



Sun Java System Application Server Enterprise Edition 8.2 リ リースノート (UNIX 版)



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

Part No: 820-0500
2008 年 5 月

本書で説明する製品で使用されている技術に関連した知的所有権は、Sun Microsystems, Inc. に帰属します。特に、制限を受けることなく、この知的所有権には、米国特許、および米国をはじめとする他の国々で申請中の特許が含まれています。

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

本製品には、サードパーティーが開発した技術が含まれている場合があります。

本製品の一部は Berkeley BSD システムより派生したもので、カリフォルニア大学よりライセンスを受けています。UNIX は、X/Open Company, Ltd. が独占的にライセンスしている米国ならびにほかの国における登録商標です。

Sun、Sun Microsystems、Sun のロゴマーク、Solaris のロゴマーク、Java Coffee Cup のロゴマーク、docs.sun.com、Java、Solaris は、米国およびその他の国における米国 Sun Microsystems, Inc. (以下、米国 Sun Microsystems 社とします) の商標もしくは登録商標です。Sun のロゴマークおよび Solaris は、米国 Sun Microsystems 社の登録商標です。すべての SPARC 商標は、米国 SPARC International, Inc. のライセンスを受けて使用している同社の米国およびその他の国における商標または登録商標です。SPARC 商標が付いた製品は、米国 Sun Microsystems 社が開発したアーキテクチャーに基づくものです。

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

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

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

概要

Sun Java™ System Application Server Enterprise Edition 8.2 製品は、J2EE アプリケーションと Web サービスの構築と管理の作業を大幅に簡略化します。ソフトウェア障害やハードウェア障害が発生した場合にも稼働し続けるスケーラブルなサービスを実現するために、優れたパフォーマンス、クラスタリング、および高可用性機能を提供します。

- 3 ページの「このリリースノートについて」
- 4 ページの「リリースノート改訂履歴」
- 4 ページの「Application Server のマニュアルセット」
- 5 ページの「関連マニュアル」
- 6 ページの「アクセシビリティ機能」
- 6 ページの「マニュアル、サポート、およびトレーニング」
- 6 ページの「問題の報告とフィードバックの方法」
- 7 ページの「このマニュアルに関するコメント」

このリリースノートについて

このリリースノートには、Sun Java System Application Server 8.2 のリリース時点で入手できる重要な情報が含まれています。拡張機能、既知の問題、およびその他の最新の情報が記載されています。Application Server Enterprise Edition 8.2 をご使用になる前に、このリリースノートをお読みください。

このリリースノートの最新バージョンについては、Sun Java System の [マニュアル Web サイト \(http://docs.sun.com/app/docs/coll/1310.3\)](http://docs.sun.com/app/docs/coll/1310.3) を参照してください。ソフトウェアのインストールおよびセットアップを行う前に、この Web サイトを確認してください。また、それ以降も定期的にこのサイトをチェックして、最新のリリースノートや製品マニュアルを参照してください。

このマニュアル内で参照している第三者の URL は、追加の関連情報を提供します。

注- このマニュアル内で引用する第三者の Web サイトの可用性について Sun は責任を負いません。こうしたサイトやリソース上の、またはこれらを通じて利用可能な、コンテンツ、広告、製品、その他の素材について、Sun は推奨しているわけではなく、Sun はいかなる責任も負いません。こうしたサイトやリソース上の、またはこれらを経由して利用可能な、コンテンツ、製品、サービスを利用または信頼したことによって発生した(あるいは発生したと主張される)いかなる損害や損失についても、Sun は一切の責任を負いません。

リリースノート改訂履歴

ここでは、Sun Java System Enterprise Edition Application Server 8.2 製品の初版リリース以降に各リリースノートで加えられた変更点を示します。

表 1-1 リリースノート改訂履歴

改訂日付	説明
2006 年 10 月	Sun Java System Application Server Enterprise Edition 8.2 製品のベータリリースの初版リリースノート。
2007 年 2 月	Sun Java System Application Server Enterprise Edition 8.2 製品の FCS リリースのリリースノート。
2007 年 7 月	インストールに関する既知の問題に欠陥 6396045 を追加。
2007 年 8 月	Java ES 5 でサポートされる Solaris および Linux プラットフォームを反映するように、Web Server のプラットフォーム要件を変更。
2008 年 5 月	サポートされる Web Server を 6.0、6.1、7.0 に変更。 「システムの仮想化のサポート」の節を追加。

Application Server のマニュアルセット

Application Server のマニュアルセットは、配備の計画とシステムのインストールについて説明しています。スタンドアロンの Application Server のマニュアルの URL (Uniform Resource Locator) は <http://docs.sun.com/app/docs/coll/1310.4> です。Sun Java Enterprise System (Java ES) Application Server のマニュアルの URL は <http://docs.sun.com/app/docs/coll/1310.3> です。Application Server の概要については、次の表の順番で各マニュアルを参照してください。

表 1-2 Application Server のマニュアルセットの内容

マニュアル名	説明
『リリースノート』	ソフトウェアとマニュアルに関する最新情報。リリースノートには、サポートされているハードウェア、オペレーティングシステム、Java Development Kit (JDK™)、およびデータベースドライバの包括的な概要が表形式で記載されています。
『クイックスタートガイド』	Application Server 製品の使用を開始するための手順。
『インストールガイド』	ソフトウェアとそのコンポーネントのインストール。
『配備計画ガイド』	最適な方法で確実に Application Server を導入するための、システムニーズや企業ニーズの分析。サーバーを配備する際に注意すべき一般的な問題や懸案事項についても説明しています。
『開発者ガイド』	J2EE コンポーネントおよび API 用のオープン Java 標準モデルに従い、Application Server 上で実行することを目的とする Java 2 Platform, Enterprise Edition (J2EE™ プラットフォーム) アプリケーションの作成と実装。開発ツール、セキュリティ、デバッグ、配備、ライフサイクルモジュールの作成などについての情報も提供します。
『J2EE 1.4 Tutorial』	J2EE アプリケーションの開発のための J2EE 1.4 プラットフォーム技術および API の使用。
『管理ガイド』	管理コンソールからの、Application Server サブシステムおよびコンポーネントの設定、管理、および配備。
『高可用性 (HA) 管理ガイド』	高可用性データベースのための、インストール後の設定と管理に関する解説。
『Administration Reference』	Application Server 設定ファイル domain.xml の編集。
『アップグレードと移行』	新しい Application Server プログラミングモデルへのアプリケーションの移行 (特に Application Server 6.x または 7 からの移行)。このマニュアルでは、製品仕様の非互換性をもたらす可能性のある、隣接した製品リリース間の相違点や設定オプションについても説明しています。
『パフォーマンスチューニングガイド』	パフォーマンスを向上させるための Application Server の調整。
『トラブルシューティングガイド』	Application Server の問題の解決。
『Error Message Reference』	Application Server のエラーメッセージの解決。
『リファレンスマニュアル』	Application Server で使用できるユーティリティコマンド。マニュアルページのスタイルで記述されています。asadmin コマンド行インタフェースも含まれます。

関連マニュアル

Application Server は、単体で購入することが可能です。あるいは、ネットワークまたはインターネット環境にわたって分散しているエンタープライズアプリケーションをサポートするソフトウェアインフラストラクチャーである Java ES のコンポーネン

トとして購入することもできます。Application Server を Java ES のコンポーネントとして購入した場合は、<http://docs.sun.com/coll/1286.2> にあるシステムマニュアルをよく読むことをお勧めします。Java ES およびそのコンポーネントに関するすべてのマニュアルの URL は <http://docs.sun.com/prod/entsys.5> です。

その他の Sun Java System サーバーについては、次のマニュアルを参照してください。

- Message Queue のマニュアル
- Directory Server のマニュアル
- Web Server のマニュアル

さらに、次のリソースが役立つことがあります。

- J2EE 1.4 Specifications (<http://java.sun.com/j2ee/1.4/docs/index.html>)
- J2EE 1.4 Tutorial (<http://java.sun.com/j2ee/1.4/docs/tutorial/doc/index.html>)
- J2EE Blueprints (<http://java.sun.com/reference/blueprints/index.html>)

アクセシビリティ機能

このメディアの出版以降にリリースされたアクセシビリティ機能を入手するには、米国リハビリテーション法 508 条に関する製品評価資料を Sun に請求し、その内容を確認して、どのバージョンが、アクセシビリティに対応したソリューションを配備するためにもっとも適しているかを特定してください。アプリケーションの更新されたバージョンは <http://sun.com/software/javaenterprisesystem/get.html> から入手できます。

アクセシビリティ機能に対する Sun のコミットメントについては、<http://sun.com/access> を参照してください。

マニュアル、サポート、およびトレーニング

Sun の Web サイトには、次に示す関連情報が示されています。

- マニュアル (<http://www.sun.com/documentation/>)
- サポート (<http://www.sun.com/support/>)
- トレーニング (<http://www.sun.com/training/>)

問題の報告とフィードバックの方法

Sun Java System Application Server に問題が発生した場合は、次のいずれかの方法で Sun のカスタマサポートにお問い合わせください。

- フィードバック送信フォーム
(<http://java.sun.com/docs/forms/J2EE14SubmittalForm.html>) — Application Server 製品についてのフィードバックを送信するためのフォームです。
- J2EE-INTEREST リスト
(<http://archives.java.sun.com/archives/j2ee-interest.html>) — J2EE に関する質疑を扱うメーリングリストです。

- [Java Developer Connection のバグデータベース \(http://developer.java.sun.com/servelet/SessionServlet?url=/developer/bugParade/index.jshtml\)](http://developer.java.sun.com/servelet/SessionServlet?url=/developer/bugParade/index.jshtml) — バグを閲覧または報告するには、この URL にある Java Developer Connection Bug Parade をご利用ください。
- [Java Technology Forums \(http://forum.java.sun.com/\)](http://forum.java.sun.com/) — Java のテクノロジーとプログラミング技法に関する知識と疑問を共有するための対話型掲示板です。Sun Java System Application Server Enterprise Edition 8.2 製品について議論するには、ここの J2EE SDK フォーラムをご利用ください。
- [Sun Software Support Services \(http://www.sun.com/service/sunone/software\)](http://www.sun.com/service/sunone/software) — Knowledge Base、オンラインサポートセンター、および Product Tracker、さらには保守プログラムやサポートに関する問い合わせ番号へのリンクが記載されています。
- 保守契約を結んでいるお客様の場合は、専用ダイヤルをご利用ください。
最善の問題解決のため、サポートに連絡する際には次の情報をご用意ください。
- 問題が発生した箇所や動作への影響など、問題の具体的な説明
- マシン機種、OS バージョン、および製品のバージョン (問題に関係するパッチおよびその他のソフトウェアを含む)
- 問題を再現するための具体的な手順の説明
- エラーログやコアダンプ

このマニュアルに関するコメント

弊社では、マニュアルの改善に努めており、お客様からのコメントおよびご忠告をお受けしております。コメントを送るには、<http://docs.sun.com> にアクセスして「コメントの送信」をクリックしてください。オンラインフォームに、マニュアルの完全なタイトルと Part No. を記入してください。Part No. は 7 桁または 9 桁の数字で、マニュアルの表紙またはドキュメントの URL にあります。たとえば、このマニュアルの Part No は 820-0500 です。

Application Server Enterprise Edition 8.2 の概要

Sun Java System Application Server Enterprise Edition 8.2 は、大規模な本稼働環境を持つ J2EE アプリケーションおよび Java テクノロジベースの Web サービスを開発および配備するための、J2EE 1.4 プラットフォーム互換サーバーです。

この章で説明する内容は、次のとおりです。

- 9 ページの「8.2 リリースの新機能」
- 11 ページの「ハードウェアとソフトウェアの要件」
- 21 ページの「Enterprise Edition 8.2 リリースで修正されたバグ」
- 24 ページの「HADB に関する追加情報」
- 34 ページの「互換性の問題」
- 36 ページの「J2EE のサポート」
- 37 ページの「サポートされている別の Java バージョンへの切り換え」
- 38 ページの「高パフォーマンス」
- 38 ページの「スケーラビリティ」
- 38 ページの「JavaServer Faces 1.1 のサポート」

8.2 リリースの新機能

Application Server Enterprise Edition 8.2 には、次の拡張機能が含まれています。

- 改善された管理機能 — Application Server では、複数のマシンで構成される複雑なエンタープライズ配備を、セキュリティ保護された状態でリモート管理できます。このリモート管理には、ブラウザベースのコンソールか、スクリプトを使用できるコマンド行インタフェースを使用できます。また、セキュリティ保護された状態でリモートから管理および監視機能にプログラムアクセスするための JMX ベースの豊富な API も用意されています。
- **Message Broker** — Application Server には、高可用性、信頼性、高パフォーマンス、およびスケーラブルなメッセージングを実現する統合エンタープライズクラスメッセージブローカが含まれています。
- **Message Queue 3.7** — Application Server に MQ 3.7 が実装されるようになりました。

- プラットフォームサポートの拡張 — サポートするオペレーティングシステム、データベース、ロケール、およびハードウェアが追加されています。
- **Sun Java Enterprise System** — Sun Java Enterprise System の主要な構成要素として、Application Server はポータルサービスおよびネットワークアイデンティティサービスと緊密に統合されています。
- 移行ツールとアップグレードツール — これらのツールにより、J2EE アプリケーションの規格合致性および移植性を保証し、別の J2EE Application Server (JBoss、WebLogic、WebSphere) からの移行に役立ち、以前のバージョンの Sun ONE Application Server や iPlanet Application Server からのアップグレードを支援することが可能になります。
- **Java 2 Standard Edition 5.0** のサポート — Application Server は、管理および監視のための拡張機能とパフォーマンスおよびスケーラビリティについての改善点を数多く含む Java 2 Standard Edition 5.0 をサポートしています。
- **Java Web Services Developer Pack 1.6 (JWSDP)** のプラグインサポート — すべての JWSDP プラグインがサポートされるようになりました。JWSDP 1.6 は <http://java.sun.com/webservices/downloads/1.6/index.html> から無料でダウンロードできます。
- **Java DB** データベースのサポート — Application Server には、[Apache Derby](http://db.apache.org/derby/) (<http://db.apache.org/derby/>) ベースの Java DB データベースが含まれています。Pointbase データベースとの下位互換性は維持されていますが、Application Server で作成した新しいデータベースではデフォルトで Java DB を使用します。Application Server 8.x からアップグレードしたあとは、既存のドメインでは引き続き PointBase を使用しますが、アップグレード後に作成した新しいドメインでは Java DB を使用します。
- **JDBC ドライバー** — Application Server には、Sun の JDBC ドライバが付属しています。
- **Web サービスセキュリティ** — これらのコンテナメッセージのセキュリティメカニズムは、SOAP Web サービス呼び出しのメッセージレベルでの認証（たとえば XML 電子署名および暗号化）を、OASIS WS-Security 規格の X509 およびユーザー名/パスワードのプロファイルによって実装しています。
- **WS-I Basic Profile 1.1** — J2EE 1.4 の仕様で規定されているように、このリリースは Web サービスアプリケーションの相互運用を可能にするための Web Services Interoperability (WS-I) Basic Profile 1.1 を実装しています。
- **iWay** アダプタによるバックエンド接続 — Sun Microsystems は、現時点で 22 の iWay アダプタを、主要なバックエンドシステム (SAP、Siebel、Oracle、CICS、および IBM MQ Series) 向けに再販およびサポートしています。これは、Application Server 環境内から既存の IT アプリケーションを利用できるようにするためです。これらのアダプタは、J2EE Connector Architecture 1.5 仕様と Web サービス (SOAP) 規格をサポートしており、バックエンドアプリケーションに接続するための時間を節減する開発者ツールを含んでいます。

- 最新の **HADB** 管理システム – UNIX™ プラットフォームは、新しい高可用性データベース (HADB) 管理システム (HADB Version 4.4.3) を備えています。このシステムには、データベースサーバー、ODBC 2.5 ドライバ、JDBC 3.0 Type 4 のドライバ、clusql (SQL 文の入力と実行を行うための対話型プログラム)、および管理システムが含まれています。このバージョンにより、SSH や RSH に依存することがなくなりますが、UDP マルチキャスト用にネットワークを設定する必要があります。HADB の要件と制限事項の詳細は、『[Sun Java System Application Server Enterprise Edition 8.2 高可用性 \(HA\) 管理ガイド](#)』を参照してください。
- **Solaris 10** ゾーンのサポート – Application Server は、Solaris 10 システムの大域ゾーンまたは非大域ゾーンのどちらにもインストールできます。Solaris ゾーンについては、[Solaris ゾーン \(http://www.sun.com/bigadmin/content/zones/\)](http://www.sun.com/bigadmin/content/zones/) のページを参照してください。
- 動的コンテンツテクノロジーのサポートの中止 – CGI-bin や SHTML などの動的コンテンツテクノロジーは、サポートされなくなりました。

ハードウェアとソフトウェアの要件

ここでは、Sun Java System Application Server Enterprise Edition 8.2 製品をインストールする前に満たしている必要のある要件を示します。

- 11 ページの「プラットフォーム要件」
- 12 ページの「システムの仮想化のサポート」
- 13 ページの「重要パッチ情報」
- 13 ページの「JDBC ドライバとデータベース」
- 14 ページの「バンドルされている Java DB データベースの使用」
- 18 ページの「サポートされている Web サーバー」
- 18 ページの「ブラウザ」
- 19 ページの「HADB の要件とサポートされているプラットフォーム」
- 20 ページの「Sun Java System Application Server のアップグレード」
- 20 ページの「その他の要件」

プラットフォーム要件

次の表に、Sun Java System Application Server Enterprise Edition 8.2 製品でサポートされているオペレーティングシステムを示します。また、Application Server をインストールして実行するための最小および推奨のメモリ要件も示します。

表 2-1 Sun Java System Application Server 8.2 のプラットフォーム要件

オペレーティングシステム	最小メモリー	推奨メモリー	最小ディスク容量	推奨ディスク容量	JVM ¹
Sun Solaris 9、10 (SPARC) Solaris 9、10 (x86)	512M バイト	1G バイト	250M バイトの空き領域	500M バイトの空き領域	J2SE_5_08
Sun Java Desktop System	512M バイト	1G バイト	250M バイトの空き領域	500M バイトの空き領域	J2SE_5_08
Redhat Enterprise Linux 3.0 U1、4.0	512M バイト	1G バイト	250M バイトの空き領域	500M バイトの空き領域	J2SE_5_08
Windows Server 2000 SP4+ Windows 2000 Advanced Server SP4+ Microsoft Windows Server 2003 Windows XP Pro SP1+	1G バイト	2G バイト	500M バイトの空き領域	1G バイトの空き領域	J2SE_5_08

¹ サポートされるのは、64 ビットではなく 32 ビットの JVM だけです。

注 - 上に示した Application Server のシステム要件と 19 ページの「[HADB の要件とサポートされているプラットフォーム](#)」に示した HADB のシステム要件は、厳密に同じではありません。これは記載の誤りではありません。Application Server と HADB サーバーを別々のマシンで実行するのはめずらしいことではありません。

UNIX では、オペレーティングシステムのバージョンを `uname` コマンドで確認できます。ディスク容量は `df` コマンドで確認できます。

注 - Microsoft Windows プラットフォームで Application Server を実行している場合は、FAT や FAT32 ではなく NTFS ファイルシステムを使用する必要があります。

システムの仮想化のサポート

システムの仮想化は、複数のオペレーティングシステム (OS) インスタンスを共用ハードウェア上で個別に実行できるようにするテクノロジーです。機能的にいうと、仮想化された環境でホストされる OS に配備されたソフトウェアは、通常はベースとなるプラットフォームが仮想化されていることを認識しません。Sun では、精選されたシステムの仮想化と OS の組み合わせについて、その Sun Java System 製品のテストを行っています。これは、Sun Java System 製品が、適切な規模と構成の仮想化された

環境で、仮想化されていないシステム上の場合と同様に引き続き機能することを実証するためのテストです。仮想化された環境における Sun Java System 製品のサポートについては、[System Virtualization Support in Sun Java System Products](#) を参照してください。

重要パッチ情報

Solaris のパッチ要件

Solaris 9 または 10 (x86 または SPARC) を使用する場合には、「Sun 推奨パッチクラス」をインストールすることをお勧めします。このパッチクラスは、SunSolve の [& セキュリティーパッチ \(http://sunsolve.sun.com/\)](http://sunsolve.sun.com/) から入手できます。

RedHat Enterprise Linux 3.0 の追加パッケージ要件

インストールプログラムをはじめとして、この製品のネイティブコンポーネントを実行するには、標準の RedHat Enterprise Linux 3.0 ディストリビューションには含まれていないパッケージをインストールする必要があります。

compat-libstdc++-7.3-2.96.118.i386.rpm をインストールする必要があります。

このパッケージは、<http://rpm.pbone.net/index.php3/stat/4/idpl/843376/com/compat-libstdc++-7.3-2.96.118.i386.rpm.html> からダウンロードできます。

JDBC ドライバとデータベース

Sun Java System Application Server は、対応する JDBC ドライバを備えた任意の DBMS への接続をサポートするように設計されています。Sun がテストし、J2EE 準拠のデータベース設定を行うのに使用できることが確認されたコンポーネントを、次の表に示します。

表 2-2 J2EE 準拠の JDBC ドライバ

JDBC ベンダー	JDBC ドライバのタイプ	サポートされるデータベースサーバー
i-net Software	Type 4	Oracle (R) 8.1.7、9i、9.2.0.3+、10.1.x、10.2.x Sybase ASE 12.5 Microsoft SQL Server 2000 4.0 Service Pack 1
IBM	Type 2	IBM DB2 8.1 Service Pack 3+
Java DB	Type 4	Apache Derby 10.1.3
PointBase	Type 4	PointBase Network Server 5.2

表 2-2 J2EE 準拠の JDBC ドライバ (続き)

JDBC ベンダー	JDBC ドライバのタイプ	サポートされるデータベースサーバー
DataDirect	Type 4	Oracle (R) 8.1.7、9i、9.2.0.3+、10.1.x、10.2.x Sybase ASE 12.5.2 Microsoft SQL Server IBM DB2 8.1 Service Pack 3+
MySQL	Type 4	5.x
Sun Java System JDBC Driver for Oracle	Type 4	Oracle (R) 9.2.0.3、10G
Sun Java System JDBC Driver for DB2	Type 4	IBM DB2 8.1 Service Pack 3+
Sun Java System JDBC Driver for Sybase	Type 4	Sybase ASE 12.5.2
Sun Java System JDBC Driver for Microsoft SQL Server	Type 4	Microsoft SQL Server 2000 4.0 Service Pack 1
Oracle	Type 4、Type 2	Oracle (R) 9.2.0.3、10G

バンドルされている Java DB データベースの使用

この節では、Application Server 8.2 にバンドルされている Java DB データベースの実装を使用する手順について説明します。

- 14 ページの「Java DB データベースの起動と停止」
- 15 ページの「Java DB ユーティリティースクリプト」
- 15 ページの「Pointbase から Java DB へのテーブルのエクスポート」

Java DB データベースの起動と停止

Sun Java System Application Server 8.2 には、Java DB ネットワークサーバーの起動と停止を行うための 2 つの新しい `asadmin` コマンドが導入されています。

- `start-database` コマンドを使用すると、Java DB Network Server のインスタンスを起動できます。

```
start-database [--dbhost 0.0.0.0] [--dbport 1527] [--dbhome path/derby]
```

ホストのデフォルト値は `0.0.0.0` です。この値によって、Java DB は IP/ホスト名のインタフェースと同様に `localhost` で待機できます。`dbhome` プロパティの値は、Java DB データベースの場所です。デフォルトの `path` は `<appserver_install_dir>/derby` です。

- `asadmin stop-database` コマンドを使用すると、実行中の Java DB ネットワークサーバーのインスタンスをシャットダウンできます。

```
stop-database [--dbhost 0.0.0.0] [--dbport 1527]
```

Java DB ユーティリティースクリプト

Application Server 8.2 とともに出荷される Java DB の構成には、Java DB の使用に役立つ便利なスクリプトも含まれます。<appserver_install_dir>/derby/frameworks/NetworkServer/bin ディレクトリには、次の使用可能なスクリプトがあります。

- startNetworkServer.ksh/bat: ネットワークサーバーを開始するスクリプト
- stopNetworkServer.ksh/bat: ネットワークサーバーを停止するスクリプト
- ij.ksh/bat: 対話式の JDBC スクリプト記述ツール
- dblook.ksh/bat: データベースの DDL の全部または一部を表示するスクリプト
- sysinfo.ksh/bat: Java DB 環境に関するバージョン情報を表示するスクリプト
- NetworkServerControl.ksh/bat: NetworkServerControl API でコマンドを実行するためのスクリプト

▼ Java DB ユーティリティースクリプトを実行するために環境を設定するには

- 1 <appserver_install_dir>/derby ディレクトリをポイントするように、DERBY_INSTALL 環境変数を設定します。
- 2 CLASSPATH 環境変数の設定を解除します。
- 3 次のプロパティを設定することもできます。
 - a. DERBY_SERVER_HOST をネットワークサーバーの待機先ホストとして設定します。
0.0.0.0 に設定すると、すべてのリスナーを有効にできます。
 - b. DERBY_SERVER_PORT をネットワークサーバーの待機先ポート番号として設定します。

参照 これらのユーティリティの詳細については、Derby の [ツール](http://db.apache.org/derby/docs/10.1/tools/) (<http://db.apache.org/derby/docs/10.1/tools/>) および [Admin](http://db.apache.org/derby/docs/10.1/adminguide/) (<http://db.apache.org/derby/docs/10.1/adminguide/>) に関するガイドを参照してください。

Pointbase から Java DB へのテーブルのエクスポート

この例は、Pointbase のテーブルの DDL をキャプチャーし、Netbeans 5.0 を使用して Java DB に同じテーブルを作成する方法を示しています。これと同じ事を行うには、コマンドツールと unload database コマンドを使用する方法もあります。

```
./startcommander.sh
Do you wish to create a new Database. (Yes (Y) or No (N))? [default: N]:
Enter product to connect with: (Embedded (E) or Server (S))? [default: E]: e
Enter driver to use? [default: [com.pointbase.jdbc.jdbcUniversalDriver]]:
Enter database URL? [default: [jdbc:pointbase:embedded:sample]]:
Enter Username? [default: PBPUBLIC]:
Enter Password? [default: PBPUBLIC]:
```

PointBase Commander 5.2 ECF build 294 size restricted version EMBEDDED

Interactive SQL command language. SunOS/5.9

(C) Copyright 2004 DataMirror Mobile Solutions, Inc. All rights reserved.

Licensed to: Sun_customer_demo_use
For commercial version contact PointBase at:
pointbase.com
PHONE: 1-877-238-8798 (US & CANADA)
1-408-961-1100 (International)
WEBSITE: www.pointbase.com

```
SQL>unload database sampledб.sql;
SQL> unload database sampledб.sql;
SQL> 13 Row(s) Unloaded. (PBPUBLIC.CUSTOMER_TBL)
SQL> 4 Row(s) Unloaded. (PBPUBLIC.DISCOUNT_CODE_TBL)
SQL> 30 Row(s) Unloaded. (PBPUBLIC.MANUFACTURE_TBL)
SQL> 11 Row(s) Unloaded. (PBPUBLIC.MICRO_MARKETS_TBL)
SQL> 9 Row(s) Unloaded. (PBPUBLIC.OFFICE_TBL)
SQL> 4 Row(s) Unloaded. (PBPUBLIC.OFFICE_TYPE_CODE_TBL)
SQL> 15 Row(s) Unloaded. (PBPUBLIC.ORDER_TBL)
SQL> 6 Row(s) Unloaded. (PBPUBLIC.PRODUCT_CODE_TBL)
SQL> 30 Row(s) Unloaded. (PBPUBLIC.PRODUCT_TBL)
SQL> 10 Row(s) Unloaded. (PBPUBLIC.SALES_REP_DATA_TBL)
SQL> 10 Row(s) Unloaded. (PBPUBLIC.SALES_REP_TBL)
SQL> 52 Row(s) Unloaded. (PBPUBLIC.SALES_TAX_CODE_TBL)
SQL> 12 Table(s) Unloaded.
SQL> quit;
```

この例の unload database コマンドを実行した結果は、sampledb.sql ファイルに書き込まれます。sampledb.sql ファイルには、必要なテーブルとインデックスを作成するために必要となる DDL がすべて含まれています。また、データベースにデータを挿入しなおすための DML も含まれます。コマンドの RUN コマンドは、生成されたスクリプトを使用して別の Pointbase データベースにデータをインポートするために使用されます。生成されたファイル内の INSERT 文と関連するデータの例を次に示します。

```
INSERT INTO "ADVENTURE"."CATEGORY" (
"CATID", "LOCALE", "NAME", "DESCRIPTION", "IMAGEURI" )
```

```
VALUES( ?, ?, ?, ?, ? );
{
'ISLAND          ','en_US','Island Adventures','Experience an island /
paradise in a way fit for your needs.','Island_Adventures.gif'
'JUNGLE          ','en_US','Jungle Adventures','Experience a jungle /
paradise in a way fit for your needs.','Jungle_Adventures.gif'
'MOUNTAIN        ','en_US','Mountain Adventures','Experience an /
elevated paradise with a view.','Mountain_Adventures.gif'
'ORBITAL         ','en_US','Orbital Adventures','Experience a vacuum /
paradise with a beautiful view and where no one can hear you scream.',' /
'Space_Adventures.gif'
'WESTERN         ','en_US','Western Adventures','Enjoy the Wild West. /
','Western_Adventures.gif'
'SOUTH_POLE      ','en_US','South Pole Adventures','Experience a /
frozen paradise in a way fit for your needs.','SouthPole_Adventures.gif'
};
```

コマンドの `unload database` コマンドによって生成されたファイルを、DDL だけで構成されるように簡単に編集できます。たとえば、`insert` 文を処理するプログラムを記述することは難しくありません。簡単なテストとして、Pointbase sample データベースに対して `unload database` コマンドを使用し、生成されたスクリプトを編集して次のように変更します。

- すべての `CREATE Table` 文の末尾から `Organization Heap` という部分を削除します。
- `COMMIT` コマンドを削除します。
- ブール値 `datatype` を `smallint` に変更します。
- すべての `INSERT` 文とそれに関連するデータを削除します。

次に、簡単な Ant スクリプトを使用し、`sql` ターゲットを使用して DDL を実行します。最後に、`sun-appserv-samples` データベースに対して同じ試行を繰り返します。このとき、生成された SQL ファイルに次の変更を追加する必要があります。

- サンプルのデータベースに対して、ここで説明した変更をすべて加えます。
- `create user` コマンドを削除します。
- `SET PATH` コマンドを削除します。
- `Decimal` の精度を 38 から 31 以下に変更します。
- `float` の精度を 64 から 52 以下に変更します。
- `CREATE PROCEDURE` コマンドの `SPECIFIC` キーワードは現在サポートされていません。
- `GRANT` コマンドを削除します。

Pointbase の Java プロシージャを Java DB で動作するように変換するには、`CREATE PROCEDURE` 文と同様に Java コードも変更する必要があります。Java DB の Java プロシージャの詳細については、[Derby Reference manual](http://db.apache.org/derby/docs/10.1/ref/) (<http://db.apache.org/derby/docs/10.1/ref/>) を参照してください。Boolean データ型は、次期リリースの Java DB でサポートされる予定です。

サポートされている **Web** サーバー

ここでは、Sun Java System Application Server Enterprise Edition 8.2 向けにサポートされている Web サーバーを示します。

表 2-3 サポートされている Web サーバー

Web サーバー	バージョン	オペレーティングシステム
Sun Java System Web Server	6.0、 6.1、 7.0	Solaris SPARC 9、 10 Solaris x86 9、 10 Red Hat Enterprise Linux 3 および 4
Apache Web Server	1.3+、 1.4、 2.0	Solaris SPARC 9、 10 Solaris x86 10 Red Hat Enterprise Linux 3 および 4 Windows Server 2003 Windows 2000 Advanced Server SP4+ Windows Server 2000 SP4+ Windows XP Pro SP1+
Microsoft IIS™	5.0+	Windows Server 2003 Windows 2000 Advanced Server SP4+ Windows Server 2000 SP4+ Windows XP Pro SP1+

ブラウザ

ここでは、Sun Java System Application Server Enterprise Edition 8.2 でサポートされているブラウザを示します。

表 2-4 サポートされている Web ブラウザ

ブラウザ	バージョン
Mozilla	1.4、 1.5、 1.6、 1.7.x
Netscape Navigator	4.79、 6.2、 7.0、 8.x
Internet Explorer	5.5 Service Pack 2、 6.0
Firefox	1.4、 1.5

HADB の要件とサポートされているプラットフォーム

11 ページの「ハードウェアとソフトウェアの要件」に示されている要件に加えて、システムが、下に示されている HADB を実行するための要件を満たしていることを確認してください。

注 - 11 ページの「プラットフォーム要件」に示した Application Server のシステム要件とここに示した HADB のシステム要件は、厳密に同じではありません。これは記載の誤りではありません。Application Server と HADB サーバーを別々のマシンで実行するのはめずらしいことではありません。

- 19 ページの「サポートされるプラットフォーム」
- 19 ページの「HADB サーバーホストの要件」
- 20 ページの「HADB 管理ホストの要件」
- 20 ページの「HADB クライアントホストの要件」

注 - システムの Java コンポーネントは JDK 1.4.2_02 を使用して構築され、JDK 1.5_09 上でテストされています。

サポートされるプラットフォーム

- **Solaris (SPARC)** - Solaris 8 MU7、Solaris 9 MU7、Solaris 10 RR。
- **Solaris (x86)** - Solaris 9 MU7、Solaris 10 RR。
- **RedHat Enterprise Linux** - 2.1 U5 (ext3 はサポートされず、ext2 ファイルシステムのみがサポートされている)、3.0 U4 (ext2 と ext3 の両方がサポートされている。U4 より前のアップデートは、過剰なスワッピングのために推奨されない)。これらのオペレーティングシステムバージョンでは、HADB が 32 ビットモードでのみテストされていることに注意してください。また、HADB が、オペレーティングシステム内のバグのために 64 ビットモードで動作している RedHat Enterprise Linux 3.0 をサポートしていないことにも注意してください (HADB への影響の詳細は、55 ページの「高可用性」の既知のバグ 6249685 を参照)。
- **Microsoft Windows** - Microsoft Windows 2000 Advanced Server Service Pack 4 および Microsoft Windows 2003 Enterprise Edition。HADB が、間もなく登場する、64 ビットモードの Microsoft Windows オペレーティングシステムバージョンをまったくサポートしていないことに注意してください。

HADB サーバーホストの要件

- 最小メモリー - ノードあたり 512M バイト。
- 最小空きディスク容量 - ホストあたり HADB バイナリ用に 70M バイト。さらに、データデバイスにもディスク容量が必要、ノードあたりテストインストール用に 512M バイト。
- 推奨メモリー - ノードあたり 1G バイト。

- 推奨空きディスク容量 - ホストあたり HADB バイナリ用に 70M バイト。さらに、データデバイスにもディスク容量が必要、ノードあたりテストインストール用に 1200M バイト。

注 - HADB データとログファイルを格納するデバイスで書き込みキャッシュが無効になっていることを確認してください。一部の Solaris プラットフォーム (たとえば、Solaris x86) では、デフォルトで書き込みキャッシュが有効になります。

HADB 管理ホストの要件

- 最小メモリー - 128M バイト
- 最小空きディスク容量 - ノードあたり HADB バイナリ用に 70M バイト

HADB クライアントホストの要件

- 最小メモリー - 120M バイト
- 最小空きディスク容量 - 20M バイト

Sun Java System Application Server のアップグレード

Application Server の以前のどのリリースからも、上書きアップグレードはサポートされていません。以前のバージョンの Application Server から現在のバージョンへのアップグレードに関する詳細な指示については、『Application Server Enterprise Edition Upgrade and Migration Guide』を参照してください。

その他の要件

Sun Java System Application Server ソフトウェアをインストールする前に、次の追加の要件が満たされるようにしてください。

- 空き領域 — 一時ディレクトリには、少なくとも、Sun Java System Application Server のインストール用に 35M バイトの空き領域と、SDK のインストール用に 250M バイトの空き領域が必要です。
- アンインストールプログラムの使用 — システムから Application Server を削除する必要がある場合は、ソフトウェアに付属のアンインストールプログラムを使用することが重要です。他の方法を使って削除すると、同じバージョンの再インストールや新しいバージョンのインストールを行うときに問題が発生する可能性があります。
- 空きポート — 利用可能な未使用ポートが 7 つ必要です。
 - インストールプログラムは、使用されているポートを自動的に検出し、その時点で未使用のポートをデフォルトの設定として示します。デフォルトでは、HTTP については 8080、HTTPS については 8181、Administration Server については 4849 が初期デフォルトポートです。

- インストールプログラムは、使用されているポートを検出し、それ以外の2つのポートを割り当てます。そのポートは、Sun Java System Message Queue (デフォルトでは、7676) と、IIOP (デフォルトでは、IIOP の場合は3700、IIOP/SSL の場合は1060 と1061) です。これらのデフォルトポート番号が使用されている場合、インストールプログラムは動的ポートの範囲から無作為にポート番号を割り当てようとします (これは利用可能な次のポート番号とはかぎらない点に注意)。

インストール済みサーバーの起動 (UNIX) — インストール済みサーバーを置き換える場合以外は、Sun Java System Application Server 8.2 のインストール処理を開始する前に、インストール済みサーバーを起動してください。インストールプログラムが使用中のポートを検出するため、使用中のポートがほかの目的に使用されることはありません。

- インストール済みサーバーの置き換え (UNIX) (UNIX) — 古いバージョンの Sun Java System Application Server がインストールされており、それを現在のバージョンの Application Server に置き換える場合は、新しいサーバーをインストールする前に、インストール済みサーバーを停止してください。インストールプログラムのアップグレードウィザードを使用してサーバーをアップグレードしてください。
- ファイアウォールのシャットダウン (Microsoft Windows) — Sun Java System Application Server ソフトウェアをインストールする前に、すべてのファイアウォールソフトウェアを停止する必要があります。一部のファイアウォールソフトウェアはデフォルトですべてのポートを無効にするからです。インストールプログラムは、使用可能なポートを正確に検出する必要があります。

互換性に関するその他の情報については、『[Sun Java System Application Server Enterprise Edition 8.2 アップグレードと移行](#)』を参照してください。

Enterprise Edition 8.2 リリースで修正されたバグ

ここでは、お客様から報告された問題のうち、Sun Java System Application Server Enterprise Edition 8.2 製品で解決されたものを示します。

バグ番号	説明
6368745	AS: AS7 (Java ES 2) から AS8.2 (Java ES 5) にアップグレードできません
6432308	AS、JES5b7a: JES2 から JES5 への asupgrade が失敗します
6378409	AS 8.2: 8.2 に含まれている jsf ライブラリが原因で下位互換性が壊れています
6371534	AS82EE: インストールパスに空白が含まれていると、Windows で configure-ha-cluster がハングアップします

バグ番号	説明
6242761	マニュアルに従って init でノードエージェントを起動しようとすると、エラーが生成されます
6267772	Borland Optimizelt の設定に関する説明が正しくありません
6273226	Windows サービスとして実行されるサーバーまたはノードエージェントを実行する場合 -xrs JVM オプションを追加する必要がある旨の説明テキストをマニュアルに追加します
6361145	8.1EE から 8.2EE に上書きアップグレードするときに、LB プラグインをアップグレードできません
6362881	8.1ur2 から 8.2ee にアップグレードするときに、インストーラにアップグレードオプションが表示されません
6325988	FVD/codeBase での最初の着信 RMI-IIOP 要求で相互運用性に関する問題が発生します
6363689	JES5 ASE8.2 build03 - インスタンスを停止できません
6364900	ある Web アプリケーションに別の Web アプリケーションが含まれていると、フェイルオーバー時にセッション値が失われます
6370993	クラスタのアプリケーションコンテキストルートを「/」に変更すると、セッションフェイルオーバーが失敗します
6373729	ORB 衝突が原因で、Appserver 8.1 のコードで WebLogic 9.0 と通信できません
6377594	Weblogic の initialcontext ファクトリで検索に関する問題が発生します
6381538	スタンドアロンクライアントに NPE で障害が発生します
6406055	警告: 「IOP00110205: (BAD_PARAM) オブジェクト参照が外部 ORB から発せられています」 org.omg.CORBA.
6388329	Access Manager のアップグレード後に Application Server で JSP コンパイルエラーが発生します
6419659	transport-guarantee が CONFIDENTIAL になっていると、要求が LB プラグインによって正しくリダイレクトされません
6390584	OutOfMemoryError: PermGen 空間
6401424	PDF ファイルの要求に対して、libns-httpd40.so の service_plain_range で SEGV が発生します。
6401704	AppServer 8.# の WebDAV サポートが必要です
6416478	JSP の TestSuite の障害: javax.servlet.jsp.el.ELEXception
6438908	relativeRedirectAllowed=true のときにヘッダーの場所が壊れます

バグ番号	説明
6456553	Cookie を応答に追加すると、 <code>java.lang.IllegalArgumentException</code> がスローされます
6295010	ファイアウォールと衝突するアイドルタイムアウトの通常プールの接続がチェックされません
6350435	2つのデータベースに対する XA 操作時に、Application Server がデータベースの障害処理に失敗します
6377830	同じ接続を次のユーザーが使用すると、 <code>setAutoCommit</code> の <code>false</code> 設定が伝播されます
6399830	IT 319: パスワードエイリアス機能が <code>domain.xml</code> で機能していません
6360040	SJAS 8.x: AppServer LDAP レルムバインドユーザーがすべてのグループとメンバーにアクセスする傾向があります
6370095	<code>acceptor-thread</code> を 10 より高く設定できません
6399365	<code>InvokerServlet</code> が Enterprise Edition だけで機能するようになっていません
6303835	余分なログ: サーバーログに誤解を招くセキュリティーメッセージがあります
6349541	8.1 EE UR2 - 特定の IP アドレスにバインドするように SSL リスナーを設定できません...
6380040	ログファイルのクリーンアップの自動化が必要です
6387278	クライアント認証が壊れているか、スレッドに対して安全ではありません (<code>ProgrammaticLogin</code>)
6407896	<code>getUserPrincipal()</code> をオーバーライドする <code>HttpServletRequestWrapper</code> によって <code>ClassCastException</code> が発生します
6321194	ラウンドロビンポリシーが機能していません
6362269	インストールパスに空白が含まれていると、Windows でベリファイアが正しく動作しません
6365888	デフォルトのコネクタ接続プールからの接続が使用対象に含まれません
6369554	接続をアプリケーションに付与する前に接続プールで接続の妥当性検査を行う必要があります
6370574	「あとで設定」を使って AS をアップグレードしたあと、 <code>/var/opt/SUNWappserver</code> ディレクトリがありません
6371723	すべての Web サーバー、特に Apache <code>mod_loadbalancer</code> で、 <code>lbplugin</code> からメモリーがリークします
6395390	処理を継続する http 要求でラウンドロビンが起こりません。

バグ番号	説明
6402713	ロードバランサがHTTPS 要求への接続に失敗します
6409992	証明書を使った 8.1pe から 8.2EE へのアップグレードに失敗しました
6413224	アップグレードツールがアップグレード証明書オプションをスキップしました
6422893	HTTPS ルーティングが機能しません
6424051	8.xPE から 9.1 EE へのアップグレードで、既存の管理資格と MP を使用する必要があります
6424053	8.XEE から 9.1EE へのアップグレードが start-domain 例外のために失敗します
6430394	n/w 停止があると、メッセージが失われます。
6444052	JMS Version 1.5 用の Generic RA が AS 8.2 EE に統合されます
6444308	EE のブロック: AS 8.1 UR2 EE -> 8.2 EE SS: 8.2 の domain1 を起動できません。正しくない 8.1 UR2 のドメインが起動されます
6444368	win2003 の並列 GUI アップグレードで、8.0PE UR1 から 9.1 ee へのアップグレードがハン グアップします
6446558	connector-connection-pool リソースで手動のトランザクション復旧が機能しません。
6447895	組み込み RA を使用するリソースでトランザクション復旧が機能していません。
6454007	アップグレードツールに必要な入力に変更されます
6455396	8.1EE から 9.1EE SBS へのアップグレード後に、node-agent とインスタンスの起動に失敗します。
6374533	パフォーマンスと安定性の理由から、Application Server は XWSS 1.0 ではなく XWSS 1.1 をバンドルすべきです
6358422	Appserver 7.1/8.1 EE: Web サーバー LB のプロキシプラグインは、キープアライブ接続を正しくサポートすべきです
6382063	com.sun.enterprise.iop.IORToSocketInfoImpl にメモリーリークがあります

HADB に関する追加情報

ここでは、Application Server 8.2 に含まれている HADB 実装に関する重要な追加情報を示します。

- [25 ページの「HADB の拡張機能」](#)
- [26 ページの「HADB のファイルシステムサポート」](#)
- [26 ページの「高可用性データベースのアップグレード」](#)

- 33 ページの「既知の SQL の制限事項」
- 34 ページの「高可用性ロードバランス」

HADB の拡張機能

- データベース管理に使用するパスワードの変更を可能にするために、新しい管理コマンド `hadbm setadminpassword` が実装されました。このコマンドは、使用する管理エージェントを示すオプションと、古いパスワードおよび新しいパスワードを取ります。詳細は、`hadbm setadminpassword` のマニュアルページを参照してください。
- 既存の管理コマンド `hadbm listpackages` が変更されています。以前、このコマンドはオペランドを取らず、関連する管理ドメイン内のすべてのパッケージを表示していました。変更によって、オプションのパッケージ名オペランドが導入され、その名前を持つパッケージのみが表示されます。そのオペランドが指定されない場合は、すべてのパッケージが表示されます。詳細は、`hadbm listpackages` のマニュアルページを参照してください。
- 既存の管理コマンド `hadbm createdomain` が変更されています。*hostlist* オペランドが、管理エージェントのポート番号も指定できるように拡張されました。この方法により、*hostlist* オペランドのみを使用してドメインが完全に指定されます。下位互換性のために、以前の動作も引き続きサポートされています。詳細は、`hadbm createdomain` のマニュアルページを参照してください。
- 管理システムからのエラーメッセージの一部が変更されています。これらの変更は、エラーメッセージのわかりやすさ、一貫性、および正確性を向上させることを目的にしています。実際の変更は、このリリースノートには示されていません。
- インストールとアンインストールの動作が若干変更されています。HADB のインストールまたはアンインストールでは、ソフトリンク `/opt/SUNWhadb/4` が常に保持されるべきですが、必ずしもそのとおりにはなっていません。
- コマンド行でパスワードをコマンドオプションとして入力する場合がありますが、この方法は推奨されません。これは、コマンド行オプションとしてパスワードを取るすべての `hadbm` コマンドに当てはまります。`hadbm` コマンドでは、従来より、パスワードを次の方法で入力できるようになっています。
 1. パスワードファイル
 2. コマンド行オプション
 3. 対話型の入力

2つ目の方法のコマンド行オプションは安全でないと見なされるため、推奨されません。この方法でパスワードが入力されると、警告メッセージが表示されます。代わりに、1つ目の方法のパスワードファイルか、または3つ目の方法の対話型の入力を使用してください。コマンド行でのパスワードの使用は、次のリリースでは廃止される予定です。これは、コマンド行のパスワードオプションを取るすべての `hadbm` コマンドに適用されることに注意してください。

- HADB は、JGroups Version 2.2 を使用するようにアップグレードされており、そのソースコードは HADB とともに配布されます。以前の HADB バージョンからのオンラインアップグレードをサポートするために、JGroups 2.1 および 2.2 の両方が HADB とともに提供されます。JGroups 2.1 の場合は、バイトコードのみが提供されます。

HADB のファイルシステムサポート

次のファイルシステムを使用するよう HADB を設定する場合には、重要な考慮事項がいくつかあります。

- **ext2** および **ext3**– HADB は、Red Hat Application Server 3.0 用に ext2 および ext3 ファイルシステムをサポートしている。Red Hat Application Server 2.1 については、HADB は ext2 ファイルシステムしかサポートしていない。
- Veritas– Solaris プラットフォームで Veritas File System を使用すると、「WRN: Direct disk I/O mapping failed」というメッセージが履歴ファイルに書き込まれる。このメッセージは、データデバイスおよびログデバイスについて HADB が直接入出力を有効にできないことを示している。直接入出力は、ディスクページに書き込むための CPU コストを節減することによってパフォーマンスを向上させる。また、「ダーティー」なデータページを管理するためのオペレーティングシステムのオーバーヘッドを減らす。

Veritas File System で直接入出力を利用するには、次の方法の 1 つを使います。

- オプション `mincache=direct` でマウントされたファイルシステム上に、データデバイスとログデバイスを作成します。このオプションは、そのファイルシステム上に作成されるすべてのファイルに適用されます。詳細は、`mount_vxfs(1M)` コマンドを参照してください。
- Veritas Quick 入出力ユーティリティを使用して、ファイルシステムファイルに対する raw 入出力を行います。詳細は、『VERITAS File System 4.0 Administrator's Guide for Solaris』を参照してください。

これらの設定は、Application Server 8.2 ではテストされていないことに注意してください。

Application Server ソフトウェアでの HADB のインストールと設定については、『Application Server Enterprise Edition 高可用性 (HA) 管理ガイド』を参照してください。

高可用性データベースのアップグレード

- 27 ページの「アップグレード前の作業とデータ移行」
- 27 ページの「アップグレード手順」
- 28 ページの「アップグレードのテスト」
- 29 ページの「配備とアップグレードに関する特別な情報」

▼ アップグレード前の作業とデータ移行

始める前に ユーザーは、HADB 履歴ファイル、管理エージェント設定ファイル、ログファイルとリポジトリ、およびインストールパスの外部にあるすべてのデータデバイスを保管するようにしてください。この作業がまだ完了していない場合は、アップグレードの前に実行してください。管理リポジトリと設定ファイルを移動するには、次の手順に従います。

- 1 すべての古い管理エージェントを停止し、**HADB ノード**は動作したままにします。
- 2 各ホスト上で、リポジトリディレクトリを新しい場所に移動します。
- 3 各ホスト上で、dbconfig ディレクトリを新しい場所にコピーします。
- 4 各ホスト上で、mgt.cfg ファイルをアップデートし、dbconfig とリポジトリディレクトリの正しいパスを設定します。
- 5 アップデートされた mgt.cfg ファイルを使用して管理エージェントを起動します。

▼ アップグレード手順

HADB Version 4.4.x から Version 4.4.3 にアップグレードするには、次の手順を実行します。

- 1 必要に応じて、上で説明したアップグレード前の作業を実行します。
- 2 **HADB Version 4.4.3** をすべての **HADB** ホストにインストールします。パスは **Version 4.4.x** とは別のパス、たとえば /opt/SUNWhadb/4.4.3 にします。
- 3 **HADB Version 4.4.3** を、**HADB** ホストとは別の hadbm クライアントホストにインストールします。
- 4 すべての **HADB** ホスト上で実行されているすべての管理エージェントを停止します。
- 5 新しいバージョンのソフトウェアを使用して(ただし、設定ファイルは古いまま)、管理エージェントプロセスを起動します。残りの手順では、新しいバージョンの bin ディレクトリにある hadbm コマンドを使用してください。
- 6 管理ドメインでパッケージを登録します。デフォルトのパッケージ名が **V4.4** になるので、同じ名前を持つ既存のパッケージとの競合を避けるために別のパッケージ名が必要になる場合があります。

```
hadbm registerpackage --packagepath=/opt/SUNWhadb/4.4.3 V4.4.3
```
- 7 hadbm listpackages コマンドを実行し、ドメインに新しいパッケージが登録されていることを確認します。

- 8 新しい hadbm **Version 4.4.3** を使用してデータベースを再起動します。デバイスと履歴ファイルを移動する必要がある場合は、オンラインアップグレードを、デバイスと履歴ファイル用の新しいパスの設定とともに1回の操作で実行します。

```
hadbm set packagename=V4.4.3,devicepath=new_devpath,  
historypath=new_histpath
```

そうでない場合、つまりデバイスと履歴ファイルがすでにインストールディレクトリの外にある場合は、ノードの順次再起動のみを行う次のコマンドを実行します。

```
hadbm set packagename=V4.4.3 database name
```

- 9 データベースが「実行中」の状態にあり (hadbm status コマンドを使用して確認)、かつ正常に動作してクライアントトランザクションを処理していることを確認します。
 - 10 すべての正常に動作している場合は、あとで古いインストールを削除することができます。古いパッケージの登録を解除する前に、古いパッケージへのすべての参照を `ma` リポジトリから削除します。そうしないと、「使用中のパッケージ」のために `hadbm unregisterpackage` が失敗します。たとえば、ダミーの再設定操作 `hadbm set connectiontrace=same as previous value` によって、古いパッケージへのすべての参照が削除されます。ここで、古いパッケージの登録を解除します。
- ```
hadbm unregisterpackage [--hosts=host-list] old pacakge name
```
- 11 ファイルシステムから古いインストールを削除します。

## ▼ アップグレードのテスト

Solaris で、アップグレードの成功をテストするには、アップグレードが正常に実行されたことを確認します。

- 1 動作中のプロセスが新しいバイナリを使用していることを確認してください。すべての **HADB** ノードで、次のことを確認します。

```
new path/bin/ma -v
new path/bin/hadbm -v
```

- 2 データベースが動作中かどうかを確認します。次のコマンドによって、すべての **HADB** ノードが「実行中」の状態にあることが表示されます。

```
new path/bin/hadbm status -n
```

- 3 **HADB** を使用している製品のポインタが、新しい **HADB** パスを指すように変更されていることを確認します。

- 4 HADB を使用している製品では、独自のアップグレードテストを実行して、HADB アップグレードも動作していることを確認できます。

オンラインアップグレードの後、新しいバージョンが正常に動作しない場合は、以前の HADB バージョンの使用に戻してください。ただし、管理エージェントリポジトリが変更されている場合は、HADB 自体はダウングレードできますが、新しい管理エージェントを引き続き動作させる必要があります。

#### 配備とアップグレードに関する特別な情報

ここでは、HADB の配備とアップグレードに関する追加情報を示します。

- [29 ページの「配備」](#)
- [32 ページの「4.4.1 から 4.4.2 へのオンラインアップグレード」](#)

#### 配備

- デバイス、ログ、および履歴ファイルはローカルディスクにのみ格納するようにし、リモートマウントのファイルシステムは使用しないでください。
- ホストに複数のノードが配置されている場合は、各ノードに所属するデバイスを別のディスクに保管することをお勧めします。そうしないと、ディスクの競合によってパフォーマンスが低下します。この問題の症状は、「BEWARE - last flush/fputs took too long」などのメッセージによって履歴ファイルで確認できます。1つのノードに複数のデータデバイスファイルがある場合は、これらのデバイスファイルに別々のディスクを使用することをお勧めします。
- ローカルディスク (可能であれば、データデバイスに使用されているものとは別のディスク) を使用して、HADB ホストに HADB バイナリをインストールしてください。NFS の遅延またはディスクの競合によって、履歴ファイルに「Process blocked for nnn, max block time is nnn」という警告が出力され、ノードが再起動される可能性があります。
- HADB デバイス、履歴ファイル、管理エージェントディレクトリ、およびエージェント設定ファイルを HADB パッケージのパスには配置しないでください。これを行うと、新しいバージョンにアップグレードし、古いパッケージのパスを削除したときに問題が発生します。
- このリリースの HADB では、最大 28 ノード (24 のアクティブデータノード、および 4 つのスペア) が正式にサポートされています。
- JDBC ドライバおよび HADB サーバーには同じバージョンを使用することをお勧めします。
- IPv6 はサポートしていません。IPv4 のみです。
- Windows でのコマンド行の長さは、2048 バイトに制限されています。
- ネットワークを UDP マルチキャスト用に設定する必要があります。
- RedHat Enterprise Linux 3.0 Update 1 ~ 3 は、過剰なスワッピングが見られるため、配備プラットフォームとしてはお勧めできません。この問題は、RedHat Enterprise Linux 3.0 Update 4 では修正されています。

- NSUP をリアルタイム優先度を使用して実行する可能性は次のとおりです。

ノードスーパーバイザー (NSUP) のプロセス (clu\_nsup\_srv) は、タイムリーな方法による「ハートビート」メッセージ交換を利用して、HADB の高可用性を保証します。NSUP がほかのプロセスと同じ場所に配置されていると、このタイミングが影響を受け、リソースの枯渇が発生します。その結果、誤ったネットワークパーティションが発生し、ノードが再起動します (その前に、履歴ファイルに「Process blocked for n seconds」という警告が出力される)。それにより、トランザクションの中止やその他の例外が発生します。

この問題を解決するには、clu\_nsup\_srv (installpath/lib/server に格納されている) の suid ビットを設定し、そのファイルをルートが所有する必要がある必要があります。これを手動で行うには、次のコマンドを実行します。

```
chown root clu_nsup_srv
chmod u+s clu_nsup_srv
```

これにより、clu\_nsup\_srv プロセスは、起動されると、ユーザー root として実行されます。さらに、起動後、自分自身にリアルタイム優先度を自動的に与えることができるようになります。setuid を使用することによるセキュリティへの影響を避けるために、リアルタイム優先度は最初の間だけ設定され、優先度の変更されたらプロセスは有効な uid に戻ります。ほかの HADB プロセスは、優先度をタイムシェアするために自分の優先度を低くします。

NSUP がリアルタイム優先度を設定できない場合は、「Could not set realtime priority」(unix: errno will be set to EPERM) という警告を出力します。この警告は ma.log ファイルに書き込まれ、リアルタイム優先度を使用しないで処理が継続されます。

リアルタイム優先度を設定できない例として、次のような場合があります。

- Solaris 10 の非大域ゾーンにインストールされている場合
- Solaris 10 で、PRIV\_PROC\_LOCK\_MEMORY (プロセスが物理メモリー内のページをロックできる) または PRIV\_PROC\_PRIOCNTL 特権、あるいはその両方が無効になっている場合
- ユーザーが setuid 権限を無効にしている場合
- ユーザーがソフトウェアを tar ファイルとしてインストールしている場合 (App.server 用のルート以外のインストールオプション)

clu\_nsup\_srv プロセスは CPU の消費が少なく、フットプリントも小さいため、リアルタイム優先度を使用して実行してもパフォーマンスには影響しません。

- Solaris での HADB 用の IP ネットワークマルチパスの設定 (Solaris 9 でのみテスト済み)

可能な範囲で最高のネットワーク可用性を保証するために、HADB を実行している Solaris ホストをネットワークマルチパスを使用して設定することをお勧めします。ネットワークマルチパスの設定は、『IP Network Multipathing Administration

Guide』で詳細に説明されています。HADB でマルチパスを使用することにした場合は、後述されている HADB 用のマルチパス設定への対応に進む前に、『IP Network Multipathing Administration Guide』の「Administering Network Multipathing」の節を参照してマルチパスを設定してください。『IP Network Multipathing Administration Guide』は、Solaris 9 の『System Administrator Collection』に含まれており、<http://docs.sun.com> からダウンロードできます。

- ネットワークインタフェース障害検出時間の設定

HADB でマルチパスのフェイルオーバーを適切にサポートするには、`/etc/default/mpathd` 内の `FAILURE_DETECTION_TIME` パラメータで指定されているネットワークインタフェース障害検出時間が 1000 ミリ秒を超えないようにする必要があります。元の値がこの値を超えている場合は、このファイルを編集して、このパラメータの値を **1000** に変更します。

```
FAILURE_DETECTION_TIME=1000
```

変更を有効にするために、次のコマンドを実行します。

```
pkill -HUP in.mpathd
```

- HADB で使用する IP アドレス

『Solaris IP Network Multipathing Administration Guide』で説明されているように、マルチパスを使用するには、物理ネットワークインタフェースをマルチパスインタフェースグループにグループ化する必要があります。このようなグループ内の各物理インタフェースには、物理インタフェースアドレスとテストアドレスの 2 つの IP アドレスが関連付けられます。データの送信に使用できるのは物理インタフェースアドレスのみであり、テストアドレスは Solaris 内部の使用のためにのみ用意されています。`hadbm create --hosts` が実行されると、各ホストは、マルチパスグループの 1 つの物理インタフェースアドレスによってのみ指定されます。

- 例

ホスト 1 とホスト 2 のそれぞれに、2 つの物理ネットワークインタフェースが含まれていると仮定します。各ホスト上で、これらの 2 つのインタフェースをマルチパスグループとして設定し、`ifconfig -a` を実行すると次の出力が得られます。

ホスト 1

```
bge0: flags=1000843<mtu 1500 index 5 inet 129.159.115.10 netmask ffffffff00
broadcast 129.159.115.255 groupname mp0
bge0:1: flags=9040843<mtu 1500 index 5 inet 129.159.115.11 netmask ffffffff00
broadcast 129.159.115.255
bge1: flags=1000843<mtu 1500 index 6 inet 129.159.115.12 netmask ffffffff00
broadcast 129.159.115.255 groupname mp0
bge1:1: flags=9040843<mtu 1500 index 6 inet 129.159.115.13 netmask ff000000
broadcast 129.159.115.255
```

## ホスト 2

```
bge0: flags=1000843<mtu 1500 index 3 inet 129.159.115.20 netmask ffffffff00
broadcast 129.159.115.255 groupname mp0
bge0:1: flags=9040843<mtu 1500 index 3 inet 129.159.115.21 netmask ff000000
broadcast 129.159.115.255
bge1: flags=1000843<mtu 1500 index 4 inet 129.159.115.22 netmask ffffffff00
broadcast 129.159.115.255 groupname mp0
bge1:1: flags=9040843<mtu 1500 index 4 inet 129.159.115.23 netmask ff000000
broadcast 129.159.115.255
```

ここで、両方のホスト上の物理ネットワークインタフェースは、bge0 および bge1 として表示されています。『IP Network Multipathing Administration Guide』で説明されているように、bge0:1 および bge1:1 として表示されているのはマルチパステストインタフェースです(そのため、ifconfig 出力ではこれらのインタフェースが「非推奨」としてマークされている)。

この環境で HADB を設定するには、各ホストから 1 つの物理インタフェースアドレスを選択します。この例では、ホスト 1 からは 129.159.115.10 を、ホスト 2 からは 129.159.115.20 を選択します。ホストあたり 1 つのデータベースノードを含むデータベースを作成するには、hadbm create に次の引数を使用します。

```
--host 129.159.115.10,129.159.115.20
```

ホストあたり 2 つのデータベースノードを含むデータベースを作成するには、次の引数を使用します。

```
--host 129.159.115.10,129.159.115.20,129.159.115.10,129.159.115.20
```

どちらの場合も、両方のホストの ma.server.mainternal.interfaces 変数を 129.159.115.0/24 に設定してください。

### 4.4.1 から 4.4.2 へのオンラインアップグレード

オンラインで 4.2 または 4.3 から 4.4 にアップグレードすることはできません。ただし、4.4 の将来のバージョンではオンラインアップグレードがサポートされます。4.4.1 から 4.4.2 にアップグレードするには、次の手順を実行します。

1. すべての HADB ホストに 4.4.2 をインストールします。パスは 4.4.1 とは別のパス、たとえば /opt/SUNWhadb/4.4.2-6 にします。
2. hadbm クライアントホストに新しいバージョンをインストールします。
3. HADB ホスト上で実行されているすべての管理エージェントを停止します。
4. 新しいバージョンのソフトウェアを使用して(ただし、設定ファイルは古いまま)、管理エージェントプロセスを起動します。残りの手順では、新しいバージョンの bin ディレクトリにある hadbm コマンドを使用してください。

5. 管理ドメインでパッケージを登録します。ここで、デフォルトのパッケージ名が V4.4 になるため、同じ名前を持つ既存のパッケージとの競合を避けるために別のパッケージ名が必要になる場合があります。

```
hadbm registerpackage --packagepath=/opt/SUNWhadb/4.4.2-6 V4.4.2
```

6. 新しいバージョンを使用してデータベースを再起動します。次のコマンドでは、ノードの順次再起動が実行されます。

```
hadbm set packagename=V4.4.2 database_name
```

7. データベースが「実行中」の状態にあり (hadbm status コマンドを使用して確認)、かつ正常に動作してクライアントトランザクションを処理していることを確認します。
8. すべてが正常に動作している場合は、あとで古いインストールを削除することができます。

古いパッケージの登録を解除する前に、古いパッケージへのすべての参照をマリポジトリから削除します。そうしないと、「使用中のパッケージ」のために hadbm unregisterpackage が失敗します。ダミーの再設定操作 (hadbm set connectiontrace= <same\_as\_previous\_value> など) によって、古いパッケージへのすべての参照が削除されます。ここで、古いパッケージの登録を解除します。

```
hadbm unregisterpackage [--hosts=<host_list>] <old_package_name>
```

HADB のインストール手順 (<http://clustra.norway.sun.com/intraweb/download/products/hadb/packages/pdf/4.4.2-6.pdf>) で説明されているように、ファイルシステムから古いインストールを削除します。

### 既知の SQL の制限事項

- テーブルに UNIQUE 二次インデックスを作成することはできません。
- 式 (DISTINCT 列) は、これが選択された唯一の式でないかぎり、収集式では許可されません。
- すべてのテーブルを、主キーを指定して作成する必要があります。つまり、主キーのないテーブルはサポートされていません。
- FULL OUTER JOIN はサポートされていません。
- テーブルサブクエリーである IN サブクエリーはサポートされていません。次に例を示します。

```
SELECT SNAME FROM S WHERE (S1#,S2#) IN (SELECT S1#,S2# FROM SP
WHERE P#='P2')
```

- NOT NULL と PRIMARY KEY 以外の制約はサポートされていません。
- リソースに新しい所有者を割り当てることができます。ただし、これを行う場合、現在の所有者に付与されている特権は新しい所有者に付与されません。

- 2つ以上の入れ子の NOT EXISTS サブクエリーで、各サブクエリーがクエリーの外側のレベルに (直接) 関連付けられていないものはサポートされていません。
- 列の特権はサポートされていません。
- 行の値コンストラクタは、VALUES 句でのみ許可されています。
- 行の値コンストラクタでは、サブクエリーは値式とは見なされません。
- 主キーを作成するとき、次のデータ型は使用できません。
  - REAL
  - FLOAT
  - DOUBLE PRECISION
  - DECIMAL
  - NUMERIC

## 高可用性ロードバランス

Application Server には、HTTP、IIOP、および JMS クライアント向けの負荷分散、HTTP セッションのフェイルオーバーのサポート、EJB クラスタリングおよびフェイルオーバーのサポート、高可用性 EJB タイマー、分散トランザクションリカバリ、アプリケーションのローリングアップグレードのサポート、および J2EE アプリケーションの一時的な状態を保存するための高可用性データベースが組み込まれています。

可用性により、クラスタ内の Application Server インスタンスのフェイルオーバー保護が可能になります。ある Application Server インスタンスがダウンすると、そのサーバーに割り当てられていたセッションを別の Application Server インスタンスが引き継ぎます。セッション情報は、HADB に格納されます。HADB は、HTTP セッションの持続性、ステートフルセッション Bean、およびシングルサインオン資格をサポートします。

## 互換性の問題

Sun Java System Application Server Enterprise Edition の次のメジャーリリースでは、次に示す非互換性が導入されます。

- HTTP サービスでは、パフォーマンス向上のために引き続き DNS キャッシュが使用されますが、DNS キャッシュの監視は使用できなくなります。
- HTTP ファイルキャッシュに対するサポートが変更されるため、設定と監視に変更が発生します。
- アクセスログローテーションのサフィックスの形式が、<http://java.sun.com/j2se/1.5.0/docs/api/java/text/SimpleDateFormat.html> に指定されている、日付と時刻オブジェクトでサポートされている形式に変更されます。このリリースでのデフォルト値、「%YYYY;%MM;%DD;-%hh;h:mm;m:ss;s」は引き続きサポートされますが、その他の種類はサポートされません。

- サポートされなくなる `domain.xml` の要素、属性、およびプロパティはすべて、サーバーログでは警告として、またアップグレードログファイルでは非推奨としてフラグが付けられます。
- `server.http-service.dns` ノードは、監視ビューでは使用できなくなります。
- `server.http-service.file-cache` ノードの属性の一部は、削除される可能性があります。その結果、これらのノードから削除された属性にアクセスしようとする `asadmin` 監視コマンドはすべて失敗します。

## Deploytool

Deploytool は使用できなくなります。これと等価な機能は NetBeans IDE にあります。詳細については、<http://www.netbeans.org/kb/> にある NetBeans 4.1 用の J2EE 1.4 Tutorial を参照してください。

## ベリファイア

- ベリファイアの GUI モード (`verifier -u` で起動される) は使用できなくなります。これと等価な機能は NetBeans IDE にあります。
- ベリファイアツールを使用する場合のアプリケーション確認のデフォルトモードは、「J2EE ルールの確認」から「J2EE ルールおよび Sun Application Server 設定ルールの確認」に変更されます。つまり、ベリファイアはデフォルトでは、アプリケーションが J2EE ルールを満たしているかどうか、および Sun Application Server 上で動作するように設定されているかどうかをテストします。ベリファイアコマンドには、J2EE ルールのみに関してアプリケーションをテストするためのコマンド行スイッチが用意されます。

## クラスローダの変更

現在のリリースでは、`domain.xml` (アプリケーションサーバーの設定ファイル) の `classpath-prefix`、`server-classpath`、および `classpath-suffix` 属性に追加された JAR およびディレクトリエントリは、JVM システムクラスパスでも使用できます。この動作に依存しているアプリケーションは、クラス `java.lang.ClassLoader` の次のメソッドを使用して、JVM システムクラスパスのクラスやその他のリソースにアクセスしている可能性があります。

- `getSystemClassLoader()`
- `getSystemResource()`
- `getSystemResourceAsStream()`
- `getSystemResources`

次のメジャーリリースでは、`classpath-prefix`、`server-classpath`、および `classpath-suffix` に追加された JAR およびディレクトリエントリは、JVM システムクラスパスでは使用できなくなります。アプリケーションが前述のいずれかのメソッドを使用している場合は、システムクラスパスでリソースが使用できることを前提としていない等価なメソッドを使用することを強くお勧めします。JVM システ

ムクラスパスに依存しない等価なメソッドは、`java.lang.ClassLoader` にあり、可能な場合は使用するようになっています。次に例を示します。

例 2-1 古いコード

```
java.net.URL url = ClassLoader.getSystemResource
("com/acme/tools/tools.properties");
```

例 2-2 推奨される変更

```
java.net.URL url = this.getClass().getClassLoader().getResource
("com/acme/tools/tools.properties");
```

コードを変更できない場合は、次のリリースで追加される、JVM システムクラスパスを設定するための新しい設定オプションの使用を選択することもできます。

Web サービスのセキュリティ設定

Web サービスのセキュリティは、`wss-client-config.xml` および `wss-server-config.xml` ファイルを使用して設定できます。これらの設定ファイルの内容や名前は変更される可能性が高いことに注意してください。ただし、等価な機能は引き続き使用できます。

J2EE のサポート

Sun Java System Application Server Enterprise Edition 8.2 は、J2EE 1.4 プラットフォームをサポートしています。J2EE 1.4 プラットフォームで利用できる拡張 API を、次の表に示します。

表 2-5 J2EE 1.4 プラットフォームで使用可能な API

| API                                 | 説明                                              |
|-------------------------------------|-------------------------------------------------|
| コンポーネント                             |                                                 |
| Application および Application Client  | XML スキーマによる標準配備記述子の実装                           |
| Enterprise JavaBeans (EJB) 2.1      | タイマーサービスと EJB Web サービスエンドポイント                   |
| Java Servlet 2.4                    | Web サービスエンドポイントのフィルタ                            |
| JavaServer Pages (JSP) 2.0 アーキテクチャー | 式言語とタグライブラリ                                     |
| J2EE Connector Architecture 1.5     | 着信リソースアダプタと Java Message Service (JMS) プラグイン可能性 |
| Web サービス                            |                                                 |

表 2-5 J2EE 1.4 プラットフォームで使用可能な API (続き)

|                                                             |                                                                                                     |
|-------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|
| Java Web Services Developer Pack 1.5                        | XML アプリケーション、Web サービス、および Web アプリケーションを構築、テスト、および配備するための統合ツールキット                                    |
| Java API for XML-based Remote Procedure Calls (JAX-RPC) 1.1 | WSDL と Java テクノロジーのマッピングと、Web サービスクライアントおよびエンドポイントの開発のサポート                                          |
| WS-I Basic Profile 1.0                                      | WSDL と SOAP による相互運用の有効化要素                                                                           |
| SOAP with attachment API for Java (SAAJ) 1.2                | SOAP ベースのメッセージングのための API で、ファイルの添付された SOAP メッセージの作成を促進します                                           |
| Java APIs for XML Registries (JAXR) 1.0                     | Universal Description Discovery and Integration (UDDI および ebXML) などの XML レジストリにアクセスするための統一された標準 API |
| その他                                                         |                                                                                                     |
| J2EE Deployment 1.1                                         | J2EE のコンポーネントおよびアプリケーションの配備を可能にする標準 API                                                             |
| J2EE Management 1.0                                         | J2EE プラットフォームを管理するための情報モデルの定義                                                                       |
| Java Management Extensions (JMX) 1.2                        | 標準的な管理 API                                                                                          |
| Java Authorization Contract for Containers (JACC) 1.0       | J2EE Application Server と 認証ポリシープロバイダとの間でのセキュリティー規約の定義                                              |
| Java API for XML Processing (JAXP) 1.2                      | XML ドキュメントを解析および変換するための API で、XML スキーマの処理もサポートします                                                   |
| JMS 1.1                                                     | J2EE アプリケーションコンポーネントがメッセージを作成、送信、受信、および読み取りできるためのメッセージング標準で、キューとトピックのための統一された API もサポートします          |
| JavaMail 1.3                                                | メールシステムをモデル化する抽象クラスの集合で、API の小規模なアップデートも含まれます                                                       |

## サポートされている別の Java バージョンへの切り換え

Sun Java System Application Server 8.2 には、配下の JVM として J2SE 5.0 以降が必要です。Java を別のバージョンに切り替える場合は、次の一般的な手順に従います。  
(Windows および Unix)

### ▼ サポートされている別の Java バージョンへ切り換えるには

- 1 必要に応じて、**Java SDK (JRE ではない)** をダウンロードしてインストールします。  
Java SDK は、<http://java.sun.com/j2se> からダウンロードできます。

**2 Application Server**を完全に停止します。

次のコマンド行が使えます。

```
as-install/bin/asadmin stop-domain
```

または、管理コンソールの GUI で次の手順に従います。

a. 「アプリケーションサーバー」ノードをクリックします。

b. 「インスタンスの停止」をクリックします。

**3** `install_dir/config/asenv.conf` ファイル (**Windows** では `asenv.bat`) を編集して、新しい **J2SE** のホームディレクトリをポイントするように `AS_JAVA` の値を変更します。

**4** `as-install/samples/common.properties` ファイルを編集して、**J2SE** の新しいホームディレクトリを参照するように `com.sun.aas.javaRoot...` で始まる行を変更します。

**5 Application Server**を再起動します。

```
as-install/bin/asadmin start-domain
```

## 高パフォーマンス

Application Server は、高パフォーマンスの EJB コンテナ、Web コンテナ、および Web サービスを組み込んでおり、Sun Java System Message Queue ソフトウェアによる並行メッセージデリバリーをサポートします。

## スケーラビリティ

Application Server は、サーバーインスタンスのクラスタ化と要求のロードバランスによる水平方向のスケーラビリティをサポートします。同時に、マルチプロセッサの大規模マシンをサポートする最高級の垂直方向スケーラビリティも実現します。統合メッセージブローカは、より高度なスケーラビリティと可用性のためにクラスタ化できます。HTTP クライアント、RMI/IIOP ベースのリッチクライアントアプリケーション、Web サービスクライアント、および JRM クライアントからのクライアントアクセスは、Application Server クラスタに負荷分散できます。

## JavaServer Faces 1.1 のサポート

Sun Java System Application Server Enterprise Edition 8.2 は、JavaServer Faces 1.1 テクノロジをサポートしています。JavaServer Faces テクノロジは、自身の状態、イベント、ハンドリング、および入力チェックを管理するユーザーインタフェースコンポーネントに対応する一連のサーバーサイド API から成ります。これらの API は、ページナビゲーションを規定し、国際化とアクセシビリティをサポートします。JSP のカスタムタグライブラリを使用して、カスタム UI コンポーネントを追加できます。

JavaServer Faces テクノロジで開発している間、開発チームの各メンバーは、処理の一部分にだけ集中することができます。単純なプログラミングモデルでは、これらの部分をつなぎ合わせることで、ずっと効率的でシンプルな開発サイクルが実現されます。



## 既知の問題点と制限事項

---

この章では、Sun Java System Application Server Enterprise Edition 8.2 ソフトウェアに関する既知の問題とそれに関連する回避策について説明します。問題の説明にプラットフォームが明記されていない場合、その問題はすべてのプラットフォームに当てはまります。この節は次の項目から構成されています。

- 41 ページの「管理」
- 47 ページの「Apache とロードバランサプラグイン」
- 49 ページの「アプリケーションクライアント」
- 49 ページの「付属の Sun JDBC ドライバ」
- 52 ページの「コネクタ」
- 53 ページの「マニュアル」
- 55 ページの「高可用性」
- 62 ページの「インストール」
- 66 ページの「J2EE Tutorial」
- 66 ページの「ライフサイクル管理」
- 67 ページの「ロギング」
- 67 ページの「メッセージキュー」
- 68 ページの「監視」
- 70 ページの「持続性」
- 70 ページの「PointBase」
- 71 ページの「サンプル」
- 74 ページの「セキュリティ」
- 75 ページの「アップグレードユーティリティ」
- 79 ページの「Web コンテナ」

### 管理

ここでは、管理上の既知の問題とその解決方法を示します。

domain1が存在しない場合、package-appclient スクリプトが動作しない。(ID 6171458)

#### 説明

デフォルトでは、asenv.conf によって参照される domain1 の AS\_ACC\_CONFIG 変数の値が \$INSTALL/lib/package-appclient.xml にハードコードされています。domain1 を削除して新たなドメインを作成した場合、AS\_ACC\_CONFIG 変数は新たなドメイン名で更新されません。その結果、package-appclient スクリプトの処理が失敗します。

#### 解決方法

次のいずれかの操作を行います。

- domain1 はそのままにしておき、その前後に別のドメインを作成します。
- domain1 を削除し、\$INSTALL/lib/package-appclient.xml の domain1 にハードコードされた値を新しいドメイン名で置き換えます。domain1 がない場合、新たなドメインが作成されるたびにこれを行う必要があります。

負荷分散プラグインをインストールすると、既存のプラグインが上書きされる。(ID 6172977)

#### 説明

7.1EE などからすでにロードバランサプラグインがインストールされている Application Server のインストールに対して負荷分散プラグインをインストールすると、プラグインを実行する新しいサーバーインスタンスを作成しても、メッセージの表示なしで既存のロードバランサが 8.2EE プラグインに置き換えられます。

プラグインファイルは、デフォルトで install\_dir/plugins/lbplugin ディレクトリの下にインストールされるため、1つの Application Server インストールで利用できるプラグインは1つのバージョンだけになります。コンソールインストーラはアンインストールが実行されていることを示すメッセージを表示しますが、このメッセージは見逃しやすいことに注意してください。

#### 解決方法

だれもがこの問題を経験するわけではありません。この問題が発生した場合は、古い Application Server インストールを削除して、アップグレードインストールではなく新規インストールを実行してください。

**JES2 AS7 と比べると、JES3 Application Server 8.2 の asadmin スクリプトにいくつかの変更がある (ID 6189433、6189436)**

Application Server 7.x と比べると、Application Server 8.2 の asadmin コマンドにはいくつかの変更が加えられています。たとえば、7.x では、サーバーインスタンスを起動するコマンドは次のとおりです。

```
asadmin start-instance
```

8.2 では、同等のコマンドは次のとおりです。

```
asadmin start-domain --user admin domain1
```

最新の asadmin コマンド構文の詳細については、次のマニュアルを参照してください。

- 『Sun Java System Application Server Enterprise Edition 8.2 管理ガイド』
- 『Sun Java System Application Server Enterprise Edition 8.2 Reference Manual』
- 『Sun Java System Application Server Enterprise Edition 8.2 アップグレードと移行』

### Application Server のデフォルトポートが変更される (ID 6198555)

#### 説明

JES2/Application Server 7.x から JES5/Application Server 8.2 にアップグレードすると、デフォルトのポートが変更されたために、互換性が失われたりエラーが発生したりすることがあります。

#### 解決方法

Application Server 8.2 で使用されるデフォルトポートの一覧については、このリリースノートの [20 ページ](#)の「[その他の要件](#)」を参照してください。

### バックアップしたドメインを別の名前で復元できない。 (ID 6196993)

#### 説明

同一の Application Server インストール上では、`backup-domain` コマンドと `restore-domain` コマンドを使用してドメインをミラーリングできません。これは、`asadmin restore-domain` コマンドにドメイン名を変更するオプションがあっても、元の名前ではなく、別の名前でドメインを復元できないからです。バックアップされたドメインの名前を正常に変更したように見えても、名前を変更されたドメインの起動は失敗します。ドメイン設定のエントリは変更されておらず、`startserv` および `stopserv` は元のドメイン名を使用してパスを設定するからです。

#### 解決方法

`restore-domain` で使用するドメイン名は、元の `backup-domain` コマンドで使ったドメイン名と同じである必要があります。Application Server 8.2 の `backup-domain` コマンドと `restore-domain` コマンドが動作するのは、同一マシンで同一ドメインをバックアップおよび復元する場合だけです。

JMX エージェントを追加した **Application Server** の起動はサポートされていない。(ID 6200011)

## 説明

J2SE 1.4.x、5.0、またはそれ以降のバージョンは、Application Server で設定できます。J2SE 5.0 プラットフォームの重要な特徴は、JMX エージェントを起動できることです。サーバーの起動時にシステムプロパティを明示的に設定すると、JMX エージェントがアクティブになります。

次に例を示します。

```
name="com.sun.management.jmxremote" value="true"
name="com.sun.management.jmxremote.port" value="9999"
name="com.sun.management.jmxremote.authenticate" value="false"
name="com.sun.management.jmxremote.ssl" value="false"
```

JMX プロパティを設定してサーバーを起動すると、Application Server VM 内で新しい jmx-connector サーバーが起動します。この場合は、望ましくない副作用の 1 つとして、管理機能が悪影響を受け、Application Server の管理 GUI や CLI で予期しない結果が発生することがあります。問題は、組み込みの jmx-connector サーバーと新たな jmx-connector サーバーとの間で衝突が発生することにあります。

## 解決方法

jconsole または何らかの JMX 互換クライアントを使用する場合には、Application Server とともに起動する標準の JMX コネクタサーバーを再利用することを検討してください。

サーバーの起動時に、次に示すような行が server.log に作成されます。ここで指定されている JMXServiceURL に接続し、資格を正常に指定した後、同様の管理および設定操作を実行することができます。次に例を示します。

```
[#|2004-11-24T17:49:08.203-0800|INFO|sun-appserver-ee8.1|javax.enterprise.
system.tools.admin|_ThreadID=10;|ADM1501: Here is the JMXServiceURL for the
JMXConnectorServer: [service:jmx:rmi:///jndi/rmi://hostname:8686/management/
rmi-jmx-connector]. This is where the remote administrative clients should
connect using the JSR 160 JMX Connectors.|#]
```

詳細については、『Sun Java System Application Server 8.2 管理ガイド』を参照してください。

UNIX で、**Application Server** の **start** および **stop** スクリプトに関する実行権に過度の制限がかけられている。(ID 6206176)

#### 説明

ユーザー「A」としてログインして `asadmin restore-domain` コマンドを実行すると、そのスクリプトのアクセス権は 744 (rwxr--r--) になります。そのあとでユーザー「B」としてドメインを起動または停止しようとする、たとえば「B」が `root` であっても、その試みは失敗します。ユーザー「A」についてだけスクリプトが実行可能だからです。

#### 解決方法

スクリプトのアクセス権を次のようにして変更します。

```
chmod 755 appserv/domains/domain-name/bin/*
```

どの **Web** サービスのエンドポイント URL を使用しても、ロードバランサの設定ファイルが作成されない。(ID 6236544、6275436)

#### 説明

Web サービスの URL をエクスポートする EJB モジュールを含むアプリケーションを使用してロードバランサを設定しても、作成された `loadbalancer.xml` ファイルに、その Web サービスのコンテキストルートが存在しません。

#### 解決方法

1. `loadbalancer.xml` ファイルを編集して、作成されなかった Web モジュールを次のように追加します。

```
<web-module context-root="context-root-name"
 disable-timeout-in-minutes="30" enabled="true"/>
```

2. `context-root-name` 値を、EJB として公開された Web サービスのコンテキストルート名に置き換えます。

設定内の **Java** ホームの設定が反映されない。(ID 6240672)

#### 説明

Application Server のドメインまたはサーバーが、関連付けられた設定の `java-config` 要素の `java-home` 属性によってポイントされる JDK を使用しません。

#### 解決方法

該当のサーバーインストール内のすべてのドメインに対して Application Server プロセスが使用する JDK は、`appserver-installation-dir/config/asenv.conf` ファイルによって決まります。使用される JDK は、このファイル内のプロパティ `AS_JAVA` に

よって決まります。これはインストール時に設定されます。インストール後に別のJDKをApplication Server プロセスが使用するようには、別のJDKをポイントするようにこの値を編集します。この変更によって、このインストール内のすべてのドメインが影響を受けることに注意してください。

---

注-asenv.conf ファイルを手動で変更する場合は有効性がチェックされないため、変更時に注意が必要です。AS\_JAVA の値を修正する場合は、製品のマニュアルでJDKのバージョンの最低限の要件を確認してください。

---

sun-appserv-admin を使用して **Application Server** を再起動すると、LoginException エラーが発生する。(ID 6288893)

#### 説明

この問題は%CONFIG\_HOME%の間違った値によって発生します。

#### 解決方法

1. 既存の名前を asant.bak に変更します。
2. <as\_install>/lib/install/templates/ee (SE/EE バージョンの場合) にある asant.template ファイルを <as\_install>/bin/ ディレクトリにコピーし、このファイルの名前を asant に変更します。
3. 新しくコピーされた <as\_install>/bin/asant スクリプトを編集して、%CONFIG\_HOME% トークンを <as\_install>/config に置き換えます。
4. 元の asant.bak ファイルに対して行なった手作業の変更がある場合は、それを新しい asant スクリプトに結合します。

**Application Server** のマニュアルに .asadmintruststore ファイルが記述されていない。(ID 6315957)

#### 説明

このファイルがサーバー管理者の home ディレクトリに存在しないと、そのサーバー上にホストされている特定のアプリケーションをアップグレードしたときに重大なバグが発生する場合があります。

#### 解決方法

- 可能であれば、そのサーバーをインストールしたユーザーが asadmin start-domain domain1 コマンドを実行してください。
- そのユーザーがこのコマンドを実行できない場合は、.asadmintruststore を、インストールしたユーザーの home ディレクトリから実行中のユーザーの home ディレクトリに移動またはコピーしてください。

- このファイルをインストールユーザーの home ディレクトリから実行中のユーザーの home ディレクトリに(コピーではなく)移動した場合は、アップグレードまたはインストールしたユーザーのホームディレクトリ (Java ES では、通常 root) に .asadmintruststore ファイルが存在しなくなるため、バグ 6309079、6310428、および 6312869 で説明されているようなアプリケーションのアップグレードに関する問題が発生する可能性があることに注意してください。

ドメインのマスターパスワードに特殊文字が含まれる場合、ドメインの起動が失敗する。(ID 6345947)

#### 説明

ドメインのマスターパスワードにパーセント文字(%)が含まれるときにドメインが起動しません。

#### 解決方法

ドメインのマスターパスワードにパーセント文字(%)を含めないようにしてください。これは、新しいドメインを作成するとき、および既存のドメインのマスターパスワードを変更するときに適用されます。

magnus.conf および obj.conf でのロードバランサの設定変更が上書きされる (ID 6394181)

#### 説明

セキュリティー保護された http-listener を作成し、lbplugin をインストールしたあとで、webserver\_instance\_dir/config の下の magnus.conf および obj.conf ファイルが変更され、lbplugin の内容が削除されます。

インストーラは、ロードバランサプラグインのインストールの一部として、Application Server の magnus.conf および obj.conf 設定ファイルを変更します。Application Server 管理コンソールにログインし、ロードバランサがインストールされたインスタンスのインスタンス設定を管理しようとする、Application Server は、設定の手動編集を検出したことを示す警告メッセージを表示します。この警告は、実際にはインストーラによって加えられた変更を示しています。

#### 解決方法

インストーラによって加えられた変更が上書きされていないことを確認します。

## Apache とロードバランサプラグイン

ここでは、Apache Web Server およびロードバランサプラグインに関する既知の問題と、それに関連する解決法を示します。

『高可用性 (HA) 管理ガイド』に、**Apache** で openssl を使用する場合は誤った手順が記載されている。(ID 6306784)

openssl のコンパイルと作成を行う場合は、次のコマンドを実行します。

```
cd openssl-0.9.7e
config
make
```

また、Apache 1.3 では、mod\_ssl ソースのディレクトリ名も、使用している Apache のリリースに応じて変わります。たとえば、Apache 1.3.33 の場合、この名前は mod\_ssl-2.8.22-1.3.33 になります。

『高可用性 (HA) 管理ガイド』に、**Apache 2.0** で証明書を使用するための手順が記載されていない。(ID 6307976)

Apache のセキュリティを実行するには、証明書を使用する必要があります。認証局から証明書を取得するための手順については、[modssl FAQ](http://www.modssl.org/docs/2.8/ssl_faq.html#ToC24) ([http://www.modssl.org/docs/2.8/ssl\\_faq.html#ToC24](http://www.modssl.org/docs/2.8/ssl_faq.html#ToC24)) にある証明書に関する情報を参照してください。

**Apache Web Server** をルートとして起動する必要がある。(ID 6308021)

Solaris では、Application Server がルートの下にインストールされている場合、Apache Web Server をルートとして起動する必要があります。Java Enterprise System は、ルートとしてインストールされます。Apache 2.0 の場合、ルートとして起動された後、Apache はユーザーが指定した別のユーザーに切り替えて動作します。そのユーザーは、/conf/httpd.conf ファイルで指定します。多くのシステムでは、ルートとして起動するには、httpd.conf ファイルを編集して正しいグループを指定する必要があります。次の行を置き換えます。

```
Group #-1
```

次の行に置き換えます。

```
Group nobody
```

ユーザーおよびグループの使用に関する詳細情報は、httpd.conf ファイルに記載されています。

**Solaris** で **Apache Web Server 2.0** とともに openssl を使用するための手順への追加。(ID 6308043)

Apache 2.0 とロードバランサプラグインをインストールした後、ssl.conf と ssl-std.conf を次のように編集します。

次の行を置き換えます。

```
<VirtualHost _default_:9191>
```

次の行に置き換えます。

```
<VirtualHost machine_name:9191>
```

ここで *machine\_name* はマシンの名前であり、9191 はセキュリティーポート番号です。

## アプリケーションクライアント

ここでは、アプリケーションクライアントに関する既知の問題とその解決方法を示します。

アプリケーションクライアントアーカイブのライブラリ JAR が MANIFEST ファイルを上書きする (ID 6193556)。

### 説明

クライアント JAR (この場合は reporter.jar) 内に最上位レベルの JAR ファイルがある場合、クライアント JAR を配備すると、その JAR の MANIFEST ファイルがクライアント JAR の MANIFEST ファイルを上書きします。

### 解決方法

現時点ではありません。

CGI-bin や SHTML 機能などの動的コンテンツ技術がサポートされていない。 (ID 6373043)

### 説明

CGI-bin や SHTML などの動的コンテンツ技術はサポートされなくなりました。

### 解決方法

代わりに JSP または Web サービスの技術を使用してください。

## 付属の Sun JDBC ドライバ

ここでは、Sun の JDBC ドライバに関する既知の問題とその解決方法を示します。

TRANSACTION\_SERIALIZABLE 遮断レベルを Microsoft SQL Server 向けの付属の Sun ドライバとともに使用するアプリケーションは、2つの並行トランザクションが実行されていて、その1つがロールバックされた場合、準備済みステートメントを使って更新するときにハングアップすることがある。 (ID 6165970)

希望の遮断レベルを接続に対して設定するには、対応する接続プールをその遮断レベルで作成する必要があります。接続プールの設定の詳細は、『[Sun Java System Application Server Enterprise Edition 8.2 管理ガイド](#)』を参照してください。

## PreparedStatement エラーが発生する。(ID 6170432)

### 説明 1

1 つのトランザクションで 3000 を超える PreparedStatement オブジェクトを生成する場合、DB2 では次のエラーが発生する可能性があります。

```
[sunm][DB2 JDBC Driver] No more available statements. Please recreate your package with a larger dynamicSections value.
```

### 解決法 1

次のプロパティを接続プール定義に追加して、ドライバが DB2 パッケージをより大きな動的セクション値に再バインドするようにします。

```
createDefaultPackage=true replacePackage=true dynamicSections=1000
```

接続プールの設定の詳細は、『[Sun Java System Application Server Enterprise Edition 8.2 管理ガイド](#)』を参照してください。

### 説明 2

前述の PreparedStatement エラーに関連して、次のエラーメッセージがスローされることがあります。

```
[sunm][DB2 JDBC Driver][DB2]Virtual storage or database resource is not available.
```

### 解決法 2

DB2 サーバー設定パラメータ `APPLHEAPSZ` の値を増やします。適度な値は 4096 です。

### 説明 3

遮断レベル `TRANSACTION_SERIALIZABLE`。アプリケーションが遮断レベル `TRANSACTION_SERIALIZABLE` を採用し、前述したパラメータの 1 つを使用している場合、そのアプリケーションは接続を取得するときにハングアップすることがあります。

### 解決法 3

希望の遮断レベルを接続に対して設定するには、対応する接続プールをその遮断レベルで作成する必要があります。手順については、『[Sun Java System Application Server Enterprise Edition 8.2 管理ガイド](#)』を参照してください。

## Sybase Adaptive Server 用の付属の Sun ドライバでの遮断レベルの設定にかかわる問題。(ID 6189199)

### 説明

- TRANSACTION\_SERIALIZABLE 遮断レベルを Sybase Adaptive Server 向けの付属の Sun ドライバとともに使用するアプリケーションは、2つの並行トランザクションが実行されていて、その1つがロールバックされた場合、準備されているステートメントを使用して更新するときにハングアップすることがあります。接続ロールバックは次のメッセージとともに失敗し、ロールバックされた接続はそれ以降は使用できません。

```
java.sql.SQLException:[sunm][Sybase JDBC Driver]Request cannot be submitted
due to wire contention
```

- Sybase Adaptive Server は TRANSACTION\_REPEATABLE\_READ 遮断レベルをサポートしません。ただし、DatabaseMetaData をクエリーすると、付属の Sun ドライバは、この遮断レベルがこのデータベースによってサポートされていると返答します。この遮断レベルを使用するアプリケーションは処理に失敗します。
- 付属の Sun ドライバを使用するアプリケーションは、TRANSACTION\_READ\_UNCOMMITTED 遮断レベルを設定できません。DataBaseMetaData に対する最初のアクセスの時点で、アプリケーションは次の例外をスローします。

```
java.sql.SQLException:[sunm][Sybase JDBC Driver][Sybase]The optimizer could
not find a unique index which it could use to perform an isolation level 0
scan on table 'sybsystemprocs.dbo.spt_server_info'.
```

### 解決方法

現時点ではありません。

**Solaris 10 および Enterprise Linux 3.0 で、Sun に付属している Oracle JDBC ドライバでは接続を作成できない。(ID 6247468)**

Sun JDBC Oracle データソース (com.sun.sql.jdbcx.oracle.OracleDataSource) を使用する場合は、JDBC 接続プールに次のプロパティを設定します。

```
<property name="serverType" value="dedicated"/>
```

このプロパティの値は、Oracle サーバーのリスナーの設定方法によって異なります。「共有」モードで設定した場合は、上の値を「dedicated」に変更する必要があります。

java.lang.SecurityException: **Sealing violation exception (ID 6554602)**

### 説明

JDBC 10.2 ドライバを使用して開始するとき、CLASSPATH に複数の JDBC jar ファイルが含まれる場合に `java.lang.SecurityException: Sealing violation exception` が発生する可能性があります。

Oracle が提供している詳細な説明については、次の Oracle マニュアル ID を参照してください。

Note:405446.1 Subject: JDBC Driver 10.2 Uses Sealed JAR files and May Cause SecurityException Sealing Violation

### 解決方法

(Oracle 推奨) CLASSPATH には JDBC ドライバ JAR ファイルが 1 つだけ含まれるようにしてください。

## コネクタ

この節では、J2EE のコネクタアーキテクチャに関する既知の問題とその解決方法を示します。

**DAS インスタンスを再起動したあと、`cascade` が `false` に設定されている場合にコネクタモジュールの配備取り消しが失敗する。(ID 6188343)**

### 説明

このシナリオでは、スタンドアロンまたは埋め込みのコネクタモジュールが DAS とコネクタ接続プールに配備され、その配備済みモジュール用にリソースが作成されます。DAS インスタンスを再起動したあと、`cascade` が `false` に設定されている場合にコネクタモジュールの配備取り消しが次の例外で失敗します。

```
[#
|2004-10-31T19:52:23.049-0800|INFO|sun-appserver-ee8.1|javax.enterprise.system
.core|_ThreadID=14;|CORE5023:Error while unloading application [foo]|#]
```

### 解決方法

DAS インスタンスを再起動したあとは、スタンドアロンまたは埋め込みのコネクタの配備を取り消すために、カスケード式配備取り消しを使用 (`cascade` オプションを `true` に設定) します。

## JMS create-jms-resource; CLI によってデフォルト値が正しく設定されない (ID 6294018)

### 説明

コマンド行から `asadmin create-jms-resource` コマンドで新しい JMS リソースを作成するときは最小プールサイズと最大プールサイズを指定できないため、`asadmin` コマンドがデフォルトのプールサイズ値 (最小が 8、最大が 32) を使用してリソースを作成すべきです。ただし、この場合には当てはまりません。代わりに、コマンド行からリソースを作成すると、デフォルトの最小プールサイズと最大プールサイズがそれぞれ 1 と 250 になります。

### 解決方法

コマンド行から JMS リソースを作成したあとで、管理コンソールを使用して最小プールサイズ値と最大プールサイズ値を変更します。

## マニュアル

ここでは、マニュアル上の既知の問題とその解決方法を示します。

### Javadoc に矛盾がある。(さまざまな ID)

いくつかの AMX インタフェースおよびメソッドについて、Javadoc が欠けているか間違っています。

- `NumConnAcquired` および `NumConnReleased` 統計情報の取得メソッドが `ConnectorConnectionPoolStats` および `AltJDBCConnectionPoolStats` から抜けている。これらの取得メソッドは、将来のリリースで `getNumConnAcquired()` および `getNumConnReleased()` として追加される予定。
- `EJBCacheStats` 内でメソッド `getPassivationSuccesses()`、`getExpiredSessionsRemoved()`、`getPassivationErrors()`、`getPassivations()` を呼び出すと、例外がスローされる。これは将来のリリースで解決される予定。
- サーバーを起動したあと、すべての AMX MBeans が登録されて利用できるようになるまでに数秒を要することがある。将来のリリースでは、AMX MBeans が完全にロードされたことを確認できるようになる予定。
- 定数 `XTypes.CONNNECTOR_CONNECTION_POOL_MONITOR` のスペルが間違っている ("NNN" の部分)。これは将来のリリースで訂正される予定。

付属の ANT によって `java.lang.NoClassDefFoundError` がスローされる。(ID 6265624)

### 説明

スレッド「main」で `java.lang.NoClassDefFoundError: org/apache/tools/ant/launch/Launcher` の例外がスローされます。

## 解決方法

付属の ANT を Application Server の外部で使用することはお勧めできません。

### ログオプションのマニュアルに誤りがある (ID 6463965)

『[Sun Java System Application Server Enterprise Edition 8.2 パフォーマンスチューニングガイド](#)』で、ログオプションについて次のように記述されていますが、これは誤りです。

The Administration GUI provides the following two logging options: (Administration GUI は次の2つのログオプションを提供します。)

- *Option 1* – Log stdout ( System.out.print) content to the event log (オプション 1 – stdout (System.out.print) コンテンツをイベントログに記録する)
- *Option 2* – Log stderr ( System.err.print) content to the event log (オプション 2 – stderr (System.err.print) コンテンツをイベントログに記録する)

これらのログオプションは、Application Server Enterprise Edition 8.2 では存在しなくなりました。

### Application Server 8.2 の HTTP ファイルキャッシュ機能に関して相反する情報がある (ID 6474799)

Application Server Enterprise Edition 8.2 のマニュアルでは、『[Sun Java System Application Server Enterprise Edition 8.2 パフォーマンスチューニングガイド](#)』の「HTTP ファイルキャッシュ」で HTTP ファイルキャッシュ機能について説明しています。しかし、この機能は Application Server Enterprise Edition 8.2 に含まれていませんでした。この機能は Application Server 9.0 で再導入されたことに注意してください。

### ラップされた接続からの物理接続の取得に関するマニュアルが正しくなくなった (ID 6486123)

ほかの欠陥 (6295215 など) により、『[Sun Java System Application Server Enterprise Edition 8.2 Developer's Guide](#)』の「Obtaining a Physical Connection from a Wrapped Connection」(第 11 章)、『[Sun Java System Application Server Enterprise Edition 8.2 Developer's Guide](#)』の第 11 章「Using the JDBC API for Database Access」に記載されているコードは正しくなくなりました。具体的には、次の行を

```
Connection drivercon = ds.getConnection(con);
```

次のように修正する必要があります。

```
Connection drivercon = ((com.sun.gjc.spi.DataSource)ds).getConnection(con);
```

## 高可用性

ここでは、高可用性データベース (HADB) に関する既知の問題とその解決方法を示します。

### ダブルネットワークでの HADB 設定。(ID なし)

2つのサブネット上にダブルネットワークで設定された HADB は、Solaris SPARC 上では正常に動作します。しかし、一部のハードウェアプラットフォームでのオペレーティングシステムまたはネットワークドライバの問題が原因で、Solaris x86 および Linux プラットフォームではダブルネットワークを適切に処理できない場合があります。これにより、HADB について次の問題が発生します。

- Linux では、メッセージ送信の際に HADB プロセスがブロックされることがある。これにより、HADB ノードが再起動し、ネットワークパーティションが発生する。
- Solaris x86 では、ネットワーク障害が発生した場合、もう一方のネットワークインタフェースへの切り替えを妨げる問題が発生することがある。この問題は常に発生するとは限らないため、ネットワークが1つしかないよりも2つあった方が安全である。この問題は、Solaris 10 で部分的に解決されている。
- トランキングがサポートされない。
- Microsoft Windows 2003 では、HADB はダブルネットワークをサポートしていない (ID 5103186)。

### HADB データベースの作成が失敗する。(ID なし)

新しいデータベースを作成すると、使用可能な共有メモリーセグメントが少なすぎるとい、次のエラーで失敗することがあります。

#### 説明

```
HADB-E-21054: System resource is unavailable: HADB-S-05512: Attaching shared memory segment with key "xxxxx" failed, OS status=24 OS error message: Too many open files.
```

#### 解決方法

共有メモリーが設定されており、その設定が機能していることを確認します。特に、Solaris 8 では、`/etc/system` ファイルを調べて、変数 `shmsys:shminfo_shmseg` の値がホストあたりのノード数の6倍以上になっていることを確認します。

共有メモリーセグメントがロックされ、ページアウトできない。(ID 5052548)

#### 説明

HADB 4.3-0.16 以降は、共有メモリーセグメントを作成してそれに接続するときに Intimate Shared Memory を使用するように設定されています (`SHM_SHARE_MMU` フラグを

使用)。このフラグを使用すると、必然的に共有メモリーセグメントが物理メモリーにロックされ、ページアウトできなくなります。このため、ローエンドマシンへのインストールでは、問題が発生する可能性が高くなっています。

したがって、Application Server 7.0 EE の使用時に開発者のマシンで 512M バイトのメモリーと十分なスワップ空間が利用でき、その後 7.1 EE 以降をインストールした場合、デフォルトの `clsetup` クラスタを設定するときに問題が発生します。このクラスタでは 2 つの HADB ノードが作成されて、それぞれの `devicesize` は 512 になり、両方のノードに必要な共有メモリーをサポートするのに十分な物理 RAM がないことになります。

### 解決方法

Application Server と HADB を共存させるときは、推奨されている容量のメモリーを使用するようにしてください。詳細については、[19 ページの「HADB の要件とサポートされているプラットフォーム」](#)を参照してください。

リソース (ディスクおよびメモリースペース) が利用可能かどうかを `hadbm set` がチェックしない。 (ID 5091280)

### 説明

`hadbm set` を使用してデバイスまたはバッファのサイズを増やす場合、管理システムは、データベースの作成やノードの追加の際にはリソースが利用可能かどうかをチェックしますが、デバイスまたはメインメモリーのバッファサイズを変更するときには利用可能なリソースが十分にあるかどうかをチェックしません。

### 解決方法

設定属性 `devicesize` または `buffersize` を増やす前に、すべてのホスト上にディスクおよびメモリーの空きスペースが十分にあることを確認してください。

`packagepath` の混在パスがサポートされない。 (ID 5091349)

### 説明

同一のソフトウェアパッケージを、同じ名前での別のホストの別の位置で登録することはできません。次に例を示します。

```
hadbm registerpackage test --packagepath=/var/install1 --hosts europa11
Package successfully registered.
hadbm registerpackage test --packagepath=/var/install2 --hosts europa12
hadbm:Error 22171: A software package has already been registered with
the package name test.
```

## 解決方法

HADB は、データベースクラスタ内のノードをまたがる混在パスをサポートしません。HADB サーバーのインストールディレクトリ (`---packagepath`) は、すべての参加ホストについて同一にしてください。

`createdomain` が失敗することがある。(ID 6173886、6253132)

## 説明

複数のネットワークインタフェースを備えたホスト上で管理エージェントを実行している場合に、すべてのネットワークインタフェースが同じサブネット上に存在しないと、`create domain` コマンドが失敗することがあります。

`hadbm:Error 22020: The management agents could not establish a domain, please check that the hosts can communicate with UDP multicast.`

管理エージェントは、特に設定されていないかぎり、UDP マルチキャスト用の「最初の」インタフェース(この「最初」は、`java.net.NetworkInterface.getNetworkInterfaces()` の結果によって定義される)を使用します。

## 解決方法

もっとも良い解決法は、使用するサブネットを管理エージェントに通知することです。たとえば、設定ファイル内の `ma.server.mainternal.interfaces` を `ma.server.mainternal.interfaces=10.11.100.0` に設定します。あるいは、サブネット間のルーターを、マルチキャストパケットをルーティングするように設定することもできます。このとき、管理エージェントはマルチキャストアドレス 228.8.8.8 を使用します。

管理エージェントの新しい設定を再試行する前に、管理エージェントリポジトリのクリーンアップが必要になる場合があります。ドメイン内のすべてのエージェントを停止し、リポジトリディレクトリ(管理エージェント設定ファイル内の `repository.dr.path` で識別される)内のすべてのファイルとディレクトリを削除します。この操作は、新しい設定ファイルを使用してエージェントを再起動する前に、すべてのホスト上で実行する必要があります。

HADB インスタンスの削除後にディレクトリをクリーンアップする必要がある。(ID 6190878)

## 説明

HADB インスタンスの削除に続いて `configure-ha-cluster` コマンドで新しいインスタンスを作成しようとする、失敗します。問題は、元の HADB インスタンスの古いディレクトリが `ha_install_dir/rep/*` と `ha_install_dir/config/hadb/instance_name` に残ることにあります。

## 解決方法

HADB インスタンスの削除後に、手動でこれらのディレクトリを削除するようにしてください。

**HADB の起動、停止、および再設定が失敗またはハングアップすることがある。(ID 6230792、6230415)**

## 説明

Solaris 10 Opteron では、`hadbm` コマンドを使用して HADB を起動、停止、または再設定すると、次のいずれかのエラーで失敗またはハングアップする場合があります。

```
hadbm:Error 22009: The command issued had no progress in the last
300 seconds.
HADB-E-21070: The operation did not complete within the time limit,
but has not been cancelled and may complete at a later time.
```

このエラーは、`clu_noman_srv` プロセスが使用するファイル (`nomandev`) への読み取り/書き込みに不整合があった場合に発生することがあります。この問題は、HADB 履歴ファイルで次のメッセージを検索することにより検出できます。

```
n:3 NSUP INF 2005-02-11 18:00:33.844 p:731 Child process noman3 733
does not respond.
n:3 NSUP INF 2005-02-11 18:00:33.844 p:731 Have not heard from it in
104.537454 sec.
n:3 NSUP INF 2005-02-11 18:00:33.844 p:731 Child process noman3 733
did not start.
```

## 解決方法

問題を手動で再現できていないため、次の回避策はまだ検証されていません。ただし、影響を受けるノードに対してこのコマンドを実行すれば、問題は解決されます。

```
hadbm restartnode --level=clear nodeno dbname
```

ノードのすべてのデバイスが再初期化されるわけではないことに注意してください。再初期化する前に、ノードの停止が必要になる場合があります。

管理エージェントが、例外「`IPV6_MULTICAST_IF failed`」で終了する。(ID 6232140)

## 説明

複数の NIC カードが実装された、Solaris 8 を実行しているホスト上で起動されている場合、IPv6 と IPv4 が有効になったカードが混在していると、管理エージェントが例外「`IPV6_MULTICAST_IF failed`」で終了することがあります。

## 解決方法

環境変数 `JAVA_OPTIONS` を `-Djava.net.preferIPv4Stack=true` に設定します。次に例を示します。

```
export JAVA_OPTIONS="-Djava.net.preferIPv4Stack=true"
```

あるいは、この問題が発生しない Solaris 9 以降を使用します。

`clu_trans_srv` を中断できない。(ID 6249685)

## 説明

Red Hat Enterprise Linux 3.0 の 64 ビットバージョンには、非同期入出力の実行中に `clu_trans_srv` プロセスを中断不可能なモードに陥らせるバグが存在します。つまり、`kill -9` が機能せず、オペレーティングシステムの再起動が必要になります。

## 解決方法

Red Hat Enterprise Linux 3.0 の 32 ビットバージョンを使用します。

`hadbm` が大文字を含むパスワードをサポートしていない。(ID 6262824)

## 説明

パスワードが `hadb` に格納されるときに、パスワード内の大文字は小文字に変換されます。

## 解決方法

大文字を含むパスワードは使用しないでください。

**HADB Version 4.4.2.5 から HADB Version 4.4.1.7 にダウングレードすると、`ma` が各種のエラーコードで失敗する。(ID 6265419)**

## 説明

以前の HADB バージョンにダウングレードすると、管理エージェントが各種のエラーコードで失敗する場合があります。

## 解決方法

HADB データベースのダウングレードは可能ですが、リポジトリオブジェクトが変更されている場合は管理エージェントをダウングレードできません。ダウングレードのあとも、最新の HADB バージョンの管理エージェントを使用し続ける必要があります。

インストールまたは削除を行なっても、symlinkが保持される。(ID 6271063)

#### 説明

HADB c パッケージ (Solaris: SUNWhadb, Linux: sun-hadb-c) バージョン <m.n.u-p> のインストールまたは削除に関しては、symlink /opt/SUNWhadb/<m> はいったん作成されると、その後は何も手を加えられません。そのため、切り離された symlink が存在することがあり得ます。

#### 解決方法

使用中の場合を除き、インストールの前またはアンインストールの後に symlink を削除します。

大域ゾーンとローカルゾーンの管理エージェントが干渉することがある。(ID 6273681)

#### 説明

Solaris 10 では、大域ゾーンで ma-initd スクリプトを使用して管理エージェントを停止すると、ローカルゾーンの管理エージェントも停止されます。

#### 解決方法

管理エージェントを大域ゾーンとローカルゾーンの両方にインストールしないでください。

セッションオブジェクトがタイムアウトし、**MA** で削除されたとき、hadbm/ma はより適切なエラーメッセージを出力するべきである。(ID 6275103)

#### 説明

場合によっては、サーバー上のリソース競合の問題によって管理クライアントが切断されることがあります。再接続時、「hadbm:Error 22184:A password is required to connect to the management agent」という紛らわしいエラーメッセージが返されることがあります。

#### 解決方法

サーバー上にリソースに関する問題があるかどうかを確認し、適切な処置 (たとえば、リソースの追加) を取ってから、操作を再試行します。

ルート以外のユーザーが **HADB** を管理できない。(ID 6275319)

#### 説明

Java Enterprise System を使用して (ルートとして) インストールすると、ルート以外のユーザーは HADB を管理できなくなります。

## 解決方法

HADB を管理するには、常にルートとしてログインします。

管理エージェントは特殊用途のインタフェースを使用するべきではない。(ID 6293912)

## 説明

0.0.0.0 のような IP アドレスを含む特殊用途のインタフェースを、管理エージェント内の HADB ノードが使用する有効なインタフェースとして登録するべきではありません。このようなインタフェースを登録すると、IP アドレスの代わりにホスト名を使用して `hadbm create` コマンドを発行するユーザーによってこのインタフェース上に HADB ノードが設定された場合に、問題が発生する場合があります。その場合、これらのノードは通信できなくなり、`create` コマンドはハングアップします。

## 解決方法

複数のインタフェースを備えたホスト上で `hadbm create` を使用する場合は、DDN 形式を使用して IP アドレスを常に明示的に指定します。

**Windows 上で再構築が失敗する。(ID 6291562)**

## 説明

Windows プラットフォームでは、特定の設定および負荷の下で、オペレーティングシステム内で多数の再構築の失敗が発生する場合があります。この問題は、20 を超えるノードが設定されている状況で、複数のテーブルスキャン (`select *`) を並列に実行している場合に発生しています。症状としては、トランザクションが頻繁に中止され、修復またはリカバリの完了に長い時間がかかるため、システムのさまざまな部分で頻繁なタイムアウトが発生していることが考えられます。

## 解決方法

この問題を修正するには、Windows レジストリ変数

`HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters` をデフォルトの 100 より大きい値に設定します。この値を 0x1000 (4096) に増やすことをお勧めします。詳細は、Microsoft サポートページの記事 [811003](http://support.microsoft.com/default.aspx?scid=kb;en-us;811003)

(<http://support.microsoft.com/default.aspx?scid=kb;en-us;811003>) を参照してください。

`hadbm start <db_name>` を実行すると、入力したパスワードの一部がマスクされずに表示される。(ID 6303581、6346059、6307497)

## 説明

マシンに負荷がかかっていると、マスキング機構が機能せず、入力したパスワードの一部の文字が表示されることがあります。これは軽度のセキュリティ上のリスクの原因となるので、パスワードは常にマスクすべきです。

### 解決方法

パスワードを独自のパスワードファイルに入れて (通常 Application Server 8.1 以降推奨されている方法)、`--adminpassword` または `--dbpasswordfile` オプションでこれらのファイルを参照します。

大域ゾーンにインストールされた **JES5 HADB** に疎ローカルゾーンからアクセスできない (ID 6460979)

### 説明

Application Server を Solaris 大域ゾーンの `/usr/SUNWappserver` にインストールすると、その Application Server インスタンスとともにインストールされた HADB コンポーネントは疎ローカルゾーンでは利用できません。

問題は、HADB が大域ゾーンの `/opt/SUNWhadb` にインストールされても、このディレクトリが疎ローカルゾーンから読み込み可能でないことにあります。残念ながら、JES5 の HADB バンドルは再配置できません。

### 解決方法

Application Server HADB コンポーネントは再配置できないため、HADB へのアクセス元にしたそれぞれの疎ローカルゾーンに HADB コンポーネントを別途インストールする必要があります。

## インストール

ここでは、インストール上の既知の問題とその解決方法を示します。

一部の **Linux** システムで、「完了」ボタンをクリックしたあとにインストールの終了でハングアップする。 (ID 5009728)

### 説明

この問題は、いくつかの Linux システム上で発生していました。これは Java Desktop System 2 でもっとも一般的に見られますが、Linux Red Hat ディストリビューションでも見られます。

インストーラの最後の画面で「完了」ボタンをクリックすると、インストーラは製品の「バージョン情報」ページまたは製品登録ページを表示するブラウザウィンドウの起動に失敗し、コマンドプロンプトに戻ることなくハングアップしたままになります。

### 解決方法

インストールプログラムを起動した端末ウィンドウで `Ctrl+C` を押すことにより、インストールプログラムを終了します。そのあとで、製品の「バージョン情報」ペー

ジまたは登録ページを表示するブラウザウィンドウが起動することがあります。ブラウザウィンドウが現れない場合には、ブラウザを起動してから次の URL を入力して「バージョン情報」ページを確認してください。

```
file://install_dir/docs-ee/about.html
```

製品を登録するインストールオプションを選択した場合には、「バージョン情報」ページ上の登録ページへのリンクをたどってください。

**Windows** では、インストール中に imq ディレクトリを作成する必要がある。(ID 6199697)

#### 説明

Windows では、Application Server Enterprise Edition をインストールした直後に、ディレクトリ `drive:\as\domains\domain1\imq` が存在しない旨のメッセージを出力して Message Queue ブローカが起動に失敗します。

`domain1` を起動してからブローカを起動した場合には、Application Server によってディレクトリが作成され、この問題は発生しません。

#### 解決方法

1. ブローカを作成する前に `var_home_dir_location` を作成します。次のようにします。

```
$imqbrokerd -varhome var_home_dir_location
```

次に例を示します。

```
$imqbrokerd -varhome D:\as\domains\domain1\imq
```

`compat-libstdc++` がいない場合に、**Application Server** を **RHLAS 3.0** および **RHLAS 4.0** に設定できない (ID 6396102)

#### 説明

Application Server Enterprise Edition 8.2 を Red Hat Linux Advanced Server (RHLAS) 3.0 または 4.0 システムにインストールする場合は、システムに `compat-libstdc++` ライブラリがインストール済みでないと失敗します。Application Server では、RHLAS システム上に `compat-libstdc++` ライブラリが必要ですが、これはデフォルトではインストールされません。この問題は RHLAS システムでのみ発生します。

#### 解決方法

Application Server ソフトウェアをインストールする前に、<http://rpm.pbone.net/index.php3/stat/4/idpl/843376/com/compat-libstdc++-7.3-2.96.118.i386.rpm.html> から `compat-libstdc++` RPM をダウンロードしてインストールします。

サーバーが 64 ビットモードで実行されていると、lbplugin(libpassthrough.so) を使用できない (ID 6480952)

## 説明

Application Server Enterprise Edition 8.2 を 64 ビットモードの Web Server 7.0 とともに実行しているときに、64 ビットバージョンのロードバランサプラグインを実行しようとする、次のエラーが発生して失敗します。

```
failure: CORE2253: Error running Init function load-modules: dlopen
of /export/home/mareks/opt/webserver7/plugins/lbplugin/bin/libpassthrough.so
failed (ld.so.1: webservd: fatal: /export/home/mareks/opt/webserver7/plugins/
lbplugin/bin/libpassthrough.so: wrong ELF class: ELFCLASS32)
failure: server initialization failed
```

問題は、Application Server Enterprise Edition 8.2 用の 64 ビットのロードバランサプラグインがなく、64 ビットの Web Server には 64 ビットのプラグインが必要であることにあります。

Web Server が 64 ビットモードで動作しているか 32 ビットモードで動作しているかは、次のコマンドを使用して判別できます。

```
wadm get-config-prop --user=admin --config=xxx --password-file=xxx platform
```

## 解決方法

Application Server Enterprise Edition 8.2 用の 64 ビットのロードバランサは計画されていません。この問題を回避するには、Web Server 7.0 の逆プロキシ機能を使用するか、32 ビットモードで動作するように Web Server 7.0 を設定してください。手順については、Web Server のマニュアルを参照してください。

「入力行が長すぎます」というメッセージが表示されて、asant deploy を実行できない (Windows 2000) (ID 6485174)

## 説明

Application Server 8.2 を Windows 2000 のデフォルトの場所にインストールすると、asant deploy の実行時に次のエラーが発生することがあります。

```
$ C:/Sun/JavaES5/appserver/bin/asant deploy
The input line is too long.
The syntax of the command is incorrect.
```

問題は、Windows 2000 のコマンド行を 1000 文字より長くすることができず、システム構成によってはデフォルトの ANT\_OPTS 環境が原因で asant deploy コマンド行が長くなることにあります。これは Windows 2000 だけの問題です。

## 解決方法

Windows 2000 で、Application Server を C:\JES5\_AS などの非常に短いディレクトリパスにインストールします。

**JES5 b12 の AS インストールで、common.properties 内に誤ったサーバーインスタンス AppServer1 がある (ID 6485254)**

## 説明

Windows 上の JES 5 b12 を使用していて、コンポーネント選択パネルの上位レベルで Application Server を選択した場合、ノードエージェントサブコンポーネントもデフォルトで選択されます。その後、インストールプロセスによって、ノードエージェントと、そのノードエージェントに属する AppServer1 というサーバーインスタンスが作成されます。これが正しい動作です。

ただし、ノードエージェントサブコンポーネントの選択を解除した場合でも、ドメインの common.properties ファイルでは AppServer1 インスタンスが作成されます。たとえば次のようになります。

```
domain.name=domain1
appserver.instance=AppServer1
```

これ以降、asant を使用してアプリケーションを配備しようとしても失敗します。

## 解決方法

common.propeties ファイルを編集して、appserver.instance=AppServer1 を appserver.instance=server で置換します。

**ラップされた接続からの物理接続の取得に関するマニュアルが正しくなくなった (ID 6486123)**

ほかの欠陥 (6295215 など) により、『[Sun Java System Application Server Enterprise Edition 8.2 Developer's Guide](#)』の「[Obtaining a Physical Connection from a Wrapped Connection](#)」(第 11 章)、『[Sun Java System Application Server Enterprise Edition 8.2 Developer's Guide](#)』の第 11 章「[Using the JDBC API for Database Access](#)」に記載されているコードは正しくなくなりました。具体的には、次の行を

```
Connection drivercon = ds.getConnection(con);
```

次のように修正する必要があります。

```
Connection drivercon = ((com.sun.gjc.spi.DataSource)ds).getConnection(con);
```

**Application Server が NFS をサポートしない。 (6396045)**

このバージョンのソフトウェアでは、Application Server でネットワークファイルシステム (NFS) をサポートしません。

## 解決方法

ありません。

## J2EE Tutorial

Sun Java System Application Server Enterprise Edition 8.2 で J2EE 1.4 Tutorial を実行するには、次の作業を実行します。

- 「About this Tutorial」の章の「About the Examples」で説明されているファイル例 `/common/build.properties` を編集する場合には、ポート 4848 を 4849 に変更します。
- `deploytool` を使用する場合、例を配備する前にサーバー `localhost:4849` を追加します。
- 管理コンソールを使用して何らかのリソースを作成する場合には、「ターゲット」タブを使用してサーバーをターゲットとして指定します。コマンド行または `asant` ターゲットを使用する場合、サーバーがデフォルトのターゲットになるため、特別な処置は必要ありません。

## ライフサイクル管理

ここでは、ライフサイクル管理に関する既知の問題とその解決方法を示します。

`ejb-timer-service` プロパティ `minimum-delivery-interval` を `9000` に設定したあとで、`ejb-timer-service` プロパティ `redelivery-interval-in-millis` を `7000` に設定しようとすると、`set` コマンドが次のエラーで失敗する。(ID 6193449)

### 説明

```
[echo] Doing admin task set
[exec] [Attribute(id=redelivery-interval-internal-in-millis) : Redelivery-Interval (7,000) should be greater than or equal to Minimum-delivery-interval-in-millis (9,000)]
[exec] CLI137 Command set failed.
```

- `minimum-delivery-interval` は、同一の周期タイマーの最小発生間隔。
- `redelivery-interval-in-millis` は、失敗した `ejbTimeout` のあとに再発生を試みるまでタイマーサービスが待機する時間。

これは、再発生間隔のプロパティを最小発生間隔のプロパティと関連付けるロジックが間違っていて、GUI または CLI を使用して再発生間隔よりも最小発生間隔が大きくなるような値を設定できないという問題です。

`minimum-delivery-interval-in-millis` を、`ejb-timer-service` プロパティの `redelivery-interval-in-millis` 以上の値に設定する必要があります。問題は、`redelivery-interval-in-millis` の値が `minimum-delivery-interval-in-millis` の値よりも大きいことを検証するための Application Server の処理にエラーがあることです。

## 解決方法

次のように、これらプロパティのデフォルト値を使用します。

```
minimum-delivery-interval(default)=7000
redelivery-interval-in-millis(default)=5000
```

これらデフォルト以外の値を指定するとエラーが発生します。

## ロギング

この節では、ログに関する既知の問題とその解決方法を示します。

`access.failure` のデバッグ文を設定すると、**Application Server** の起動時にハングする。(ID 6180095)

### 説明

JVM の `java.security.debug` オプションを設定すると、サーバーインスタンスの起動がデッドロックで動かなくなります。たとえば、`domain.xml` で次の設定を行うと、この問題が発生します。

```
<jvm-options>-Djava.security.debug=access,failure</jvm-options>
```

### 解決方法

現時点ではありません。このフラグは設定しないでください。

**JES3 Application Server** 以降ログの場所やインスタンスの場所が変更されている (ID 6189409)

Sun Java System 8.2 では、7.x と比べると、ログやサーバーインスタンスのデフォルトの場所が変更されています。

詳細は、『[Sun Java System Application Server Enterprise Edition 8.2 管理ガイド](#)』または『[Sun Java System Application Server Enterprise Edition 8.2 アップグレードと移行](#)』を参照してください。

## メッセージキュー

ここでは、Java メッセージキューに関する既知の問題とその解決方法を示します。

タイミングに依存する特定の場合に、**JMS** 再接続が正常に完了しない。(ID 6173308、6189645、6198481、6199510、6208728)

### 説明

タイミングに依存する場面での再接続の失敗は、さまざまな問題によって引き起こされます。

## 解決方法

これらの問題は、次の方法で回避できます。

- 関連するブローカを再起動する
- 関連する Application Server インスタンスを再起動する

非同期メッセージリスナーの動作が、appclient で **8.0** から **8.1 Update 2** に変更された。(ID 6198465)

## 説明

最新の変更により、非同期メッセージリスナーが app-client コンテナの唯一の稼働しているスレッドである場合、残っている appclient 仮想マシンはデーモンとして存在します。この動作は、ACC で非同期受信を実行する過去のアプリケーションの影響です。この問題は、JMS メッセージリスナーを設定してメインスレッドを終了するアプリケーションクライアントに影響します。

## 解決方法

メインスレッドを終了しないでください。メッセージリスナーがメインスレッドに通知するのを待ってから、メインスレッドを終了します。

# 監視

ここでは、監視上の既知の問題とその解決方法を示します。

コネクタサービスとコネクタ接続プールの監視レベルを変更できない。(ID 6089026)

## 説明

「監視レベル」設定ページを使用して、「コネクタサービス」または「コネクタ接続プール」の監視レベルを LOW または HIGH に変更して保存しても、ドメインの domain.xml ファイルのどちらの値も変更されません。しかし、「JMS サービス」の監視レベルを LOW または HIGH に変更して保存すると、「コネクタサービス」と「コネクタ接続プール」の値も同時に変更されます。この問題は、コマンド行から同等のコマンドを実行すると発生しません。

## 解決方法

監視レベルを変更するときは、「監視レベル」ページの JMS サービスコンポーネントだけを使用してください。

HTTP サービスの監視統計には有益な情報を提示しないため、無視する必要があるものがある。(ID 6174518)

HTTP サービスの一部の要素の監視統計を参照した場合、示される値のいくつかは現在の値に対応していないか、または常に 0 になっています。特に、次の HTTP サービス統計は Application Server に適用できる情報を表していないため、無視すべきです。

■

```

http-service
load1MinuteAverage
load5MinuteAverage
load15MinuteAverage
rateBytesTransmitted
rateBytesReceived

```

■ pwc-thread-pool (要素)

該当する監視名を持つ統計をすべて削除した場合でも、配備を取り消された EJB モジュールに対する監視 MBean が削除されない。(ID 6191092)

## 説明

次に例を示します。

```

EJBModuleMonitorMap().size() = 1 eventhough ejb module is
undeployed EJBModuleMonitor().getName() = sqe_ejb_s1_01

```

これは、EJB モジュールとアプリケーションの両方に当てはまります。MBean API 経由のプログラムを使用しても、asadmin list/get を使用しても、空の監視 MBean が残っています。

## 診断

```

asadmin list -m "server.applications" shows the following output:
server.applications.MEjbApp
server.applications.__ejb_container_timer_app
server.applications.adminapp
server.applications.admingui
server.applications.com_sun_web_ui
server.applications._export_install_nov-11_domains_domain1_applications
_j2ee-modules_sqe_ejb_s1_01

```

次のようにして統計を調べることができます。

```

bin/asadmin list -m "server.applications._export_install_nov-11_domains
_domain1_applications_j2ee-modules_sqe_ejb_s1_01"
server.applications._export_install_nov-11_domains_domain1_applications_
j2ee-modules_sqe_ejb_s1_01.SQEMessage
server.applications._export_install_nov-11_domains_domain1_applications_
j2ee-modules_sqe_ejb_s1_01.TheGreeter

```

いったん配備を取り消します。

```

_export_install_nov-11_domains_domain1_applications_j2ee-modules_sqe_
ejb_s1_01

```

ここで list コマンドを実行すると、まだアプリケーションが残っています。

```
asadmin list -m "server.applications"
server.applications.MEjbApp
server.applications.__ejb_container_timer_app
server.applications._export_install_nov-11_domains_domain1_applications_
j2ee-modules_sqe_ejb_s1_01
server.applications.adminapp
server.applications.adminui
server.applications.com_sun_web_ui
```

しかし、何の監視統計も含まれていません。

```
asadmin list -m "server.applications._export_install_nov-11_domains_
domain1_applications_j2ee-modules_sqe_ejb_s1_01"
Nothing to list at server.applications.-export-install-nov-11-domains-
domain1-applications-j2ee-modules-sqe-ejb-s1-01.
```

ある文字列で始まる有効な名前を取得するには、ワイルドカード文字(「\*」)を使用します。たとえば、server で始まるすべての監視可能エンティティの名前を一覧表示するには、list "server.\*" を使用します。

### 解決方法

これは無害です。何の問題もなくモジュールを再配備できます。ルート監視 MBean は削除されませんが、その内容は空です。

## 持続性

ここでは、Java Data Objects およびコンテナ管理による持続性に関する既知の問題とその解決方法を示します。

**JD076018:** 循環依存関係が原因で持続オブジェクトをフラッシュできない。(ID 6500961)

### 説明

この例外は、トランザクションで変更(または作成)されたインスタンス間に外部キー依存関係のチェーンがあるため、データベースで循環依存関係が生じた場合にスローされます。

### 解決方法

元の一連の操作を複数のトランザクションに分割します。

## PointBase

ここでは、PointBase に関する既知の問題とその解決方法を示します。

アプリケーションの接続プールに対して遮断レベルを設定すると、**PointBase**で例外が発生する。(ID 6184797)

#### 説明

PointBase データベースインストールを指している JDBC 接続プールについて、`transaction-isolation-level` プール属性をデフォルト (`Connection.TRANSACTION_READ_COMMITTED`) 以外の任意の値に設定すると、例外が発生します。ただし、その他のデータベースを指すプールについてデフォルト以外の値にこのパラメータを設定しても、例外はスローされません。

#### 解決方法

PointBase データベースを指す JDBC 接続プールについては、`transaction-isolation-level` を設定しないでください。

ネットワークサーバードライバと組み込みドライバを一緒に使用すると、**PointBase** が例外をスローする。(ID 6204925)

#### 説明

ネットワークサーバードライバと組み込みドライバを同時に使用すると、PointBase が例外をスローすることがあります。

#### 解決方法

組み込みドライバとネットワークサーバードライバの両方ではなく、どちらか一方だけを使用してください。

デフォルトの**PointBase** データベースが上書きされるというアップグレードの問題がある。(ID 6264969、6275448)

#### 説明

Application Server Enterprise Edition 8.2 にアップグレードすると、アップデートリリースパッチによって Pointbase デフォルトデータベースが上書きされます。

#### 解決方法

アップグレードの前に存在していたスキーマまたはデータを、すべて再作成または再入力します。テーブル生成オプションを使用して CMP Beans を含むアプリケーションを配備した場合は、テーブルを再作成するために、アプリケーションの配備の取消または再配備を行う必要があります。

## サンプル

ここでは、Application Server 8.2 製品に付属するサンプルコードに関する既知の問題とその解決方法を示します。

**setup-one-machine-cluster**が、**Windows**ではハングアップするが、**Solaris**では動作する。mqfailoverをCtrl+Cでキャンセルし、再実行する必要がある。(ID 6195092)

## 説明

`install_dir\samples\ee-samples\failover\apps\mqfailover\docs\index.html`を参照してから、次のコマンドを実行します。

### ■ コンソール 1

```
cd install_dir\samples\ee-samples asant start-mq-master-broker1
```

### ■ コンソール 2

```
cd install_dir\samples\ee-samples asant start-mq-cluster-broker1
```

### ■ コンソール 3

```
cd install_dir\samples\ee-samples asant start-mq-cluster-broker2
```

### ■ コンソール 4

```
cd install_dir\samples\ee-samples asadmin start-domain domain1
```

別の Enterprise Edition サンプル用に `asant setup-one-machine-cluster-without-ha` または `asant setup-one-machine-cluster-with-ha` を実行済みであれば `asant configure-mq` を、それ以外の場合には `asant setup-one-machine-cluster-and-configure-mq` を実行します。この場合、次に示すように、コマンドが正常に実行されたように見えます。

```
start_nodeagent: [echo] Start the node agent cluster1-nodeagent
[exec] Command start-node-agent executed successfully.
```

しかし、このあとシステムはハングアップします。

## 解決方法

現時点ではありません。この問題は、Windows でこの ant ターゲットを使用するすべての Enterprise Edition サンプルに同様に影響します。回避策は、ハングアップしたプロセスを Ctrl+C でキャンセルしてから再実行することです。

**MQ** フェイルオーバーのサンプルアプリケーションを実行する前に、`asadmin deploy` 命令の後で **JMS** リソースを作成する必要があることが、マニュアルに明記されていない。(ID 6198003)

## 説明

次のエラーがスローされます。

```

/opt/SUNWappserver/domains/domain1/config/sun-acc.xml -name
MQFailoverTestClient -textauth -user j2ee -password j2ee
Nov 18, 2004 10:50:17 PM com.sun.enterprise.naming.NamingManagerImpl
bindObjects
SEVERE: NAM0006: JMS Destination object not found: jms/durable/TopicA
Nov 18, 2004 10:50:18 PM com.sun.enterprise.naming.NamingManagerImpl
bindObjects
SEVERE: javax.naming.NameNotFoundException
javax.naming.NameNotFoundException

```

asadmin deploy コマンドを使用して手動配備を行う場合に JMS リソースを手動で作成する必要があること、そして、サンプルアプリケーションを配備するために用意されている ant ターゲットを使用する必要があることが、マニュアルに明記されていません。

### 解決方法

build.xml スクリプト用に asant deploy ターゲットを使用します。これにより、アプリケーションを実行するために必要とされる JMS リソースが作成されます。

**Linux で、webservices/security サンプルでの証明書の作成中に実行時エラーが表示される。(ID 6198239)**

### 説明

Linux で *install\_dir/samples/webservices/security* のサンプル (basicSSL) を配備するときに、証明書が作成されず、次のようなエラーがスローされます。

```

generate_certs: [echo] ***Exporting certificate from NSS database
[exec] Result: 1 [echo] ***Generating Java Keystore from generated
certificate [exec] keytool error: java.lang.Exception: Input not an
X.509 certificate [exec] Result: 1 [echo] ***Generating Java trust
store from generated certificate [exec] keytool error: java.lang.
Exception: Input not an X.509 certificate [exec] Result: 1
.
.
.
generate_certs: [echo] ***Exporting server certificate from NSS database to
a PKCS12 certificate file [exec] /opt/sun/appserver/lib/pk12util: /usr/lib/
libnss3.so: version 'NSS_3.9' not found (required by /opt/sun/appserver/lib/
pk12util) [exec] /opt/sun/appserver/lib/pk12util: /usr/lib/libnss3.so:
version 'NSS_3.6' not found (required by /opt/sun/appserver/lib/pk12util)
[exec] /opt/sun/appserver/lib/pk12util: /usr/lib/libnss3.so: version
'NSS_3.7' not found (required by /opt/sun/appserver/lib/pk12util) [exec]
Result: 1

```

問題は、Linux での NSS ライブラリの場所が Solaris での場所と異なることにあります。Linux 上に配備する場合、LD\_LIBRARY\_PATH が適切な NSS ライブラリを指している

ることを確認する必要があります。LD\_LIBRARY\_PATHを環境に設定するか、install\_dir/bin/asant シェルラッパースクリプトに設定します。

### 解決方法

次のいずれかの操作を行います。

- LD\_LIBRARY\_PATH=/opt/sun/private/lib を設定します。
- 次の行を install\_dir/bin/asant スクリプトに追加します。

```
LD_LIBRARY_PATH=$AS_NSS:$LD_LIBRARY_PATH;export LD_LIBRARY_PATH
```

**8.0 Platform Edition から 8.2 Enterprise Edition へのアップグレード後にサンプルドキュメントが見つからない**

### 説明

Application Server Platform Edition 8.0 から Application Server Enterprise Edition 8.2 にアップグレードしたあとで、サンプルページにアクセスしようとすると、HTTP 404 の「ファイルが見つかりません」というエラーが表示されることがあります。

### 解決方法

サンプルドキュメントを 8.0 ドメインから 8.2 ドメインにコピーしてください。

サンプルを疎ローカルゾーンで実行すると、実行時に失敗する。(ID 6460970)

### 説明

Application Server Enterprise Edition 8.2 を Solaris 大域ゾーンにインストールしたあとに続けて Application Server ドメインを疎ローカルゾーンにインストールした場合、疎ゾーンのドメインのファイルアクセス権が配備プロセス中に十分開かれていないと、サンプルアプリケーションを実行するときに問題が発生することがあります。

### 解決方法

配備プロセス中に、Application Server がクライアント JAR ファイル xmsClient.jar を検出してサンプルの場所 (/usr/SUNWappserver/appserver/samples/webservices/security/ejb/apps/xms/xmsClient.jar) にコピーできるようにしてください。通常、これはサンプルハーネスによって自動的に実行されますが、xmsClient.jar に対するアクセス権が開かれていないと失敗します。

## セキュリティー

ここでは、Application Server と Web アプリケーションのセキュリティーおよび証明書に関する既知の問題とその解決方法を示します。

**Enterprise Edition** 上で **J2SE 5.0** を使用して **WebServiceSecurity** アプリケーションを実行できない。(ID 6183318)

#### 説明

**WebServiceSecurity** アプリケーションは、次の理由から、**J2SE 5.0** では実行できません。

- **J2SE 5.0** の **PKCS11** は、**UNWRAP** モードをサポートしない
- **J2SE 5.0** の **PKCS11** は、**PKCS11** による **RSA/ECB/OAEPWithSHA1AndMGF1Padding** をサポートしない

**J2SE** チームは、このバグのために「**CR 6190389: Add support for RSA-PKCS1 and RSA-OAEP wrap/unwrap mechanisms**」をファイルしています。

#### 解決方法

**J2SE 1.4.2** で別の **JCE** プロバイダ (デフォルトで含まれているもの以外) を使用します。この構成では、ハードウェアアクセラレータはサポートされません。

**SSL** 終了が機能しない。(ID 6269102)

#### 説明

ロードバランサ (ハードウェア) を **SSL** 終了用に設定すると、リダイレクト中に **Application Server** がプロトコルを **https** から **http** に変更します。

#### 解決方法

ハードウェアロードバランサと **Application Server** の間にソフトウェアロードバランサを追加します。

## アップグレードユーティリティ

この節では、アップグレードユーティリティに関する既知の問題とその解決方法を示します。

**Application Server Enterprise Edition 8** から **Application Server Enterprise Edition 8.2** へのアップグレードで、*install\_dir*/domains ディレクトリ以外のカスタムパスに作成されたドメインが直接アップグレードされない。(ID 6165528)

#### 説明

アップグレードユーティリティを実行しているときに、*install\_dir* をソースインストールディレクトリとして指定すると、そのアップグレードプロセスは、*install\_dir*/domains ディレクトリの下に作成されたドメインだけをアップグレードします。その他の場所に作成されたドメインはアップグレードされません。

## 解決方法

アップグレードプロセスを起動する前に、すべてのドメインディレクトリを、それぞれの場所から *install\_dir* /domains ディレクトリに移動します。

一部の **Linux** システムで「アップグレードウィザードを起動」ボタンをクリックしたあとに、「同じ場所にアップグレード」を実行するインストーラがアップグレードツールの起動に失敗する。(6207337)

## 説明

この問題は複数の **Linux** システムで発生しています。Java Desktop System 2 でもっとも一般的ですが、RedHat ディストリビューションでも発生しています。

インストーラの最後の画面で「アップグレードツールの起動」ボタンをクリックすると、インストーラはアップグレード処理を完了するためのアップグレードツールの起動に失敗し、コマンドプロンプトに戻ることなくハングアップしたままになります。

## 解決方法

この問題は、コマンド行インストールモードを使って代替アップグレードを実行している場合には発生しません。

## ▼ コマンド行インストールモードを使用するには

- 1 GUI モードで代替アップグレードを実行してこの問題が発生した場合には、インストールプログラムを起動した端末ウィンドウで **Ctrl+C** を押すことにより、そのインストールプログラムを終了します。

- 2 その端末ウィンドウから次のコマンドを使ってアップグレードツールを起動します。

```
install_dir/bin/asupgrade --source install_dir/domains --target install_dir
--adminuser adminuser--adminpassword adminpassword --masterpassword changeit
```

*adminuser* および *adminpassword* は、アップグレード中のインストールで使用されている値に一致する必要があります。

- 3 アップグレードツールがアップグレードプロセスを完了したあとは、ブラウザを起動して次の **URL** を入力することにより、「バージョン情報」ページを参照できます。

```
file:///install_dir/docs/about.html
```

製品を登録するインストールオプションを選択した場合には、「バージョン情報」ページ上の登録ページへのリンクをたどってください。

**8.0 Platform Edition (PE) から 8.1 Enterprise Edition (EE) UR2 へのアップグレード中およびその後、自己署名付き証明書が信頼されない。(ID 6296105)**

アップグレード後、ターゲットの `domain.xml` から次のエントリを削除し、サーバーを再起動します。

```
<jvm-options>-Djavax.net.ssl.keyStore=${com.sun.aas.instanceRoot}
/config/keystore.jks</jvm-options>-
<jvm-options>Djavax.net.ssl.trustStore=${com.sun.aas.instanceRoot}
/config/cacerts.jks</jvm-options>
```

**Application Server の JES2 から JES5 へのアップグレード後にポートの衝突が発生する**

#### 説明

Application Server 7.x から 8.2 に更新すると、古いインストールと新しいインストールの間でポートの衝突(おそらく、デフォルトポートの **8080** と **8181**)が発生することがあります。

#### 解決方法

ポートの衝突を解決するには、Application Server 8.2 で使用するポートを変更してください。

**サンプルスクリプトで使用する Derby データベースが間違った場所に作成される。(ID 6377804)**

#### 説明

このバグには、次の 2 つの面があります。

1. Derby データベースを使用するサンプルアプリケーション設定スクリプトを実行すると、Derby データベースが現在のディレクトリまたは `<install_root>/bin` の下に作成されます。
2. サンプル build Ant スクリプトによって、管理パスワードファイルを保存する `password.txt` ファイルが現在のディレクトリの下に作成されますが、このディレクトリはルート以外の疎ゾーンシナリオでは書き込み可能ではありません。

#### 解決方法

1. Derby データベースの場所 – `start-database` コマンドで `--dbhome` オプションを使用して、`--dbhome` に指定された値の場所にデータベースを作成します。たとえば、次の例は、`start-database` 用の `asadmin` コマンドの構文です。

```
start-database [--dbhost 0.0.0.0] [--dbport 1527] [--dbhome db_directory] [--echo=false]
[--verbose=false]
```

2. password.txt ファイルの場所 – 設計上、サンプルディレクトリは書き込み可能であると想定されています。これは、すべてのビルドコマンドにそのディレクトリでの password.txt ファイルの作成が含まれているためです。書き込み可能な場所にサンプルの作業用コピーをインストールするようにしてください。

**8.0UR1PE から 8.2EE へのアップグレード中に LoginException が発生し、アップグレードプロセスが異常終了する (ID 6445419)**

#### 説明

この問題は、デフォルト以外の管理資格を使用してアップグレードインストールを実行すると発生します。

#### 解決方法

ファイルベースのインストーラを使用して 8.xPE から 8.2EE への並列アップグレードを実行するとき、新しい Application Server 用には次の管理資格を使用してください。

- 管理ユーザー: admin
- 管理パスワード: adminadmin
- マスターパスワード: changeit

アップグレードの実行後に、必要に応じてこれらのパスワードを変更できます。

**アップグレードツールが「ソースインストールディレクトリ」フィールドの既存の無効なディレクトリ入力の検出に失敗する (ID 6460122)**

#### 説明

アップグレードツールが「ソースインストールディレクトリ」フィールドの既存の無効なディレクトリ入力の検出に失敗するので、ディレクトリ設定は正しいという印象を与えます。

想定では、「ソースインストールディレクトリ」に不正なパスを入力すると「無効なディレクトリ」というメッセージがポップアップ表示されるべきです。「ソースインストールディレクトリ」に /opt/SUNWappserverEE81UR2/ と入力すると、無効なディレクトリというメッセージが正しくポップアップ表示されます。しかし、/opt/SUNWappserverEE81UR2/domains と入力すると、パスが無効でも、アップグレードツールは警告を表示せずにアップグレードプロセスを続けます。この問題は、入力する値によって動作が異なるという点を除けば、ID 6440710 に似ています。

#### 解決方法

Application Server 7 または 8.x から Application Server 8.2 にアップグレードするとき、マニュアルで推奨されている値でソースディレクトリをまずシードする必要があります。上書きアップグレードの場合はドメインルート、並列アップグレードの場合はドメインディレクトリです。

セミコロン (;) 文字の管理ユーザー名/管理パスワードを無効にすべきである (ID 6473341)

#### 説明

Application Server Enterprise Edition 8.2 インストールでは、管理ユーザー名に特殊文字は使用できません。特殊文字を使用すると、ドメインの作成は失敗します。ただし、管理パスワードには特殊文字を使用できます。

#### 解決方法

Application Server 7 から Application Server 8.2 にアップグレードするときに、管理ユーザー名に特殊文字が含まれていないことを確認してください。

## Web コンテナ

ここでは、Web コンテナに関する既知の問題とその解決方法を示します。

**Windows** で、`--precompilejsp=true` を使用してアプリケーションを配備すると、そのアプリケーションの **JAR** ファイルがロックされ、その後の配備取り消しや再配備が失敗することがある。 (ID 5004315)

#### 説明

Microsoft Windows にアプリケーションを配備するときに JSP のプリコンパイルを要求すると、それ以降、そのアプリケーションの配備取り消しや、そのアプリケーション (または同一モジュール ID を持つ任意のアプリケーション) の再配備を試みても、予期したとおりに動作しません。この問題は、JSP のプリコンパイル処理でアプリケーションの JAR ファイルが開かれたまま閉じられないため、Microsoft Windows がこれらのファイルを配備取り消しで削除することや、これらのファイルを再配備で上書きすることを許可しないことにあります。

配備取り消しは、Application Server からアプリケーションが論理的に削除されるという点では成功します。また、`asadmin` ユーティリティからエラーメッセージは返されませんが、そのアプリケーションのディレクトリとロックされた `jar` ファイルはサーバー上に残っています。サーバーのログファイルには、ファイルとアプリケーションディレクトリの削除に失敗した旨のメッセージが出力されます。

配備取り消し後のアプリケーションの再配備が失敗するのは、既存のファイルとディレクトリをサーバーが削除しようとして失敗するからです。これは、最初に配備されたアプリケーションと同じモジュール ID を持つアプリケーションを配備しようとしたときにも発生します。アプリケーションのファイルを保持するディレクトリの名前を、サーバーはモジュール ID から決定するからです。

同様の理由から、配備取り消しをせずにアプリケーションを再配備しようすると失敗します。

## 診断

アプリケーションを再配備しようとする、または、配備取り消しを行ってから配備しようすると、asadmin ユーティリティは次のようなエラーを返します。

```
An exception occurred while running the command. The exception
message is: CLI171 Command deploy failed : Deploying application in
domain failed; Cannot deploy. Module directory is locked and can't
be deleted.
```

## 解決方法

アプリケーションを配備するときに `--precompilejsp=false` (デフォルトの設定) を指定すると、この問題は発生しません。そのアプリケーションを最初に使用するとき JSP コンパイルが起動されるため、最初の要求に対する応答時間は、その後の要求に比べて長くなります。

また、プリコンパイルを行う場合には、そのアプリケーションを配備取り消しまたは再配備する前に、サーバーを終了して再起動する必要があります。シャットダウンすると、ロックされている JAR ファイルが解放されるため、再起動後の配備取り消しや再配備が成功します。

空の `<load-on-startup>` 要素を持つ **Servlet 2.4** ベースの `web.xml` を含んだ **WAR** ファイルを配備できない。 (ID 6172006)

## 説明

`web.xml` のオプションの `load-on-startup` 要素は、サーブレットを宣言する Web アプリケーションの起動の一環として、そのサーブレットをロードおよび初期化すべきことを示します。

この要素のオプションの内容は、Web アプリケーションのその他のサーブレットとの関係で、そのサーブレットをロードおよび初期化する順序を示す整数です。空の `<load-on-startup>` は、そのサーブレットを含む Web アプリケーションの起動時にそのサーブレットがロードおよび初期化される場合、その順序は意味を持たないことを表します。

`web.xml` の Servlet 2.4 スキーマでは、空の `<load-on-startup>` はサポートされなくなりました。つまり、Servlet 2.4 ベースの `web.xml` を使用する場合は整数値を指定する必要があります。 `<load-on-startup/>` の場合と同様に、空の `<load-on-startup>` を指定すると、`web.xml` が `web.xml` の Servlet 2.4 スキーマに対する妥当性検証に失敗するため、Web アプリケーションの配備も失敗します。

下位互換性の問題もあります。空の `<load-on-startup>` は、Servlet 2.3 ベースの `web.xml` では有効です。

## 解決方法

Servlet 2.4 ベースの web.xml を使用する場合は、`<load-on-startup>0</load-on-startup>` を指定して、サーブレットの読み込み順序が問題にならないことを示します。

リソースに制約のあるサーバー上で JSP ページをコンパイルできない。(ID 6184122)

## 説明

JSP ページにアクセスしてもコンパイルに失敗し、サーバーログには「Unable to execute command」というエラーメッセージと次のスタックトレースが記録されます。

```
at org.apache.tools.ant.taskdefs.Execute$Java13CommandLauncher.
exec(Execute.java:655) at org.apache.tools.ant.taskdefs.Execute.
launch(Execute.java:416)
at org.apache.tools.ant.taskdefs.Execute.execute(Execute.java:427)
at org.apache.tools.ant.taskdefs.compilers.DefaultCompilerAdapter.
executeExternalCompile(DefaultCompilerAdapter.java:448)
at org.apache.tools.ant.taskdefs.compilers.JavacExternal.execute
(JavacExternal.java:81)
at org.apache.tools.ant.taskdefs.Javac.compile(Javac.java:842)
at org.apache.tools.ant.taskdefs.Javac.execute(Javac.java:682)
at org.apache.jasper.compiler.Compiler.generateClass(Compiler.java:396)
```

## 解決方法

JSP のコンパイルスイッチを「fork」から「false」に設定します。

これは、次のいずれかの方法で行えます。

- グローバルに行うには、次のように、`${S1AS_HOME}/domains/domain1/config/default-web.xml` 内の JspServlet の fork init パラメータを false に設定します。

```
<servlet> <servlet-name>jsp</servlet-name>
<servlet-class>org.apache.jasper.servlet.JspServlet</servlet-class>
.... <init-param>
<param-name>fork</param-name> <param-value>>false</param-value>
</init-param> </servlet>
```

- Web アプリケーションごとに、`sun-web.xml` の JSP 設定プロパティ fork を false に設定します。次のようにします。

```
<sun-web-app> <jsp-config> <property name="fork" value="false" />
</jsp-config> </sun-web-app>
```

これらのいずれかを設定することにより、ant が javac コンパイルのための新規プロセスを生成することが防止されます。

**Application Server** で、auth-passthrough **Web Server 6.1** アドオンがサポートされない。  
(ID 6188932)

## 説明

Sun Java System Application Server Enterprise Edition 8.2 では、Sun Java System Application Server Enterprise Edition 7.1 で使用できる auth-passthrough プラグイン機能が提供する機能に対するサポートが追加されています。ただし、Application Server Enterprise Edition 8.2 での auth-passthrough プラグイン機能の設定方法は異なります。

Application Server Enterprise Edition 7.1 での auth-passthrough プラグイン関数は、次に示す 2 層配備のシナリオで有効でした。

- Application Server インスタンスは、企業ファイアウォールの内側にある 2 番目のファイアウォールによって保護される。
- Application Server インスタンスへの直接のクライアント接続は許可されない。

このようなネットワークアーキテクチャーの場合、クライアントは、service-passthrough プラグイン関数で設定されたフロントエンド Web サーバーに接続し、HTTP 要求を、プロキシされた Application Server インスタンスに転送して処理します。Application Server インスタンスは、要求をクライアントホストから直接にではなく、Web サーバープロキシからしか受信できません。その結果、プロキシされた Application Server インスタンス上に配備され、クライアントの IP アドレスなどのクライアント情報を照会する任意のアプリケーションは、中継された要求の実際の発信元ホストであるプロキシホストの IP を受信します。

## 解決方法

Application Server Enterprise Edition 7.1 では、プロキシされた Application Server インスタンス上で、そのインスタンス上に配備された任意のアプリケーションがリモートクライアントの情報を直接使用するように auth-passthrough プラグイン関数を設定できました。その場合は、プロキシされた Application Server インスタンスが、service-passthrough プラグインを実行している中間の Web サーバー経由ではなく、要求を直接受信したかのように見えます。

Application Server Enterprise Edition 8.2 では、domain.xml 内の <http-service> 要素の authPassthroughEnabled プロパティを TRUE に設定することにより、auth-passthrough 機能を有効にすることができます。次に例を示します。

```
<property name="authPassthroughEnabled" value="true"/>
```

Application Server Enterprise Edition 7.1 にある auth-passthrough プラグイン関数のセキュリティに関する同じ注意点が、Application Server Enterprise Edition 8.2 にある authPassthroughEnabled プロパティにも適用されます。authPassthroughEnabled によって、認証目的に使用される可能性のある情報(要求発信元の IP アドレスや SSL クライアント証明書など)を上書きすることが可能になるため、authPassthroughEnabled を TRUE に設定して Application Server Enterprise Edition 8.2 イン

スタンスへの接続を許可する場合は、その対象を信頼できるクライアントまたはサーバーだけに限定することがきわめて重要です。予防措置として、`authPassthroughEnabled` を `TRUE` に設定するのは、企業ファイアウォールの内側にあるサーバーだけにすることをお勧めします。インターネット経由でアクセス可能なサーバーでは、決して `authPassthroughEnabled` を `TRUE` に設定しないでください。

プロキシ Web サーバーが `service-passthrough` プラグインを使用して設定されており、要求を `authPassthroughEnabled` が `TRUE` に設定された Application Server 8.1 Update 2 インスタンスに転送するシナリオでは、SSL クライアント認証は Web サーバープロキシ上で有効になり、プロキシされた Application Server 8.1 Update 2 インスタンス上で無効になる可能性があることに注意してください。この場合、プロキシされた Application Server 8.1 Update 2 インスタンスは、SSL 経由で認証されたかのように引き続き要求を処理し、クライアントの SSL 証明書を、それを要求している任意の配備されたアプリケーションに提供します。

`--enabled=false` で作成された HTTP リスナーが無効にならない。(ID 6190900)

#### 説明

`--enabled=false` フラグで `httplistener` を作成すると、リスナーは無効になりません。フラグ `--enabled` は、リスナーの作成と同時に使用すると効果がありません。

#### 解決方法

リスナーを使用可能状態で作成して、あとで手動で無効にしてください。

`verify_file_user_exists_common` が実行されないために、**Windows** での再配備が失敗する。(ID 6490227)

#### 説明

Windows で、配備前にユーザーを作成するアプリケーションを再配備すると、`verify_file_user_exists_common` が呼び出されても実行されないために `create-file-user` コマンドが失敗することがあり、ユーザーがすでに存在することを通知することに失敗します。`deploy` ターゲットの実行はこの時点で停止し、配備と配備取消しが失敗します。

#### 解決方法

まず `keydel` ターゲットを使用してファイルユーザーを削除し、次に `deploy` ターゲットをもう一度実行します。

```
asant keydel
asant deploy
```

