとせず、rename() が成功かどうかを確認していませんでした。</p> <p>また、時間ベースのログ ローテーションの場合に、マルチ CPU マシンで、同じミリ秒内に複数のローテーションが行われていました。</p> <p>この問題は、コードの修正で解決しました。</p>

---

変更要求番号	説明
CR105516	<p data-bbox="448 256 1231 352">WebLogic Server 7.0 の以前のサービス パックでは、複数のフェイルオーバーが必要なときにステートフル セッション EJB のフェイルオーバーが機能していませんでした。</p> <p data-bbox="448 366 1237 461">同じブラウザ ウィンドウで次のように呼び出しを行うと、クラスタの 1 つのノードだけが生き残る場合に <code>java.rmi.ConnectException</code> が送出されていました。</p> <ol data-bbox="448 475 1231 899" style="list-style-type: none"><li>1. 3 つすべてのクラスタ ノードが動作しています。</li><li>2. ノード 1 に対して呼び出しを行い、EJB を作成して、リモートを http セッションに格納します (HTTP セッションのレプリケーションが有効になっている )。</li><li>3. ノード 1 を強制停止します。</li><li>4. セカンダリ ノード 2 に対して呼び出しを行い、EJB のリモートがレプリケートされた HTTP セッションから取得され、EJB の呼び出しが成功します。この EJB の呼び出しの後、再びリモートが HTTP セッションに格納されます。</li><li>5. ノード 2 を強制停止します。</li><li>6. ノード 3 に対して呼び出しを行い、HTTP セッションから EJB のリモートを取得します。</li></ol> <p data-bbox="448 913 1231 1008">WebLogic Server はノード 2 の EJB をルックアップしようとし、ノード 3 は使用しようとしませんでした。ノード 3 で次の例外が送出されていました。</p> <pre data-bbox="448 1022 1231 1446">java.rmi.ConnectException: Could not establish a connection with -3088833905169218734S:172.23.64.38:[7001,7001,7002,7002 ,7001,7002,-1]:mydomain:managed2, java.rmi.ConnectException: Destination unreachable; nested exception is: java.net.ConnectException: Connection refused: connect; No available router to destination at weblogic.rjvm.RJVMImpl.getOutputStream(RJVMImpl.java:275) at weblogic.rjvm.RJVMImpl.getRequestStream(RJVMImpl.java:408 ) at weblogic.rmi.internal.BasicRemoteRef.getOutboundRequest(B asicRemoteRef.java:97) at weblogic.rmi.cluster.ReplicaAwareRemoteRef.invoke(Replica AwareRemoteRef.java:255)</pre> <p data-bbox="448 1454 962 1480">この問題は、コードを修正して解決されました。</p>

---

変更要求番号	説明
CR106957	<p>WebLogic Server 7.0 の以前のサービス パックでは、AIX 5.2 上で IBM の MQWorkflow と一緒に動作している場合に、WebLogic Server でサブレットが MQWorkflow Java API を呼び出したときに、MQWorkflow でエラーが生じていました。この問題は、ネイティブ IO (パフォーマンス パック) が無効になっている場合、または libmuxer.so ライブラリがクラスパスから削除されている場合は Windows では起こりませんでした。</p> <p>分析の結果、「言語コード」(エンコーディングパラメータ)が適切な「en-us」に設定されず、「c」に設定されていたことが判明しました。この問題は、コードを修正して解決されました。</p>
CR108727	<p>WebLogic Server 7.0 の以前のサービス パックでは、管理サーバが以下のコマンドライン オプションで起動していました。</p> <pre data-bbox="377 656 964 755">weblogic.management.discover.interval = 60 weblogic.management.discover.retries = 6 weblogic.management.internal.debug = true</pre> <p>障害の後 (java.rmi.ConnectException が送出される)、管理対象サーバは検出されて、正常に再接続されましたが、OutOfMemoryError が生じていました。</p> <p>その後は、管理対象サーバ上の MBean の呼び出しが次の例外で失敗していました。</p> <pre data-bbox="377 939 1116 994">java.rmi.NoSuchObjectException: RemoteInvokable - id: '267'</pre> <p>分析の結果、ManagedServerRediscoveryChecker スレッドが停止して、管理対象サーバが検出されなくなっていたことが判明しました。この問題は、必要なときに必ず検出スレッドが実行されるように再試行ロジックを変更して解決しました。</p>
CR109307	<p>以前は、SP3 ドメインでは SP2 の管理対象サーバを起動できませんでした。コード変更の結果、異なるサービス パックが互換性を持つようになり、「管理ドメイン内のサーバは、管理サーバが管理対象サーバと同じかそれ以上のサービス パック レベルにある限りどのサービス パック レベルであってもかまわない」という WebLogic Server の互換性に関する一文と調和するようになりました。</p>



変更要求番号	説明
CR109391、 CR123398、CR174565	<p>以前のサービス パックでは、WebLogic Server は <code>ServletContext</code> の中にあればシリアライズできないオブジェクトであってもデシリアライズしようとしていました。その結果として、<code>NotSerializableException</code> が送出されていました。</p> <p><code>ServletContext</code> で、シリアライズできないオブジェクトはデシリアライズされなくなりました。</p>
CR109688、CR135235	<p>特定の状況下では、<code>InitialContext</code> をルックアップしているシンプルな Java クライアントが偽の「バージョン情報がない」エラーを受信していました。</p> <p>問題は、ヘッダバッファの改行が <code>line.separator</code> プロパティではなく「<code>\n</code>」でマーキングされていたことです。この問題は、コードの変更で解決されました。</p>
CR110233、CR121682	<p>プロダクション モードで動作しているサーバインスタンスが不適切に <code>\applications</code> フォルダをチェックし、そのチェックを記録するための一時ファイルにエントリを作成していました。開発モードでは、サーバの起動時に、アプリケーションは自動的に <code>\applications</code> フォルダでアンデプロイされて再デプロイされます。</p> <p>コードの修正で、この問題は解決されました。</p>

変更要求番号	説明
CR111239	<p>失われた JMS 接続がクリーンアップされていませんでした。クライアント (トピック サブスクライバ) は、JMS サーバとは異なるマシンで動作してました。クライアントとサーバ間の顧客のネットワーク接続が失われたときに、サブスクライバ、セッション、および接続のクローズと再確立の後、最初のサブスクライバが削除されていないことが Administration Console で示されていました (コンソールでは 2 つのサブスクライバが表示されていた)。接続ファクトリの <code>&lt;MessagesMaximum&gt;</code> 値に達すると、新しいメッセージが発行されるたびに「保留メッセージ数」値が増加します。</p> <p>分析の結果、ネットワークの停止が原因で、クライアントの <code>peerGone</code> メッセージがサーバインスタンスに到達していなかったことが判明しました。<code>peerGone</code> を宣言した後、クライアントはサーバインスタンスに再接続しようとしていました。サーバインスタンスは古い <code>rjvm</code> で <code>peerGone</code> を宣言することなく古い接続を新しい接続に置き換えて、そのために JMS オブジェクトがメモリに残される結果となっていました。</p> <p>同じクライアントからの同じプロトコルの重複接続を検出したときに、サーバが古い <code>rjvm</code> で <code>peerGone</code> を宣言し、新しい <code>rjvm</code> と新しい接続の新しい <code>ConnectionManager</code> を作成するようにコードを変更して、この問題は解決しました。</p>
CR116707	<p>WebLogic Server は特定の <code>Factory/Trader</code> パターンが使用されたときに <code>ReplicaAwareRemoteRef</code> 警告ログを提供し、次のメッセージをクライアント側で表示していました。</p> <pre data-bbox="377 1013 1177 1065">[ReplicaAwareRemoteRef] : WARNING: client-side RA stub didn't find environment on thread</pre> <p>コード変更の後は、不適切なログがクライアントに対して出力されなくなりました。</p>

変更要求番号	説明
CR116712	<p>以前のサービス パックでは、INVOKE が管理ツールを使用して WLECConnectionPoolRuntime で発行された場合に WLECConnectionRuntime MBeans が更新されていませんでした。</p> <p>分析の結果、resetConnectionPool() が WLECConnectionPoolRuntime MBean で呼び出されたときに、プールがリセットされていないことが判明しました。古い接続が削除されていませんでした。</p> <p>この問題は、コードを修正して解決されました。現在、古い接続に対応する MBean は resetConnectionPool() の呼び出しの前に登録が解除されるようになっています。</p>
CR120492、CR122551	<p>HTTP トンネリングのパフォーマンスを向上させるためのコード変更で、キーブライブ接続が維持されるようにトンネリングされる応答のコンテンツ長が設定され、HTTP URL 接続のキャッシュも可能になりました。</p>
CR121801	<p>WebLogic Server 8.1 で導入された wlconfig ツールが、サービス パック 5 から 7.0 で利用できるようになりました。</p>
CR134971	<p>(サーバサイドから) 例外メッセージ フィールドの一部として &lt;&lt;no stack trace available&gt;&gt; が送信されたとき、rmi レイヤはサーバサイド スタック トレースとして例外を繰り返し追加していました。</p> <p>そのためログ ファイルが一杯になりました。この状況は、サーバでヒープが不足し、アプリケーション コード (EJB + サブレット) によって NullPointerException が送出されたときに明らかになりました。</p> <p>WebLogic Server は例外メッセージ フィールドを解析してこの特殊な例外を捕捉し、例外は 1 回だけ追加されるようになりました。</p>

---

変更要求番号	説明
CR122112	<p data-bbox="384 256 1161 352">weblogic.management.timer.Timer クラスを使用して EJB に対する呼び出しが行われた場合に、EJB のホーム インタフェースのルックアップで次の例外が送出されていました。</p> <pre data-bbox="384 361 1174 1182">java.lang.ClassCastException: Cannot narrow remote object to examples.ejb20.timer.SourceDataHome at weblogic.iiop.PortableRemoteObjectDelegateImpl.narrow(Por tableRemoteObjectDelegateImpl.java:200) at javax.rmi.PortableRemoteObject.narrow(PortableRemoteObjec t.java:132) at examples.ejb20.timer.webapp.TimerServlet.callEjb(TimerSer vlet.java:61) at examples.ejb20.timer.webapp.TimerServlet.access\$000(Timer Servlet.java:23) at examples.ejb20.timer.webapp.TimerServlet\$MyNotificationLi stener.handleNotification(TimerServlet.java:81) at weblogic.time.common.internal.TimerListener\$1.run(TimerLi stener.java:48) at weblogic.security.service.SecurityServiceManager.runAs(Se curityServiceManager.java:685) at weblogic.time.common.internal.TimerListener.deliverNotifi cation(TimerListener.java:44) at weblogic.management.timer.Timer.deliverNotifications(Time r.java:382) at weblogic.time.common.internal.TimerNotification\$1.run(Tim erNotification.java:112) at weblogic.security.service.SecurityServiceManager.runAs(Se curityServiceManager.java:685) at weblogic.time.common.internal.TimerNotification.execute(T imerNotification.java:109) at weblogic.kernel.ExecuteThread.execute(ExecuteThread.java: 234) at weblogic.kernel.ExecuteThread.run(ExecuteThread.java:210)</pre> <p data-bbox="384 1194 1167 1317">分析の結果、呼ばれる側の ContextClassLoader を WebLogic Server が Timer クラスに格納していなかったことが判明しました。そのために、通知が追加されたクラスロードとは異なるクラスロードで通知が実行されていました。結果として、ClassCastException が送出されていました。</p> <p data-bbox="384 1329 1161 1456">この問題は、通知が追加されたときに呼ばれる側の ContextClassLoader を保存するコード変更で解決されました。通知イベントを配信する一方で、保存されたクラスロードは配信側 ExecuteThread の contextClassLoader として設定されます。</p>

---

変更要求番号	説明
CR122361	<p>監査有効化メッセージが、USER ID を記録していませんでした。</p> <pre data-bbox="448 300 1231 352">&lt;Aug 29, 2003 12:31:25 AM EDT&gt; &lt;Info&gt; &lt;Configuration Audit&gt; &lt;159909&gt; &lt;Configuration Auditing is enabled&gt;</pre> <p>反対に、無効化メッセージは USER を記録していました。</p> <pre data-bbox="448 404 1237 456">&lt;Aug 29, 2003 12:31:31 AM EDT&gt; &lt;Info&gt; &lt;Configuration Audit&gt; &lt;159910&gt; &lt;USER system, Configuration Auditing is disabled&gt;</pre> <p>コード変更の後、両方の監査メッセージとも USER ID を記録するようになりました。</p>
CR122706	<p>別のクラスタの EJB を呼び出す、クラスタにデプロイされたクラスタ化されている Web アプリケーションが、AssertionError を送出していました。生成されたコードではインタフェースのメソッドを計算する際に適切なメソッドを使用していなかったため、生成されたスタブから間違ったクラスローダが使用されるようになっていたことが、分析でわかりました。</p> <p>コードを変更してこの問題を解決しました。</p>
CR122878	<p>MBean が KernelID で削除されていました。その結果、削除操作が何によって開始されても、監査プロバイダはカーネルが MBean を削除したと記録していました。</p> <p>コード変更の後、MBean は現在の認証されたサブジェクトによって削除されるようになりました。</p>
CR123571	<p>同じマシン上で複数の T3 クライアントを同時に起動すると、それらの複数のクライアントが同じ JVMID を取得し、例外が生じるか、クライアントがハングする可能性があります。この問題は、同じマシン上で同時に複数の T3 クライアントを起動したときのみ発生していました。</p> <p>この問題は、JVMID を生成するコードを修正して解決しました。</p>

## サービス パック 1 から 6 で解決済みの問題

変更要求番号	説明
CR129094、 CR133591、 CR135254、 CR135255、 CR124861、CR132275	AIX プラットフォームでの t3 プロトコルの TCP ウィンドウ縮小に関連するパフォーマンス上の問題が解決されました。
CR135488	<p>以前は、セキュリティ プロバイダ ディレクトリのファイル (デフォルトは /lib/mbeantypes) は、そのファイル拡張子に関係なくセキュリティ プロバイダとして扱われ、サーバの起動時にロードされていました。</p> <p>WebLogic Server 7.0 のサービス パック 5 からは、拡張子が .jar または .zip のファイルだけがセキュリティ プロバイダとして扱われます。他の拡張子を使用するセキュリティ プロバイダは、ロードされなくなりました。ドメインでセキュリティ プロバイダに別のファイル拡張子が使用されている場合、サービス パック 5 ではそれらを変更してください。</p>

# デプロイメント

---

変更要求番号	説明
CR110897、 CR125329、 CR129494	<p>アプリケーションをデプロイした後でアンデプロイし、その後でサーバを再起動した場合に、サーバから <code>IllegalArgumentException</code> が送出されていました。</p> <p>分析の結果、J2EE コンテナが起動中に、コンフィグレーションされているがデプロイされていないアプリケーションについて処理していなかったことが分かりました。そのようなアプリケーションは &lt;Application&gt; スタンザ内で <code>Deployed="false"</code> となっていることから識別されます。</p> <p>J2EE コンテナのコードが変更されて、このような周辺条件の場合にも処理が行われるようになりました。</p>

---

変更要求番号	説明
CR127393	<p>以下のメモリ リークが確認されました。</p> <ol style="list-style-type: none"><li>1) EJB オブジェクトにおけるメモリ リーク。EO オブジェクトの <code>ClientRuntimeDescriptors</code> は、アプリケーションがアンデプロイされた後もヒープに保持されていました。</li><li>2) J2EEApplicationContainer の EJBCache におけるメモリ リーク。EJBCache オブジェクトは、アプリケーションがアンデプロイされた後にクリーンアップされませんでした。</li><li>3) <code>WebAppServletContext</code> の実行時 MBean におけるメモリ リーク。<code>WebAppServletContexts</code> に固有の実行時 MBean は、アプリケーションのアンデプロイ後に登録解除されていませんでした。</li><li>4) <code>KeepAliveCache</code> のタイマーにおけるメモリ リーク。<code>KeepAliveCache</code> のタイマー オブジェクトは、アプリケーション固有のクラスローダに保持されていました。</li><li>5) J2EEApplicationContainerFactory の <code>ddMap</code> におけるメモリ リーク。記述子オブジェクトは、アプリケーションがアンデプロイされた後に登録解除されていませんでした。</li><li>6) Introspector キャッシュが原因のメモリ リーク。アプリケーション固有のクラスは、Struts を使用するアプリケーションをアンデプロイした後も Introspector キャッシュによって保持されていました。</li></ol> <p>以下の機能を実行するコードが追加されました。</p> <ol style="list-style-type: none"><li>1) EO オブジェクトをアンエクスポートすると、アプリケーションのアンデプロイ時に、EO オブジェクトに関連付けられている <code>ClientRuntimeDescriptor</code> オブジェクトが削除されるようになりました。</li><li>2) アプリケーションのアンデプロイ時に EJBCache をクリアします。</li><li>3) アプリケーションのアンデプロイ時に、<code>WebAppServletContext</code> に固有の実行時 MBean を登録解除します。</li><li>4) <code>KeepAliveCache</code> は、アプリケーションのコンテキストに入る前に、タイマーを初期化します。</li><li>5) アプリケーションのアンデプロイ時にデプロイメント記述子をクリアします。</li><li>6) アプリケーションのアンデプロイ時に Introspector キャッシュをフラッシュします。</li></ol>
CR136176	<p><code>LoadBeforeAppActivation="true"</code> を設定した結果、<code>StartupClass</code> が 2 回呼び出されました。</p> <p><code>PostDeploymentStartupClass</code> に該当しない場合はサーバが無視するようにコードを追加しました。</p>



変更要求番号	説明
CR102296	<p>ブール型属性の <code>LoadBeforeAppActivation</code> が <code>StartupClassMBean</code> に追加されたため、起動クラスがデプロイされる順序には 3 つの異なる設定があります。</p> <p>デフォルト：サーバインスタンスが JMS および JDBC サービス、EJB、およびアプリケーションをデプロイした後。</p> <p><code>LoadBeforeAppActivation=true</code>: サーバインスタンスが JMS および JDBC サービスをデプロイした後で、EJB とアプリケーションをデプロイする前。</p> <p><code>LoadBeforeAppDeployment=true</code>: サーバインスタンスが JMS および JDBC サービス、EJB、およびアプリケーションをデプロイする前。</p>
CR109645	<p>削除後にアプリケーションが 2 度目にコンフィグレーションされるときに、アプリケーション内のコンポーネントには正しい対象がありませんでした。管理 MBean に対する変更はコンフィグレーション MBean に伝播されていませんでした。</p> <p>現在は、コンフィグレーション MBean を更新する場合、対象がクラスタであれば、WebLogic Server はクラスタ内のサーバを取得して、各サーバごとに、サーバ名に基づいてコンフィグレーション MBean のオブジェクト名を作成してから、コンフィグレーション MBean を更新します。</p>
CR111065	<p>ある特定の状況では、<code>SlaveDeployer</code> の初期化から、デプロイメントタスクのデルタリストが作成されていました。</p> <p>コードの修正により、デプロイメントタスクのデルタの作成には、<code>MasterDeployer</code> が常に使用されるようになりました。</p>
CR134122、 CR134119	<p><code>StatefulEJBObject.remove()</code> を呼び出すと、このメソッドに対するパッチがない場合でも、EJB オブジェクトがアンエクスポートされました。</p> <p>この問題は、コードを変更して解決されました。</p>
CR128932	<p>Administration Console は、クラスタ内の管理対象サーバの 1 つが正常に停止した場合に、クラスタに割り当てられたアプリケーションがデプロイされていないことを報告しました。</p> <p>コードを変更してこの問題は解決しました。</p>

## EJB

変更要求番号	説明
CR127334、 CR133975	既にバッチセッションされて削除された EJB に対して <code>remove()</code> が呼び出され、EJB の <code>EOImpl</code> インスタンスがガベージコレクションされない場合に、メモリリークが発生しました。  コードを変更してこの問題は解決しました。
CR132510	この問題は、集合値 <code>cmr-field</code> が、作成元のトランザクションとは別のトランザクションでアクセスされたために発生しました。  現在のトランザクションが、生成された <code>oneToManySet</code> の作成元のトランザクション ( <code>createTx</code> ) と同じかどうかをチェックするためのコードが追加されました。同じでない場合は、 <code>oneToManySet</code> の <code>createTx</code> を現在のトランザクションに設定します。
CR131848	BMP の「 <code>cache-between-transactions</code> 」が <code>true</code> に設定されている場合、実行時に <code>ClassCastException</code> が送出されました。  <code>bean.java.rmi.RemoteException: EJB</code>  <code>ClassCastException</code> を回避するために BMP および CMP20 Bean のチェックが追加されました。
CR103038	[トランザクション中の削除を許可] を「 <code>False</code> 」に設定している場合に、アプリケーションがトランザクション中にステートフルセッション Bean を削除しようとすると、不正確なエラーメッセージが表示されました。  <code>weblogic.ejb20.locks.LockTimedOutException</code> は EJB 2.0 仕様で求められた例外ではなく、以下の必要な例外で置き換えられています。 <code>javax.ejb.RemoveException</code>
CR132853、 CR128980、 CR135722	メッセージ駆動型 Bean が同期的なメッセージポーリング方法を使用しているときに、Sonic JMS サーバを使用すると、メッセージ駆動型 Bean コンテナのポーリングの最適化によって、メッセージの受信が遅延する場合があります。  この問題を回避するには、最適化されたポーラーを使用しないでください。Sonic のメッセージ配信方式はこの方法ではうまく動作しないため、Sonic JMS サーバを連続的にポーリングするポーラーを使用してください。  新しいメッセージ駆動型 Bean の動作は、TIBCO および Sonic JMS プロバイダにのみ適用できます。

変更要求番号	説明
CR133602	<p>XA 接続が使用される場合、WebLogic Server は SonicMQ 4.* 用に特別な処理コードを使用します。このコードは SonicMQ 5.* では不要になりました。このため、接続リークが発生しました。</p> <p>WebLogic Server MDB コンテナにある SonicMQ 5.* では特別なコードを使用しないでください。</p>
CR133774	<p>weblogic-application.xml DTD の max-beans-in-cache 要素に関する次の部分の記述が正しくありません。「0 を指定すると、max-beans-in-cache は無制限になります。max-beans-in-cache のサイズに 0 を指定すると、キャッシュの実際のサイズが 0 に設定されて、CacheFullException を引き起こします。」</p> <p>上記の注意は DTD から削除されています。また、weblogic-application.xml の max-beans-in-cache に対して 0 の値の設定を許可しないように、準拠チェックが追加されました。キャッシュに大きなサイズ (無限) を設定する必要がある場合は、DD で max-beans-in-cache の値を java.lang.Long.MAX_VALUE に設定することができます。これにより weblogic-ejb-jar.xml の同じ要素の動作にも準拠します。</p>
CR126413	<p>WebLogic JDriver (Type 2) を使用する場合、blob/clob cmp-field をデータベースに永続化しているときに、CMP Bean が ClassCastException を送出しました。</p> <p>Oracle Type4 および WebLogic Type2 ドライバの両方で LOB (Blob および Clob) のキャストを処理するコードを追加しました。</p>
CR129185	<p>WebLogic Server は Optimistic 同時方式の Bean を別のトランザクションでロードしました。現在のトランザクションをサスペンドして新しいトランザクションを開始し、その Bean をロードしてから、Oracle 以外のデータベースに対する古いトランザクションを再開することで、SELECT 中にロックを取得しないようにしました。Sybase のデフォルトの動作では、SELECT 中に共有ロックが取得されますが、その文が完了する前にロックが解放されるため、ロードの動作を変更しました。concurrency-strategy が Optimistic である場合、WebLogic Server は、isolation-level が Read-Uncommitted または Read-Committed よりも高い場合にのみ、Oracle 以外のデータベースに対するトランザクションをサスペンドしてから再開するようになりました。</p>

変更要求番号	説明
CR127397	<p>エンティティ Bean は、<code>ActivatableServerRef</code> を使用して <code>activate</code> を呼び出し、Bean インスタンスを取得します。Bean のメソッドを呼び出した後、<code>ActivatableServerRef</code> は <code>deactivate</code> を呼び出して、Bean をプールに解放します。問題は、アプリケーションから EJB が呼び出されるときに、<code>deactivate</code> が呼び出されず、Bean が解放されないことでした。</p> <p><code>ActivatableServerRef</code> は <code>notifyRemoteCallBegin</code> および <code>notifyRemoteCallEnd</code> を明示的に呼び出すようになりました。</p>
CR127361	<p>メンバー変数としてリモート オブジェクトを含む Bean のパッシベーションを行うときに、<code>EJBReplacer</code> はリモート オブジェクトを考慮しないため <code>NotSerializableException</code> を送出しました。</p> <p>コードの変更後、Bean インスタンスがメンバー変数としてリモート オブジェクトを含む場合は、<code>EJBReplacer</code> が必要に応じてリモート オブジェクトを交換するようになりました。</p>
CR079164	<p>自動主キー生成テーブル設定中のエラーに関してサーバが表示した、役に立たないデプロイメント エラー メッセージは、ネストされた例外メッセージをより明確に提示する詳細なメッセージに置き換えられました。</p>
CR093520	<p>あるタイムゾーンのマシンで JSP をプリコンパイルし、それと同じ JSP を別のタイムゾーンのサーバにデプロイする場合、<code>WebLogic Server</code> は JSP を再コンパイルすることがありました。この問題は、<code>WebLogic Server</code> が、(JAR コーティリティによって埋め込まれた) JSP のローカル タイムスタンプと、生成されたクラス ファイル内のタイムスタンプを比較して、JSP をチェックしたために発生しました。</p> <p>コンパイル時にタイムゾーンを保存し、そのタイムゾーンをデプロイメント時に使用して再コンパイルが必要かどうかを決定することで、この問題を解決しました。</p>

変更要求番号	説明
CR106136	<p>以前の WebLogic Server 7.0 サービス パックでは、<code>delay-updates-until-end-of-tx</code> が <code>false</code> に設定されている場合、EJB 1.1 CMP Bean のゲッター メソッドは <code>isModified()</code> を呼び出しませんでした。</p> <p>セッション EJB はエンティティ EJB のゲッター メソッドを呼び出しました。両方の EJB にはコンテナ管理のトランザクションがあり、トランザクション属性は <code>Required</code> に設定されていました。ゲッター メソッドが呼び出されるたびに、続いて <code>ejbStore()</code> が呼び出されます。<code>delay-updates-until-end-of-tx</code> は <code>false</code> でした。ただし、Bean の <code>ejbStore</code> が呼び出される前に、コンテナはトランザクションがコミットするまで <code>isModified</code> メソッドを呼び出しませんでした。</p> <p><code>ejbStore</code> は、Bean の <code>isModified</code> メソッドの結果に応じて、<code>postInvoke()</code> から呼び出されるようになりました。この修正で問題は解決されました。</p>
CR108084	<p>送信側が JMS メッセージをポストして、メッセージ駆動型 Bean がトリガされたときに、MDB がデータベースのデータを取得しない場合、メッセージが消失します。この問題は、確認済みの Oracle XA の制限によって発生しました。この制限では、サスペンドされたトランザクションが、開いている結果セットを無効にします。</p> <p>コードを変更して問題を解決しました。</p>

---

変更要求番号	説明
CR110917	<p data-bbox="315 256 1163 352">大きな負荷がかかっているときに WebLogic Server のステートフル セッション Bean サンプル (examples.ejb20.basic.statefulSession) を実行すると、OutOfMemoryError が発生しました。</p> <p data-bbox="315 366 1130 392">「out-of-the-box」コンフィグレーションは次の設定のように変更されました。</p> <pre data-bbox="315 406 673 466">home-is-clusterable = true replication-type = InMemory</pre> <p data-bbox="315 479 1163 539">アプリケーション アーカイブは 2 つの管理対象サーバから成るクラスタにデプロイされました。Java クライアントは次の手順を実行しました。</p> <ul data-bbox="315 565 740 743" style="list-style-type: none"><li>• ホームをルックアップする。</li><li>• EJB のインスタンスを作成する。</li><li>• buy および sell メソッドを呼び出す。</li><li>• EJB の remove メソッドを呼び出す。</li></ul> <p data-bbox="315 756 1163 852">そのような要求でクラスタをロードすると、メモリ不足の状況が生じます。クライアントのアクションが停止したときに、サーバ インスタンスは回復しませんでした。コンソールは高いヒープ使用率を報告し続けました。</p> <p data-bbox="315 866 1163 927">分析の結果、セカンダリの EJLObject がアンエクスポートされず、その結果メモリ リークになったことがわかりました。</p> <p data-bbox="315 940 1163 994">この問題は、セカンダリの EJLObject をアンエクスポートするようにコードを変更して解決されました。</p>

---

変更要求番号	説明
CR111551	<p>以前のサービスの WebLogic Server 7.0 で、EJBC は <code>java.lang.ClassCastException: oracle.sql.BLOB</code> エラーを引き起こす CMP コードを生成しました。</p> <p>分析の結果、クエリに対して生成された CMP RDBMS コードが正しくないことがわかりました。WebLogic Server 7.0 SP05 より前では、サーバサイド JDBC は RMI ドライバを経由して、プール ドライバ、次に DBMS ドライバにアクセスします。WebLogic Server 7.0 SP05 からは、RMI ドライバは外部クライアントでのみ呼び出されるようになりました。サーバ サイド クライアント コードはプール ドライバに直接アクセスし、DBMS ドライバの BLOB を RMI ラップでラップしないで直接返します。この変更では、コードを <code>oracle.sql.BLOB</code> instead of <code>weblogic.jdbc.common.OracleBlob</code> に直接キャストすることが必要です。</p> <p>この問題は、SP05 以降の動作を反映するコードを生成するように EJBC を修正することで解決されました。</p>
CR111670	<p><code>connectionPool</code> の作成に失敗した後で (たとえば、無効な URL またはドライバクラスを使用する)、プールを正常に作成することはできませんでした。</p> <p>この問題は、<code>connectionPool</code> が正常に作成されるかどうかに関係なく、<code>connectionPool</code> を作成する前に <code>connectionPool MBean</code> が作成されたために発生しました。以降にプールを作成しようとすると、同じタイプの 2 つ目の MBean を作成するときに失敗します。</p> <p><code>connectionPool</code> の作成中に例外が発生した場合は、関連付けられた MBean を削除するように <code>JDBCService</code> を変更することで、問題を解決しました。</p>
CR112074	<p>ビジネス メソッドのコンテナ管理の関係において、<code>update</code> の呼び出しとそれに続く <code>remove</code> の呼び出しを実行するときに、トランザクションの途中で <code>ejbStore</code> が呼び出されました。</p> <p><code>BaseEntityManager.caseDeleteRemove()</code> メソッドの内部で <code>flushModifiedBeans()</code> メソッドが呼び出されたために、余分な <code>ejbStore</code> が発生しました。</p> <p>カスケード削除の場合に <code>flushModifiedBeans()</code> が呼び出されるようにコードを変更して、この問題を解決しました。</p>

---

変更要求番号	説明
CR112558	<p>フェイルオーバがクラスタ内のステートフル セッション Bean に対して機能しませんでした。次の例外が発生しました。</p> <pre>Start server side stack trace: java.rmi.NoSuchObjectException: Unable to locate EJBHome: 'de.roland24.common.system.session.BackendSessionHome' on server: 't3://192.168.201.52:0 at weblogic.ejb20.internal.HomeHandleImpl.getEJBHome (HomeHandleI mpl.java:78) at weblogic.ejb20.internal.HandleImpl.getEJBObject (HandleImpl.ja va:188) at weblogic.servlet.internal.session.SessionData.getAttribute (Se ssionData.java:426)</pre> <p>この問題はコードを変更して修正されました。</p>

---



---

変更要求番号	説明
CR112265	<p>EJBGEN タグを使用してコンテナ管理による永続性 EJB をビルドするとき、以下の例外を受け取りました。</p> <pre data-bbox="377 335 1224 1124">&lt;Jul 17, 2003 2:49:20 PM MDT&gt; &lt;Info&gt; &lt;EJB&gt; &lt;010097&gt; &lt;Exception during the invocation of EJB "Customer(Application: classproject, EJBComponent: xpersistence_ejb. jar)" with primary key "[CustomerPK guidStr:4E9A50DD-8E1B-F2B5-24E7-860ADE3BE707 ]": java.lang.NullPointerException at com.zbc.las.commercial.persistence.test.CustomerPK.hashCode(CustomerPK.java:24) at com.zbc.las.commercial.persistence.test.CustomerPK.equals(CustomerPK.java:32) at com.zbc.las.commercial.persistence.test.CustomerBean_b3dzi6__WebLogic_CMP_RDBMS_customerOrders_Set.add(CustomerBean_b3dzi6__WebLogic_CMP_RDBMS_customerOrders_Set.java:289) at com.zbc.las.commercial.persistence.test.CustomerBean_b3dzi6__WebLogic_CMP_RDBMS_customerOrders_Set.addAll(CustomerBean_b3dzi6__WebLogic_CMP_RDBMS_customerOrders_Set.java:328) at com.zbc.las.commercial.persistence.test.CustomerBean_b3dzi6__WebLogic_CMP_RDBMS.setCustomerOrders(CustomerBean_b3dzi6__WebLogic_CMP_RDBMS.java:672) at com.zbc.las.commercial.persistence.test.CustomerBean_b3dzi6_ELOImpl.setCustomerOrders(CustomerBean_b3dzi6_ELOImpl.java:268) at java.lang.reflect.Method.invoke(Native Method) at com.zbc.las.framework.core.persistence.EjbHelper.setRelations(EjbHelper.java:594) at com.zbc.las.framework.core.persistence.EjbHelper.setRelations(EjbHelper.java:351) at com.zbc.las.commercial.persistence.test.CustomerBean.setOrders(CustomerBean.java:302) at ....</pre> <p>この問題は、コードを修正して解決されました。</p>
CR112703	<p>JMSConnectionPoller は、リモートのユーザ名とパスワードを取得するために WebLogic Security フレームワークに渡す必要のある、ユーザの ID を認識しませんでした。このリモートのユーザ名とパスワードは、リモート JMS サーバ (MQ、WebLogic など) との JMS 接続を確立するために必要です。</p> <p>この問題は修正により解決されていますが、ユーザは MDB のセキュリティ ID をコンフィグレーションする必要があります。手順については、『WebLogic エンタープライズ JavaBeans プログラマーズ ガイド』の「<a href="#">メッセージ駆動型 Bean のセキュリティ ID のコンフィグレーション</a>」を参照してください。</p>

---

変更要求番号	説明
CR112838、 CR112225	<p>BLOB については、その生成されたコードで、WebLogic Server は <code>ObjectOutputStream.writeObject</code> と <code>ObjectInputStream.readObject</code> を呼び出して、データベースでの読み書きの前にオブジェクトをシリアライズまたはデシリアライズします。それらの呼び出しは、余分なヘッダ情報を追加します。<code>writeObject</code> メソッドはオブジェクトのクラス、クラスのシグネチャ、クラスの非 <code>transient</code> および非静的フィールドの値を書き込み、加えてそのすべてのスーパータイプが書き込まれます (これが、データベースで余分なヘッダが見られる理由)。それらの呼び出しで、ユーザが WebLogic Server のみを使用して BLOB を設定および取得する場合に問題が起こることはありません。なぜなら、<code>readObject</code> を使用してバイトが適切なオブジェクトに変換されるからです (例の余分なヘッダ情報が必要)。ただし、BLOB が他のベンダまたはプログラムによって次のようにしてデータベースに直接挿入された場合は、</p> <pre data-bbox="314 661 1116 739">OutputStream os = ((weblogic.jdbc.common.OracleBlob) lob).getBinaryOutputStream(); os.write(this.tiffImage); // byte[] tiffImage</pre> <p>問題が起こることがあります。WebLogic Server が <code>readObject</code> を使用しても、ヘッダ情報がないからです。WebLogic Server を使用して挿入されたデータについては、バイトを読み込む他のプログラムは余分なヘッダ情報を直に取得し、そして失敗します。</p> <p><code>&lt;serialize-byte-array-to-oracle-blob&gt;</code> が、永続性の動作をコントロールするために追加されています。この要素を使用すると、OracleBlob にマップされた <code>byte[]</code> 型の <code>cmp-field</code> がシリアライズされるべきかどうかを指定できます。このタグは、<code>weblogic-cmp-rdbms</code> 記述子の新しい互換性スタンプに追加されています。</p> <p>SP2 より前のバージョンでは、OracleBlob にマップされた <code>byte[]</code> 型の <code>cmp-field</code> をシリアライズするのがデフォルトの動作でした。現在は、<code>byte[]</code> は BLOB から取得された <code>OutputStream</code> に直接書き込まれます。以前の動作に戻すには、このタグの値を <code>true</code> に設定します。</p>
CR113161、 CR12220、 CR110917	サーバのコアにあるクラスタ化可能なステートフル EJB で発生したメモリ リークは、コードの変更により解決されました。

変更要求番号	説明
CR116696、 CR121886	<p>byte[] データが 4K より大きい場合、preparedStatement の setBytes() メソッドに問題がありました。</p> <p>setBinaryStream() メソッドを setBytes() メソッドで置き換えることで、この問題を解決しました。その結果、cmp11 で生成されるコードが変わりました。この修正を利用するユーザは、EJB に対して ejbc を再実行し、新しいコードを生成する必要があります。新しいコードを生成しないと、この変更の効果はありません。</p>
CR120450	<p>CMP-field と CMR-field を同じカラムにマップしたときに java.lang.IndexOutOfBoundsException が発生しました。</p> <p>コードを変更してこの問題を解決しました。</p>

変更要求番号	説明
CR121378	<p>WebLogic Server 7.0 SP03 で、Bean 管理による永続性 (BMP) EJB を使用する場 合、BMP Bean を使用して flushModifiedBeans を試行したときに ClassCastException が発生しました。</p> <pre data-bbox="314 366 1166 760"> &lt;Aug 25, 2003 9:39:35 AM EDT&gt; &lt;Info&gt; &lt;EJB&gt; &lt;010040&gt; &lt;Exception in ejbStore: java.lang.ClassCastException: com.logica.mdch.morse.business.meterpoints.MeterPointXBean_vm 8ya9_Impl java.lang.ClassCastException: com.logica.mdch.morse.business.meterpoints.MeterPointXBean_vm 8ya9_Impl at weblogic.ejb20.manager.ExclusiveEntityManager.flushModified(E xclusiveEntityM anager.java:693) at weblogic.ejb20.internal.TxManager\$TxListener.flushModifiedKey s(TxManager.jav a:749) at weblogic.ejb20.internal.TxManager.flushModifiedBeans (TxManag e r.java:329) at weblogic.ejb20.manager.BaseEntityManager.flushModifiedBeans (B aseEntityManag e r.java:1644) at </pre> <p>分析の結果、他の EJB とのトランザクションに關与している Exclusive 同時方 式の BMP が、ファインダを実行する前に、変更された Bean をデータベースと 同期させているときに、例外が発生したことがわかりました。 ExclusiveEntityManger の flushModified のコードでは BMP の場合を考慮せず、 デフォルトとしてコンテナ管理による永続性を想定していました。</p> <p>ClassCastException を回避できるように問題を解決しました。変更の結果、TX の内部で Bean に対してファインダが呼び出されるときに、BMP EJB の ejbStore メソッドがトランザクションで複数回呼び出される可能性があります。 ユーザ コードでは、トランザクションにおける ejbStore の複数の呼び出しに対 応する必要があります。</p>

---

変更要求番号	説明
CR122052、 CR110440、 CR121949、 CR132495	組み合わせキャッシュ (アプリケーション レベルのキャッシュ) を使用するエンティティ キャッシュのタイムアウトに関して、いずれかの Bean の <code>idle-timeout-seconds</code> が 0 に設定されている場合、 <code>idle-timeout-seconds &gt; 0</code> に設定されている他の Bean も、タイムアウト後にキャッシュから削除されません。  <code>idle-timeout-seconds</code> の登録に関するコードを変更することで、問題を解決しました。
CR123213	BMP <code>ejbRemove</code> から呼び出された場合に、ステートレス セッション Bean がフリー プールに返されませんでした。その結果、プールは最大サイズに達し、インスタンスの取得中にタイムアウトしました。  コードを変更してこの問題を解決しました。

---

## JCOM

---

変更要求番号	説明
CR095485	COM オブジェクトがロード中にクローズされると、ソケットが <code>CLOSE_WAIT</code> 状態のままになる可能性があります。  ソケットが <code>endOfStream</code> でクローズされるようになりました。

---

## JDBC

変更要求番号	説明
CR120455	WebLogic Server 6.1 サービス パック 2 でメモリ リークが検出されました。このリークは、WebLogic XA ドライバと共に使用しているデータベースの BLOB カラムへのアクセスに <code>TxDatasource</code> を使用した場合に発生していました。この問題は、コードを修正して解決されました。
CR127460	<code>rmi_jndi</code> および <code>rmi_driver</code> 層での WebLogic Server <code>jDriver for Oracle</code> バージョン 817、901 および 920 に関して、 <code>JDBCHelper</code> クラスを終了する際の接続で複数の <code>ClassCastException</code> が送出されました。 コードを変更して問題を解決しました。
CR126511	Oracle OCI NativeXA では、接続が作成されたスレッド以外のスレッドがその接続を使用することはできません。 WebLogic Server の接続プール アルゴリズムは、どのスレッドでも接続を使用できることを前提としていました。以前の WebLogic Server 接続プール アルゴリズムで Oracle OCI NativeXA を使用すると、 <code>XAException</code> を受け取りました。 コードを変更して、WebLogic Server 8.1 から <code>PinnedToThread</code> 接続プール アルゴリズムをバックポートし、どのスレッドもすべての接続プールに対する独自の接続を保持できるように、接続をスレッドに固定できるようになりました。

変更要求番号	説明
CR108826	<p>現在のマルチプールは時間に基づくフェイルオーバを定義するオプションを提供していません。望ましいのは、セカンダリ接続プールへのフェイルオーバがトリガされるような状況です。フェイルオーバがプール内のすべての接続に対して行われてから、プールは正常な接続がないと判断します。</p> <p>ネットワークの障害を回避するために、10 分間ですべてのチェックが失敗したらセカンダリ データベースにフェイルオーバーするような、時間に基づくオプションが求められます。プライマリが正常な状態になったらプライマリにフェイルバックできるフェイルバック メカニズムも必要です。たとえば、セカンダリがリモート データセンタにある場合、リモートで実行するときは応答時間が長くなります。プライマリ データベースが停止しているときは対応できますが、プライマリがバックアップされたときは対応しません。</p> <p>つまり、プライマリ データベースが停止した場合は、セカンダリ データベースへ接続をフェイルオーバーする前に、xx 時間待機する必要があります。</p> <p>プライマリ データベースが使用可能になっても、指定するまでプライマリに戻るべきではありません。処理内容を指定するオプションや実行時チェックを用いし、これらのフラグが有効な場合はプライマリに戻るなどの処理を行うことができます。</p>
CR130306、 CR135909	<p><code>jta.DataSource</code> で <code>refreshAndEnlist</code> を実行するときに、WebLogic Server は <code>tx.enlist()</code> を呼び出しましたが、<code>refreshAndEnlist</code> 呼び出しで例外が発生すると、接続はプールに返されませんでした。</p> <p>例外を捕捉し、接続をプールに解放するようにコードを変更しています。</p>
CR129379	<p>EJB トランザクションが多数の新しいエンティティを作成するか、JDBC を使用する多数の Bean に関与した場合、WebLogic Server では Oracle カーソルが不足する危険が生じました。予期される Oracle ドライバのバグを回避しようとして、WebLogic Server がトランザクションの終了まで JDBC 文のクローズを遅延し、その文のカーソルを保持していたためです。</p> <p>この動作は変更されて、トランザクションが終了するまでセッションがカーソルを保持する必要はなくなりました。</p>
CR127949、 CR131743	<p><code>Statement.getResultSet()</code> が不要な新しい <code>ResultSet</code> ラップを生成することがありました。</p> <p>コードを変更して問題を解決しました。</p>

変更要求番号	説明
CR127720	<p>新しいバージョンの JDBC ドライバは接続のトランザクション状態を追跡します。ある接続でローカル トランザクションがアクティブな場合、XA 操作を実行できないため、その接続に対して <code>xa_start()</code> を呼び出すと、<code>XAER_PROTO</code> または <code>XAER_RMERR</code> が発生しました。その結果、アプリケーションは、ローカル トランザクションが開始されたものの終了しなかった箇所を、コード内で絞り込むという面倒な処理を行わなければなりませんでした。</p> <p>回復メソッドで、特別な XA 接続がプールに 2 度解放されるのを防ぐようにコードを変更することで、問題を解決しました。</p>
CR126808	<p>それぞれアプリケーション スコープのデータソースを持つ 2 つのアプリケーションが同じサーバにデプロイされていて、データソースが同じ名前である場合、<code>InstanceAlreadyExistsException</code> が送出されました。</p> <p>必要に応じて、プール名をアプリケーション固有の名前にコンフィグレーションするコードを追加しました。</p>
CR133835	<p>文または結果セットが閉じられていない場合、ガベージ コレクションされるときに、必要な場合は <code>finalize()</code> メソッドがそれらを閉じます。必要性のテストには、接続が既に閉じられていた場合に役に立たない例外を送出するメソッドが含まれていました。例外は送出されなくなりました。</p>
CR099363	<p>負荷テストにおいて、接続プールの更新間隔が 15 分に設定されていて、予約時の接続テストとリリース時の接続テストが <code>false</code> に設定されているときに、その JDBC 接続プールは <code>weblogic.common.ResourceException: No available connections in pool</code> 例外を送出しました。この問題は、プール内の接続の一部だけが使用されていると、その他の接続はすべて、コンフィグレーションされた更新テスト間隔が経過した時点でのテスト用に予約されてしまうために発生しました。</p> <p>コードを変更して、この状況を解決する新しいアルゴリズムを実装しました。</p>
CR103603	<pre data-bbox="314 1168 1174 1255">DatabaseMetaData meta = dbCon.getMetaData(); ResultSet rs = meta.getColumns(null, meta.getUserName(), "dual".toUpperCase(), "%");</pre> <p>ユーザ トランザクションまたは トランザクション 対応 EJB で呼び出す場合、接続が閉じられるときに <code>ResultSet</code> が明示的に閉じられない場合、以降の呼び出しでは Oracle データベース カーソルをリークします。カーソルはサーバが停止するまで保持されて、データベースがカーソル不足になるリスクが生じます。</p> <p>WebLogic Server は接続を閉じるときに <code>ResultSet</code> を明示的に閉じるようになりました。</p>



変更要求番号	説明
CR108478	<p>Oracle 920 thin ドライバを使用する 2 つの JDBC 接続プールが含まれるコンフィグレーションで、1 つ (非 XA) は JMSJDBCStore 用に、もう 1 つ (XA) はデータ挿入用に使用する場合、グローバル トランザクション中にクライアントが JMS キューとデータベースにメッセージを置きました。クライアントはこのアクションを繰り返し処理してから終了しました。この後、クラッシュ (<code>kill -9</code> または <code>System.exit()</code>) 後に WebLogic Server を再起動すると、クライアントがグローバル トランザクションを再び開始するときに、次の例外が送出されます。</p> <pre>Unknown Exception during dataInsert java.sql.SQLException: XA error: XAER_PROTO</pre> <p>より強力なエラー処理コードを追加して、回復動作を修正しました。</p>
CR108826	<p>WebLogic Server 7.0SP5 では、マルチプールに以下の拡張が施されました。</p> <ul style="list-style-type: none"><li>● 接続要求のルーティングの改良。</li><li>● 失敗した接続プールが回復したときの自動フェイルバック。</li><li>● マルチプール内の使用中の接続プールのフェイルオーバー。</li><li>● マルチプールに対するフェイルオーバーおよびフェイルバック コールバック。</li></ul> <p>詳細については、『WebLogic JDBC プログラマーズ ガイド』の「<a href="#">マルチプールのフェイルオーバーの拡張</a>」を参照してください。</p>
CR110486	<p>リモートクライアントは JDBC 接続の Prepared Statement キャッシュをクリアできるようになりました。</p> <p>以前は、<code>weblogic.jdbc.rmi.SerialConnection</code> クラスのオブジェクトは <code>WLConnection</code> を実装していませんでした。したがって、リモートクライアントは <code>clearPrepareStatementCache()</code> メソッドを呼び出すことができませんでした。</p> <p>この制限は、コードを変更して修正されました。</p>

---

変更要求番号	説明
CR112295	<p>マルチプールの DataSource を管理サーバにデプロイできましたが、管理対象サーバへのデプロイは失敗して、次の ResourceException が発生しました。</p> <pre>weblogic.common.ResourceException: DataSource (multiDataSource) can't be created with non-existent Pool (connection or multi) (multiPool) at weblogic.jdbc.common.internal.DataSourceManager.createDataSou rce(DataSourceManager.java:251) at weblogic.jdbc.common.internal.DataSourceManager.createAndStart DataSource(DataSourceManager.java:106) at weblogic.jdbc.common.internal.JDBCService.addDeployment (JDBCSe rvice.java:190) at weblogic.management.mbeans.custom.DeploymentTarget.addDeployme nt(DeploymentTarget.java:330) at weblogic.management.mbeans.custom.DeploymentTarget.addDeployme nts(DeploymentTarget.java:590) at weblogic.management.mbeans.custom.DeploymentTarget.updateServe rDeployments(DeploymentTarget.java:568) at weblogic.management.mbeans.custom.DeploymentTarget.updateDeplo yments(DeploymentTarget.java:240) at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method) at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccess orImpl.java:39) at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMeth odAccessorImpl.java:25) at java.lang.reflect.Method.invoke(Method.java:324) at ...</pre> <p>分析の結果、DataSourceManager.validateConnectionPool() メソッドの間に例外が発生したことがわかりました。この問題は、コードを修正して解決されました。</p>
CR117729	<p>weblogic.JTAJDBC が有効になっている場合、WebLogic Server はドライバのベンダとバージョンを次のよう出力します。</p> <pre>&lt;Aug 5, 2003 3:22:55 PM PDT&gt; &lt;Debug&gt; &lt;JDBC XA&gt; &lt;000000&gt; &lt; -tx:null- -pool:OracleThinXAPool- Vendor:Oracle 8.1.7 XA&gt;</pre> <p>Oracle の場合、WebLogic Server は実際のバージョンに関係なく、バージョンを「Vendor:Oracle8.1.7 XA」と出力しました。</p> <p>コードを変更して問題を解決しました。</p>

---

変更要求番号	説明
CR120531、 CR126189	<p>WebLogic Server 7.0 の以前のサービス パックでは、接続が既に解放されている場合に、<code>weblogic.Admin RESET_POOL</code> (または相当する API) を使用して接続プールをリセットすると、次のような例外が発生しました。</p> <pre data-bbox="377 366 1210 470">&lt;Aug 13, 2003 1:33:11 PM EST&gt; &lt;Error&gt; &lt;HTTP&gt; &lt;[WebAppServletContext (7096795,jdbc_webapp,/jdbc_webapp)] Servlet failed with Exception java.lang.Error: 1 Was already released:weblogic.jdbc.common.internal.Connection</pre> <p>ガーベジ コレクションの後、サーバは次のような接続リークの警告を表示しました。</p> <pre data-bbox="377 565 1224 782">&lt;Aug 13, 2003 1:33:19 PM EST&gt; &lt;Warning&gt; &lt;JDBC&gt; &lt;A JDBC pool connection leak was detected. A Connection leak occurs when a connection obtained from the pool was not closed explicitly by calling close() and then was disposed by the garbage collector and returned to the connection pool. The following stack trace at create shows where the leaked connection was created. Stack trace at connection create: at weblogic.jdbc.pool.Connection.&lt;init&gt;(Connection.java:55) at [...]</pre> <p>接続リークの警告がなくなるようにコードを修正しました。プールをリセットするときに接続が既に解放されている場合は、サーバは今後も前述のエラーを適切に表示します。</p>
CR120971	<p>標準の JDBC 呼び出しに JTS 接続を使用する前に、ベンダ接続を取得する場合、<code>jts.Connection.getVendorConnection()</code> メソッドは <code>null</code> を返しました。すべての標準呼び出しは基底の接続を確立しましたが、<code>getVendorConnection()</code> は確立しませんでした。</p> <p>この問題は、コードを修正して解決されました。</p>
CR127891	<p>接続リーク ファイルの形式を変更して情報を読みやすくしました。</p>
CR131575、 CR132652	<p>WebLogic Server は JDBC 接続リークについて警告する、次のコードで始まるバグを送出することがありました。</p> <pre data-bbox="377 1303 1210 1381">[SerialConnection] : Connection Leak detected!!!!!!java.lang.Throwable: StackTrace at creation of connection: /n</pre> <p>この警告を送信したリーク検出コードは使用されなくなりました。コードを変更してこの問題を解決しました。</p>

## jDriver

変更要求番号	説明
CR125135	<p>デフォルトでは、WebLogic jDriver for Oracle/XA データソースの <code>oracleXATrace</code> パラメータの値は <code>false</code> ではなく <code>true</code> に設定されています。このため、<code>config.xml</code> で <code>oracleXATrace=?false?</code> を設定してトレース ファイルを特に無効にしない限り、ドライバは時間がたつとサイズが大きくなる <code>xa_poolname*.trc</code> の形式でトレース ファイルを作成します。</p> <p>値が指定されていない場合は <code>oracleXATrace</code> のデフォルト値を <code>false</code> に設定するようにコードを修正しました。</p>
CR129220	<p>WebLogic Server Oracle jDriver はガベージ コレクションで <code>Clob</code> オブジェクトを適切に解放していませんでした。</p> <p>コードを変更してこの問題を解決しました。</p>
CR113462	<p>Oracle の <code>NLS_LANG</code> を設定した環境で <code>BigDecimal</code> 型を使用する場合に、WebLogic Server は <code>NumberFormatException</code> を送出しました。この問題は、アメリカ式の数字の定義に合わせるために Oracle の <code>NLS_NUMERIC_CHARACTERS</code> パラメータを使用する場合は発生しませんでした。この問題は、コードを修正して解決されました。</p>

# JMS

変更要求番号	説明
CR135798	<p>トランザクション MDB の場合にメッセージングパイプラインが適切に管理されていませんでした。優先順位の高い新しいメッセージがキューに到達した場合、既にパイプラインにあるメッセージは発信されませんでした。そのため、目的の動作を実現するには、メッセージパイプラインのサイズを最小限に抑えることが重要でした。</p> <p>トランザクション MDB からの確認応答の際に、ウィンドウ数をデクリメントするようにコードを修正しました。トランザクション MDB のメッセージングパイプラインは <code>MessagesMaximum</code> 設定を受け付けます。</p>
CR110120	<p>ある管理対象サーバから別の管理対象サーバに JMS サーバを移行し、メッセージを送信した後で、その JMS サーバを最初の管理対象サーバに再び移行すると、次のような例外が発生しました。</p> <pre data-bbox="377 751 1224 1029">&lt;Jun 23, 2003 4:18:49 PM EDT&gt; &lt;Error&gt; &lt;JMS&gt; &lt;040366&gt; &lt;JMS Distributed Destination member "S2_BE1_T1" in "DT1" unable to accept connection from remote member "S1_BE1_T1" due to exception weblogic.jms.common.JMSEException: Failed to setup system subscription because destination S2_BE1_T1 is not available (shutdown, suspended or deleted). &lt;Jun 23, 2003 4:19:54 PM EDT&gt; &lt;Error&gt; &lt;JMS&gt; &lt;040366&gt; &lt;JMS Distributed Destination member "S1_BE1_T1" in "DT1" unable to accept connection from remote member "S2_BE1_T1" due to exception weblogic.jms.common.JMSEException: Consumer already exists</pre> <p>分析の結果、JMS が複数の WebLogic Server Aggregatable オブジェクトを 1 つのサーバインスタンスに内部的にバインドする一方で、WebLogic Server JNDI は 1 つのサーバにつき 1 つの Aggregatable のみを想定しており、その結果、最初の JMS Aggregatable が 2 番目の JMS Aggregatable で置き換えられたことがわかりました。WebLogic Server JNDI では、この状況に対処する JMS の内部的な API が提供されるようになったため、すべてのオブジェクトが保持されます。</p>

変更要求番号	説明
CR132606	<p>同じ WLS サーバにある分散トピック メンバーに対して、コンソールの保留中メッセージのカウンタが間違っていました。リモートの分散トピック メンバーに対するカウンタは適切でした。</p> <p>他の非リモート分散トピック メンバーに転送する場合に、分散トピック参照のカウントを適切に管理するようにコードを変更しました。</p> <p>すべてのメッセージが正常に受信および応答確認された後で、分散トピック メンバーの予期しない保留中メッセージが表示されることはなくなりました。</p>
CR128596	<p>分散送り先を使用したブリッジを介して 7.0 SP2 クラスタから 8.1 SP2 クラスタへメッセージを送信する場合に、8.1 SP2 クラスタのいずれかのサーバからパウンスされると、メッセージ フローが滞留することがありました。</p> <p>コードを変更してこの問題を解決しました。</p>
CR127260	<p>MessagingBridgeRuntime MBean は次の <code>weblogic.Admin</code> コーティリティ コマンドで停止しませんでした。</p> <pre data-bbox="314 760 1163 840">java weblogic.Admin -url t3://127.0.0.1:7001 -username weblogic -password weblogic INVOKE -type MessagingBridgeRuntime -method stop</pre> <p>MessagingBridge start および stop メソッドは、MessagingBridge の実行時の起動および停止方法を提供する例外を送出するようになりました。</p>
CR126183	<p>最大アイドル時間の設定を過ぎた後に、アイドル状態のブリッジがメッセージをロギングしていました。</p> <p>アイドル状態のブリッジによってロギングされる「Bridge X start transferring messages」という繰り返しのログ メッセージを抑制するコードを追加しました。</p> <p>ブリッジを停止して再起動したり、例外が発生してブリッジを再起動したりする場合、「Bridge "bridgename" starts transferring messages」というログ メッセージが表示されますが、アイドル状態のブリッジによってメッセージが繰り返しロギングされることはありません。</p>
CR125693、 CR092468	<p>JMSConnectionFactoryMBean の <code>flowMinimum</code> 属性の有効な最小値が 1 に変更されました。したがって、整合性を保つために、<code>flowMaximum</code> 属性の有効な最小値も 1 に変更されました。</p>
CR133155	<p>WebLogic Server は起動時に JDBC ストアから JMS メッセージを回復するのに時間がかかり過ぎました。<code>getTables</code> プリフィックスに JMS を含めることで問題を解決しました。</p>

変更要求番号	説明
CR103001、 CR127436	メッセージングブリッジ送り先をホストしているサーバが利用できない場合に、WebLogic Server は長過ぎる例外を表示しました。短い例外メッセージを表示するようにコードを修正しました。
CR107729、 CR113714、 CR124985	コンシューマのない送り先で期限切れのメッセージが蓄積する問題は、コードを変更してアクティブなメッセージ有効期限のサポートを追加することで解決しました。この機能を使用すると、コンシューマを持たない送り先から期限切れメッセージを削除できます。アクティブなメッセージ有効期限を有効にするには、スキャンの間にシステム プロパティ <code>-Dweblogic.jms.ExpirationScanIntervalSecs="number of second(s)"</code> を使用します。ゼロおよび負の数を指定するとこの機能は無効になります。
CR108665	JMS サーバは、キュー ブラウザの参照リストにまだあるかどうかに関係なく、期限切れのページアウト メッセージを永続ストアから削除することがありました。その結果、多くのメッセージが列挙値リストでまだ使用可能である場合でも、ページング IO 例外により列挙値の処理が終了することがありました。 コードの変更によってページングの処理が改良され、問題は解決されました。
CR111159	JMS のフロントエンドとバックエンドが同じ場所になく、フロントエンドの <code>FEProducerSendRequest</code> が <code>Failure/JMSException</code> を受け取った後で、別のスレッドで <code>FEProducerFiniteStateMachine</code> を再開すると、フェイルオーバーされません。 分析の結果、プロデューサ ロード バランサが送り先メンバーを選択するものの、メッセージを送り先に追加しようとするバックエンドが例外を送出する場合、フロントエンドは例外を適切に処理せず、送信先として別の分散送り先メンバーを選択しました。 フロントエンド プロデューサがバックエンドから例外を捕捉して、使用可能な別の分散送り先メンバーを選択して再試行するようにコードを変更することで、問題を解決しました。他のメンバーが使用可能でない場合は、送信側アプリケーションに例外が返されます。
CR110991、 CR117044	<code>JMSServerRuntimeMBean</code> のメソッド <code>getHealthState</code> はパブリック メソッドではなくなりました。

---

変更要求番号	説明
CR112845	<p>クライアントとフロント エンド (JMS 接続ファクトリが割り当てられているサーバ) が同じ場所にあり、JMS 分散送り先メンバーが別の VM 上にある場合、JMS ディスパッチャは JMS 送信操作のフェイルオーバを処理できませんでした。</p> <p>コードを変更してこの問題を解決しました。</p>
CR120619	<p>JMS テンプレートを持つ JMS サーバからすべての対象を削除して、JMS サーバに再び割り当てようとすると、WebLogic Server は例外を送出しました。</p> <pre data-bbox="314 522 1170 951">&lt;Aug 13, 2003 4:50:58 PM PDT&gt; &lt;Error&gt; &lt;JMS&gt; &lt;Failed to deploy JMS Server "server_name" due to weblogic.jms.common.JMSEException: Error initializing JMSServer server_name. weblogic.jms.common.JMSEException: Error initializing JMSServer server_name at weblogic.jms.backend.BackEnd.initialize(BackEnd.java:448) at weblogic.jms.JMSService.createBackEnd(JMSService.java:906) at weblogic.jms.JMSService.addJMSServer(JMSService.java:1273) at weblogic.jms.JMSService.addDeployment(JMSService.java:1169) at weblogic.management.mbeans.custom.DeploymentTarget.addDeployment(DeploymentTarget.java:364) at weblogic.management.mbeans.custom.DeploymentTarget.addDeployment(DeploymentTarget.java:150) at java.lang.reflect.Method.invoke(Native Method) [...]</pre> <p>JMS サービスが一時的送り先ファクトリを RMI ランタイムにエクスポートし、ファクトリが JNDI からアンバインドされるときにファクトリ参照が削除されなかったために、この問題が発生しました。ファクトリがアンバインドされるときに一時的送り先ファクトリの参照が削除されるようにコードを修正しました。JMS テンプレートを持つ JMS サーバの対象を削除して再び割り当てても例外が発生しなくなりました。</p>

---



変更要求番号	説明
CR121011、 CR131700	<p>ストアにメッセージが格納されていて、そのメッセージの ReplyTo が分散送り先であり、ストアと同じクラスタにない場合、JMSServer の再起動時に NPE が発生します。</p> <p>ドメイン間通信に MessagingBridge を使用する WebLogic Server が 2 つありました。送信側アプリケーションが JMSReplyTo フィールドをドメイン 1 にある DistributedDestination に設定した後、メッセージがブリッジ経由で送信され、ドメイン 2 にある JMSStore に到着しました。ドメイン 2 にある JMSServer を再起動すると、JMS はストアに書き込まれた分散送り先メンバー インスタンスの configMBeanName を検索しようとしたのですが、この名前はドメイン 1 に属しているのを見つかりませんでした。ここで障害が発生しました。ログには次のようなメッセージが含まれていました。</p> <pre>#####Aug 7, 2003 1:36:13 PM BST&gt; &lt;Info&gt; &lt;JMS&gt; &lt;hwbachp&gt; &lt;managed5_hwbwl98&gt; &lt;main&gt; &lt;kernel identity&gt; &lt;&gt; &lt;040108&gt; &lt;User connection factory "BridgeJMSnnn" is started.&gt; #####Aug 7, 2003 1:36:13 PM BST&gt; &lt;Info&gt; &lt;JMS&gt; &lt;nnn&gt; &lt;managed5_nnnn&gt; &lt;main&gt; &lt;kernel identity&gt; &lt;&gt; &lt;040108&gt; &lt;User connection factory "Wnnn_JMSConnectionFactory" is started.&gt; #####Aug 7, 2003 1:36:13 PM BST&gt; &lt;Info&gt; &lt;JMS&gt; &lt;nnn&gt; &lt;managed5_nnnn&gt; &lt;main&gt; &lt;kernel identity&gt; &lt;&gt; &lt;040108&gt; &lt;User connection factory "nnnQueueFactory" is started.&gt; #####Aug 7, 2003 1:36:13 PM BST&gt; &lt;Info&gt; &lt;JMS&gt; &lt;hwbachp&gt; &lt;managed5_hwbwl98&gt; &lt;main&gt; &lt;kernel identity&gt; &lt;&gt; &lt;040108&gt; &lt;User connection factory "WLI_B2B_TopicFactory" is started.&gt; #####Aug 7, 2003 1:36:13 PM BST&gt; &lt;Info&gt; &lt;JMS&gt; &lt;hnnn&gt; &lt;managed5_nnnn&gt; &lt;main&gt; &lt;kernel identity&gt; &lt;&gt; &lt;040108&gt; &lt;User connection factory "RNQueueFactory" is started.&gt;</pre> <p>JMS が configMBeanName を見つけられない場合、そのインスタンス名を使用して replyTo フィールドの DestinationImpl (非 DD キュー) を作成するようにコードを変更して、問題を解決しました。replyTo キューは、分散送り先からこのシナリオの通常の送り先にダウングレードされました。</p> <p>変更後の動作：サーバの再起動時にメッセージがストア内にあり、そのメッセージには a) ReplyTo がある、b) この ReplyTo はこのストアが属するクラスタと同じクラスタにはない、c) この ReplyTo は分散送り先である場合、この ReplyTo は通常の送り先として組み込まれます。configMBeanName が (別のクラスタから) 見つからないために replyTo が分散送り先から通常の送り先にダウングレードされるので、この replyTo を使用するクライアントは LoadBalancing を取得しません。</p>

変更要求番号	説明
CR121041	<p>同じドメインの 2 つの管理対象サーバで動作する 2 つの JMS トピックの間のメッセージングブリッジ (同じマシン上で動作) で問題が発生しました。WebLogic Server の起動時にメッセージングブリッジを起動すると、次の警告メッセージを送出し始めました。メッセージは転送されませんでした。</p> <pre> #####&lt;Aug 20, 2003 12:49:11 PM MST&gt; &lt;Warning&gt; &lt;MessagingBridge&gt; &lt;slsol4.bea.com&gt; &lt;server2&gt; &lt;ExecuteThread: '10' for queue: 'default'&gt; &lt;kernel identity&gt; &lt;&gt; &lt;200026&gt; &lt;Bridge "Test Messaging Bridge" encountered some problems to one of its adapters or underlying systems. It stopped transferring messages and will try to reconnect to the adapters shortly. (Exception caught was java.lang.Exception: javax.resource.spi.IllegalStateException: Managed connection is closed at weblogic.jms.adapter.JMSManagedConnection.checkIfDestroyed(JMSManagedConnection.java:263) at weblogic.jms.adapter.JMSManagedConnection.sendEvent(JMSManagedConnection.java:231) at weblogic.jms.adapter.JMSBaseConnection.throwResourceException(JMSBaseConnection.java:1266) at ... </pre> <p>分析の結果、ブリッジは、恒久トピックに対してコンフィグレーションされていて、使用可能な JMS ストアがなかったため、メッセージリスナの作成に失敗したことがわかりました。リソース例外をログに記録しようとすると、ブリッジで内部エラーが発生したため、ユーザはブリッジが失敗した理由を確認できませんでした。</p> <p>ブリッジが適切なリソース例外を送出できるようにコードを変更して、問題を解決しました。現在はブリッジによって正しい例外がロギングされます。</p> <pre> &lt;Aug 27, 2003 9:32:35 AM CDT&gt; &lt;Warning&gt; &lt;MessagingBridge&gt; &lt;200026&gt; &lt;Bridge "TopicBridge" encountered some problems to one of its adapters or underlying systems. It stopped transferring messages and will try to reconnect to the adapters shortly. (Exception caught was java.lang.Exception: javax.resource.ResourceException: Error setting message listener . . . . . . ----- Linked Exception ----- javax.resource.ResourceException: Error creating asynchronous consumer or setting message listener . . . . . . ----- Linked Exception 2 ----- weblogic.jms.common.JMSException: No store for durable consumer . . . </pre>

---

変更要求番号	説明
CR121741	<p>MBean を使用して MQ-WLS ブリッジを停止しているときに矛盾した動作が起きました。次の抜粋は停止で使用されたコードです。</p> <pre>MessagingBridgeMBean bridge = (MessagingBridgeMBean)mbeanHome. getAdminMBean(bridgeName, "MessagingBridge"); bridge.setStarted(false); for(int i=0;i&lt;serverList.length;i++) { target=(TargetMBean) mbeanHome.getAdminMBean(serverList[i], "Target"); bridge.removeTarget(target);</pre> <p>ブリッジの割り当てが解除されて、次の例外により停止することがありました。</p> <pre>&lt;Aug 27, 2003 9:24:08 AM PDT&gt; &lt;Info&gt; &lt;MessagingBridge&gt; &lt;200034&gt; &lt;Bridge "reverseBridge" is shut down.&gt; &lt;Aug 27, 2003 9:24:08 AM PDT&gt; &lt;Error&gt; &lt;Connector&gt; &lt;190008&gt; &lt;Error closing connection instance for XA Transaction Resource Adapter.&gt; javax.resource.spi.IllegalStateException: Managed connection is closed at weblogic.jms.adapter.JMSManagedConnection.checkIfDestroyed (JMSManagedConnection.java:263) at weblogic.jms.adapter.JMSManagedConnection.cleanup(JMSManagedCo nnection.java:128) at weblogic.connector.common.internal.XATransResourceFactory.clea nUp (XATransResourceFactory.java:325) at ....</pre> <p>分析の結果、MessagingBridge.shutdown() が MessagingBridge.cleanup() を呼び出した後、ソース接続に対する recover() を呼び出したことがわかりました。recover() はトランザクションセッションに対する有効な操作ではないため、例外が発生しました。</p> <p>recover() を呼び出す前に、セッションがトランザクションセッションではないことを検証するように、コードの変更を実装しました。</p>

---

---

変更要求番号	説明
CR121760	<p data-bbox="314 256 1181 447">WebLogic Server メッセージング ブリッジを使用して WebMethods 6.0 JMS と WebLogic Server を統合する場合、<code>InvalidDestinationException</code> を受け取りました。WebLogic Server にメッセージを送信するときに、WebMethods 6.0 は <code>ack</code> を要求します。WebMethods がタイプ <code>com.wm.broker.jms.Destination</code> を想定したときに、ブリッジはタイプ <code>weblogic.jms.DestinationImpl</code> の <code>ack</code> を送信します。次の例外が送出されました。</p> <pre data-bbox="314 465 1181 1116">javax.jms.InvalidDestinationException: Destination must be of class: com.wm.broker.jms.Destination at com.wm.broker.jms.Message.setJMSDestination(Message.java:896) at weblogic.jms.client.JMSProducer._send(JMSProducer.java:380) at weblogic.jms.client.JMSProducer.send(JMSProducer.java:172) at weblogic.jms.adapter.JMSBaseConnection.sendInternal(JMSBaseCon nection.java:571) at weblogic.jms.adapter.JMSBaseConnection.send(JMSBaseConnection. java:528) at weblogic.jms.adapter.JMSConnectionHandle.send(JMSConnectionHan dle.java:131) at java.lang.reflect.Method.invoke(Native Method) at weblogic.connector.common.internal.ConnectionWrapper.invoke(Co nnectionWrapp e r.java:101) at \$Proxy115.send(Unknown Source) at weblogic.jms.bridge.internal.MessagingBridge.onMessageInternal (MessagingBrid ge.java:1160) at weblogic.jms.bridge.internal.MessagingBridge.onMessage(Messagi ngBridge.java: 1093) at com.wm.broker.jms.MessageConsumer.deliverMessage(MessageConsum er.java:682) at com.wm.broker.jms.Session\$MessageDeliveryThread.run(Session.ja va:1431)</pre> <p data-bbox="314 1133 1181 1220">WLS は 2 つの JMS 実装間にブリッジ機能を提供しており、要求はメッセージの 確認応答を正しいタイプに変換することでした。この要求は通常のメッセージに 対しては適切に行われているようです。</p> <p data-bbox="314 1237 1181 1359">分析の結果、標準の WebLogic JMS クライアントは送信する WebMethods メッ セージを渡され、WebLogic 送り先を使用して WebMethods メッセージに対し <code>setJMSDestination</code> を呼び出そうとしたことがわかりました。このとき、 WebMethods は <code>InvalidDestinationException</code> を送出しました。</p> <p data-bbox="314 1376 1181 1428"><code>InvalidDestinationException</code> を捕捉して無視するようにコードを変更すること で、問題を解決しました。</p>

---

変更要求番号	説明
CR122749	<p>キュー内のすべてのメッセージに <code>JMSCorrelationID</code> が設定されていない場合、<code>DestinationKey JMSCorrelationID</code> をキーにしてソートすると <code>NullPointerException</code> になる可能性があります。<code>DestinationKey JMSType</code> の場合にも同じ失敗が起きる可能性があります。</p> <p><code>compareTo</code> を呼び出す前に、<code>null</code> の <code>JMSCorrelationID</code> と <code>null</code> の <code>JMSTypes</code> がないかどうかチェックするようにコードを変更しました。その結果、ソートは <code>NullPointerException</code> を送出せずに適切に動作します。</p>
CR123194	<p>以前の WebLogic Server 7.0 サービス パックで、サーバインスタンスが停止してもクライアントがアクティブな場合、JMS は JMS 仕様で想定される <code>JMSException</code> ではなく、実行時例外 <code>weblogic.rmi.extensions.RemoteRuntimeException</code> を送出しました。</p> <p>サービス パック 5 でコードを変更して問題を解決しました。</p>

---

変更要求番号	説明
CR123675	<p>非恒久メッセージに対する最適化コードの問題が原因で、送り先が無効になることがありました。そのため、負荷の大きな状況でメッセージをページアウトするときに、次のような例外が発生しました。</p> <pre data-bbox="314 361 1174 873">&lt;Sep 12, 2003 9:54:12 PM EDT&gt; &lt;Error&gt; &lt;Kernel&gt; &lt;BEA-000802&gt; &lt;ExecuteRequest failed java.lang.NullPointerException. java.lang.NullPointerException at weblogic.jms.common.MessageImpl.writeExternal(MessageImpl.java :1622) at weblogic.jms.common.TextMessageImpl.writeExternal(TextMessageI mpl.java:92) at weblogic.jms.store.ObjectIOByPassImpl.writeObject(ObjectIOBypa ssImpl.java:155) at weblogic.jms.store.BufferDataOutputStream.writeObject(BufferDa taOutputStream.java:175) at weblogic.jms.store.FileIOStream.write(FileIOStream.java:506) at weblogic.jms.store.StoreRequest.doTheIO(StoreRequest.java:282) at weblogic.jms.store.JMSStore.execute(JMSStore.java:493) at weblogic.kernel.ExecuteThread.execute(ExecuteThread.java:197) at weblogic.kernel.ExecuteThread.run(ExecuteThread.java:170)</pre> <p>この問題は、コードを修正して解決されました。</p>

---

---

変更要求番号	説明
CR128723、 CR081630	<p data-bbox="377 262 1193 352">コンシューマのロード バランサにおけるタイミングの問題が原因で、<code>ServerAffinity</code> を有効にすると、ローカル メンバーの方が適切な場合に、誤ってリモート メンバーを選択するという動作が起きました。</p> <p data-bbox="377 366 1231 421">以前のサービス パックでは、<code>ServerAffinity</code> を有効にすると以下のプリファレンスが設定されます。</p> <ol data-bbox="377 439 986 586" style="list-style-type: none"><li>1) コンシューマを持たないローカル メンバーを取得する</li><li>2) ローカル メンバーを取得する</li><li>3) コンシューマを持たないリモート メンバーを取得する</li><li>4) リモート メンバーを取得する</li></ol> <p data-bbox="377 604 1231 659">サービス パック 5 では、<code>ServerAffinity</code> を有効にすると以下のプリファレンスが設定されます。</p> <ol data-bbox="377 677 986 826" style="list-style-type: none"><li>1) コンシューマを持たないローカル メンバーを取得する</li><li>2) コンシューマを持たない任意のメンバーを取得する</li><li>3) ローカル メンバーを取得する</li><li>4) 任意のメンバーを取得する</li></ol>

---

# JSP

変更要求番号	説明
CR172250	<p>weblogic.jspc のプロパティには下位互換性がありませんでした。</p> <p>weblogic.jspc の新しいコマンドライン フラグ <code>backwardcompatible</code> を追加しました。true に設定すると、Web アプリケーションは WebLogic Server 6.x 以前との下位互換性を維持してコンパイルされます。新しい「backwardcompatibility」フラグを使用すると、WebLogic Server 6.x と同じように Web アプリケーションをコンパイルできます。</p>
CR133760	<p>サーバは、「<code>compilerclass</code>」文字列ではなく「<code>compilerClass</code>」文字列を使用して、サーブレット コンフィグレーションから初期化パラメータを読み込もうとしました。JSP 記述子要素は <code>compilerclass</code> であるため、JSPServlet はそれに応じて変更されました。JSPServlet のコードに、メソッド <code>config.getInitParameter("compilerClass")</code> を <code>config.getInitParameter("compilerclass")</code> に変更するコードを追加しました。そのため、JSP 記述子要素の「<code>compilerclass</code>」で対応されるようになりました。</p>
CR093625	<p>サーバが JSP ページの JSP include に対して出力を誤って送っていました。</p> <p>コードの変更に従い、すべてのラッパーが削除された場合は <code>isResponseWrapper</code> がリセットされます (たとえば、すべてのラッパーが JSP 本文タグからの <code>NestedBodyResponse</code> ラッパーだった場合。ラッパーが <code>weblogic.servlet.internal.RemoveWrapperOnForward</code> を実装しているとき、それらは転送時に削除されるためです)。</p>
CR127836	<p>ルート コンテキストのサブコンテキストにある JSP は、展開形式ではなく、WAR ファイルにパッケージ化された Web アプリケーションとしてデプロイされている場合、プリコンパイルされました。</p> <p>コードを変更してこの問題を解決しました。</p>
CR126007	<p>JSP が別の JSP から転送されたときに、マルチバイト文字列パラメータが失われました。この問題は、パラメータ エンコーディングが EUC-JP または ISO2022JP で、POST メソッドによって送信された場合にのみ発生しました。</p> <p>エンコーディングが変更されていない場合は POST パラメータを再解析しないようにコードを変更しました。</p>



変更要求番号	説明
CR092039	<p>JSP の文字セット値の周囲に余分な引用符がある場合、WebLogic Server は <code>UnsupportedEncodingException</code> を送出しました。たとえば、次のタグは例外を引き起こしました。</p> <pre data-bbox="377 361 1180 387">&lt;%@ page contentType="text/html; charset=\"Shift_JIS\" %&gt;</pre> <p>この問題は、コードを修正して解決されました。</p>
CR093520	<p>あるタイムゾーンのマシンで JSP をプリコンパイルし、それと同じ JSP を別のタイムゾーンのサーバにデプロイする場合、WebLogic Server は JSP を再コンパイルすることがありました。この問題は、WebLogic Server が、(JAR コーティリティによって埋め込まれた) JSP のローカル タイムスタンプと、生成されたクラス ファイル内のタイムスタンプを比較して、JSP をチェックしたために発生しました。</p> <p>コンパイル時にタイムゾーンを保存し、そのタイムゾーンをデプロイメント時に使用して再コンパイルが必要かどうかを決定することで、この問題を解決しました。</p>
CR098514	<p>JSP カスタム タグの属性にある文字列リテラルの途中で改行するとコンパイルエラーが発生しました。次に例を示します。</p> <pre data-bbox="377 838 1233 1003">&lt;%@ taglib prefix="c" uri="http://java.sun.com/jstl/core";; %&gt; &lt;c:if test="{ sessionScope.sessionID == null     or sessionScope.dummy == null }"&gt;     There is no Session ID! &lt;/c:if&gt;</pre> <p>上記のコードによって次のエラーが発生しました。</p> <pre data-bbox="377 1055 1224 1150">unclosed string literal     _c_if0.setTest(weblogic.utils.StringUtils.valueOf("{ sessionScope.sessionID == null //[ /ifTest.jsp; Line: 17]</pre> <p>JSP カスタム タグの属性値の中に改行文字を格納できるようにコードを変更することで、問題を解決しました。</p>
CR104429、 CR134313	<p>WebLogic Server は、更新された JSP クラス ファイルが <code>jspWorkingDir</code> にコピーされた後で JSP ページを再コンパイルしました。</p> <p>新しいパラメータ <code>-Dweblogic.jsp.alwaysCheckDisk</code> を <code>true</code> に設定すると、WebLogic Server は古くなった JSP クラスをディスクからチェックします。このパラメータはデフォルトで <code>false</code> に設定されているため、デフォルトの動作は変更ありません。</p>

変更要求番号	説明
<p>CR111024、 CR134046、 CR111897</p>	<p>Cache タグを使用した次のフラグメントのある JSP で、</p> <pre data-bbox="314 296 1161 461"> &lt;wl:cache name="email_portlet" key="request.email_portlet_key" scope="session" timeout="60"&gt; time=&lt;%= new java.text.SimpleDateFormat ("kk:mm:ss.SSS").format(new Date()) %&gt;&lt;br&gt; key=&lt;%= key %&gt;&lt;br&gt; &lt;% for (int i=0; i&lt;100; i++) { // just a delay try { Thread.sleep(10); } catch (Exception e) { } } refreshed = true;%&gt; </pre> <p>10 ユーザを同時にロードする場合、10 スレッドが weblogic.cache.CacheSystem.waitOnLock() でデッドロックしました。 コードを変更してこの問題を解決しました。</p>
<p>CR111423、 CR127336</p>	<p>WAR ファイルにパッケージ化される場合、アプリケーションのサブコンテキストに格納されたプリコンパイル済み JSP が、WebLogic Server にデプロイされる時に常に再コンパイルされました。JSP が展開されたアーカイブディレクトリからデプロイされる場合、または JSP が WAR ファイルのルート コンテキストにある場合は、この問題は発生しませんでした。</p> <p>この問題は、jar および zip フォーマットで使用されるタイムスタンプの四捨五入動作の違いが原因で発生しました。四捨五入の違いによって、WAR ファイル内のクラス ファイルで (1 秒差の) 古いタイムスタンプが記録され、サーバによるクラスの再コンパイルがトリガされました。</p> <p>コンパイル済み JSP クラスのタイムスタンプを 1 秒進めるようにコードを修正しました。それによって、JSP は再コンパイルされなくなります。</p>
<p>CR111655</p>	<p>これまで、日本語文字を含む JSP では JavaServer Pages 標準タグ ライブラリ (JSTL) を使用できませんでした。そのような JSP を実行すると、次の行で始まるエラーが発生しました。</p> <pre data-bbox="314 1107 1089 1216"> java.io.IOException: javax.servlet.jsp.JspException: The taglib validator rejected the page: "org.xml.sax.SAXParseException: An invalid XML character (Unicode: 0x82) was found in the CDATA section." </pre> <p>ページ ディレクティブで pageEncoding 属性を指定しない場合、タグ ライブラリの検証に使用されるバイト ストリームはデフォルトのエンコーディングを使用して作成されました。</p> <p>pageEncoding が指定されていない場合、contentType 属性で定義された文字エンコーディングを必ず使用するようにコードを変更して、問題を修正しました。</p>

変更要求番号	説明
CR112794	<p>適切に解析されていないコピーで JSP が更新されたため、デッドロックが発生しました。</p> <p>JSP 更新中のチェックを追加するようにコードを変更することで、問題を解決しました。</p>
CR117477	<p><code>response.sendRedirect</code> で、Content-Type が <code>text/html</code> ではなく <code>text/plain</code> にデフォルトで設定されました。その結果、WebLogic Server は <code>text/html</code> Content-Type を必要とするコンテンツに対して 503 ステータスのページを返しました。</p> <p>コードの変更後は、<code>sendRedirect</code> がアクティブになるまでは、Context-Type がまだ定義されていない場合 <code>text/html</code> がデフォルトのタイプになります。</p> <p>この変更後は、<code>text/plain</code> Content-Type に依存している受信者は失敗することに注意してください。</p>
CR120914	<p>WebLogic Server のフォーム検証タグ ライブラリを使用する場合、以降の JSP でリクエスト パラメータが使用できませんでした。この問題は、コードを修正して解決されました。</p>
CR123520、 CR106226	<p>WebLogic Server 7.0 サービス パック 4 で、JSP 本体タグで <code>PageContext.include()</code> または <code>forward()</code> メソッドが使用され、インクルードまたは転送されるサーブレットで <code>HttpServletResponse.getOutputStream()</code> が呼び出されると、次の例外が発生しました。</p> <pre>java.lang.IllegalStateException: Cannot use ServletOutputStream because a Writer is being used. Use getWriter() instead.</pre> <p>このインクルード要求はおそらく JSP 本体タグにネストされていました。コードを変更して問題を解決しました。</p>

# JTA

変更要求番号	説明
CR111475	<p>JTARuntime MBean は、調整を行っていないサーバについての統計値をインクリメントしていました。</p> <p>トランザクションを開始し、管理対象サーバにあるリソースを取得して（その管理対象サーバをコーディネータとする）、次に管理サーバにあるリソースを取得し、トランザクションをコミットすると、両方のサーバの JTARuntime MBean が TransactionTotalCount として 1 を返しました。</p> <p>コードの変更により、WebLogic Server は調整を行っているサーバについての統計のみを更新するようになりました。</p>
CR122842	<p>新しいリソースを古いリソースと同じ名前で登録する場合、以降の XA トランザクション処理を新しいリソースに対して行うことができませんでした。元のリソースにアクセスしようとし続けました。</p> <p>同じ名前の 2 番目のリソースに置き換えられるようにコードを追加しました。</p>
CR127412	<p>PinnedToThread が有効になっている場合、または Oracle OCI NativeXA ドライバを使用する場合、アプリケーションは接続を取得できませんでした。</p> <p>この問題は、コードを変更して解決されました。</p>
CR113226	<p>リソース名が 64 文字を超える場合、WebLogic Server は接続プールのテスト中に次の例外を送出する可能性があります。</p> <pre>java.sql.SQLException: XA error: XAER_RMERR : A resource manager error has occurred in the transaction branch start() failed on resource 'weblogic.jdbc.jta.DataSource' null at weblogic.jdbc.jta.DataSource.enlist (DataSource.java:1167) at weblogic.jdbc.jta.DataSource.refreshXAConnAndEnlist (DataSource .java:1133) at weblogic.jdbc.jta.Connection.getXAConn (Connection.java:153) at weblogic.jdbc.jta.Connection.prepareStatement (Connection.java: 241) [...]</pre> <p>この問題は、最初の 64 文字だけがユニークかどうかをテストされたために発生しました。64 文字より長いリソース名を適切に処理するようにコードを変更しました。</p>

変更要求番号	説明
CR134081	<p><code>unregister</code> の呼び出しやリソースの障害によってリソース アダプタが登録解除されると、そのリソース アダプタを対象とするすべてのトランザクション処理が失敗します。</p> <p>サービス パック 5 からは、同じ名前の 2 つ目のリソースアダプタが登録される場合、最初のリソース アダプタを対象としていたすべてのトランザクションは、2 つ目のリソース アダプタ上で動作するようになりました。</p>
CR126201	<p>複数サーバのドメインで、別のアドレスまたはポート番号を使用するために管理対象サーバを再起動すると、JTA サブシステムはアドレス情報の更新に失敗しました。変更後のサーバを再起動すると次の例外が発生しました。</p> <pre> javax.naming.CommunicationException.  Root exception is java.net.ConnectException: t3://ip_address:port_number: Destination unreachable; nested exception is: java.net.ConnectException: Connection refused; No available router to destination [...]</pre> <p>アドレスまたはポートの変更に応じて、新しいアドレス情報を管理サーバから取得するようにコードを修正しました。</p>

## ノード マネージャ

変更要求番号	説明
CR104285、 CR124729	<p>ノード マネージャの共有オブジェクト コードは、サーバ インスタンスの起動時に特定のコード パスがとられたときにセグメント違反を引き起こすことがありました。IBM の zLinux JDK を使用したときにも同じコード パスで問題が発生していました。これらの問題は、ノード マネージャのコード修正により解決しました。</p>
CR127930	<p><a href="http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA04_51.00.jsp">http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA04_51.00.jsp</a> のセキュリティ勧告に関する情報を確認してください。</p>
CR125829	<p><a href="http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA03_42.00.jsp">http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA03_42.00.jsp</a> のセキュリティ勧告に関する情報を確認してください。</p>

## 操作と管理

変更要求番号	説明
CR099488	<p>WebLogic Server は管理ポートの URL を誤って計算していました。</p> <p>最初に管理ポート (有効な場合) に基づいて、次に管理チャンネル (有効な場合) に基づいて、URL が正しく計算されるようにコードを変更しました。</p>
CR128713	<p>JMSQueue の ErrorDestination を設定するときに、ユーザは beanshell 例外を受け取ることがありました。特定の状況では JMSQueue に親が設定されませんでした。リーガル チェックは親を使用するため、ErrorDestination の設定は失敗しました。</p> <p>チェックのために JMSQueue の名前から親の JMSServer を取得するコードを追加して、リーガル チェックを修正しました。</p>
CR134167	<p>EXISTS_POOL が予期されるとおりに動作しませんでした。プールが存在していても、存在する場合もしない場合もある実行時 MBean を探していました。文書化されていないコマンド DELETE_POOL は接続プールを削除する内部的な実装でしたが、この機能は外部に対して DESTROY_POOL として文書化されていました。</p> <p>ヘルプ メニューには文書化されていないコマンドもいくつか表示されました。</p> <p>EXISTS_POOL コマンド実装は、コンフィグレーション MBean を探して、プールが存在するかどうかをチェックするようになりました。</p> <p>DESTROY_POOL コマンドは基底の DELETE_POOL 実装を使用するようになりました。</p> <p>weblogic.Admin ヘルプ メニューに表示されていた文書化されていないコマンドはすべて無効になりました。コマンドには TEST_POOL、REMOVE_POOL、SUSPEND_POOL、SHUTDOWN_POOL、RESUME_POOL、DELETE_POOL があります。これらは今後サポートされず、7.0 SP5 では動作しません。</p> <p>ヘルプ メニューにこれらのコマンドは表示されなくなりました。</p>

変更要求番号	説明
CR136210	<p>パラメータを渡す必要がある場合に、コードは <code>runAs</code> を実行しようとしていました。スタックにパラメータを渡す代わりに、コードはパラメータをオブジェクト変数に隠そうとしていました。また、トークン方式を使用して、<code>runAs()</code> から呼び出すログ メソッドを特定しようとしていました。これらのオブジェクト変数は同期化されませんでした。その結果、同じメッセージが何度もログに記録されることがありました。</p> <p>コードをより明確でスレッド セーフにする匿名の内部クラスを使用するように、コードを変更しました。ログ ハンドラのコードはログ ファイルに書き込むために同期化されました。</p>
CR130441	<p><code>dev2dev</code> のサンプル セキュリティ プロバイダは、7.0 の以前のサービス パックでは正常に機能していませんでした。サンプルをビルドするとき、サーバを起動してドメインを設定しようとする、多数の <code>javax.management.MBeanException</code> エラーを受け取りました。</p> <p>サーバが「<code>commotype</code>」を管理ツールに対するフラグと見なせるように、コードを変更しました。</p>
CR101109	<p>クラスタ内の管理対象サーバに対する JDBC ロギングを有効にすると、コンソールは起動時にエラーを表示しました。このエラーは JDBC ロギングに影響を与えず、JDBC ロギングは適切に動作しました。</p> <p>コードを追加して、管理対象サーバでのサーバ デバッグ メッセージのロギングを修正しました。</p>
CR121228	<p>ユーザ ID が監査ログ メッセージに記録されるようになりました。</p>
CR121216	<p><code>DomainMBean.AdministrationMBeanAuditingEnabled</code> 属性の <code>@exclude</code> および <code>@non-configurable</code> タグが削除されました。</p>
CR132445	<p><code>weblogic.Admin</code> はデフォルト URL で適切に動作していませんでした。次に例を示します。</p> <pre data-bbox="377 1220 1112 1246">java weblogic.Admin -username ss -password ss VERSION</pre> <p>上記のコマンドは結果を返していませんでしたが、<code>-url t3://localhost:7001</code> を追加すると正しい結果を返しました。</p> <p>コードの変更後、<code>weblogic.Admin</code> は入力 URL が <code>null</code> かどうかをチェックするようになりました。<code>weblogic.Admin</code> は既に内部的にデフォルト URL を持っているためです。</p>

変更要求番号	説明
CR111199	<p>管理サーバを起動して、2つの管理対象サーバを同時に起動すると、CPUを大幅に消費しました。一方の管理対象サーバが初期化を完了して、もう一方が初期化を開始すると停止しました。</p> <p>kill -QUIT pid を使用した結果、以下のようにメイン スレッドが HashMap.rehash でスタックしていたことがわかりました。</p> <pre> "main" prio=5 tid=0x29240 nid=0x1 runnable  [0xffbec000..0xffbedbd4] at  java.util.HashMap.rehash(HashMap.java:292) at  java.util.HashMap.put(HashMap.java:344) at  weblogic.management.internal.DynamicMBeanImpl\$XInfo.get(DynamicMBeanImpl.java:2270) at  weblogic.management.internal.DynamicMBeanImpl.&lt;init&gt;(DynamicMBeanImpl.java:195) at  weblogic.management.internal.DynamicMBeanImpl.&lt;init&gt;(DynamicMBeanImpl.java:167) at  weblogic.management.runtime.RuntimeMBeanDelegate.&lt;init&gt;(RuntimeMBeanDelegate.java:70) at  weblogic.jms.frontend.FEConsumer.&lt;init&gt;(FEConsumer.java:100)  at weblogic.jms.frontend.FESession\$3.run(FESession.java:952)  at .... </pre> <p>サイズを変更しなくてもすべてのMBean属性に対応できるようにデフォルトのHashMapのサイズを増やし、HashMapへのスレッドアクセスを同期化するようにコードを変更して、問題に対応しました。</p>
CR111287	<p>Convert weblogic.properties ユーティリティは web.xml と weblogic.xml に DOCTYPE を追加しませんでした。</p> <p>web.xml と weblogic.xml の先頭にそれぞれ次の文を追加するように、ユーティリティを修正しました。</p> <pre> &lt;!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN" "http://java.sun.com/dtd/web-app_2_3.dtd";&gt; &lt;!DOCTYPE weblogic-web-app PUBLIC "-//BEA Systems, Inc.//DTD Web Application 7.0//EN" "http://www.bea.com/servers/wls700/dtd/weblogic700-web-jar.dtd";&gt; CONFIGURATION: WebLogic Server 7.0 SP2 </pre>



変更要求番号	説明
CR120987	<p>管理対象サーバのログが、指定された RootDirectory ではなく NodeManager のホーム ディレクトリ (NODEMGR_HOME) に誤って作成されました。access.log および tlog ファイルは RootDirectory に正しく作成されました。</p> <p>管理対象サーバのリモート スタートができるように、あるマシンで NodeManager がコンフィグレーションされていました。</p> <p>コンソールにある管理対象サーバの [ リモート スタート ] タブで、管理対象サーバの RootDirectory とログの FileName が指定されました。</p> <p>config.xml には管理対象サーバの次のスタンザが含まれていました。</p> <pre data-bbox="377 557 1228 1117">&lt;Server ExpectedToRun="true" ListenAddress="172.23.64.45" ListenPort="27023" Machine="MyMachine" Name="m1" ServerVersion="7.0.3.0" StdoutSeverityLevel="64"&gt; &lt;COM Name="m1"/&gt; &lt;ExecuteQueue Name="default" ThreadCount="15"/&gt; &lt;IIOP Name="m1"/&gt; &lt;JTAMigratableTarget Cluster="" Name="m1" UserPreferredServer="m1"/&gt; &lt;JTARecoveryService Name="m1"/&gt; &lt;KernelDebug Name="m1"/&gt; &lt;Log FileName="m1/m1.log" Name="m1"/&gt; &lt;SSL Name="m1"/&gt; &lt;ServerDebug Name="m1"/&gt; &lt;ServerStart Name="m1" OutputFile="/home/support/BEA/wlserver7.0sp3/user_projects/ja y1/. /NodeManagerClientLogs/jay1_m1/startserver_1061321008754. log" Password="{3DES}7v6S0vZT+xUhhbWkLEq23A==" RootDirectory="/home/support/BEA/wlserver7.0sp3/user_projects /jay1" Username="system"/&gt; &lt;WebServer Name="m1"/&gt; &lt;/Server&gt;</pre> <p>この問題は、コードを変更して修正されました。</p>
CR121839	<p>コマンドラインから有効にする場合、DomainMBean インスタンスは監査状態を反映しませんでした。</p> <p>-Dweblogic.AdministrationMBeanAuditingEnabled スイッチによってコマンドラインで MBean 監査を有効にすると、Domain MBean インスタンスでも、DomainConfig MBean インスタンスでも、属性が true に設定されていませんでした。監査は有効になりましたが、MBean にその事実が反映されませんでした。</p> <p>ドメインの MBean 属性をサーバ起動時にシステム プロパティから設定するようにコードを修正することで、問題を解決しました。</p>

---

変更要求番号	説明
CR121728	<p>機能強化：</p> <p>カスタム認証プロバイダの MBean を、<code>WL_HOME\server\lib\mbeantypes</code> ディレクトリにある WebLogic インストール ツリーの内部に配置する必要はなくなりました。</p> <p><code>WL_HOME\server\lib\mbeantypes</code> は MBean タイプをインストールするためのデフォルト ディレクトリです。ただし、WebLogic Server に追加のディレクトリで MBean タイプを検索させるには、サーバの起動時に</p> <pre>-Dweblogic.alternateTypesDirectory=&lt;dir&gt;</pre> コマンドライン フラグを使用します。<dir> はディレクトリ名のカンマ区切りのリストです。このフラグを使用する場合、WebLogic Server は常に、最初に <code>WL_HOME\server\lib\mbeantypes</code> から MBean タイプをロードします。次に、追加のディレクトリを検索して、そのディレクトリにある有効なアーカイブを ( 拡張子に関係なく ) すべてロードします。たとえば、 <pre>-Dweblogic.alternateTypesDirectory = dirX,dirY</pre> の場合、WebLogic Server は最初に <code>WL_HOME\server\lib\mbeantypes</code> から MBean タイプをロードし、次に <code>dirX</code> および <code>dirY</code> にある有効なアーカイブをロードします。 <p>このオプションを使用し続けないと、サーバを起動できなくなることに注意してください ( たとえば、このオプションを使用してユーザを作成した後で、<code>alternateTypesDirectory</code> オプション使用しないことにした場合など )。</p> <p>詳細については、『WebLogic Security サービスの開発』の「<a href="#">WebLogic Server 環境に MBean タイプをインストールする</a>」を参照してください。</p>

---

変更要求番号	説明
CR122204	<p>管理サーバを再起動せずに、カスタム COMMO Bean を持つ管理対象サーバを起動すると、管理対象サーバは COMMO Bean のインスタンスを作成できませんでした。次の例外が発生しました。</p> <pre>javax.Management.InstanceAlreadyExistsException is thrown, and the Detailed message in the exception is, The Object name specified 'wldomain:Name=t,Server=managed1,Type=AppViewRuntime' is not unique across the domain. Please choose an unique Object Name</pre> <p>分析の結果、停止時に、管理サーバの MBeanServer で MBean が登録解除されなかったことがわかりました。このため、起動時に管理対象サーバは Bean を再作成できませんでした。管理サーバのインスタンス化済み Bean のリストには、その Bean 名の Bean インスタンスがまだあったからです。MBean 名はドメイン全体でユニークでなければなりません。</p> <p>管理対象サーバのサーバ固有の Bean を、サーバの停止時にすべて登録解除するようにコードを変更することで、問題を解決しました。管理対象サーバが停止するとき、管理サーバは PeerGoneEvent を受け取って、その管理対象サーバに関連付けられているすべての MBean を登録解除します。</p>
CR136718	<p>「ルート ディレクトリ」のコンフィグレーションが、ノード マネージャを使用する場合に機能するようになりました。</p>

## プラグイン

変更要求番号	説明
CR134413	<p>Apache プラグインが原因で、302 応答の http ヘッダと本文が重複していました。プラグインとバックエンド サーバの間に問題はありませんでした。Apache サーバが 302 応答を追加していました。</p> <p>request_handler メソッドの戻り値を OK に戻すコードが追加されました。</p>
CR113033	<p>以前のサービス パックでは、ISAPI プラグインが _wl_proxy フォルダの WLTempDir フラグを認識していませんでした。このフラグを使用するようにコードが修正されました。</p>

## サービス パック 1 から 6 で解決済みの問題

変更要求番号	説明
CR132699	<p>アプリケーション サーバがダウンした場合に、Apache アクセス ログが 500 エラーを記録すべきところ、代わりに次のような 200 コードが返されていました。</p> <pre>[Mon Jan 05 12:58:21 2004] [notice] Apache/2.0.44 (Unix) configured -- resuming normal operations [Mon Jan 05 12:58:25 2004] [error] CONNECTION_REFUSED [os error=13, line 1566 of ../nsapi/URL.cpp]: 172.18.137.68:7232 errno = 115 [Mon Jan 05 12:58:27 2004] [error] CONNECTION_REFUSED [os error=13, line 1566 of ../nsapi/URL.cpp]: 172.18.137.68:7232 errno = 115 [Mon Jan 05 12:58:29 2004] [error] CONNECTION_REFUSED [os error=13, line 1566 of ../nsapi/URL.cpp]: 172.18.137.68:7232 errno = 115 [Mon Jan 05 12:58:31 2004] [error] CONNECTION_REFUSED [os error=13, line 1566 of ../nsapi/URL.cpp]: 172.18.137.68:7232 errno = 115 [Mon Jan 05 12:58:33 2004] [error] CONNECTION_REFUSED [os error=13, line 1566 of ../nsapi/URL.cpp]: 172.18.137.68:7232 errno = 115 [Mon Jan 05 12:58:35 2004] [error] CONNECTION_REFUSED [os error=13, line 1566 of ../nsapi/URL.cpp]: 172.18.137.68:7232 errno = 115</pre> <p>この問題は解決済みです。</p>
CR130060	<p>IIS プラグインのパフォーマンス上の問題は、サービス パック 5 で、指定されたコンテンツ長と等しいデータが既に読み込まれているかどうかを確認されるようにするコード変更によって解決されています。</p>
CR129471	<p>以前のサービス パックでは、Apache プラグインは <code>WLTempDir</code> パラメータを認識していませんでした。この問題は修正されています。</p>
CR129342	<p>ISAPI プラグインは、<code>WL-PATH-TRIM</code> 値の代わりに <code>WL-PATH-TRIM HTTP</code> ヘッダ値を WebLogic Server に送信していました。</p> <p>コードを変更してこの問題を解決しました。</p>

変更要求番号	説明
CR129138	<p>NSAPI プラグインがバックエンドの WebLogic Server インスタンスで名前の解決を実行したときに、その名前の解決では <code>sysGetHostByName</code> が使用され、<code>sysGetHostByName</code> では <code>getHostByName</code> が呼び出され、<code>getHostByName</code> では開いているファイル記述子の数量制限がある内部メソッドが呼び出されて、名前の解決が失敗することがありました。</p> <p>クッキー解析を修正し、JVMID を代用してプライマリ サーバおよびセカンダリサーバを見つけるようにすることで、この問題は解決しました。</p>
CR129026、 CR129323	ISAPI プラグインのメモリ リークが、コードの変更で修正されました。
CR127973	<p>サブレット セッションに永続クッキーを追加した後で、ISAPI プラグインが失敗することがありました。</p> <p>クッキー解析コードを修正して問題は解決されました。</p>
CR127658	<p>プールから接続を取得しても、サーバ インスタンスが既にその接続を閉じている場合は、<code>HALF_OPEN_SOCKET_RETRY</code> 例外が送出されて、前の接続オブジェクトが削除され、同じサーバに接続する新しいオブジェクトが作成されていました。</p> <p>この問題は、<code>HALF_OPEN_SOCKET_RETRY</code> 例外を適切に処理するコードを追加することで解決しました。</p>
CR127231	<p>503 HTTP ステータスを受信した後に、リクエストがクラスタ内の次に利用可能なサーバにフェイルオーバーされていませんでした。</p> <p><code>READ_ERROR_FROM_SERVER</code> 例外または <code>CONNECTION_REFUSED</code> 例外が送出されるまで、同じサーバで繰り返し試行されていました。</p> <p>503 HTTP ステータス エラーを取得した時点でサーバに「bad」とマークし、次に利用可能なサーバを取得して、リクエストを再送信するコードが追加されました。現在は、すべてのリクエストが次に利用可能なサーバに正常にフェイルオーバーされます。</p>
CR132840	アプリケーション サーバがダウンしたときに、Apache アクセス ログで 500 エラーではなく 200 コードが不適切に記録されていました。コードを変更してこの問題を解決しました。

変更要求番号	説明
CR135002	複数の仮想ホストがある Apache コンフィグレーションで、1つの仮想ホストだけが WebLogic Server プラグインで SecureProxy=ON がコンフィグレーションされており、それ以外の仮想ホストでは SecureProxy または WLProxySSL が使用されない場合に、SSL がコンフィグレーションされていない仮想ホストで、プラグインがバックエンド WebLogic Server との SSL 接続を試行していました。これが原因で、パフォーマンス上の問題が生じていました。 コードを変更してこの問題は解決しました。
CR112503	NSAPI プラグインではヘッダの文字比較で大文字と小文字が区別されていましたが、現在は区別されません。
CR123775、 CR123120	Post メソッドを使用して Apache プラグイン経由でリクエストを転送すると、プラグインは content-length を 0 に設定し、wlproxy.log は次のメッセージを記録していました。 POST and PUT requests *must* contain a Content-Length この問題は、コードを修正して解決されました。
CR121726、 CR121341	のセキュリティ勧告に関する情報を確認してください。
CR110991、 CR117044、 CR120970、 CR120978	プライマリ サーバから PROTOCOL_ERROR メッセージを受信したリクエストは、次のようなメッセージも受信していました。 Mon Jun 30 21:52:57 2003 general list: trying connect to '10.84.133.182'/7305/7306 at line 1257 for .. このメッセージは、セカンダリ サーバではなく一般リストの別のサーバにフェイルオーバーが適用されていることを示しています。 この問題は、コードを修正して解決されました。

変更要求番号	説明
CR126982	<p>WLExcludePathOrMimeType が設定された場合に、WebLogic Server へのリクエストでそのファイル タイプが削除されましたが、iplanet で代わりにそれらのファイルを提供することができていませんでした。</p> <p>たとえば、.jpg の含まれている .jsp に対する次のリクエストは、</p> <pre data-bbox="377 406 1123 517">&lt;Object name="test5" ppath="*/weblogic/*"&gt; Service fn="wl_proxy" WebLogicHost="lorna" WebLogicPort="7001" PathTrim="/weblogic" Debug="ALL" DebugConfigInfo="ON" WLExcludePathOrMimeType="*.jpg" &lt;/Object&gt;</pre> <p>WebLogic Server にプロキシされ、Iplanet は jpg を提供するのに失敗していました。Iplanet のアクセス ログには、次のメッセージが記録されていました。</p> <pre data-bbox="377 604 1224 1081">10.40.4.117 - - [28/Oct/2003:11:45:34 -0500] "GET /weblogic/images/logo_tm_onwt.jpg HTTP/1.1" 500 305 I get the following in wlproxy.log: Tue Oct 28 11:45:35 2003 ===== new request ===== Tue Oct 28 11:45:35 2003 INFO: SSL is not configured Tue Oct 28 11:45:35 2003 URI=[/hello.jsp] Tue Oct 28 11:45:35 2003 attempt #0 out of a max of 5 Tue Oct 28 11:45:35 2003 general list: trying connect to '10.40.4.117'/7001/7001 at line 1224 for '/hello.jsp' Tue Oct 28 11:45:35 2003 INFO: New NON-SSL URL Tue Oct 28 11:45:35 2003 Going to check the general server list Tue Oct 28 11:45:35 2003 WLS info : 10.40.4.117:7001 recycled? 0 Tue Oct 28 11:45:35 2003 Hdrs from Client:[accept]=[*/*] Tue Oct 28 11:45:35 2003 Hdrs from Client:[accept-language]=[en-us] Tue Oct 28 11:45:35 2003 Hdrs from Client:[accept-encoding]=[gzip, deflate] Tue Oct 28 11:45:35 2003 Hdrs from Client:[user-agent]=[Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0)]</pre> <p>WLExcludePathOrMimeType が設定されている場合は、WebLogic Server がリクエストに対応せず、制御が Web サーバに渡されて、Web サーバでリクエストの処理を継続できなければなりません。</p> <p>この問題は、コードを修正して解決されました。</p>
CR126568	<p>NSAPI プラグインを通じて WebLogic Server に送信された、最後に %0A のある POST リクエストが、正常に処理されていませんでした。このリクエストは本文のストリームに余計なデータを追加し、ヘッダが本文の最後に現れていました。WebLogic Server に直に送信されたリクエストは、正確に処理されていました。</p> <p>HTTP/0.9 応答を正確に検出および処理するようにプラグインのコードを修正することで、この問題は解決しました。</p>

変更要求番号	説明
CR126103	<p>負荷テストにおいて、HP11.00 上で動作する NSAPI が 2 台の Solaris マシンにまたがる 6 ノードのクラスタ (各マシンに 3 つの WebLogic Server インスタンス) にプロキシしているときに、メモリ消費が絶え間なく上昇し、約 50 分後に、ns-httpd プロセスがクラッシュしていました。</p> <p>同じ負荷テストが、HP11.00 または Solaris ではクラッシュしませんでした。</p> <p>分析の結果、プログラムのヒープ領域にシステム メモリを割り当てるネイティブのシステム呼び出し proxy.cpp used strdup() のコードに問題があることが判明しました。WebLogic Server は、Iplanet の FREE マクロを使用して、不要になった割り当て済みの領域を解放します。FREE では strdup() の呼び出しで割り当てられた領域が解放されないため、メモリ リークが発生していました。</p> <p>この問題は、proxy.cpp のすべてのネイティブ strdup() システム呼び出しを Iplanet の STRDUP マクロで置換し、どの領域を解放するのか FREE マクロが指示されるようにすることで解決しました。</p>
CR125690	<p>9 つの IIS サーバとクラスタ化されている 9 つの WebLogic Server インスタンスが含まれているコンフィグレーションで、IIS が数時間おきにクラッシュし、Event 37 がイベント ログに書き込まれていました。wlproxy のログには、次のメッセージが記録されていました。</p> <pre data-bbox="314 869 1157 921">Thu Oct 09 13:01:46 2003 *****Exception type [WRITE_ERROR_TO_CLIENT] raised at line 1269 of .\iisproxy.cpp</pre> <p>診断の結果、初期バッファを拡張しているときに Reader::fill() メソッドが十分なメモリを割り当てていないことが判明しました。バッファの最後を示す 4 バイトが失われ、その結果としてコア ダンプが生じていました。この問題は、コードを修正して解決されました。</p>
CR125545	<p>クライアントが (たとえば応答が完了する前にブラウザを閉じることで) 応答が送信されてくるのを中断した場合に、プラグインが 500 [WRITE TO CLIENT ERROR] を Web サーバのログ ファイルに書き込んでいました。このことが原因で、状態モニタ ツールが 500 エラーを Web サーバの問題として解釈してしまう恐れがありました。この問題は、コードを修正して解決されました。</p>
CR124464	<p>リクエストが HALF_OPEN_SOCKET_RETRY, "Unexpected EOF reading HTTP status - request retry" メッセージを受け取った直後に、プラグインが iPlanet のクラッシュを引き起こす恐れがありました。</p> <p>例外オブジェクトから例外コードを取得し、それからオブジェクトを削除するようにコードが修正されました。</p>



---

変更要求番号	説明
CR124433	IIS で <code>WlForwardPath=/</code> がコンフィグレーションされている場合に、サーバがダウンしていてもプラグインがリクエストを転送しようとしていました。クライアントにエラー ページが提供されることはありませんでした。この状況ではパスが適切に除外されるようにプラグインが修正されました。

---

---

変更要求番号	説明
CR123925	<p data-bbox="314 256 1163 317">プラグインは、500 エラー メッセージでブラウザに応答することがありました。この問題には、以下の 3 つの症状もありました。</p> <ol style="list-style-type: none"><li data-bbox="314 343 1163 1029">1. IIS のアクセス ログが次のメッセージを示していました。<pre data-bbox="353 392 1112 440">Out-of-process+ISAPI+extension+request+failed. 500 1726 99122 2003 84078</pre></li><li data-bbox="314 468 1163 1029">2. Windows のイベント ログが、Event ID 37 を記録していました。<pre data-bbox="353 517 1163 1029">Event Type: Warning Event Source: W3SVC Event Category: None Event ID: 37 Date: 8/26/2003 Time: 6:45:03 PM User: N/A Computer: name Description: Out of process application '/LM/W3SVC/2/Root/caf' terminated unexpectedly. For additional information specific to this message please visit the Microsoft Online Support site located at:http://www.microsoft.com/contentredirect.asp.</pre></li><li data-bbox="314 1055 1163 1345">3. wlproxy.log のエントリが次のように示していました。<pre data-bbox="353 1104 1083 1307">Fri Nov 21 19:06:31 2003 Write to the browser failed: calling URL::close at line 1270 of .\iisproxy.cpp Fri Nov 21 19:06:31 2003 *****Exception type [WRITE_ERROR_TO_CLIENT] raised at line 1271 of .\iisproxy.cpp</pre></li></ol> <p data-bbox="314 1321 827 1345">この問題は、コードを修正して解決されました。</p>

---

変更要求番号	説明
CR123120	<p>POST メソッドがプラグインを通じて使用され、かつ Content-Length が定義されていない場合に、プロキシ ログ ファイルに次のようなメッセージが記録されていました。</p> <pre>POST and PUT requests *must* contain a Content-Length</pre> <p>Content-Length が定義されていない場合に、0 のコンテンツ長が設定されるようにコードが修正されました。</p>
CR122755	<p>? .wforward? が手動で URL に追加された場合に、プラグイン フィルタが迂回されていました。初期リクエストの MIME タイプが .wforward である場合に 404 エラーが送出されるように、コードが修正されました。</p>
CR122754	<p>プラグイン パラメータ WLExcludeByPathOrMimeType が、MIME タイプで転送を行うときに機能していませんでした。この問題は、コードを修正して解決されました。</p>
CR122207	<p>KeepAliveEnabled と DynamicServerList が両方とも有効になっている場合に、プラグインがソケットを CLOSE_WAIT 状態のままにする恐れがありました。この問題は、コードを修正して解決されました。</p>
CR121688、 CR121943	<p>URL で感嘆符 !? が ?%21? で置換されている場合に、プラグインがクッキーの解析に失敗していました。この置換は通常、URL 書き換えの使用時に WAP ゲートウェイによって行われます。URL の文字を正しく解析するように、コードが修正されました。</p>
CR113093	<p>次のように、httpd.conf で複数の MatchExpression パラメータを使用してリクエストをさまざまな場所に転送する場合に、</p> <pre>MatchExpression *.jsp WebLogicHost=localhost WebLogicPort=8001 MatchExpression *.html WebLogicCluster=localhost:8001,localhost:8003</pre> <p>各リクエストが同じグローバルパラメータ情報を上書きし、リクエストが間違った場所に転送されていました。上の例では、*.jsp のリクエストがポート 8003 のサーバに転送されることになります。</p> <p>各リクエストがそれ専用のパラメータ情報を使用するように、コードが修正されました。</p>

## サービス パック 1 から 6 で解決済みの問題

変更要求番号	説明
CR111167	<p>WebLogic Server 7.0 の以前のサービス パックでは、ISAPI プラグインを使用すると、HTTP 応答の Date ヘッダが 2 つになっていました (1 つは WebLogic Server によって挿入され、1 つは IIS で挿入される)。この Date ヘッダの重複は、Date ヘッダは 1 つだと想定している特定のキャッシング サービスで問題を発生させていました。</p> <p>この問題は、WebLogic Server で挿入された Date ヘッダを除去するように ISAPI プラグインを更新することで解決しました。</p>
CR110664	<p>プラグインのコードが例外を捕捉できず、それが原因で iPlanet サーバが <code>sendResponse</code> フェーズでクラッシュしていました。例外が捕捉されるように、プラグインのコードが修正されました。</p>
CR109755	<p>ワイルドカード文字 (*/*) 以外の正規表現が含まれたコンフィグレーション パラメータが、プラグインで無視されていました。この問題は、次のようにしてパラメータを使用するときに 404 エラーを生じされる恐れがありました。</p> <pre>LocationMatch "/weblogic/(abc def)/ghi"</pre> <p>この問題は、コードを修正して解決されました。</p>
CR108092	<p>WebLogic Server 7.0 の以前のサービス パックでは、修正されたクッキーに遭遇したときに ISAPI プラグインが未処理の例外エラーを Windows のイベント ログに記録していました。イベント テキストは、次のような行で始まっていました。</p> <pre>The HTTP Server encountered an unhandled exception while processing the ISAPI Application.</pre> <p>この問題は、ISAPI プラグインのコード修正で解決しました。</p>
CR105123	<p><code>iisproxy.ini</code> ファイルで仮想ディレクトリをコンフィグレーションし、MIME タイプをワイルドカード * (すべてをプロキシ) でコンフィグレーションして、<code>DefaultFileName</code> を追加した場合に、ファイル名のないディレクトリに対するリクエストで、<code>DefaultFileName</code> が使用されていませんでした。</p> <p>コードを変更してこの問題は解決しました。</p>
CR105173、 CR135917	<p>クライアントが (たとえば応答が完全に受信される前にブラウザを閉じることで) 応答が送信されてくるのを中断した場合に、500 [WRITE TO CLIENT ERROR] が Web サーバのログに不適切に記録されていました。</p> <p>この種の応答の失敗で、500 エラーがログに記録されることはなくなりました。</p>

変更要求番号	説明
CR091910	<p>WebLogic Server 7.0 の以前のサービス パックでは、<code>&lt;IfModule mod_weblogic.c&gt;</code> を使用している場合に Apache プラグインが <code>PathPrepend</code> を読み込みませんでした。この問題は、Apache 1.3.x および Apache 2.0.43 用のプラグインで発生していました。この問題は、<code>PathPrepend</code> プロパティと <code>MatchExpression</code> プロパティが <code>&lt;IfModule mod_weblogic.c&gt;</code> タグの中で共存している場合にのみ発生していました。<code>PathTrim</code> プロパティでも同じことが起こっていました。</p> <p>この問題は、コードを修正して解決されました。</p>
CR106764	<p>Netscape プラグインのスレッドは長い時間 ( デフォルトは 5 分、<code>WLIOWaitTimeoutSecs</code> でコンフィグレーションすることも可能 ) 重要なロックを取得することができたので、他のすべてのスレッドがブロックされて、Web サーバがハングしたようになっていました。この問題は、プラグインのコードを修正することで解決しました。</p>
CR087204	<p><code>PathTrim</code> パラメータと <code>PathPrepend</code> パラメータを一緒に使用しても、末尾に不要な <code>/</code> の付いた URL が作られなくなりました。</p>

## RMI

変更要求番号	説明
CR124596	<p>BEA ORB に追加されたオプションの機能により、ブートストラップ時に再接続が行われ、ハードウェア ロードバランサが接続の試行を適切にバランシングできるようにになりました。</p> <p>この機能と制限の詳細については、『WebLogic RMI over IIOP プログラマーズガイド』の「<a href="#">ハードウェア ロード バランサと RMI over IIOP の併用</a>」を参照してください。</p>
CR125261	<p>ファット クライアント アプリケーションが、サービス パック 3 およびサービス パック 4 を適用した WebLogic Server 7.0 を使用している場合に、ステートレスセッション EJB を呼び出すクライアントをインスタンス化し、メモリ リークを起していました。</p> <p>この問題は、Appmanager のコンテキスト クラスローダを使用することで解決しました。この変更後、<code>-Dweblogic.LoadStubUsingContextClassLoader</code> オプションは使用できなくなります。</p>
CR127333	<p>サーバが応答を書き込んでいる途中で RMI/IIOP クライアントが強制停止されたときに、それ以降の一部のクライアント要求が <code>MarshalException</code> で失敗していました。初期クライアント接続が切断された場合に、間接マップがプールに戻されるときにクリアされていませんでした。その特定のマップをプールから使用する以降のクライアントは、<code>MarshalException</code> で失敗していました。</p> <p>コードを修正した結果、クライアントが強制的に終了された場合でも間接マップが適切にクリアされるようになりました。</p>
CR121079	<p>最初のリクエストを行うときに、WebLogic Server-IIOP ランタイムがワイド文字列の送信に適切なコードセットを選択していませんでした。</p> <p>現在は、すべてのリクエストでワイド文字列の送信に適切なコードセットが選択されます。WebLogic Server との相互運用が正しく機能するようになっています。</p>
CR128594	<p>以前のサービス パックでは、WebLogic Server は AIX 上の Java オブジェクトを読み込むときにタイプコードのエイリアスを処理できていませんでした。エイリアス ラッパーを破棄するように、コードが修正されました。</p>

---

変更要求番号	説明
CR124377	<p>シンクライアント.jar (wlclient.jar) を使用しているクライアント アプリケーションが EJB にアクセスしたときに、WebLogic Server は <code>java.rmi.UnmarshalException</code> を送出することがありました。次に示すのは、サーバ側の例外の一部です。</p> <pre>java.rmi.UnmarshalException: error unmarshalling arguments; nested exception is: java.io.IOException: Serializable readObject method failed internally.java.rmi.UnmarshalException: error unmarshalling arguments; nested exception is: java.io.IOException: Serializable readObject method failed internally at com.ejb_cvps36_EOImpl_WLSkel.invoke(Unknown Source) [...]</pre> <p>次に示すのは、クライアント側の例外の一部です。</p> <pre>java.rmi.MarshalException: CORBA MARSHAL 0 No; nested exception is:org.omg.CORBA.MARSHAL: vmcid: 0x0 minor code: 0 completed: No at com.sun.corba.se.internal.iiop.ShutdownUtilDelegate.mapSystemE xception(ShutdownUtilDelegate.java:97) at javax.rmi.CORBA.Util.mapSystemException(Util.java:65) [...]</pre> <p>この問題は、クライアントで <code>weblogic.jar</code> を使用したときには発生しませんでした。この問題に対応するように、コードが修正されました。</p>
CR109844、 CR113715	<p><code>InitialContext</code> の作成とクローズが繰り返されているときの <code>DGCServerHelper</code> クラスのメモリ リークに起因するメモリ不足エラーで、分散ガベージ コレクションを管理するヘルパー オブジェクトが不必要に作成されます。</p> <p>この問題は、コードの変更で解決されました。</p>

---

---

変更要求番号	説明
CR106281	<p>WebLogic Server 7.0 の以前のサービス パックでは、負荷が大きいときにレプリケートされたセッション Bean が原因で、ヒープの使用量が上昇し、OutOfMemoryErrors が生じていました。</p> <p>分析の結果、ガベージ コレクションが強制されたときでも <code>examples.ejb20.basic.statefulSession.TraderBean_5ysgq2_EOImpl</code> オブジェクトがガベージ コレクションされていなかったことが判明しました。クラスタ化されたステートフル セッション Bean では、<code>weblogic.rmi.cluster.PrimarySecondaryRemoteObject</code> の <code>EOImpl</code> オブジェクト プライマリへの強い参照が保持されていました。Bean で <code>remove</code> が呼び出されないので、<code>EOImpl</code> の参照が <code>eoMap</code> から削除されることはありませんでした。</p> <p><code>session-timeout-seconds</code> が経過した後、パッシベーションされた Bean が削除されたときに EO がアンエクスポートされるように、コードが修正されました。</p>
CR131692	<p>ejb メソッドから派生例外を送出するときに、iiop レイヤが基本例外ではなく派生例外の Full Value Description を追加していました。クライアントが戻って基本例外の情報を取得するときに、サーバはその情報を持たず、例外を送出していました。</p> <p>派生クラスの Full Value Description を作成するときに、基本クラスの Full Value Description も作成されるようにコードが追加されました。</p>

---



# セキュリティ

変更要求番号	説明
CR135710	<p>backendstandard には、アンデプロイなどの一部の操作を遅延させる <code>wait(1000)</code> がありました。</p> <p>コードを変更して <code>wait(1000)</code> を削除し、<code>tp.wait()</code> を導入しました。</p>
CR172774	<p>以下のように、LDAP で証明書を保持するバイナリ フィールド ("userCertificate" または "userCertificate;binary") に合わせて <code>CertificateAttribute</code> を定義する必要があります。現在のデフォルトは "userCertificate" です。</p> <p>1. userCertificate</p> <p>LDAP ブラウザから証明書をロードする場合、バイナリ型の属性 <code>userCertificate</code> が作成されます。この属性の値は証明書のデータです。証明書にアクセスするには <code>CertificateAttribute="userCertificate"</code> を定義する必要があります。</p> <p>2. userCertificate;binary</p> <p>次のように <code>ldapmodify</code> を使用して新しい属性を追加する場合、</p> <pre>ldapmodify -p 1155 -D Principal -w Password dn: cn=support@bea.com,ou=Certs,dc=bea,dc=com changetype: modify add: userCertificate userCertificate;binary:: MIICxDCCAI2gAwIBAgIBIDANBgkqh..... userCertificate;binary という属性が作成されて、そこに証明書データがロードされます。この証明書にアクセスするには CertificateAttribute="userCertificate;binary" を定義する必要があります。</pre>
CR121114、 CR134158、 CR134159、 CR135017、 CR130506	<p><code>SocketImpl</code> クラスが <code>IOException</code> を捕捉せず、再送出を続けた結果、ソケットが閉じなくなっていました。</p> <p>この問題は、ソケットを閉じ、<code>IOException</code> を送出するコードを追加することで解決しました。この変更の結果、アイドル状態のソケットは存在しなくなりました。</p>

## サービス パック 1 から 6 で解決済みの問題

変更要求番号	説明
CR120748	以前のリリースの WebLogic Server には、LDAP X509 ID アサータがありませんでした。 LDAP X509 ID アサータの使用を許可するコードが追加されました。
CR112147	weak、strong、authenticate などの既存の ServletAuthentication API メソッドが、LoginException を呼び出し元に伝達していませんでした。weak および authenticate メソッドと同じように機能する、2 つのオーバーライドされたログイン メソッドが追加されました。strong メソッドと同じように機能する assertIdentity メソッドも追加されています。これらの新しいメソッドは、LoginException を呼び出し側コードに伝達します。
CR124746、 CR175051、 CR175045	のセキュリティ勧告に関する情報を確認してください。
CR132503	Web サービスのクライアントはローカル ID を直接 Certicom SSL コンテキストにロードしていましたが、Web サービスの <code>HttpsURLConnection</code> クラスは接続をその特定の Certicom SSL コンテキストと関連付けていませんでした。 接続は現在は、 <code>HttpsURLConnection</code> クラス (SSL コンテキストをパラメータとして取る) によって SSL コンテキストと関連付けられるようになりました。
CR130355	重複したエントリを追加したときに、例外が無害であっても、メソッドを完了することなく <code>DefaultCredentialMapperLDAPHelper</code> が例外を送出していました。 コードの修正によって、 <code>DefaultCredentialMapperLDAPHelper</code> が重複エントリの追加時に例外を無視するようになりました。
CR128002	接続プールが正しく管理されていず、プールのサイズがリクエストの生成する負荷に対して十分ではありませんでした。これらの問題が原因で、ユーザの認証時にパフォーマンス上の問題が生じていました。 プールの問題は、コードの修正で解決しました。さらに、LDAP 接続プールのサイズを設定するオプションも追加されました。そのオプションは、 <code>-Dweblogic.security.providers.authentication.LDAPDelegatePoolSize</code> です。

変更要求番号	説明
CR127259	<p>DefaultAuditRecorder のバックアップ ファイル名の月フィールドで、間違っ た値が示されていました。</p> <p>表現を修正するコードが追加されました。現在は、正しい値が示されるよう なっています。</p>
CR125409	<p>2 つの管理対象サーバが同時に起動された場合に、アプリケーションのデプロイ の過程で LDAP エントリを作成するために同じコードが 2 回実行されていま した。同期がなされていないために、LDAP で重複エントリが作成されていま した。</p> <p>id カウンタを更新する前に同期化の過程でエントリの有無をチェックするコード が追加されました。重複エントリが誤って作成されている場合に、WebLogic Server はエントリの作成を防止する例外を送出します。</p>
CR064593、 CR121607	<p>SSL の存続期間とキャッシュ サイズの制御に使用される以下のコマンドライン 引数は、WebLogic Server 7.0 で無視されなくなりました。</p> <pre data-bbox="370 748 951 861">-Dweblogic.security.SSL.sessionCache.size= sessioncachesize -Dweblogic.security.SSL.sessionCache.ttl= sessioncachetimetolive</pre>
CR097734、 CR090472	<p>リクエストをプロキシするときに SSLLayeredSocket が SSLIOContext を作成 せず、プロキシ サーバを通じて SSL のトンネリングを行っている間に NullPointerException が生じていました。</p> <p>SSLIOContext を作成し、それをコンテキストテーブルに追加するコードの変 更で、この問題は解決しました。</p>
CR105933	<p>接続フィルタのルールで LDAP プロトコルが許されていなかったために、 WebLogic Server は LDAP サーバからの接続をフィルタ処理することができま せんでした。</p> <p>フィルタ処理可能なプロトコルのリストに LDAP が追加されたので、ldap から の接続がフィルタ処理可能になりました。</p>
CR106192	<p>セキュリティ デバッグ フラグ &lt;ServerDebug DebugSecurityAtn="true" DebugSecurityAtz="true" ...&gt; を StdoutDebugEnabled="true" StdoutSeverityLevel="64" と一緒に設定した場合に、ユーザのパスワードが クリア テキストで stdout に記録されていました。</p> <p>この問題は、ログ ファイルに書き込まれる文からパスワード文字列を削除する ことで解決しました。</p>

変更要求番号	説明
CR106294、 CR132596、 CR120273	埋め込み LDAP トランザクションの処理で、コンフィグレーションされた待機時間だけアプリケーションの削除が遅延されていました。コードを変更してこの問題は解決しました。
CR106532	<p>双方向 SSL が有効になっている場合、サーバは IdentityAssertion を自動的に呼び出します。これはクライアント / サーバのシナリオでは適切ですが、接続が WebLogic Server のインスタンス間の場合は問題が生じます。そのシナリオは次のとおりです。</p> <p>クライアントが WebLogic Server のインスタンス 1 にアクセスし、双方向 SSL で適切に認証されます。インスタンス 1 はインスタンス 2 の EJB を双方向 SSL で呼び出し、インスタンス 2 はインスタンス 1 の証明書に基づいて認証を行います。以下の 2 つの問題があります。</p> <ol style="list-style-type: none"><li>1. 真のユーザはインスタンス 1 ではなくクライアントであるのに、インスタンス 2 ではインスタンス 1 の証明書の DN に基づいてユーザはインスタンス 1 だと認識します。</li><li>2. 認証が必須であってはなりません。SSL 接続ではクライアント証明書を強制できなければなりません、信頼性のある WebLogic Server ドメインを使用する場合には IdentityAssertion チェックを無効にします。</li></ol> <p>コードの修正によって、証明書の認証がコンフィグレーション可能になりました。双方向 SSL の過程で提示された証明書を処理して IdentityAssertion を実行したくない場合は、サーバの起動時に</p> <p><code>-Dweblogic.security.disableIdentityAssertion=true</code> を使用します。</p>
CR107363	<p><a href="http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA03_40.00.jsp">http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA03_40.00.jsp</a> のセキュリティ勧告を参照してください。</p>

変更要求番号	説明
CR108021	<p>セキュリティ サブジェクトと一緒に格納された情報の損失を引き起こしていた問題が、次のようにして解決されました。</p> <p>双方向の証明書あるいはヘッダまたはクッキーのトークンを使用してユーザがログインするときに新しいキャッシュが使用されて、ID アサーションのパフォーマンスが向上します。サーバは 6.x 以前の WebLogic Server インスタンスから、または run-as タグ付きのサーブレットまたは EJB が実行されたときに RMI 呼び出しを受信します。ユーザが削除されても、キャッシュの TTL に達するまでその削除されたユーザについて ID アサーションが継続して行われます。キャッシュの TTL のデフォルトは 5 分であり、次のようにしてコマンドラインで調整できます。</p> <pre data-bbox="370 597 989 730">-Dweblogic.security.identityAssertionTTL=N for N &gt; 0      sets the cache TTL to N seconds N = 0      sets the cache TTL to infinity N &lt; 0      disables the cache altogether</pre>
CR108624、 CR128228	<p>パフォーマンス チューニングの拡張機能として、グループ メンバーシップ検索の深さを制限するための新しい属性が追加されました。メンバーシップの階層に基づいて検索を調整し、グループ メンバーが見つからないことがわかっている検索を排除できます。</p> <p>新しい属性は、GroupMembershipSearching と MaxGroupMembershipSearchLevel です。</p>
CR110242	<p>realm.getGroups() の呼び出しで使用される LDAPConnection がその結果を保持し、getGroups() の繰り返し処理でも使用されるプールへ復帰していませんでした。これが原因で、開いている LDAP 接続が急増していました。</p> <p>コードの修正後、LDAP 接続はグループ イテレータに対する getGroups() 呼び出しに関連付けられ、イテレータにそれ以上要素がなくなった時点で閉じるようになりました。</p>

---

変更要求番号	説明
CR110977	<p>ACLCacheTTLPositive で <code>java.lang. Integer.MAX_VALUE</code> を指定すると、<code>java.lang. IllegalArgumentException</code> が送出されていました。この問題は、イベントが次の順序で行われた後に発生していました。</p> <ol style="list-style-type: none"><li>1. CompatibilityRealm モードで WebLogic Server を起動します。</li><li>2. キャッシング レルムで、ACL キャッシュを有効にし、ACL Cache Positive TTL = <code>java.lang.Integer.MAX_VALUE</code> の値を指定し、変更を保存します。</li><li>3. WebLogic Server を再起動します。</li></ol> <p>WebLogic Server から次の例外が送出されます。</p> <pre>&lt;Jun 25, 2003 4:31:44 PM PDT&gt; &lt;Notice&gt; &lt;Management&gt; &lt;140005&gt; &lt;Loading configuration D:\opt\weblogicapps\ecommerce\.\config.xml&gt; &lt;Jun 25, 2003 4:31:47 PM PDT&gt; &lt;Info&gt; &lt;Logging&gt; &lt;000000&gt; &lt;FileLogger Opened at D:\opt\weblogicapps\ecommerce\.\logs\ecommerce\myserver.log&gt; &lt;Jun 25, 2003 4:31:50 PM PDT&gt; &lt;Critical&gt; &lt;WebLogicServer&gt; &lt;000364&gt; &lt;Server failed during initialization. Exception:java.lang.IllegalArgumentException: ttl&lt;= 0 java.lang.IllegalArgumentException: ttl &lt;= 0 at weblogic.security.acl.TTLCache.&lt;init&gt;(TTLCache.java:195) at weblogic.security.acl.TTLCache.&lt;init&gt;(TTLCache.java:173) at weblogic.security.acl.CachingRealm.&lt;init&gt;(CachingRealm.java:47 8) at weblogic.security.acl.CachingRealm.&lt;init&gt;(CachingRealm.java:37 5) at weblogic.security.acl.CachingRealm.&lt;init&gt;(CachingRealm.java:35 0) at ...</pre> <p>この問題は、<code>ttlcache</code> の作成コードを修正することで解決しました。</p>
CR111305	<p>ログインしたクライアントが Web アプリケーション リソースにその表示前にアクセス可能かどうかを、<code>webflowCheckAccess()</code> メソッドが明示的にチェックするようになりました。</p>
CR112971、 CR104667	<p>WebLogic Server 7.0 の以前のサービス パックと一緒にリリースされていた NetScape LDAP API に、複数のスレッドが <code>LDAPConnThread</code> の <code>deregister()</code> を呼び出したときにデッドロックを生じさせるバグがありました。</p> <p>サービス パック 5 のコード変更で、この問題は解決しました。</p>

---

変更要求番号	説明
CR113459	<p>以前のサービス パックでは、ノード マネージャのプロパティ ファイルを削除すると問題が起きていました。ノード マネージャのプロパティ ファイルは常に、<code>&lt;saved logs dir&gt;/NodeManagerInternal</code> ディレクトリに作成されます。NodeManagerInternal の内容を定期的に保管および削除することでノード マネージャのプロパティ ファイルが削除されるので、ノード マネージャのプロパティ ファイルに格納された証明書のパスワードを解読できず、ノード マネージャが正しく機能していませんでした。</p> <p>ノード マネージャのプロパティ ファイルを <code>user.dir</code> で指定されたディレクトリ (ファイルが NodeManagerInternal がない場合) または <code>user.dir</code> ディレクトリに作成するようにコードを修正することで、この問題は解決しました。</p>
CR120233	<p>IPlanet 5.1 Authenticator を使用してカスタム監査プロバイダおよびカスタム ロール マッピング プロバイダをコンフィグレーションするときに <code>java.lang.ArrayStoreException</code> が送出される問題が、コードの修正によって解決されました。</p>

---

変更要求番号	説明
CR120850	<p data-bbox="307 256 1166 418">weblogic.net.http.HttpURLConnection が、https.nonProxyHosts 環境変数を無視していました。WebLogic Server 内で動作しているか、スタンドアロンの Java プログラムであるかに関係なく、クライアントからのリクエストが、https.nonProxyHosts で対象ホストが指定されている場合でも常にプロキシを経由していました。</p> <p data-bbox="307 430 1166 621">https.nonProxyHosts が SSLParams で検出されると、プロキシされるべきホストの内部リストが作成されます。SSLSocket が作成および初期化されると、ホストがこのリストに照らし合わせて比較されます。ホストがリストにある場合、そのホストはソケットを開くために使用されます。ホストがリストにない場合は、proxyHost を使用してソケットが開かれます (proxyHost がない場合はすべて無効であり、ホストを使用してソケットが開かれる)。</p> <p data-bbox="307 633 1166 696">コードを変更してこの問題を解決しました。現在は、ユーザが以下のプロパティを設定できるようになっています。</p> <ul data-bbox="307 708 846 916" style="list-style-type: none"><li>• http.proxyHost</li><li>• http.proxyPort</li><li>• http.nonProxyHosts</li><li>• https.proxyHost</li><li>• https.proxyPort</li><li>• https.nonProxyHosts (新しいプロパティ)</li></ul> <p data-bbox="307 928 1166 1020">http と https のどちらの場合でも、ソケットが開かれる proxyHost を定義できます。しかし、proxyHost が定義されている場合でも、直接接続されるべきホストのリスト (nonProxyHosts) を定義することもできます。</p> <p data-bbox="307 1032 1166 1124">https.nonProxyHosts が追加されたことで、https.proxyHost が定義されている場合でも、ssl を使用して常に直接接続されるべきホストを指定することができるようになりました。</p>

---



変更要求番号	説明
CR120852	<p>動的グループ属性 memberURL の文字列値で、LDAP URL の LDAP 検索フィルタ部分に予約文字 (スラッシュ「/」など) が含まれている場合に、MalformedURLException が送出されていました。たとえば、次の URL では、</p> <pre>ldap://text_replaced:389/dc=text_replaced??sub?(&amp;( (objectclass=person)objectclass=groupofuniquenames))(uid=*text_replaced/2*)</pre> <p>次の例外が送出されていました。</p> <pre>&lt;Jun 23, 2003 12:30:35 PM EDT&gt; &lt;Debug&gt; &lt;SecurityDebug&gt; &lt;000000&gt; &lt;advance dyn group entry = LDAPEntry: cn=slashgroup,ou=Groups,dc=text_replaced; LDAPAttributeSet: LDAPAttribute {type='cn', values='dynamicgroup001,slashgroup'} LDAPAttribute {type='memberURL', values='ldap:///dc=text_replaced??sub?(&amp;( (objectclass=person)(objectclass=groupofuniquenames))(uid=*htext_replaced/2*)'}&gt; &lt;Jun 23, 2003 12:30:35 PM EDT&gt; &lt;Debug&gt; &lt;SecurityDebug&gt; &lt;000000&gt; &lt;advance member url = ldap:///dc=text ...</pre> <p>分析と調査の結果、以下のことが判明しました。</p> <ul style="list-style-type: none"> <li>LDAP URL で「/」は不正ではないが、URL の区切り文字なので避ける必要がある。</li> <li>WebLogic Server は、動的グループの繰り返し処理の間に LDAPUrl オブジェクトを使用してフィルタを取得していた。その url フィルタ部分に「/」のような特殊文字が含まれている場合、LDAPUrl クラスはそれらの文字を適切に処理しません。</li> </ul> <p>LDAPRealm オブジェクトを使用せずにフィルタを取得するようコードを修正して、この問題は解決しました。</p>
CR120932	<p>CertificateServlet で提供された HTML ページが、ラジオ ボタンに間違っただけの値を表示していました。ラジオ ボタンで 2048 が表示されていましたが、そのボタンを使用すると 1024 ビットの証明書になっていました。この問題は、コードの変更で解決されました。</p>
CR121043、 CR134995	<p>JSP で直接設定されたプロパティが、HTTP クライアントで取得および設定されていませんでした。</p> <p>コードを変更してこの問題は解決しました。</p>

変更要求番号	説明
CR121135、 CR123865	<p>LANG が <code>univ.utf8</code> に設定されている場合に、特定の状況で 2 つの WebLogic Server インスタンス間の双方向 SSL 通信を失敗させる内部例外が、JDK 1.3.1_08 から送出されていました。</p> <p>コードを変更してこの問題は解決しました。</p>
CR121646	<p>Extended Key Usage が <code>critical</code> に設定されているクライアント証明書が WebLogic Server に送信されたときに、<code>BAD_CERTIFICATE</code> エラーが受信され、SSL 接続が終了していました。</p> <p>分析の結果、WebLogic Server が SSL の Enhanced KeyUsage をサポートしていないことが判明しました。</p> <p>この問題は、EnhancedKeyUsage のサポートを追加することで解決しました。現在は、Enhanced Key Usage が <code>critical</code> に設定されている証明書を WebLogic Server が受け入れるようになっています。</p>
CR122642	<p>サービス パック 2 からサービス パック 3 にアップグレードした後に、[セキュリティ   ユーザ   レalm   myrealm   ユーザ] 画面を表示しようとすると、WebLogic Administration Console がハングしていました。</p> <p>サービス パック 5 のコード変更で、この問題は解決しました。</p>
CR124164、 CR090738	<p>資格エンジンは以前は、ロール式のオペランドの数を 100 に制限していました。この制限は、さまざまな属性値を制限していました。たとえば、<code>weblogic-ejb-jar.xml</code> の <code>security-role-assignment</code> 内の <code>role</code> にマップできる <code>principal-name</code> 要素の最大数は 50 に制限されていました。</p> <p><code>principal-name</code> 要素が 50 を超えると、アプリケーションのデプロイメント時に <code>weblogic.entitlement.data.EnCreateException</code> が送出されていました。</p> <p>ロール式のオペランド数が制限されないようにコードを修正することで、この問題は解決しました。</p>

変更要求番号	説明
CR126106、 CR090472	アプリケーションのデプロイメント時に、LDAP のエントリがまず管理サーバに書き込まれ、その後に管理対象サーバに書き込まれていました。以前のサービスパックでは、管理サーバへの書き込みが失敗した場合に、管理対象サーバへの書き込みなしに呼び出しが戻され、それが原因で EJB セキュリティ レイヤが適切なロールを返さなくなっていました。 LDAP のコードを修正することで、この問題は解決しました。
CR133655、 CR128002	接続プールのコードが正しく機能せずに、LDAP 検索の速度が低下し、それが原因でたとえばサードパーティ LDAP との認証が遅くなっていました。 コードを変更してこの問題を解決しました。

## サブレット

変更要求番号	説明
CR128624、 CR171907	<p><a href="http://localhost:7001/WebApp/target.jsp#section4">http://localhost:7001/WebApp/target.jsp#section4</a> のように HTML アンカー タグが含まれている URL をアプリケーションでエンコードしようとしたときに、結果の URL は</p> <p><a href="http://localhost:7001/WebApp/target.jsp#section4;jsessionid=16OXDN2vax5g2lbGucG4uspB9h6vzhaZw8KtDL15urAhK96dqvfo!-1835542719">http://localhost:7001/WebApp/target.jsp#section4;jsessionid=16OXDN2vax5g2lbGucG4uspB9h6vzhaZw8KtDL15urAhK96dqvfo!-1835542719</a> のように間違っただけになっていました。</p> <p>アンカー タグは、無視されるように URL の末尾にあるべきです。正しい URL は、</p> <p><a href="http://localhost:7001/WebApp/target.jsp;jsessionid=16OXDN2vax5g2lbGucG4uspB9h6vzhaZw8KtDL15urAhK96dqvfo!-1835542719#section4">http://localhost:7001/WebApp/target.jsp;jsessionid=16OXDN2vax5g2lbGucG4uspB9h6vzhaZw8KtDL15urAhK96dqvfo!-1835542719#section4</a> のようになります。</p> <p>オリジナルの URL にクエリ文字列がない場合にアンカーを探し、エンコードされた URL の末尾にそのアンカーを付けるように、コードが追加されました。</p>
CR123333	<p>Web アプリケーションのフォームベースの認証は、「j_username」および「j_password」でマルチバイト文字を許可せず、それによって JSP でのリクエストエンコーディングが妨げられていました。マルチバイト文字が原因で認証が失敗していました。</p> <p>JSP で文字エンコーディングを指定するには、「j_character_encoding」を次のように使用します。</p> <pre>&lt;form method="POST" action="j_security_check"&gt; &lt;input type="text" name="j_username"&gt; &lt;input type="password" name="j_password"&gt; &lt;input type="hidden" name="j_character_encoding" value="Shift_JIS" &gt; &lt;/form&gt;</pre> <p><b>注意：</b> j_character_encoding は J2EE に準拠していません。 コードを変更してこの問題は解決しました。</p>

変更要求番号	説明
CR132522	<p>デフォルト Web アプリケーションのない Web サーバで、見つからないリソースに対する HTTP リクエストが次のように間違っただけの日付ヘッダのある応答を受信していました。</p> <pre>HTTP/1.1 404 Not Found Date: Thu, 01 Jan 1970 00:00:00 GMT</pre> <p>このヘッダは、RFC2616 のセクション 14.18 によれば妥当ではありません。コードを変更してこの問題を解決しました。</p>
CR129237	<p><code>HttpServletRequest.getServletPath()</code> または <code>HttpServletRequest.getPathInfo()</code> を呼び出したサーブレットは、<code>ServletMapping</code> が <code>*.&lt;value&gt;</code> に対応していて、URL にピリオド「.」が含まれている場合に間違っただけの値を受け取っていました。</p> <p>解析エラーを修正することで、この問題は解決しました。</p>
CR129211	<p><code>ServletOutputStream.write(byte)</code> を使用してバッファ サイズ以上を書き込むと、無限ループが生じていました。</p> <p>バッファが一杯で自動フラッシュが <code>false</code> に設定されている場合の境界条件のチェックを更新するコードの修正で、この問題は解決しました。</p>
CR128986	<p>プロトコル例外の後に、<code>ensureContentLength</code> メソッドの問題が原因で新しいライセンスを処理しようとしているときにサーバがハングしていました。</p> <p>コードを変更してこの問題を解決しました。</p>
CR128420	<p><code>breakUpAndWriteItOutAsNecessary()</code> が、1 行に最大 72 バイトの制限を実施するためにマニフェスト ヘッダを切り離そうとし、1 度に 1 行ずつ出力ストリームに書き込んでいました。この問題は、新しい行の開始オフセットが間違っていたことが原因でした。</p> <p>コードを変更してこの問題を解決しました。</p>
CR127708	<p><code>HttpParsing.unescape()</code> が、パス <code>/foo/..%c0%/..</code> を <code>/foo/..</code> にデコードしていました。その理由は、<code>sun.io.ByteToCharUTF8.convert()</code> が、UTF8 で有効ではないバイト (<code>0xc0</code> など) に遭遇した場合に残りのバイトの変換を止めてしまうからです。この問題は、JDK 1.4 では発生していませんでした。</p> <p>この問題は、無効な UTF-8 シーケンスを <code>U+FFFD</code> 文字で置換する <code>WebLogic Server UTF-8 コンバータ</code> によって解決しました。この修正の結果、<code>HttpParsing.unescape()</code> はパス <code>/foo/..%c0%/..</code> を <code>/foo/..?%..</code> にデコードします。</p>

## サービス パック 1 から 6 で解決済みの問題

変更要求番号	説明
CR125846	<p>サーブレット仕様によれば、ワイルドカード パターンよりも完全パターンの方が優先されなければなりません。しかし、これは正確に機能していませんでした。たとえば、「/TestPath/*」が「WildcardServlet」に対応し、「/TestPath」が「ExactMatchServlet」に対応する場合に、入力の相対 URI が「/TestPath」の場合は、WildcardServlet ではなく ExactMatchServlet が提供されなければなりません。</p> <p>この問題は、パターン マッチングを適切に変更することで修正されました。</p>
CR125048	<p>server/bin/fastfile.dll からのデバッグ情報が、Administration Console で出力されていました。</p> <p>ソース コードを変更することなくバイナリを再構築することで、この問題は解決しました。</p>
CR133558	<p>null 値が渡されたときに、ServletContext.setAttribute が NullPointerException を送出していました。</p> <p>この問題は、setAttribute() に null 値が渡されたときに removeAttribute() を呼び出すようにするコード変更で解決しました。</p>

変更要求番号	説明
CR084455	<p>ローテーション時にすべての HTTP リクエストを拡張フォーマットでログに記録するように WebLogic Server がコンフィグレーションされている場合に、ローテーションがスケジュールされた後に約 1 分間隔で次のエラーが送出されました (1 分はログ フラッシュの時間間隔の設定)。</p> <pre data-bbox="370 409 1231 939"> #####&lt;Aug 20, 2002 9:00:00 PM PDT&gt; &lt;Error&gt; &lt;HTTP myapp&gt; &lt;host2&gt; &lt;myserver&gt; &lt;ExecuteThread: '6' for queue: 'default'&gt; &lt;system&gt; &lt;&gt; &lt;000000&gt; &lt;Exception flushing HTTP log file&gt; java.io.IOException: Failed to rename log file on attempt to rotate logs at weblogic.servlet.logging.LogManagerHttp.rotateLog(LogManagerHttp.java:168) at weblogic.servlet.logging.LogManagerHttp.access\$2(LogManagerHttp.java:148) at weblogic.servlet.logging.LogManagerHttp\$RotateLogTrigger.trigger(LogManagerHttp.java:432) at weblogic.time.common.internal.ScheduledTrigger.executeLocally(ScheduledTrigger.java:238) at weblogic.time.common.internal.ScheduledTrigger.execute(ScheduledTrigger.java:229) at weblogic.time.server.ScheduledTrigger.execute(ScheduledTrigger.java:69) at weblogic.kernel.ExecuteThread.execute(ExecuteThread.java:139) at weblogic.kernel.ExecuteThread.run(ExecuteThread.java:120) #####&lt;Aug 20, 2002 9:00:05 PM PDT&gt; &lt;Error&gt; &lt;HTTP myapp&gt; &lt;host2&gt; &lt;myserver&gt; &lt;ExecuteThread: '7' for queue: 'default'&gt; &lt;system&gt; &lt;&gt; &lt;000000&gt; &lt;Exception flushing HTTP log file&gt;... </pre> <p>分析の結果、ログが不適切にフラッシュされていたことが判明しました。この問題は、ローテーション時にログがフラッシュされず、ログ ファイル名で null 値がチェックされるようにコードを修正することで解決しました。</p>
CR088785、 CR122315	<p>HTTP アクセス ログで、cs-uri フィールドが指定されている場合に、URI の基本部分とクエリ部分の両方ではなく基本部分のみが記録されていました。この問題は、コードを修正して解決されました。</p>

変更要求番号	説明
CR092625、 CR112910	<p>HTTP のロギングを無効にして起動され、かつログ ファイルが存在していない管理対象サーバで HTTP のロギングを有効にした場合に、WebLogic Server が <code>NullPointerException</code> を送出していました。その管理対象サーバへの各 HTTP アクセスで、次の例外が送出されていました。</p> <pre>java.lang.NullPointerException at weblogic.servlet.logging.LogManagerHttp.log(LogManagerHttp.java:292) at weblogic.servlet.internal.HttpServer.log(HttpServer.java:865) at weblogic.servlet.internal.ServletResponseImpl.send(ServletResponseImpl.java:1044) at weblogic.servlet.internal.ServletRequestImpl.execute(ServletRequestImpl.java:2265) at weblogic.kernel.ExecuteThread.execute(ExecuteThread.java:139) at weblogic.kernel.ExecuteThread.run(ExecuteThread.java:120)</pre> <p>この問題は、コードを修正して解決されました。</p>
CR095945	<p><code>CGIServlet</code> は、WAR Web アプリケーションの CGI スクリプトを、それらが実行できるように抽出します。以前、サーブレットは現在の作業ディレクトリを設定せずにそれらのスクリプトを呼び出していました。つまり、サブスクリプトを適切に呼び出すことができていなかったのです。コードの変更によって、現在の作業ディレクトリが設定されるようになり、CGI スクリプトが WAR Web アプリケーションのサブスクリプトを呼び出せるようになりました。</p>
CR096459	<p>HTTP 1.0 でサーブレットまたは JSP のフラッシュを実行するときに、WebLogic Server がときどきソケットを閉じることができず、ソケットがタイムアウトになるまでクライアントが待機するようになっていました。この問題は、コードを修正して解決されました。</p>
CR103339	<p>サーブレットの出力で、<code>Transfer-Encode</code> タイプの値として、HTTP/1.1 仕様で要求されている小文字の <code>chunked</code> ではなく、先頭を大文字にした <code>Chunked</code> が使用されていました。小文字の「<code>chunked</code>」が <code>Transfer-Encode</code> タイプとして使用されるように、コードが修正されました。</p>
CR103482、 CR120943、 CR129041	<p>セカンダリ サーバの孤立セッションの問題をそのセッションを無効化することで解決した回避策が、負荷テスト時に「プライマリが見つからない」エラーを生じさせる競合状態を作り出していました。</p> <p>コードを変更してこの問題を解決しました。</p>



変更要求番号	説明
CR103925、 CR125206	setAttribute メソッドは、hashCode の同一性のみをチェックしていました。現在は、equals メソッドの結果で古いオブジェクトと新しいオブジェクトをチェックするようにもなりました。
CR104245	<p>負荷が大きい状況で、2 サーバ コンフィグレーションのプライマリ サーバとセカンダリ サーバの両方が同じレプリケート セッションにアクセスすると、セッション レプリケーションのデッドロックが発生します。両方のサーバが同じセッション データをレプリケートしようとし、プライマリ ロールのサーバは、ReplicatedSessionData をロックして、セカンダリ サーバを更新しようとし、同時に、もう一方のサーバも同じことを実行し、最初のサーバがそのセカンダリ コピーを更新するのを待ちます。レプリケーション呼び出しは最初のサーバのレプリケーション スレッドキューに入り、同じセッション データのプライマリとして機能している同じサーバの別のスレッドによって保持されたセッション データのロックを取得できるまで更新を終了できません。各サーバはプライマリとしてレプリケーションを試行し、セカンダリとして同じセッションを更新しようとし続けて、相互のデッドロックに陥ります。</p> <p>サーバがデッドロックに陥ると、セッションのアクセスとレプリケーションが関わるすべてのリクエストがブロックされ、リクエストが蓄積されていきます。2 つのサーバがプライマリとセカンダリの状態を行き来し続けると、「Got stale replication request」がログに記録されます。</p> <p>同期メソッドの代わりに同期オブジェクトを使用するコード変更で、この問題は解決しました。</p>
CR108034	ユーザが接続を切断することで生成されていた不適切なエラー メッセージが、生成されないようになりました。
CR108350	Administration Console では、ログ ファイル フォーマットを共通ログ フォーマット (CLF) と拡張ログ フォーマット (ELF) の間で動的に変更できるという情報が誤って示されていました。実際には、そのような変更にはサーバの再起動が必要です。このコンフィグレーション パラメータの変更にはサーバの再起動が必須であることを適切に示すように、コードが変更されました。

変更要求番号	説明
CR108385	<p>WebLogic Server クラスタの前でプロキシとして機能している <code>HttpClusterServlet</code> を使用してクラスタ インスタンスにチャンクデータをポストしようとしたときに、<code>NumberFormatException</code> が生成されていました。</p> <p><code>HttpClusterServlet</code> のプロキシを経由しないダイレクトのリクエストは、エラーがでることなく成功していました。</p> <p>チャンク転送エンコーディングされたデータは既にコア サーバでデコードされているということを考慮し、<code>HttpClusterServlet</code> が不必要にデータを再デコードすることがないようにコードを変更して、この問題は修正されました。</p>
CR108577	<p>アンデプロイメントで、モジュールの順序が考慮されていませんでした。このロジックは修正済みです。現在、モジュールはそれらがデプロイされた正確に逆の順序で非アクティブ化およびロールバックされます。</p>
CR108607	<p>WebLogic Server では、アプリケーションが FOP (Formatting Objects Processor) 出力メソッドのある XSL を使用してアーカイブ ファイルから画像を取得するとエラーが生成されていました。この問題は、展開形式でデプロイされたアプリケーションでは起こりませんでした。この問題は、コードを修正して解決されました。</p>
CR109479、 CR110838	<p><code>JnlpDownloadServlet</code> は、WAR ファイルのタイムスタンプを利用して、クライアントのリソースが Web サーバ上のものより古いかどうかを判断します。タイムスタンプを取得するために、<code>JnlpDownloadServlet</code> は <code>URLConnection.getLastModified()</code> メソッドを呼び出します。しかし、<code>getLastModified()</code> メソッドは 0 を返していました。</p> <p><code>ZipURLConnection.getLastModified()</code> が zip ファイル (したがって WAR ファイル) の変更時刻を返すように、コードが修正されました。</p>
CR109885	<p>障害の発生したセカンダリ ノードへのセッション レプリケーションで、プライマリ ノードがハングしていました。プライマリ ノードは、セカンダリ ノードへの接続を試行し、他のスレッドが待ち続けなければならないロックを保持しているようになっていました。</p> <p>コードを変更してこの問題は解決しました。</p>
CR109958	<p>プロキシ サーブレットを使用してリクエストを処理している場合に、WebLogic Server は HTTPS を使用する受信リクエストでのみ <code>SecureProxy</code> 設定に従っていました。受信リクエストが HTTP を使用している場合は、プロキシ サーブレットで <code>SecureProxy</code> が有効になっていても WebLogic Server は HTTPS 接続を使用していませんでした。この問題は、コードを修正して解決されました。</p>

変更要求番号	説明
CR110798	JSP で <code>weblogic.servlet.security.ServletAuthentication.killCookie(req)</code> を呼び出すと、クリーンアップされることなくセッションが WebLogic Server に留まっていた。 <code>killCookie(req)</code> が完了する直前にセッションが無効化されるように、コードが修正されました。
CR110832	<code>config.xml</code> の <code>WebAppComponent</code> で <code>DebugEnabled</code> が <code>true</code> に設定されている場合に、WebLogic Server は Web アプリケーションのデプロイメントで <code>NullPointerException</code> を送出していました。この問題は、コードを修正して解決されました。
CR110910	<code>KeepAliveEnabled</code> が <code>true</code> に設定されている場合、大きなファイルのダウンロードがキャンセルされた後に <code>HttpClusterServlet</code> が <code>NumberFormatException</code> を送出していました。 分析の結果、クライアントがファイルのダウンロードをキャンセルしたときに、残りのデータが入力ストリームに残されていたことが判明しました。ソケットが以降のリクエストで再利用される場合に、サーブレットが残りのデータを読み込んでしまい、その結果として例外が送出されていました。 この問題は、ダウンロードがキャンセルされた場合に入力ストリームから残留データが排出されるようコードを修正して解決しました。
CR110914	<code>TrackingEnabled</code> が <code>false</code> に設定されている場合、リクエストごとに新しいセッションが作成されていましたが、それらのセッションは無効化されていませんでした。 <code>TrackingEnabled</code> が <code>false</code> に設定されている場合にセッションが直ちに無効化されるように、コードが修正されました。これは正常な動作です。
CR111090、 CR090886	<code>-Dweblogic.webservice.transport.http.full-url</code> が <code>True</code> に設定されている場合、Web サービスは HTTPS プロトコルを使用して呼び出すことができませんでした。 コードを変更してこの問題は解決しました。
CR111531	[Sending UTF-8 Only] オプションが無効になっている場合、Weblogic Server 7.0 は 2 バイト文字の含まれているファイル名を処理しませんでした。名前に 2 バイト文字のあるファイルにアクセスしようとすると、404 (File not found) エラーが生じていました。 この問題は、コードを修正して解決されました。

---

変更要求番号	説明
CR111752	<p>特定の条件下では、<code>useByteStream</code> が <code>true</code> に設定されている <code>CGIServlet</code> を使用した場合に、WebLogic Server が <code>NullPointerException</code> を送出していました。この問題は、1つのフレームに静的な URL リンクがあり、もう1つのフレームで <code>CGIServlet</code> が使用されているフレームセットを使用している場合に発生していました。他方のフレームでページが完全にダウンロードされる前に静的リンク付きのフレームをユーザが選択すると、IO 例外が捕捉されて次のようにユーザに示されていました。</p> <pre data-bbox="307 494 1106 574">java.lang.NullPointerException at weblogic.utils.Executable\$Drainer.run(Executable.java:366)</pre> <p>この問題は、コードを修正して解決されました。</p>
CR112799	<p>WebLogic Server は、<code>IOException</code> が発生した後も出力ストリームに書き込もうとしていました。<code>IOException</code> を処理しない Web アプリケーションで予期しないソケットの切断が起こった場合に、このことで CPU の使用率が 100% になっていました。</p> <p><code>Org.apache.xml.serialize.XMLSerializer</code> は、プロセスが終了するまで <code>IOException</code> を無視します。これが原因で、HTTP 応答の一部として HTML ドキュメントを返す途中で <code>IOException</code> が発生したときに問題が生じていました。</p> <p><code>IOException</code> が発生した後は出力ストリームへの書き込みが停止されるように、コードが修正されました。</p>

---

変更要求番号	説明
CR112910、 CR092625	<p>HTTP のロギングを無効にして起動され、かつログ ファイルが存在していない管理対象サーバで HTTP のロギングを有効にした場合に、WebLogic Server が <code>NullPointerException</code> を送出していました。その管理対象サーバへの各 HTTP アクセスで、次の例外が送出されていました。</p> <pre data-bbox="370 401 1231 791">java.lang.NullPointerException at weblogic.servlet.logging.LogManagerHttp.log(LogManagerHttp.java:292) at weblogic.servlet.internal.HttpServer.log(HttpServer.java:865) at weblogic.servlet.internal.ServletResponseImpl.send(ServletResponseImpl.java:1044) at weblogic.servlet.internal.ServletRequestImpl.execute(ServletRequestImpl.java:2265) at weblogic.kernel.ExecuteThread.execute(ExecuteThread.java:139) at weblogic.kernel.ExecuteThread.run(ExecuteThread.java:120)</pre> <p>この問題は、コードを修正して解決されました。</p>
CR114936	<p>クライアントが進行中のリクエストをキャンセルした場合でも、WebLogic Server は入力ストリームを排出していました。このために、CPU リソースが消費されていました。</p> <p>入力ストリームが排出されないよう接続が破棄されるように、コードが修正されました。この修正の結果として、クライアントへの書き込み中に <code>IOException</code> が発生した場合、<code>GenericProxyServer</code> はこの接続を再利用せず、次のリクエストでは新しい接続を作成します。</p>
CR116209	<p>顧客がシリアライズ可能な <code>ServletContext</code> 属性を追加し、それをシリアライズできない値で上書きした場合は、元の値によって新しい値がマスクされていました。この問題は、属性のハッシュテーブルから古い値が削除されないことが理由で発生していました。</p> <p>シリアライズ可能な値をシリアライズできない値で置換する場合に、必要であれば、属性のハッシュテーブルから値が削除されるようにコードが修正されました。</p>

変更要求番号	説明
CR120228、 CR124369	展開された EJB および Web アプリケーションとそれらの MANIFEST ファイル (クラスパスに共有クラス) が格納された展開されたエンタープライズ アプリケーションは、WebLogic Server 7.0 SP2 では機能しましたが、WebLogic Server 7.0 SP3 では機能していませんでした。共有クラスが見つかりませんでした。この問題は、MANIFEST ファイルのクラスパス エントリを展開されたアプリケーションのクロスローダに追加するコード変更で解決しました。
CR120281	HttpServletRequest.getParameterValues (String) からパラメータ値が 2 度返されることがありました。分析の結果、転送されたリクエストのクエリ パラメータが、適切に解析されたパラメータ値が 2 回返される方法で解析されていたことが判明しました。 コードを変更してこの問題は解決しました。
CR120440	複数の Web アプリケーションがシングル サインオン コンフィグレーションにデプロイされ、1 つのアプリケーションが <code>weblogic.servlet.security.ServletAuthentication.invalidateAll (request)</code> を呼び出したときに、その他のアプリケーションの <code>HttpSessionListeners</code> がセッションのタイムアウトが起こるまで呼び出されていませんでした。最初の Web アプリケーションと関連するセッションだけが無効化用に登録されていたことが、この問題の原因でした。ユーザが認証された後に、それ以降のセッションは登録されていませんでした。 すべての Web アプリケーションのセッション ID とコンテキストパスの両方が、 <code>invalidateAll (request)</code> で必要に応じて無効化できるよう登録されるように、コードが修正されました。
CR121094、 CR129364	クラスタとそのクラスタにもマップされた VirtualHost の両方にデプロイされた Web アプリケーションを再デプロイすると、 <code>InstanceNotFoundException</code> が生じていました。この問題は、コードを変更して解決されました。

変更要求番号	説明
CR121175	<p>HTTPClusterServlet が、セッション クッキーの JVMID で識別されたプライマリ インスタンスとは異なるサーバ インスタンスにリクエストを転送していました。HTTPClusterServlet は、クラスタ化されていない複数の管理対象サーバにリクエストをプロキシしていました。バックエンド サーバ インスタンスの Web アプリケーションは、index.jsp と frameset.jsp で構成されていました。Web アプリケーションの起点は index.jsp であり、そこでセッションが作成されます。index.jsp は &lt;jsp:forward&gt; を使用してリクエストを frameset.jsp ( フレームセット ) に転送します。</p> <p>プロキシ (HTTPClusterServlet) を通じて最初に index.jsp にアクセスすると、セッションが作成され、クッキーが設定されます。index.jsp が frameset.jsp に転送するときには、( フレームセットの JSP ごとの ) 新しいリクエストがクッキーと一緒に送信されます。クッキーがリクエストにあることもありますが、リクエストはサーバリストのサーバ ( プライマリ サーバ以外 ) に送信され、セッションは失われます。プロキシ ログには、次のエントリが記録されます。</p> <pre data-bbox="370 748 1190 881">&lt;Thu Aug 21 15:41:15 PDT 2003&gt;: Found cookie: -1339390245 &lt;Thu Aug 21 15:41:15 PDT 2003&gt;: In-bound headers: &lt;Thu Aug 21 15:41:15 PDT 2003&gt;: Content-Length: 117 &lt;Thu Aug 21 15:41:15 PDT 2003&gt;: ##### Trying to connect with server -1189081773!172.17.26.74!7201!443</pre> <p>この問題は、サーバリストを更新するロジックを変更することで解決しました。リスト中のサーバの JVMID を更新すると、そのサーバはリストから削除され、更新されて、それからリストに戻されるようになりました。単純にオブジェクトを更新しても、リストはソートされませんでした。</p>
CR121343、 CR133921、 CR129538、 CR129537、 CR174336	<p>複数のフレームが使用されている場合に、セカンダリ JVMID の算出時に競合状態が生じていました。あるスレッドで、セカンダリ JVMID の算出によりメンバー変数値がリセットされ、それによって競合状態が生じていたようでした。コード変更の後、セカンダリ JVMID の算出で競合状態が生じなくなりました。</p>

変更要求番号	説明
CR121359	<p>JSTL の終了タグのタグ名と右山括弧 (&gt;) の間にスペースがある場合に (例 : &lt;/c:if &gt;)、weblogic.utils.ParsingException が発生していました。</p> <pre data-bbox="307 335 1166 1012"> &lt;%@ taglib uri="/WEB-INF/c.tld" prefix="c" %&gt; Full stack trace: &lt;Aug 25, 2003 12:46:23 PM EDT&gt; &lt;Error&gt; &lt;HTTP&gt; &lt;101017&gt; &lt; [ServletContext (id=53171 81,name=TempApp,context-path=/TempApp)] Root cause of ServletException weblogic.utils.ParsingException: Could not complete parsing, unmatched tags: if otherwise when choose if forEach if at weblogic.servlet.jsp.JspLexer.parse (JspLexer.java:976) at weblogic.servlet.jsp.JspParser.doit (JspParser.java:90) at weblogic.servlet.jsp.JspParser.parse (JspParser.java:213) at weblogic.servlet.jsp.Jsp2Java.outputs (Jsp2Java.java:119) at weblogic.utils.compiler.CodeGenerator.generate (CodeGenerator.j ava:258 ) at weblogic.servlet.jsp.JspStub.compilePage (JspStub.java:353) at weblogic.servlet.jsp.JspStub.prepareServlet (JspStub.java:211) at weblogic.servlet.jsp.JspStub.prepareServlet (JspStub.java:164) at weblogic.servlet.internal.ServletStubImpl.getServlet (ServletSt ubImpl.java:517) at weblogic.servlet.internal.ServletStubImpl.invokeServlet (Servlet StubImpl.java:351) at weblogic.servlet.internal.ServletStubImpl.invokeServlet (Servlet StubImpl.java:306) . </pre> <p>タグ名と「&gt;」の間でスペースを 1 回まで使えるようにコードを変更することで、この問題は解決しました。</p>
CR121846	<p>WebLogic Server 7.0 の以前のサービス パックでは、サーバが、拡張ログ フォーマットのヘッダを書き込む前に標準ログ エントリをログ ファイルに書き込むことができました。このような状況は、ログ ローテーションで、複数のスレッドが新しいログ ファイルに同時に書き込もうとしたときに起こる可能性があります。</p> <p>ログ ヘッダが書き込まれる後まで、ログ ローテーションを処理するスレッドが新しいログ ファイルに排他的にアクセスするように、コードが修正されました。</p>
CR122177	<p>ファイルが見つからないときに、FileServlet が 404 ではなく 200 の応答コードを返していました。ファイルが見つからないときは 404 が返されるように、コードが修正されました。</p>



変更要求番号	説明
CR122551、 CR122556、 CR129248、 CR134888	<p>コンテンツ長が -1 で、書き込まれたバイトが 0 の場合に、トンネリングされたリクエストについて ServletOutputStreamImpl が次の例外を送出していました。</p> <p>Servlet failed with IOException.java.net.ProtocolException: Didn't meet stated Content-Length, wrote: '0' bytes instead of stated: '-1' bytes.</p> <p>コードを変更してこの問題は解決しました。</p>
CR124524、 CR127621	<p>weblogic.management.configuration.WebServerMBean の値 maxKeepAliveSecs が、最大 300 秒までコンフィグレーションできるようになりました。</p>
CR127888、 CR111024	<p>キャッシュ フィルタの使用時に、負荷のかかっている状態でサーバがハングしていました。分析の結果、ロックされている範囲と比べてロックされているキーのバランスが取れていないことが判明しました。</p> <p>コードを変更してこの問題を解決しました。</p>
CR134414	<p>以前のサービス パックでは、シリアライズ可能なサブレット リクエスト属性を追加して、それをシリアライズできない値で上書きすると、元の値によって新しい値がマスクされていました。</p> <p>コードを変更してこの問題は解決しました。</p>

## SNMP (Simple Network Management Protocol)

変更要求番号	説明
CR113122	<p>sysUpTime で WebLogic Server SNMP エージェントから返された値が、SNMP が初期化されてから経過した時間を正確に示していませんでした。</p> <p>コードを変更してこの問題は解決しました。</p>

## Web アプリケーション

変更要求番号	説明	問題が修正されたリリース
CR175651	<p data-bbox="310 388 407 409">対象事項</p> <p data-bbox="310 427 911 644">コンテナが、Web アプリケーションのアンデプロイメント中または再デプロイメント中に、処理中の Web アプリケーション リクエストの完了を待機しませんでした。その結果 Web アプリケーションがコンテキストの使用に依存している場合に、コンテキストが有効でなくなっていることに起因して、アンデプロイ後に <code>NullPointerException</code> が発生していました。</p> <p data-bbox="310 661 407 682">解決方法</p> <p data-bbox="310 699 911 921">解決策として、起動スクリプト ファイルに <code>-D</code> オプション (すなわち <code>weblogic.http.requestCompletionTimeoutSecs</code>) が導入されました。このフラグに指定される値は、コンテナが処理中の作業の完了を待機する秒数を示します。デフォルトの値は 0 秒です。つまり、このフラグを指定しない場合、コンテナは完了を待機しません。</p>	7.0 SP4

# WebLogic Tuxedo Connector

変更要求番号	説明
CR125533	<p>CLASSPATH に EJB JAR ファイルが含まれていない場合、セッション Bean のサービス メソッドの呼び出しで、TypedFML._tmpresend の呼び出しにつながるオブジェクト レプリケーション ロジックが開始されま す。CARRAY フィールドが null の場合、_tmpresend はフィールド長を (FLDID_SIZE + FLDLEN_SIZE ではなく) 0 として記録します。最終的に は、_tmppostrecv が呼び出され、フィールド長は FLDID_SIZE + FLDLEN_SIZE だと見なされます。_tmpresend はこの情報を記録しな かったため、負の値がフィールド長として読み込まれていました。現在、 フィールド長で行われるチェックは、それが 0 であるかどうかをチェッ クします。これが、NegativeArraySize 例外の理由です。</p> <p>フィールド長をチェックし、それが 0 以下であるかどうかを判別するた めのコードが _tmppostrecv で追加されました。</p>
CR074356	<p>リダイレクトに Name 属性がなかったため、複数の tBridge リダイレクト をお互いに区別することができませんでした。</p> <p>tBridge リダイレクトに、コンフィグレーション可能な Name 属性が追加 されました。</p>

変更要求番号	説明
CR110812	<p>Tuxedo Connector キュー ブリッジは、WebLogic Tuxedo デバッグ オプションが設定されていない場合でも WebLogic Server のログにデバッグメッセージを書き込んでいました。Administration Console の [Tuxedo Queuing Bridge Connections] タブ ページのタイムアウト値で指定された間隔で、デバッグメッセージはログに記録されていました。たとえば、タイムアウト値が 60 秒に設定されている場合は、次のようにデバッグメッセージが記録されていました。</p> <pre> #####&lt;25-Jun-03 12:49:30 BST&gt; &lt;Debug&gt; &lt;WTC&gt; &lt;jdunn01&gt; &lt;adminserver&gt; &lt;Thread-11&gt; &lt;kernel identity&gt; &lt;59:223361a20602d183&gt; &lt;180046&gt; &lt;/tBexec/tuxQ2jmsQ/TPException TPETIME(13):0:0:TPED_MINVAL(0):QMNONE(0):0&gt; #####&lt;25-Jun-03 12:50:30 BST&gt; &lt;Debug&gt; &lt;WTC&gt; &lt;jdunn01&gt; &lt;adminserver&gt; &lt;Thread-11&gt; &lt;kernel identity&gt; &lt;60:223361a20602d183&gt; &lt;180046&gt; &lt;/tBexec/tuxQ2jmsQ/TPException TPETIME(13):0:0:TPED_MINVAL(0):QMNONE(0):0&gt; #####&lt;25-Jun-03 12:51:31 BST&gt; &lt;Debug&gt; &lt;WTC&gt; &lt;jdunn01&gt; &lt;adminserver&gt; &lt;Thread-11&gt; &lt;kernel identity&gt; &lt;61:223361a20602d183&gt; &lt;180046&gt; &lt;/tBexec/tuxQ2jmsQ/TPException TPETIME(13):0:0:TPED_MINVAL(0):QMNONE(0):0&gt; ..... </pre> <p>この問題は、コードを変更して解決されました。</p>

変更要求番号	説明
CR121355	<p data-bbox="448 256 1237 352">応答メッセージの <code>TypedCArray</code> バッファが、不適切に 4 バイト アラインでパディングされていました。 <code>TypedCArray</code> オブジェクトでは、パディングを行うべきではありません。</p> <p data-bbox="448 366 1237 461">CARRAY メッセージ バッファを使用して <code>wtc</code> サービスに対して <code>tpcalls</code> を行う Tuxedo クライアントのテストでは、次のような結果が出ていました。</p> <ul data-bbox="448 475 1210 795" style="list-style-type: none"><li data-bbox="448 475 1210 539">• <code>wtc</code> サービスが 174 バイトを送り返す場合、<code>tuxedo</code> クライアントは 176 バイトを受信する</li><li data-bbox="448 560 1210 624">• <code>wtc</code> サービスが 175 バイトを送り返す場合、<code>tuxedo</code> クライアントは 176 バイトを受信する</li><li data-bbox="448 645 1210 710">• <code>wtc</code> サービスが 176 バイトを送り返す場合、<code>tuxedo</code> クライアントは 176 バイトを受信する</li><li data-bbox="448 730 1210 795">• <code>wtc</code> サービスが 177 バイトを送り返す場合、<code>tuxedo</code> クライアントは 180 バイトを受信する</li></ul> <p data-bbox="448 808 1237 873">この問題は、サービスが別の <code>tuxedo</code> ドメイン ゲートウェイで動作している場合には起こっていませんでした。</p> <p data-bbox="448 881 962 907">この問題は、コードを修正して解決されました。</p>
CR121432	<p data-bbox="448 933 1237 1064"> <code>weblogic.wtc.tbridge.tBexec</code> クラスで、 <code>tpenqueue</code> を実行しているときに、コードが <code>null</code> の <code>replyQ</code> をチェックし、見つかった場合はコンフィグレーションされているリクエスト キューの名前で置換していました。</p> <p data-bbox="448 1072 1237 1133">このチェック機能は廃止され、 <code>NULL</code> は <code>replyQ</code> でコンフィグレーション可能な値になっています。</p>
CR122953	<p data-bbox="448 1159 1237 1255"> <code>weblogic.wtc.jatmi.TypedFML32</code> クラスの <code>Fchg(FmlKey key, Object value)</code> メソッドが、前のすべてのエントリを規定どおりに <code>null</code> 値で適切に設定していませんでした。</p> <p data-bbox="448 1263 962 1289">この問題は、コードを修正して解決されました。</p>

## Web サービス

変更要求番号	説明
CR122845	<p>SOAP Attachments API for Java (SAAJ) が実装されていなかったため、バージョン 7.0 では SOAPFaultException 例外の <code>java.xml.soap.Detail</code> オブジェクトを作成することができませんでした。</p> <p>この問題は、WebLogic API <code>weblogic.webservice.util.FaultUtil</code> をエクスポートするコード変更で解決されました。このコード変更には、Detail オブジェクトを作成するための <code>newDetail()</code> メソッドが含まれています。</p> <p>注意: <code>weblogic.webservice.util.FaultUtil</code> API は、WebLogic Server のバージョン 8.1 以降では非推奨になります。それらのバージョンでは、SAAJ API を使用して Detail オブジェクトを作成します。</p>
CR120594	<p><code>clientgen</code> の <code>typeMappingFile</code> で指定されたカスタム型マッピングファイルは無視されていました。</p> <p>この問題は、カスタム型マッピングファイルを設定し、デフォルトのマッピングファイルをオーバーライドすることで解決しました。</p>
CR103985	<p><code>startWebLogic.cmd</code> スクリプトの「Setting <code>javax.xml.soap.MessageFactory</code>」でのプロパティの設定が、JSP およびサーブレットの <code>org.xml.sax.driver</code>、<code>org.xml.sax.parser</code>、<code>javax.xml.soap.MessageFactory</code>、および <code>javax.xml.rpc.ServiceFactory</code> で機能していませんでした。</p> <p>コード変更の後、WebLogic Server は実際に設定する前にプロパティが既に設定されているかどうかを確認し、設定されていない場合はデフォルト値に設定します。</p>
CR129061	<p>Apache SOAP クライアントはパラメータを取るメソッドでは WebLogic Server Web サービスに適切に接続しますが、パラメータなしのメソッドの場合はサーバ側で次のエラーが生成されていました。</p> <pre>javax.xml.soap.SOAPEXception: Found SOAPElement ['']. But was not able to find a Part that is registered with this Message which corresponds to this SOAPElement</pre> <p>コードを変更してこの問題を解決しました。</p>
CR126896	<p><a href="http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA04_47.0.0.jsp">http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA04_47.0.0.jsp</a> のセキュリティ勧告に関する情報を確認してください。</p>

変更要求番号	説明
CR089677	クラス <code>webllogic.xml.schema.binding.FaceUtils</code> は <code>webservices.jar</code> にありましたが、 <code>webserviceclient.jar</code> にはありませんでした。クライアントが複合パラメータ (Java オブジェクトのリストなど) を Web サーバに渡すときにこのクラスが存在しないことが明らかになると、その不在が原因でエラーが発生していました。
CR090950、CR172603	クライアント側のハンドラで <code>SOAPElement.getElementName()</code> にアクセスすると、 <code>ClassCastException</code> が生じていました。 <code>SOAPElement</code> が、 <code>javax.xml.soap.Name</code> を実装していない名前クラスで作成されていたからです。この名前クラスは、 <code>javax.xml.soap.Name</code> に変換されるようになりました。
CR091230	<code>clientgen Ant</code> タスクは、WSDL から Java コードを生成するときにアンダースコア ( <code>_</code> ) を削除していました。コードを変更してこの問題を解決しました。
CR097344	<code>autotype</code> コード ジェネレータが、型のない属性または型が匿名の属性を適切に処理していませんでした。コードの修正で、この問題は解決されました。
CR099339	<code>-Dwebllogic.webservice.verbose=true</code> でサーバを起動すると、詳細出力でリクエストと応答 XML ではなく応答 XML が 2 回表示されます。最初の出力のタイトルは、それが応答であっても「Request」となります。 コード変更の後は、 <code>Dwebllogic.webservice.verbose</code> が有効になっている場合に、SOAP リクエストと応答が適切にサーバ側の <code>stdout</code> に出力されるようになりました。
CR105715	WebLogic Server の <code>clientgen Ant</code> タスクが、WSDL ファイルにハイフンが残ることを不適切に許していました。これが原因で、生成された Java コードは Java では許されていないハイフンのあるクラス名やメソッド名を持っていました。 <code>NameUtils</code> の修理により、 <code>clientgen</code> がハイフンを削除するようになりました。また、JAXRPC のメソッドとクラスについては、生成された文字列が Java のキーワードでもある場合、WebLogic Server は JAXRPC 仕様 1.0 および 1.1 に従って <code>_</code> を先頭に付けます。

---

変更要求番号	説明
CR106741	<p>clientgen の複数のサービスで useServerTypes を指定しても、正しく機能していませんでした。複数のサービス エントリを持つ servicegen を実行して、useServerTypes=true を指定したクライアントを生成する場合に、型ファイルがクライアント用にエンタープライズ アプリケーションから必ずしもコピーされず、型コードの生成でゲッターとセッターのメソッド名から「_」が削除されていました。</p> <p>現在は、クライアントが useServerTypes=true を設定する servicegen タスクで複数のサービスを指定できます。たとえば、次のように指定しても機能するようになりました。</p> <pre>&lt;target name="ear"&gt;   &lt;servicegen     destEar="RetailServicesServerBEA"     warName="RetailServicesServerBEAWebApp"&gt;     &lt;service       javaClassComponents="com.hp.wsmo.demo.retail.webservice. products.ProductsWS"targetNamespace="http://localhost:70 01/javaaclass";         serviceName="Products"         serviceURI="/Products"         expandMethods="true"&gt;         &lt;client           packageName="com.hp.wsmo.demo.retail.client.bea.products"           clientJarName="PRODUCTS.jar"             useServerTypes="true" /&gt;         &lt;/service&gt;       &lt;/service&gt;       javaClassComponents="com.hp.wsmo.demo.retail.webservice. orders.OrdersWS"targetNamespace="http://localhost:7001/ javaaclass";         serviceName="Orders"         serviceURI="/Orders"         expandMethods="true"         style="rpc"&gt; &lt;client           packageName="com.hp.wsmo.demo.retail.client.bea.orders"             clientJarName="ORDERS.jar"             useServerTypes="true" /&gt;         &lt;/service&gt;</pre>

---



変更要求番号	説明
CR107542、CR136469	<p>7.0の以前のサービス パックでは、リクエストをソケットに書き込むときに URL パスのみが Request-Line で使用され、Web サービスの URI の HTTP クエリと参照パラメータは Web サービスに渡されていませんでした。</p> <p>クエリと参照パラメータは、それらがある場合はリクエストの URI に追加されるようになりました。</p>
CR111879	<p>Date を引数として使用する Web サービスをテストするときに、Web サービスで、ユーザ名とパスワードを要求するセキュリティ ダイアログが表示されていました。この問題は、引数が文字列に変更されたときには起こりませんでした。</p> <p>セキュリティ ダイアログを排除し、Date 引数を使用するときに適切にサービスを呼び出すコード変更で、この問題は解決しました。</p>
CR112443、 CR109898、CR126134	<p>メソッドにパラメータとして <code>org.w3c.dom.Document []</code> を渡すと、次の行で始まる <code>SOAPException</code> が送出されていました。</p> <pre data-bbox="448 788 1224 984"> javax.xml.soap.SOAPException: Found SOAPElement [&lt;anyType&gt; &lt;this xmlns="mynamespace"&gt; &lt;f:that xmlns:f="yournamespace"&gt; &lt;or&gt; a lot of random &amp;lt; &lt;/or&gt; &lt;f:the&gt; &lt;/f:the&gt; &lt;f:other&gt; foo bar blaz&lt;/f:other&gt; &lt;/f:that&gt; &lt;/this&gt; &lt;/anyType&gt;]. But was not able to find a Part that is registered with this Message which corresponds to this SOAPElement ... </pre> <p>コードを修正してこの問題を解決しました。</p>
CR112940	<p>サービス パック 5 の WebLogic Server では、以前は欠けていた <code>SOAPHeader</code> クラスの以下のメソッドが実装されました。</p> <ul data-bbox="448 1135 908 1367" style="list-style-type: none"> <li>• <code>examineHeaderElements</code></li> <li>• <code>extractHeaderElements</code></li> <li>• <code>examineMustUnderstandHeaderElements</code></li> <li>• <code>examineAllHeaderElements</code></li> <li>• <code>extractAllHeaderElements</code></li> </ul>

変更要求番号	説明
CR120076	<p>リクエストの偽の認可ヘッダが原因で、Web アプリケーションでセキュリティ制約が定義されていない場合でも、Web サービス エンジンが基本認証を実行しようとしていました。</p> <p>コードを変更してこの問題は解決しました。</p>
CR120548	<p>WebLogic Server 7.0 の SP02 と SP03 では、ユーザが次のクライアント呼び出しを使用して <code>webservices.basic.statelessSession</code> サンプル Web サービスにアクセスした場合に、</p> <pre data-bbox="377 517 1173 574">String returnVal = port.sayHello(4, "\n\n &lt;--spaces and \n\n");</pre> <p>先頭のスペース (改行とスペース) が削除され、サーバ側では現れていませんでした。</p> <p>この問題は、コードを修正して解決されました。</p>
CR120741	<p>クライアントが静的であるか、動的であるか、あるいは Web サービスのテスト ホーム ページであるかに関係なく、Web サービスのオペレーションに <code>char</code> を渡すときに問題が発生していました。パラメータとして <code>char</code> を渡すと、次のアサーション エラーが生じていました。</p> <pre data-bbox="377 852 1173 1420">run: [java] weblogic.utils.AssertionError: ***** ASSERTION FAILED *****[ invalid class: class java.lang.String ] [java] at weblogic.utils.Debug.assertion(Debug.java:84) [java] at weblogic.xml.schema.binding.internal.builtin.JavaCharSerializer.getContentFromObject(JavaCharSerializer.java:23) [java] at weblogic.xml.schema.binding.internal.builtin.XSDSimpleTypeSerializer.writeContentToStream(XSDSimpleTypeSerializer.java:137) [java] at weblogic.xml.schema.binding.internal.builtin.XSDSimpleTypeSerializer.writeContents(XSDSimpleTypeSerializer.java:130) [java] at weblogic.xml.schema.binding.CodecBase.serializeInitialWrite(CodecBase.java:370)...</pre> <p>この問題は、コードを修正して解決されました。</p>

変更要求番号	説明
CR121394、CR128988	<p>ISAPI フィルタを使用して Web サービスでメソッドを呼び出すと、次の例外が生じていました。</p> <pre>java.lang.IllegalArgumentException: Illegal MimeHeader name or value</pre> <p>コードを変更してこの問題は解決しました。</p>
CR122716	<p>基本型のネームスペースが異なる場合に、Clientgen では拡張型の生成されたクラスをコンパイルすることができませんでした。</p> <p>分析の結果、異なるパッケージの基本クラスがインポートされないことが判明しました。完全なパッケージ名を使用して基本クラスの名前が生成されるようになって、この問題は解決しました。</p>

## WebLogic Server 7.0 サービス パック 4 のソリューション

以下の節では、WebLogic Server 7.0 サービス パック 4 (SP4) で解決された問題を説明します。ここで示す解決済みの問題のリストは、定期的に更新されます。

- [4-176 ページの「Administration Console」](#)
- [4-177 ページの「クラスタ」](#)
- [4-177 ページの「コネクタ」](#)
- [4-178 ページの「EJB」](#)
- [4-178 ページの「EJB」](#)
- [4-179 ページの「JDBC」](#)
- [4-180 ページの「JMS」](#)
- [4-181 ページの「JSP とサーブレット」](#)
- [4-181 ページの「JTA」](#)
- [4-182 ページの「ノード マネージャ」](#)

- [4-182 ページの「セキュリティ」](#)
- [4-183 ページの「ツール」](#)
- [4-184 ページの「Web サービス」](#)

## Administration Console

変更要求番号	説明
CR091853	<p>3 ノードのクラスタが複数の独立した Solaris 2.8 マシン上にある場合は、デフォルトの実行キューにアクセスするときに管理サーバから InstanceNotFound 例外が送出されていました。</p> <p>この問題は解決済みです。</p>
CR104066	<p>アプリケーション ポーラーが原因で CPU の使用率が高くなるがありました。</p> <p>現在、<code>adminMBeanHome.getMBeansByType("Application")</code> は各ポーリング間隔で 1 回のみ実行されるようになっています。</p>
CR105456	<p>互換性セキュリティの使用時に Administration Console で [セキュリティ   レalm   CompatibilityRealm   プロバイダ   裁決   RealmAdapterAdjudicator] ノードの [詳細] タブを選択すると、<code>javax.management.AttributeNotFoundException</code> が生じます。</p> <p>[詳細] タブは、属性 <code>RequireUnanimousPermit</code> を探していました。この属性は、<code>RealmAdapterAdjudicator</code> MBean では指定されなくなっています。</p> <p>この問題は解決済みです。</p>
CR106122	<p>WebLogic Server 7.0 では、以前は MLET 要素の NAME 属性を指定する必要がありました。</p> <p>JMX 仕様に準拠するために、NAME 属性は必ずしも指定する必要がなくなりました。</p>

変更要求番号	説明
CR110224	<p>クラスタ環境では、<code>getMBeansByType</code> を使用して MBean にアクセスするために初期コンテキストが管理対象サーバから取得された場合に、管理サーバを再起動すると次の行で始まる例外が送出されていました。</p> <pre>java.rmi.ConnectException: This RJVM has already been shutdown</pre> <p>管理対象サーバの再接続で JNDI をリバインドすることで問題は解決されました。</p>
CR110557	<p>Monitors グループのみに属するユーザは、デプロイメント、サーバの状態、およびクラスタをモニタすることができませんでした。この問題は解決済みです。</p>

## クラスタ

変更要求番号	説明
CR108127	<p>プライマリ サーバ A からセカンダリ サーバ B に切り替えると、サーバ B が A をセカンダリとして再登録することなくプライマリ サーバとして再登録されていました。サーバ A からサーバ B に切り替えて、再びサーバ A に戻すと、セッションが古くなっていました。</p> <p>サーバ A のプライマリ ステータスは、サーバ B がプライマリ サーバになったときに削除されるようになりました。</p>

## コネクタ

変更要求番号	説明
CR106960	<p>RAR が EAR に含まれている場合、その EAR の再デプロイメントは失敗していました。コネクタ モジュールのコードが再デプロイメントを適切に処理しなかったためです。</p> <p>コネクタ モジュールの再デプロイ ロジックを修正することで問題は解決しました。</p>

## EJB

変更要求番号	説明
CR106041	<p>ejbc で、BMP (Bean 管理の永続性) Bean のオブティミスティックな同時方式を許可しない準拠チェックが実行されるようになりました。</p> <p>オブティミスティックな同時方式は、EJB 2.0 の CMP (コンテナ管理の永続性) Bean の機能です。「WebLogic Server EJB コンテナとサポートされるサービス」の「<a href="#">EJB の同時方式</a>」を参照してください。</p>
CR103391	<p>Microsoft の JDBC ドライバには制限があり、そのため、行とカラムから成る特定の結果セットについて、getXXX メソッドを 1 行につき 1 度しか呼び出すことができません。この制限は、クエリ カラムの中にテキストカラムまたは画像カラムが含まれている場合にのみ適用されていました。WebLogic Server で生成された JDBC は (たとえば getLong(1) を使用して) 行からキー値を取得し、その結果セットを、getLong(1) を再実行して、最初のカラム値を含むすべてのカラム値を読み込む別のルーチンに渡していました。この再実行により、Microsoft のドライバは例外を送出していました。</p> <p>この問題は、resultSet の主キー カラムが 2 回解析されるのを防止するコード修正で解決されました。</p>

変更要求番号	説明
CR105966	<p>weblogic-cmp-rdbms-jar.xml の field-group エントリに CMR フィールドが含まれていると、次のエラーでコンパイルが失敗していました。</p> <pre>D:\AA\bea70sp1\weblogic700\samples\server\src\examples\ejb20\relationships\Fathr\EJB_Problem2&gt;java weblogic.ejbcfatherError.jar fathersonError.jar ejbcgen\temp_lr_sahre_jb8\SonBean_1saa_WebLogic_CMP_RDBMS.java:909: incompatible types found    : &lt;null&gt; required: int __WL_bean.__WL_isLoaded[null] = true; 1 error Exec failed .. exiting</pre> <p>この問題は解決済みです。</p>
CR106774	<p>WebLogic Server で生成された EJB SQL クエリでは、「&gt;= AND &lt;=」が演算子として使用されていました。IBM DB2 は「&lt;=」を処理せず、したがって同じ機能の「BETWEEN」を必要としません。</p> <p>DB2 では、生成された EJB SQL が「&gt;= AND &lt;=」の代わりに BETWEEN を使用するようになり、NOT BETWEEN も適切に処理するようになりました。</p>

## JDBC

変更要求番号	説明
CR109941	<p>weblogic-application.xml で指定された情報に基づいて localDataSource オブジェクトを作成すると、NullPointerException が送出されていました。</p> <p>メソッド weblogic.jdbc.common.internal.LocalDataSource.defineDriverProps は、PoolParamsMBean、ConnectionCheckParamsMBean、および XaParamsMBean について null 値をチェックしていませんでした。</p> <p>この問題は解決済みです。</p>

# JMS

---

変更要求番号	説明
CR108665	<p>JMS サーバは、キュー ブラウザの参照リストにまだそのメッセージがあるのに永続ストアから期限切れメッセージを削除し、その結果としてページング IO 例外が送出されていました。</p> <p>コードの変更によってページングの処理が改良され、問題は解決されました。</p>
CR109599	<p>Microsoft SQLServer 2000 を使用して JMS JDBC ストアを作成すると、メッセージがキューにある状態でサーバを再起動した場合に例外が生じていました。</p> <p>クエリとカラム取得の順序を変更することで問題は解決しました。</p>

---



## JSP とサーブレット

変更要求番号	説明
CR103304	jsp_precompile は、カスタム クラスの拡張版を含むすべての種類の JSP で機能するようになりました。
CR096041、 CR108111	<p>WebLogic Server で、ウェルカム ファイルを返すときにリダイレクト (HTTP ステータス コード 302) が使用されなくなりました。</p> <p>8.1 SP2 および 7.0 SP4 より前のバージョンの WebLogic Server では、302 (moved temporary) ステータス コードとウェルカム ファイルの場所を返していました。</p> <p>WebLogic Server バージョン 8.1 SP2、7.0 SP4、およびそれ以降では、200 (OK) ステータス コードが返されてウェルカム ファイルの内容が表示されます。</p> <p>この変更の結果、WebLogic Server のパフォーマンスは向上していますが、ウェルカム ファイルの相対 URL が別の場所を指している場合 404 Not Found エラーになることがあります。</p>
CR107419	<p>次のような、2 バイト文字をパラメータの値として設定する JSP コードでは、</p> <pre>&lt;jsp:include page="included.jsp"&gt;   &lt;jsp:param name="title"     value="&lt;%= java.net.URLEncoder.encode(\" [double byte characters]\" ) %&gt;"/&gt; &lt;/jsp:include&gt;</pre> <p>指定された 2 バイト文字が URL エンコーディングされたコードに変更されていました。この問題は解決済みです。</p>

## JTA

変更要求番号	説明
CR111719	XA 接続プールで必須の属性 KeepXAConnTillTxComplete は、XA アイソレーション レベルが変更されている場合のみチェックを行うようになりました。

## ノード マネージャ

変更要求番号	説明
CR105924	<p>ノード マネージャは、管理サーバが再起動した後に管理対象サーバを停止することができませんでした。</p> <p>ノード マネージャは、停止時に管理対象サーバのステータスを適切に更新するようになっています。</p>

## セキュリティ

変更要求番号	説明
CR105443	<p><a href="http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/SA_BEA03_36.00.jsp">http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/SA_BEA03_36.00.jsp</a> のセキュリティ 勧告を参照してください。</p>
CR110892	<p><a href="http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA03-35.jsp">http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA03-35.jsp</a> のセキュリティ 勧告を参照してください。</p>
CR112886	<p>ヘッダの ID アサーション命名規約 (WL-PROXY-CLIENT-*) が、WL-PROXY-CLIENT-KEYSIZE、WL-PROXY-SECRET-KEYSIZE、および WL-PROXY-CLIENT-IP と対立していました。</p> <p>それらのヘッダを ID アサーション ヘッダとしてフィルタ処理することで、Client-Cert ID アサーション エラーが防止されました。</p>

## ツール

---

変更要求番号	説明
CR112500	<p>weblogic.Admin コマンドは、指定されたユーザとパスワードが認証されないときに正確な終了コードを返しませんでした。たとえば、存在しないユーザの例を次に示します。</p> <pre>java weblogic.Admin -username george -password hamilton -url t3://localhost:7001 lock</pre> <p>この例ではエラー メッセージとユーザが認証されなかったことを示すスタックトレースが返されましたが、終了コードは成功を示していました。この問題は解決済みです。</p>

---

## Web サービス

変更要求番号	説明
CR104199	<p>.NET C# クライアントを使用して WebLogic Server Web サービスにアクセスし、要求した 2 つの文字列が返される場合に (最初の文字列は結果、2 番目の文字列は in/out パラメータ)、返された 2 つの文字列の順序が正しくありませんでした。2 番目の文字列が結果として返され、in/out パラメータは null として返されていました。コードの変更により、返される最初のアクセサが戻り値になります。</p>
CR092268	<p>ファイアウォールの内側で動作するクライアントは、Web サービス スタブでプロパティを設定する必要がありました。</p> <p>この問題は、2 つの新しいプロパティ <code>weblogic.webservice.client.proxyusername</code> と <code>weblogic.webservice.client.proxypassword</code> の導入によって解決されました。</p>
CR100098	<p>SSL で保護された Web サービスの動的クライアントは、ポート番号がエンドポイント URL に含まれていない場合に機能しませんでした。</p> <p>デフォルトの SSL リスン ポート 443 が現在は自動的に指定されます。</p>
CR103512 CR107171	<p>スタブで設定された一部のプロパティが、MessageContext にコピーされていませんでした。</p> <p>コードを変更した後、WebLogic Server は以下の使用事例をサポートするようになりました。</p> <p>エンドポイントは、次のフローで修正できます。</p> <ul style="list-style-type: none"><li>• Stub-&gt;MessageContext in handleRequest-&gt;MessageContext in handleResponse-&gt;StubCall-&gt;MessageContext in handleRequest-&gt;Call</li></ul> <p>タイムアウトは、次の順序で設定できます。</p> <ul style="list-style-type: none"><li>• Stub-&gt;MessageContext in handleRequest</li><li>• Call-&gt;MessageContext in handleRequest</li></ul>
CR108579	<p>配列またはユーザ定義の型を使用した、ドキュメントリテラル形式の Web サービスからの SOAP 応答に、不適切にネームスペースで修飾された要素が含まれていて、その結果として .Net が一部のデータを無視してました。</p> <p>この問題は、ネームスペース情報を SOAP 要素に渡すことで解決しました。</p>

変更要求番号	説明
CR109150	<p>weblogic.webservice.tools.versioning.VersionMaker ユーティリティで作成されたポータブル スタブを使用すると、NoClassDefFoundError 例外が生じていました。</p> <p>wsclient70.jar には、クライアントサイド メッセージ ロガーではなくサーバサイドメッセージ ロガーが含まれていました。wsclient70.jar を更新することで、この問題は解決しました。</p>
CR110168	<p>clientgen タスクで作成された、配列を返すサービスのクライアントは、サーバから返された XML が明示的に配列のサイズを指定していない場合には失敗していました。</p> <p>サイズの指定されていない SOAP 配列の場合、オブジェクトは配列が作成される前にリストに格納されるようになりました。</p>
CR110384	<p>wspackage に codecDir が欠けていることで、NullPointerException が生じていました。</p> <p>codecDir は現在は wspackage で必須ではありません。</p> <p>この問題は解決済みです。</p>
CR111187	<p>weblogic.webservice.binding.soap.HttpResponse クラスで、メソッド getBodyAsString が「String index out of range」エラーを送出していました。</p> <p>コードの変更により、メソッドはエンドの代わりにボディの長さを使用するようになり、問題は解決されます。</p>

# WebLogic Server 7.0 サービス パック 3 のソリューション

以下の節では、WebLogic Server 7.0 サービス パック 3 で解決された問題を説明します。

- [4-187 ページの「Administration Console」](#)
- [4-195 ページの「コネクタ」](#)
- [4-198 ページの「コア サーバ」](#)
- [4-208 ページの「デプロイメント」](#)
- [4-210 ページの「ドメイン コンフィグレーション ウィザード」](#)
- [4-210 ページの「EJB」](#) [4-225 ページの「JDBC」](#)
- [4-232 ページの「jDriver」](#)
- [4-234 ページの「JMS」](#)
- [4-245 ページの「JNDI」](#)
- [4-247 ページの「JSP とサーブレット」](#)
- [4-261 ページの「JTA」](#)
- [4-266 ページの「プラグイン」](#)
- [4-269 ページの「RMI/RMI-IIOP」](#)
- [4-272 ページの「セキュリティ」](#)
- [4-280 ページの「システム管理」](#)
- [4-288 ページの「ツール」](#)
- [4-293 ページの「Web サービス」](#)
- [4-299 ページの「WebLogic Tuxedo Connector」](#)
- [4-300 ページの「XML」](#)

# Administration Console

変更要求番号	説明
CR069284	Administration Console の [ ドメイン   コンフィグレーション   アプリケーション ] タブで [ 自動デプロイを有効化 ] 属性をコンフィグレーションできなくなりました。自動デプロイメントの制御については、『WebLogic Server アプリケーションの開発』の「 <a href="#">自動デプロイメント</a> 」を参照してください。
CR069927	[ サーバ   モニタ   プロセス出力 ] を選択し、表示オプションを選択しても、出力が表示されませんでした。ノード マネージャが動作している状態で [ ノード マネージャ出力を表示 ] を選択すると、ノード マネージャがクラッシュしました。コードの修正で、この問題は解決されました。
CR076953	状況依存 ( 右クリック ) メニュー項目の [ 個別の Bean のポリシーとロールを定義 ... ] を使用してロールを作成し、( 同じ状況依存メニューとフォームを使用して定義された ) ポリシーにロールを追加する場合、目的のリソースは外部 JCom クライアントからは使用できません。 状況依存メニュー項目ではスコープ ロールが作成され、スコープ ロールの外部にあるクライアントはリソースを使用できません。 JCom ノードに [Define JCom Roles] メニュー項目を追加し、修正されたスコープに現在のクライアントが含まれるようにすることで、この問題を解決しました。
CR078764	Web サービスに対する Administration Console のセキュリティ設定が正しく機能しませんでした。 Administration Console で Web サービスに対してポリシーとロールを設定できないようにコードを変更しました。
CR081673	Administration Console で作成されたレルムには RealmMBean の UserLockoutManager がありませんでした。 Administration Console で新しいレルムを作成すると ULMbean が作成されるようになりました。
CR081673	新しいセキュリティ レルムを作成すると、[ ユーザ ロックアウト ] タブに「User lockout feature not supported by the configured providers」というメッセージが表示されました。この問題は、新しいセキュリティ レルムの作成時にユーザ ロックアウト マネージャが作成されるようにコードを修正することで解決されました。

変更要求番号	説明
CR082335	<p>ユーザおよびグループのリスト用に Administration Console によって作成されるカーソルがクローズされない場合に、メモリ リークが発生する可能性があります。</p> <p>コードの修正によって、カーソルはクローズされます。</p>
CR083530	<p>[ サーバ   ログ   HTTP ] コンソール ページに、新しい属性 <code>LogTimeInGMT</code> が追加されました。この属性では、ホスト コンピュータで指定されたローカル タイムゾーンに関係なく、HTTP ログ メッセージのタイム スタンプがグリニッジ標準時で書き込まれることを指定します。</p>
CR088262	<p>JMS サーバの移行可能な対象が保存され、対象が以前の設定に戻される問題が修正されました。</p>
CR089652	<p>Administration Console から管理対象サーバの SSL ポートを無効にすると、次のような <code>AssertionError</code> が発生しました。</p> <pre>weblogic.utils.AssertionError: ***** ASSERTION FAILED ***** [ ServerIron for server not found ]</pre> <p>検証プロセスで管理対象サーバのコンフィグレーション MBean をチェックするようにすることで、問題を修正しました。</p>
CR090059	<p>WebLogic Server Administration Console では、WebLogic Security サービスのセキュリティ チェックの実行方法を指定する必要があります。これを指定するには、WebLogic Server の起動時に設定するコマンドライン引数である <code>fullyDelegateAuthorization</code> フラグを使用します。</p> <p><code>fullyDelegateAuthorization</code> フラグの値が <code>false</code> の場合、WebLogic Security サービスは関連付けられているデプロイメント記述子 (DD) でセキュリティが指定されている URL および EJB リソースでのみセキュリティ チェックを行います。これはデフォルト設定です。</p> <p>『WebLogic リソースのセキュリティ』の「<a href="#">fullyDelegateAuthorization フラグについて</a>」を参照してください。</p>
CR090266	<p><code>config.xml</code> の <code>Application</code> タグには、Administration Console から設定できない <code>LoadOrder</code> 属性があります。</p> <p>コードを変更して、モジュール ページに設定を追加しました。</p>



変更要求番号	説明
CR090266	<p>Administration Console から config.xml 内の Application タグの LoadOrder 属性を設定できませんでした。</p> <p>アプリケーション モジュール (Application、ConnectorComponent、EJBComponent、WebAppComponent、JDBCPoolComponent) に LoadOrder 属性を追加して、この問題を修正しました。</p>
CR090481	<p>以前のサービス パックでは、NTFS ボリュームに WebLogic Server をインストールした場合、代わりにログ ファイルの場所を指定すると、Administration Console でサーバ ログを表示できませんでした。ログの場所の設定は次のとおりです。</p> <pre data-bbox="366 560 1042 661">&lt;Server Name="myserver" ServerVersion="7.0.0.0"&gt;   &lt;Log FileName="c:\temp\myserver.log" Name="myserver"/&gt; &lt;/Server&gt;</pre> <p>サーバは java.io.FileNotFoundException を送出しました。</p> <p>FileName 属性の呼び出しをカスタムのゲッターに移動して (以前の FileStreamHandler への依存を解消して)、LogFileSearchServlet が MBean から直接 FileName 属性を取得するようにすることで、この問題を解決しました。</p>
CR091170	<p>Examples サーバ上で新しい Web アプリケーション、EJB、またはコネクタを複製しようとする、Administration Console には次のようなメッセージの予期しないエラー ページが表示されました。</p> <pre data-bbox="366 977 1210 1003">javax.management.InvalidAttributeValueException: Path is NULL</pre> <p>解決策として、ApplicationTable、EJBComponentTable、WebAppComponentTable、ConnectorComponentTable から [ クローン ] アイコンを削除しました。</p>
CR091325	<p>Administration Console から JMS トピックおよびキューのポリシーをコンフィグレーションできるようになりました。</p>
CR091359	<p>Administration Console のオプション</p> <p>[Web Application   webapp1   モニタ   すべての アクティブな Web アプリケーションのモニタ ...] は、EAR ファイル内の Web アプリケーションでは失敗していました。</p> <p>この問題は、コードを修正して解決されました。</p>
CR093289	<p>Administration Console に表示されるログ メッセージで、UTF8 エンコーディングの使用時にそのテキストの一部が切り捨てられる問題が修正されました。</p>

変更要求番号	説明
CR095556	<p>アクセス ログの [ ローテーション時間 ] の日付の書式として 00:00 を指定すると、Administration Console は次のような <code>AssertionError</code> を送出していました。</p> <pre>weblogic.utils.AssertionError: ***** ASSERTION FAILED ***** [ Could not parse LogRotationTimeBegin: 00:00 because Unparseable date: "00:00" ]</pre> <p>Log MBean と WebServer MBean に時間の書式の検証を追加することで、問題を修正しました。</p>
CR095694	<p>日本語の <code>weblogic.properties</code> ファイル コンバータと関連するファイル ブラウザの表示上の問題が修正されました。</p>
CR098353	<p>TransactionLogFileWritePolicy 属性が 7.0 SP01 で追加されましたが、Administration Console でエクスポートされませんでした。</p> <p>この属性を設定するフィールドが、[ サーバ   ログ   JTA ] タブに追加されました。</p>
CR099721	<p>Administration Console を使用して [CookieMaxAgeSecs] を編集した後で、値が編集前の値に戻ることがありました。</p> <p>コードを変更して値の下限を削除することで、問題を修正しました。</p>
CR102982	<p>[ クライアント証明書プロキシを有効化 ] チェック ボックスがサーバの [ チューニング ] タブに表示されていましたが、この属性はチューニングに関係ありません。</p> <p>[ クライアント証明書プロキシを有効化 ] フィールドは、サーバの [ コンフィグレーション   一般 ] タブに移動しました。</p>
CR103104	<p>UserReader MBean は Administration Console にユーザを表示しませんでした。認証プロバイダは、UserReader ではなく UserEditor を実装する場合に Console にユーザを読み込むことができます。ユーザを読み込もうとすると、「ユーザの作成をサポートする認証プロバイダがありません。」というメッセージが表示されます。グループの場合 (GroupEditor および GroupReader インタフェース) にも同じことが発生します。</p> <p>Administration Console は <code>userEditor.CreateUser</code> メソッドを呼び出し、<code>UserReader.listUsers</code> メソッドを呼び出しませんでした。</p> <p>適切なプロバイダ タイプを探すように MBean を修正しました。</p>

---

変更要求番号	説明
CR105624	<p data-bbox="366 256 1233 352">サーバが MSI (管理対象サーバ独立) モードで起動された場合、管理対象サーバのアドレスとポートから WebLogic Server Administration Console にアクセスできませんでした。</p> <p data-bbox="366 366 998 392">コードを変更して、この不適切なアクセスを制限しました。</p>
CRO68646	<p data-bbox="366 418 1233 543">weblogic.Admin ユーティリティでは、指定されたプロトコルに関係なく T3 プロトコルが使用されていました。たとえば、コマンド「<code>java weblogic.Admin - url http://localhost:7001</code>」では T3 を使用して、localhost:7001 でリスンするサーバインスタンスに接続していました。</p> <p data-bbox="366 557 1233 682">このユーティリティは、指定すれば HTTP または HTTPS を使用するよう更新されています。それらのプロトコルを使用するには、HTTP トンネリングを有効にする必要があります。Administration Console オンライン ヘルプの「<a href="#">HTTP プロトコルのコンフィグレーション</a>」を参照してください。</p>

---

## クラスタ

変更要求番号	説明
CR086319	<p>特定の状況で、管理サーバの下で動作する管理対象サーバの停止時に、ノードマネージャが不要なスタックトレースを書き込みました。</p> <p>この問題は、クエリの応答が返される前に管理対象サーバが一定期間待機するために <code>URLConnection.getInputStream</code> が返されないことで発生しました。これは、管理対象サーバを継続的にモニタするための意図的な動作でした。ただし、管理対象サーバが停止するときに、遅れたサーブレット応答は送信されず、ソケットがクローズされました。続いてノードマネージャは接続を再び開こうとしましたが、サーバが動作していないため <code>openServer</code> 呼び出しは失敗しました。これに応じて、<code>openServer</code> によって <code>ConnectException</code> が送出されログに記録されました。</p> <p>ノードマネージャは不要なスタックトレースを書き込まなくなりました。</p>
CR093809	<p>スタックトレースが、セッションのルックアップエラーから生じていました。セカンダリサーバインスタンスが停止されていました。対応した要求のログエントリを作成するときに、HTTPサーバはセッションからユーザ情報を取得しようとしていました。結果として、それがダウンしていてもセカンダリサーバをルックアップしようとして、次のようなスタックトレースが生じていました。</p> <pre>&lt;Dec 26, 2002 7:17:55 PM EST&gt; &lt;Error&gt; &lt;HTTP Session&gt; &lt;Error looking up session for id:2LcR6Oo12IC5qNtFZhThJClYVYEXt9KWG2ciXmAXZ4XiBAHPiHx4!-137950 51 94!stcasfi01b.usa-ed.net!8001!7002!-1127797657!stcasfi02b.usa-ed. net!8001!7002 java.rmi.ConnectIOException: Server is being shut down Start server side stack trace: java.rmi.ConnectIOException: Server is being shut down at weblogic.rjvm.RJVMImpl.dispatchRequest(RJVMImpl.java:692) at weblogic.rjvm.RJVMImpl.dispatch(RJVMImpl.java:666) at...</pre> <p>プライマリサーバは利用可能なサーバのリストから新しいセカンダリを適切に選択していたので、セッションデータは失われませんでした。</p> <p>この問題は、コードを修正して解決されました。現在、サーバはセッションではなく <code>ThreadLocal</code> からユーザ情報を取得します。</p>
CR094561	<p><a href="http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA03-26.01.jsp">http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA03-26.01.jsp</a> のセキュリティ勧告に関する情報を確認してください。</p>

変更要求番号	説明
CR094837	<p>次のような手順で <code>ClassCastException</code> が発生しました。サーバを定義する。サーバをクラスタに割り当てる。サーバと同じ名前の Unix マシンを作成する。Unix マシンとサーバを互いに割り当てる。管理サーバを停止および再起動する。クラスタまたはサーバリストからサーバ名を選択する。管理対象サーバを起動する場合にも <code>ClassCastException</code> が発生しました。</p> <p>起動時に <code>UnresolvedMBean</code> を解決するプロセスでの問題を修正して、この問題を解決しました。</p>
CR104430	<p>クライアントがクラスタ内の 1 番目と 2 番目のサーバ (クライアントのプライマリサーバとセカンダリサーバ) から 3 番目のサーバに切り替えるときに、クラスタ化されたサーバでセッションが失われる可能性がありました。プロセスは最初の 2 つのサーバからセッションを削除します。クライアントがプライマリサーバに戻ると、プライマリサーバは 3 番目のサーバ上ではなく、セカンダリサーバ上でセッションを探しました。</p> <p>3 番目のサーバ上でセッション情報からセッションを再作成し、プライマリおよびセカンダリサーバからセッションを完全に削除するようにコードを修正して、問題を解決しました。</p>
CR105482	<p>2 つの実行中の管理対象サーバが含まれるクラスタで新しい管理対象サーバを起動すると、実行中のいずれかのサーバで <code>ConcurrentModificationException</code> が発生しました。</p> <p>クラスタのメンバーが同期化の要件を満たすようにコードを変更して、問題を修正しました。</p>
CR105698	<p>2 つのクラスタ間に同じ JNDI 名の 2 つの分散キューがあり、メッセージ駆動型 Bean が両方のクラスタにデプロイされている場合、メッセージはいずれかのクラスタのみで消費されました。</p> <p>メッセージ駆動型 Bean はドメイン内の分散キューの集合全体に機能し、メンバーを適切に追加するようになりました。</p>

---

変更要求番号	説明
CR106384	<p>クラスタ内に 1 つの管理サーバと 2 つの管理対象サーバがありました。application1.ear には EJB と Web アプリケーションが含まれていました。EJB はクラスタにデプロイされていました。Web アプリケーションは仮想ホストにデプロイされ、仮想ホストはクラスタに割り当てられていました。</p> <p>application1.ear には、仮想ホストのデフォルト Web アプリケーションとしてコンフィグレーションされた application1.war が含まれていました。さらに、各クラスタ ノードには独自のデフォルト Web アプリケーションがありました。</p> <p>デプロイメントは成功しましたが、application1.ear をアンデプロイしようとする以下例が発生しました。アプリケーションは、ドメイン全体を再起動しても再デプロイできませんでした。</p> <pre>Exception caught for task Unprepare application application1 on mycluster,application1: Start server side stack trace: java.lang.reflect.UndeclaredThrowableException: java.lang.reflect.InvocationTargetException: javax.management.ListenerNotFoundException: listener: ServletContext (id=559103557,name=DefaultWebApp,context-path=)</pre> <p>Web コンテナは、すべての WAR をアンデプロイして、デフォルト Web アプリケーションをアンデプロイしようとしてしました。デフォルト Web アプリケーションはすでにアンデプロイされているため、最初の WAR は成功しましたが、2 番目の WAR は失敗しました。</p> <p>対象の Web アプリケーションのコンテキストが http サーバ上に存在しない場合、デフォルト Web アプリケーションがアンデプロイされないようにコードを変更しました。その修正で、問題は解決されました。</p>
CR107273	<p>クラスタで、MulticastTTL に値「0」を設定すると、WebLogic Server は設定を無視してデフォルト値の 1 を取得し、次のようなエラーを送出しました。</p> <pre>"clusterdomain:Name=mycluster,Type=Cluster" is smaller than the minimum allowed: 1</pre> <p>値を「0」にできるようにコードを変更して問題を解決しました。</p>

---

変更要求番号	説明
CR134233、 CR102655	<p>WebLogic Server 6.1 SP04 では、シリアライズされていないオブジェクトのレプリケートが失敗した後にセッション レプリケーションが機能を停止していました。シリアライズされていないデータをセッション オブジェクトに設定する JSP の呼び出しでは、次のような予測可能な例外が送出されていました。</p> <pre>&lt;Mar 19, 2003 2:58:23 PM PST&gt; &lt;Error&gt; &lt;Cluster&gt; &lt;All session objects should be serializable to replicate. Please check the objects in your session. Failed to replicate non serializable object&gt;</pre> <p>その後、それ以降のすべてのリクエストについて、セッションがレプリケートされず、次の例外が生じていました。</p> <pre>&lt;Mar 19, 2003 2:58:48 PM PST&gt; &lt;Debug&gt; &lt;Cluster&gt; &lt;Unable to create secondary for -8956818414963087828&gt;</pre> <p>この問題は、コードを修正して解決されました。現在は、シリアライズできないオブジェクトに遭遇したときに、メソッドは他のセカンダリを試すことなく復帰します。</p>
CR110385	<p>2 つの管理対象サーバ間の NetworkChannel で URL を作成するときに HomeHandleImpl クラスで間違ったポート番号が作成されないように、listenPort と sslListenPort はデフォルト値で初期化されるようになりました。</p>

## コネクタ

変更要求番号	説明
CR069426	<p>max-capacity と initial-capacity が同じ場合に RAR をデプロイすると、コンフィグレーション例外が送出されていました。</p> <p>max-capacity と initial-capacity が等しいときに値が考慮されないよう、weblogic-ra.xml の .xpi の capacity-increment のチェックが変更されました。また、max-capacity と initial-capacity が等しい場合には増加容量をゼロに設定できるようになっています。</p>
CR086862	<p>「Unable to locate context: java:/comp/env/WebLogic Server-connector-resref」などの、説明のつかないログ メッセージがコネクタのログ ファイルに記録されていました。</p> <p>コネクタの担当チームが、拡張の意味で大量のログ メッセージを削除しました。</p>

変更要求番号	説明
CR091106	コネクタ RAR ファイルの再デプロイ後、WebLogic Server の組み込み LDAP サーバは管理対象サーバへのデータの複製を停止していました。 コードの変更によって、レプリケーションの動作は改善されます。
CR100707	7.0 SP3 より前の WebLogic Server で JCA を使用すると、 <code>java.naming.CompositeName</code> で <code>java.lang.ClassCastException</code> が送出されてきました。WebLogic Server 8.1 で JCA を使用した場合は、Web アプリケーションの場合に、 <code>resource-ref</code> のエントリ (Auth、SharingScope) で <code>null</code> が返されてきました。 この問題は内部エラーが原因で発生し、コードの修正で解決されました。
CR101761	7.0 SP3 より前の WebLogic Server で JCA を使用する場合に、 <code>java.naming.CompositeName</code> で <code>java.lang.ClassCastException</code> が送出されてきました。WebLogic Server 8.1 で JCA を使用した場合は、Web アプリケーションの場合に、 <code>resource-ref</code> のエントリ (Auth、SharingScope) で <code>null</code> が返されてきました。 この問題は内部エラーが原因で発生し、コードの修正で解決されました。



---

変更要求番号	説明
CR101906	<p>クライアントはEJBのメソッドを呼び出しました。ManagedConnection (MC) とEISの間で接続が失敗したときに、XA トランザクションの場合に、コンテナはXA トランザクションがロールバックされた後で60秒ごとにXAResource.rollback と ManagedConnection.cleanup を呼び出しました。EIS に対する以降の呼び出しは java.lang.NullPointerException 例外になりました。</p> <pre>&lt;Mar 21, 2003 5:50:52 PM PST&gt; &lt;Error&gt; &lt;Connector&gt; &lt;190041&gt; &lt;&lt; JCA Resource Adapter for ClearPath MCP_eis/comsRAJNDINAMEBrazil &gt; The returned ManagedConnection instance is null&gt; &lt;Mar 21, 2003 5:50:52 PM PST&gt; &lt;Info&gt; &lt;EJB&gt; &lt;010051&gt; &lt;EJB Exception during invocation from home: drummer.CedarBank.CedarBankBeanBrazil_ris6bf_HomeImpl@12fc36 threw exception: java.lang.NullPointerException java.lang.NullPointerException at drummer.CedarBank.CedarBankBeanBrazil.cedarBank(CedarBankBeanB razil.java:130) at drummer.CedarBank.CedarBankBeanBrazil_ris6bf_EOImpl.cedarBank( CedarBankBeanBrazil_ris6bf_EOImpl.java:46) at drummer.CedarBank.CedarBankBeanBrazil_ris6bf_EOImpl_WebLogic Serverkel.invoke(Unknown Source) at weblogic.rmi.internal.BasicServerRef.invoke(BasicServerRef.jav a:362) at weblogic.rmi.cluster.ReplicaAwareServerRef.invoke(ReplicaAware ServerRef.java:114) at weblogic.rmi.internal.BasicServerRef\$1.run(BasicServerRef.java :313) at weblogic.security.service.SecurityServiceManager.runAs(Securit yServiceManager.java:785) at weblogic.rmi.internal.BasicServerRef.handleRequest(BasicServer Ref.java:308) at weblogic.rmi.internal.BasicExecuteRequest.execute(BasicExecute Request.java:30) at weblogic.kernel.ExecuteThread.execute(ExecuteThread.java:153) at weblogic.kernel.ExecuteThread.run(ExecuteThread.java:134)</pre> <p>WebLogic Server は接続を破棄する前にトランザクションの完了を待機していませんでした。</p> <p>コードの修正で、この問題は解決されました。</p>

---

---

変更要求番号	説明
CR106388	<p>ra-link-ref 要素を使用する RAR (「論理的な」RAR) をデプロイするときに、論理的な RAR デプロイメントは、weblogic-ra.xml デプロイメント記述子で指定された正しいプール名または max-capacity を取得しませんでした。そのため WebLogic Integration で予期しない動作が発生しました。</p> <p>要素は正しく機能するようになりました。</p>

---

## コア サーバ

---

変更要求番号	説明
CR070887	<p>JDK1.3.1 では、java.net.HttpURLConnection で getInstanceFollowRedirects () および setInstanceFollowRedirects () というメソッドが追加されました。これらのメソッドを使用すると、特定の HttpURLConnection で URL のリダイレクトを無効にできます。</p> <p>HttpURLConnection は、setInstanceFollowRedirects () で設定されたフラグを無視していました。</p> <p>この問題は、InstanceFollowRedirects の設定に従ってプロセスがリダイレクトするように WebLogic Server の HttpURLConnection リダイレクト ロジックをコード修正して解決されました。</p>
CR071415	<p>WebLogic Server のセキュリティ マネージャが有効になっていて、ポリシー ファイルが設定されているときに、JSP のコンパイル時に javac に path /usr/lib が付加されましたが、java.security.AccessControlException になりました。</p> <p>Executable.resolveExecutable () は、javac コンパイラの位置を確認するときに間違ったパスを検索していました。</p> <p>Executable.resolveExecutable () が java.library.path を検索する前に JavaCompiler で指定された絶対パスを検索するように、コードを変更しました。</p>

---

---

変更要求番号	説明
CR077170	<p>WebLogic Server 6.1 SP03 では、管理サーバが停止された後に管理対象サーバを停止すると例外が発生しました。この問題は、アクションが次の順序で行われたときに発生しました。</p> <ol style="list-style-type: none"><li>1. 管理サーバと管理対象サーバを起動する。</li><li>2. 管理サーバを停止する。</li><li>3. 管理対象サーバの ServerConfigMBean の「停止」メソッドを呼び出す。</li></ol> <p>次の例外が送出されました。</p> <pre>Unexpected Exception Start server side stack trace: java.rmi.ConnectException: Unable to get direct or routed connection to: '-19546889165621794S:dwarkamai:[7001,7001,-1,-1,7001,-1,-1]:OA Mdomain:adminServer' at weblogic.rmi.internal.BasicOutboundRequest.sendReceive(BasicOu tboundRequest.java:109) at weblogic.rmi.internal.BasicRemoteRef.invoke(BasicRemoteRef.jav a:127) at...</pre> <p>weblogic.Admin からの shutdown コマンドは、成功していました。</p> <p>この問題は、管理サーバがダウンした後で管理対象サーバを停止するときに発生する例外を捕捉することで解決しました。</p>
CR083209	<p>NTSocketedMuxer で Windows XP をサポートできるようにコードが修正されました。</p>

---

---

変更要求番号	説明
CR083762	<p data-bbox="310 256 1170 352">クライアント上のデータソースから接続を取得しようとしたときに <code>NoSuchObjectExceptions</code> が原因で障害が発生しました。次のメッセージが送出されました。</p> <pre data-bbox="310 366 1170 986">Error dropping table: Unexpected Exception java.rmi.NoSuchObjectException: The object identified by: '275' could not be found. Either it was has not been exported or it has been collected by the distributed garbage collector. at weblogic.rjvm.BasicOutboundRequest.sendReceive(BasicOutboundRequest.java:109) at weblogic.rmi.internal.BasicRemoteRef.invoke(BasicRemoteRef.java:120) at weblogic.jdbc.common.internal.RmiDataSource_WebLogicServletub.getConnection(Unknown Source) at com.bea.cts.CTSDeployment.executeSQL(CTSDeployment.java:2445) at com.bea.cts.CTSDeployment.undeploy(CTSDeployment.java:711) at com.sun.cts.harness.SuiteSynchronizer.continueToUndeployApps(SuiteSynchronizer.java:1082) at com.sun.cts.harness.SuiteSynchronizer.undeployApps(SuiteSynchronizer.java:982) at com.sun.cts.harness.SuiteSynchronizer.doDeployment(SuiteSynchronizer.java:289) at com.sun.cts.harness.CTSGUIHarnessObserver.starting(CTSGUIHarnessObserver.java:167) at javasoft.sqe.javatest.TestRunner\$Notifier.starting(TestRunner.java:287) at javasoft.sqe.javatest.TestRunner\$Worker.runTest(TestRunner.java:239) at javasoft.sqe.javatest.TestRunner\$Worker.run(TestRunner.java:221)</pre> <p data-bbox="310 999 1170 1060">分析の結果、クライアントにまだその参照があるにもかかわらず、データソースがガベージコレクションされていることが判明しました。</p> <p data-bbox="310 1074 1170 1133">この問題は、クライアントが参照するオブジェクトでガベージコレクションが行われないようにガベージコレクションをコード修正して解決されました。</p>
CR087364	<p data-bbox="310 1164 1170 1225">Administration Console で [ トンネリングを有効化 ] 属性を変更するには、新しい値が有効になるようにサーバを再起動する必要があります。</p> <p data-bbox="310 1239 1170 1289">この問題は、サーバを再起動することなくトンネリングを有効化できるようにコード修正して解決されました。</p>

---

変更要求番号	説明
CR087808、 CR095487	<p>Sunblade 100 シングル CUP Solaris マシンでは、2 つの独立したサーバインスタンスが相互に InitialContext をルックアップできませんでした。一方のサーバインスタンスが InitialContext をルックアップし、同じマシン上の 2 番目のサーバインスタンスが同じマシン上の InitialContext をルックアップしようとした後、次の例外が発生しました。</p> <pre>&lt;2002/10/09 4:23:56:JST&gt; &lt;Debug&gt; &lt;ConnectionFactory&gt; &lt;Attempt to sendMsg using a closed connection&gt; javax.naming.CommunicationException. Root exception is java.rmi.ConnectException: Attempt to sendMsg using a closed connection at weblogic.rmi.internal.BasicOutboundRequest.sendReceive(BasicOutboundRequest.java:85) at weblogic.rmi.cluster.ReplicaAwareRemoteRef.invoke(ReplicaAwareRemoteRef.java:262) at weblogic.rmi.cluster.ReplicaAwareRemoteRef.invoke(ReplicaAwareRemoteRef.java:229) at weblogic.rmi.internal.ProxyStub.invoke(ProxyStub.java:35) at \$Proxy45.lookup(Unknown Source) at weblogic.jndi.internal.WLContextImpl.lookup(WLContextImpl.java:341) at javax.naming.InitialContext.lookup(InitialContext.java:345) at com.bea.samples.servlet.TestServlet.service(TestServlet.java:40) at javax.servlet.http.HttpServlet.service(HttpServlet.java:853) at weblogic.servlet.internal.ServletStubImpl.invokeServlet(ServletStubImpl.java:265) at weblogic.servlet.internal.ServletStubImpl.invokeServlet(ServletStubImpl.java:200) at weblogic.servlet.internal.WebAppServletContext.invokeServlet(WebAppServletContext.java:2546) at weblogic.servlet.internal.ServletRequestImpl.execute(ServletRequestImpl.java:2260) at weblogic.kernel.ExecuteThread.execute(ExecuteThread.java:139) at weblogic.kernel.ExecuteThread.run(ExecuteThread.java:120)</pre> <p>この問題は、Sun Enterprise 上では再現されませんでした。localhost の代わりに IP アドレスを使用してルックアップを実行すると、Sun Blade でこの問題は発生しませんでした。分析の結果、クライアントの RJVM が重複する T3JVMConnections を閉じようとするのが判明しました。サーバに対して CMD_REQUEST_CLOSE が発行され、RJVM が閉じられました。ただし、サーバがこのメッセージをキューに入れると、RJVM は閉じたものとしてマークされました。</p> <p>コードの変更により、重複接続を検出しているときの RJVM の停止が防止され、CMD_REQUEST_CLOSE はサーバに配信されません。</p>

変更要求番号	説明
CR088022	<p>WebLogic Server Administration Console では、ノード マネージャで管理サーバおよび管理対象サーバを起動するときに接続情報が表示されません。</p> <p>コードの修正により、ナビゲーション ツリーでサーバ名を右クリックして [ 接続を表示 ] を選択することで接続情報を表示することができます。</p>
CR089454	<p>リクエスト キューによる実行キューの待機を -D フラグで制限して、受信トラフィックを抑制できるようになりました。</p>
CR092454	<p>クライアント仮想マシンから CommoMBeanServerProxy クラスを介して COMMO Bean にアクセスするメソッドは、「bean」の値を設定するコードが Kernel.isServer() によって回避された後で、ObjectName が MBean インスタンスまたは MBean タイプ オブジェクト名のどちらであるかを確認しようとすると、Null ポインタ例外をトリガすることがありました。</p> <p>コードの変更により、早い回避が防止されます。</p>
CR092704	<p>WebLogic Server 6.1 SP02 では、ソケット リーダー スレッドがブロックされることが原因で頻繁にハングが発生しました。POLL ロックを所有するスレッドが SSL ソケットを閉じようとしたましたが、sendRecord() メソッドにおいて必要な出力ストリームのロックを取得できないために、処理を進めることができませんでした。スタックトレースは次のようなものでした。</p> <pre data-bbox="310 899 1166 1182"> "ExecuteThread: '23' for queue: 'default'" daemon prio=5 tid=0x6d6c40 nid=0x24 waiting for monitor entry [0xe4e81000..0xe4e81a28] at weblogic.security.SSL.SSLSocket.sendRecord(SSLSocket.java:1049 ) at weblogic.security.SSL.SSLSocket.sendAlert(SSLSocket.java:1007) at weblogic.security.SSL.SSLSocket.close(SSLSocket.java:1153) at weblogic.security.SSL.SSLSocket.close(SSLSocket.java:1141) at weblogic.socket.SocketMuxer.closeSocket(SocketMuxer.java:236 </pre> <p>アポートによる停止に対するクローズでは SSL ソケットがソケットにデータを書き込まないよう、コードが修正されました。</p>

変更要求番号	説明
CR092933	<p>次のコンフィグレーションで、スレッド ダンプ エラーが発生しました。</p> <pre>VM: java.version='1.3.1_02' os.name='windows 2000' java.vendor.url='http://java.sun.com/'</pre> <p>Hotspot クライアントおよび Version 1.3.1_x 仮想マシンに対して JVM_DumpAllStacks シンボルがグローバルに宣言されていなかったため、スレッド ダンプが失敗しました。</p> <p>1.3.1_X Hotspot クライアント / サーバ JVM において、weblogic.Admin を使用するスレッド ダンプが可能になりました。</p>
CR093320	<p>サブレット コンテナに関する内部的な問題により、クラスがロードできなくてもソースが返されていました。この問題は、WebLogic Server 8.1 サーバ上の JDBC 接続プールからデータベース接続を取得する WebLogic Server 7.0SP2 クライアントをテストしているときに確認されました。このエラーが原因で、クライアントはデータベース接続を取得できませんでした。</p> <p>この問題は、コードを修正して解決されました。</p>
CR094697	<p>クライアントで「Server XYZ failed to shutdown successfully ...」というエラーメッセージが表示されてもサーバが適切に停止する問題が、コードの修正で解決されました。</p>
CR094724	<p>HPUX 上の WebLogic Server ネイティブ ライブラリが、(aCC ではなく) cfront を使用してコンパイルされていました。aCC は、1998 年に cfront に取って代わった ANSIC コンパイラです。</p> <p>その結果、互換性のないランタイム ライブラリがロードされていました (cfront のコンパイルでは libc.2、aCC でコンパイルされた SUN の Java では libCsup.2)。互換性のないランタイム ライブラリはクラッシュを引き起こします。</p> <p>この問題は、hpux11 で aCC を使用するよう WebLogic Server のビルドを修正して解決されました。</p>

---

変更要求番号	説明
CR095487	<p>同じサーバを示す異なるサーバ URL で複数の <code>getInitialContext()</code> 呼び出しを試行する t3 クライアント。通常の状態では、クライアントはそれが重複接続であると判断し、CLOSE メッセージをサーバに送信して、自分側の接続を閉じます。サーバは、この重複接続を削除します。状況によっては、CLOSE メッセージは非同期 IDENTIFY リクエストのためにキューに入れられ、クライアントは CLOSE の配信を待つことなく接続の自分の側を閉じます。サーバはこのソケットが削除されていることを検出し、そうでなければ正常な RJVM を停止する <code>ConnectionManager</code> の <code>shutdown</code> を発行します。</p> <p>コードの修正によって、CLOSE メッセージが配信されないときに重複接続を閉じるときの RJVM 停止が防止されます。</p>
CR096049	<p>クラスタが管理サーバと 2 つの管理対象サーバで構成されているシステムでは、管理対象サーバの再起動に問題があります。管理対象サーバの片方が動作しているマシンを再起動した後、その管理対象サーバは再起動できず、次の例外が送出されます。</p> <pre data-bbox="310 751 1170 977">&lt;Jan 22, 2003 5:27:49 PM EST&gt; &lt;Error&gt; &lt;Deployer&gt; &lt;149204&gt; &lt;The Deployment framework was unable to register with the Data Replication Service. RegisterException due to underlying exception java.rmi.RemoteException: Failed to send message to URL t3://admtest1:7001; nested exception is: javax.management.InstanceNotFoundException: myDomain:Location=AdminServer,Name=***servername***,ServerRuntime=AdminServer,Type=DeploymentTaskRuntime</pre> <p>この問題は、管理対象サーバがコンフィグレーション MBean に基づいてアプリケーションをロードできるようにコードを修正して解決されました。</p>

---



変更要求番号	説明
CR096291	<p>BeanA の doSomethingOnA で 1 つの EJB ビジネス メソッドを呼び出す起動クラス。このメソッドは BeanB のホーム インタフェースを取得し、次の ClassCastException が送出されます。</p> <pre data-bbox="413 361 1233 1185"> &lt;29 janv. 03 13:21:19 CET&gt; &lt;Info&gt; &lt;EJB&gt; &lt;010051&gt; &lt;EJB Exception during invocation from home: test_ccex.a.BeanABean_124zg1_HomeImpl@5c5ca2 threw exception: java.lang.ClassCastException: Cannot narrow remote object to test_ccex.b.BeanBHome java.lang.ClassCastException: Cannot narrow remote object to est_ccex.b.BeanBHome at weblogic.iop.PortableRemoteObjectDelegateImpl.narrow(Porta bleRemoteObjectDelegateImpl.java:223) at javax.rmi.PortableRemoteObject.narrow(PortableRemoteObject. java:132) at test_ccex.a.BeanABean.doSomethingOnA(BeanABean.java:49) at test_ccex.a.BeanABean_124zg1_EOImpl.doSomethingOnA(BeanABea n_124zg1_EOImpl.java:46) at java.lang.reflect.Method.invoke(Native Method) at com.csg.am.dcf.weblogic.startup.BeanExecutor.startup(BeanExe cutor.java:183) at weblogic.t3.srvr.StartupClassRunner.invokeStartup(StartupCl assRunner.java:141) at weblogic.t3.srvr.StartupClassRunner.invokeClass(StartupClas sRunner.java:122) at weblogic.t3.srvr.StartupClassRunner.access\$0(StartupClassRu nner.java:113) at weblogic.t3.srvr.StartupClassRunner\$1.run(StartupClassRunne r.java:86) at weblogic.security.service.SecurityServiceManager.runAs(Secu rityServiceManager.java:744) at weblogic.t3.srvr.StartupClassRunner.run(StartupClassRunner. java:71) at java.lang.Thread.run(Thread.java:479) </pre> <p>コードの修正で、RemoteObjectReplacer のクラスロードの問題は解決されました。</p>
CR097077	<p>多くの発信 RMI 呼び出しを行うカスタム レルムを使用するときに、発信呼び出しを行うことによってリーダー スレッドがブロックされ、応答を読み込むためのスレッドが残されませんでした。この問題は、リーダー スレッドの RJVM レイヤからの BootServicesImpl.invoke メソッドが原因で発生します。</p> <p>BootServicesImpl は、デフォルトの実行キューにディスパッチできるように RMI レイヤに移動されました。</p>

変更要求番号	説明
CR099307、 CR102154、 CR100469、 CR104186、 CR104352、 CR103539	重複した t3 接続をクローズするときに、ロックが取得される順序が原因で、スレッド (PosixSocketMuxer および RJVM ConnectionManager) がデッドロック状態に入るおそれがありました。 マルチプレクサで要求をディスパッチするときに FDRRecord ロックが行われないうちになりました。
CR099489	InitialContext からルックアップした後、PortableRemoteObject を使用してホームインタフェースが限定されたときに <code>java.lang.ClassCastException</code> が送出されていました。 コードの修正により、返される型は正しく限定されました。
CR100177	<code>weblogic.security.service.ServerResource</code> の使用が、Administration Console でのサーバリソースの作成方法、およびドキュメントと一致していませんでした。T3 サーバを保護する目的で使用する場合、WebLogic リソースの名前は常に「T3Srvr」でした。他のすべてのケースでは、基盤のサーバの名前が WebLogic リソースの名前として使用されていました。 この問題は、基盤のサーバの名前を常に WebLogic リソースの名前として使用することで解決されました。
CR101720	接続フィルタを有効にすると、 <code>Connection rejected</code> メッセージが蓄積されて、ログファイルが一杯になっていました。 新しい <code>ConnectionLogger</code> プロパティを使用すると、そのようなメッセージをログファイルに記録するかどうかをコンフィグレーションできます。
CR102848	CoreHealthMonitor は、大量の作業を実行しているときに <code>ExecuteThreadManager</code> ロックを保持して、デッドロックを招き、Administration Console がフリーズしました。CoreHealthMonitor が <code>ExecuteThreadRuntimeMBean</code> を使用してスタックスレッドのリストを取得し、ETM ロックを保持しているときに <code>ExecuteThreadManager</code> 内のすべての場所からのロギングを回避するように、コードを変更しました。

---

変更要求番号	説明
CR103721	<p>t3 クライアントは、クラスタ化された EJB からコンテンツ スイッチを介して <code>InitialContext</code> を取得するときにタイムアウトし、次のような例外が送出されました。</p> <pre>javax.naming.CommunicationException. Root exception is java.net.ConnectException: t3://10.161.39.60:9001: Bootstrap to: 10.161.39.60/10.161.39.60:9001' over: 't3' got an error or timed out at weblogic.rjvm.RJVMFinder.findOrCreate(RJVMFinder.java:180) at weblogic.rjvm.ServerURL.findOrCreateRJVM(ServerURL.java:262) at weblogic.jndi.WLInitialContextFactoryDelegate.getInitialContext (WLInitialContextFactoryDelegate.java:323) at weblogic.jndi.WLInitialContextFactoryDelegate.getInitialContext (WLInitialContextFactoryDelegate.java:221) at weblogic.jndi.WLInitialContextFactory.getInitialContext(WLInitial ContextFactory.java:149) at javax.naming.spi.NamingManager.getInitialContext(NamingManager .java:668) at javax.naming.InitialContext.getDefaultInitCtx(InitialContext.j ava:246) at javax.naming.InitialContext.init(InitialContext.java:222) at javax.naming.InitialContext.&lt;init&gt;(InitialContext.java:198) at Test.main(Test.java:34)</pre> <p>ロード バランサで t3 クライアントのサポートを取り入れて、この問題を解決しました。</p>
CR103967	<p>WebLogic Server 実行スレッドの子スレッドは、正しいコンテキスト クラスローダを継承していませんでした。この問題は、サブレットまたはタイマー (<code>java.util.Timer</code>) によって作成される EJB にも当てはまります。</p> <p><code>java.lang.Thread</code> が独自のメンバー変数を子スレッドに直接割り当てるため、WebLogic Server 実行スレッドは独自のコンテキスト クラスローダを保持し、これらの実行スレッドの子スレッドは継承しません。</p> <p><code>java.lang.Thread</code> に同じコンテキスト クラスローダを設定して、この問題を解決しました。</p>

---

## デプロイメント

変更要求番号	説明
CR089760	<p>weblogic.Deployer ユーティリティで、EAR ファイルをデプロイするときに「alternate descriptor」エラーが生成されました。次に例を示します。</p> <pre>Module: <i>module_name</i> Error: Alternative descriptor <i>descriptor_name</i> can't be found.</pre> <p>この問題は、EAR が抽出されたディレクトリで WebLogic Server が記述子ファイルを検索しなかったために発生しました。</p> <p>weblogic.Deployer は、EAR を正しくデプロイするように修正されました。</p>
CR090678	<p>ドメインの管理サーバを再起動すると、そのドメイン内のすべての管理対象サーバで自動的に EJB が再デプロイされました。</p> <p>管理サーバを起動してもドメインの管理対象サーバで EJB が再デプロイされないようにコードが修正されました。</p>
CR090678	<p>EAR ファイルの一部としてデプロイされたアプレットにアクセスするために、WebLogic Server では不適切な CODEBASE 値を必要としていました。たとえば、Web アプリケーション (MyWar.war) にパッケージ化され、EAR ファイル (MyEar.ear) の一部としてデプロイされたアプレットでは、次のような CODEBASE が必要でした。</p> <pre>CODEBASE=/bea_WebLogic Server_internal/classes/MyEar@MyWar.war</pre> <p>MyWar.war は、アプレットを格納する WAR ファイルの名前です。これは、Web アプリケーションで MyWar の CONTEXT-ROOT が明示的に定義されている場合でも同じでした。この問題は、CONTEXT-ROOT ではなく Web アプリケーションの URI を使用してデプロイメントが行われたことが原因で発生しました。</p> <p>現在は、CONTEXT-ROOT (EAR ファイルの application.xml 記述子で定義されている場合) を使用して Web アプリケーションがデプロイされるようにコードが修正されています。たとえば、上記の EAR ファイルで Web アプリケーションに MyWar の CONTEXT-ROOT が指定されている場合、アプレットの適切な CODEBASE は次のようになります。</p> <pre>CODEBASE=/bea_WebLogic Server_internal/classes/MyEar@MyWar</pre>

変更要求番号	説明
CR100857	<p>weblogic.Deployer で (-activate -upload フラグで) Web アプリケーションを更新しても、適切に動作しませんでした。Web アプリケーションはロードされる前に、非アクティブ化され、ロールバックされて破棄されました。これは別の修正によって発生したリグレッションです。</p> <p>適切な状態で Web アプリケーションをロールバックするようにコードを変更しました。</p>
CR101755	<p>Web コンテナが Tomcat 4.0 で EJB コンテナが WebLogic Server7.0SP1 であるときに、名前が異なり、別々のクラスローダを使用するものの、同じクラスを使用している 2 つの Web アプリケーションが Tomcat 上にデプロイされている場合、1 つ目の Web アプリケーションは EJB を呼び出せますが、2 つ目の Web アプリケーションは ClassCastException により呼び出しに失敗します。</p> <p>新しい -D フラグを使用すると、クライアントは必ずコンテキスト クラスローダ上にスタブを生成することになります。-D フラグは次のとおりです。</p> <pre>-Dweblogic.LoadStubUsingContextClassLoader="true"</pre>
CR101760	<p>停止している管理対象サーバから Web アプリケーションを削除すると、次のようなエラーになりました。</p> <pre>&lt;Warning&gt; &lt;Management&gt; &lt;149311&gt; &lt;Rejecting deployment operations to non-running server, mgd1&gt;</pre> <pre>&lt;Warning&gt; &lt;Deployer&gt; &lt;149004&gt; &lt;Failures detected initiating weblogic.management.ManagementException: Rejecting deployment operations to non-running server, mgd1 task for application Remove application MyWebApp on mgd1&gt;</pre> <p>Web アプリケーションは削除されませんでした。</p> <p>コードを変更し、WebServer および VirtualHost MBean から関連する WebAppComponentMBean 参照を削除することで問題を修正しました。</p>
CR102928	<p>VirtualHost にデプロイされた EAR 内の Web アプリケーションは、ドメインの再起動時に再デプロイされませんでした。</p> <p>仮想ホストに割り当てられたアプリケーションがドメインの再起動時に再起動されるようにコードを変更して、問題を解決しました。</p>
CR108934	<p><a href="http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA03-28.01.jsp">http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA03-28.01.jsp</a> のセキュリティ勧告に関する情報を確認してください。</p>

## ドメイン コンフィグレーション ウィザード

---

変更要求番号	説明
CR101796	WebLogic Server 7.0 SP2 では、サイレント モードで使用されるコンフィグレーション ウィザード テンプレートによって、使用できないドメインが作成されました。  silent.xml のコードを変更して問題を修正しました。

---

## EJB

---

変更要求番号	説明
CR056023	データベース内の対応する行が select for update または update 文でロックされる場合、CMP エンティティ EJB はタイムアウトしませんでした。トランザクションをロールバックする前に実行中の JDBC 文に取り消しが送信されるように、コードを変更しました。データベース内の対応する行がロックされる場合でも、トランザクションのタイムアウトが発生するときに EJB がタイムアウトするようになりました。
CR063837	存在しないデータベース テーブルおよびカラムがあるかどうかを EJB コンテナが検証しようとするときのエラーが送出されました。  weblogic.utils.AssertionError: ***** ASSERTION FAILED ***** [ Table: Cmp_Birthday Full Table Check failed, but table all columns were found! ] at weblogic.ejb20.utils.TableVerifier.verifyTableAndColumnsExist (TableVerifier.java:371) at weblogic.ejb20.utils.TableVerifier.verifyTableExistsAndCreateMaybe (TableVerifier.java:391)  この問題は、コードを修正して解決されました。
CR068572	EJB クエリ言語 (EJB-QL) は、新しい Upper および Lower 関数で、大文字 / 小文字を区別しない検索をサポートするようになりました。

---

変更要求番号	説明
CR081512 CR097906 CR094280	<p>同時方式がオプティミスティックである場合、トランザクション内で行われた更新が Finder メソッドで見つかりませんでした。つまり、include-updates 機能が失敗していました。</p> <p>オプティミスティックな同時方式で include-updates が機能するようにコードが修正されました (Oracle データベースの場合)。</p>
CR081951	<p>クライアントがステートレス セッション Bean を呼び出して、その Bean が Bean 管理の永続性エンティティ Bean を呼び出していました。そのエンティティ Bean は、TreeMap をセッション Bean に返していました。その結果、負荷のある状態で次の例外が送出されていました。</p> <pre data-bbox="360 586 1227 838">Tue Jul 16 11:46:54 PDT 2002:&lt;E&gt; &lt;Adapter&gt; Exception thrown by rmi server: [386851657836047808S172.17.24.65: [8001,8001,8002,8002,8001,-1]/ 262] java.util.ConcurrentModificationException at java.util.TreeMap\$Iterator.next(TreeMap.java:1023) at java.util.TreeMap.writeObject(TreeMap.java:1498) at java.lang.reflect.Method.invoke(Native Method) at java.io.ObjectOutputStream.invokeObjectWriter(ObjectOutputStrea m.java:1864)...</pre> <p>分析の結果、TreeMap の writeObject () に対する現在の呼び出しがオブジェクトのシリアライゼーションを完了して復帰する前にエンティティ Bean が以降の呼び出しを処理し始め、その結果として ConcurrentModificationException が送出されることが判明しました。この動作は、Bean が EJB コンテナ レイヤでロックされ、RMI レイヤでシリアライズされることに起因しています。</p> <p>この問題は、RMI アクティベーション コールバックを使用して EJB をプールに戻すことで解決しました。</p>
CR087151	<p>stateless-bean-is-clusterable が False に設定されたクラスタ内のステートレス セッション Bean が次のようなエラーを生成しました。</p> <pre data-bbox="360 1185 1227 1350">&lt;Mar 14, 2003 3:35:00 PM PST&gt; &lt;Error&gt; &lt;Cluster&gt; &lt;Conflict start: You tried to bind an object under the name ejb20-statelessSession-TraderHome_EO in the JNDI tree. The object you have bound from 172.17.24.112 is non clusterable and you have tried to bind more than once from two or more servers. Such objects can only deployed from one server.&gt;</pre> <p>クラスタ化されていないスタブで、JNDI バインディングはレプリケートされなくなりました。</p>

---

変更要求番号	説明
CR087151	<p>ステートレス セッション Bean では、weblogic-ejb-jar.xml に次の行があり、</p> <pre data-bbox="297 331 1170 378">&lt;stateless-bean-is-clusterable&gt;False&lt;/stateless-bean-is-clusterable&gt;</pre> <p>Bean がクラスタにデプロイされている場合に、次のエラーが生成されます。</p> <pre data-bbox="297 435 1170 656">&lt;Mar 14, 2003 3:35:00 PM PST&gt; &lt;Error&gt; &lt;Cluster&gt; &lt;Conflict start: You tried to bind an object under the name ejb20-statelessSession-TraderHome_EO in the JNDI tree. The object you have bound from 172.17.24.112 is non clusterable and you have tried to bind more than once from two or more servers. Such objects can only deployed from one server.&gt;</pre> <p>コードの修正により、クラスタ化されていないスタブの JNDI バインディングはレプリケートされません。</p>
CR088156、 CR124471	<p>PersistenceManagerImpl.releaseResources() メソッドが、別のスレッド (weblogic.transaction.internal.ServerTransactionImpl.commit()) でトランザクションをコミットするために接続が使用されている JDBC 文を閉じるときに、次のデッドロックが生じました。</p> <pre data-bbox="297 899 1170 1119">FOUND A JAVA LEVEL DEADLOCK: ----- "ExecuteThread: '180' for queue: 'default'": waiting to lock monitor 0xcbb28 (object 0xde1d070, a oracle.jdbc.driver.OraclePreparedStatement), which is locked by "ExecuteThread: '73' for queue: 'default'" "ExecuteThread: '73' for queue: 'default'": waiting to lock monitor 0xcbc78 (object 0xdec416b8, a oracle.jdbc.driver.OracleConnection), which is locked by "ExecuteThread: '180' for queue: 'default'"</pre> <p>調査の結果、EJB コンテナが早まって文を閉じてしまうことが判明しました。現在は、PersistenceManagerImpl.releaseResources() が適切なタイミングで文を閉じるようにコードが修正されています。</p>

---



変更要求番号	説明
CR089759	<p>UserTransaction コンテキストを使用して <code>ejbStore()</code> から <code>context.getCallerPrincipal()</code> を呼び出すと、次の例外が送出されていました。</p> <pre>&lt;Oct 24, 2002 5:52:24 AM PDT&gt; &lt;Info&gt; &lt;EJB&gt; &lt;Exception from ejbStore:javax.ejb.EJBException:.ejbStore: nulljavax.ejb.EJBException:.ejbStore: null at AccountBean.ejbStore(AccountBean.java:99)</pre> <p>この問題は、EJB 仕様のセクション 21.2.5.1 に準拠するために SP02 で導入された変更の結果です。この問題は修正されています。</p>
CR089822	<p><code>weblogic.ejbc</code> の <code>-classpath</code> オプション (コンパイル時のクラスパスの設定に使用され、システムまたはシェルのクラスパスをオーバーライドする) が機能するようになりました。以前は次のエラーで失敗していました。</p> <pre>"ERROR: Error from ejbc: Unable to load a class specified in your ejb-jar.xml: examples.ejb20.basic.statelessSession.TraderBean" "ERROR: ejbc found errors"</pre>
CR089936	<p>WebLogic Server クラスタにデプロイされた、クラスタ化されているコンテナ管理の永続性 Bean で、WebLogic Server が <code>ejbload()</code> を間違っって呼び出していました。<code>ejbLoad()</code> は、EJB が同じクラスタ内の別のサーバから修正された後に、Bean が要求されるたびに呼び出されていました。</p> <p>同時方式が <code>Optimistic</code> で、<code>cache-between-transactions</code> と <code>home-is-clusterable</code> が両方とも有効になっていました。</p> <p>この問題は、<code>ejbLoad()</code> の最初の呼び出しの後に Bean の状態が適切に更新されなかったことが原因で発生しました。</p> <p>Bean の状態が正しく更新されるようにコードが修正されました。</p>
CR090104	<p>記述子にアイソレーション レベルを指定した EJB が実行時に例外を受け取っていました。WebLogic Server は、誤ったベンダの回避策をチェックしていました。<code>supportsTxIsolationUponEnlistment</code> ではなく <code>supportsTxIsolation</code> をチェックしていました。また、トランザクションがアクティブなときにアイソレーション レベルを設定または再設定する際のルールや動作は、さまざまなデータベース ベンダやドライバによって異なります。たとえば、WebLogic Server <code>jDriver/XA for Oracle</code> を使用する場合、アイソレーション レベルはトランザクション プランチの作成時にのみ設定し、プランチを結合または再開するときには設定できません。</p> <p>コードを変更して両方の問題を修正しました。</p>

---

**変更要求番号**   **説明**

---

CR090192    max-beans-free-pool が No LIMIT に設定されているステートレス セッション Bean がデプロイメント時に applications ディレクトリにコピーされると、デプロイメントは次のエラーにより失敗しました。

```
<Nov 5, 2002 4:59:00 PM EST> <Error> <Deployer> <BEA-149201> <The Slave Deployer failed to complete the deployment task with id 7 for the application _appsdir_BasicStatelessTraderBean_jar. weblogic.management.ApplicationException: Prepare failed. Task Id = 7 { Module Name: BasicStatelessTraderBean, Error: [EJB:011025]The XML parser encountered an error in your deployment descriptor. Please ensure that your DOCTYPE is correct. You may wish to compare your deployment descriptors with the WebLogic Server examples to ensure the format is correct. The error was: Path .weblogic-ejb-jar.weblogic-enterprise-bean.caching-descriptor.max-beans-in-free-pool.: [EJB:010136]Param must be an integer.
```

問題は、「NO LIMIT」のホワイトスペースの解析にありました。

max-beans-in-free-pool が NO LIMIT として指定されている場合に、それが int の最大値に設定されるようにコードが修正されました。

---

CR090515    Administration Console の [ 待機総数 ] フィールドで、Exclusive 同時方式のエンティティ Bean およびステートフル セッション Bean へのアクセスを待っているクライアントの数が間違っ報告されていました。この数は、クライアントがロックを取得するか、タイムアウトになったときに正しくデクリメントされるようになりました。

---

CR091263    ステートレス セッション Bean が、関連 Bean を更新しようとして失敗しました。Oracle 制約違反の SQL 例外が原因でトランザクションがロールバックされた後、ステートレス セッション Bean にアクセスする試みで次のエラーが発生しました。

```
com.cpships.ecomm.exception.FatalException: EJB Exception: javax.ejb.TransactionRolledbackLocalException: Illegal Reentrant call to RelatedCompanyHomeRemote with primary key: 30: weblogic.ejb20.InternalException: Illegal Reentrant call to RelatedCompanyHomeRemote with primary key: 30
```

調査の結果、weblogic.ejb20.manager.DBManager.remove() メソッドが実行された後、setBusy フラグ (Bean が使用できるかどうか、つまり Bean がビジー状態かどうかを示す) が正しく False に設定されないことが判明しました。この問題は解決済みです。

---

---

変更要求番号	説明
CR091352	<p>4 レベルのリレーションシップ キャッシングで、ejbc のコード生成が次の例外を発生して失敗していました。</p> <pre>Found 4 semantic errors compiling "D:/labs/cluster70/src/com/bea/ps/cm_r_orders/ejbcgen/com/bea/ps/cm_r_orders/CustomerBean_1q1rpd_WebLogic_CMP_RDBMS.java": 789. if (__WL_bean_orderLines != null) &lt;-----&gt; *** Error: No entity named "__WL_bean_orderLines" was found in this environment. 791. __WL_bean_orderLines.__WL_add_WL_item_field(__WL_eo); &lt;-----&gt; *** Error: "__WL_bean_orderLines" is either a misplaced package name or a non-existent entity. 1310. if (__WL_bean_orderLines != null) { &lt;-----&gt; *** Error: No entity named "__WL_bean_orderLines" was found in this environment. 1312. __WL_bean_orderLines.__WL_add_WL_item_field(__WL_eo); &lt;-----&gt; *** Error: "__WL_bean_orderLines" is either a misplaced package name or a non-existent entity.</pre> <p>分析の結果、キャッシングが4レベル以上の場合、prevCmrFieldName 文字列を作成するときに RDBMSCodeGenerator.java によって余分な文字が削除されていることが判明しました。</p> <p>RDBMSCodeGenerator.java の修正により、この問題は解決されました。</p>
CR091436	<p>ルックアップに外部 InitialContextFactory() 呼び出しを使用するメッセージ駆動型 Bean が繰り返し接続の試行を行い、かつ JMS プロバイダがアクセス不能の場合、接続や他のリソースが急速に増えて、サーバのリソースが枯渇していました。</p> <p>より詳しい調査の結果、JMSConnectionPoller() が InitialContexts を適切に閉じていないことが判明しました。この問題は、コード上で修正されました。</p>
CR091774	<p>コンテナ管理による関係の Finder メソッドが実行されたときに、EJB が cache-between-transactions を True に設定してオプティミスティックな同時方式を使用する場合でも、WebLogic Server が同じエンティティ EJB への2回目のアクセス用に間違っデータベースから状態をロードしようとしていました。この問題は、コード上で修正されました。</p>

---

変更要求番号	説明
CR094073	BMP エンティティ Bean の処理が、EJB 2.0 仕様に準拠して変更されました。EJB の仕様に従い、EJB コンテナはファインダ メソッドの実行前に EJBStore を呼び出してエンティティ Bean の状態を同期化するようになりました。この変更は、findByPrimaryKey メソッドに影響しません。
CR094524	WebLogic Server 6.1 SP3 では、多対 1 のコンテナ管理の関係を持つ読み込み専用の EJB で LockTimedOutException が生じていました。 この問題は、if condition チェックを rdb.isReadOnly() から READONLY_EXCLUSIVE_CONCURRENCY チェックに変更して解決されました。
CR094629、 CR098386	1 対 1 のコンテナ管理の関係を持つ 2 つの Bean が、同じトランザクションで作成されました。両方の Bean で delay-database-insert-until が Commit に設定されていても、架空の SQL SELECT 文が発行されました。大量のデータが扱われる状況では、一括挿入の真のメリットは余計な SELECT の結果としてパフォーマンスが低下したせいで実現されていませんでした。 余計な SELECT が発行されないようにコードが変更されています。
CR094776	エンティティ EJB の作成が試行されたが失敗した場合に、Administration Console の [ プールの使用中 Bean 数 ] がまるで作成が成功したかのようにインクリメントされました。 この問題は修正済みです。

---

**変更要求番号**    **説明**

---

CR094861    String[] を持つ複数の属性が EJB に含まれている場合に、ejbc コンパイラで次のエラーが送出されました。

```
C:\public\383031>java weblogic.ejbc
ejb11_basic_containerManaged.jar
ejbcgen\examples\ejb11\basic\containerManaged\AccountBean_rp7qqq
__WebLogic_CMP_R DBMS.java:499: byteArray is already defined in
ejbFindByPrimaryKey(java.lang.String) byte[] byteArray =
__WL_rs.getBytes(5); ^
ejbcgen\examples\ejb11\basic\containerManaged\AccountBean_rp7qqq
__WebLogic_CMP_R DBMS.java:611: byteArray is already defined in
ejbFindAccount(double) byte[] byteArray = __WL_rs.getBytes(5); ^
ejbcgen\examples\ejb11\basic\containerManaged\AccountBean_rp7qqq
__WebLogic_CMP_R DBMS.java:732: byteArray is already defined in
ejbFindBigAccounts(double) byte[] byteArray =
__WL_rs.getBytes(5); ^
ejbcgen\examples\ejb11\basic\containerManaged\AccountBean_rp7qqq
__WebLogic_CMP_R DBMS.java:838: byteArray is already defined in
ejbFindNullAccounts() byte[] byteArray = __WL_rs.getBytes(5); ^
ejbcgen\examples\ejb11\basic\containerManaged\AccountBean_rp7qqq
__WebLogic_CMP_R DBMS.java:1236: byteArray is already defined in
ejbLoad() byte[] byteArray = __WL_rs.getBytes(5); ^ 5 errors Exec
failed .. exiting
```

この問題は、EJB 1.1 でのみ発生しました。

生成されたコードのバイト配列でユニークな変数が作成されるようにコードが変更されました。

---

CR095023    EJB デプロイメント記述子は、トークン "\${APPNAME}" を使用して、JNDI 名にバインドして参照することができます。ただし、LocalJNDI 名は、デプロイメント時に解決する "\${APPNAME}" トークンをサポートしていませんでした。現在は、デプロイメント時に、アプリケーション名トークンの置換用に LocalJNDI 名が変換されるようになりました。

---

---

**変更要求番号**   **説明**

---

CR095030      WebLogic Server 7.0 SP1 では、ステートフル セッション EJB のデプロイメントが次の `ClassCastException` で失敗しました。

```
Unable to deploy EJB: MetaDataBS from
csam_dcf3_server_ejb_depl.jar: java.lang.ClassCastException:
weblogic.ejb20.deployer.SessionBeanInfoImpl at
weblogic.ejb20.deployer.ClientDrivenBeanInfoImpl.setMethodDescri
ptors(ClientDrivenBeanInfoImpl.java:745) at
weblogic.ejb20.deployer.ClientDrivenBeanInfoImpl.prepare(Client
DrivenBeanInfoImpl.java:907) at
weblogic.ejb20.deployer.EJBDeployer.setupBeanInfo(EJBDeployer.ja
va:1041) at
weblogic.ejb20.deployer.EJBDeployer.prepare(EJBDeployer.java:124
5) at
weblogic.ejb20.deployer.EJBModule.prepare(EJBModule.java:242) at
weblogic.j2ee.J2EEApplicationContainer.prepareModule(J2EEApplica
tionContainer.java:1504) at
weblogic.j2ee.J2EEApplicationContainer.prepare(J2EEApplicationCo
ntainer.java:690) at
weblogic.j2ee.J2EEApplicationContainer.prepare(J2EEApplicationCo
ntainer.java:576) at
weblogic.management.deploy.slave.SlaveDeployer.processPrepareTas
k(SlaveDeployer.java:1064) at
weblogic.management.deploy.slave.SlaveDeployer.prepareUpdate(Sla
veDeployer.java:732) at
weblogic.drs.internal.SlaveCallbackHandler$1.execute(SlaveCallba
ckHandler.java:24) at
weblogic.kernel.ExecuteThread.execute(ExecuteThread.java:153) at
weblogic.kernel.ExecuteThread.run(ExecuteThread.java:134)
```

この問題は、`setMethodDescriptors()` がセッション Bean を間違っ  
てエンティティ Bean として識別していたことが原因で発生しました。

この問題は修正されています。

---

CR095090      同じトランザクション内で、`delay-database-until-insert` が Commit に設  
定され、`ejbRemove()` が呼び出された場合に、Bean の削除は  
`java.rmi.NoSuchObjectException` によって失敗します。

`delay-database-insert-until` が Commit に設定されている場合はコンテナ  
キャッシュをチェックして Bean を削除するようにコードが変更されました。

---

---

変更要求番号	説明
CR095141	<p>クラスタ化されたコンフィグレーションと最近未使用 (NRU) キャッシング方式で、インスタンス化された Bean インスタンスの数が <code>max-beans-in-cache</code> を超えたときに、コンテナはステートフル セッション Bean のパッシベーションを正しく行いませんでした。</p> <p>クラスタ化された環境で、WebLogic Server は <code>ReplicatedStatefulSessionManager</code> を使用して新しい <code>NRUCache</code> を作成します。</p> <p>調査の結果、レプリケートされたステートフル <code>SessionManager</code> がキャッシュに登録されないために、Bean はアクティブ キューから非アクティブ キュー、さらに空きキューへと正しく移動されないため、<code>CacheFullExceptions</code> になることが判明しました。</p> <p>コードの修正で、この問題は解決されました。</p>
CR095173	<p><code>idle-timeout-seconds</code> 要素は、ステートフル セッション Bean でパッシベーションが行われるまで、つまりキャッシュから削除されてディスクに書き込まれるまで EJB コンテナが待機する時間を指定します。EJB コンテナでは、パッシベーションされた EJB をディスクから削除するまでの待機時間を指定するためにもこの要素を使用していました。ただし、<code>idle-timeout-seconds</code> より長い時間ステートフルセッション Bean がディスクに残ることを望むユーザもいました。ステートフル セッション Bean がキャッシュに留まる時間と、それらがディスクに留まる時間を別々の要素で指定できることが望まれていたのです。</p> <p>WebLogic Server 7.0 サービス パック 3 では、新しい要素の <code>session-timeout-seconds</code> が導入されます。この要素では、アイドル状態のステートフル セッション Bean がディスクから削除されるまでの EJB コンテナの待機時間を指定します。</p>
CR095290	<p><code>delay-database-insert-until</code> が <code>Commit</code> に設定され、同時方式がオプティミスティックであるエンティティ EJB をトランザクションの中で更新しようとするアプリケーションの試みは、<code>NullPointerException</code> で失敗しました。</p> <p>問題は、<code>delay-database-insert-until</code> の値が <code>Commit</code> の場合、WebLogic Server は EJB が作成された時点でオプティミスティック同時方式のバージョン カラムを初期化しないということです。</p> <p>オプティミスティック同時方式のバージョン カラムが一括挿入用に初期化されるようコードが変更されました。</p>

---

---

**変更要求番号**   **説明**

---

CR095545      ejb-name が同じでも、JNDI マッピングは異なる 2 つのメッセージ駆動型 Bean を同じ WebLogic Server インスタンスにデプロイしようとしたときに、最初の Bean は正常にデプロイされましたが、2 番目の Bean のデプロイメントは次の例外で失敗しました。

```
weblogic.management.ManagementException: - with nested exception:
[javax.management.InstanceAlreadyExistsException:
jpdomain:Location=jpserver,Name=MessageQueueHandlerBean,ServerRun
time=jpserver,Type=EJBMessageDrivenRuntime] at
weblogic.management.runtime.RuntimeMBeanDelegate.register(Runtime
MBeanDelegate.java:96) at
weblogic.management.runtime.RuntimeMBeanDelegate.<init>(RuntimeM
BeanDelegate.java:83) at
weblogic.management.runtime.RuntimeMBeanDelegate.<init>(RuntimeM
BeanDelegate.java:53) at
weblogic.management.runtime.RuntimeMBeanDelegate.<init>(RuntimeM
BeanDelegate.java:63) at
weblogic.ejb20.deployer.MessageDrivenRuntimeMBean.<init>(Message
DrivenRuntimeMBean.java:18) at
weblogic.ejb20.deployer.MessageDrivenBeanInfoImpl.deploy(Message
DrivenBeanInfoImpl.java:450) at
weblogic.ejb20.deployer.Deployer.deployDescriptor(Deployer.java:
1299) at
weblogic.ejb20.deployer.Deployer.deploy(Deployer.java:1005)
```

この問題の原因は、MessageDrivenRuntimeMBean の名前がユニークでないことにありました。現在は、名前がユニークになるようにコードが変更されています。

---

CR095632      基本的な Java クライアントは、ステートレス セッション Bean を Facade として使用して、エンティティ EJB を作成し、ファインダ メソッドと読み込み専用 Bean を使用してその EJB を読み込み、エンティティ Bean を更新して、再びエンティティ EJB を読み込みました。

transaction-attribute が Supports に設定されている場合、クライアントは更新された値を読み込みました。transaction-attribute が Required または RequiresNew に設定されている場合は、クライアントは古い値を読み込みました。

調査の結果、キャッシュのタイムアウトが 0 に設定されているために、ファインダメソッドがタイムアウトになった Bean を更新しなかったことが判明しました。キャッシュのタイムアウトが 0 でも Bean が更新されるようにコードが変更されました。

---



変更要求番号	説明
CR096182	動的な EJB-QL は基底の SQL に <code>MAX_INT</code> として誤って長い引数を渡し、検索が失敗しました。コードの修正で、この問題は解決されました。
CR096395	エンティティ Bean がアンデプロイされる時に、コンテナは <code>unsetEntityContext</code> の呼び出しに失敗しました。エンティティ Bean がアンデプロイされる時にプールをクリーンアップするために、 <code>BaseEntityManager</code> に <code>undeploy</code> メソッドが追加されました。その修正で、問題は解決されました。
CR096800	<p>WebLogic Server 6.1 が WebLogic Server 7.0 の呼び出し側に対して EJB 例外を送出すると、呼び出し側ではエラー メッセージを伴うシリアライゼーション エラーを代わりに受け取ります。</p> <p>これは、WebLogic Server 6.1 で <code>EJBException</code> のシリアルバージョン ユニーク識別子 (SVUID) が間違っていたためです。</p> <p>WebLogic Server 7.0 は現在 WebLogic Server 6.1 と通信するときに正しい SVUID を識別して受け付け、その接続上で送信します。7.0 に取り入れられた以前の解決策には不備がありました。</p>
CR096848	<p>6.1 SP04 では、CMP 2.0 Bean で <code>&lt;is-modified-method-name&gt;</code> が呼び出されていませんでした。Bean のメソッドを呼び出すたびに <code>ejbStore()</code> を呼び出し、後でストアを回避するかどうかを判定していました。そのため、<code>&lt;is-modified-method-name&gt;</code> を頻繁に使用するアプリケーションでは、パフォーマンスの問題が発生していました。</p> <p>この問題は、CMP EJB 2.0 用に <code>&lt;is-modified-method-name&gt;</code> 関数を実装することで解決されました。</p>
CR097890	<p>WebLogic Server は、同じ時間に <code>JMSPoller</code> の複数のトリガをスケジューリングしていました。それらすべてがデフォルトの JMS キューを占有して、デッドロックが起きていました。</p> <p>トリガが重複してスケジューリングされないようにコードが変更されました。</p>

変更要求番号	説明
CR097913	<p>動的な EJB-QL 要求で COUNT と EXISTS を結合すると、SQL が無効になり、次の例外が送出されました。</p> <pre data-bbox="297 326 1130 482">[java] javax.ejb.FinderException: Exception in dynamicQuery while preparing or executing statement: 'weblogic.jdbc.rmi.SerialStatement@62b30' [java] java.sql.SQLException: ORA-00937: not a single-group group function [java] [java] java.sql.SQLException: ORA-00937: not a single-group group function</pre> <p>サブクエリの解析時に main select list に不要なカラムが追加されていました。パーサのコードを修正して問題は解決されました。</p>
CR098188	<p>メッセージ駆動型 Bean が、別のサーバ上の JMS キューからメッセージを消費していました。その JMS サーバは機能していませんでした。</p> <p>weblogic.ejb20.internal.MDListener ではなく、UserTransaction.commit method から例外が送出されました。MDListener は、動作を継続していました。それ以降の JMS からの MDB の呼び出しは、すでにスレッドにトランザクションがあるために UserTransaction.begin で失敗しました。</p> <p>MDListener が新しいトランザクションを開始する前に現在のトランザクションをサスペンドするようコードが変更されています。</p>
CR098923	<p>JAR のすべてのメッセージ駆動型 EJB で実行されるように mdbPoolInfo.start() メソッドが変更されました。その JAR の最初の MDB で失敗しても、すべての MDB で実行されます。</p>
CR099420	<p>2 次元配列が引数として指定された場合、ejbc コンパイラは次のエラーで失敗しました。</p> <pre data-bbox="297 1090 1143 1220">ERROR: Error from ejbc: Unable to set the method permission for method "isAuthorized(java.lang.String, [[java.lang.String)". No matching method could be found. Please verify the method signature specified in the ejb-jar.xml file matches that of your EJB. ERROR: ejbc found errors.</pre> <p>weblogic.ejb20.deployer.mbimpl.MethodDescriptorImpl クラスの makeMethodParameter() メソッドは、渡される配列の次元に応じて適切なメソッドパラメータシグネチャを生成するように修正されました。</p>

---

変更要求番号	説明
CR100822	<p>クラスタ化されたメッセージ駆動型 Bean の動作を変更して、パフォーマンスを最適化しました。以前は、MDB が JMS に接続する場合、JMS 接続がクラスタ全体でバランシングされるように、標準のクラスタ ロードバランシング アルゴリズムが使用されました。</p> <p>可能な場合、MDB はメッセージをホストする (接続の受信元である) 同じ JVM に、JMS 接続を導きます。これにより、リソースにアクセスするときに MDB によって作成されるホップ数が削減されます。</p>
CR100832	<p>オブティミスティックな同時方式が有効な場合、暗黙の書き込みがチェックされないため、カラムのバージョン番号は正しく更新されませんでした。オブティミスティックな同時方式で暗黙の書き込みがチェックされるようにコードを変更しました。</p>
CR101226	<p>WebLogic Server 7.0 SP1 で、自動テーブル作成が断続的に次のエラーにより失敗しました。</p> <pre data-bbox="360 725 1209 1255">weblogic.utils.AssertionError: ***** ASSERTION FAILED ***** [ Table: Cmp_Birthday Full Table Check failed, but table all columns were found! ] at weblogic.ejb20.utils.TableVerifier.verifyTableAndColumnsExist(Ta bleVe rifier.java:371) at weblogic.ejb20.utils.TableVerifier.verifyTableExistsAndCreateMay be(Ta bleVerifier.java:391) at weblogic.ejb20.cmp.rdbms.RDBMSPersistenceManager.populateFieldSQ LType Map(RDBMSPersistenceManager.java:499) at weblogic.ejb20.cmp.rdbms.RDBMSPersistenceManager.setup(RDBMSPers isten ceManager.java:122) at weblogic.ejb20.manager.BaseEntityManager.setupPM(BaseEntityManag er.ja va:192) at weblogic.ejb20.manager.BaseEntityManager.setup(BaseEntityManager .java :168) at weblogic.ejb20.manager.DBManager.setup(DBManager.java:123) at weblogic.ejb20.deployer.ClientDrivenBeanInfoImpl.deploy(ClientDr ivenB eanInfoImpl.java:807) at weblogic.ejb20.deployer.Deployer.deployDescriptor(Deployer.java: 1234)</pre> <p>調査の結果、クエリの発行に <code>preparedStatement</code> が使用されていたことがわかりました。<code>prepared statement</code> は Oracle データベースから削除できるため、断続的なアサーション エラーが発生する可能性があります。</p> <p><code>preparedStatement</code> の代わりに <code>Statement</code> を使用するようにコードを修正しました。</p>

---

変更要求番号	説明
CR101248	<p>CMP Bean が JDBC 接続プールをリークしていました。 <code>ejbFindByPrimaryKey()</code> が <code>releaseResources()</code> を呼び出すときに問題が発生しました。</p> <p><code>ResumeTransaction</code> が失敗した場合、WebLogic Server がリソースを解放して JDBC 接続リークの可能性を回避するように、<code>RDBMSCodeGenerator</code> を変更しました。</p>
CR102180	<p>100 個の EJB を含む JAR ファイルで <code>ejbc</code> コンパイラが失敗しました。 オペレーティングシステムに関係なく渡されるファイル数に基づいた <code>@tempfile</code> 機能を使用することで、問題を修正しました。</p>
CR102481	<p>一部の Bean 管理のステートフル セッション Bean では、キャッシュが一杯でアクティベーションとパッシベーションが発生するときに、Bean インスタンスの数が <code>max-beans-in-cache</code> よりも大きい場合に、Bean が境界を定めるトランザクションで <code>IllegalStateException</code> が発生しました。</p> <p>コードの変更により、パッシベーション リクエストごとに 1 つのリプレーサが作成され、そのため各スレッドがそれ専用のリプレーサを持つこととなります。</p>
CR104046	<p><code>delay-database-insert-until</code> が <code>Commit</code> に設定されている場合に、「複数テーブル」の Bean を作成するときにオプティミスティックな同時方式のバージョン カラムが初期化されませんでした。</p> <p><code>RDBMSCodeGenerator.perhapsAssignOptimisticField()</code> から、CMP Bean がマップされたすべてのテーブルのバージョン カラムを設定する生成済みコードが返されるように、コードを変更しました。</p>
CR105857	<p>Bean 管理の永続性を持ち、同時方式が <code>exclusive</code> に設定されている EJB が、キャッシュ内の Bean を検索するのではなく、<code>ejbFindByPrimaryKey</code> を呼び出しました。</p> <p>EJB がキャッシュ内の Bean を検索するようにコードを変更して、問題を解決しました。</p>

## 相互運用性

---

変更要求番号	説明
--------	----

---

CR095804	WebLogic Server 7.0 SP2 クライアントが WebLogic 8.1 サーバ インスタンスから Coordinator リモート オブジェクトを取得しようとする、リモート スタブを CoordinatorOneway インタフェースにキャストするときに <code>ClassCastException</code> が送出されていました。このため、8.1 調整サーバへコミットを渡すことができませんでした。 クラスロード機能を修正するコードの変更で問題は解決されました。
----------	---

---

## JDBC

---

変更要求番号	説明
--------	----

---

CR087870	WebLogic Server 7.0 の以前のリリースでは、高可用性アルゴリズムの JDBC マルチプールは最初の接続プールの DBMS がネットワークから削除された場合はフェイルオーバーされませんでした。この問題は、コードの修正で解決されました。
CR088525	以前の WebLogic Server 7.0 サービス パックで、値が「/////」の JSP パラメータが、 <code>getParameter()</code> によって値が「/」であると解釈されました。 この問題は、コードを修正して解決されました。
CR090379	WebLogic Server 7.0 の以前のリリースでは、JDBC 接続プールを通じて 1 つの JDBC オブジェクトを指し示す複数の <code>ResultSet</code> を誤って作成すると <code>OutOfMemoryError</code> が発生しました。各 <code>ResultSet</code> からは、RMI オブジェクトが生じました。 <code>ResultSet</code> を閉じても、すべての RMI オブジェクトが閉じたわけではありません。この問題は、コードの修正で解決されました。

---

---

変更要求番号	説明
CR092453	<p>WebLogic Server 6.1 SP04 では、Oracle 9.2 に付属する CLASSES12.zip の Oracle Thin XA により、EJB を呼び出すステートレス セッション Bean において、「Configuration Changes saved to the repository」というメッセージの後で XAER_PROTO が発生していました。</p> <pre>START SLEEP 2: After updating thevalue to 1... DONE SLEEP 2: After updating thevalue to 1... START SLEEP 2: After updating thevalue to 2... DONE SLEEP 2: After updating thevalue to 2... START SLEEP 2: After updating thevalue to 3... DONE SLEEP 2: After updating thevalue to 3... START SLEEP 2: After updating thevalue to 4... DONE SLEEP 2: After updating thevalue to 4... START SLEEP 2: After updating thevalue to 5... DONE SLEEP 2: After updating thevalue to 5... Current value is 5 &lt; Dec 6, 2002 10:26:59 PM MST&gt; &lt;Info&gt; &lt;Management&gt; &lt;Configuration changes for domain saved to the repository.&gt; SQLException -- XA error: XAER_PROTO : Routine was invoked in an improper context start() failed on resource 'OracleXA' null Current value is 0</pre> <p>この問題は、Oracle クライアント 9.2.0.[01] の確認済みの問題が原因で発生しており、9.2.0.2 で解決されます。WebLogic Server では、9.2.0.[01] の問題を回避するためにコード修正が行われました。</p>
CR092791	<p>WebLogic Server 6.1 で、Oracle Thin Client ドライバに基づく接続プールを介して、特定の Oracle オブジェクト (Array、Struct、その他) を使用できませんでした。Oracle Thin Client ドライバによって返されるオブジェクトがシリアライズ可能でもリモートでもないため、RMI を通じて渡すことができません。</p> <p>WebLogic Server 7.0 SP3 では、新しいパッケージ <code>weblogic.jdbc.vendor.oracle</code> にこれらのオブジェクトのプロキシが含まれており、接続プールを介して渡されるようになりました。</p>

---

変更要求番号	説明
CR093038	<p>WebLogic Server 7.0 SP3 では、Oracle 仮想プライベート データベース (Virtual Private Database : VPD) がサポートされています。VPD を使用することで、アプリケーション定義のファイングレイン アクセス コントロールをサーバで実施し、Oracle 9i データベース サーバ内のアプリケーション コンテキストのセキュリティを確保できます。</p> <p>詳細については、『WebLogic JDBC プログラマーズ ガイド』の「Oracle 仮想プライベート データベースによるプログラミング」を参照してください。</p>
CR093649	<p><code>ejb2jsp</code> コーティリティは、String 以外のパラメータ タイプを含むメソッドに対して間違ったタグ クラスを生成していました。</p> <p>タグ ライブラリ ジェネレータが <code>getAttributeString</code> ではなく <code>getAttribute</code> を使用するように変更して、問題を修正しました。</p>
CR094645	<p>0 以外の開始インデックスを <code>getStatementProfiles()</code> メソッドに渡すと、トレースは、指定した開始インデックスではなく、.tsf ファイルの先頭からデータを返しました。</p> <p>この問題は、コードの修正で解決されました。</p>
CR094729	<p><code>JDBCConnectionPoolRuntimeMBean.getStatementProfiles()</code> メソッドは、それが呼び出された MBean インスタンスの <code>poolName</code> に基づいて結果をフィルタ処理しませんでした。<code>getStatementProfiles</code> メソッドを呼び出した場合、その結果にはトレースがアクティブ化されているすべての接続プールの文が含まれていました。<code>JDBCConnectionPoolRuntimeMBean</code> インスタンスは 1 つの接続プールに専用のものなので、その結果は誤ったものでした。</p> <p>この問題は、MBean インスタンスの <code>poolName</code> に基づいて結果を適切にフィルタ処理するように</p> <p><code>JDBCConnectionPoolRuntimeMBean.getStatementProfiles()</code> メソッドを修正して解決されました。</p>
CR095059	<p><code>JDBCStatementProfile.getTimeTaken()</code> は、JDBC の操作が実行に要した実際の時間を返す代わりに常にゼロを返したので、その操作の開始時刻と終了時刻が常に等しくなっていました。</p> <p><code>weblogic/jdbc/common/internal/ProfileStorage.java</code> のコード修正で、この問題は解決されました。</p>

変更要求番号	説明
CR095962	JDBC 接続を置き換えようとするスレッドが <code>ResourceAllocator</code> の <code>resetThisOne()</code> メソッドで不必要に同期がとられ、リソースが浪費されていました。スレッドが平行して接続を置き換えるように <code>ResourceAllocator</code> のコードが変更されました。
CR095994	以前のバージョンの WebLogic Server は、接続リーク プロファイルがなく、 <code>weblogic.jdbc.rmi.internal.ConnectionImpl</code> に接続リーク検出口ジックがありませんでした。 <code>SerialConnection.java</code> および <code>ConnectionImpl.java</code> のコードを変更して、この問題を解決しました。
CR096268	DBMS が利用できない場合に、 <code>makeConnection()</code> in <code>ConnectionEnvFactory.java</code> がクライアントの代わりに 2 回 DBMS に接続しようとしていました。 この問題は、接続が 1 回のみ試行されるように修正されています。
CR096922	負荷のある状態で、WebLogic Server が <code>ResourceAllocator.markBorrowed()</code> を呼び出し、JMX が <code>ConnectionPoolRuntimeMBeanImpl.getConnectionDelayTime()</code> を呼び出すと、デッドロック状態が発生しました。 <code>ResourceAllocator.java</code> のコード修正で、この問題は解決されました。
CR097832	Weblogic Server 6.1 SP04 シン ドライバ、Solaris 8、JDK 1.3.1_04、および Oracle 8.1.7.4. というコンフィグレーションでは、接続の更新が有効になっていると、デッドロックが発生しました。スタック トレースは次のようなものでした。 "ExecuteThread: '35' for queue: 'default'" daemon prio=5 tid=0x45ac68 nid=0x30 waiting for monitor entry [0xce901000..0xce9019d8] at <code>weblogic.jdbc.common.internal.ConnectionEnv.destroy(ConnectionEnv.java:571)</code> at <code>weblogic.common.internal.ResourceAllocator</code> <code>weblogic.jdbc.common.internal.ConnectionEnv.destroy</code> と <code>weblogic.common.internal.ResourceAllocator.release</code> は両方とも同期メソッドです。 この問題は、 <code>weblogic/jdbc/common/internal/ConnectionEnv.java</code> のコード修正で解決されました。



---

変更要求番号	説明
CR099363	<p>WebLogic Server 6.1 SP04 では、負荷テストにおいて、接続プールの更新間隔を 15 分に設定し、<code>TestConnsOnReserve</code> と <code>TestConnsOnRelease</code> を <code>false</code> に設定すると、次の例外が送出されました。</p> <pre data-bbox="364 366 1233 418">weblogic.common.ResourceException: No available connections in pool ODSConnectionPool</pre> <p>この問題は、プール内の接続の一部だけが使用されていると、その他の接続はすべて、更新テスト間隔が経過した時点でのテスト用に予約されてしまうために発生しました。このような状態で、クライアントが接続を要求すると、例外が送出されました。</p> <p>次の 2 つの動作を実装するようにコードを修正して、この問題は解決されました。</p> <p>他に何も変更せず、まったく同じ条件で使用すると、更新テストは一度にただ 1 つの接続を予約してテストし、直ちに接続を解放します。これにより、すべての接続がロックされる問題は解決されます。</p> <p>プールの定義に対してドライバ プロパティ <code>secondsToTrustAnIdlePoolConnection</code> を追加し、このプロパティに正の整数を設定した場合には、プールは、その更新間隔中に DBMS に対して正常に接続されたことがわかっているプール接続については、テストを行いません。これで <code>refresh</code> と <code>testConnsOnReserve</code> が高速化されます。</p>
CR099507	<p><code>Prepared Statement</code> を閉じるときに、<code>ConnectionEnv.java</code> の <code>cleanUpStatementForReUse()</code> メソッドが、MS SQL ドライバで実装されていない <code>clearBatch()</code> を呼び出していました。その結果として、次の例外が送出されました。</p> <pre data-bbox="364 1065 1233 1117">java.sql.SQLException: This JDBC 2.0 method is not implemented exception.</pre> <p>害のない呼び出しが許可され、例外が送出されないようにコードが変更されました。</p>
CR101419	<p><code>build-jdbc.xml</code> で、<code>RmiDataSource</code> がクラスタ対応として作成されませんでした。そのため、<code>DataSource</code> のフェイルオーバーが妨げられていました。</p> <p><code>DataSources</code> および <code>TXDataSources</code> をクラスタ対応として作成し、問題を解決しました。</p>

---

---

**変更要求番号**   **説明**

---

CR102698      weblogic.Admin CREATE\_POOL で接続プールを作成しようとする、次の例外が発生しました。

```
./wlg-create-pool.sh No permission to create ConnectionPool
Start server side stack trace:
weblogic.common.ResourceException: No permission to create
ConnectionPool at
weblogic.jdbc.common.internal.JDBCService.createPool(Ljava.util
.Properties;Lweblogic.secur
ity.acl.internal.AuthenticatedSubject;)V(Unknown Source) at
weblogic.jdbc.common.internal.ConnectionPool.createPool(Ljava.u
til.Properties;Lweblogic.se
curity.acl.internal.AuthenticatedSubject;)V(Unknown Source) at
weblogic.jdbc.common.internal.ConnectionPool.createPool(Ljava.u
til.Properties;)V(Unknown S
ource) at
weblogic.jdbc.common.internal.ConnectionPool_WebLogic
Serverkel.invoke(ILweblogic.rmi.spi.InboundReque
st;Lweblogic.rmi.spi.OutboundResponse;Ljava.lang.Object;)Lweblo
gic.rmi.spi.OutboundResponse;(Unknown Source) at
weblogic.rmi.internal.BasicServerRef.invoke(Lweblogic.rmi.inter
nal.MethodDescriptor;Lweblo
gic.rmi.spi.InboundRequest;Lweblogic.rmi.spi.OutboundResponse;)
V(Unknown Source) at
weblogic.rmi.internal.BasicServerRef$1.run()Ljava.lang.Object;(
Unknown Source) at
weblogic.security.service.SecurityServiceManager.runAs(Lweblogi
c.security.acl.internal.Aut
henticatedSubject;Lweblogic.security.acl.internal.Authenticated
Subject;Ljava.security.PrivilegedExcep
tionAction;)Ljava.lang.Object;(Unknown Source) at
weblogic.rmi.internal.BasicServerRef.handleRequest(Lweblogic.rm
i.spi.InboundRequest;)V(Unk
nown Source) at
weblogic.rmi.internal.BasicExecuteRequest.execute(Lweblogic.ker
nel.ExecuteThread;)V(Unknown Source) at
weblogic.kernel.ExecuteThread.execute(Lweblogic.kernel.ExecuteR
equest;)V(Unknown Source) at
weblogic.kernel.ExecuteThread.run()V(Unknown Source) at
java.lang.Thread.startThreadFromVM(Ljava.lang.Thread;)V(Unknown
Source)
```

調査の結果、WebLogic Server が属性として非推奨の `aclName` を要求していたことがわかりました。コードの修正で、この問題は解決されました。

---

変更要求番号	説明
CR102740	JDBC 接続プールの <code>xAPreparedStatementCacheSize</code> が Administration Console で公開されました。
CR102792	SP2 の変更では、 <code>isolation-level</code> の設定を許可しないドライバでアイソレーションレベルが許可されていました。DB2 JDBC ドライバではアイソレーションレベルの変更が許可されているため、チェックが呼び出されず、アイソレーションレベルの変更は不適切な時点で行われます。 つまり、デプロイメント記述子で <code>isolation-level</code> を設定している EJB は、デプロイメント記述子のアイソレーションレベルがデフォルトの「READ-COMMITTED」と同じ場合でも、例外を送出することがあります。 アイソレーションレベルが現在のレベルと同じ場合は設定が行われないようにコードを変更して、この問題を修正しました。
CR103256	コードを変更した結果、 <code>JDBCSessionContext</code> および <code>JDBCSessionData</code> の <code>bubblecaches</code> に関する JDBC のパフォーマンスが向上しました。
CR103321	<code>JDBCConnectionPoolRuntimeMBean</code> の <code>getConnectionsTotalCount</code> メソッドは予期されるとおりに機能しませんでした。この <code>JDBCConnectionPoolRuntimeMBean</code> でプールが初期化されてからの JDBC 接続の総数を返すものですが、代わりに、任意の時点の接続の最大数 ( <code>MaxCapacity</code> 以下) を返していました。 <code>ResourceAllocator.java</code> の <code>getTotalConnections()</code> のコードを変更して問題を修正しました。
CR103560	SQL UPDATE の <code>PreparedStatement</code> は、同じ接続上で <code>CallableStatement</code> を使用した後に機能しませんでした。データベースで何も変更されなかったときに <code>getUpdateCount</code> はゼロを返しました。これは、JDBC 接続の <code>weblogic.oci.min_bind_size</code> プロパティを設定するときに発生しました。コードを変更してこの問題を解決しました。
CR104523	<code>DatabaseMetaData.getDriverVersion()</code> は無効なバージョン文字列を返していました。コードの修正で、この問題は解決されました。

---

変更要求番号	説明
CR106522	WebLogic Server jDriver for MS SQLServer は Microsoft JDBC ドライバの <code>jdbc:microsoft:...</code> URL を誤って受け付ける可能性があります。JVM が両方のドライバを使用する場合、最初に WebLogic Server ドライバをロードすると、Microsoft ドライバが使用できなくなります。 この jDriver は非推奨になりました。
CR106767	アプリケーションが複数回 JDBC オブジェクトをクローズしようとする、WebLogic Server は例外を送出するようになりました。

---

## jDriver

---

変更要求番号	説明
CR088387	<p>jDriver 上で XADataSource を使用すると、ヒープ サイズが減少して <code>OutOfMemoryError</code> が発生しました。</p> <p><code>weblogic.jdbc.oci.Connection</code> クラスでは、トランザクションの結果セットの LOB フィールドは、<code>Connection.addLob()</code> メソッドで接続に登録されます。登録された LOB は、以下のいずれかの条件が発生すると、ネイティブ jDriver ライブラリ内の対応するオブジェクトと共に解放されます。</p> <ul style="list-style-type: none"><li>• 接続レベル (この Connection は <code>weblogic.jdbc.oci.Connection</code>) で <code>Connection.commit()/Connection.rollback()/Connection.close()</code> が呼び出される。</li><li>• SQL 文が実行されて、接続が <code>autoCommit</code> モードに設定される (つまり、非 TX データソースまたはプールからの直接接続)。</li><li>• OCI SQL 呼び出しからの戻りコードが、データベース レベルでコミット/ロールバックが発生したことを示している。</li></ul> <p>XADataSource が使用されているときは、上記の条件はどれも当てはまりません。結果として、結果セット内の jDriver の LOB フィールドは、JVM ヒープ内では解放されませんでした。</p> <p><code>closeLob ()</code> を呼び出すよう <code>weblogic.jdbc.oci.xa.DataSrcThreadInfo.java</code> のコードを修正することで、この問題を解決しました。</p>

---

変更要求番号	説明
CR090025	<p>WebLogic Server 6.1 SP04 では、jDriver for Oracle 9.2 が AL32UTF8 文字セット (Unicode バージョン 3.1) をサポートしていません。NLS_LANG を AMERICAN_AMERICA.AL32UTF8 に設定すると、 weblogic.jdbc.oci.Connection.&lt;init&gt;(Connection.java:246) で次のエラーが発生します。</p> <pre>java.sql.SQLException: Unsupported Oracle codeset: al32utf8. Set weblogic.codeset in your connection Properties to a valid JDK codeset which is compatible with the Oracle codeset defined in your NLS_LANG setting.</pre> <p>NLS_LANG=AMERICAN_AMERICA.UTF8 のときは、「ORA-01461: can bind a LONG value only for insert into a LONG column」というエラーが発生します。これは、クライアントとデータベースの文字セットが一致していないことを示しています。</p> <p>weblogic/db/oci/OciConnection のコードを修正することで、この問題は解決しました。</p>
CR090761	<p>WebLogic Server 7.0 の以前のリリースでは、WebLogic jDriver for Oracle で StringIndexOutOfBoundsException が発生しました。この問題は、ResultSet が中サイズ (数百行) の複数のクエリでときどき発生しました。この問題は、コードの修正で解決されました。</p>
CR091151	<p>WebLogic Server 6.1 SP03 および SP04 では、jDriver for Oracle の ResultSet.getBigDecimal() メソッドが正しい値を返しませんでした。たとえば、9999999999.999999 という値は 9999999999.999998 に丸められていました。</p> <p>この問題は、コードを修正して解決されました。</p>
CR096730	<p>WebLogic Server 7.0、7.0 SP1、および 7.0 SP2 に付属の WebLogic jDriver for MS SQL Server は、MS SQL Server 2000 SP3 データベースへの接続で使用されるときに次のエラーを送出しました。</p> <pre>java.sql.SQLException: I/O exception while talking to the server, java.io.EOFException Unable to connect, please check your server's version and availability.</pre> <p>この問題は、コードを修正して解決されました。</p>
CR098071	<p>WebLogic Server の以前のリリースにバンドルされていた Oracle Thin ドライバ 9.2.0 には、ときどきデータ エラーを引き起こすエラーがありました。現在は、ドライバが Oracle 提供の新しいバージョンに変わっています。</p>

# JMS

変更要求番号	説明
CR101298	<p>長期間継続する JMS 接続で、一定間隔のハートビートチェックがありませんでした。</p> <p>JMS がアイドル状態の場合は、接続を有効に保つために、接続が 5 分ごとにデータベースを ping するようにコードを変更しました。</p>
CR080289	<p>WebLogic Server 7.0 では、UserPassword が null の場合にメッセージングブリッジが正しく機能しませんでした。メッセージングブリッジが null 以外の UserName と null の UserPassword でコンフィグレーションされている場合、明確なエラーメッセージなく起動に失敗しました。</p> <p>この問題は、コードを修正して解決されました。</p>
CR083933	<p>WebLogic Server 6.1 SP03 では、負荷の大きい状況下で JMS サーバが <code>NullPointerException</code> を送出していました。</p> <pre>&lt;Warning&gt; &lt;JTA&gt; &lt;XA resource [JMS_JMSServer1JDBCStore] has not responded in the last 120 second(s).&gt; &lt;Warning&gt; &lt;JTA&gt; &lt;Resource JMS_JMSServer1JDBCStore was not assigned to any of these servers JMSServer1 &gt;</pre> <p>この問題は、実行スレッド 400、JMS スレッド 200、接続プールの接続 100 というコンフィグレーションで確認されました。例外は、負荷テストで、恒久サブスクライバに対する毎秒約 10 個のメッセージ (約 2K) を JMS サーバに対して発行すると発生しました。分散トランザクションはなく、JMS JDBC ストアに対する接続プールには Sybase ドライバが使用されていました。</p> <p>この問題は、コードを修正して解決されました。</p>

---

**変更要求番号 説明**

---

- CR089583      WebLogic Server 6.1 SP02 では、トランザクション恒久サブスクリバのサブスクリブを取り消そうとすると `JMSEException` が発生しました。
- ```
weblogic.jms.common.JMSEException: Subscription clientID.vIJAY in use, uncommitted/unacknowledged messages at weblogic.jms.backend.BEConsumer.delete(BEConsumer.java:1784) at weblogic.jms.backend.BEManager.removeSubscription(BEManager.java:204)
```
- 分析の結果、確認応答されていないメッセージのカウントをメッセージごとに 2 回インクリメントし、1 回だけデクリメントしていることが判明しました。そのため、確認応答されていないメッセージのカウントは常に 0 より大きくなり、恒久サブスクリバのサブスクリブを取り消そうとすると例外が送出されました。 >
- この問題は、確認応答されていないメッセージのカウントと懸案の恒久サブスクリバに対するコードの修正で解決されました。
- 
- CR089682      WebLogic Server 6.1 SP03 では、メッセージングブリッジの接続再試行の値（下限、上限、および増分）が正しく機能しませんでした。この問題は、メッセージングブリッジが 2 つの WebLogic JMS サーバ間でメッセージを送信するようにコンフィギュレーションされており、JMS サーバが両方とも同じ WebLogic Server インスタンスで動作している場合に発生しました。JMS サーバの 1 つを対象から外すと、次のエラーが発生しました。
- ```
<Error> <Connector> <Error granting connection request.>
```
- 再試行の試みは、接続再試行パラメータによって指定された間隔で行う必要があります。ただし、起動時のデフォルト値（下限 = 15、上限 = 60、増分 = 5）では、次のような動作になりました。
1. 接続を試みて失敗する - OK
  2. 15 秒後に 2 回目の接続を試みる - OK
  3. それ以降は、5 秒間隔で接続を試みる - NOT OK
- ```
<Error> <Connector> <Error granting connection request.>
<Error> <Connector> <Error granting connection request.>
<Error> <Connector> <Error granting connection request.>
<Error> <Connector> <Error granting connection request.>
<Error> <Connector> <Error granting connection request.>
```
- デフォルトパラメータのいずれかを変更すると、増分は正しくなりました。しかし、上限値には達しませんでした。デフォルト値の場合、55 までしか達しませんでした。
- コードの修正で、再試行値のロジックが修正されました。
-

| 変更要求番号   | 説明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CR090403 | <p>WebLogic Server 7.0 SP02 では、JTA トランザクション内で JMS キューに複数のメッセージを入れると、その後トランザクションがロールバックされた場合に、Administration Console で不正確な情報が表示されました。</p> <p>コードの修正で、トランザクションのロールバックでの <code>receivedCounts</code> が調整されました。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| CR091195 | <p>WebLogic Server 6.1 SP04 においてのみ、MDB が受信した永続キュー メッセージに対して適切に確認応答が行われず、Administration Console のモニタ機能で「BYTES PENDING」と表示されました。MDB が受信したメッセージは、サーバをシャットダウンして再起動すると再配信されました。</p> <p>EJB のトランザクション記述子が「NotSupported」に設定されている時の MDB の確認応答に関するコードを修正することで、この問題は解決しました。</p>                                                                                                                                                                                                                                                                                                                                                                                         |
| CR091827 | <p>WebLogic Server 6.1 SP02 では、Oracle Thin Driver を使用して接続プールが作成され、JMS JDBC ストアが (JDBC のロギングを有効にして) そのプールを使用するようにコンフィグレーションされている場合に、「Connection has already been closed」という SQL 例外が生成されました。このエラー メッセージの後、JDBC ログ ファイルには何も記録されませんでした。</p> <p>コードの修正で、JDBC のログ エラーが防止されます。</p>                                                                                                                                                                                                                                                                                                                                                                                   |
| CR091844 | <p>WebLogic Server 7.0 SP01 では、JMS キューからまだメッセージを受信している <code>ServerSessionPool</code> のある <code>QueueConnection</code> で「stop」を呼び出すと、エンドレス ループが生じ、CPU 使用率が 100% になっていました。</p> <p>この問題は、コードを修正して解決されました。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| CR092110 | <p>WebLogic Server 7.0 SP01 では、JMS 接続ファクトリのフロー制御設定が <code>FlowMinimum</code> を 2、<code>FlowMaximum</code> を 8 に設定するようにコンフィグレーションされていました。サーバを再起動した後は、次のコンフィギュレーション例外が送出されました。</p> <pre data-bbox="292 1203 1170 1359">&lt;Critical&gt; &lt;WebLogic Server&gt; &lt;000364&gt; &lt;Server failed during initialization. Exception:weblogic.management.configuration.ConfigurationException: invalid value '8' for attribute 'FlowMaximum' of MBean "mydomain:Name=MyJMS Connection Factory,Type=JMSSConnectionFactory".</pre> <p>この問題は、<code>FlowMinimum</code> フィールドと <code>FlowMaximum</code> フィールドのリーガルチェックを廃止することで解決しました。これは、このリリースで実行できるリーガルチェックの制限です。</p> |



| 変更要求番号   | 説明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CR092464 | <p>WebLogic Server 6.0 SP02 と SP04 では、恒久トピック サブスクライバが JMS サーバに接続され、メッセージがそのトピックに送信されると、そのメッセージが受信側で受信されてコミットされていても Bytes Current Count の値がメッセージのサイズ分だけ増やされていました。Messages Current Count の対応する値は影響を受けませんでした。</p> <p>恒久トピック サブスクライバが JMS サーバに接続されていない状態で、トピックに送信されたメッセージがある場合は、Messages Current Count と Bytes Current Count の両方が増やされました。恒久トピック サブスクライバが JMS サーバに再び接続されると、それに応じて Messages Current Count と Bytes Current Count が両方とも減らされました。この不一致の結果として、恒久トピック サブスクライバに未処理のメッセージがなくても「Bytes Current Count」の値はゼロになりません。</p> <p>分析の結果、backend/BEConsumer.java addMessages() において bytesCurrentCount が不適切に更新され、Administration Console に正しくない統計が表示されていることが判明しました。</p> <p>この問題は、コードを修正して解決されました。</p> |
| CR092468 | <p>WebLogic Server 7.0 SP01 では、JMS 接続ファクトリの (フロー制御の) FlowMinimum の最小許容値 0 がメッセージ プロデューサに悪い影響を持つという分析結果が出ました。値を 0 にするとプロデューサでメッセージの送信を完全に止められるように思えますが、実際には、メッセージは少しずつ送信され続けるようになっていました。</p> <p>この問題は、FlowMinimum の最小許容値を 1 に変更することで解決しました。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| CR093177 | <p>WebLogic Server 7.0 SP01 では、メッセージングブリッジに関する Administration Console のラベル、オンライン ヘルプ、およびログメッセージが、[最大待機時間] 属性および [増加遅延] 属性の測定方法と一致していませんでした。これらの属性は、「ミリ秒」ではなく「秒」を単位とします。</p> <p>この問題は、それらの属性の単位が「秒」であることを反映して Administration Console のラベル、オンライン ヘルプ、およびログメッセージを修正することで解決しました。</p>                                                                                                                                                                                                                                                                                                                                                                                                          |

---

| 変更要求番号   | 説明                                                                                                                                                                                                                                                                                                                   |
|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CR093712 | <p>WebLogic Server SP04 では、メッセージングブリッジの [最大待機時間] の設定値に達した後、メッセージがサーバのログファイルに記録されていました。多くのメッセージングブリッジがコンフィグレーションされている場合は、ログファイルがすぐに一杯になり、場合によっては非常に大きくなるようになっていました。それらのメッセージは、現在はメッセージングブリッジで実行時のデバッグが有効になっている場合だけ生成されるようになりました。</p>                                                                                  |
| CR094414 | <p>WebLogic Server 7.0 SP02 のメッセージングブリッジのドキュメントでは、メッセージングブリッジの [Execute Thread Pool Size] フィールドの値は -1 でも有効であり、この値を使用すれば、このスレッドプールが無効になって、すべてメッセージングブリッジが WebLogic Server のデフォルト スレッドプールを使用するようになるとされていましたが、このフィールドの値 -1 は実際には許可されていませんでした。</p> <p>コードの修正により、[Execute Thread Pool Size] を「-1」に設定することができるようになりました。</p> |
| CR094670 | <p>WebLogic Server 7.0 SP01 では、信頼性のあるセキュリティ関係を確立していない WebLogic Server ドメインをメッセージングブリッジを使用して相互運用するときには次のセキュリティ エラー メッセージが生成されました。</p> <pre data-bbox="292 881 911 942">java.lang.SecurityException: Invalid Subject:<br/>principals=[searchuser]</pre> <p>WebLogic Server のセキュリティでは、信頼性のないドメイン間の同期処理が可能となりました。</p>    |
| CR096229 | <p>WebLogic Server 7.0 では、分散送り先を通じてアクセスされた場合、物理送り先のセキュリティに効果がありませんでした。たとえば、分散送り先に物理的なキューメンバーが 2 つ (SecQ1 と SecQ2) あり、いずれかの物理キューにメッセージを「送信」するための JMS セキュリティ特権がユーザに与えられていない場合でも、そのユーザは分散送り先を通じて正常にメッセージを送信することができました。</p> <p>この問題は、分散送り先の物理送り先メンバーと照らし合わせてセキュリティをチェックすることで解決されました。</p>                                 |

---

| 変更要求番号   | 説明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CR096854 | <p>ユーザがクラスタ内の 1 つのサーバでメッセージングブリッジを設定し、その後、そのブリッジをクラスタ内の別のサーバに移行しました。その場合に、2 番目のサーバを起動すると次の例外が発生しました。</p> <pre>(java.lang.Exception: java.lang.NullPointerException at weblogic.jms.bridge.internal.MessagingBridge.getConnections(Mess agingBridge.java:714) at weblogic.jms.bridge.internal.MessagingBridge.execute(MessagingBr idge.java:919) at weblogic.kernel.ExecuteThread.execute(ExecuteThread.java:153) at weblogic.kernel.ExecuteThread.run(ExecuteThread.java:134)</pre> <p>メッセージングブリッジは移行可能なターゲットを対象とされた場合に初期化手順の 1 段階をスキップし、その結果としてブリッジはコンフィグレーションからリソースアダプタの情報を取得できませんでした。NullPointerException は、ブリッジが後でリソースアダプタにアクセスしようとしたときに発生していました。</p> <p>この問題は、移行可能ターゲットの場合のメッセージングブリッジの初期化コードを修正して解決されました。</p> |
| CR097451 | <p>WebLogic Server 7.0 SP01 では、ステートレス EJB のトピックメッセージが接続の開始前に TopicSubscriber クライアントに配信されていました。</p> <p>コードの修正により、接続の開始前にメッセージが配信されることがなくなります。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| CR097538 | <p>WebLogic Server 7.0 SP01 では、JMS セレクタの長さ属性が 3 未満の場合に次の解析セレクタエラーが発生しました。</p> <pre>weblogic.jms.common.InvalidSelectorException: Error parsing selector</pre> <p>この問題は、コードを修正して解決されました。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| CR097963 | <p>WebLogic Server 7.0 では、JMS サーバに着信キューと発信キューがあり、別のサーバにある MDB が 1 対 1 の関係でその着信キューからのメッセージを消費し、その発信キューへのメッセージを生成する場合に、それはすべて MDB によって開始されたトランザクションの中で行われていました。</p> <p>そのトランザクションは、MDB がメッセージを発信キューに入れたときに終了していました。JMS サーバが停止して再起動した場合、メッセージは失われていました。10,000 メッセージの負荷で、一部 (少量) のメッセージが失われていました。</p> <p>コードの修正により、XASession は CLIENT_ACK 確認応答を実行します。</p>                                                                                                                                                                                                                                                                                                                                                               |

---

| 変更要求番号   | 説明                                                                                                                                                                                                                                                                                                                                         |
|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CR098049 | <p>WebLogic Server 7.0 SP01 では、重複したメッセージ ID が JMS で生成される可能性があります。</p> <p>この問題は、コードを修正して解決されました。</p>                                                                                                                                                                                                                                        |
| CR098268 | <p>WebLogic Server 7.0 SP01 では、メッセージ駆動型 MBean が、XAConnectionFactory を使用している場合に XAConnection を開いていました。つまり、トランザクション非対応 MDB の場合、コンテナは XASession を使用して接続し、それがクライアントの確認応答だと気付いていなかったため、「確認応答」は呼び出されず、メッセージは確認応答されていませんでした。</p> <p>コードの修正によってこの動作は変更されました。そのため、返される接続ファクトリのタイプに関係なく、MDB のコンテナは XA が必須の場合に XAConnection と XASession のみを作成します。</p> |

---

---

**変更要求番号 説明**

---

|          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CR098280 | <p data-bbox="356 253 1257 343">XAQueueConnection または XATopicConnection オブジェクトを使用して XA 非対応セッションを作成する場合の、セッション作成メソッドの動作が変更されました。この変更の前は、次のような動作でした。</p> <ul data-bbox="356 361 1257 534" style="list-style-type: none"><li data-bbox="356 361 1257 387">• XAQueueConnection.createQueueSession は XAQueueSession を作成する</li><li data-bbox="356 404 1257 534">• XATopicConnection.createTopicSession は XATopicSession を作成する</li></ul> <p data-bbox="356 447 1257 534">いずれの場合も、ユーザ指定の確認応答モードとトランザクション フラグは無視されて、AUTO_ACKNOWLEDGE モードと <i>false</i> のトランザクション設定で置き換えられました。</p> <p data-bbox="356 543 1257 633">一方、新しい動作では、XAQueueConnection.createQueueSession および XATopicConnection.createTopicSession メソッドが、QueueConnection および TopicConnection の対応するメソッドとまったく同様に、次のように動作します。</p> <ul data-bbox="356 642 1257 772" style="list-style-type: none"><li data-bbox="356 642 1257 668">• QueueConnection.createQueueSession は QueueSession を作成する</li><li data-bbox="356 677 1257 703">• TopicConnection.createTopicSession は TopicSession を作成する</li><li data-bbox="356 711 1257 737">• XAQueueConnection.createQueueSession は QueueSession を作成する</li><li data-bbox="356 746 1257 772">• XATopicConnection.createTopicSession は TopicSession を作成する</li></ul> <p data-bbox="356 781 1257 841">さらに、ユーザ指定の確認応答モードとトランザクション フラグ設定は、これらの 4 つの各メソッドで受け付けます。</p> <p data-bbox="356 850 1257 1058">上記の 4 つの接続メソッドは、接続ファクトリで XAServerEnabled フラグが有効になっている場合、異なる動作をします。このフラグが有効な場合、4 つのすべてのメソッドは、サーバ上で呼び出された場合は XAQueueSession (または XATopicSession) セッションを作成し、クライアント上で呼び出された場合は XA 非対応の QueueSession または TopicSession セッションを作成します。作成されたセッションはユーザ指定の確認応答モードを受け付けますが、XA をサポートしているため、トランザクション フラグは無視されます。</p> <p data-bbox="356 1067 1257 1258">以前のバージョンの WebLogic Server 7.0 では、XAConnectionFactoryEnabled フラグが有効になっているときに接続ファクトリから作成される接続オブジェクトは、XAQueueConnection または XATopicConnection オブジェクトのように動作しました。新しい動作の変更によって、XAQueueConnectionFactory または XATopicConnectionFactory に接続を明示的にキャストして、いずれかの createXA メソッドを呼び出さない限り、この動作は見えなくなりました。</p> <p data-bbox="356 1267 1257 1347">この変更の前は、接続ファクトリで XAConnectionFactoryEnabled フラグを設定する場合、接続ファクトリをいずれかの XA 接続ファクトリ クラスにキャストしていなくても、createQueueSession と createTopicSession の動作が異なっていました。</p> |
|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

---

---

| 変更要求番号   | 説明                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CR098280 | <p>WebLogic Server 7.0 SP01 では、クライアントサイドおよびサーバサイドで、XAConnection オブジェクトの <code>createSession()</code> メソッドを呼び出すと XASession オブジェクトが作成されていました。</p> <p>XASession オブジェクトの <code>createSession()</code> を呼び出すと非 XA「セッション」オブジェクトが作成されるように変更されています。</p>                                                                                                                                                                                                                                  |
| CR098692 | <p>WebLogic Server 6.1 SP04 では、ガベージ コレクションまたはトランザクション タイムアウトの後に JMS キューでのメッセージの同期受信が停止しました。</p> <p>この問題は、JMS ディスパッチャのコードをリファクタリングしてスレッド管理を改善することで解決しました。</p>                                                                                                                                                                                                                                                                                                                     |
| CR099455 | <p>WebLogic Server 7.0 SP01 では、コンシューマに向けて送信されたメッセージがフロントエンド セッションに到着したが、そのメッセージがすでに閉じているコンシューマに向けられている場合、リストが特に考慮なく切り詰められて、メッセージが失われる可能性があります。</p> <p>この問題は、潜在的な競合状態を防止するコード修正で解決されました。</p>                                                                                                                                                                                                                                                                                      |
| CR100015 | <p><code>dispatch-policy</code> がメッセージ駆動型 Bean で対応されるようになりました。つまり、メッセージ駆動型 Bean をユーザ定義の実行キューに割り当てることができるようになったのです。</p>                                                                                                                                                                                                                                                                                                                                                           |
| CR100076 | <p>WebLogic Server 7.0 では、JMS サーバが WebLogic Server インスタンス (server1) から別のインスタンス (server2) に移行された場合、その JMS サーバは server1 で非アクティブ化され、そのすべての送り先がサスペンドされていました。ただし、送り先がサスペンドされるときに、メモリ内のメッセージはクリーンアップされませんでした。そのため、JMS サーバが server1 に再び移行されたときに、移行前に JMS サーバ上にあったすべてのメッセージが server2 でキューから出されている場合でも表示されていました。JMS サーバが server2 に移行された後に server1 でメモリ内のメッセージがクリーンアップされたときには、重複メッセージはないはずです。</p> <p>この問題は、JMS サーバを別の WebLogic Server インスタンスに移行するときにメモリ内のメッセージのリストをクリーンアップすることで解決しました。</p> |

---

---

| 変更要求番号   | 説明                                                                                                                                                                                                                                                                                                                                                                                                  |
|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CR100083 | <p>WebLogic Server 7.0 では、JMS サーバと関連付けられた実行時 MBean (JMSServerRuntimeMBean と JMSDestinationRuntimeMBean) が、JMS サーバがサスペンドされた後に管理サーバから登録解除されませんでした。その結果、JMS サーバが移行可能な対象をターゲットとしている場合に、それらの MBean の 2 つのインスタンスが、JMS サーバが別のサーバに移行された後も Administration Console のモニタ ページに表示されます。1 つのインスタンスが元のサーバインスタンスにあり、もう 1 つのインスタンスが JMS サーバの移行先サーバにあったのです。</p> <p>実行時 MBean の登録解除と登録のプロセスが、現在は適切に処理されるようになっていきます。</p> |
| CR101298 | <p>ファイアウォールの生存期間が切れるなどして、長い時間アイドル状態の JMS JDBC 接続が「bad」とマークされた場合に、JMS がその「bad」の接続を使用しようとして失敗していました。この失敗がすぐに復帰するか、8 ~ 10 分ほどかかるかは、チューニング パラメータによって決まります。</p> <p>この問題は、JMS がアイドル状態の場合に、接続を維持するために 5 分ごとにデータベースに ping を実行するようコードを修正して解決されました。5 分が選ばれたのは、どのファイアウォールのタイムアウトよりも短い時間だと考えられたからです。</p>                                                                                                        |
| CR103213 | <p>サーバが JMS メッセージの送信に失敗したときに、クライアントでエラー メッセージが表示されませんでした。トランザクション マネージャはメッセージが異常であると宣言し、JMS はリソースの取得に失敗したときにメッセージをロールバックしていました。</p> <p>トランザクションの取得の失敗はクライアントに通知されるようになりました。コミットが失敗したことも報告されます。</p>                                                                                                                                                                                                  |
| CR105237 | <p>IIOP プロトコルで JMS を使用しているときに、クライアントによる異常な切断がサーバで正しく検出されませんでした。そのため、クライアントに関連付けられたステートフル情報 (JMS クライアント ID など) で問題が発生しました。</p> <p>異常な切断は適切に検出されて処理されるようになりました。</p>                                                                                                                                                                                                                                   |

---

---

**変更要求番号**   **説明**

---

CR105337      負荷のかかる状況で、テスト JMS アプリケーションに次の例外が発生しました。

```
java.lang.OutOfMemoryError weblogic.jms.common.TransactionRolledBackException: at
weblogic.jms.backend.BEConsumer.expireTimeout(BEConsumer.java:1620) at
weblogic.jms.backend.BEXATranEntryBlockingConsumer.startRollback(BEXATranEntryB
lockingConsumer.java:72) at
weblogic.jms.backend.BEXAResource.rollback(BEXAResource.java:1205) at
weblogic.transaction.internal.ServerResourceInfo.rollback(ServerResourceInfo.java:1400) at
weblogic.transaction.internal.ServerResourceInfo.rollback(ServerResourceInfo.java:664) at
weblogic.transaction.internal.ServerSCInfo.startRollback(ServerSCInfo.java:365) at
weblogic.transaction.internal.ServerTransactionImpl.localRollback(ServerTransactionImpl.j
ava:1521) at
weblogic.transaction.internal.ServerTransactionImpl.globalRollback(ServerTransactionImpl.
java:2142) at
weblogic.transaction.internal.TransactionImpl$1.execute(TransactionImpl.java:1656) at
weblogic.kernel.ExecuteThread.execute(ExecuteThread.java:211) at
weblogic.kernel.ExecuteThread.run(ExecuteThread.java:198)
weblogic.jms.common.JMSException: Only one thread may use a JMS Session at a time. at
weblogic.jms.frontend.FESession.rollbackAfterRecover(FESession.java:1773) at
weblogic.jms.frontend.FESession.recover(FESession.java:1373) at
weblogic.jms.frontend.FESession.invoke(FESession.java:2253) at
weblogic.jms.dispatcher.Request.wrappedFiniteStateMachine(Request.java:609) at
weblogic.jms.dispatcher.DispatcherImpl.dispatchSyncNoTran(DispatcherImpl.java:339) at
weblogic.jms.client.JMSSession.recoverGuts(JMSSession.java:868) at
weblogic.jms.client.JMSSession.rollback(JMSSession.java:632) at
com.ncr.crm.framework.jms.DefaultJmsQueueReceiver.rollback(DefaultJmsQueueReceiver.
java:184) at
com.ncr.crm.framework.jms.DemoMessageConsumer.run(JmsMemoryLeakDemo.java:163)
at java.lang.Thread.run(Thread.java:479) ----- Linked Exception -----
weblogic.jms.common.JMSException: Only one thread may use a JMS Session at a time. at
weblogic.jms.frontend.FESession.recover(FESession.java:1219) at
weblogic.jms.frontend.FESession.invoke(FESession.java:2253) at
weblogic.jms.dispatcher.Request.wrappedFiniteStateMachine(Request.java:609) at
weblogic.jms.dispatcher.DispatcherImpl.dispatchSyncNoTran(DispatcherImpl.java:339) at
weblogic.jms.client.JMSSession.recoverGuts(JMSSession.java:868) at
weblogic.jms.client.JMSSession.rollback(JMSSession.java:632) at
com.ncr.crm.framework.jms.DefaultJmsQueueReceiver.rollback(DefaultJmsQueueReceiver.
java:184) at...
```

調査の結果、WebLogic JMS のガベージ コレクション機能は、リンクされた長いリストを扱うときにメモリを再要求するために速度が遅くなることが判明しました。リストを分割するようにコードを変更しました。その修正で、問題は解決されました。

---



---

| 変更要求番号   | 説明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CR105980 | 分散キューの各メッセージごとに <code>QueueReceiver</code> を作成して、 <code>QueueReceiver</code> が停止された後で永続化しているときにメモリ リークが発生しました。<br>コードを変更して、 <code>BasicServiceOffer</code> および <code>DistributedDestinationImpl</code> オブジェクトのメモリ リークを修正しました。                                                                                                                                                                                                                                                                    |
| CR106789 | 使用されていない JMS オブジェクトが開いたままであることが原因で、メモリ リークが発生していました。<br>次の修正によって問題は解決されました。 <ul style="list-style-type: none"><li>- 同じ送り先上で作成されたすべてのプロデューサの送り先が正規化された。</li><li>- <code>FEProducer</code> を <code>DispatcherWrapperState</code> に置き換えるときに、正しい <code>DispatcherWrapperState</code> からリスナが削除された。</li><li>- <code>FEProducer</code> を無条件にクローズするときに、正しい <code>DispatcherWrapperState</code> からリスナが削除された。</li><li>- <code>BackEnd</code> が停止するときに、<code>FrontEnd</code> 上の一時的な送り先のリークが削除された。</li></ul> |

---

## JNDI

---

| 変更要求番号   | 説明                                                                                                                      |
|----------|-------------------------------------------------------------------------------------------------------------------------|
| CR084381 | JNDI コンテキストの <code>list("")</code> メソッドが、正しいクラス名を返していませんでした。<br>現在は、「list」メソッドを使用したときに WebLogic Server から実際のクラス名が返されます。 |
| CR087237 | サーバで <code>kill -STOP</code> が呼び出されたときに RMI のフェイルオーバが長い時間かかっていました。<br>コードの変更により、フェイルオーバに必要な時間が短縮されました。                  |

---

---

|                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CR093700              | <p>JNDI ツリーがリバインドで適切に更新されていませんでした。</p> <p>クラスター ノードの 1 つからクラスター可能リモート オブジェクトを JNDI アンバインドしても、そのオブジェクトがアンバインドされたノードに呼び出しが届いたときにそのオブジェクトがまだアクセス可能な状態になっていました。この問題は修正済みで、リモート メソッド呼び出しでは次に利用可能なノードが選択されます。</p>                                                                                                                                                                                                                                                                                           |
| CR093799              | <p>複合名に対してコンテキスト ルックアップを行うと、名前の全体または一部が解決できない場合に <code>NameNotFoundException</code> が発生していました。</p> <p><code>RemainingName</code> が <code>CompositeName</code> である場合は、列挙があるはずですが、残りの名前は単一文字列として扱われ、WebLogic Server は <code>CompositeName</code> の作成時に「.」をセパレータとして渡していました。現在は、「.」セパレータが「/」に適切に置換され、列挙が正しく作成されるようになっています。</p>                                                                                                                                                                            |
| CR103148、<br>CR110890 | <p>WebLogic Server 7.0 SP2 では、<code>LoadBeforeAppDeployments</code> が <code>true</code> に設定されている場合、起動クラスは <code>LOCAL_JNDI_NAME</code> のローカル インスタンス <code>MBeanHome (weblogic.management.home.localhome)</code> にアクセスできませんでした。次のようなエラー メッセージにより失敗しました。</p> <pre data-bbox="294 795 1166 855">Unable to resolve 'weblogic.management.home.localhome' Resolved:<br/>'weblogic.management' Unresolved:'home'</pre> <p>他の JNDI 名のバインディングを起動クラスで使用できるように <code>AdminService</code> へ移動することで、問題を修正しました。</p> |
| CR106853              | <p>動的プロキシ オブジェクトをローカル JNDI ツリーにバインドしようとすると、<code>ClassNotFoundException</code> が発生しました。</p> <p>コードを変更して、<code>resolveProxyClass()</code> を実装し、アプリケーションからインタフェースをロードするためにスレッド コンテキスト クラスローダを使用することで、問題を修正しました。</p>                                                                                                                                                                                                                                                                                     |

---

# JSP とサーブレット

| 変更要求番号   | 説明                                                                                                                                                                                                                                                                                                                        |
|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CR083597 | <p>&lt;%@ page contentType="text/html; charset=UTF-8" %&gt;などのページ ディレクティブを使用してコンテンツ タイプが設定されるときに、そのコンテンツ タイプは正しく設定されず、目的の出力が得られませんでした。</p> <p>この問題は、コードの変更によって取り除かれました。</p>                                                                                                                                               |
| CR088044 | <p>HTTP のログ オプションとして「拡張フォーマット」を選択し、サーバを再起動すると、Null ポインタ例外が送出されました。</p> <p>logManagerHttp の作成時に access.log ファイルを開くコードを追加することで、この問題は取り除かれました。</p>                                                                                                                                                                           |
| CR088350 | <p>HttpProxyServlet は、サードパーティのサーバに到達できませんでした。</p> <p>この問題は、サーバが ContentLength を送信せず、データをいつまでも送信し続けることが原因で発生しました。コンテンツの長さがなく、チャンク転送されていない場合、サーバはストリームの最後まで読み込みます。</p> <p>HttpProxyServlet は、バイト単位で読み書きするように変更されています。</p>                                                                                                   |
| CR088747 | <p>weblogic.xml で CookiePath を明示的に設定していない Web アプリケーションに対するリクエストの後で、ユーザ エージェントは WebLogic Server セッションを再確立できませんでした。</p> <p>WebLogic Server は、同じドメインからの 2 番目のクッキーが存在し、それがリクエストの URL と一致するパスを指定している場合でも、weblogic.xml で指定されたクッキー名と名前が同じ最初のクッキーを使用します。</p> <p>以前は、クッキー ベクトルの後方へ向かって JSESSIONID を検索していましたが、検索方向は前方に変更されています。</p> |
| CR088759 | <p>redirect-with-absolute-url 機能が、j_security_check では機能しませんでした。応答の Location ヘッダがまだ絶対パスでした。</p> <p>redirect-with-absolute-url が FormSecurityModule で対応されるようにコードが修正されています。</p>                                                                                                                                              |

---

|          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CR089347 | <p>ネットワーク チャネルでは、<code>getServerPort</code> メソッドは、<code>FrontendHTTPPort</code> 値が <code>config.xml</code> の <code>WebServer</code> 要素で設定されている場合に <code>FrontendHTTPPort</code> の値を返します。</p> <p>(WebLogic Server の前にプロキシがある場合など) <code>HOST</code> ヘッダに WebLogic Server デフォルト リスン ポート以外のポートが含まれている場合は、<code>FrontendHTTPPort</code> を指定することで、<code>HOST</code> ヘッダで指定されたポートではなく WebLogic Server ポートが <code>getServerPort()</code> から返されるようになります。ただし、この <code>FrontendHTTPPort</code> 値は、別のポートで動作しているネットワーク チャネルの場合も含めてすべての <code>getServerPort()</code> の呼び出しで返されていました。</p> <p><code>request.getAttribute("weblogic.servlet.network_channel.port")</code> を使用して、リクエストが受信されたポートの値が返されるようにコードが修正されています。</p> |
| CR089582 | <p>リクエストに <code>Host</code> ヘッダがない場合に、不正確な <code>Date</code> ヘッダが返されていました。</p> <p>この問題は、エラー メッセージを送信する前にリクエストの開始時刻を設定することで修正されました。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| CR089868 | <p>CR083654 を修正するために作成されたパッチを適用すると、マルチバイトのヘッダが不正確になっていました。</p> <p>この問題は、<code>WebServer</code> に <code>UseHeaderEncoding</code> オプションを追加することで解決されました。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| CR090188 | <p>チャンク転送では、WebLogic Server はデータに付加されたチャンク サイズを扱っていません。</p> <p><code>setChunking</code> は、<code>ServletRequestImpl</code> の <code>mergePostParams</code> から、<code>request.setInputStream</code> 直後の <code>MuxableSocketHttp</code> に移動しています。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| CR090465 | <p>ユーザ定義のエラー ページがある Web アプリケーションをホストすると、WebLogic Server が、400 番のエラーに対して <code>Connection: close</code> ヘッダを設定しないで次の応答を返し、直ちに接続を閉じていました。</p> <pre>HTTP/1.1 400 Bad Request Date: Thu, 01 Jan 1970 00:00:00 GMT Content-Length: 463 Content-Type: text/html Last-Modified: Thu, 07 Nov 2002 21:56:52 GMT &lt;!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2//EN"&gt; &lt;html&gt;</pre> <p>この問題は、コードを修正して解決されました。</p>                                                                                                                                                                                                                                                                                           |

---

---

|          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CR090555 | <p>Http ヘッダに X_WEBLOGIC_CLUSTER_LIST がない場合は、X_WEBLOGIC_CLUSTER_HASH をチェックするコードのためにサーバリストが作成されませんでした。</p> <p>この問題は、応答ヘッダでハッシュの後にリストが送信される場合にサーバでハッシュを格納できるようにすることで修正されました。</p>                                                                                                                                                                                                                                                                                                                                                                                                                  |
| CR091759 | <p>RequestDispatcher.forward が、URL ヘッダのバージョン番号を削除していました。</p> <p>この問題は、forward() の実行時に setRequestURI() で processPathParameters を false に設定して修正されました。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| CR091831 | <p>HttpURLConnection の WebLogic Server 実装は、POST メソッドの使用時にサーバ側でキープアライブ接続がタイムアウトになっているかどうかをチェックしなかったため、フラッシュ時にエラー「Connection aborted by peer: socket write error」が発生していました。</p> <p>サーバでキープアライブを無効にすると、問題がなくなります。キープアライブの期間を長くしても、異なる長さのスリープで同じ問題が発生しました。</p> <p>タイムスタンプの更新前に HttpClient が非 null であるようにするチェック機能が追加されました。</p>                                                                                                                                                                                                                                                                      |
| CR091922 | <p>サブレット コンテナがエラー ページにログイン例外を渡していませんでした。現在、エラー JSP は例外にアクセスできます。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| CR092039 | <p>WebLogic Server は、JSP タグの charset の値に余分な引用符があると UnsupportedOperationException を送出していました。</p> <p>この問題は、ヘッダの属性値で HTTP 仕様のサポートを追加することで解決しました。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| CR092039 | <p>WebLogic Server 6,1 SP03 では、JSP タグの charset の値に余分な引用符があると、JspParser から UnsupportedOperationException が送出されていました。</p> <pre>&lt;%@ page contentType="text/html; charset=\"Shift_JIS\" %&gt;</pre> <p>のように余分な引用符があると、次のエラーが発生していました。</p> <pre>java.lang.RuntimeException: Unknown/unsupported charset: "Shift_JIS" - java.io.UnsupportedEncodingException: Charset: '"Shift_JIS"' not recognized, and there is no alias for it in the WebServerMBean</pre> <p>WebLogic Server が、HTTP 仕様を適切にサポートしていませんでした。</p> <p>この問題は、引用符付き文字列およびコメントのサポートを Content-Type ヘッダの charset 属性値に追加することで解決されました。</p> |

---

## サービス パック 1 から 6 で解決済みの問題

---

|          |                                            |
|----------|--------------------------------------------|
| CR092173 | ファイル サープレットの GET 機能はバイトの範囲をサポートするようになりました。 |
|----------|--------------------------------------------|

---

|          |                                                                                                             |
|----------|-------------------------------------------------------------------------------------------------------------|
| CR092255 | クッキーの引用符が、WebLogic Server によって取り除かれていました。その結果、特定の環境ではクッキーの値が読み取れなくなっていました。現在は、引用符がクッキーに残されるようにコードが変更されています。 |
|----------|-------------------------------------------------------------------------------------------------------------|

---

|          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CR092377 | <p>Web アプリケーションの web.xml ファイルで、</p> <pre>&lt;user-data-constraint&gt;<br/>&lt;description&gt;USE SSL&lt;/description&gt;<br/>&lt;transport-guarantee&gt;CONFIDENTIAL&lt;/transport-guarantee&gt;<br/>&lt;/user-data-constraint&gt;</pre> <p>リソースに対してこのように設定し、HTTP を使用してその保護されたリソースにアクセスすると、リソースを表示する許可が与えられていないことを通知するエラーがサーバから送出されていました。サーバは、サープレット仕様 2.3 のページ 87 に従って認証し、HTTPS に切り替える機会をユーザに与えていませんでした。</p> <p>現在、コンテナは元のリクエストが HTTP 経由で、リソースが INTEGRAL または CONFIDENTIAL に設定されている場合はクライアントを HTTPS ポートにリダイレクトするようになっています。</p> |
|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

---

|          |                                                                                                                                                  |
|----------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| CR092428 | <p>WebLogic Server は、Netscape クッキーの「,」を文字として受け入れていませんでした。</p> <p>カンマは現在は Netscape のクッキーで許可されています。その理由は、RFC2109 クッキーでのみカンマは区切り文字として使用できるからです。</p> |
|----------|--------------------------------------------------------------------------------------------------------------------------------------------------|

---

|          |                                                                                                                                                       |
|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| CR092545 | <p>リクエストがプロキシ経由で発行され、その最初のリクエストが終了する前に同じリクエストが再び発行されると、例外が発生していました。</p> <p>プロキシは、クライアントへの書き込みに失敗しても（接続がブラウザによって閉じられても）IIS から入力ストリームを排出するようになりました。</p> |
|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------|

---

---

|          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CR092778 | <p>WebLogic Server 6.1 SP04 では、HttpRequest の JISAutoDetect エンコーディングで、次の UnsupportedEncodingException が送出されました。</p> <pre>java.io.UnsupportedEncodingException: JISAutoDetect at sun.io.Converters.getConverterClass (Converters.java:102) at sun.io.Converters.newConverter (Converters.java:133) at sun.io.CharToByteConverter.getConverter (CharToByteConverter.java:62) at weblogic.servlet.internal.ServletRequestImpl.setCharacterEncoding (ServletRequestImpl.java:344) ...</pre> <p>この問題は、WebLogic Server 6.1 SP03 では起こりませんでした。</p> <p>この問題の原因は、ServletRequestImpl.java が sun.io.ByteToCharConverter.ではなく CharToByteConverter クラスを使用していたことです。CharToByteConverter は、入力コンバータ用です。CharToByte は、出力コンバータ用です。JISAutoDetect は、入力ストリームでのみ使用できます。</p> <p>この問題は、コードを修正して解決されました。</p> |
| CR093014 | <p>Java のコメントが複数のスクリプトレットにまたがると、jspc が失敗していました。</p> <p>ScriptletScopeLexer が Java のコメント全体をスキップするようにするコードを追加して問題は解決されました。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| CR093112 | <p>JSP に <code>&lt;%@ include file="./file.inc" %&gt;</code> のような行があり、file.inc に <code>&lt;%@ taglib uri="/c.tld" prefix="c" %&gt;</code> のような行があり、c.tld にバリデータがある場合は、<code>&lt;%@ include file="./file.inc"%&gt;</code> およびタグライブラリの使用に関する JSP コンパイラ エラーが返されていました。</p> <p>この問題は、XML フォーマットに変換される JSP ページの currentURI の参照を追加することで修正されました。この修正の結果、アプリケーション ルートを基準とした、インクルードされた JSP の相対的な位置を実装で計算することができます。</p>                                                                                                                                                                                                                                                                                                                                                                    |
| CR093167 | <p>HTTPClusterServlet を使用してバックエンド クラスタ上のセキュアな Web アプリケーションにアクセスする場合、ユーザ名とパスワードの入力後に Internet Explorer が 30 秒間「ハング」します。</p> <p>WebLogic Server は現在は、Connection: Close ヘッダ要求がブラウザにプロキシされている場合はクライアントとプロキシ サーバ間の接続を閉じるようになりました。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |

---

## サービス パック 1 から 6 で解決済みの問題

---

CR093209      プロパティがアプリケーションの java プロトコルに設定された、正常にデプロイされている Web アプリケーションがアクセスされたときに、WebLogic Server から NullPointerException が送出されていました。

現在は、`source.getURL()` の null 値がチェックされます。

---

CR093209      新規にインストールした WebLogic Server 6.1 SP04 では、プロパティが Java プロトコルに設定されている、正常にデプロイされた Web アプリケーションにアクセスすると、null ポインタ例外が送出されていました。ブラウザからは、500 番のサーバ内部エラーが返されました。この問題は、アップグレードインストールでは発生しませんでした。

```
<Dec 16, 2002 11:59:05 AM PST> <Error> <HTTP>
<[WebAppServletContext(2092664,sampleApp,/sampleApp)] Servlet
failed with
Exception
java.lang.NullPointerException
at weblogic.servlet.JSPServlet.service(JSPServlet.java:132)
at javax.servlet.http.HttpServlet.service(HttpServlet.java:853)
at
weblogic.servlet.internal.ServletStubImpl.invokeServlet(Servlet
StubImpl.java:262)
at
weblogic.servlet.internal.ServletStubImpl.invokeServlet(Servlet
StubImpl.java:198)
at
weblogic.servlet.internal.WebAppServletContext.invokeServlet(We
bAppServletContext.java:2637)
at
weblogic.servlet.internal.ServletRequestImpl.execute(ServletReq
uestImpl.java:2359)
at
weblogic.kernel.ExecuteThread.execute(ExecuteThread.java:139)
at weblogic.kernel.ExecuteThread.run(ExecuteThread.java:120)
```

分析の結果、`ZipSource.getURL()` のエラーであることが判明しました。この問題は、`source.getURL()` の null 値をチェックするロジックを追加することで解決しました。

---



---

|          |                                                                                                                                                                                                                                                                                                                                             |
|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CR093625 | <p>サーバが、JSP ページの JSP includes に対して出力を誤って送っていました。</p> <p>この問題は、includes、forwards、wrapped-responses、および JSP BodyTags の処理を統一するようにサーブレット エンジンを変更して解決されました。</p>                                                                                                                                                                                  |
| CR093634 | <p>新しいパラメータ [WebLogic プラグインを有効化] が WebLogic Server に追加されました。getRemoteAddr を呼び出すと、クライアントの IP アドレスではなく HttpClusterServlet の IP アドレスが返されていました。その理由は、HttpClusterServlet が独自のヘッダ「WL-Proxy-Client-IP」を設定せず、したがってソケットの IP アドレスを使用していたからです。</p> <p>この問題はコードで修正されました。</p>                                                                            |
| CR093755 | <p>無害の警告メッセージがコードから削除されました。</p> <p>context.log.error("Warning: One of the getParameter family of methods called after reading from the ServletInputStream(), can't mix these two!"); が無効になりました。</p>                                                                                                                                         |
| CR094190 | <p>JVM 仕様に従って、メソッド サイズのサイズ制限は 64K です。WebLogic Server 6.1 SP03 で、JSP に多数の本文タグがあるときに、生成された java コードが __jspService メソッドの 64K の制限を超えました。</p> <p>この問題は、コードを変更して解決されました。フォーム &lt;my:tag ... /&gt; の BodyTag が空である場合、生成されるコード (通常は本文の範囲を設定する) が、StandardTagLib.java の静的メソッドへの単一の呼び出し (JSP ソースから普通に呼び出されたら BodyTag に認識させる) によって置き換えられるようになりました。</p> |
| CR094416 | <p>以前のバージョンの WebLogic Server 7.0 では、更新されていなくてもすべての JSP を事前にコンパイルしていました。</p> <p>コードの変更により、WebLogic Server のインスタンスが再起動したときに一時ディレクトリが上書きされないようになりました。</p>                                                                                                                                                                                      |

---

- 
- CR094488      HTTP を使用して開始されたセッションで、セッション データを損失することなく HTTPS リソースに安全にアクセスできるようにする新しい機能が追加されました。この新しい機能を有効にするには、config.xml の WebServer 要素に AuthCookieEnabled="true" を追加します。
- ```
<WebServer Name="myserver" AuthCookieEnabled="true"/>
```
- このように設定すると、HTTPS 接続を通じて認証する場合に新しいセキュアなクッキーがブラウザに送信されます。いったん設定すると、クッキーがブラウザから送信される場合にのみ、セッションは他のセキュリティ制約 HTTPS リソースにアクセスできます。
- 注意：** 普通の HTTP で認証する場合、HTTPS リソースでセキュアなクッキーは設定されず、必要ともされません。保護されていない HTTPS リソースにアクセスする場合、クッキーは検証されません ( ブラウザから送信されないため)。この動作により、ブラウザはユーザのログインなしに、保護されていない HTTPS リソースにアクセスできます。
- 
- CR094663      3 サーバクラスと IIS プロキシ プラグインを使用して負荷テストを実行し、クラスタ内のいずれか 1 つのサーバを停止すると、IIS プロキシから 500 エラーが送出されました。
- 原因は、解析エラーとサーバ アフィニティの問題にありました。
- これらの問題は、コードで修正されています。
- 
- CR095713      Unix プラットフォーム (Solaris、Linux) の場合に、セカンダリ サーバでレプリケート セッションが失われていました。
- web.xml (j2ee 記述子) の session-config 要素は非推奨ではありません。値は分単位です。weblogic.xml (BEA 固有の記述子) の要素は秒単位の値をとり、web.xml で定義された値に優先します。
-

CR096399

JSP コンパイル エラーが不適切に処理されていました。フィルタは

HttpServletResponseWrapper getWriter ではなく OutputStream と通信して、次の結果になりました。

```
java.lang.IllegalStateException: strict servlet API: cannot call
getWriter() after getOutputStream() at
weblogic.servlet.internal.ServletResponseImpl.getWriter(Servlet
ResponseImpl.java:171) at
weblogic.servlet.internal.ServletRequestImpl.reportJSPFailure(S
ervletRequestImpl.java:227) at
weblogic.servlet.internal.ServletRequestImpl.reportJSPCompilati
onFailure(ServletRequestImpl.java:239) at
weblogic.servlet.jsp.JspStub.reportCompilationFailure(JspStub.j
ava:523) at
weblogic.servlet.jsp.JspStub.compilePage(JspStub.java:430) at
weblogic.servlet.jsp.JspStub.prepareServlet(JspStub.java:210)
at
weblogic.servlet.jsp.JspStub.prepareServlet(JspStub.java:164)
at
weblogic.servlet.internal.ServletStubImpl.getServlet(ServletStu
bImpl.java:517) at
weblogic.servlet.internal.ServletStubImpl.invokeServlet(Servlet
StubImpl.java:351) at
weblogic.servlet.internal.ServletStubImpl.invokeServlet(Servlet
StubImpl.java:445) at
weblogic.servlet.internal.TailFilter.doFilter(TailFilter.java:2
0) at
weblogic.servlet.internal.FilterChainImpl.doFilter(FilterChainI
mpl.java:27) at WebLogic ServerBugFilter.doFilter(WebLogic
ServerBugFilter.java:27) at
weblogic.servlet.internal.FilterChainImpl.doFilter(FilterChainI
mpl.java:27) at
weblogic.servlet.internal.WebAppServletContext$ServletInvocatio
nAction.run(WebAppServletContext.java:5418) at
weblogic.security.service.SecurityServiceManager.runAs(Security
ServiceManager.java:744) at
weblogic.servlet.internal.WebAppServletContext.invokeServlet(We
bAppServletContext.java:3086) at
weblogic.servlet.internal.ServletRequestImpl.execute(ServletReq
uestImpl.java:2544) at
weblogic.kernel.ExecuteThread.execute(ExecuteThread.java:153)
at weblogic.kernel.ExecuteThread.run(ExecuteThread.java:134)...
WebLogic ServerBugFilter.doFilter after
```

ラッパー対応の RequestCallback を実装して問題を修正しました。

---

CR097719      WebLogic Server SP02、SP03、および SP04 では、WAP デバイスから特定の POST リクエストを取得するときに、Web アプリケーションで `java.util.ConcurrentModificationException` が送出されていました。

```
<29.jan.03 13:03:05 WET> <Error> <HTTP>
<[WebAppServletContext(2810713,TnMFF,/TnMFF)] Servlet failed
with Exception java.util.ConcurrentModificationException at
java.util.HashMap$HashIterator.next(HashMap.java:736) at
weblogic.utils.enumerations.IteratorEnumerator.nextElement(Iter
atorEnumerator.java:25) at
com.colibria.core.xmlswitch.XMLSwitch.getQueryStringXML(XMLSwit
ch.java:347) at ...
```

分析の結果、charset がリクエストの content-type と関連付けられている場合に、アプリケーションでコードがそのエンコーディングを使用してデータを再び読み取り、クエリ パラメータを再設定していることが判明しました。アプリケーションは既に列挙オブジェクトを持っており、それを検索して (`request.getParameterNames`)、パラメータの値を取得しようとします (`request.getParameterValues(k)`)。その時点で、クエリのパラメータは消去されており、`getParameterValues` メソッドがそれを設定しようとした結果、例外が発生します。

文字セットがリクエストの content-type と関連付けられている場合にデータを再び読み取ることができるよう、消去されたクエリ パラメータを設定するようにコードを修正して、この問題は解決しました。

---

CR098366      サーバが起動した直後に、一部のサーバ リクエストが拒否されていました。その理由は、リスン スレッドがリクエストの受信を始める前に MBean がサーバの状態を更新しなかったからです。

この問題は、Web コンテナ内でプール フラグを使用してリスン スレッドのサーバ状態を更新することで解決されました。

---

CR098518      次の EJB/Web アプリケーションの設定によってクラス キャスト例外が発生しました。

```
Servlet Filter1 --> WebApp1 -- forward to WebApp2 ctx --> Servlet
Filter2(WebApp2) --> Ejb Lookup (fails)
```

2 つの Web アプリケーション間で転送するときに、サーブレット フィルタではなくサーブレット内で EJB ルックアップを行うことで、この問題を修正しました。

---

---

CR100172	<p>複数のチャンクになったリクエストがサーブレットにポストされると、多くのリクエストが <code>NumberFormatException</code> で失敗しました。ほとんどの場合、代替リクエストは正常に処理され、次のリクエストが失敗しました。この問題は、リクエストが同じ接続で行われている場合に発生しているようでした。例外のスタックトレースは次のようなものでした。</p> <pre data-bbox="356 364 1243 616">&lt;ExecuteThread: '11' for queue: 'default'&gt; &lt;kernel identity&gt; &lt;&gt; &lt;101017&gt; &lt;[ServletContext (id=4864139,name=387551,context-path=)] Root cause of ServletException&gt; java.lang.NumberFormatException: at java.lang.Integer.parseInt (Integer.java:430) at weblogic.utils.http.HttpChunkInputStream.readChunkSize (HttpChun kInputStream.java:118) at weblogic.utils.http.HttpChunkInputStream.initChunk (HttpChunkInp utStream.java:72)</pre> <p>分析の結果、<code>PostInputStream</code> に 2 つの問題があることが判明しました。</p> <ul data-bbox="356 685 1243 789" style="list-style-type: none"><li>• <code>PostInputStream</code> を再利用するとき、制御変数 <code>readAllChunks</code> をリセットしていなかった。</li><li>• <code>isChunkComplete</code> がチャンクの終了を正しく検出していなかった。</li></ul> <p><code>PostInputStream</code> でストリームを最後まで読み取り、<code>HttpChunkInputStream</code> でチャンクの終わりを正しく検出するようコードを修正して、この問題は解決しました。</p>
CR100298	<p>フィルタサーブレットを使用する、つまり <code>HTTPServletRequest</code> に加えて <code>HTTPRequest</code> ラッパークラスを使用した Web サービスが <code>ClassCastException</code> エラーになりました。WebLogic Server はリクエストのタイプが <code>weblogic.servlet.internal.ServletRequestImpl</code> であると予期していたためです。</p> <p>コードの修正で、この問題は解決されました。</p>
CR100572	<p>URI が正しくないリクエストをプラグインから受け取ると、WebLogic Server は次のスタックトレースを送出していました。</p> <pre data-bbox="356 1189 1243 1302">&lt;Mar 8, 2003 3:27:20 PM PST&gt; &lt;Error&gt; &lt;Socket&gt; &lt;BEA-000421&gt; &lt;Uncaught Throwable in processSockets java.lang.NullPointerException. java.lang.NullPointerException... この問題は、コードを修正して解決されました。</pre>
CR100837	<p>WebLogic Server 6.1 SP04 が、JSP ページの <code>JSP includes</code> に対して出力を誤って送っていました。</p> <p>この問題は、コードを修正して解決されました。</p>

---

---

CR102036 削除された JSP ページに最初にアクセスすると、「404 File Not Found」ではなく JspFileNotFoundException が発生しました。このページへの以降のアクセスでは適切に 404 が返されました。

削除された JSP ページが常に 404 を返すように問題を修正しました。

---

CR102446 WebLogic Server 7.0 の以前のバージョンで、JSP が管理サーバの MBeanServer 上に ModelMBean を登録するように、管理サーバ上で動作する Web アプリケーションから JSP を呼び出し、その後管理対象サーバを停止して再起動すると、サーバは次のメッセージにより失敗しました。

```
<Critical> <WebLogicServer> <000364> <Server failed during
initialization.
Exception:weblogic.management.configuration.ConfigurationExcept
ion: - with nested exception:
[java.lang.ClassCastException:
javax.management.modelmbean.ModelMBeanInfoSupport]
java.lang.ClassCastException:
javax.management.modelmbean.ModelMBeanInfoSupport at
weblogic.management.commo.Commo.createMBean(Commo.java:530) at
weblogic.management.internal.RemoteMBeanServerImpl.createCommoM
BeanLocally(RemoteMBeanServerImpl.java:941) at
weblogic.management.ManagedServerAdmin.syncToAdminServerCommo (M
anagedServerAdmin.java:406) at
weblogic.management.ManagedServerAdmin.initializeCommo (ManagedS
erverAdmin.java:444)
    at weblogic.management.Admin.start(Admin.java:307)
    at weblogic.t3.srvr.T3Srvr.initialize1(T3Srvr.java:679)
    at weblogic.t3.srvr.T3Srvr.initialize(T3Srvr.java:589)
    at weblogic.t3.srvr.T3Srvr.run(T3Srvr.java:277)
    at weblogic.Server.main(Server.java:32)
```

weblogic.management.commo.Commo.java のコードを変更して問題を解決しました。

---

CR102628

次の JSP コードでは、

```
<jsp:plugin type="applet"  
code="examples.applets.PhoneBook1.class"  
codebase="/bea_WebLogic  
Server_internal/classes/DefaultWebApp@DefaultWebApp/"  
height="800" width="500" type="applet" jreversion="1.3.1_06"  
nspluginurl="http://java.sun.com/products/plugin/1.1.3/plugin-i  
ninstall.html";  
iepluginurl="http://java.sun.com/products/plugin/1.1.3/  
jinstall-113-win32.cab#Version=1,1,3,0" >
```

以下の HTML が生成されます。

```
<embed type="application/x-java-applet;version=1.3.1_06"  
pluginspage="http://java.sun.com/products/plugin/1.1.3/plugin-i  
ninstall.html"; height="800" width="500"  
code="examples.applets.PhoneBook1.class"  
codebase="/bea_WebLogic  
Server_internal/classes/DefaultWebApp@DefaultWebApp/">
```

生成された HTML コードは Netscape で失敗しました。Netscape は SUN のサイトからではなく WebLogic Server からプラグインをダウンロードしようとしていました。

pluginspage 行を codebase 行の前に移動すると、コードは Netscape 上で正しく動作しました。

アプレット属性の順序が正しくなるようにコードを修正して、問題を解決しました。

---

CR102675 あるページで taglib ディレクトリを通じて taglib を 1 回インポートし、そのページに含まれる 2 番目のページでも同じ taglib をインポートする場合、無効な XML は次のようなブラウザ出力になります。

```
Error 500--Internal Server Error From RFC 2068 Hypertext Transfer Protocol -- HTTP/1.1:
```

```
10.5.1 500 Internal Server Error The server encountered an unexpected condition which prevented it from fulfilling the request.
```

ログ ファイルの出力は次のようになります。

```
<[ServletContext (id=287644,name=jspVal.war,context-path=/jspVal)] Servlet
```

```
failed withIOException java.io.IOException:
javax.servlet.jsp.JspException: The taglib validator
rejected the page: "Exception TestValidator XML parsing:
org.xml.sax.SAXParseException:
```

```
Attribute "xmlns:valtest" was already specified for element
"jsp:root". -- stack trace written to stderr., "...
```

```
weblogic.servlet.jsp.jsp2xml.Jsp2XmlOutputter のコードの変更により、JSP の XML 表示の生成中に、重複しているタグプレフィックスは無視されます。
```

---

CR102769 サブレットまたは JSP から別の JSP ページにリクエストを転送したときに、queryString が access.log に記録されませんでした。また、access.log は転送されたリクエストは記録しましたが、元のリクエストは記録しませんでした。RequestDispatcherImpl のコードを変更して問題を修正しました。

---

CR103289 HttpClusterServlet は POST データのセッション ID を正しく解析しませんでした。HttpClusterServlet からクラスタにリクエストを送信して、セッションを確立し、クッキーなしで POST パラメータにセッション ID を含めて 2 番目のリクエストを送信すると、サブレットはセッションを認識しません。サブレットは POST データからセッション ID を抽出できるようになりました。

---



---

CR103473      `jspc` のコマンドラインのコンパイルでパフォーマンスが改良されました。複数のコンパイラ プロセスではなく、1 つのコンパイラ プロセスが各 JSP に分岐されるようになりました。さらに、複数の JSP で使用される TLD は、JSP ごとではなく、すべての JSP で 1 回だけ解析されるようになりました。

---

CR105395      JSP がユーザ ログインなしで `Security.getCurrentUser()` を呼び出すと、次の例外が発生しました。

```
java.lang.NullPointerException at
weblogic.security.acl.Security.getCurrentUser (Security.java:301
)
```

`wlrealm null` 設定を許可して例外を送出しないようにコードを変更しました。

---

## JTA

---

### 変更要求番号    説明

---

CR088171      WebLogic Server 7.0 の以前のリリースでは、調整サーバがトランザクションの開始サーバでない場合にサーバによって開始されたトランザクションをコミットまたはロールバックすると予期しない例外が送出されていました。コミットまたはロールバックの完了を待っている従属的な開始サーバは、予期しない例外 (たいていは調整サーバの再起動から生じる `PeerGoneException`) によって割り込まれ、トランザクションのステータスが「unknown」に変更されていました。再起動した調整サーバは、トランザクションを適切に解決できませんでした。コードの変更により、調整サーバで再起動後にトランザクションを適切に解決できるようになりました。

---

---

**変更要求番号**   **説明**

---

CR091628    `javax.transaction.xa.XAResource` インタフェースでは、メソッド `setTransactionTimeout()` が定義されています。このメソッドを使用すると、トランザクション マネージャは、`XAResource` インスタンスに関する操作のために、リソース マネージャにトランザクション タイムアウトを設定できます。アプリケーションがグローバルトランザクションでリソース マネージャにアクセスするとき、WebLogic Server のトランザクション マネージャはこのメソッドを呼び出していませんでした。

以下のプロパティが、`config.xml` ファイルで `JDBCConnectionPool` オブジェクトに追加されています。

- `XASetTransactionTimeout`— トランザクション マネージャが `XAResource.start` の呼び出し前に `XAResource.setTransactionTimeout` を呼び出すかどうかを指定します。`true` または `false` を指定できます。`true` に設定すると、トランザクション マネージャは秒単位のグローバルトランザクション タイムアウトをセッション タイムアウトとして Oracle に渡します。`false` に設定すると、トランザクション マネージャは `XAResource.setTransactionTimeout` を呼び出しません。このパラメータのデフォルト値は、`false` です。
- `XATransactionTimeout`— `XASetTransactionTimeout` が `true` に設定されている場合に、`XATransactionTimeout` 値がセッション タイムアウトとして Oracle に渡されます (`OpenString` で `SetTm` を設定するのと同じ)。このパラメータを `0` に設定すると、セッション タイムアウトが秒単位のグローバルトランザクション タイムアウトに設定されます。このパラメータのデフォルト値は、`0` です。

これらのパラメータは、XA JDBC ドライバを使用してデータベース接続を作成する接続プールでのみ有効です。非 XA JDBC ドライバを使用する接続プールでは無視されます。これらのパラメータは、Administration Console では表示されません。

---

CR091882    WebLogic Server 7.0 の以前のリリースでは、トランザクションが複数の WebLogic Server インスタンスにまたがる場合に問題の発生するおそれがあります。一部のコンフィグレーションでは、トランザクションが余計な状態を積み重ねて完了までの時間が延び、トランザクションがタイムアウトになる場合があります。この問題は、コードの変更によって修正されました。

---

変更要求番号	説明
CR091925	WebLogic Server 7.0 の以前のリリースでは、例外が捕捉されない場合にクラスタ化された管理対象サーバの回復移行サービスで問題が発生するおそれがありました。その結果、クラスタの他のメンバーが動作していない場合に管理対象サーバの起動が失敗する場合があります。この問題は、コードの変更によって修正されました。
CR092301、 CR107866	WebLogic Server 7.0 の以前のリリースでは、コンフィグレーションによっては、トランザクションの調整中に <code>NullPointerException</code> 例外が発生するおそれがあります。この問題は、コードの変更によって修正されました。
CR093406	WebLogic Server 7.0 では、サーバの再起動時のトランザクション回復処理が新しく検出されたリソースのために遅延していました。WebLogic Server 7.0 SP3 では、サーバによって検出されるとすぐにリソースでトランザクション回復処理が開始されます。
CR097013	サブコーディネータサーバは、サーバの停止後に不明なトランザクションを自動的に回復せず、DBA による手動のロールバックまたはコミットが必要でした。調査の結果、サブコーディネータのコードでは、サーバの再起動後にリモートの参加コンポーネントを追跡できないことがわかりました。リモートの参加コンポーネントを正しく追跡するために、コードにチェックポイントを追加しました。
CR098273	WebLogic Server 7.0 では、トランザクション ログの古いエントリが原因で、サーバの再起動時に不要な回復のオーバーヘッドが生じることがありました。この問題は、コードを修正して解決されました。
CR098318	WebLogic Server 7.0 SP1 では、トランザクションの古い状態が、誤ってリモートサーバにトランザクションの影響を及ぼす可能性のある専用の通信スレッドと関連付けられ、その結果として <code>java.lang.IllegalStateException</code> 例外が生成される場合があります。この問題は、コードの変更によって修正されました。
CR099554	WebLogic Server 7.0 SP1 では、環境によっては、JMS メッセージが保留状態にあるサーバが停止されるか、クラッシュした場合に、保留中のメッセージがサーバの再起動時に回復されません。この問題は、コードを修正して解決されました。

---

変更要求番号	説明
CR099830	<p>JTA の移行は、トランザクションがクラスタ内の 2 つのサーバに伝播された後に <code>NullPointerException</code> または <code>ConnectException</code> で失敗する場合があります。</p> <p><code>NullPointerException</code> &lt;Feb 28, 2003 6:38:44 PM JST&gt; &lt;Warning&gt; &lt;JTA&gt; &lt;110213&gt; &lt;The activation of Transaction Recovery Service for server [serverB] fails. java.lang.NullPointerException at...&gt;</p> <p><code>ConnectException</code> &lt;2003/02/27 13:11:20:JST&gt; &lt;Warning&gt; &lt;JTA&gt; &lt;110213&gt; &lt;The activation of Transaction Recovery Service for server [server1] fails.&gt; java.rmi.ConnectException: Destination unreachable; nested exception is: java.net.ConnectException: Connection refused: connect; No available router to destination at...</p> <p>調査の結果、サーバが応答しないときの通知と、そのサーバの情報がキャッシュから削除されたときの通知でコードが競合状態を引き起こしていることが判明しました。サーバが応答しないという通知がキャッシュのクリアの後である場合、もう一方の管理対象サーバは応答しない管理対象サーバの <code>tlog</code> の位置を探そうとしたときに <code>NullPointerException</code> に遭遇します。</p> <p>コードの修正で、この問題は解決されました。</p>
CR100898	<p>トランザクションがロールバックした場合、トランザクションに参加するリソースを持たないサーバがトランザクションに参加していると、トランザクションが完了しない場合があります。トランザクションは調整を行っているサーバ上に存在し続け、ロールバックするリソースがないことをサーバに通知する再試行を続けていました。最終的に、トランザクションは破棄されました。すべての参加リソースがロールバックの通知を受けましたが、完了した後で、調整を行っているサーバ上でコールバックが処理されない場合があります。</p> <p>リソースを持たないサーバにも対応し、すべてのサーバ / リソースが通知を受けたら直ちに状態をロールバック済みに設定するよう、ロールバック非同期応答処理のロジックを変更することで、この問題は解決しました。</p>

---

変更要求番号	説明
CR101135	<p>トランザクション タイムアウト処理の過程で、準備されている各トランザクションについて、現在のサーバがそのトランザクションの従属参加コンポーネントである場合に、トランザクション マネージャはコーディネータに一方通行の RMI メッセージを送信してトランザクションのステータスを確認します。コーディネータが再起動されている場合は、JTA サブシステムがトランザクション ログからコミット レコードの読み込みを終了する前に、このステータス要求がディスパッチされることがあります。</p> <p>このケースでは、ステータス要求がディスパッチされ、トランザクション マネージャの確認されたトランザクションのマップにクエリ対象のトランザクション ID が存在するかどうかチェックされていました。tlog がまだ処理されていないので、コーディネータの再起動より前にトランザクションがコミット ポイントを過ぎていても、TM はロールバックの決定で従属参加コンポーネントに回答していました。</p> <p>このシナリオでは、トランザクションの結果が混在していました。つまり、ステータス要求を行った従属参加コンポーネントのリソースはロールバックされ、トランザクションの他のリソースはコミットされます。</p> <p>JTA サブシステムの初期化の前に <code>checkStatus()</code> リクエストが処理されないよう、コードが修正されました。これで、トランザクションの結果が混在しなくなります。</p>
CR106174	<p>トランザクションの 1 つを除いてすべての参加リソースが <code>prepare</code> コマンドに <code>XA_RDONLY</code> フラグで応答する場合に、トランザクション マネージャはコミット レコードを不必要に書き込んでいました。</p> <p>コードを変更してこの問題を解決しました。ログ レコードを書き込む前に、サーバは第 2 フェーズを必要とする参加リソースの数をチェックするようになりました。数が 1 の場合、トランザクションはコミット レコードを生成しません。</p>
CR106174	<p>トランザクションの 1 つを除いてすべてのリソースが <code>prepare</code> に <code>XA_RDONLY</code> フラグで応答する場合、トランザクション マネージャは 1 フェーズ コミットを発行して参加リソースを保留しなければなりません。ただし、このシナリオの場合、トランザクション マネージャはコミット レコードを不必要に書き込んでいました。</p> <p>ログ レコードを書き込む前に、トランザクション マネージャが第 2 フェーズを必要とする参加リソースの数をチェックし、数が 1 の場合は書き込みがスキップされるようにコードを変更しました。その修正で、問題は解決されました。</p>

---

変更要求番号	説明
CR106177	<p>リモートの <code>beforeCompletion</code> コールバックを処理するときに、トランザクション マネージャは登録されたオブジェクトを呼び出す前にトランザクション コンテキストを再開していました。静的に登録されたリソースは、トランザクションがスレッド上で再開される時に取得されませんでした。そのため <code>beforeCompletion</code> ロジックはトランザクションの一部として追加の更新を実行できなくなりました。</p> <p>トランザクション マネージャは静的に登録されたリソースを取得できるようになりました。</p>
CR106177	<p>リモートの <code>beforeCompletion</code> コールバックを処理するときに、TM は登録されたオブジェクトを呼び出す前にトランザクション コンテキストを再開します。ただし、静的に登録されたリソースは、トランザクションがスレッド上で再開される時に取得されませんでした。</p> <p>そのようなリソースの取得を実行して、<code>beforeCompletion</code> ロジックがトランザクションの一部として追加の更新を実行できるようにコードを変更しました。その修正で、問題は解決されました。</p>

---

## 操作と管理

---

変更要求番号	説明
CR098464	<p>LogManager で同期が使用される場合、スレッドでデッドロックが発生していました。</p> <p>ある程度の同期をなくすことでこの問題は修正され、デッドロックは発生しなくなりました。Eliminating some synchronization in LogManager fixed the problem. There is no more deadlock.</p>

---

## プラグイン

---

変更要求番号	説明
CR096850	<p>アプリケーション ( 負荷テスト中のアプリケーションを含む ) から iPlanet サーバへの接続が複数回試行された後、サーバがクラッシュしなくなりました。</p>

---

---

CR086224	<p>NSAPI プラグインは POST データから SESSIONID を読み込みませんでした。</p> <p>分析の結果 <code>getPreferredServersFromCookie()</code> の問題が判明しました。POST データとして <code>session.getId()</code> が保持されるときに、<code>getId()</code> に余分な文字列 <code> time</code> が含まれていました。</p> <p>この問題は、コードを修正して解決されました。</p>
CR091791	<p>バックエンド サーバから新しい動的サーバリストを受け取ると、Apache プラグインは <code>keepAliveSecs</code> を 0 にリセットしていました。その結果、タイマがプールからすべての接続を削除していました。このため、事実上、接続プールは壊されていました。</p> <p>コードの修正で、この問題は解消されました。</p>
CR092484	<p>クラスタで、動的サーバリストが Apache プラグインによって不適切に解析されていました。</p> <p>この問題は解決済みであり、サーバリストは 5.1 までバージョンを遡って正しく解析されるようになっています。</p>
CR092756	<p>バックエンド WebLogic Server から HTTP 503 の応答を受け取っても、ISAPI プラグインがフェイルオーバーしていませんでした。</p> <p>コードの修正で、WebLogic Server から HTTP 503 の応答を受け取ったときにプラグインがフェイルオーバーするようになりました。</p>
CR093530	<p><code>PathTrim</code> の値は大文字と小文字が区別され、ISAPI プラグインでは大文字と小文字が区別されないようにする必要がありました。</p> <p>コードの修正により、<code>PathTrim</code> 値で大文字と小文字が区別されないようになりました。</p>
CR093595	<p>長い URI および <code>weblogichost</code> の代わりに <code>weblogiccluster</code> を使用している場合に、Apache プラグインが予期せず停止していました。</p> <p>コードの修正により、バッファ サイズが URI の長さの限界まで増やされ、それによって問題が解決されました。</p>
CR094768	<p>クッキーの POST の解析が、ISAPI プラグインで正しく実装されていませんでした。</p> <p>ISAPI プラグインは現在は、<code>content-type</code> が <code>application/x-www-form-urlencoded</code> の場合のみ POST を解析します。</p>

---

## サービス パック 1 から 6 で解決済みの問題

---

CR096625	<p>X_WEBLOGIC_FORCE_JVMID ヘッダが、Apache プラグインによってクラスタ内のサーバインスタンスに送信されていました。</p> <p>X_WEBLOGIC_FORCE_JVMID は、サーバリストがクラスタ化されていず、現在のサーバに <code>jvmid</code> がまだない場合以外は送信するべきではありません。</p> <p>この問題は、コードを修正して解決されました。</p>
CR097132	<p>Apache プラグインを、Linux-IA64 用に作成する必要がありました。</p> <p>バイナリ ファイルとライブラリが作成されており、新しいプラグインが入手可能になっています。</p>
CR097202	<p>WebLogic Server とブラウザの間で 128 ビットのステップアップ証明書を使用するプラグインを含むコンフィグレーションでは、WebLogic 側でブラウザの暗号強度を定義する必要がありました。ブラウザには、40 ビットまたは 128 ビットの強度がありました。ブラウザの暗号強度を取得するには、次のコードを使用していました。</p> <pre>(Integer) httpRequest.getAttribute("javax.servlet.request.key-size")</pre> <p>ただし、ブラウザは 128 ビットより小さい暗号サイズを使うように設定されていましたが、返される値は常に 128 でした。</p> <p>この問題は、HTTPS_KEYSIZE で接続の強度が指定されることが原因でした。128 ビットの証明書がプラグインでは使用されていたので、ブラウザで使用される強度に関係なく 128 が返されていました。リクエストが WebLogic Server に転送されると、ブラウザの強度に応じて 40 または 128 が返されました。プラグインは、HTTPS_KEYSIZE ヘッダまたは HTTPS_SECRETKEYSIZE ヘッダを変更しません。</p> <p>このニーズを満たすために、WL-Proxy-Client-Keysize および WL-Proxy-Client-Secretkeysize という 2 つの新しいヘッダが実装されました。両方のヘッダの値とも、<code>request.getHeader()</code> で取得できます。</p>
CR103161	
CR100361	<p>READ_ERROR_FROM_CLIENT、READ_ERROR_FROM_SERVER、READ_ERROR_FROM_FILE、WRITE_ERROR_TO_FILE という新しい例外型が、NSAPI プラグイン用に追加されました。</p>
CR101428	<p>ISAPI プラグインの使用時、リクエストに複数のクッキーが含まれているときに、JSESSIONID という名前ではないが、JSESSIONID クッキーの直後にくるクッキーは取り除かれて、WebLogic Server に配信されませんでした。</p> <p>この問題は、コードを修正して解決されました。</p>

---



CR101596	<p>ssl_certchain_verify_callback メソッドで、証明書の検証が失敗した事例のエラー条件として SSLNoErr が返されていました。たとえば、CA 証明書が critical とマークされていない基本的な制約を持ち、(デフォルトの「strong」ではなく) strict 設定が有効になっている事例や、証明書が CA としてマークされていない事例などがあります。</p> <p>この問題は、このような失敗の事例で (SSLNoErr ではなく) X509CertChainInvalidErr を返すことで解決されました。</p>
CR102616	<p>NSAPI プラグインは、名前が DNS サーバからの IP アドレスのリストを返し、プラグインが WebLogicHost = 'DNS name' でコンフィグレーションされる場合に、DNS 名に対する動的な DNS ルックアップをサポートするようになりました。</p>

## RMI/RMI-IIOP

変更要求番号	説明
CR108317	<p>一部のアクセサは、ejbc 生成の IDL (Interface-Definition-Language) で正しくマッピングされていませんでした。この問題は解決済みです。</p>
CR090979	<p>また別のサーバにアクセスするサーバにアクセスするクライアントでプリンシパルと資格を設定した場合に、プリンシパルのサーバ間の資格伝播が適切ではありませんでした。</p> <p>コードの修正により、適切な資格伝播が行われるようになりました。</p>
CR090987	<p>WebLogic は、サーバから ROID を取得してそれをクライアントに返すことに失敗し、次の例外を送出していました。</p> <pre data-bbox="361 1119 1237 1229">java.rmi.UnmarshalException: failed to unmarshal class weblogic.cluster.replication.ROID; nested exception is: java.rmi.UnmarshalException: null; nested exception is: org.omg.CORBA.MARSHAL vmcid: 0x0 minor code: 0 completed: No</pre> <p>コードの修正により、クラスは正しくマーシャリングされます。</p>
CR093107	<p>CosNaming で使用されるクラスタ化された参照に実際の参照が 1 つ含まれており、IIOP のフェイルオーバー処理が JNDI レベルで正しく機能しなくなっています。参照が JNDI にバインドされていませんでした。コードを修正してこの問題を解決しました。</p>
CR093561	<p>コードの修正で、クラスタ対応オブジェクトの置換が改善されました。</p>

---

変更要求番号	説明
CR094795	<p>ファット クライアントはステートレス EJB に接続して、Vector を拡張するクラスを返すメソッドを呼び出します。Vector を拡張するクラスは、その独自のシリアライゼーションを使用してシリアライゼーションの最適化を向上させます。ファット クライアントは、この Vector クラスの内容を表示します。RMI-IIOP がプロトコルとして使用される場合、Vector は空です。</p> <p>分析の結果、writeReplace の呼び出しが遅すぎることが判明しました。リポジトリ ID がすでに書き込まれているので、解決すべきクラスではなく実際のクラスが読み込まれていました。</p> <p>コードの修正により、Externalizable クラスで writeReplace が正しく呼び出されます。</p>
CR096476	<p>MBean プロパティ Server.IIOP.IdleConnectionTimeout は、接続が閉じるまでにアイドル状態でいられる時間をコンフィグレーションします。WebLogic Server には、同様のクライアントサイド プロパティがありませんでした。</p> <p>コードの修正により、アイドル状態のタイムアウトと保留のタイムアウトが実装されました。保留のタイムアウトは、DGC の期間の長さで数で管理します。</p> <p>アイドル時間のタイムアウトを 100 秒 (デフォルトは 240 秒) に設定するには、次のようにします。</p> <pre data-bbox="337 869 901 895">-Dweblogic.iiop.IdleConnectionTimeout=100</pre> <p>保留のタイムアウトを 5 分 (デフォルトは 60 秒) に設定するには、次のようにします。</p> <pre data-bbox="337 973 874 999">-Dweblogic.DGCIdlePeriodsUntilTimeout=5</pre> <p>期間の長さを設定するには、次のようにします。</p> <pre data-bbox="337 1052 861 1078">-Dweblogic.PeriodLength=120000 (for 120 s)</pre> <p>接続は、その接続のアイドル期間が経過していない場合にタイムアウトになります。保留のタイムアウトは常にアイドル状態のタイムアウトより長いものと想定されます。保留のタイムアウトは、すでに応答を待っている発信接続で使用します。</p>
CR098713	<p>WebLogic Server は IIOP valuetype の発信チャンキングをサポートしていません。つまり、Externalizable クラスへの変更をサポートしていませんでした。</p> <p>コードの修正により、発信チャンキングのサポートが実装されました。</p>

---

変更要求番号	説明
CR099693	<p>7.0 (G.A.、SP01、および SP02) では、クライアントが t3 プロトコルを使用してファイアウォールから WebLogic Server にアクセスすると、サーバは初期化されていない接続上でメッセージをルーティングしようとするため、クライアントはエラーを受け取りました。</p> <p>サーバの <code>ExternalDNSName</code> が指定されました。初期接続時に、クライアントはサーバインスタンスの <code>ExternalDNSName</code> へリクエストを送信しました。サーバインスタンスからの応答にある JVMID は、サーバの内部 IP アドレスと内部 DNS 名を必要としています。クライアントからの以降のリクエストは認識されませんでした。</p> <p>コードの修正で、この問題は解決されました。</p>
CR100480	<p>コードの修正によって、複数のスレッドで競合する認証が実行されている場合に、IIOP クライアントのコンテキスト ID (シンクライアントを含む) が正しく維持されるようになりました。</p>
CR100831	<p>クライアントがあるクラスタ内のサーバ上にあり、別のクラスタ内のサーバのレプリカ対応スタブを使用している場合、ローカルサーバはレプリカリストに載ることができず、そのために優先ホストが無視されていました。</p> <p>この問題は、ローカルアフィニティ (リモートクラスタの呼び出し時に取得されない) をチェックするようにコードを修正することで解決されました。ローカルアフィニティを取得できない場合は、優先ホストがレプリカリストと照合されず、レプリカリストに一致がない場合にのみ、ホストはロードバランシングアルゴリズムを使用してリストから選択されることとなります。</p>
CR101186	<p>バグが原因で、IIOP 経由での例外の相互運用性が実現されていませんでした。具体的には、JDK 1.4.x で動作するサーバ (つまり 8.1) で送出された例外が、JDK 1.3.1 で動作する 7.0.x サーバで理解できていませんでした。これは、MARSHAL 例外です。2 つのバグがありました。1 つはシーケンスの処理に、もう 1 つはクラスが FVD (FullValueDescription) で記述される方法にありました。現在は、IIOP 経由で例外の相互運用性が実現されています。</p>
CR107137	<p>CORBA および IIOP を介したクライアント / サーバアクセスで、マーシャリングエラーとインダイレクションエラーが生成されました。</p> <p>コードを変更して、インダイレクション中の <code>hashCodes</code> の生成を修正しました。IndirectionWrapper についても修正しました。</p>

## セキュリティ

変更要求番号	説明
CR080488、 CR091569	<p>ユーザ / グループ管理のセキュリティ サービス プログラミング インタフェース (SSPI) をサポートするサンプル カスタム認証プロバイダが、dev2dev のサンプル セキュリティ プロバイダのコード例に含まれていました。また、カスタム監査プロバイダもコード例に追加されました。</p> <p>詳細については、<a href="http://dev2dev/codelibrary/code/security_prov81.jsp">http://dev2dev/codelibrary/code/security_prov81.jsp</a> を参照してください。</p>
CR084874	<p>2 つの信頼性のあるドメインの中で、Java クライアントがドメイン 1 のサーバ 1 のステートレス セッション Bean 1 を呼び出します。この Bean は、ドメイン 2 のサーバ 2 のステートレス セッション Bean 2 を呼び出します。最初のサーバを呼び出すユーザは、ロール Admin (デフォルト) を持つグループ Administrators のユーザ system です。</p> <p>Bean 1 から Bean 2 への呼び出しで t3 プロトコルが使用されている場合、プリンシパルは適切に伝播されていました。</p> <p>Bean 1 から Bean 2 への呼び出しが IIOP 呼び出しで、ロール Admin をユーザ system にマッピングするセキュリティ ロールの割り当てが weblogic-ejb-jar.xml にある場合、プリンシパルは適切に伝播されていました。</p> <p>ただし、Bean 1 から Bean 2 への呼び出しが IIOP 呼び出しで、セキュリティ ロールの割り当てでロール Admin がグループ Administrators にマッピングされている場合は、org.omg.CORBA.NO_PERMISSION エラーが発生していました。</p> <p>プリンシパルの伝播は、現在はあらゆる状況で適切に行われます。</p>

---

変更要求番号	説明
CR085094	<p>1つの管理サーバと1つの管理対象サーバを含むドメインを作成した後で、2つのサーバに対して <code>-Djava.security.manager</code> を有効にすると <code>java.io.Filepermissions</code> エラーが返されて、AdminServer ログで <code>&lt;Error&gt;</code> <code>&lt;EmbeddedLDAP&gt;</code> <code>&lt;000000&gt;</code> <code>&lt;Error parsing Entry #19: access denied (java.io.FilePermission .\myserver\ldap\ldapfiles\EmbeddedLDAP.data read)&gt;</code> と報告されました。</p> <p>この問題は、<code>weblogic.policy</code> ファイルを更新することで解決されました。ファイルには、ユーザアプリケーションへのパーミッションの付与に関する情報と、<code>grant</code> 文のサンプルが追加されました。<code>weblogic.policy</code> は WebLogic Server Examples、Pet Store、および Workshop に対して機能します。ユーザコンフィグレーションで使用するには、変更が必要になります。手順はファイル上部のコメントに示されています。</p>
CR090101	<p>WebLogic Server は証明書チェーンの各証明書が認証局によって発行されたことを保証しませんでした。この問題は、誰かが信頼性のある認証局から個人用証明書を取得し、その証明書を使用して他の証明書を発行しても、WebLogic Server は無効な証明書を検出できないことを意味しました。このリリースでは、WebLogic Server で使用するすべての X509 V3 CA 証明書は CA として定義される基本制約拡張を備えている必要があります。このため、証明書チェーンのすべての証明書が認証局によって発行されたことが保証されます。デフォルトでは、この条件を満たしていない認証局の証明書は拒否されます。証明書検証に合格しない証明書チェーンを持つ WebLogic Server が起動された場合、クライアントでそれを拒否できることを示す情報メッセージがログに記録されます。</p> <p>詳細については、「<a href="#">SSL 証明書検証</a>」を参照してください。</p>
CR090147	<p>Active Directory LDAP 認証プロバイダを使用している場合に、ユーザの検索フィルタがユーザ名での「\」の使用をサポートしていませんでした。LDAPv3 の属性構文の定義の仕様では、ユーザ検索フィルタでの「\」の使用がサポートされています。「\」の使用がサポートされるように、ユーザ検索フィルタのコードが更新されています。</p>

---

変更要求番号	説明
CR090218	<p>メソッド <code>getCurrentUser</code> と <code>getCurrentSubject</code> は、現在のスレッドに関連付けられたユーザのユーザ ID を返します。WebLogic Server 7.0 の以前のサービスパックでは、<code>RequestDispatcher.forward</code> を使用してあるサーブレットから別のサーブレットに転送した場合、これらのメソッドは <code>null</code> (または空の文字列) を返しました。</p> <p>1 つの JSP またはサーブレットから別の JSP またはサーブレットへ転送されるリクエストが、セッションで認証されたユーザを送信するようにコードを修正して、この問題を解決しました。</p>
CR091972	<p>WebLogic Server の以前のリリースでは、SSL のライセンス チェックで 128 ビットとライセンスを使用した IP アドレスが必要とされていました。ライセンス チェックの元々の目的は以下のようなものでした。</p> <ul style="list-style-type: none"><li>• WebLogic Server の実行時を除いて、顧客が Certicom SSL ランタイムを使用できないようにするため。これは、BEA System が Certicom と結んだ契約の 1 つの条件でした。</li><li>• SSL が使用できる強度を決めるため (国内または輸出)。これは政府の規制でした。</li></ul> <p>BEA では、サーバ上で動作していない Java クライアントがサーバから Java クライアントマシンにライセンスをコピーすることを推奨していました。しかし、SSL コードのライセンス チェックの実装は IP の検証を要求していました。このため、サーバの SSL ライセンスはクライアントマシンで使用することができませんでした。ライセンス チェックの IP 検証は現在は廃止になっています。現在は、ライセンスをサーバから Java クライアントマシンにコピーすることができます。</p>
CR092236	<p>Active Directory LDAP 認証プロバイダを使用している場合に、ユーザ検索フィルタがユーザ名での自然言語文字の使用をサポートしていませんでした。LDAPv3 の属性構文の定義の仕様では、ユーザ検索フィルタでの自然言語文字の使用がサポートされています。自然言語文字の使用がサポートされるように、ユーザ検索フィルタのコードが更新されています。</p>
CR093533	<p>互換性レルムのレルム アダプタ認証プロバイダは、ユニークな ID で User オブジェクトを作成するのではなく、User オブジェクトの認証済みユーザ名を使用していました。この問題により、単純認証 (ユーザ名とパスワード) を実行するカスタム セキュリティ レルムとの下位互換性が壊されていました。</p> <p>レルム アダプタ認証プロバイダのコードが、ユニークな ID で User オブジェクトが作成されるように修正されています。</p>

変更要求番号	説明
CR093813	<p>ノード マネージャ キー ファイル <code>weblogic.nodemanager.keyPassword</code> の暗号化されたプライベート キーにアクセスするために使用するパスワードが、プレーン テキストでアクセス可能なものでした。</p> <p>ノード マネージャ プロパティ ファイル <code>nodemanager.properties</code> を作成して、問題を解決しました。</p>
CR094038	<p>WebLogic Security フレームワーク内のリソース エンジンで、階層キャッシングの最適化が廃止されています。リソース キャッシュは、リソースをキーとして使用するよう更新されています。キャッシュは、各キャッシュ ヒットで <code>resource.equals()</code> を実行して、ハッシュ コードの衝突を防止するようになりました。リソースの文字列表現と親がリソースにキャッシュされるので、階層がキャッシュされた後は、キャッシュを 1 回ルックアップするだけで、リソースのパス名を取得できます。この変更で、WebLogic Server のパフォーマンスが向上します。</p>
CR094215	<p><code>weblogic.security.ntrealm.NTRealm username password</code> を呼び出すと、常に例外が送出されていました。この問題は解決済みです。</p>
CR094548	<p>WebLogic Server 7.0 の以前のバージョンでは、2 つのグループがネストされていることを WebLogic Server に認識させるには、LDAP 内のグループを互いに明示的に追加する必要がありました。LDAP ツリー内でグループを別のグループに追加して暗黙的なネストを行うと、認識されませんでした。</p> <p>階層によるグループ メンバーシップ チェックを実行するように LDAP プロバイダを変更して、この問題を解決しました。次のようにプロパティを設定します。</p> <pre>-Dweblogic.security.hierarchyGroupMemberships="true"</pre>
CR095689	<p>双方向 SSL と</p> <p><code>weblogic.servlet.security.ServletAuthentication.strong()</code> メソッドを使用すると、例外が発生していました。サーブレット コードの認証部分の <code>import</code> が、サーブレット仕様で定義されている <code>java.security.cert.X509Certificate</code> ではなく <code>javax.security.cert.X509Certificate</code> を使用していました。サーブレットのコードでは、<code>java.security.cert.X509Certificate</code> が使用されるようになりました。</p>

変更要求番号	説明
CR095789	<p>ユーザが非常に多いグループを扱うときに、LDAP レルム v2 のパフォーマンスが低下していました。この問題が原因で、WebLogic Portal サーバがハングしてタイムアウトになる可能性があります。この問題は、次のメソッドの呼び出しに関連していました。</p> <pre data-bbox="297 392 919 444">netscape.ldap.LDAPMessageQueue.waitForMessage (LDAPMessageQueue.java:179)</pre> <p>具体的には、groupDN の多くのグループが LDAP レルム v2 で検索されており、それらすべてのグループで Group.isMember メソッドが連続して呼び出された場合にその呼び出しがハングします。通常、サーバは数百の呼び出しが連続して行われた後にハングします。</p> <p>LDAP レルム v2 のコードがこの問題を解決するために更新されており、ユーザの多いグループを扱うときにパフォーマンスが向上しています。</p>
CR095836	<p>Administration Console の [ドメイン   Domain Wide Security Settings   組み込み LDAP] で、バックアップ分数のデフォルト値が正しく計算されて報告されるようになりました。</p>
CR096488	<p>RDBMS セキュリティ レルムを使用する 6.1 セキュリティ コンフィグレーションを使用すると、サーバが (したがって互換性セキュリティも同様に) 起動しません。</p> <ol data-bbox="297 899 1174 1359" style="list-style-type: none"><li>1. RDBMS レルムの config.xml エントリでは、RealmClassName、DatabaseDriver、および DatabaseURL 属性を明示的に示す必要があります。6.x で RDBMS セキュリティ レルムを使用する場合、それらの値は RDBMS コード サンプルとサンプル データベースが使用されているものと想定していました。</li><li>2. WebLogic Server バージョン 7.0 で使用するためには、HimselfAAA のコードを再コンパイルする必要があります。WebLogic Server 6.x では、RDBMS コード サンプルでプライベートな Admin API がエクスポートされていました。この問題は確認済みです。詳細については、「<a href="#">セキュリティのアップグレード</a>」を参照してください。</li><li>3. セキュリティ レルムをロードする Weblogic Server のコードは、RealmClassName 属性が空である場合の適切な対処を行っていませんでした。このコードはすでに更新されており、レルム クラス名を指定する必要があることを明確に示す例外が送出されます。</li></ol>



変更要求番号	説明
CR096589	<p><code>weblogic.servlet.security.ServletAuthentication.strong()</code> はクラスキャスト例外により失敗しました。</p> <p>調査の結果、間違った証明書クラスをインポートしていたことがわかりました。コードの修正で、この問題は解決されました。</p>
CR096934	<p><code>WebLogicBeanMaker</code> は <code>AuthenticationMBI</code> クラス コンストラクタによって生成された例外を認識していなかったため、<code>AuthenticationProvider</code> の構築でエラーが認識されませんでした。</p> <p>エラーの伝播を確実にするようにコードを変更して、問題を解決しました。</p>
CR098083	<p><code>WebLogic Server</code> の過去のリリースでは、有効性を証明できないパスワードを受信したときに <code>WebLogic</code> 資格マッピングから通知が出力されていませんでした。この問題は解決済みです。<code>WebLogic</code> 資格マッピング プロバイダは、有効性を証明できない暗号化パスワードを受信したときに例外を送出するようになりました。</p>
CR098084	<p><code>WebLogic Server</code> の過去のリリースでは、<code>WebLogic</code> 資格マッピング プロバイダによって組み込み LDAP サーバに格納されたパスワードがクリア テキストでした。<code>WebLogic</code> 資格マッピング プロバイダは、組み込み LDAP サーバに格納するときにパスワードを暗号化するようになりました。</p>
CR098242	<p>ノード マネージャでは、ユーザの認可レベルに関係なく、クリアされたユーザ名とパスワードが提供されるため、管理対象サーバを認可なしで起動することができました。</p> <p>ノード マネージャはユーザに対するセキュリティ チェックを実行するようになりました。</p>
CR100174	<p><code>Bouncy Castle JCE</code> プロバイダは 7.0 SP3 より前ではテストされておらず、SSL と一緒に機能するには特別なコードが必要です。このリリースでは、<code>BouncyCastle</code> プロバイダ JCE がプラグインされた時点で検出し、代わりに内部の暗号化方式を使用するように処理するコードを SSL に追加しました。サーバサイド アプリケーションで <code>BouncyCastle JCE</code> プロバイダを使用しても、例外が発生しなくなりました。プラグインされても SSL はそのプロバイダを使用しませんが、例外は発生しません。</p>

---

変更要求番号	説明
CR100592	<p>サポートされている LDAP サーバの 1 つでユーザ アカウントが無効になった場合 (たとえばアカウントが長い期間無効になっているか、障害が原因でロックされている場合)、LDAP セキュリティ レalm からそのアカウント (ユーザ名 / パスワード) を使用するためには、サーバを再起動する必要があります。次の例外が送出されました。</p> <pre data-bbox="297 430 1150 482">&lt;[WebAppServletContext (2052089, WebCM-Outages, /Webbed-Outages)] Servlet failed with Exception netscape.ldap.LDAPException</pre> <p>この問題は解決済みです。</p>
CR100762	<p>WebLogic Server の以前のリリースでは、実行時に SunJCE プロバイダをアンロードするコードがまだ使用されていました。この問題により、SSL プロトコルがサーバインスタンスで有効になっている場合に WebLogic Server で Sun JCE プロバイダを使用することができませんでした。</p> <p>この問題は解決済みです。</p> <p>JDK 1.4 クライアントと Sun JCE を使用している顧客は、JDK 1.4 JCE の実装のパフォーマンスの問題に起因するパフォーマンスの低下を経験する場合があります。クライアントが Sun JCE プロバイダを必要としない場合は、次のように JDK/re/lib/security/java.security ファイルから Syncs プロバイダをコメントアウトします。</p> <pre data-bbox="297 890 1085 1029">security.provider.1=sun.security.provider.Sun security.provider.2=com.sun.net.ssl.internal.ssl.Provider security.provider.3=com.sun.rsajca.Provider #security.provider.4=com.sun.crypto.provider.SunJCE #security.provider.5=sun.security.jgss.SunProvider</pre>
CR100797	<p>WebLogic Server のデフォルト接続フィルタが HTTPS プロトコルを適切に処理していませんでした。この問題は解決済みです。</p>

---

変更要求番号	説明
CR101652	<p>WebLogic Server の以前のリリースでは、XML サーバが SSL プロトコルをサポートしていなかったためサードパーティ XML サーバとのセキュアな接続を確立できませんでした。</p> <p>次の回避策を利用できます。</p> <ol style="list-style-type: none"><li>1. WebLogic クライアント (WebLogic Server にデプロイされた JavaServer Page (JSP)) を作成します。</li><li>2. Java の仕様に従って Java 認証局クラスを作成します。この認証局は、XML サーバに接続するために使用する名字とパスワードを取得する必要があります。</li><li>3. <code>weblogic.policy</code> ファイルで Java 認証局クラスのネット パーミッションを指定します。ネット パーミッションは、Java VM のすべてのユーザに与えられます。これで潜在的なセキュリティの弱さが生じるので、慎重に使用する必要があります。</li><li>4. <code>java.net.HttpConnection</code> を使用して、XML サーバとの接続を確立します。</li><li>5. WebLogic Server の起動時に <code>-DUseSunHttpHandler</code> をコマンドライン引数として指定します。</li></ol>
CR102351	<p>Eracom および Entrust サードパーティ セキュリティ プロバイダはこのリリースで動作確認されました。</p>
CR102351	<p>7.0 SP2 では SSL 暗号化処理に JCE プロバイダを使用できるようになりました。ただし、すべての JCE プロバイダが同じように機能するわけではありません。各プロバイダでは、特別な処理と予測が必要です。ERACOM/Entrust JCE プロバイダが SSL と共に機能するには、特別なコードが必要です。</p> <p>コードを SSL に追加して、ERACOM/Entrust プロバイダがプラグインされた時点で検出し、内部の暗号化方式を使用するようにしました。ERACOM/Entrust JCE プロバイダはサーバサイド アプリケーションで機能するようになりました。</p>
CR105809	<p><a href="http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA03-33.jsp">http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA03-33.jsp</a> のセキュリティ勧告に関する情報を確認してください。</p>
CR106027	<p><a href="http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA03-28.01.jsp">http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA03-28.01.jsp</a> のセキュリティ勧告に関する情報を確認してください。</p>
CR106357	<p>複数の認可プロバイダをインストールした場合、Web アプリケーションに対してポリシーを定義できませんでした。</p> <p>この問題は修正済みです。</p>

## システム管理

変更要求番号	説明
CR063279	WebLogic Server の config.xml 内の複数の要素に同じオブジェクト名がある場合、後の方の要素が前の要素のコンフィグレーションを上書きします。これは通知されずに行われていました。WebLogic Server は標準出力にエラー メッセージ (ID=150012) を送出するようになりました。
CR082422	Unix では、新しくデプロイされたファイルは、ルート ディレクトリに移動してからそのファイルの親ディレクトリに移動しない限り、ファイル リストまたはディレクトリ リストに表示されませんでした。 file_chooser.jsp のコードを変更して、この問題を解決しました。
CR091205	WebLogic Server SNMP MIB では、applicationRuntimeIndex と cacheMonitorRuntimeIndex で OID (1.3.6.1.4.1.140.625.105.1.1) が重複していました。cacheMonitorRuntime は現在はユニークな OID (1.3.6.1.4.1.140.625.10) を使用しています。
CR091224	EJBComponentRuntimeMBean.getEJBRuntimes() を呼び出すと、ASSERTION FAILED エラーに続いて ArrayStoreException が生じる問題が修正されました。
CR092877	削除された管理対象サーバの ServerLifecycleRuntimeMBean と ServerConfigMBeans は、管理サーバが再起動されるまで存在し続けました。管理対象サーバを削除するとこれらの MBean が削除されるようになりました。
CR092946	SNMP を使用し、管理サーバを起動するときに次の例外が発生するバグを修正しました。java.lang.ClassNotFoundException: weblogic.management.snmp.agent.VirtualHost
CR093481	WebLogic Server の起動時に weblogic.Domain と weblogic.apache.xerces.maxentityrefs が両方とも設定される場合に不要なエラー メッセージを表示していた問題が修正されました。
CR093816	動的 MBean の使用時に NPE を生じさせる問題が修正されました。

変更要求番号	説明
CR093824	<p>ドメイン全体の管理ポートを使用している場合に次の例外を生じさせていた、管理サーバの再起動時の問題が修正されました。この例外はドメイン全体の管理ポートが使用されていない場合には発生せず、ドメインが最初に起動されるときにも例外は送出されませんでした。</p> <pre>&lt;Emergency&gt; &lt;Configuration Management&gt; &lt;150009&gt; &lt;Errors detected attempting to connect to admin server ...</pre>
CR094966	<p>管理対象サーバが動作している状態で管理サーバを再起動させた後に、アプリケーションのデプロイメントを失敗させていた問題が修正されました。</p>
CR096544	<p>HP-UX で、WebLogic Server の以前のバージョンから 7.0 に以降するときに、OutOfMemory エラーで失敗することがありました。</p> <p>この問題は、Ant スクリプト内にハードコード化された JDK への参照によって発生しました。</p> <p>参照を変数に置き換えて、問題を修正しました。</p>
CR096726	<p>MBeanHome.deleteMBean() の呼び出し時に NPE を生じさせていた問題が修正されました。</p>
CR097071	<p>システム管理サブシステムの SUID の非互換性が修正されました。</p>
CR099002	<p>T3S プロトコル用のバインディングが新しく実装されました。</p>

---

**変更要求番号**   **説明**

---

CR099474   RequiredModelMBean の `getAttributes()` は、`javax.management.Attribute` オブジェクトを含む `javax.management.AttributeList` を返しますが、WebLogic Server 7.0 の JMX 実装では、RequiredModelMBean によって返される `AttributeList` には、`javax.management.Attribute` オブジェクトとしてではなく、MBean 属性の値が直接含まれていて、次の例外が発生しました。

```
java.lang.ClassCastException: java.lang.String at
weblogic.management.internal.RemoteMBeanServerImpl.getAttributes(RemoteMBeanServerImpl.java:210) at
weblogic.management.internal.RemoteMBeanServerImpl_WebLogicServerkel.invoke(Unknown Source)

    at
weblogic.rmi.internal.BasicServerRef.invoke(BasicServerRef.java:362) at
weblogic.rmi.internal.BasicServerRef$1.run(BasicServerRef.java:313) at
weblogic.security.service.SecurityServiceManager.runAs(SecurityServiceManager.java:821) at
weblogic.rmi.internal.BasicServerRef.handleRequest(BasicServerRef.java:308) at
weblogic.rmi.internal.BasicExecuteRequest.execute(BasicExecuteRequest.java:30) at
weblogic.kernel.ExecuteThread.execute(ExecuteThread.java:213)
at weblogic.kernel.ExecuteThread.run(ExecuteThread.java:189) End
server side stack trace

at
com.adventnet.beasupport.client.JMXConverter.convertInvocationTargetException(JMXConverter.java:744) at
com.adventnet.beasupport.client.ProxyMBeanServer.getAttributes(ProxyMBeanServer.java:511) at
com.adventnet.beasupport.client.WebLogicClient.getAttributes(WebLogicClient.java:593) at
com.adventnet.beasupport.client.WebLogicClient.cmdline(WebLogicClient.java:1279) at
com.adventnet.beasupport.client.WebLogicClient.access$000(WebLogicClient.java:15) at
com.adventnet.beasupport.client.WebLogicClient$2.run(WebLogicClient.java:1293)

GetAttributes メソッドは Attribute オブジェクトを返すようになりました。
```

---

---

**変更要求番号**   **説明**

---

CR100756      weblogic.Admin コーティリティから JMSServer の Store プロパティを次のように設定すると、

```
java weblogic.Admin -url t3://localhost:7001 -username system
-passwd weblogic SET -mbean
examples:Name=examplesJMSServer,Type=JMSServer -property Store
examples:Name=examplesJMSFileStore,Type=JMSStore
```

次の例外が送出されました。

```
<Mar 10, 2003 3:32:15 PM EST> <Error> <Management> <141033>
<Error importing MBean
examples:Name=examplesJMSFileStore,Type=JMSStore to server
examplesServer.javax.management.InstanceNotFoundException:
examples:Name=examplesJMSFileStore,Type=JMSStore
```

コンフィグレーション MBean の WebLogicObjectName の属性を設定するときにチェックすることで、この問題は部分的に修正されました。コードでは、Bean が有効なものかどうかをチェックして、InvalidAttributeValueException を送出します。これは管理サーバのログに記録され、JMSServer は使用可能なままですが、ファイルストアは設定されません。

---

---

**変更要求番号**   **説明**

---

CR101738      管理対象サーバは次のスタック トレースにより失敗しました。

```
java.rmi.ConnectException: This RJVM has already been shutdown
-1425673232662082784S:WebLogic
ServerDEBUG1: [53101,53101,53102,53102,53101,53102,-1]:WebLogic
Servere1:ADMIN Start server side stack trace:
java.rmi.ConnectException: This RJVM has already been shutdown
-1425673232662082784S:WebLogic
ServerDEBUG1: [53101,53101,53102,53102,53101,53102,-1]:WebLogic
Servere1:ADMIN at
weblogic.rjvm.RJVMImpl.getOutputStream(RJVMImpl.java(Compiled
Code)) at
weblogic.rjvm.RJVMImpl.getRequestStream(RJVMImpl.java(Compiled
Code)) at
weblogic.rmi.internal.BasicRemoteRef.getOutboundRequest(BasicRe
moteRef.java(Compiled Code)) at
weblogic.rmi.internal.BasicRemoteRef.invoke(BasicRemoteRef.java
(Compiled Code)) at
weblogic.management.internal.AdminMBeanHomeImpl_WebLogic
Serververtub.getMBean(Unknown Source) at
weblogic.management.deploy.slave.SlaveDeployer.getTargetVirtual
Hosts(SlaveDeployer.java(Compiled Code)) at
weblogic.management.deploy.slave.SlaveDeployer.getCompsFromTask
(SlaveDeployer.java(Compiled Code)) at
weblogic.management.deploy.slave.SlaveDeployer.processRemoveOrD
eactivate(SlaveDeployer.java:1240) at
weblogic.management.deploy.slave.SlaveDeployer.commitUpdate(Sla
veDeployer.java:1533) at
weblogic.drs.internal.SlaveCallbackHandler$2.execute(SlaveCallb
ackHandler.java:34) at
weblogic.kernel.ExecuteThread.execute(ExecuteThread.java(Compil
ed Code)) at..
```

調査の結果、管理対象サーバ上で動作するアプリケーション スレーブ デプロイヤーは、初期化されたときに、管理サーバの Admin MBeanHome をロックアップしてキャッシュし、使用していたことがわかりました。

管理サーバの Admin MBeanHome を、スレーブ デプロイヤーの初期化時にキャッシュするのではなく、必要なときにロックアップするようにコードを変更しました。その修正で、問題は解決されました。

---



変更要求番号	説明
CR101856	<p>クラスタに 1 つの管理サーバと 2 つの管理対象サーバが含まれていました。Application1.ear に EJB と Web アプリケーションが含まれていました。EJB はクラスタにデプロイされ、Web アプリケーションは仮想ホストにデプロイされて、その仮想ホストはクラスタに割り当てられていました。Application1.ear には、仮想ホストのデフォルト Web アプリケーションとしてコンフィグレーションされた Application1.war が含まれていました。さらに、各クラスタ ノードには独自のデフォルト Web アプリケーションがありました。</p> <p>Application1.ear をアンデプロイすると、次の例外が発生しました。</p> <pre>deactivating application application1 on server2 deactivating application application1 on server1 deactivated application application1 on server2 unpreparing application application1 on server2 deactivated application application1 on server1 unpreparing application application1 on server1 failed application application1 on server2 failed application application1 on server1 Exception caught for task Unprepare application application1 on mycluster,application1: Start server side stack trace: java.lang.reflect.UndeclaredThrowableException: java.lang.reflect.InvocationTargetException: javax.management.ListenerNotFoundException: listener: ServletContext (id=559103557,name=DefaultWebApp,context-path=)</pre> <p>アプリケーションは、ドメイン全体を再起動しても再デプロイできませんでした。調査の結果、Application Mbean unstageTargets() メソッドが、仮想ホストのアンステージングで不正なインデックスを使用していることがわかりました。コードの修正で、この問題は解決されました。</p>
CR102149	<p>Solaris 上で、トランザクション セッションで同期的に 21,000 の永続化されたメッセージを受信すると、WebLogic Server インスタンスは OutOfMemoryError になりました。</p> <p>解放される長いリストを再要求するときに、ガベージ コレクタの速度が低下しました。リンクされたリストを分割して、問題を修正しました。</p>
CR105736	<p>7.0 で SessionID のフォーマットが変更されたため、Cisco Loadbalancer などの一部のロードバランシング アプリケーションはセッションの維持を使用できなくなりました。</p> <p>新しいサーバ起動フラグ</p> <p>-Dweblogic.servlet.useExtendedSessionFormat=true によって、ロードバランシング アプリケーションでセッションの維持に必要な情報を保持します。</p>

## サービス パック 1 から 6 で解決済みの問題

変更要求番号	説明
CR105777	<p>RemoteException が迅速なキャッシングの valuetype として扱われないため、例外の伝播が正しく機能していませんでした。</p> <p>実際の valuetype のみをキャッシュすることで問題を修正しました。</p>
CR105831	<p>展開された Web アプリケーションが docroot からクラスをロードできました。</p> <p>この不適切なクラスロードの動作は無効になりました。</p>
CR106198	<p>GroupReaderMBean で、メンバー名にカンマが含まれているときに isMember が失敗しました。</p> <p>カンマを含むメンバー名を許可するようにコードを変更して、この問題を解決しました。</p>

---

変更要求番号	説明
CR106687	<p>Examples サーバ上のリンクがローカル ファイルを指している場合、ユーザがリモートのブラウザからそのリンクにアクセスしても機能しません。</p> <p>リモートユーザがサンプルのドキュメントを参照できるように、examplesWebApp の weblogic.xml に virtual-mapping を追加して、問題を修正しました。</p> <p>アクティブ化するには、&lt;virtual-directory-mapping&gt; の &lt;local-path&gt; 値をサンプルのインストール場所 (d:/bea/weblogic/samples/server/src など) で置き換えて、weblogic.xml の virtual-mapping を正しくコンフィグレーションします。</p>
CR107043	<p>bootclasspath に weblogic.jar を指定して WebLogic Server で t3 接続を確立すると、次の例外が発生しました。</p> <pre data-bbox="360 661 1231 1397">java.lang.NullPointerException at weblogic.rmi.internal.StubGenerator.getStubClass(StubGenerator. java:632) at weblogic.rmi.internal.StubGenerator.generateStub(StubGenerator. java:656) at weblogic.rmi.internal.StubGenerator.generateStub(StubGenerator. java:651) at weblogic.rmi.extensions.StubFactory.getStub(StubFactory.java:59 ) at weblogic.jndi.WLInitialContextFactoryDelegate.newRootNamingNode Stub(WLInitialContextFactoryDelegate.java:489) at weblogic.jndi.WLInitialContextFactoryDelegate.newRemoteContext( WLInitialContextFactoryDelegate.java:452) at weblogic.jndi.WLInitialContextFactoryDelegate.newContext(WLInit ialContextFactoryDelegate.java:372) at weblogic.jndi.WLInitialContextFactoryDelegate.getInitialContext (WLInitialContextFactoryDelegate.java:339) at weblogic.jndi.WLInitialContextFactoryDelegate.getInitialContext (WLInitialContextFactoryDelegate.java:221) at weblogic.jndi.WLInitialContextFactory.getInitialContext(WLIniti alContextFactory.java:149) at javax.naming.spi.NamingManager.getInitialContext(NamingManager. java:671) at javax.naming.InitialContext.getDefaultInitCtx(InitialContext.ja va:242) at javax.naming.InitialContext.init(InitialContext.java:218)</pre> <p>スタブの生成に引数として指定できるクラスローダを使用するようにコードを変更して、問題を修正しました。</p>

---

## ツール

変更要求番号	説明
CR074714	WebLogic Builder の [WebLogic 設定] パネルで、RAR の [ 増加容量 ] が [ 最大容量 ] と [ 初期容量 ] の差より大きい場合に、WebLogic Builder は RAR のロードに失敗し、「weblogic.xml.process.: capacity-increment should be less than or equal to ( max_capacity-initial_capacity)」で始まる SAXProcessorException を出力していました。 コードを変更してこの問題を解決しました。
CR072039	[EJB Relations] パネルに示された関連が、名前のアルファベット順にソートされるようになりました。
CR073975	メソッド名とトランザクション属性が、新しいメソッドを作成する上で必須の情報になりました。
CR074198	Bean クラスのメソッドが、[Add method] ダイアログの [Method name] 選択リストに表示されなくなりました。ホーム、ローカル ホーム、リモート インタフェース、およびローカル インタフェースのメソッドは表示されます。
CR074591	デプロイメント ダイアログは、事前に更新しなくても、失敗したデプロイメントのステータスが正確に表示されるようになりました。
CR075935	エンティティ Bean の [ 詳細設定 ] パネルでは、選択リストに、コンテナ管理の永続性エンティティ Bean が修正されたときに無効化できるすべての読み込み専用エンティティ Bean が表示されていませんでした。現在は、そのような Bean がすべてリストに表示されます。
CR078807	デプロイメント記述子で参照されるクラスをコンパイルせずにプロジェクトを開くと、Builder から誤った AssertionError が送出されていました。現在は、クラスが見つからないことを通知する警告がユーザに示されます。
CR081472	WebLogic Builder で、ローカル インタフェースを持たない Bean に誤って JNDI 名が割り当てられることがなくなりました。
CR082276	WebLogic Builder の検証ユーティリティが、アーカイブされたモジュールではなく展開されたモジュールで更新されたクラス ファイルを検出します。Builder は、プロジェクトを閉じて再び開いたときにアーカイブされたモジュールと展開されたモジュールの両方で更新されたクラス ファイルを検出します。

変更要求番号	説明
CR085188	プロジェクトからすべての CMP フィールドを削除すると、Builder によって <code>weblogic-rdbms-cmp-jar.xml</code> が壊される場合があります。プロジェクトからすべての CMP フィールドを削除しようとすると、警告が表示されます。
CR085270	Builder の関連の追加ウィザード、ファインダの追加ウィザード、および CMP の追加ウィザードで、関係の作成を中断またはキャンセルできるようになりました。
CR088145	Builder で、無効な CMP ノードがアクティブになったときに警告が表示されるようになりました。
CR091665	以前のリリースでは、Bean の [CMP Fields] ノードで行を追加し、その後にテーブルビューでそれを削除すると、 <code>AssertionError</code> が送出されていました。この問題は、Builder のユーザインタフェースで不完全に定義されたプロパティが原因でしたが、現在は修正されています。
CR091720	CMP Bean と同様に、BMP Bean でも詳細設定パネルが表示されるようになりました。
CR093246	新しいデプロイメント記述子要素 <code>&lt;allow-remove-during-transaction&gt;</code> が <code>weblogic-ejb-jar.xml</code> デプロイメント記述子に追加されました。このタグはステートフル セッション Bean 用に追加されたものであり、 <code>&lt;stateful-session-descriptor&gt;</code> タグの子です。ただし、WebLogic Builder で表示されることはありません。 この問題は、WebLogic Builder のセッションの [ 詳細設定 ] パネルに [Allow remove during transaction] チェック ボックスを追加することで解決されました。
CR093579	<code>weblogic.Deployer</code> を使用して、Deployers グループのユーザとしてクラスターにアプリケーションをデプロイすると、アプリケーションはデプロイされますが、次の例外が発生しました。 <pre>weblogic.management.NoAccessRuntimeException: Access not allowed for subject: principals=[velvet, Deployers], on ResourceType: Cluster Action: execute, Target: addDeployment     at weblogic.management.internal.Helper\$IsAccessAllowedPrivilegeAct ion.run(Helper.java:2034)</pre> クラスターのデプロイメントに対する ACL を追加して問題を修正しました。
CR093658	<code>VirtualHostMBean</code> の <code>getLogFileName</code> を呼び出すと、不正なパスが返されました。

変更要求番号	説明
CR093833	<p>weblogic.Deployer を使用して、サーバを指定しないで管理対象サーバに再デプロイすると、アプリケーションは管理サーバにデプロイされました。</p> <p>weblogic.Deployer ユーティリティを変更して問題を解決しました。</p>
CR094825	<p>weblogic.Admin は無効な URL 引数を適切に処理しませんでした。次に例を示します。</p> <pre data-bbox="297 461 865 487">java weblogic.Admin https://hostname:8888</pre> <p>サーバが動作していない場合は「Failed to connect」エラー メッセージになるはずですが、説明もなくコマンドが終了したり、ハングしたりすることがありました。</p> <p>問題は解決済みです。javax.net.ssl.SSLSocketImpl.close によって送出された例外が捕捉され、適切なエラー メッセージが表示されるようになりました。</p>
CR095915	<p>同じプロジェクトの別の関連と名前が同じ関連を作成することが、WebLogic Builder ではできなくなりました。</p>
CR096615	<p>エンティティ Bean の [ チューニング ] パネルのキャッシュ テーブルが、キャッシュの参照がない場合は無効になるようになりました。キャッシュの参照がない場合にキャッシュ テーブルの要素をクリックすると、エラーが発生していました。</p>
CR096719	<p>WebLogic Builder は、エンティティ Bean のプール設定をエクスポートしていませんでした。</p> <p>この問題は、エンティティ Bean のプール パネルをチューニング サブノードに追加することで解決されました。</p>
CR096848	<p>6.1 SP04 では、CMP 2.0 Bean で &lt;is-modified-method-name&gt; が呼び出されていませんでした。Bean のメソッドを呼び出すたびに ejbStore() を呼び出し、後でストアを回避するかどうかを判定していました。そのため、&lt;is-modified-method-name&gt; を頻繁に使用するアプリケーションでは、パフォーマンスの問題が発生していました。</p> <p>この問題は、CMP EJB 2.0 用に &lt;is-modified-method-name&gt; 関数を実装することで解決されました。</p>
CR098134	<p>weblogic.Admin はパスワードを要求しますが、入力する前に例外を送出しました。</p> <p>コードを変更してこの問題を解決しました。</p>

変更要求番号	説明
CR098709 Tools	SET 操作を実行する時に、クライアントは MBean に対応する WebLogicObjectName を作成しました。ただし、親など他のメンバーは正しく設定されませんでした。そのため WebLogicObjectName の一貫性がなくなりました。 値が WebLogicObjectName に属するすべての属性を修正して問題を解決しました。
CR098985	setWebLogic ServerEnv.sh スクリプトでは extEnv.sh スクリプトの検索に失敗する場合があります。 extEnv.sh スクリプトの相対アドレスがすべてのシステム シェルで機能しないことが問題でした。 参照を全体で読み込み可能な相対アドレスに置き換えて、問題を修正しました。
CR099260	関連の追加ウィザードで、既存の関連の名前を変更できるようになりました。既存の関連の名前フィールドは、以前は読み込み専用でした。
CR099865	以前は、Bean のデフォルト トランザクション属性をリセットすると、プロジェクトの他の Bean のトランザクション属性もリセットされていました。複数のトランザクション属性を持つ複数のメソッドが、1 つのトランザクション属性を持つ 1 つのメソッドとして再定義されることがありました。この動作は発生しなくなりました。
CR099911	以前は、Bean 管理の永続性を利用する一部の Bean が、デプロイメント記述子 ファイルのトランザクション タイムアウト設定と一致しない期間でタイムアウトになっていました。BMP Bean は、デプロイメント記述子に従って CMP Bean と同じように動作するようになりました。
CR099913	WebLogic Builder の以前のバージョンでは、一部のコントロールが不適切に利用可能になっていました。たとえば、アプリケーションで自動キー生成が無効になっていても Oracle シーケンスの名前値を入力することができました。 WebLogic Builder のインタフェース全体でコントロールが適切に有効および無効になるようになりました。

変更要求番号	説明
CR099913	WebLogic Builder の自動キー生成フレームで、すべてのフィールドが一貫性を持って有効または無効になりませんでした。たとえば、 <automatic-key-generation> タグのない JAR を開いて、[Automatic Key Generation] タブをロードした場合に、チェック ボックスのチェックは外されていますが、それでも値を入力することができました。 この問題は、フィールドが規則に従って無効になるようにコードを修正することで解決されました。
CR099917	トランザクション間のキャッシングの設定が永続的ではなかったので、Bean の [チューニング] パネルを表示し、同時方式を選択して、 [cache-between-transactions] をチェックすることができましたが、プロジェクトを開き直すと、設定が「False」に戻っていました。この設定は永続化されるようになりました。
CR100288	ファインダ メソッドの更新が、永続化されないことがありました。自動コミットの設定が原因でしたが、問題は修正されています。
CR100756	weblogic.Admin で JMSFileStore を設定するときに間違った Type 設定を使用すると、再起動するまでサーバが使用できなくなることがありました。 間違った Type 設定がある場合は、管理サーバのログに記録される InvalidAttributeValueException が発生するようになりました。
CR103174	デバッグ モードで起動されたサーバが、config.xml からデバッグ フラグが削除されるまでデバッグ モードを保持するように、サーバを起動するコマンドラインのデバッグ フラグはドメイン コンフィグレーション ファイルに永続化されていました。 コードの修正により、コマンドライン フラグは config.xml に書き込まれなくなりました。
CR105436	java weblogic.marathon.ddinit.WebInit stageDir は weblogic.xml および web.xml 内のサーブレット コンポーネントの生成に失敗しました。 問題の原因はコード内のリグレッションです。修正して問題を解決しました。



# Web サービス

---

**変更要求番号**   **説明**

---

CR106013      クラスパス内の `weblogic.jar` のクライアント コードで `java.net.HttpURLConnection.getResponseCode` は 200 を返しましたが、コンテナ内で動作しているときに `java.net.HttpURLConnection.getResponseCode` は同じ URL に対して 404 を返しました。404 メッセージの原因は、WebLogic Server の内部から `HttpURLConnection` を使用すると、EJB コードが `weblogic.net.http.HttpURLConnection` のインスタンスを受け取ることでした。`setRequestProperty()` は、値が空の文字列であるヘッダを無視しました。SOAPAction ヘッダは `http://www.xmethods.net/interfaces/query` をホストする Apache サーバで必要でした。

使用するプロトコル ハンドラを指定できるので、Sun の実装をコード内で使用することができます。次の単純なサーブレットで例示します。

```
package test;
import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.net.*;

public class ProtocolHandlerTest extends HttpServlet {
    public void service(HttpServletRequest request,
        HttpServletResponse response) { try { URL testUrl_1 = new
        URL(null, "http://www.yahoo.com:80/";, new
        sun.net.www.protocol.http.Handler());
            HttpURLConnection huc_1 = testUrl_1.openConnection();
            URL testUrl_2 = new URL(null,
        "http://www.yahoo.com:80/");;
            HttpURLConnection huc_2 = testUrl_2.openConnection();
            System.out.println("huc_1 =
        "+huc_1.getClass().getName());
            System.out.println("huc_2 =
        "+huc_2.getClass().getName());
        } catch (java.net.MalformedURLException mue) {
        } catch (java.io.IOException ioe) { } } }
```

---

変更要求番号	説明
CR078871	デプロイされた Web サービスの WSDL を生成するときに、WebLogic Web サービスの実行環境は、操作を実装する EJB メソッドによって送出される例外を WSDL の <code>&lt;soap:fault&gt;</code> 要素に正確にマッピングするようになりました。以前は、それら Web サービス固有の例外は間違っ <code>&lt;soap:body&gt;</code> 要素にマッピングされていました。
CR079631	<code>clientgen</code> Ant タスクは、 <code>clientgen</code> の呼び出しが含まれている <code>build.xml</code> ファイルの <code>&lt;classpath&gt;</code> 子要素を正しく受け入れるようになりました。
CR079776	UDDI クライアント API の <code>weblogic.uddi.client.service.Inquiry.findService</code> メソッドの Javadoc が更新され、このメソッドを使用するときにはサービス名だけでなくビジネスキーも指定するように明示されました。
CR082779	<code>weblogic.webservice.client.WebLogic ServerSLAdapter</code> インタフェースの実装が、それが開いた SSL 接続にユーザ証明書を伝播できるようになりました。以前は、SSL を介してクライアント側の証明書を使用した Web サービスの呼び出しで認証を実行することができませんでした。
CR083861	WebLogic Web サービス ランタイムが、WebLogic Web サービスを実装するステートレス セッション EJB から送出されたときに <code>javax.xml.rpc.soap.SOAPFaultException</code> 例外を変更しなくなりました。以前は、誤って <code>EJBException</code> にラップされていました。
CR084960	Web サービス ランタイムが、呼び出し対象の Web サービスが例外を送出した場合に HTTP 500 「Internal Server Error」 例外をクライアントに適切に伝播するようになりました。実際の例外は、適切に SOAP 障害に格納されます。

---

**変更要求番号**   **説明**

---

CR087303      次の WSDL ファイルの処理中に、clientgen ant タスクは無効なクラスを作成しました。

```
wSDL:message name="UploadCBProfileRequest"> <wSDL:part
name="body" element="tns:TRAMSDATA"/> <wSDL:part
name="authHeader" element="tns:AuthenticationHeader"/>
</wSDL:message> <wSDL:portType name="RelMgrPortType">
<wSDL:operation name="uploadCBProfile"> <wSDL:input
message="tns:UploadCBProfileRequest"/> </wSDL:operation>
</wSDL:portType> <wSDL:binding name="RelMgrSOAPBinding"
type="tns:RelMgrPortType"> <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/;>
<wSDL:operation name="uploadCBProfile"> <soap:operation
soapAction="" style="document"/> <wSDL:input> <soap:body
parts="body" use="literal"/> <soap:header wSDL:required="true"
message="tns:UploadCBProfileRequest" part="authHeader"
use="literal" xmlns:wSDL="http://schemas.xmlsoap.org/wSDL"/;>
</soap:header> </wSDL:input> </wSDL:operation> </wSDL:binding>
```

結果として生成されたタスクには、重複した uploadCBProfile() 引数が含まれていました。

調査の結果、コードでは、メッセージにヘッダ部分が既にある場合でも追加されることが判明しました。コードの修正で、この問題は解決されました。

---

CR087529      WebLogic Server 6.1SP03 では、

```
samples\examples\webservices\rpc\weatherEJBにあるサンプル Web サービスのクライアントが wsgen で生成され、client.jar のみを使用してデプロイされて
いました (weblogic.jar なし)。非 SSL Web サービスが Java クライアントに
よって呼び出されると、次の例外が送出されていました。
```

```
<< Missing class files in servicegen client.jar. Getting
Exception in thread "main" java.lang.NoClassDefFoundError:
weblogic/net/http/URLConnection at
weblogic.soap.http.SoopContext.lookup(SoopContext.java:87) at
javax.naming.InitialContext.lookup(InitialContext.java:350) at
com.verizon.client.client.main(client.java:72) >>
```

分析の結果、何も手を加えていない最低限の機能には weblogic.net.http.URLConnection クラスが必要であり、 client.jar にこのクラスを含める必要があることが判明しました。

この問題は、weblogic.net.http.URLConnection を client.txt に追加することで解決しました。

---

変更要求番号	説明
CR090882	<p>WebLogic Server で JRockit JVM を使用する場合に、UDDIExplorer を使用して Web サービスを正常にプライベート UDDI レジストリにパブリッシュできるようになりました。以前は、サービスをプライベート レジストリにパブリッシュすると、UDDIExplorer エラー「E_fatalError(10500): a serious technical error has occurred while processing the request」が返されました。同じパブリッシュ操作が、デフォルトの Sun JVM では常に適切に機能してきました。</p>
CR092549	<p>非組み込みデータ型コンポーネント (シリアライゼーション クラス、Java クラス など) を生成する Web サービス Ant タスクが、サポートされていない XML スキーマ データ型 xsd:union を、サポートされていない XML スキーマ型のデフォルト Java クラス SOAPElement に正しく変換できるようになりました。以前は、Ant タスクはエラーを生じて失敗していました。</p>
CR095044	<p>WebLogic Web サービス ランタイムが、WebLogic Server から受け取った例外の詳細を CDATA にラップするようになり、SOAP 障害の詳細に無効な XML が含まれているかもしれなかった以前の問題が修正されました。無効な XML の例としては、&lt;no stacktrace available&gt; などの WebLogic Server メッセージが &lt;&gt; の中に配置され、無効な &lt;&lt;no stacktrace available&gt;&gt; になっていました。</p>
CR095109	<p>verbose モードが明示的に有効化されていない限り、ハンドラ実装の操作がある Web サービスをデプロイするときに、WebLogic Server を起動したコマンド ウィンドウに次の警告が出力されなくなりました。</p> <p style="padding-left: 40px;">WARNINIG: Unable to find a javaType for the xmlType:datatype. Make sure that you have registered this xml type in the type mapping Using SOAPElement instead.</p> <p>Web サービス ランタイムがこのケースでは必要のない型マッピング情報をチェックしているので、この警告は紛らわしく、誤解を生みます。</p>
CR095561	<p>servicegen Ant タスクが、typeMappingFile 属性と一緒に使用され、指定された types.xml ファイルに JAX-RPC 命名規約に従っていない非組み込み Java データ型のクラス名が含まれている場合にエラーを返さなくなりました。独自のシリアライゼーション クラス、非組み込み Java データ型などを作成する場合は、Java データ型にどのような名前でも付けることができます。</p>
CR095741	<p>WebLogic Web サービス ランタイムが、xsd:dateTime の時刻部分を適切に扱うようになりました。以前は、タイムゾーンのオフセットが適切に処理されなかったことが原因で、タイムゾーン次第で時刻が変わっていました。</p>

変更要求番号	説明
CR096906	WebLogic Web サービス ランタイムが、 <code>xsd:dateTime</code> データ型を適切に処理するようになりました。以前は、タイムゾーンが正であるときに <code>dateTime</code> 値のタイムゾーン オフセット部分から「+」の文字が省略されていました。
CR103937	SSL 接続を中止したときに、しばらくの間 <code>CLOSE_WAIT</code> が続いて、ソケットが開いたままになることがあります。 コードを変更してソケットのリークを修正しました。
CR104719	次のシグネチャで Web サービスを実装すると、 <pre>public void echoDom(Document doc)</pre> コンパイル エラーが発生しました。 ビルド時または実行時にドキュメント スタイルの <code>type-mapping</code> に <code>void</code> を追加できないようにし、ドキュメント スタイルの <code>void</code> 部分の書き込みをできないようにし、ドキュメント スタイルのドキュメント シリアライザでラッパー要素を使用できないようにすることで、この問題を解決しました。
CR104783	<code>clientgen</code> は複合型に対するシリアライザとデシリアライザを作成しませんでした。 <code>clientgen</code> はモデル グループへの単一の参照を含む複合型を扱うようになりました。

---

**変更要求番号**   **説明**

---

CR106392   invocation-style="one-way" で実行すると、以下の抜粋に示すように、その操作の <binding> セクションに <output> が含まれる WSDL になります。

```
<portType name="MyServicePort">
  <operation name="sayNothing">
    <input message="tns:sayNothing" />
  </operation>
</portType>

<binding name="MyServicePortSoapBinding"
type="tns:MyServicePort">
  <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"; />
  <operation name="sayNothing">
    <soap:operation soapAction="" style="document" />
  <input>
    <soap:body use="literal" namespace="http://tempuri.org"; />
  </input>
  <output>
    <soap:body use="literal" namespace="http://tempuri.org"; />
  </output>
</operation>
</binding>
```

この動作は「rpc」と「document」の両方のスタイルの Web サービスで確認されました。この WSDL に対して Microsoft の WSDL.EXE を実行すると、検証が失敗し、続いて .Net クライアント プロキシまたはスタブの生成が失敗します。

一方向のバインディングでは出力メッセージの生成を停止するようにコードを変更して、問題を解決しました。

---

CR107254   Hewlett-Packard は HP Apache-based Web Server バージョン 1.3.x のサポートを中止したため、WebLogic Server は HP 用の Apache 1.3 プラグイン を削除しました。

---

---

**変更要求番号** **説明**

---

- CR108332 シンクライアント (wlclient.jar) を使用する場合、IIOP ストリームの破損を示すマーシャリング例外が発生することがありました。この問題は、J2SE 1.4 の複数のマイナーバージョンにおいて、プロトコルとして IIOP または T3 が指定されているものの、クライアントのクラスパスに weblogic.jar がない場合に発生しました。
- この問題は、カスタム マーシャリングされた valuetype のエンコーディングとデコーディングの問題と、valuetype をマーシャリングする時の WebLogic Server によるインダイレクションの追跡方法の問題が原因で発生しました。
- これらの問題は解決されました。
- 

## WebLogic Tuxedo Connector

---

**変更要求番号** **説明**

---

- CR074963 大きな Field Manipulation Language (FML) テーブル (約 4000 エントリ) を作成するときには、mkfldclass [32] によってハッシュ テーブルへの各エントリをインスタンス化するメソッドが生成されていました。大きな FML テーブルの場合、この 1 つのメソッドで JVM のサイズ制限を超えてしまいます。
- この問題は、動的なフィールド テーブル ロード メカニズムを実装することで解決されました。詳細については、「[mkfldclass32 クラスに対する DynRdHdr プロパティの使用](#)」を参照してください。
-

---

変更要求番号	説明
CR089100	<p><code>viewj</code> メソッドと <code>viewj32</code> メソッドは、多数のフィールドが含まれるビュー ファイルをコンパイルするときに巨大な Java クラスを作成する場合があります (コンストラクタ メソッドのコードのバイト数が 65535 を超える)。</p> <p>分析の結果、コンストラクタのコードの大部分はフィールドや配列要素に <code>null</code> 値を割り当てるものであり、削除可能であることが判明しました。</p> <p>この問題は、ビュー ファイルのコンパイル時に冗長なコードを抑制するようにコードを修正することで解決されました。</p>
CR092860	<p>WTC を使用して Tuxedo 8 Corba オブジェクトにアクセスすると、JavaIDL Reader スレッド リークが発生していました。シンプルなサンプル アプリケーションを実行すると、WebLogic Server が動作する JVM で JavaIDL Reader スレッドが連続的に発生していました。それらのスレッドは、サーバが停止されるまで破棄されませんでした。</p> <p>分析の結果、<code>finalize()</code> が適切なきに呼び出されていないことが判明しました。<code>ORB.destroy()</code> の呼び出しを追加するようにコードが修正されて、問題は解決しました。</p>

---

## XML

---

変更要求番号	説明
CR087201	<p>(Apache の Xerces に基づく) 組み込み XML パーサが、日本語の文字が含まれる XML ファイルの解析時に <code>java.lang.ArrayIndexOutOfBoundsException</code> 例外を送出しなくなりました。</p>
CR090137	<p>(Apache の Xalan に基づく) 組み込み XML トランスフォーマが、XSL スタイルシートを使用して XML ファイルを HTML ファイルに変換するときに <code>&lt;/META&gt;</code> タグを適切に追加するようになりました。以前は、出力 HTML ファイルで <code>&lt;/META&gt;</code> タグが生成されず、XML が無効になっていました (HTML ファイルはブラウザで適切に表示されていた)。</p>

---



---

変更要求番号	説明
--------	----

---

CR095073	以前は、特定のタイプの XML ファイルで、反復処理が増加して解析の応答時間が遅くなっていました。この問題は現在は修正されています。
----------	--

---

---

変更要求番号	説明
CR100068	<p data-bbox="299 256 1161 317">次のような条件で、日本語文字で JSP の JSTL タグを使用すると衝突が発生しました。</p> <ol data-bbox="299 331 1042 465" style="list-style-type: none"><li data-bbox="299 331 1042 392">1. JSP のページ エンコーディングを Shift_JIS で定義する。&lt;%@ page pageEncoding="Shift_JIS" %&gt;</li><li data-bbox="299 401 827 427">2. JSP でマルチバイト文字 (日本語) を使用する。</li><li data-bbox="299 440 563 465">3. JSTL タグを使用する。</li></ol> <p data-bbox="299 482 854 508">JSP を実行すると次のようなエラーが発生しました。</p> <pre data-bbox="299 522 1170 1373">java.io.IOException: javax.servlet.jsp.JspException: The taglib validator rejected the page: "org.xml.sax.SAXParseException: An invalid XML character (Unicode: 0x82) was found in the CDATA section., " at weblogic.servlet.jsp.Jsp2Java.outputs(Jsp2Java.java:124) at weblogic.utils.compiler.CodeGenerator.generate(CodeGenerator.java:258) at weblogic.servlet.jsp.JspStub.compilePage(JspStub.java:353) at weblogic.servlet.jsp.JspStub.prepareServlet(JspStub.java:211) at weblogic.servlet.jsp.JspStub.prepareServlet(JspStub.java:164) at weblogic.servlet.internal.ServletStubImpl.getServlet(ServletStubImpl.java:517) at weblogic.servlet.internal.ServletStubImpl.invokeServlet(ServletStubImpl.java:351) at weblogic.servlet.internal.ServletStubImpl.invokeServlet(ServletStubImpl.java:445) at weblogic.servlet.internal.ServletStubImpl.invokeServlet(ServletStubImpl.java:306) at weblogic.servlet.internal.WebAppServletContext\$ServletInvocationAction.run(WebAppServletContext.java:5445) at weblogic.security.service.SecurityServiceManager.runAs(SecurityServiceManager.java:780) at weblogic.servlet.internal.WebAppServletContext.invokeServlet(WebAppServletContext.java:3105) at weblogic.servlet.internal.ServletRequestImpl.execute(ServletRequestImpl.java:2588) at weblogic.kernel.ExecuteThread.execute(ExecuteThread.java:213) at weblogic.kernel.ExecuteThread.run(ExecuteThread.java:189)</pre> <p data-bbox="299 1390 1130 1442">makeXMLStream のエンコーディングが正しくないことが判明しました。エンコーディングを UTF-8 に変更して問題を解決しました。</p>

---

# WebLogic Server 7.0 サービス パック 2 のソリューション

以下の節では、WebLogic Server 7.0 サービス パック 2 で解決された問題を説明します。

- [4-304 ページの「Administration Console」](#)
- [4-306 ページの「クラスローダ」](#)
- [4-307 ページの「クラスタ」](#)
- [4-310 ページの「コネクタ」](#)
- [4-310 ページの「コア サーバ」](#)
- [4-314 ページの「デプロイ」](#)
- [4-315 ページの「EJB」](#)
- [4-319 ページの「サンプル」](#)
- [4-320 ページの「インストーラ」](#)
- [4-320 ページの「インターナショナルライゼーション」](#)
- [4-320 ページの「jCOM」](#)
- [4-321 ページの「JDBC」](#)
- [4-322 ページの「jDriver」](#)
- [4-323 ページの「JMS」](#)
- [4-326 ページの「JNDI」](#)
- [4-327 ページの「JTA」](#)
- [4-327 ページの「JTS」](#)
- [4-327 ページの「その他」](#)
- [4-330 ページの「ノード マネージャ」](#)
- [4-330 ページの「プラグイン」](#)

- [4-332 ページの「RMI/RMI-IIOP」](#)
- [4-333 ページの「セキュリティ」](#)
- [4-336 ページの「サブレットと JSP」](#)
- [4-342 ページの「ツール」](#)
- [4-344 ページの「WebLogic Tuxedo Connector」](#)
- [4-345 ページの「Web サービス」](#)
- [4-346 ページの「XML」](#)

## Administration Console

変更要求番号	説明
CR074370	デプロイメント記述子エディタで EJB 記述子の値を変更すると、 <code>javax.management.AttributeNotFoundException</code> が発生していました。この問題は解決済みです。
CR074653	Administration Console デプロイメント記述子エディタで、メモリまたはディスクへの書き込み時に「&」が「&amp;」で置換されるようになりました。「&」は XML ファイルで「&amp;」としてエスケープしなければならない特殊文字であり、これを行わないとデプロイメントが失敗する可能性があるからです。
CR074677	WebLogic Server のバージョン 6.x で作成されたドメインをアップグレードすると、ユーザ system は、Administration Console へのアクセスを明示的に許可するようになりました。
CR074933	翻訳されたオンライン ヘルプが表示されませんでした。いまは表示されるようになっています。
CR079805	対象サーバに割り当てる Web アプリケーションが複製サーバに割り当てる Web アプリケーションとして表示されるようになりました。
CR081366	1 つまたは複数のサーバがコンフィグレーションされているマシンで、その削除が失敗しなくなりました。
CR081445	Netscape 4.7 ブラウザで EJB のモニタ ページが空白になることがなくなりました。

変更要求番号	説明
CR082345	WebLogic 7.0.0.0 では、JDBC ドライバが \$CLASSPATH にない場合、コンソールから NPE が送出されました。この問題はこのリリースで修正されています。
CR082942	ユーザ パスワードを変更して [ 続行 ] をクリックしたときに NullPointerException が送出されなくなりました。
CR083242	レルム アダプタ ATN プロバイダが 7.x セキュリティ レルムに追加されたときにサーバを再起動できるようになりました。
CR083330	Administration Console でマルチプールをクラスタに割り当てることができるようになりました。
CR083465	[ サーバ   サービス   ブリッジ   すべてのメッセージ ブリッジ ランタイムのモニター ] をクリックしても 404 エラーが発生しなくなりました。
CR083530	ログのタイムスタンプがグリニッジ標準時 (GMT) で押されるように指定するためのコンフィグレーション オプションが Administration Console に追加されました。
CR084558	サンプルの WebLogic Portal Server に付属している Pipeline EJB の EJB デプロイメント記述子を編集する場合に、javax.management.AttributeNotFound などの MBean 例外で失敗しなくなりました。
CR084856	日本語環境で [ セキュリティ   ユーザ   ユーザのロックを解除 ] が文字化けすることがなくなりました。
CR085101	Administration Console を使用して EJB メソッドで適切にポリシーを定義できるようになりました。
CR086107	Administration Console でグローバル Admin/Operator/Monitor/Deployer ロールを使用できるようになりました。
CR087802	EJB デプロイメント記述子を編集しても、NullPointerException が送出されなくなりました。
CR089044	ブラウザがサポートされていないときにブラウザ警告ページが Administration Console に表示されなくなりました。
CR092102	Administration Console がスレッド プールをモニターするように設定されている場合に、管理サーバが OutOfMemory 例外を送出しなくなりました。

## クラスローダ

変更要求番号	説明
CR056911	<p>.ear ファイル内の 2 つの別々の .war ファイルがそれぞれ独自のシングルトンインスタンスをインスタンス化し、アプリケーションの実行時に結果が不正になっていました。</p> <p>この問題は修正済みです。</p>
CR081377	<p>デプロイメント時にコンテキスト クラス ローダを通じてクラスをロードできるようになり、ClassNotFound 例外が送出されなくなりました。</p>
CR082466	<p>ClassFinder に複数のエントリが追加されていました。その結果、ClassLoader.getResources が重複したエントリのある列挙値を返していました。</p> <p>この問題は修正されています。</p>
CR083752	<p>クライアントクラスは、java コマンドラインで -Xbootclasspath 引数を使用して weblogic.jar を指定する場合に InitialContext を正常に作成できるようになりました。</p>

# クラスタ

変更要求番号	説明
CR043366	<p>WebLogic Server 6.1 では、どのプラットフォームで動作する場合でも、クラスタ アドレスを次のようなカンマ区切りの IP アドレスのリストとして指定すると、  <code>IPAddress1, IPAddress2</code>            または  <code>IPAddress1:7011, IPAddress2:7011</code>            クラスタ内の管理対象サーバの起動時に次のようなエラー メッセージが発行されました。</p> <pre>&lt;main&gt; &lt;system&gt; &lt;&gt; &lt;000101&gt; &lt;Cannot resolve ClusterAddress: IPAddress1:7011, IPAddress2:7011</pre> <p>クラスタは正常に起動しますが、カンマ区切りのリストは解決されませんでした。ドメインを開発モードに設定する場合、カンマ区切りの IP アドレスのリスト形式で指定されるクラスタ アドレスをサポートするように、コードの変更が実装されました。この形式はプロダクション モードの操作ではサポートされません。この形式を使用すると、EJB ハンドルが正しく管理されず、フェイルオーバに問題が発生する可能性があります。</p> <p>プロダクション モードで、クラスタ アドレスは、クラスタ内の管理対象サーバにマップされた DNS 名として指定する必要があります。</p>
CR074236	<p>クラスタワイドの JNDI ツリーの作成と MDB のデプロイメントに関連するタイミングの問題により、起動時に <code>NameNotFoundException</code> が発生しました。MDB の <code>ejbCreate</code> メソッドは、クラスタ内の単一のノードに固定された EJB のサービスを呼び出しました。アプリケーションの要件に関する理由のため、このノードは最初に起動されます。2 番目のノードが起動されるときに、<code>initial-beans-in-pool</code> の設定が 0 より大きい MDB がデプロイされると、Bean は <code>NameNotFoundException</code> を送りました。これは、固定された EJB のホーム インタフェースがまだ JNDI ツリーにバインドされていないために発生しました。初期プールの設定が 0 の MDB がデプロイされる場合は、ノードは正常に起動し、JNDI が追いつく機会があるため、最初のメッセージがキューに入れられるときに Bean は正しく動作しました。</p> <p><code>ClusterMBean</code> の <code>MemberWarmupTimeoutSeconds</code> 属性を実装することで、この問題は解決されました。デフォルト値は 0 で、クラスタが待機しないことを意味します。デフォルトでは、クラスタの起動シーケンスは変更されません。値を &gt;0 に設定すると、クラスタ メンバーは起動時にその時間だけ待機して、他のメンバーと同期します。すべての MDB は待機時間の経過後にデプロイされます。</p>

---

変更要求番号	説明
CR078455	<p>6 つのノードを持ち、Web アプリケーションをホストするクラスタを 7 時間実行すると、繰り返しハングするようになりました。F5/Big IP ロード バランサが、WebLogic Server 6.1 SP2 を実行する 3 つのサーバから成るクラスタにリクエストを転送し、次に IIS および ISAPI プラグインが、6 つの WebLogic Server 6.1 SP2 サーバから成るクラスタにプロキシする、というコンフィグレーションでした。ホスト マシンは JDK 1.3.1 を実行する Solaris 8 デュアル CPU サーバでした。</p> <p>プラグインはプライマリまたはセカンダリ以外のサーバインスタンスにリクエストを転送し、WebLogic Server はリモート <code>ROIDImpl</code> をルックアップして呼び出していました。</p> <p><code>ROIDLookup</code> および Replication Manager RMI オブジェクトのローカル スタブを作成して、スタブの JNDI ルックアップを避けるという修正によって、この問題は解決されました。また、<code>ROIDLookup</code> の呼び出しではレプリケーション スレッドを使用して、デフォルト実行スレッドが使用できないときでも確実に実行し、起こり得るハングを回避しています。</p>
CR078677	<p>Web アプリケーションはクラスタにデプロイされました。WebLogic Server 6.1 SP2 の管理対象サーバの正常停止中に、<code>HttpClusterServlet</code> はフェイルオーバーしませんでした。Web アプリケーションはアンデプロイされ、停止シーケンス中にリクエストを受信したときにコンテキストへの一致を見つけることができませんでした。透過的なフェイルオーバーの代わりに、「page not found」エラーがブラウザに表示されました。この問題はサーバがクラッシュする場合には発生せず、正常停止中にのみ発生しました。</p> <p>サーバ状態のチェックと、コンテキストが <code>null</code> の場合に <code>SERVICE_UNAVAILABLE</code> (503) を返すロジックを実装して、プラグインがクライアントに 404 (NOT FOUND) を返さずにフェイルオーバーできるようにしたことで、この問題は修正されました。</p>

---



変更要求番号	説明
CR080153	<p>NT マルチホーム システムでは、管理対象サーバの検出モードにある管理サーバは、URL ホストがホスト名と同じで、ホスト名が複数の IP アドレスに解決される場合に、再起動しませんでした。ホスト名が複数のアドレスに解決される場合、管理サーバはどの管理対象サーバを管理しているのが確実に特定できないためエラーが発生し、管理サーバの再起動は完了しませんでした。</p> <p>アドレスがマルチホームの場合、管理対象サーバの起動時に管理サーバに対してドット区切りの IP アドレスが指定されたかどうかを特定するためのチェックが Admin.java に追加されました。それに当てはまる場合は、次のメッセージが表示されます。</p> <pre>'The admin server url "http://qa700:9001"; cannot be used with managed-server-discovery-mode because the hostname resolves to multiple ip addresses. Please turn off discovery mode or set just one ip address for this host.'</pre>
CR080976	<p>タイムアウト オプションを指定して管理サーバを停止すると次のエラー メッセージが出力される問題が修正されました。</p> <pre>Expected RemoteException, RuntimeException, or Error but received: 'class weblogic.rjvm.PeerGoneException.'</pre> <p>エラーは訂正されました。タイムアウト オプションを指定して停止すると、「The shutdown sequence has been initiated」というメッセージが出力されるようになりました。</p>
CR082443	<p>クラスタのデバッグ フラグを有効にできないため、ログ ファイルのサイズが大きくなりました。weblogic.cluster.ClusterDebug.java で、デバッグ メッセージがログに記録されるかどうかを制御するフラグが true に設定されていて、変更できなかったために、この問題が発生しました。</p> <p>config.xml で、次のようにタグを使用してフラグをコンフィグレーションできるようにしたため、問題は解決されました。</p> <pre>&lt;ServerDebug DebugCluster="true" Name="myserver"/&gt;</pre>
CR083532	<p>ReplicaAwareRemoteRef.getCurrentReplica の Null ポインタ例外により、停止が完了する前に停止プロセスが終了していました。</p> <p>このエラーは修正されました。</p>

## コネクタ

変更要求番号	説明
CR080250	weblogic-ra.xml 記述子ファイルなしで RAR をデプロイしても <code>NullPointerException</code> が発生しなくなりました。
CR082608	JCA 接続ファクトリが、利用可能な接続を使用しないで新しい接続を取得しようとする問題が修正されました。
CR086251	WebLogic Server の停止中に、メッセージングブリッジアダプタのアンデプロイメントで <code>NullPointerException</code> が送出されなくなりました。
CR089279	WebLogic Server は、非推奨となったデフォルトの weblogic-ra.xml 要素を保存しなくなりました。

## コア サーバ

変更要求番号	説明
CR032447	意図的にサーバを停止したときに「Emergency」メッセージが発行されなくなりました。
CR058358	EJB および Web アプリケーションで alt-dd デプロイメント記述子要素を適切に処理できるようになりました。
CR062756	ソケット読み取りがブロックされて、JDK バージョン 1.3.1 でサーバがハングする問題が解決されました。
CR068646	トンネリングがサーバ Bean で無効になっている場合に、HTTP トンネリングで <code>weblogic.admin</code> が成功しなくなりました。
CR070429	<code>-Dweblogic.safeCommoBoot=true</code> フラグを指定してサーバを起動することで、前回の正常な COMMO コンフィグレーションを使用して再起動できるようになりました。
CR071796	デフォルト以外のディレクトリから起動したサーバで、そのアプリケーションを確認できるようになりました。

変更要求番号	説明
CR074265	weblogic.t3.srvr.ListenThread の MAX_BACKOFF_BETWEEN_FAILURES を使用することで、サーバリスン スレッドがソケット接続を承認しようとしている場合に「Failed to listen on port」メッセージを送信する期間をコンフィグレーションでできるようになりました。詳細については、weblogic.t3.srvr.ListenThread javadoc の getMaxBackoffBetweenFailures() および setMaxBackoffBetweenFailures() を参照してください。
CR077831	AppletArchiver コーティリティを使用してクライアント jar ファイルを生成し、対象アプレットが javax.swing.JApplet クラスをそのノードで使用する場合、エラーが送出されなくなりました。
CR079547	クラスタ化環境で JMS サーバへの恒久サブスクライバを作成するときに、InvalidClientIDException 例外と NameAlreadyBound 例外が発生しなくなりました。
CR080744	さまざまなマルチプレクサの問題が解決されました。
CR081176	ServerLifecycleRuntimeBean の getState および getStateVal プロパティの動作がほぼ同じになりました。
CR082026	WebLogic Server がクライアントでソケットを登録するたびに JavaSocketMuxer で新しいスレッドが作成されることがなくなりました。
CR082348	WebLogic Server コンソールで wlidomain を停止しても java.lang.ThreadDeath が発生しなくなりました。
CR082533	以前のバージョンおよびサービス パック (6.1SP3) では、ChangeAwareClassLoader からの一部のデバッグ メッセージが stdout に不適切に書き込まれました。このバグはこのリリースで修正されています。
CR083654	WebLogic Server 6.1SP3 のパフォーマンスは、Microsoft WAS Tool からの連続的なリクエストで若干影響を受けます。応答でマルチバイト文字セットをコンテンツ タイプとして使用するサーブレットおよび JSP の修正によって、パフォーマンスが改善されました。
CR084977	「ルート」ID が、サーバの起動後に .wlnotdelete の制御を保持しなくなりました。
CR085544	NT サービスから weblogic.Server.stop() を呼び出して WebLogic Server インスタンスを停止しても NullPointerException が送出されなくなりました。

## サービス パック 1 から 6 で解決済みの問題

変更要求番号	説明
CR085659	RJVM の同期コードが、マシンの切断中に EJB のファイルオーバを制限しなくなりました。
CR085669	HPUX で、SAP JCO が Java 仮想マシン (JVM) のクラッシュを引き起こさなくなりました。
CR086108	WebLogic Server 6.1SP3 は WebLogic Server 7.0 SP2 と比較すると CPU 時間をより多く消費するため、ListenThread においてパフォーマンスが低下します。サーバのスロットリングが、アクティブなソケットの情報をキャッシュすることでパフォーマンスを低下させていました。 この問題は修正済みです。
CR086425、 CR84172	MTU (Large Message Transmission Unit) によってパフォーマンスの低下が発生しなくなりました。
CR086552	Java のシリアライゼーション中に、java.io.OptionalDataException が送出されなくなりました (WebLogic Server 7.0SP1)。
CR086758	Web 層と EJB 層を備えたアプリケーションで、EJB 層が ConnectionManager を閉じたときにその EJB 層と対話する Web 層がハングして、以下のような例外を送出することがなくなりました。 <pre>&lt;Sep 24, 2002 8:43:58 AM PDT&gt; &lt;Info&gt; &lt;RJVM&gt; &lt;Failure in heartbeat trigger for RJVM: '7831636024374910916S:10.10.10.187:[8001,8001,8002,8002,8001,8002,-1]:webserver:sourcingWebserver' java.rmi.ConnectException: The connection manager to ConnectionManager for: 'weblogic.rjvm.RJVMImpl@3bedf2 - id: '7831636024374910916S:10.10.10.187:[8001,8001,8002,8002,8001,8002,-1]:webserver:sourcingWebserver' connect time: 'Tue Sep 24 06:15:16 PDT 2002'' has already been shut down at weblogic.rjvm.ConnectionManager.getOutputStream(ConnectionManager.java:1348) at ...</pre>

変更要求番号	説明
CR087180	<p>管理対象サーバが完全に停止する前に管理対象サーバから管理サーバへの接続が早まって終了してしまうことがなくなりました。7.0SP01 では、次のような例外が送出されていました。</p> <pre data-bbox="357 361 1233 782">&lt;Oct 1, 2002 3:22:42 PM PDT&gt; &lt;Warning&gt; &lt;rmi&gt; &lt;080006&gt; &lt;Failed to associate the transaction context with the response while marshalling an exception: java.rmi.ConnectException: The connection manager to ConnectionManager for: 'weblogic.rjvm.RJVMImpl@7d37fa - id: '2847122412264039151S:172.17.24.65: [7001,7001,7002,7002,7001,7002,-1]:mydomain:myserver' connect time: 'Tue Oct 01 15:18:40 PDT 2002'' has already been shut down java.rmi.ConnectException: The connection manager to ConnectionManager for: 'weblogic.rjvm.RJVMImpl@7d37fa - id: '2847122412264039151S:172.17.24.65: [7001,7001,7002,7002,7001,7002,-1]:mydomain:myserver' connect time: 'Tue Oct 01 15:18:40 PDT 2002'' has already been shut down at weblogic.rjvm.ConnectionManager.getOutputStream(ConnectionManager.java:1564) at</pre>
CR087254	IIOP のパフォーマンスの低下が解決されました。
CR088056	ビルドの日付がマルチプレクサ ライブラリに埋め込まれるようになりました。
CR088415	ulimit -n を unlimited に設定したときに AIX 上のサーバがクラッシュしなくなりました。
CR089035	クラスタ内の単純なステートレス セッション Bean にアクセスするときに、IllegalArgumentException: port out of range が送出されなくなりました。
CR089144	カスタム CallRouter による EJB のデプロイメントが、IllegalArgumentException で失敗しなくなりました。
CR092183	PosixMuxer が Interrupted System Call 例外を送出しなくなりました。
CR092375	メソッドが対話状態にある場合に、手動によるハンドルのルックアップとそれに続く保存およびキャッシュが機能していませんでした。これは、EJBHandle オブジェクトのシリアライゼーションとデシリアライゼーションを変更することで修正されています。

## デプロイ

変更要求番号	説明
CR110897、 CR125329、 CR129494	アプリケーションをデプロイし、その準備を解除してからサーバを再起動した場合、サーバは <code>IllegalArgumentException</code> を送出していました。  分析の結果、J2EE コンテナが、コンフィグレーションされているがデプロイはされていないアプリケーションを起動時に処理していないことが判明しました。そのようなアプリケーションは、 <code>&lt;Application&gt;</code> スタンザで <code>Deployed="false"</code> となっていました。  J2EE コンテナのコードが修正され、この周辺条件に対応するようになりました。
CR049340	コンソールでアプリケーションをインストールするか、開発モードでアプリケーションをアプリケーション ディレクトリにコピーする場合、 <code>applications.xml</code> で定義された順番でコンポーネントがデプロイされるようになりました。
CR080216	Web アプリケーションを断続的にホット デプロイまたは手動でデプロイしても、 <code>NoSuchObjectException: Bean is already undeployed</code> 例外が送出されなくなりました。
CR080537、 CR084907	起動クラスがアプリケーションより前にデプロイされるようにマークされている ( <code>LoadBeforeAppDeployments</code> が <code>True</code> に設定されている) にもかかわらず、サーバの起動時にアプリケーションがデプロイされた後でロードされる問題を修正しました。
CR080929	<code>weblogic.refresh</code> は、ワイルドカード文字を使用して指定されたファイル名に対しても機能するようになりました。
CR081311	DTD の <code>SaxParseException</code> は、重要度が <code>Info</code> ではなく <code>Warn</code> になりました。
CR082263	<code>weblogic.deploy</code> におけるクロスプラットフォームパスの非互換性の問題がなくなりました。 <code>weblogic.deploy</code> は Web アプリケーション ( <code>.war</code> ファイル) を Windows 2000 マシンから Solaris 上で動作するクラスタに正常に更新できるようになりました。
CR083179	<code>web.xml</code> および <code>weblogic.xml</code> の <code>resource-ref</code> 要素と <code>resource-description</code> 要素が同期していない場合に、エラー メッセージは一般的な情報ではなく問題の <code>ResourceReference</code> を示すようになりました。
CR083652	アプリケーションが起動時に 2 回ではなく 1 回デプロイされるようになりました。

変更要求番号	説明
CR085762	管理対象サーバに割り当てられたアプリケーションは、管理対象サーバが動作していない場合でも正しくデプロイされます。
CR085979	カスタム MBean が定義されている場合に、 <code>MBeanHome.getAllMBeans()</code> が <code>java.lang.IllegalArgumentException</code> を送出しなくなりました。
CR087830	管理対象サーバに割り当てられたアプリケーションは、Administration Console から管理サーバを停止して再起動した後も正しくデプロイされることが報告されています。
CR089098	この問題は解決済みです。 Web アプリケーションは仮想ホストに割り当てられ、仮想ホストは管理対象サーバに割り当てられています。 Web アプリケーションで JSP が変更されます。 Web アプリケーションで行われた変更は、新しいリクエストが送信されたときにまだブラウザに表示されていないため、アンデプロイとその後の再デプロイが失敗します。

## EJB

変更要求番号	説明
CR062512	7.0 において、 <code>Required</code> および <code>NotSupported</code> の TX 属性を持つステートフルセッション Bean がある場合、パフォーマンスが若干低下する問題が修正されました
CR060965	EJB QL クエリで等号記号 '=' の前後のスペースが不要になりました。

変更要求番号	説明
CR063275	<p>WebLogic Server EJB では、byte-cmp フィールドにファインダ メソッドを書き込むことができませんでした。これは、「EJB QL で許される型は、エンティティ Bean と依存オブジェクトの抽象スキーマ型、cmp-field の定義済みの型、およびリモート エンティティ Bean のエンティティ オブジェクト型である」という J2EE EJB-QL の定義に準拠していません。この問題は WebLogic Server 6.1 SP01 で発見されました。このエラーはビルド時に発生しました。</p> <pre data-bbox="299 465 1166 864">ejbc:     [java]     [java] ERROR: Error from ejbc: Error while reading 'META-INF/weblogic-cmp-rdbms-jar.xml'. The error was:     [java]     [java] invalid query: In EJB containerManaged, for a query defined in the ejb-jar.xml file with a method signature, findFk(byte[]), we failed to find a corresponding method in the remote home interface, local home interface, or bean class that matches this signature. Note that class parameters such as java.lang.String must be fully qualified, thus 'String' would not match 'java.lang.String'.     [java]</pre> <p>WebLogic Server EJB は、byte-cmp フィールドでファインダ メソッドをサポートするようになりました。</p>
CR063502	<p>java.lang.Long 用の自動主キー生成がサポートされました。</p>
CR076272	<p>ホットデプロイされた EJB のエラー レポート機能が向上しました。</p>
CR079616	<p>WHERE 句のないサブクエリで、不正な SQL が生成されなくなりました。</p>
CR079745	<p>weblogic-ejb-jar.xml に別の .xml ファイルが含まれている場合でも ejbc はデプロイメントを正確に実行するようになりました。</p>
CR080030	<p>EJB 1.1 Bean を EJB 2.0 に更新するときに、weblogic-ejb-jar.xml の is-modified-method-name タグによって java.lang.NoSuchMethodException が送出されなくなりました。</p>
CR080569	<p>EJBCacheRuntime 統計でマイナス値が報告されなくなりました。</p>
CR080985	<p>インメモリ レプリケーションのライセンスが検出されない場合、EJB コンテナはエラーを送出する代わりにこの状況を適切に処理するようになりました。</p>



変更要求番号	説明
CR081735	<code>dispatchPolicy</code> オプションが <code>ejbc</code> と <code>rmic</code> の両方のパブリック オプションになりました。
CR081753	Bean を何度デプロイおよびアンデプロイしようとしても、サーバがメモリ不足にならなくなりました。
CR081807	<code>home-is-clusterable</code> が <code>False</code> に設定されている場合に、サーバがクラスタ内の他のノードに通知を送信しなくなりました。
CR081817	<code>weblogic-ejb-jar.xml</code> ファイルで <code>dispatchPolicy</code> を設定する Bean レベルのオプションが用意されました。
CR082182	<code>ejbc</code> 出力ディレクトリのソース ファイルが、 <code>ejbc</code> の実行時に誤って削除されてしまうことがなくなりました。
CR082350	6.1 SP3 には、EJB 処理サイズが大幅に増加してしまう問題がありました。この問題は解決済みです。
CR082451	JMS プロバイダが WebLogic 固有ではない状況でサードパーティ jar が適切にロードされるようになりました。
CR082566	Microsoft SQLServer を使用する場合に Bean 単位のペシミスティックな同時方式がサポートされるようになりました。ペシミスティック ロックは、DD で <code>use-select-for-update</code> タグを使用すると実行できます。これは MSSQL Server では機能しません。該当する SQL は次のような形式になります。 <code>SELECT ... FROM ... WITH(UPDLOCK) WHERE ...</code> DD の <code>database-type</code> タグが SQL Server の場合は、すべての SELECT クエリで「WITH(UPDLOCK)」のサポートを追加してください。
CR082787	以前のバージョンでは、 <code>create()</code> および <code>remove()</code> メソッドを <code>TX_SUPPORTS</code> にすると、ステートフルセッション Bean の基本的なロックおよびロック解除機能が破壊されました。この問題は、 <code>create()</code> および <code>remove()</code> を TX が指定されていないコンテキストで実行するように制限することで修正されました。

変更要求番号	説明
CR083050	<p>WebLogic Server 6.1SP# では、<code>ejbRemove()</code> の <code>context.getCallerPrincipal()</code> を呼び出すと、以下のような例外が送出されました。</p> <pre data-bbox="299 331 1163 609">&lt;Jul 31, 2002 4:52:17 PM EDT&gt; &lt;Error&gt; &lt;HTTP&gt; &lt;[WebAppServletContext(4292442,test,/test)] Servlet failed with Exception weblogic.ejb20.interfaces.PrincipalNotFoundException: Error. Method SecurityHelper.getCallerPrincipal returned a NULL Principal. The CallerPrincipal Stack has been corrupted. One cause of corruption might be: A Bean has Created its own Threads. Note that this would be in violation of EJB2.0 Final Spec Chapter: Runtime Management 24.1.2 at weblogic.ejb20.internal.EJBRuntimeUtils.getCallerPrincipal(EJBR untimeUtils.java:526) at ...</pre> <p>これはこのリリースで修正されています。</p>
CR083239	<p>EJB QL NOT MEMBER 句によって無限ループが発生しなくなりました。</p>
CR083240	<p>EJB QL NOT MEMBER 句と WHERE 句の組み合わせによって不正なクエリが作成されなくなりました。</p>
CR083689	<p>コンテナ管理の永続的な自動キー生成を SQLServer と一緒に使用できるようになりました。</p>
CR084407	<p>以前の一部のバージョンでは、<code>ExclusiveEntityManager</code> が Bean をプールに正しく返しませんでした。<code>ExclusiveEntityManager</code> からの Bean の場合、Bean はキャッシュから出るとフリー プールに戻されませんでした。これはこのリリースで修正されています。</p>
CR084938	<p>アプリケーション スコープ キャッシュを使用する読み取り専用 Bean の <code>read-timeout-seconds</code> をコンフィグレーションできるようになりました。</p>
CR084978	<p>[追加 EJB コンパイラ オプション] フィールドが Administration Console に追加され、ヒープ サイズなどのオプションを Administration Console で <code>ejbc</code> に渡すことができるようになりました。</p>
CR085320	<p>Administration Console の「アイドル Bean 数」の値が増え続けなくなりました。</p>
CR085903	<p>メッセージ駆動型 Bean のアンデプロイメント時に <code>ejbRemove()</code> が適切に呼び出されるようになりました。</p>
CR088223	<p>バッチセッション時の Bean 管理の永続性ステートフルセッション Bean のシリアライゼーションに関する問題が修正されました。</p>

変更要求番号	説明
CR088526	WebLogic Server 6.1 SP3 では、EJB コンテナは「BMT を使用する場合、Tx はロールバックに設定されず、例外が送出される」という EJB 2.0 の規約に違反しています。このリリースでは、コンテナは例外をログに記録し（以前のバージョンでも機能）、Bean インスタンスを削除して（以前のバージョンでも機能）、トランザクションをロールバックとしてにマークします（以前のバージョンでは機能しない）。また、トランザクションはスレッドに関連付けられます。
CR090356	EJB コンテナがセキュリティ ロール情報を不適切に解析する問題が修正されました。
CR093776	weblogic-cmp-rdbms-jar.xml でカラムが LongString として定義されている場合に ejbc が不正な構文のコードを生成しなくなりました。

## サンプル

変更要求番号	説明
CR055626	webapp security サンプルの login.jsp ファイルでは、ユーザ ログイン時の URL エンコーディングが有効になりました。
CR069680	JdbcTable.jsp サンプルの問題を修正しました。ユーザ名とパスワードのデータベース接続プロパティを変数名のリテラルに設定しなくなりました。変数の内容が正しく使用されます。
CR085521	r3client Web サービス サンプルの package-summary.html ファイルが不完全だった問題を修正しました。ファイルには必要な情報がすべて含まれています。

## インストーラ

変更要求番号	説明
CR084521	コンフィグレーション ウィザードで、PetStore ドメインにコンフィグレーションの正しくない JMSJDBCStore が作成されなくなりました。
CR088287	コンソール モードの場合、コンフィグレーション ウィザードが管理対象サーバを作成するときに不正な SSL ポートを生成しなくなりました。

## インターナショナルライゼーション

変更要求番号	説明
CR074933	Administration Console のヘルプで、日本語の翻訳が表示されずに 404 または英語が表示されることがなくなりました。

## jCOM

変更要求番号	説明
CR076584	手動で介入しないと生成されたクラスがコンパイルされない com2java.exe の問題が修正されました。

# JDBC

変更要求番号	説明
CR062827	Prepared Statement コードで、null ポインタ例外が送出されなくなりました。問題は、Prepared Statement オブジェクトが null に設定されることでした。
CR072605	Prepared Statement キャッシュがデフォルトでオフになりました。デフォルトの Prepared Statement キャッシュ サイズは 0 です。
CR082011	weblogic.admin コマンドライン インタフェースで接続プールを作成しても、次の認証エラーが起こらなくなりました。 <pre>java.lang.ClassCastException: weblogic.security.service.PrincipalAuthenticator</pre>
CR083928	WebLogic Server JDBC は NLS_LANG パラメータ WE8MSWIN1252 をサポートします。
CR085559	Oracle のダウンが原因でユーザ トランザクションがロールバックされない場合に、トランザクションで使用される JDBC 接続が解放されていました。結果として、プールには使用可能な接続がありませんでした。この問題は解決済みです。
CR087803	WebLogic Server は Oracle 9.0.1 を使用する JMS Server に対してコンフィグレーション済みの JDBC ストアを作成できるようになりました。
CR087930	<p>以前は、Prepared Statement キャッシュ機能は、最初に <code>setLong()</code>、<code>setFloat()</code>、または <code>setDouble()</code> で使用された文のカラムに対する <code>setInt()</code> の呼び出しをサポートしていませんでした。また、最初は <code>setNull(i, BIGINT)</code> にバインドされていたカラムに対して <code>setInt()</code> を呼び出すと、「再解析カーソル」に関する <code>SQLException</code> が発生しました。</p> <p>JDBC Connection Pool MBean の新しい XA Prepared Statement キャッシュ (デフォルトで有効、デフォルト値は 5) を公開し、他にもコードを変更して、不足している機能を提供しました。</p> <p>[Prepared Statement キャッシュ サイズ] のデフォルト値も 0 から 5 に増加し、この機能がデフォルトで有効になりました。使用制限を含む詳細については、「<a href="#">Prepared Statement キャッシュのパフォーマンスの向上</a>」を参照してください。</p>

---

変更要求番号	説明
CR089185	<p>jDriver/Oracle XA および Oracle Thin XA を使用してトランザクションのアイソレーションを設定できるようになりました。以前は、設定しようとすると次のエラーが発生しました。</p> <pre>get exception: java.sql.SQLException: Due to vendor limitations, setting transaction isolation for "Weblogic OCI XA" JDBC XA driver is not supported</pre>
CR090378	<p>JDBC 接続プールを通じて連続的に Select を実行しても、OutOfMemory エラーが発生しなくなりました。</p> <p>WebLogic Server の以前のリリースでは、閉じられることなく破棄される JDBC オブジェクトがアプリケーション コードによって作成された場合に、そのオブジェクトは失われますが、ガベージコレクションが行われた後でもメモリが保持されるか、カーソルが開かれたままになっていました。そのようなオブジェクトがあまりにも多く作成された場合には、サーバが最終的にメモリ不足になり、データベースはカーソル不足になる可能性があります。</p> <p>WebLogic Server 7.0SP2 では、破棄された JDBC オブジェクトがガベージコレクションの前に閉じられるようにコードが修正されました。</p> <p><b>注意：</b> 破棄されたオブジェクトと同一の JDBC オブジェクトをすぐに作成する場合、WebLogic Server は元の JDBC オブジェクトのクローンを作成します。ただし、元のリーク オブジェクトが閉じられると、クローンも影響を受けます。</p> <p>『WebLogic JDBC プログラマーズ ガイド』の「<a href="#">JDBC オブジェクトを閉じる</a>」で説明されているベスト プラクティスに従うと、JDBC オブジェクトの破棄を回避できます。</p>

---

## jDriver

---

変更要求番号	説明
CR059097	<p>statement.cancel メソッドが WebLogic jDriver for Oracle と連携して動作するようになりました。</p>
CR068911	<p>ResultSet.getCharacterStream() が Oracle LONG 型で使用できるようになりました。</p>

---

---

変更要求番号	説明
CR078427	Oracle jDriver で <code>weblogic.jdbc.pool.Driver</code> を通じて <code>PreparedStatement</code> を使用する 場合、パラメータが適切にクリアされず、jDriver はデータベースに ( パラメータ の新しいデータと以前のデータが ) 結合されたデータを挿入していました。
CR078936	HP プラットフォームで、 <code>weblogic.jdbc.connectionPool.oraclePool</code> を 使用して Oracle データベースに接続しようとしても、SIGSEGV 11 HP エラーが 発生しなくなりました。
CR082484	Prepared Statement に 511 より多くのパラメータがある場合、jDriver for Oracle で <code>ArrayIndexOutOfBoundsException</code> が発生しなくなりました。制限は 512 から 1024 に変更されました。

---

## JMS

---

変更要求番号	説明
CR063743	メッセージ駆動型 Bean がオブジェクト メッセージの確認応答を行うようになり ました。メッセージはメッセージ リスト キューに適切に格納されるようになりま した。
CR077805	ヌル文字列のユーザ プロパティ — <code>setStringProperty (propertyName, null)</code> — で、プロパティ名が <code>Message.propertyExists ()</code> に反映されるよう になりました。
CR077898	<code>serverSession</code> プールを使用するときに <code>Message.getJMSDestination ()</code> が <code>null</code> を返さなくなりました。

---

---

変更要求番号	説明
CR078439	<p>以前のバージョンでは、クライアントとサーバの両方がクラスタ環境で同じ場所に配置される (サーバサイドテスト) 場合に、<code>JMSsession.commit</code> メソッドが以下のスタックトレースと同様の例外を送出しました (HP AS および Sol MS)。</p> <pre>EXCEPTION: java.lang.Error: transaction suspended twice Start server side stack trace: java.lang.Error: transaction suspended twice at weblogic.jms.dispatcher.Request.forceSuspendTransaction(Request .java:874) at weblogic.jms.dispatcher.Request.sleepTillNotified(Request.java: 200) at ...</pre> <p>この問題はこのリリースで修正されています。</p>
CR078702	<p>メッセージをリモートサーバにディスパッチするときに、<code>JMSDispatcher</code> スレッドにセキュリティ コンテキストに関する問題が発生しなくなりました。</p>
CR079944	<p>JMS キュー ブラウザの処理が修正され、メッセージ列挙全体が走査されていなくてもブラウザの終了時に関連メモリがクリーンアップされるようになりました。</p>
CR080301	<p>WebLogic Server 6.1SP2 では、<code>MessageListener</code> は <code>TransactedSession</code> がロールバックされた後にメッセージを受信しません。メッセージリスナは、一定のブロックのメッセージを処理し、一定の <code>rollback()</code> 呼び出しを実行した後で滞留する可能性があります。</p> <p>この滞留は JMS キューから 2 回目の一括フェッチが行われる際に発生する可能性があります。つまり、[最大メッセージ] パラメータが 10 の場合、最初の 10 メッセージは正しく処理されますが、メッセージリスナは 11 番目または 12 番目のメッセージで滞留します。[最大メッセージ] パラメータが 1 の場合、滞留は 2 番目のメッセージまたは 3 番目のメッセージで発生します。これは、キューのコミットにおける内部データ構造を適切に維持することで、現在のリリースで修正されました。</p>
CR081306	<p>恒久トピック サブスクリバ用の JMS メッセージの再配信のバッチ サイズが 10 に制限されなくなり、コンフィグレーション可能になりました。</p>
CR081429	<p><code>jdDriver</code> と Oracle 8.1.7. を使用して接続プールを作成し、このプールを使用して JMS 用の <code>JDBCStore</code> をコンフィグレーションし、<code>JDBC</code> を有効化したときに、<code>"java.sql.SQLException: Connection has already been closed"</code> エラーが送出されなくなりました。</p>

---



変更要求番号	説明
CR082826	JMS メッセージング ブリッジの終了、アンデプロイ、または停止時に恒久サブスクリプションが誤って解除されることがなくなりました。
CR083196、 CR088255	メッセージ キューを保持する送り先サーバが停止している場合に JMS メッセージング ブリッジがメッセージを削除することがなくなりました。
CR083290	クラスタ内のサーバの回復が改善され、サーバ間通知が適切に処理されるようになりました。ネットワークング レベル、RMI レベル、および JMS で変更が加えられました。
CR083794	ServerAffinity プロパティが False に設定されているトピックにメッセージを送信するときに NullPointerException が発生しなくなりました。
CR089114	<p>特定のクラスのアプリケーションの場合、WebLogic JMS はトピック サブスクライバ セレクタをインデックス処理することでそれらを大幅に最適化できるようになりました。これらのアプリケーションは、1 つまたは複数のユニークな ID と、それらの ID に基づいてフィルタ処理される何千ものサブスクライバを含むメッセージを持ちます。</p> <p>一般的な例としては、各サブスクライバが異なるユーザに対応し、各メッセージに 1 つまたは複数の対象ユーザのリストが含まれるインスタントメッセージングアプリケーションがあります。</p>
CR084175	MessagingBridge がメッセージを処理している間に WebLogic Server インスタンスが終了および再起動された場合、メッセージが失われなくなりました。
CR084182	MessageListener または ExceptionListener を Null に設定したときに NullPointerException が送出されなくなりました。
CR084374	メッセージング ブリッジが初期コンテキストを確立できなかったときに表示されるエラー メッセージにそのブリッジのパスワードが含まれなくなりました。
CR085179	JDBC ストアを閉じたときに未処理 JMS I/O 要求が削除されなくなりました。
CR085766	recover の後に ClearProperties を呼び出したときにメッセージ プロパティが削除されなくなりました。
CR086110	JMS セキュリティ ポリシーのメッセージ キュー参照がチェックされるようになり、匿名ユーザはメッセージ キューを参照できなくなりました。

## サービス パック 1 から 6 で解決済みの問題

変更要求番号	説明
CR086125	セレクト コードが最適化されたため、ユニークな単純セクタごとに多数のサブスクリバがある場合のパフォーマンスが 30% まで向上しました。
CR086413	分散送り先メッセージに不正な生成時間または期限切れ時間が含まれなくなりました。
CR086495	JMS Server のアンデプロイメント ( デプロイメントの削除 ) で InstanceNotFound 例外が送出されなくなりました。
CR086976	DB2 が動作する AS400 コンフィグレーションで JDBC ストアを作成できるようになりました。
CR087131	Redelivered フラグは、False に設定される必要があるときには True に設定されなくなりました。
CR087524	JMS DurableSubscriber の保留数がマイナスの値に設定されなくなりました。
CR088952	JMS クライアント プログラムが異常切断されたときに、サーバサイドの関連リソースが完全にクリーンアップされるようになりました。
CR089730	createTopic ("BackendName/TopicName") が正常に動作するようになりました。
CR090508	メッセージの replyTo が分散送り先に設定されているときにメッセージが誤ってキューに残らなくなりました。

## JNDI

変更要求番号	説明
CR083705	WebLogic Server 7.0 では JNDI ルックアップと EJB の使用に関して、パフォーマンスの低下が若干ありました。生成済みの JNDI 関連スタブを許可することでこの問題を修正しました。ejbc には、コンパイル中にスタブとスケルトンを生成する新しいフラグ (disableHotCodeGen) があります。
CR085463	WebLogic Server 6.1 SP3 では、ネットワーク接続が切断されていると、JNDI ルックアップのフェイルオーバーに長い時間がかかりました。これはこのリリースで修正されています。

## JTA

変更要求番号	説明
CR064403	XA driver + SSB を使用したトランザクション タイムアウト後に XAConnection がサーバの再起動まで使用できなくなる問題が修正されました。
CR073072	JTA トランザクションのタイムアウト値を 20,000,000 以上に設定したときに例外 (javax.transaction.SystemException: start () failed) が発生しなくなりました。
CR073649	トランザクションのタイムアウト後に XA 接続プールから接続を呼び出したときに XA-PROTO エラーが送出されなくなりました。
CR080755	Oracle XA シン ドライバに関するデッドロックの問題が修正されました。
CR090618	HP で JTA と JMS の "direct-write" ファイル I/O オプションを使用できるようになりました。

## JTS

変更要求番号	説明
CR064301	DBA は不明な分散トランザクションを手動で解決する必要がなくなりました。WebLogic Server が自動的に解決します。

## その他

変更要求番号	説明
CR042655	拡張ログ ファイル フォーマットで access.log を使用する場合、現地時間でなくすべて GMT で時刻が指定されるようになりました。
CR072188	サーバライフサイクル操作を実行するために管理サーバとノード マネージャ間で確立されるネットワーク接続がタスク完了後に閉じるようになりました。

## サービス パック 1 から 6 で解決済みの問題

変更要求番号	説明
CR073023	リモート サーバを起動しようとしたときにノード マネージャが OutputHandler エラーを送出しなくなりました。
CR068646	MBean は、HttpdEnabled が false に設定され、TunnelingEnabled が false に設定されていても HTTP でアクセスできなくなりました。
CR071796	サーバがデフォルト以外のディレクトリから起動できるようになりました。
CR079110、 CR083257	weblogic.Admin DELETE コマンドで WTCServer 実行時 MBean を削除するときに、削除の前に WTC がアンデプロイされませんでした。この問題は解決済みです。
CR079455	デプロイメントの失敗により ConfigurationError が発生して、サーバ プロセスが停止していました。この問題は解決済みです。
CR080016	アプリケーションが管理サーバではなく管理対象サーバを対象とする仮想ホストに割り当てられている場合、管理サーバの起動時に <code>java.lang.reflect.UndeclaredThrowableException</code> が発生する問題が修正されました。
CR080324	<code>WebLogicMBean.setParent()</code> は Javadoc でパブリック メソッドとして公開されています。
CR080740	NT サービスに正常停止機能が追加されました。beasvc に <code>-stopclass</code> オプションがあります。このオプションを設定すると、このクラスの <code>stop()</code> メソッドが呼び出されて停止を実行します。
CR082081	SERVERLOG コマンドライン コーティリティは、セキュアな T3 プロトコルが使用される場合に失敗なくなりました。 以前は次のようなコマンドでした。 <pre>Dweblogic.security.SSL.trustedCAKeyStore=%WL_HOME%\server\lib\cacerts -Dweblogic.security.SSL.ignoreHostnameVerification=true weblogic.Admin -url t3s://localhost:7502 -username system -password password SERVERLOG "2002/07/17 13.00" "2002/07/17 14.00"</pre> 次のようなメッセージが発生しました。Unable to get log file:unknown protocol: https
CR082910	カスタム MBean を EAR 内に登録する場合、JMX/MBean を使用して JDBC 接続プールの削除しようとしたときに、 <code>java.lang.ClassNotFoundException</code> で失敗することがなくなりました。

変更要求番号	説明
CR083220	新しい COMMO MBean をインスタンス化しようとしても、クライアントサイドで <code>javax.management.MBeanException</code> が送出されなくなりました。
CR083400	セキュアな T3 プロトコルを使用する場合に、管理対象サーバの起動がスタックオーバーフロー例外により失敗することはなくなりました。
CR084607	JMS トピックとキューが、クラスタ内で実行時に正しく作成されるようになりました。以前は、トピックは作成されたように見えてましたが、JMS サーバの送り先のリンクをクリックすると次のような例外が発生していました。 <code>java.lang.reflect.UndeclaredThrowableException</code> <code>javax.management.InstanceNotFoundException</code>
CR085315	AIX 4.3 において、管理サーバの実行中に管理対象サーバを起動したときに、WebLogic Server プロセスが <code>NullPointerException</code> により終了することがなくなりました。
CR085914	SNMP リクエストが、管理対象サーバの停止後にタイムアウトにならなくなりました。
CR086913	WebLogic Platform ドメインで、WebLogic Server は起動時に以下の 3 つのプロパティを認識するようになりました。 <code>weblogic.jws.ProductionMode</code> <code>weblogic.servlet.ClasspathServlet.disableStrictCheck</code> <code>weblogic.SystemDataStoreConfigDirectory</code>
CR088449	WebLogic Server は、起動時に以下のプロパティを認識するようになりました。 <code>weblogic.classloader.preprocessor</code> <code>weblogic.oci.selectBlobChunkSize</code> <code>weblogic.PosixSocketReaders</code>
CR090721	WebLogic java プロセスのファイル記述子数が、サーバ プロセスが不能になる程に制御できなくなることがなくなりました。
CR092471	ドメイン ログの時間に基づいたローテーションが、サーバの再起動時に失敗することがなくなりました。

## ノード マネージャ

変更要求番号	説明
CR071591	ノード マネージャは、HTTP 接続の開始を試みることで管理対象サーバの障害を検出することがなくなりました。現在は、ノード マネージャが接続障害を受信した場合、ノード マネージャの出力で接続例外が示されることがあります。
CR072188	タスクの完了後に、管理サーバはノード マネージャで確立した接続を正しく閉じるようになりました。以前は失敗し、サーバのファイル記述子の不足が生じました。
CR079205	ノード マネージャは <code>weblogic.policy</code> を正しく解析するようになりました。子プロセスのコマンドラインにプロパティをコピーするときに等号を追加しなくなりました。
CR081183	ノード マネージャがクラスタを強制停止するときに、競合状況が発生しなくなりました。この問題により <code>java.net.ConnectException:Connection refused: connect</code> エラーが発生していました。

## プラグイン

変更要求番号	説明
CR079186	すべてのプラグインは、HTTP 1.1 仕様に従って、折り返されたヘッダを正しく解析するようになりました。以前は折り返されたヘッダがあると、不正なヘッダを示すエラー メッセージが発生しました。
CR087997	Apache プラグインはセッション クッキーのプライマリ <code>jvmid</code> を正しく解析するようになりました。
CR089033、 CR088914 CR088915	証明書チェーンの攻撃やセキュリティ リスクに対するプラグインの脆弱性を修正しました。
CR079973	NSAPI で <code>Idempotent</code> が <code>Off</code> に設定されている場合、フェイルオーバは適切に無効になります。 <code>ConnectTimeoutSecs</code> を 0 に設定してもコア ダンプは発生しなくなりました。

変更要求番号	説明
CR080219	IPlanet は、すべての WebLogic Server インスタンスが停止された場合にコア ダンプしなくなりました。
CR080382	実際にはページが使用できる場合に「error page is unavailable」というメッセージは表示されなくなりました。
CR082093	負荷が大きい状態で、プロキシ プラグインがアプリケーション サーバに誤った余分な呼び出しを行っていました。iPlanet は呼び出しを受け取ると、呼び出しを 1 回だけアプリケーション サーバに送信するようになりました。
CR082096、 CR083386	ISAPI で <code>Idempotent</code> が <code>Off</code> に設定されている場合、フェイルオーバは適切に無効になります。
CR082113	以前は、ISAPI プラグインはクライアントからのデータの読み込み時にエラーを受け取ると、バックエンドの WebLogic Server に不完全なデータ (リクエスト) を送信しようとしていました。この状況は、プラグインが HTTP POST リクエストを読み込んでいるときに、エンド ユーザがブラウザの中止ボタンを押した場合などに発生していました。 送信中のデータが予定のバイト数に一致しないため、プラグインは <code>sendRequest</code> フェーズでエラーを受け取っていました。 この問題は発生しなくなりました。
CR082206	IPlanet では、プロキシからの一部の画像を除外して、パスによるプロキシを実行できる <code>WLExcludePathOrMimeType</code> フラグが用意されました。
CR082939	セッション トラッキングにクッキーではなく URLRewriting プロパティを使用する場合、セッションが維持されるようになりました。以前は、URLRewriting を設定してもセッションが正しくトラッキングされませんでした。
CR083643	NSAPI において、 <code>netbuf_getc()</code> が非常に遅く、POST データ内の ' <code>\0</code> ' と <code>0</code> を区別できないというパフォーマンスの問題が修正されました。
CR085192	ISAPI および NSAPI プラグインでは、KeepAlive 機能がデフォルトで有効になりました。
CR085922	NSAPI において、 <code>netbuf_getbytes()</code> が iWS4.x を使用する Java クライアントで失敗することがなくなりました。

---

変更要求番号	説明
CR086518	IPlanet において、クエリ文字列とアンカーの両方が含まれる URL を使用してリダイレクトを実行する場合、アンカーが誤って最後のクエリ文字列メンバーの値部分の一部と見なされる問題を修正しました。
CR089746	セッションのキャッシュと KeepAlive 機能は、セキュアなセッションに対して有効になりました。

---

## RMI/RMI-IIOP

---

変更要求番号	説明
CR079918	RMI オブジェクトをデータベースに格納するという試みが、動的プロキシ オブジェクトのシリアライズに問題があることで失敗することがなくなりました。
CR081265	クラスタへの呼び出し中に発生し、AssertionError として示されたレプリカリストの問題を修正しました。
CR081510	RMI オブジェクトの IOR における CSIv2 のサポートを示す実行時記述子は、デフォルトで Supported に設定されるようになりました。以前は、デフォルトで None に設定されていました。
CR081764	負荷が大きい状態で再起動するとサーバのハングを引き起こす、IIOP 初期化時のデッドロックを修正しました。
CR081923	IncompatibleClassChange エラーを発生させる、スタブ生成時のメモリ リークを修正しました。
CR083018	WebLogic Server で、間接的な型コードを使用して Any をアンマーシャリングできない問題が修正されました。これは IBM との相互運用性の問題を招いていました。
CR083936	WebLogic Server を使用して Orbix 2000 2.0 とのトランザクションを調整できない一連の問題を修正しました。
CR085703	例外的応答が発生する場合にリクエストが閉じてしまうことはなくなりました。
CR086300	サーバ コンテキストのレプリカ リストで、ドメイン名フィールド (WebLogic Server がサーバ名を配置する) が正しく設定されるようになりました。

---



変更要求番号	説明
CR086350	例外クラス名が大文字の「I」で終わる場合、生成された IDL が破損する問題を修正しました。
CR088777	循環参照は IDL の生成で正しく処理されるようになりました。以前は、クラスに循環参照が含まれている場合、-iioop -idl オプションを指定してクラスをコンパイルしようとするると失敗しました。
CR088778	TopLink で間接的な方法を使用しても、Assertion エラーは送出されなくなりました。
CR090010	weblogic.ejbcc が多次元配列に対する不正な IDL モジュール宣言を生成する問題を修正しました。

## セキュリティ

変更要求番号	説明
CR066491	<a href="http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA02-17.jsp">http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA02-17.jsp</a> のセキュリティ勧告を参照してください。
CR072154	sslclient サンプルの問題が以下のように修正されました。 <ul style="list-style-type: none"> <li>Package-Summary.html に空白のクラスが含まれなくなった</li> <li>WebLogic Server_home が Samples_home に置き換えられた</li> <li>MyListener クラスがキットに実装された</li> </ul>
CR072432	証明書データが Administration Console に表示されない問題が修正されました。
CR072954	SSL はアプレットで正常に動作するようになりました。
CR073187	TLS_RSA_WITH_DES_CBC_SHA 暗号スイートがエクスポート可能になりました。
082040	WebLogic 監査プロバイダのログ ファイルの場所が変更されました。監査はセキュリティ レベルムを対象にコンフィグレーションされますが、各サーバインスタンスはサーバ ディレクトリに存在する独自のログ ファイルに書き込みを行います。新しい格納場所は、 <code>WL_HOME\mydomain\myserver\DefaultAuditRecorder.log</code> です。

変更要求番号	説明
CR084170	<p>旧リリースでは、WebLogic Server は証明書チェーンの各証明書が認証局によって発行されたことを保証しませんでした。この問題は、誰かが信頼性のある認証局から個人用証明書を取得し、その証明書を使用して他の証明書を発行しても、WebLogic Server は無効な証明書を検出できないことを意味しました。このリリースでは、WebLogic Server で使用するすべての X509 V3 CA 証明書は CA として定義される基本制約拡張を備えている必要があります。このため、証明書チェーンのすべての証明書が認証局によって発行されたことが保証されます。デフォルトでは、この条件を満たしていない認証局の証明書は拒否されます。詳細については、「<a href="#">SSL 証明書検証</a>」を参照してください。</p>
CR085072	<p>weblogic.xml または weblogic-ejb-jar.xml ファイルでセキュリティ ポリシーを定義するには、ドメインに \lib ディレクトリを作成し、次に weblogic.policy ファイルで \lib ディレクトリにパーミッションを付与する必要があります。</p> <p>この確認済みの問題はこのリリースで修正されています。</p> <p>セキュリティ ポリシーの定義については、『<a href="#">WebLogic リソースのセキュリティ</a>』を参照してください。</p>
CR085203	<p>WebLogic Integration では、WebLogic Server との対話中に SSL セッションを再開する問題が発生しなくなりました。</p>
CR085805	<p>カスタム認可プロバイダのコード例は、その web.xml ファイルが不正であったため正常に機能しませんでした。このファイルは、url-pattern として */ を使用すべきときに * を使用してそのページを保護していました。</p> <p>更新されたコード例は、次の URL からダウンロードできます。</p> <p><a href="http://dev2dev.bea.com/code/">http://dev2dev.bea.com/code/</a></p>
CR086589	<p><code>javax.security.auth.login.LoginContext.logout()</code> メソッドは <code>LoginContext</code> のサブジェクトからプリンシパルおよび資格情報を削除しませんでした。このため、ユーザはログアウトされませんでした。</p> <p>この確認済みの問題はこのリリースで修正されています。</p>
CR086709	<p>保護されている JNDI リソースに、許可のないユーザはアクセスできなくなりました。</p>
CR088081	<p>階層的な WebLogic リソース (JMS、JDBC、サーバなど) は <code>getParentResource()</code> メソッドからのアクションを正常に処理するようになりました。</p>

変更要求番号	説明
CR088784	<p>JSP からステートフル セッション Bean を呼び出すと以下のセキュリティ違反が発生しました。</p> <pre>java.rmi.AccessException:Security Violation: user username has insufficient permission to access EJB</pre> <p>このエラーは、確立されたセキュリティ コンテキストが正常に閉じられないために発生しました。JSP の最後で <code>ServletAuthentication.runAs (Subject subject, HttpServletRequest request)</code> を使用すると、初期セキュリティ コンテキストが閉じられない場合にユーザが <code>HttpServletRequest</code> に追加されません。</p>
CR089059	<p>WebLogic Server は、KeyUsage 制約がクリティカルとして設定され、KeyEncipherment 制約が設定されていないデジタル証明書を拒否します。</p> <p>SSL V3.0 を使用する WebLogic Server の旧バージョンは、この状況でも証明書を拒否しませんでした。TLS V1.0 を使用する WebLogic Server のこのリリースは、この状況で証明書を拒否します。</p> <p><b>回避策:</b> パッチ CR089059_70sp1.jar を適用してください。</p>
CR090360	<p>ServerIron モニタ製品は、サーバの状態チェック中に要求を SSL ポートに送信します。この状態チェック中、SSLIOConect と関連 SSL オブジェクトが解放されず、メモリ不足エラーが発生しました。</p> <p>この確認済みの問題はこのリリースで修正されています。</p>
CR090401	<p>ユーザ グループに未認証ユーザは含まれなくなりました。</p> <p>この確認済みの問題はこのリリースで修正されています。</p>
CR091796	<p>IPlanet 認証プロバイダで動的グループが正常に動作するようになりました。</p>
CR091922	<p>WebLogic Server のサーブレット コンテナは、J2EE 仕様で指定されているとおりにログイン例外をエラー ページに受け渡しませんでした。</p> <p>この確認済みの問題はこのリリースで修正されています。</p>
CR092167、 CR085441	<p><a href="http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA03-29.jsp">http://dev2dev.bea.com/resourcelibrary/advisoriesnotifications/BEA03-29.jsp</a> のセキュリティ勧告に関する情報を確認してください。</p>
CR077050	<p>クラスタ内のレプリケートされた LDAP データベースにユーザを追加した場合、そのユーザが管理対象サーバにレプリケートされるのに非常に長い時間がかかる問題が修正されました。</p>

## サーブレットと JSP

変更要求番号	説明
CR065967	<code>type</code> オプションが指定され、型が配列であるときに (型が文字列として扱われるため) タグ ライブラリが失敗する問題が修正されました。
CR068577	生成されたコードで <code>method_releaseTags()</code> が <code>jsp:forward</code> の一部として呼び出されない問題が修正されました。JSP の本文中で転送が発生するとタグがリリースされるようになりました。
CR070417	サーブレットが HTTP/1.1 でアプレットにメッセージを送信すると、すべてのメッセージが送信されるまでアプレットがハングする問題が修正されました。これは、サーブレットが送信したメッセージをアプレットが並列受信しなかったからです。
CR075419	「%」が含まれている URL が正常にロードされるようになりました。
CR044926、 CR075799	2 バイト プラットフォームで、「}」を含む JSP ファイルから Java コードを生成できるようになりました。
CR072474	<code>jsp:usebean</code> を使用する場合、 <code>jspc</code> は非推奨のメソッドを含んだクラス ファイルを生成しなくなりました。
CR075668	JSP がルート ディレクトリ以外のディレクトリに格納され、相対パス名で表現されている場合に <code>jspc</code> が正常に機能するようになりました。
CR078551	<code>jsp:include</code> が存在しない JSP を参照すると、 <code>FileNotFoundException</code> が送出されるようになりました。
CR081353	<code>javax.servlet.jsp.PageContext.out</code> 属性は本体コンテンツ タグ用に正常にクリーンアップされ、出力が正常にブラウザに送信されるようになりました。
CR082310	<code>page contentType</code> ディレクティブを含む JSP を展開 Web アーカイブ (WAR) で削除および上書きできるようになりました。
CR083227	JSP ファイルに <code>jsp:setProperty</code> プロパティが含まれ、このタグが参照する Bean に属性 <code>String []</code> として多次元配列が含まれる場合、 <code>jspc</code> はコンパイル可能コードを生成できるようになりました。
CR083597	<code>page contentType</code> ディレクティブを使用する場合、コンテンツ タイプは正しく設定されるようになりました。

変更要求番号	説明
CR83804	WebLogic Server では、応答がクライアントにコミットされない限り、ユーザがバッファ サイズをリセットできるようになりました。7.0 および 7.0SP1 では、バッファがコミットされていない場合、サーバは <code>illegalStateException</code> を送出し、 <code>setBufferSize</code> が <code>ServletOutputStream.write()</code> の後に呼び出されました。これにより、バッファをフラッシュする前の <code>setBufferSize</code> の呼び出しに依存するサブレット / フィルタで問題が発生しました。
CR083848	別の JSP からのコンテンツを含めるときに JSP 内の日本語が文字化けしないようになりました。
CR084593	JSP は <code>contentType</code> が不正な文字セットに設定されていることを正確に検出できるようになりました。
CR085702	JSP コンパイル エラーの行番号がエラー メッセージと正確に一致するようになりました。
CR086908	<code>jspc</code> 解析エラー メッセージがより詳細になり、誤った記述子を素早く発見できるようになりました。
CR087090	<code>jspc</code> が同一のタグ ライブラリを使用する複数の JSP を呼び出す方法により、 <code>jspc</code> 実行時間が効率化されました。
CR088019	共通の一時ディレクトリを共有するために複数の仮想ホストのデプロイ済みリソースが競合する問題が修正されました。これを解決するために、一時ディレクトリのパスにサーバ名を追加しました。  この副作用として、 <code>compile-on-startup</code> JSP クラスが管理サーバから管理対象サーバにデプロイされなくなります。しかし、これは意図的な結果ではありません。適切にデプロイするには、 <code>weblogic.jspc</code> を使用してサーバの起動前にこれらのクラスを事前にコンパイルし、 <code>WEB-INF/classes</code> ディレクトリにデプロイします。
CR088301	インクルードされたリソース (「リソース A」) が、JSP <code>BodyTag</code> の内部で <code>Forward</code> ディスパッチされた別のリソース (「リソース B」) からディスパッチされたとき、サブレットの出力ストリームに送信されない問題が修正されました。
CR088525	<code>jsp:include</code> は <code>jsp:param</code> (値が <code>"/////"</code> の場合) を適切に転送するようになりました。
CR092089	Web アプリケーションに複数バイト名のファイルが存在する場合、WebLogic Server がそれをコンパイルせずディレクトリの内容を返す問題が修正されました。

## サービス パック 1 から 6 で解決済みの問題

変更要求番号	説明
CR093291	クラスパスの問題を解決したことにより、JSP をコンパイルするときに .jar ファイルを適切に発見できるようになりました。
CR061493	URL が JSP 名前 / 値パラメータとして指定されて JSP が含まれた場合、その URL は異なるエンコード体系が指定されたときに削除されなくなりました。
CR070090	WebLogic Server はキャッシュ プロキシがクッキーをキャッシュしないように <code>cache-control</code> を適切に指定するようになりました。
CR074840	<code>ppath</code> と <code>pathtrim</code> を持つプロキシがクライアントと WebLogic Server 間で使用され、フォームベースの認証が使用される場合、クライアントがプロセスにリダイレクトされない問題が修正されました。
CR077780	WebLogic Server はサブレット要求のディスパッチ後にクリーンアップとキューへの再登録を適切に行って、メモリ リークを防ぐようになりました。
CR078698	<code>access.log</code> ファイルはサイズに基づいて正確にローテーションされるようになりました。
CR079014	次に示す新しい <code>setTimeout()</code> メソッドを使用して、HTTP 永続接続の存続期間をクライアントサイドでコンフィグレーションできるようになりました。 <pre>URL u = new URL(url); URLConnection hu = new HttpURLConnection(u); hu.setTimeout(5*1000); BufferedReader in = new BufferedReader(new InputStreamReader(hu.getInputStream()));</pre>
CR079767	<code>applications/.wlnotdelete</code> ファイルに格納されていた一時ファイルは、不要になったとき (アプリケーションが再デプロイまたは削除されたとき) に削除されるようになりました。
CR80090	定義されている場合、 <code>FrontendHTTPPort</code> 、 <code>FrontendHTTPSPort</code> 、および <code>FrontendHost</code> はリダイレクトされる URL を作成するときに HOST ヘッダに優先するようになりました。
CR080384	不正なステータス行があるリクエストで検出されたときに <code>HttpClusterServlet</code> がリクエストをサーバに転送するのを停止する問題が修正されました。
CR080613	クッキー内の引用符で囲まれた値でカンマが正しく解析されない問題が修正されました。

変更要求番号	説明
CR080751	アプリケーションが <code>ServletException</code> を拡張するカスタマイズされた例外を送出するときに誤ったメッセージが表示される問題が修正されました。
CR080855	<code>absolutURI</code> を含む HTTP リクエストが適切なコンテキストパス、リクエスト URI、およびリクエスト URL で実行されるようになりました。
CR080903	デプロイメント時に Web アプリケーションが 2 度展開される問題が修正されました。
CR081071	不正なプロセスが <code>NullPointerException</code> を送出せずに適切に終了するようになりました。
CR081253	<code>TransferEncoding</code> が <code>Chunked</code> に設定されている HTTP リクエストをサーバが適切に解析するようになりました。
CR081418	<code>PrintNulls</code> オプションが <code>weblogic700-web-jar.dtd</code> から失われなくなりました。
CR081484	<code>HttpSessionListener</code> を使用し、 <code>PersistentStoreType</code> が <code>replicated</code> に設定されているときにセッション属性が失われなくなりました。
CR081521	<code>.ear</code> または <code>.war</code> ファイルの JSP から CGI スクリプトを呼び出すときに CGI 出力が正確に表示されるようになりました。
CR082156	<code>isRequestedSessionIdValid()</code> で常に値 <code>true</code> が返されていました (つまり、URL でセッション ID を変更してもリクエストセッション ID が有効となっていました)。無効なセッションが適切に検出されて無効としてレポートされるようになりました。
CR082580	URL にアクセスした場合、フォームベースの認証が起動され認証が成功したときに HTTP POST パラメータが適切に保存されるようになりました。
CR083191	Administration Console にサーブレット実行時間の無効な値が表示される問題が修正されました。
CR083487	<code>&lt;url-pattern&gt;</code> が完全一致したときに <code>CGIServlet</code> が動作するようになりました。
CR083517	<code>HttpClusterServlet</code> は、 <code>SecuryProxy</code> が設定されているときにリクエストを SSL に適切にプロキシするようになりました。
CR083624	HTTP 1.0 リクエストがヘッダで HTTP 1.1 応答を適切に返すようになりました。

## サービス パック 1 から 6 で解決済みの問題

変更要求番号	説明
CR083912	Oracle Thin Driver 8.1.7.3 を使用するとき JDBC セッションの永続が失敗する問題が修正されました。
CR084002	仕様に従って、204 HTTP 応答にメッセージ本文が含まれなくなりました。
CR084076、 CR086280	Keepalive 接続でサブレットにアクセスしているときに不正なリクエストが作成される問題が修正されました。
CR084165	HTTP セッション トラッキングは、PersistentStoreType が jdbc に設定されているときに維持されるようになりました。
CR084167	応答ラッパーを使用する場合、取り込みまたは転送によって JSP の出力を別の JSP に委託するときに <code>IllegalStateException</code> が送出されなくなりました。
CR084649	WebLogic の CGI サーバは、その中で実行するスクリプトに対する <code>SERVER_URL</code> 、 <code>HTTP_COOKIE</code> 、および <code>QUERY_STRING</code> 環境変数を提供するようになりました。
CR084785	<code>GenericProxyServlet</code> はラップされた応答を処理するようになりました。
CR084958	<code>URLConnection</code> ハンドラが <code>getErrorStream()</code> を実装するようになったため、エラー ページからカスタム応答を読み取ることができるようになりました。
CR085735	<code>URLRewritingEnabled</code> が <code>false</code> に設定されている場合にも <code>JSESSIONID</code> が URL にエンコードされてしまい、404 エラーが発生する問題が修正されました。
CR085843	<code>ejb-local-ref</code> が Web アプリケーションに対して動作するようになりました。
CR086026	<code>CacheSessionCookie</code> のデフォルトが <code>true</code> となり、最初の表示で問題が発生しなくなりました。
CR086052	発信 (WebLogic Server から) HTTPS 接続で後続のスラッシュ '/' が不要になりました。
CR086099	Java 1.3.1 と Java 1.4.0 間の <code>java.net.Socket.isConnected()</code> の動作の変更から発生した <code>NullPointerException</code> バグが修正されました。
CR086416	サブレット エンジン、コンソール ページをロードしようとするときに <code>SocketWriteError</code> を送出しなくなりました。



変更要求番号	説明
CR086481	Web アプリケーションがファイルをアップロードしようとしたときに <code>ArrayIndexOutOfBoundsException</code> 例外が発生しなくなりました。コードを再作成して、バッファ オーバーフロー、およびバッファが一部しか読み込まれないケースに対処しました。
CR086677、 CR087573、 CR088528	JDBC 永続性に関して、後続の要求がセッションの永続性の完了前に許可され、データの損失が発生する問題が修正されました。
CR087647	<code>URLConnection</code> のタイムアウトが正確に行われるようになりました。
CR087984	サーバの停止時に <code>ServletContextListener</code> を <code>HttpSessionListener</code> の前に呼び出すと予期しない結果が発生する問題が修正されました。
CR088166	サーバがより正確に XML 終了タグを検出するようになり、例外を回避できるようになりました。
CR088478、 CR086490	クラスタ化サーバ用の WebLogic Server プラグインで <code>MaxSkips</code> 機能が非推奨になり、代わりに <code>MaxSkipTime</code> を使用するようになりました。 <code>MaxSkipTime</code> は、プラグインが「bad」（障害が発生したサーバを表す）とマークされたサーバへの接続を再試行するまでの時間を設定します。
CR089229	<code>HttpClusterServlet</code> 呼び出しでパフォーマンスが 10 パーセント低下する問題が修正されました。
CR089923	<code>jspc</code> が現在のディレクトリにファイルを作成できない問題が修正されました。
CR090665	サブレット フィルタは、応答をクライアントに送信するために <code>chain.doFilter()</code> の呼び出し後に応答ラッパーの <code>flushBuffer()</code> を明示的に呼び出す必要がなくなりました。
CR091699	<code>response.sendRedirect()</code> は、絶対 URL の代わりに相対 URL にリダイレクトするようになりました。
CR091878	Web アプリケーションのアンデプロイによって <code>IndexOutOfBoundsException</code> が発生することがなくなりました。
CR092778	<code>HttpRequest</code> 用の <code>JISAutoDetect</code> エンコードで例外が発生する問題が修正されました。

---

変更要求番号	説明
CR095166	iisproxy のクッキー解析が修正されました。クライアントが複数のクッキーを送信するときにプロキシはプライマリ サーバに正確に留まるようになりました。
CR080778	セッション モニタを有効にしても、トラフィックが増加して管理サーバが新しいセッションを作成するたびにハングすることはなくなりました。

---

## ツール

---

変更要求番号	説明
CR073688	WebLogic Builder では、[ サブレット / フィルタ マップ   フィルタ マッピング   追加 ] で、サブレット名と URL パターンの両方の値を入力できるようになりました。
CR073715	.jar ファイルに複数のコンテナ管理の永続性 (CMP) 記述子ファイルが含まれている場合、WebLogic Builder にすべての記述子ファイルが表示されるようになりました (これまででは最初に発見したファイルしか表示されなかった)。
CR073877	WebLogic Builder がコンテナ管理の永続性エンティティ Bean を作成する場合、<trans-attribute> 値のデフォルトとして Required が使用されます。
CR074052	WebLogic Builder は enable-call-by-reference のデフォルト値として True を正しく表示するようになりました。
CR074059	WebLogic Builder に、デプロイメント記述子をユーザが指定した場所にエクスポートする機能が追加されました。
CR074208	WebLogic Builder で resultTypeMapping を設定する場合、新しいファインダ / 選択メソッドを追加すると正しく機能するようになりました。
CR074246	WebLogic Builder がアーカイブを最初に展開するときにタイムスタンプを保持できない問題を修正しました。この問題によって、EJB コンテナが誤ってすべての JSP をあらかじめコンパイルする場合があります。
CR074247	WebLogic Builder で、既存のファインダに含まれているのと同じメソッド名を持つファインダを追加するとツリーは更新されるがテーブルは更新されない問題が修正されました。

---

変更要求番号	説明
CR074602	WebLogic Builder はクラスを検証する前に再ロードするようになりました。
CR074603	WebLogic Builder はクラスをデプロイする前に検証するようになりました。
CR074634、 CR074955、 CR074972	WebLogic Builder の関連ウィザードを使用して既に複数の関連がコンフィグレーションされている CMP の関連を編集しても例外は送出されなくなりました。
CR074639	WebLogic Builder の編集パネルを使用してファインダ メソッドの名前を編集した場合、その変更が XML に正しく伝播されるようになりました。[ファインダ] ポップアップ ダイアログを使用して編集した場合、その変更は XML にも編集パネルにも正しく伝播されます。
CR074876	WebLogic Builder は [ オプション ] ダイアログでサーバの接続情報を保存し、正しく伝播するようになりました。 以前は [ オプション ] ダイアログの値は [server connect] ダイアログに表示される値と関連がありませんでした。修正によって、その情報は [ オプション ] 画面で入力された値と同期するようになりました。
CR075005	WebLogic Builder でコンポーネントを開くときに無効なドライブを指定しても ( テキスト ボックスに無効なドライブを入力して [Enter] を押し、[開く] を選択する )、NullPointerException 例外が発生しなくなりました。
CR075006	WebLogic Builder は 1 対 多の関係を正しく作成するようになりました。
CR080967	WebLogic Builder は weblogic-cmp-rdbms-jar.xml file を正しく解析および編集するようになりました。
CR081296	NullPointerException を発生させる DDInit の問題を修正しました。次のコマンドを発行します。 java weblogic.marathon.ddinit.EJBInit project.jar where project.jar is a well-formed ejb jar no longer provokes this error: Exception in thread "main" java.lang.NullPointerException
CR081483	WebLogic Builder では、verify-column の値が設定されていない場合、同時方式をデフォルトで Read に設定してエラーを防ぎます。
CR082562	WebLogic Builder は use-select-for-update をサポートするようになりました。

変更要求番号	説明
CR083151	コマンドラインの DDInit は EAR ファイルに対して機能するようになりました。
CR084570	DDInit は EAR ファイルの application.xml に加えて weblogic-application.xml を作成するようになりました。
CR085493	EJBJGen に global-role タグが追加されました。
CR086073	サーバ ダイアログを取り消しても、WebLogic Builder はエラー メッセージを誤って表示しなくなりました。
CR086546	DDInit は、同等でなければならぬ 2 つの関係が同等でないことを正しく判別するようになりました。
CR089867	WebLogic Builder は dbms-column-type の LongString および SybaseBinary タイプをサポートするようになりました。

## WebLogic Tuxedo Connector

変更要求番号	説明
CR077476	WTC トランザクションが TPEV_SENDFONLY の不正な処理のために誤ってロールバックされる問題が修正されました。
CR078774	Tuxedo /Q からメッセージを取得する接続を使用して、WTC/tBridge によってメッセージを処理しても、一定時間の経過後にメッセージの取得が停止しなくなりました。
CR082312	TPNOBLOCK フラグが設定された tpcall が失敗しなくなりました。
CR084435	接続ポリシーが ON_DEMAND または ON_STARTUP に設定された場合、フェイルオーバーおよびフェイルバックの状況で発生する問題を修正しました。
CR084903	WebLogic Tuxedo Connector simpappcns サンプルはマルチプラットフォーム環境をサポートするようになりました。

## Web サービス

変更要求番号	説明
CR070387	データ型の非標準のマッピングが無視されなくなりました。
CR073899	WebLogic Web サービスは、SOAP 障害の生成時に余計な非 CDATA 文字を追加しなくなりました。
CR076625	ユーザは、デプロイメント記述子要素で portTypeName を指定できるようになりました。
CR076706	XML ノードの余分なスペース文字の解析に関する問題が修正されました。
CR077855	Java クライアントが Web サービスにアクセスするときに NullPointerException を発生させる不正な content-type 設定が修正されました。
CR078115	Administration Console で [ 個別のサービスのポリシーとロールを定義 ... ] を選択したときに NullPointerException が送出されなくなりました。
CR081938	ハンドラを使用せずに添付ファイル付き SOAP を処理する Web サービス例が追加されました。
CR082626	パフォーマンスを向上させるシリアライゼーションが効率化されました。
CR082779	HttpsURLConnection が SSLContext から ID を正確に取得するようになりました。
CR083278	JAX-RPC クライアントスタブ生成が正常に機能しない問題が修正されました。
CR083410	カスタム SoapfaultException が正確に詳細フィールドを埋め込むようになりました。
CR083813	JavaBean 配列型が適切に処理されるようになりました。
CR083919	重複要素を持つ継承型に対する XML データ バインドが失敗しなくなりました。
CR084168	WSDL 生成で再帰型が処理されるようになりました。
CR084220	WebLogic 動的クライアントがスタイル style="document" WebService を呼び出せるようになりました。

変更要求番号	説明
CR084591	シリアライゼーションとデシリアライゼーションでの月と日の形式の処理が修正されました。
CR085214	日付をパラメータとして渡すと <code>java.Lang.IllegalArgumentException</code> が発生する問題が修正されました。
CR085246	XSD 列挙が WSDL 生成で認識されるようになりました。
CR085888	アノテーション ドキュメントを含むスキーマ インポートを持つ WSDL が失敗しなくなりました。
CR086231	生成される WSDL で重複する例外メッセージが発行される問題が修正されました。
CR087883	<code>xsd:base64Binary</code> 型の要素を持つ大きな値について、Web サービス応答でコンテンツ長が正確に設定されるようになりました。

## XML

変更要求番号	説明
CR080961	<code>org.xml.sax.helpers.AttributesImpl.removeAttribute()</code> の呼び出しが SAX エラーで失敗する Xerces1.3.1 の問題を修正しました。

---

変更要求番号	説明
CR081372	<p>DocumentBuilder.parse() は、入力エンコーディングが Unicode の場合、マルチスレッド セーフになりました。</p>
CR088349	<p>Java Web サービス クライアントから WebLogic Workshop で生成された Web サービスへの要求で、UTF-8 または UTF-8 のサブセットではない文字エンコーディングを使用すると、次のクライアントサイド例外により失敗しました。</p> <pre data-bbox="364 453 1110 479">javax.xml.rpc.soap.SOAPFaultException: Invalid request</pre> <p>上記のサーバサイド例外：</p> <pre data-bbox="364 531 1224 583">java.io.CharConversionException: Malformed UTF-8 char -- is an XML encoding declaration missing?</pre> <p>調査により、クライアントでは XML 宣言ではなく HTTP ヘッダでのみ文字エンコーディング情報が指定されていたことがわかりました。Workshop Web サービスは XML 宣言のエンコーディングに従って動作します。指定されていない場合は UTF-8 になります。リクエストに有効な UTF-8 文字ではない文字が含まれている場合、エラーが発生します。</p> <p>クライアントは、HTTP ヘッダと XML 宣言の両方にエンコーディング情報を含むように変更されました。その修正で、問題は解決されました。</p>

---

# WebLogic Server 7.0 サービス パック 1 のソリューション

以下の節では、WebLogic Server 7.0 サービス パック 1 で解決された問題を説明します。

- [4-349 ページの「コンソール」](#)
- [4-350 ページの「コア サーバ」](#)
- [4-352 ページの「デプロイ」](#)
- [4-353 ページの「EJB」](#)
- [4-356 ページの「サンプル」](#)
- [4-357 ページの「インターナショナルライゼーション」](#)
- [4-357 ページの「JDBC」](#)
- [4-358 ページの「JMS」](#)
- [4-359 ページの「JNDI」](#)
- [4-359 ページの「JTA」](#)
- [4-359 ページの「JTS」](#)
- [4-359 ページの「その他」](#)
- [4-365 ページの「RMI/RMI-IIOP」](#)
- [4-367 ページの「セキュリティ」](#)
- [4-371 ページの「サーブレットと JSP」](#)
- [4-375 ページの「ツール」](#)
- [4-377 ページの「WebLogic Tuxedo Connector」](#)
- [4-377 ページの「Web サービス」](#)



## コンソール

変更要求番号	説明
CR068366	コンソールのコンテキストパスの変更が、コンソールアプリケーションのデプロイに適切に影響するようになりました。
CR069887	load.mlet ファイルが myserver ディレクトリで作成される問題が修正されました。 この問題は、weblogic.Server を使用して myserver とは異なるフォルダに新しいドメインを作成すると発生しました。load.mlet ファイルは削除され、myserver ディレクトリは作成されなくなりました。
CR076739	ローカルステートレスセッション EJB デプロイメントの JNDI ツリーを表示するときに [オブジェクトクラス]、[オブジェクトハッシュコード]、および [文字列へのオブジェクト] フィールドが空白になる問題が修正されました。
CR077126	コンソールでサブレットをモニタできるように weblogic.xml param ConsoleMain Attribute をコンフィグレーションできない問題が修正されました。
CR081366	1 つまたは複数のサーバがコンフィグレーションされているマシン オブジェクトをユーザが削除できない問題が修正されました。
CR082942	コンソールでユーザのパスワードが変更されて [続行] がクリックされたときに Null ポインタ例外が送出される問題が修正されました。
CR083460	HP_UX での Netscape のサポートに合わせて WebLogic Server が更新されました。
CR083578	[Server Protocols] タブの [メッセージの最大サイズ] テキストフィールドのサイズが大きくなり、入力されたすべての桁が表示されるようになりました。
CR084114	[WebApp Monitoring] タブで、Web アプリケーションのパフォーマンスをモニタしているときに例外が送出される問題が修正されました。

## コア サーバ

変更要求番号	説明
CR055987	URLConnection のタイムアウト オプションが実装されました。
CR066230	ユーザが名前属性 (ドメイン名に合わせて config.xml ファイルで設定) を設定できるようになりました。config.xml ファイルを手作業で編集し、ドメイン名とは異なる FileName 属性を指定してドメインに Log 要素を追加した場合、サーバはドメインと同じ名前のドメイン ログを作成して使用していました。
CR074835	java.net.HttpURLConnection の変更に対応するように WebLogic Server が変更されました。
CR074888	サーバが 2 つのクラスタでプライマリ サーバが停止されようとしているときに HttpSession のフェイルオーバでエラー 503 を生じさせる問題が修正されました。
CR075321	Operator ロールで weblogic.Admin コマンドを使用して START コマンドを発行できない問題が修正されました。NoAccessRuntimeException が発生していました。
CR077831	javax クラスをローカルでロードできます。その結果、AppletArchiver ユーティリティを使用してクライアント JAR ファイルを生成するときに発生した問題が修正されました。問題のアプレットのコードで java.swing.JApplet が使用される場合に、アプレット ビューアで output.jar. java.io.FileNotFoundException エラーが生成されていました。
CR077919	オブジェクトが StubInfo 型で参照が LocalServerRef である場合に、元のオブジェクトと参照を LeasedRemoteReference に変更させるメソッド replaceObject() の問題が修正されました。
CR079738	大きなメモリ構造の参照が静的である場合に発生した問題が修正されました。この問題を修正するために、クラスローダの参照 WebAppComponent および WebAppServletContext に NULL が割り当てられました。
CR080324	setParent がパブリック API にエクスポートされていないことが原因で JMS サーバ上の送り先 MBean の配列から新しい送り先 MBean を呼び出すときにエラーを発生させた問題が修正されました。

変更要求番号	説明
CR080779	セッションのモニタを有効にした後、セッションが作成されるたびに管理サーバがハングする問題が修正されました。
CR081193	WebLogic Server が数時間動作した後に stdout へのロギングが停止する問題が修正されました。例外は送出不されず、サーバは動作を続けていました。さらに、Windows 環境では [Ctrl] + [Break] の機能が停止し、Unix プラットフォームでは kill -3 でのスレッド ダンプの生成が停止していました。
CR083485	t3 をデフォルト プロトコルとして使用した場合に発生する問題が修正されました。この問題は、t3 クライアントを JavaSocketMuxer でハングさせていました。この修正の結果、t3 クライアントはデフォルト接続にブートストラップする代わりに適切な受信時接続を使用します。

# デプロイ

変更要求番号	説明
CR070498	実際のアプリケーションを削除していない場合には、ユーザはアプリケーション MBean を削除できなくなりました。
CR080401	複数のコンテキスト ルートが正しく機能するのを妨げていた問題が修正されました。
CR080914	アプリケーション スコープの接続プールでリソース記述子を使用できるようになりました。その結果、WebLogic Server では、 <code>resource-ref-name</code> 要素で JNDI 名が定義されているかどうかを確認し、ローカル レベルでルックアップを実行して、DataSource が存在する場合にそれを <code>comp/env</code> にバインドすることができるようになりました。
CR081311	<code>web-app.xml</code> ファイルの DTD のチェックで、適切な警告重大度レベルではなく情報重大度レベルの <code>SAXParseException</code> が生成される問題が修正されました。
CR082195	<code>weblogic.Deployer</code> により、クラスタに利用できないメンバーがある場合でも、管理者がクラスタの利用可能なすべてのメンバーを正常にデプロイできるようになりました。
CR082263	Windows 2000 システムから Solaris システム上のサーバにデプロイされた Web アプリケーション WAR ファイルを、 <code>weblogic.Deploy</code> で更新できない問題が修正されました。
CR083179	Web アプリケーションの <code>web.xml</code> および <code>weblogic.xml</code> の <code>resource reference</code> デプロイメント記述子に、同期していない <code>resource-ref</code> 要素および <code>resource-description</code> 要素がある場合に表示される WebLogic Server 例外メッセージが改善されました。以前は、表示される例外メッセージに、問題を生じさせた実際の <code>resource reference</code> 名が含まれていませんでした。

## EJB

変更要求番号	説明
CR050001	マルチバイト文字の EJB-QL サポートが提供され、句 '=' および 'IN' で固定漢字文字列を処理できない EJB-QL の問題が修正されました。
CR057074	beans-in-use-number の値が beans-in-cache-number より大きくなる問題が修正されました。この情報は、コンソールのモニタ セクションに表示されます。
CR060965	パーサは、EJB QL クエリで記号 '=' の前後にスペースを許可するようになりました。
CR073297	メッセージ駆動型 Bean が送り先に再接続しようとしても失敗する問題が修正されました。再接続が失敗すると、そのたびにログ メッセージが作成されます。再接続は 10 秒ごとに試行されるので、修正前はログ ファイルがすぐに一杯になっていました。
CR074781	PointBase データベースでテーブルの自動生成が機能しない問題が修正されました。
CR075219	ロールとポリシーが EJB で施行されるようになりました。
CR075273	WAR ファイルの DDInit で weblogic.xml ファイルが出力または作成されない問題が修正されました。web.xml ファイルは出力されましたが、WEB-INF ディレクトリには作成されませんでした。
CR077104	Required トランザクション属性を持つ読み込み専用エンティティ Bean 間の EJB 2.0 コンテナ管理の関係 (CMR) の 1 対多の関係で、コレクションにアクセスしようとしたときに <code>IllegalStateException</code> が生成されなくなりました。
CR077986	EJB 2.0 <code>ejb.jar.xml</code> ファイルにおいて、ワイルドカード「*」がメソッドパーミッションの <code>&lt;unchecked/&gt;</code> で機能するようになりました。
CR078297	同時方式をオブティミスティックに設定した場合にコンテナ管理による永続性 (CMP) のキャッシングが期待通りに機能するようになりました。以前は、キャッシュが更新される代わりに、2 回目のトランザクションで <code>ejbLoad</code> が呼び出されている、つまりデータベースが外部の変更で更新されていたものと考えられます。

変更要求番号	説明
CR078350	生成されたクエリが適切でないためにクエリの生成で CTS 障害が生じる問題が修正されました。
CR079086	JNDI 情報に LDAP サーバを使用するメッセージ駆動型 Bean が、 <code>javax.management.MalformedObjectNameException</code> でデプロイに失敗することがなくなりました。
CR079471	SelectDistinct が動的クエリで機能するようになりました。EJB QL select 句に SelectDistinct のある動的クエリでは重複が返されていました。Select Distinct を使用する場合、EJB コンテナは重複した結果を排除します。
CR080115	ユーザが CMP Clob 値を出力したときに null 値が返されなくなりました。
CR080783	EJB でグローバル セキュリティ ロールが確実にマークおよび処理されるようになりました。role-name 要素にグローバルに指定されたプリンシパルが割り当てられる場合は、global-role 要素を <code>weblogic700-ejb-jar.xml</code> ファイルの <code>security-role-assignment</code> スタンザで定義する必要があります。
CR080920	完全委任のセキュリティ チェックを可能にする実行時フラグが追加されました。 以前は、セキュリティ チェックは実行されませんでした。JDBC リソースに対してコンソールで設定されたポリシーは有効にならず、不適切な JDBC リソースがコードによって設定され、新しいプールを作成するための管理リソースを設定する手段がありませんでした。
CR081398	同時方式が排他的で、トランザクション間でキャッシュを使用する Bean 間に 1 対多の関係を持つ EJB 2.0 CMP Bean で <code>illegalException</code> が生成されなくなりました。
CR081807	<code>home</code> を JNDI ツリーにバインドする前に <code>ejb.Deployer</code> がコンテキストオブジェクトで <code>replicatebindings</code> を <code>true</code> に固定することが原因で発生した問題が修正されました。この問題が原因で、WebLogic Server がクラスタの他のノードに通知を送信しようとすると、 <code>NameAlreadyBoundExcedption</code> が発生していました。

変更要求番号	説明
CR082406	WebLogic Server を Sybase データベースおよび Sybase XA ドライバと使用した場合で、EJB の CMP フィールドがバイナリ有的时候にドライバから暗黙の変換エラーが送出されなくなりました。
CR082451	JMS プロバイダが WebLogic 固有でない状況でメッセージ駆動型 Bean のサードパーティ JAR ファイルが正しくロードされず、ClassNotFoundException が発生する問題が修正されました。
CR082749	<p>WebLogic Server 7.0 SP1 の weblogic-rdbms20-persistence-700.dtd に新しいタグが追加されました。その新しいタグ use-select-for-update は、Bean 単位でペシミスティックな同時方式を強制します。このフラグの説明は次のとおりです。</p> <pre>&lt;!-- このフラグに「true」を指定すると、データベースから Bean がロードされるときに SELECT ... FOR UPDATE 句が使用されます。これは、トランザクション レベルではなく Bean レベルで設定する点で、トランザクション アイソレーション レベルの TRANSACTION_READ_COMMITTED_FOR_UPDATE とは異なります。 有効な値は「true」、「True」、「false」、または「False」 デフォルト値 : False 使用される場所 : weblogic-rdbms-bean 導入されたバージョン : WebLogic Server 7.0 SP1 --&gt; &lt;!ELEMENT use-select-for-update (#PCDATA)&gt;</pre>

## サンプル

変更要求番号	説明
CR073117	RMI-IIOP のサンプルが修正され、ejb cppclients の古い情報が更新されました。サーバ MBean の DefaultGIOPMinorVersion="1" が iiop MBean の DefaultMinorVersion='1' に変更されました。
CR074373	JMS サンプルでタイプミスが修正されました。サンプルの手順説明「WebLogic Server のコンフィグレーション」で、手順 10 の後に、JAR 名から -appsdir が抜けていました。
CR075073	セキュリティ プロバイダのサンプルが完成し、WebLogic Server の BEA Developer Center ( <a href="#">dev2dev</a> ) から入手できるようになりました。
CR077817	PointBase コンソールでデフォルト URL が間違っている問題が修正されました。デフォルト URL で、サーバの後にコロンが抜けていました。
CR078175	WebLogic Server Petstore の手順説明で Petstore EAR ファイルの間違ったディレクトリが参照される問題が修正されました。
CR079455	examples-dataSource-demoPool データソースが oraclePool に変更された場合にサンプルサーバの起動時に致命的な初期化例外が発生する問題が修正されました。この問題は、EJB サンプルの実行時に発生しました。
CR080073	PointBase ファイルがアップグレードされ、アプリケーション データでマルチバイト文字セットのデータを使って CLOB の更新を行うときに PointBase サーバがハングする問題が修正されました。
CR080501	WebLogic Server 製品配布キットに Web サービス サンプルが新たに追加されました。
CR082913	wtc.tBridge サンプルでエラーと例外を生じさせる問題が修正されました。javadocs が更新され、サンプルの実行前に JMS サーバを設定する手順説明が提供されるようになりました。



# インターナショナルライゼーション

変更要求番号	説明
CR063067	WebLogic Builder のメニュー、アクション、およびドキュメントがインターナショナルライズされました。その中には、AppFrame と MainAppFrame のメニューおよびコンソールのアクションが含まれません。

## JDBC

変更要求番号	説明
CR080487	DBMS のロールバックが失敗した場合に JTD ドライバがプール接続をリークしなくなりました。
CR082072	JTS 接続の自動コミット動作が機能しない問題が修正されました。
CR082964	Oracle 9.2 に対する以下の WebLogic Server サポートが提供されました。 <ul style="list-style-type: none"> <li>● Oracle 9.0.1 に対する WebLogic Server jDriver サポート</li> <li>● Oracle 9.2 に対する WebLogic Server jDriver サポート (動作確認済み)</li> <li>● WebLogic Server for Oracle 9.0.1 にバンドルされた従来の Oracle 提供シン ドライバ</li> <li>● WebLogic Server for Oracle 9.2 にバンドルされた新しい Oracle 提供シン ドライバ (Oracle で承認審査中)</li> </ul>

# JMS

変更要求番号	説明
CR063743	Object 型のメッセージ駆動型 Bean がオブジェクト メッセージの確認応答を行うようになりました。
CR075514	コンソールの [メッセージング ブリッジ] タブの [最大待機時間] 属性ラベルで、アイドル時間の単位として (秒) と表示されるべきときに (ミリ秒) と表示される問題が修正されました。
CR075894	[Monitor Durable Subscriber] テーブルで [削除] アイコンが表示されていませんでした。このため、Administration Console ではトピックから恒久サブスクライバを削除することができませんでした。このアイコンは現在は表示されます。
CR077814	クライアントの実行時に <code>weblogic.jms.backend.BEMessageReference</code> でメモリ リークが起これなくなりました。
CR078833	新しい TLOG および JMS ファイル ストアの書き込みオプションが追加されました。このリリースでは、JMS ファイル ストアでの同期書き込み「SynchronousWritesEnabled?」を無効にするコマンドライン -D パラメータが非推奨となっています。適切な方法は、新しいコンソールまたは MBean のコンフィグレーションを使用することです。
CR079855	独自の一般ブリッジ アダプタはコンフィグレーションできないことをユーザに示すドキュメントが用意されました。その説明によると、JMS 以外のメッセージング製品を使用するユーザはサードパーティ OEM ベンダを利用するか、BEA プロフェッショナル サービスに連絡して非 JMS のソース送り先またはターゲット送り先にアクセスできるようにする必要があります。
CR080078	メッセージが適切な送り先に配信できなかった後に JMS メッセージの再配信制限が変わらずにいることがなくなりました。

## JNDI

---

変更要求番号	説明
CR081839	T3Client から InitialContext() を取得するときに NamingException が発生しなくなりました。

---

## JTA

---

変更要求番号	説明
CR059175	SQL 文を実行している EJB の後に同じトランザクション内で RMI 呼び出しが続く場合に XAException が発生しなくなりました。

---

## JTS

---

変更要求番号	説明
CR064301	アプリケーションが異常終了したときに不明な分散トランザクションが作成される問題が修正されました。アプリケーションが再起動された後も、その不明な分散トランザクションは存在し続け、DBA が手動で解決しなければなりませんでした。

---

## その他

---

変更要求番号	説明
CR067750	アプリケーション スコープのプールの名前とグローバル プールの名前が互いに重複しているかどうかを確認するチェック機能が追加されました。
CR068158	SNMP MIB リファレンスが更新され、新しい WebLogic Server SNMP MBean および属性の説明が追加されました。

---

## サービス パック 1 から 6 で解決済みの問題

変更要求番号	説明
CR067987	WebLogic Server のマニュアルで File T3 についてのすべての言及が非推奨とされました。
CR072268	単純な型からの complexType の継承がサポートされるようになりました。
CR072949	障害が本当か偽りかをネイティブ レイヤから分析するために例外メッセージがチェックされます。問題は、ノード マネージャが管理対象サーバで forceShutdown コマンドを実行しようとし、それによってエラーがユーザに報告され、紛らわしいメッセージがノード マネージャのログに書き込まれる結果になることです。
CR073988	移行可能なサービスをホストしている管理対象サーバのモニタをノード マネージャが再開しない問題が修正されました。
CR074006	複数の MBean 属性が同じ oid を持つ問題が修正されました。
074653	「&」は XML ファイルで「&amp」としてエスケープしなければならない特殊文字であるために、メモリまたはディスクへの書き込み時に「&」が「&amp」で置換される、WebLogic Server Administration Console のデプロイメント記述子エディタの問題が修正されました。
CR075521	一部の検索で間違ったファイルを表示させる重複ファイルを削除することで、オンライン ヘルプの検索の問題が修正されました。

変更要求番号	説明
CR075538	<p data-bbox="440 256 1206 317">ノード マネージャが次のように定義するサーバの特定の状態を説明したドキュメントが提供されました。</p> <p data-bbox="440 326 1206 421">ノード マネージャでは、サーバの再起動時に使用するために、管理対象サーバについて独自の内部的な状態が定義されています。ドメインでノード マネージャを使うときは、これらの状態を観察できます。</p> <ul data-bbox="440 439 1206 881" style="list-style-type: none"><li data-bbox="440 439 1206 508">• FAILED_RESTARTING— ノード マネージャは、現在、障害が発生した管理対象サーバを再起動しています。</li><li data-bbox="440 526 1206 621">• ACTIVATE_LATER— 現在の RestartInterval において MaxRestart 回の再起動が試みられており、ノード マネージャは次の RestartInterval でさらに再起動を試みます。</li><li data-bbox="440 638 1206 769">• FAILED_NOT_RESTARTABLE— サーバで障害が発生しましたが、サーバの AutoRestart 属性または AutoKillIfFailed 属性が False に設定されているため、ノード マネージャはサーバを再起動できません。</li><li data-bbox="440 786 1206 881">• FAILED_MIGRATABLE— サーバで障害が発生しましたが、サーバの HostsMigratableServices 属性が True に設定されているため、ノード マネージャはサーバを再起動できません。</li></ul>
CR078272	<p data-bbox="440 907 1206 1072">iPlanet プラグインが 64 ビットをサポートする Tru54unix プラットフォームで機能しない問題が修正されました。64 ビットのサポートを使用するには、iPlanet 6x プラグインの以下の 2 つのバイナリをインストールし、iPlanet 4x プラグイン用の既存のバイナリを置換する必要があります。</p> <ul data-bbox="440 1090 628 1177" style="list-style-type: none"><li data-bbox="440 1090 628 1124">• libproxy.so</li><li data-bbox="440 1142 628 1177">• libproxy128.so</li></ul> <p data-bbox="440 1185 1206 1246">これらのバイナリを入手するには、<a href="#">BEA のサポート</a>に問い合わせてください。</p>
CR078782	<p data-bbox="440 1272 1206 1428">コネクタ記述子コードが weblogic-ra.xml ファイルのエンコーディングに従わない問題が修正されました。データは、JMV のデフォルトエンコーディングで読み込まれていました。現在、XML ヘッダのエンコーディング仕様はデプロイメント記述子ファイルと RAR ファイルで適切に処理されます。</p>

変更要求番号	説明
CR079186	すべてのプラグインで、折り返されたヘッダが HTTP 1.1 仕様に従って正しく解析されるようになりました。
CR079462	アプリケーション スコープのプールの詳細が <code>weblogic-application.xml</code> ファイルで確実に処理されるようになりました。この修正により、ローカル プロパティが指定されていない場合はグローバル プロパティが使用されることとなります。また、ドライバ、url、および接続プロパティが <code>weblogic-application.xml</code> の <code>jdbc-connection-pool</code> スタンザにない場合に、 <code>JDBCDataSourceFactopry</code> の定義は必ずしも使用されません。
CR079672	<code>WarClassFinder.getSource</code> で JVM のクラッシュが引き起こされる問題の解決策が提供されました。
CR079683	<code>jsp.include</code> の使用時に CTE を使用した Apache プラグインで発生する問題が修正されました。チャンク転送エンコーディングされた HEX 値は、 <code>mod_wl_ssl_so</code> の使用時にブラウザに表示されます。
CR079831	7.0 メッセージ カタログの DTD にロギング可能な属性が含まれなかった問題が修正されました (その DTD がこの機能に対応するように更新されなかったため)。これはメッセージ カタログのパーサにとっては問題ではありませんでしたが、汎用 XML エディタでは問題となっていました。
CR079973	<code>Idempotent</code> が OFF および <code>ConnectTimeoutSecs</code> がゼロに設定されているサーバへの接続が複数回試行される問題が、 <code>Idempotent</code> が OFF の場合はフェイルオーバを防止し、 <code>ConnectTimeoutSecs</code> がゼロの場合はコアダンプを防止することで解決されました。
CR080016	アプリケーションが仮想ホストをターゲットとしており、その仮想ホストが管理対象サーバおよび管理サーバをターゲットとしている場合に、WebLogic 管理サーバの起動中にそのアプリケーションをデプロイすると発生した問題が修正されました。
CR080073	マルチバイト文字セットのデータがアプリケーション データで使用されている場合に、CLOB の更新時に PointBase サーバがハングする問題が修正されました。
CR080219	WebLogic Server のすべてのインスタンスを停止するときに iPlanet でコアダンプが生じる問題が修正されました。

変更要求番号	説明
CR080249	WebLogic Server 7.0 の管理サーバが NT サービスとして設定されている場合にノード マネージャと通信できない問題が修正されました。
CR080257	HTTP リクエストの最初の行が正しくフォーマットされていない場合に <code>NegativeArraySizeException</code> が発生する問題が修正されました。
CR080382	次の設定で NSAPI プラグインを使用しているときに、予期しないエラー メッセージが iPlanet Web Server のエラー ログに記録される問題が修正されました。 <code>ErrorPage=[any]</code>
CR080746	ネイティブ Web サーバ プラグインで 64 ビット プラットフォームのサポートが提供されました。
CR080931	<code>CallableStatement</code> を使用した後に <code>PreparedStatement</code> が <code>min_bind_size</code> と機能しない Oracle <code>jdbcDriver</code> の問題が、 <code>db\oci\OciConnection.java</code> を変更することで修正されました。
CR081377	デプロイメント時にコンテキスト クラス ローダを通じてクラスをロードしたときに発生する問題が修正されました。
CR081625	WebLogic Server に PointBase 4.3 が追加されました。
CR082081	<code>t3</code> の使用時に <code>weblogic.Admin SERVERLOG</code> コマンドライン ユーティリティが失敗する問題が修正されました。
CR082093	プロキシに呼び出しが到達するたびにアプリケーション サーバに対して複数の呼び出しが行われる、WebLogic Server および NSAPI プラグインの問題が修正されました。
CR082096	<code>Idempotent</code> が OFF に設定されている場合に発生し、IIS プロキシ プラグインでサーバの後にすべての POST リクエストに対して 1 度の再試行を試みさせる問題が修正されました。
CR082113	WebLogic Server-ISAPI プラグインがクライアントからのデータの読み込み時にエラーを受信し、バックエンドの WebLogic Server に不完全なデータ (リクエスト) を送信したときに発生する問題が修正されました。
CR082939	ISAPI プラグインの問題が修正され、セッション ID の異なる形式に対応するようになりました。

## サービス パック 1 から 6 で解決済みの問題

変更要求番号	説明
CR083151	EAR ファイルでコマンドライン ddinit が機能しない、weblogic.Builder の問題が修正されました。
CR083174	セッションに新しい値を追加するときに最新の IIS プラグインが機能しない問題が修正されました。たとえば、IIS では、メソッド KeepAliveEnabled=true が正しくフェイルオーバーされません。
CR083643	POST データを読み込もうとするときの NSAPI プラグインのパフォーマンスが改善されました。問題は、POST データで \0 と 0 を区別するときに発生しました。



## RMI/RMI-IIOP

変更要求番号	説明
CR073691	RMI-IIOP のトレース機能が実装されました。
CR074790	アプレット内で RMI-IIOP を使用するとき AccessControlExceptions が生じる問題が修正されました。
CR076974	リクエストがないときに <code>weblogic.rmi.ReplyOnError.execute</code> が失敗する問題が修正されました。
CR079387	RMI-IIOP でサブコンテキストが壊れる問題が修正されました。JNDI でサブコンテキストをルックアップすると、代わりにルート コンテキストが表示されます。
CR079439	Any のメソッドで渡された文字列が不正確にコード化される問題が修正されました。この問題は、Object、Serializable、または Externalizable の宣言型を持つリモート関数に文字列を渡すときに発生します。
CR079454	interop listBindings がまったく機能しない問題が修正されました。
CR079645	OBJECT_NOT_EXIST などの CORBA プロトコル例外が間違っ RMI-IIOP <code>java.rmi.MarshalException</code> にマッピングされています。
CR079648	IOR が TransactionFactory に対して定義されず、それが原因でユーザがクライアントから OTS TransactionFactory を取得できない問題が修正されました。
CR079918	動的プロキシがシリアライズ可能でないために発生する問題が修正されました。この問題は、RMI オブジェクトが各クライアントで EJB からクライアントへのコールバック リクエストを容易にするために使用されたときに発生しました。
CR080239	IDL パッケージの名前が正しく管理されない問題が修正されました。
CR080259	boxed RMI シーケンスのコンポーネント型で valuetype の間違っ シーケンスが生成される問題が修正されました。
CR080306	特定の環境において、valuetype の例外で継承した基本クラスが脱落する問題が修正されました。

## サービス パック 1 から 6 で解決済みの問題

変更要求番号	説明
CR081135	リクエストのサービス コンテキストが大きい場合にフラグメンテーションとなる問題が修正されました。
CR081923	<code>IncompatibleClassChangeError</code> を発生させる RMI/IIOP スタブコード生成の問題が修正されました。生成されたスタブでは、 <code>RemarshalException</code> が正しく処理されませんでした。
CR083018	WebLogic Server で、間接的な型コードを使用して <code>Any</code> のメソッド型をアンマーシャリングできない問題が修正されました。

# セキュリティ

変更要求番号	説明
CR069969	WebLogic Server で信頼性のある CA をロードできるようにする新しいアルゴリズムが作成されました。以前は、ユーザが信頼性のある CA をコンフィグレーションしなかった場合、WebLogic Server では JDK cacerts ファイルからデフォルトの信頼性のある CA をロードしていました。
CR070307	相互 SSL 用にコンフィグレーションされているサーバをユーザが起動したときに発生するサーバ障害の問題が修正されました。
CR071894	デフォルト キーストアで、プライベート キーまたはルート CA キーストアがコンフィグレーションされているが、ファイルが存在しないか、指定の場所がない場合に情報メッセージがログに記録されるようになりました。
CR072902	信頼性のある証明書をロードするためのアルゴリズムが修正されました。
CR073082	Solaris 上で証明書を生成するときに、CertGen で、証明書の共通名フィールドにドメイン名が追加されることがなくなりました。Solaris 上で HostNameVerification が有効になっても、それはホスト名の不一致が原因で失敗します。
CR073665	iPlanet プロキシ サーバを通じて HTTPS をトンネリングするときに、ProtocolException が発生しなくなりました。
CR073941	デフォルト キーストアでエラー メッセージをログに記録できるようになりました。
CR074102	Administration Console でデフォルトのクッキー名を使用できなくなる問題が修正されました。別の Web アプリケーションを通じてコンソールをロードしようとした場合に、コンソールと同じクッキー (ADMINCONSOLESESSION) を持つように Web アプリケーションをコンフィグレーションしない限りは、シングル サインオンでアクセスすることができませんでした。結果として、コンソールと Web アプリケーションのシングル サインオン (SSO) メカニズムの間でクッキー名の衝突が生じていました。

変更要求番号	説明
CR074612	jdk ディレクトリで指定された cacert ファイルが存在せず、信頼性のある cacert ファイルの位置がコマンドラインで指定されたことが原因で WebLogic Server の起動時に SSL が起動しない問題が修正されました。
CR075172	SSL ハンドシェイクが、コンフィグレーションされているルート キーストアで CA 証明書を見つけることができないときに生じる問題が修正されました。
CR075901	埋め込み LDAP に 50000 ユーザがロードされている状態でユーザを表示するときの、コンソールのリストの処理方法に関する問題が修正されました。
CR076409	NTSocket Muxer のクラッシュを生じさせる問題が修正されました。この問題は、サーバへの接続を作成する Perl スクリプトの使用時に表面化しました。そのスクリプトが原因で、Web サーバがクラッシュしました。
CR076945	LDAP ファイルの破損の可能性が原因で WebLogic Server の起動時に発生する問題が修正されました。この問題では、よく <code>java.lang.OutOfMemory</code> エラーが生じます。
CR078249	LDAP V2 レルムのネストされたグループが機能しなくなる問題が修正されました。
CR078797	EJB コンテナがすべてのメソッド（除外されていないまたはチェックされているメソッドを除く）でメソッド パーミッションを確認したときに、セキュリティ制限の設定されていないメソッドでユーザが実行パーミッションを拒否される問題が修正されました。
CR078887	埋め込み LDAP サーバへの匿名ログインを無効にして、匿名ログインを禁止にする手段が提供されました。
CR080072	プライベート キーストア、エリアス、およびパスワードが <code>config.xml</code> ファイルで指定された 7.0 SSL コンフィグレーションを持つ 7.0 バージョンの WebLogic Server がデフォルトで 6.x SSL コンフィグレーションになる問題が修正されました。
CR080490	<code>DefaultKeyStore</code> プロバイダが非 JKS JDK キーストアをサポートしない問題が修正されました。

変更要求番号	説明
CR080508	厳重なサーブレット セキュリティをサポートできるように WebResource が更新されました。
CR080567	LDAP からのエラー メッセージが管理サーバのログに表示される問題が修正されました。VDE サーバは、現在、変更ログの短縮の後に正しくレプリケートするようになりました。

変更要求番号	説明
CR081494	<p>ロール マッピングおよび AccessDecision セキュリティ サービス プロバイダ インタフェース (SSPI) は、コンテキスト ハンドラを介して HttpServletRequest および HttpServletResponse オブジェクトにアクセスできるようになりました。次に例を示します。</p> <pre data-bbox="377 395 1022 505">HttpServletRequest req = (HttpServletRequest) context.getValue("HttpServletRequest"); HttpServletResponse res = (HttpServletResponse) context.getValue("HttpServletResponse");</pre> <p>応答から返される情報はクッキーに設定できます。これらのオブジェクトは URL リソースでのみ機能します。このコードは、他のすべてのタイプの WebLogic リソースでは null を返します。</p>
CR083125	<p>weblogic.security.service.URLResource クラスが Windows 上で他のプラットフォームとは異なる動作をする問題が修正されました。Windows 上で、このクラスは contextPath と URI を小文字に変換していました。この問題は、contextPath と URI が Windows 上では大文字と小文字が区別されないことが原因で発生しました。</p> <p>デフォルトの動作を変更するには、以下のフラグのいずれかを設定する必要がありますがありました。</p> <ul data-bbox="391 881 1049 991" style="list-style-type: none"><li>● -Dweblogic.security.URLResourceCaseMapping=on</li><li>● -Dweblogic.security.URLResourceCaseMapping=off</li><li>● -Dweblogic.security.URLResourceCaseMapping=os</li></ul> <p>フラグを on に設定すると、contextPath と URI が常に小文字にマッピングされます。フラグを off に設定すると、contextPath と URI は小文字にマッピングされません。フラグを os に設定すると、contextPath と URI は Windows 上では小文字にマッピングされます。</p> <p>このフラグは、Windows マシンとそれ以外のマシンが混在する環境でドメインを実行する場合にのみ設定する必要がありますがありました。その場合、フラグを on に設定することで、すべてのプラットフォームで問題が解決されましたが、Windows 以外のシステムで大文字と小文字が区別されなくなりました。</p>

## サーブレットと JSP

変更要求番号	説明
CR042655	access.log ファイルの拡張ログフォーマットの時刻書式でエラーが修正されました。時刻は GMT で表示されるようになりました。
CR074096	匿名ユーザとして実行されたときにサーブレットの <init> メソッドで MBean アクセスが存在しない問題が修正されました。run-as identify の代わりに使用される、2 つの新しい要素が weblogic.xml デプロイメント記述子ファイルに追加されました。
CR074843	<p>式が Null の場合に空の文字列を出力するために使用する JSP のリクエスト時属性で文字列 Null が出力されるように WebLogic Server がアップグレードされました。</p> <p>WebLogic Server の旧バージョンでは、JSP のリクエスト時属性 (&lt;%= expr %&gt;) は、式が Null の時には空の文字列を出力します。現在は、文字列「null」を出力するようになっています。JSP の仕様では、式のデフォルトは「null」でなければならないと規定されています。</p> <p>ただし、weblogic.xml に printNulls という名前の新しいフラグが導入されています。フラグのデフォルト値 true は、デフォルトの出力が「null」であることを意味しています。このフラグを false に設定すると、「null」の式が "" として出力されます (以前の動作)。</p> <p>weblogic.xml の printNulls タグのコンフィグレーション:</p> <pre>&lt;weblogic-web-app&gt; &lt;jsp-descriptor&gt; &lt;jsp-param&gt; &lt;param-name&gt;printNulls&lt;/param-name&gt; &lt;param-value&gt;&gt;false&lt;/param-value&gt; &lt;/jsp-param&gt; &lt;/jsp-descriptor&gt; &lt;/weblogic-web-app&gt;</pre> <p>また、次のように指定して weblogic.jspc からコンパイルできます。</p> <pre>-noPrintNulls</pre> <p>これにより、JSP の式の「null」も "" (空の文字列) として示されます。</p>
CR075805	weblogic.servlet.security.ServletAuthentication に新しい runAs メソッドを追加することで、HttpSession の現在の主体を設定する手段が実装されました。

変更要求番号	説明
CR076330	サーバが名前にピリオドのある JAR ファイルから Web アプリケーションクラスをロードできない問題が修正されました。
CR077823	プラグインのパラメータ情報の更新を反映して、『管理者ガイド』と『WebLogic Server における Web サーバ プラグインの使い方』のプラグイン情報が更新されました。
CR077944	JSP2XmlOutputter が dequote() を不適切に呼び出して解析エラーが生じる JSP タグの問題が修正されました。
CR078698	サイズが 2MB を超えたときに access.log ファイルがサイズを基準にローテーションされることを妨げる問題が修正されました。
CR078725	チャンク転送エンコーディングされた応答を HttpClusterServlet がデコードする問題が修正されました。
CR079892	複数のブラウザ ウィンドウから CGI スクリプトを実行し、そのプロセスの途中ですべてのブラウザで停止ボタンを押したときに CPU 使用率が 100% になる問題が、文字セットが Context-Type で定義されている巨大な CGI スクリプトを実行するときにハング スレッドを防ぐことで修正されました。
CR079924	weblogic.xml ファイルに新しい要素 wl-dispatch-policy を追加することで、web-app レベルでディスパッチ ポリシーを定義する機能が提供されました。
CR080090	FrontEndHTTPPort、FrontEndHTTPSPort、および FrontEndPort が普遍的に優先されるようになりました (定義されている場合)。これらは、リダイレクト時に常に使用する必要があります。
CR080445	WebLogic Server が config.xml ファイルの server セクションの WeblogicPluginEnabled=true を認識できない問題が修正されました。これが原因で、不正確なクライアント マシンのアドレスがヘッダで渡されていました。
CR080554	IIS および iPlanet が WebLogic Server に URL 書き換えのための jsessionid 識別子を渡さない問題が、EncodeSessionIdInQueryParams プロパティを導入することで修正されました。
CR080608	weblogic.servlet.security.ServletAuthentication weak 文字列で常に FAILED_AUTHENTICATION が返される問題が修正されました。



変更要求番号	説明
CR080613	引用符で囲まれた文字列値で特殊文字を使用できるように WebLogic Server が更新されました。以前は、クッキーの引用符で囲まれた値でカソマを使用できませんでした。
CR080648	指定されたロールをグローバル ロールとしてマークする <code>global-role</code> という新しい属性が追加されました。
CR080751	エラー ページのカスタム例外へのマッピングに関する問題が修正されました。この問題は、 <code>ServletException</code> を拡張したカスタム例外クラスがサーブレット 2.3 仕様に完全に準拠していなかったことが原因で発生しました。
CR080772	ユーザが文字セットに余計な引用符を付けた場合に <code>UnsupportedEncodingException</code> を発生させた問題が修正されました。
CR080791	<code>HttpClusterServlet</code> および <code>HttpProxyServlet</code> で折り返されたヘッダを正しく処理する機能が提供されました。
CR081253	WebLogic Server がチャンク コーディングされた HTTP リクエストを正しく解析できない問題が修正されました。
CR081484	<code>PersistentStoreType</code> が <code>replicated</code> として指定された <code>HttpSessionListener</code> を使用したときにセッション属性が失われる問題が修正されました。
CR081521	JSP から呼び出された CGI スクリプトが WAR ファイルまたは EAR ファイルで機能しない問題が修正されました。
CR081792	<code>HttpServletRequestWrapper</code> の型チェックが緩和されて、サーブレット エンジンで <code>HttpServletRequestWrapper</code> の代わりに <code>ServletRequestWrapper</code> を使用できるようになりました。
CR082238	<code>CacheSessionCookie</code> というセッション記述子が <code>weblogic.xml</code> ファイルに追加されました。この記述子は、デフォルトで <code>false</code> に設定されます。true に設定すると、 <code>Cache-Control</code> ヘッダが追加されません。
CR082310	ユーザが展開された WAR ファイルで「page contentType」ディレクティブのある JSP を削除または上書きできない問題が修正されました。解決策は、JSP パーサの入力ストリームが使用後に確実に閉じられるようにすることでした。

## サービス パック 1 から 6 で解決済みの問題

変更要求番号	説明
CR082461	FileServlet が機能しない問題が修正されました。docHome パラメータが指定されている場合に目的のファイル名を作成するために使用される PATH_INFO の先頭に SERVLET_PATH が付加されないようになりました。
CR082636	空の auth 制約で fullSecurityDelegation モードのアクセスが制限されない問題が修正されました。
CR083200	WebLogic Server が nonProxyHosts を正しくフィルタ処理できない問題が修正されました。WebLogic Server は、http.nonProxyHosts と nonProxyHosts の両方に対応するようになりました。
CR083517	SecureProxy が有効な場合でも HttpClusterServlet で SSL にリクエストをプロキシできない問題が修正されました。
CR083654	マルチバイト文字セットをコンテンツタイプとして使用するサーブレットおよび JSP のパフォーマンスを向上させることで、パフォーマンスの問題が改善されました。

## ツール

変更要求番号	説明
CR074535	DDinit が再帰的な関連を認識できない、weblogic.Builder の問題が修正されました。
CR074535	DDinit が再帰的な関連を認識できない、weblogic.Builder の問題が修正されました。
CR074570	DDinit が cmr フィールドを cmp フィールドと間違える、weblogic.Builder の問題が修正されました。
CR074620	関係が 1 対 1 の場合でもウィザードの [Bi-directional] フィールドがチェックされている場合は CMP フィールドタイプが有効になる、weblogic.Builder の問題が修正されました。
CR074629	ユーザが [Connect to Server] ダイアログ ボックスで何らかの参照アクションに対して取り消しボタンを押したときに [Illegal DataSource] ダイアログ ボックスが表示される、weblogic.Builder の問題が修正されました。
CR074689	[create duplicate environments] フィールドをチェックすることで不完全な XML が生成される、weblogic.Builder の問題が修正されました。
CR074764	ステートフル セッション Bean のチューニング パネルに欠けている [set cache-type] フィールドが [Session Advanced] パネルに追加されました。
CR074773	XMLElement ジェネレータで、allow-concurrent-calls の指定されたステートフルセッション EJB の XML コードを生成できない問題が修正されました。
CR074995	ユーザが関係を削除したときに、XML は削除されるが、モジュールは modified とマークされない、weblogic.Builder の問題が修正されました。モジュールが閉じるときに、ユーザには変更の保存が求められませんでした。
CR075034	[EJB Tuning Cluster] パネルで新しい多重呼び出し不変メソッドを追加すると java.lang. IllegalArgumentException が発生する、weblogic.Builder の問題が修正されました。

変更要求番号	説明
CR075521	さまざまなトピックの検索で間違っただけが表示される、weblogic.Builder オンライン ヘルプの問題が修正されました。
CR075537	実行中のドメインのクラスタに追加されたサーバがサーバの新しいクラスタにデプロイされない、weblogic.Deployer の問題が修正されました。 クラスタにサーバを追加して、管理対象サーバを再起動した場合、そのクラスタをターゲットとし、そのクラスタにデプロイされていた Web アプリケーションは追加サーバで自動的にデプロイされません。
CR075672	methods-are idempotent が非推奨になったため、weblogic.Builder で [Methods Are Idempotent] チェックボックスが削除されました。ステートレス セッション EJB で多重呼び出し不変のメソッドを追加する手段が開発されました。多重呼び出し不変のメソッドは、[クラスタ] パネルで指定できるようになりました。
CR075937	weblogic.Builder が EJB デプロイメント記述子 max-cache-size で無効な値を書き込む問題が修正されました。
CR077171	DDinit がコンテナ管理の永続性 (cmp) 記述子で間違っただけバージョンを設定する、weblogic.Builder の問題が修正されました。
CR078747	weblogic.Builder が ejb11_container_managed.jar ファイルを開くことができない問題が修正されました。
CR081213	weblogic.Builder のステート Bean の [一般] タブでステートレス セッション Bean とステートフル セッション Bean のチェックボックスを切り替えるときに NullPointerException が発生する問題が修正されました。
CR081325	DDInit が Web アプリケーションの EJB Ref と Resource Ref および Bean の Resource Ref で JNDI 名を割り当てない、weblogic.Builder の問題が修正されました。

# WebLogic Tuxedo Connector

変更要求番号	説明
CR078774	Tuxedo /Q からメッセージを取り出したときに応答がなくなる、tBridge セッションの接続の問題が修正されました。

## Web サービス

変更要求番号	説明
CR066226	WebLogic Server で、相互 SSL 認証に RPC タイプの Web サービスを提供できるようになりました。
CR069027	wsdl クライアントと協調して複合データ型の使われる Web サービスを呼び出すのを妨げる問題が修正されました。
CR072268	WebLogic Web サービスでは、単純な型から継承された complexTypes のある XML スキーマはサポートされていません。
CR073539	余計な「/」が誤って付けられていた、Web サービスのホーム ページの壊れたリンクが修正されました。
CR075634	生成されたスタブが echo"Struct で doc/lit を処理できない問題が修正されました。
CR076677	wsdl クライアントと協調して複合データ型の使われる Web サービスを呼び出すのを妨げる問題が修正されました。
CR077849	英語の環境でマルチバイト文字列が Web サービスによって壊される問題が修正されました。
CR077855	URL.getContext() の使用時に Java クライアントから Web サービスにアクセスしようとするとき NullPointerException が発生する問題が修正されました。
CR078313	スタブクライアントの .net サービスとの相互運用が失敗したときに断続的に発生する接続の問題が修正されました。

## サービス パック 1 から 6 で解決済みの問題

変更要求番号	説明
CR078442	サービスの生成で protocol=https を使用した場合と使用しない場合の両方で生成された Web サービスのホームページに HTTPS を通じてアクセスすることで認証ログイン ダイアログが表示される問題が修正されました。
CR078804	Web Services Stateless Session サンプルで非パブリック API <code>import weblogic.ejb.GenericSessionBean</code> が使用される問題が修正されました。
CR079620	SOAP メッセージが RFC2376 に準拠しました。
CR079976	clientgen などのタスクの使用時に HTTP プロキシ サーバを指定しようとして発生する問題が、proxyHost 属性および proxyPort 属性を clientgen に追加することで修正されました。
CR079984	Web サービスのサンプルに <code>webservices.jar</code> ファイルがあるかどうかを確認する機能が無効になりました。
CR080152	WebLogic Server の <code>dateTime</code> の解析に関する問題が修正されました。
CR080154	HTTP プロトコル、およびメソッド シグネチャで <code>int</code> または <code>string</code> の配列を使用したときに認証画面が表示される問題が修正されました。
CR080264	SOAP 応答が HTTP ヘッダに必須の <code>utf-8 contentType</code> を持たない問題が修正されました。
CR080961	XML パーサ Xerces 1.3.1 の <code>org.xml.sax.helpers.AttributesImpl.removeAttribute()</code> 呼び出しのバグに対する対処法が開発されました。
CR081570	プロパティの <code>set</code> または <code>get</code> メソッドがないときに <code>servicegen</code> が失敗する問題が、ゲッター メソッドとセッター メソッドの両方を持たない Java Bean プロパティを無視することで解決されました。