



Sun Java System Application Server 9.1 リリースノート



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

Part No: 820-3515
2007年9月

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

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 9.1 製品は、Java EE アプリケーションおよび Java Web サービスの開発と配備に利用できる Java EE 5 プラットフォーム互換サーバーです。このサーバーを業務用に使用する場合も費用は発生しません。Sun Java System Application Server 9.1 の開発、配備、および再頒布は自由に行うことができます。再頒布を検討している場合は、再頒布ライセンスについて [Sun OEM sales](#) にお問い合わせください。

Sun Java System Application Server 9.1 製品は、J2EE アプリケーションと Web サービスアプリケーション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。Application Server を使用して Web サービスを開発すれば、開発プロセスが単純になるだけでなく、独自の方法で柔軟に進化させることもできます。

- 3 ページの「このリリースノートについて」
- 4 ページの「リリースノートの変更履歴」
- 4 ページの「アクセシビリティ機能」
- 5 ページの「関連マニュアル」
- 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」
- 7 ページの「その他の情報」

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 9.1 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 9.1 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/1343.4?q=glassfish>) 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。

このマニュアルでは、第三者が提供しているURL で関連する追加情報を参照しません。

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。

リリースノートの変更履歴

ここでは、Sun Java System Application Server 9.1 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 リリースノートの変更履歴

改訂日付	内容
2007年2月	Sun Java System Application Server 9.1 製品の初版リリース。
2007年3月	Sun Java System Application Server 9.1 製品のベータ版1リリース。
2007年5月	Sun Java System Application Server 9.1 製品のベータ版2リリース。
2007年9月	Sun Java System Application Server 9.1 製品のFCSリリース。

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手に入れるには、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。最新バージョンのアプリケーションは、次のサイトで入手できます。
<http://sun.com/software/javaenterprisesystem/get.html>

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。

関連マニュアル

表 1-2 Application Server のマニュアルセットの内容

マニュアルタイトル	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。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、JDK、および JDBC/RDBMS の包括的な表ベースの概要を含みます。
『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。
『Installation Guide』	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念事項についても説明しています。
『Developer's Guide』	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。開発者ツール、セキュリティー、アセンブリ、配備、デバッグ、ライフサイクルモジュールの作成に関する一般情報を含みます。
『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 (HA) データベースのためのインストール後の設定と管理の手順。
『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せん。
『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。

表 1-2 Application Server のマニュアルセットの内容 (続き)

マニュアルタイトル	説明
『Reference Manual』	Application Server で使用できるユーティリティーコマンド。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を含みます。

問題の報告とフィードバックの方法

Sun Java System Application Server で問題が発生した場合は、次のいずれかの方法でカスタマー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。
- **Glassfish メーリングリスト**
(<https://glassfish.dev.java.net/servlets/ProjectMailingListList>) — いろいろな話題とフィードバックのためのさまざまな Glassfish コミュニティーメーリングリストがあります。
- **Java Developer Connection のバグデータベース** (<http://developer.java.sun.com/servlet/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/>) — 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 9.1 製品について議論するには、この Java EE 5 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>) — 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は、専用ダイヤルをご利用ください。
問題解決へのお手伝いが円滑に行えるよう、サポートに連絡する際には次の情報をご用意ください。
- 問題が発生した状況および操作への影響などの、問題の具体的説明
- 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
- エラーログまたはコアダンプ

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。

<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。このオンラインフォームでは、マニュアルのタイトルと Part No. もご記入ください。Part No. は、マニュアルのタイトルページか先頭に記述されている 7 桁または 9 桁の番号です。たとえば、このリリースノートの場合、タイトルは『Sun Java System Application Server 9.1 リリースノート』、Part No. は 820-3515 です。

その他の情報

以下の場所から有益な情報を入手することができます。

- Application Server の製品情報
(http://www.sun.com/software/products/appsrvr/home_appsrvr.html)
- Sun Java System マニュアル (<http://docs.sun.com/prod/java.sys>)
- Sun Java System プロフェッショナルサービス
(<http://www.sun.com/service/sunps/sunone>)
- Sun Java System ソフトウェア製品とサービス (<http://www.sun.com/software>)
- Sun Java System Software Support Services
(<http://www.sun.com/service/sunone/software>)
- Sun Java System サポートと Knowledge Base
(<http://www.sun.com/service/support/software>)
- Sun サポートおよびトレーニングサービス (<http://training.sun.com>)
- Sun Java System コンサルティングとプロフェッショナルサービス
(<http://www.sun.com/service/sunps/sunone>)
- Sun Java System 開発者情報 (<http://developers.sun.com>)
- Sun 開発者サポートサービス (<http://www.sun.com/developers/support>)
- Sun Java System ソフトウェアトレーニング
(<http://www.sun.com/software/training>)
- Sun Software Data Sheets (<http://www.sun.com/software>)
- Sun Microsystems の製品マニュアル (<http://docs.sun.com/>)

Application Server 9.1 について

Sun Java System Application Server 9.1 は、大規模な本稼働環境での Java EE アプリケーションおよび Java テクノロジベースの Web サービスの開発と配備に利用できる Java EE 5 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「9.1 Release の新機能」
- 12 ページの「ハードウェアとソフトウェアの要件」
- 22 ページの「Java EE 5 プラットフォームの API」
- 23 ページの「Java EE 5 SDK」
- 24 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」
- 24 ページの「既知の Java EE 5 互換性の問題」
- 25 ページの「Application Server アップデートセンターの使用」
- 26 ページの「WSIT 統合の詳細について」

9.1 Release の新機能

Application Server 9.1 は、次に示す多くの新機能を実装しています。

- **Java EE 5 プラットフォームのサポート** — Sun Java System Application Server Enterprise Edition 9.1 では、Java EE 5 仕様を実装して、次世代のエンタープライズアプリケーションおよび Web サービスにもっとも適したアプリケーションランタイムの 1 つを実現します。Application Server 9.1 は、次の Java EE 標準を実装しています。
 - Enterprise Java Beans 3.0
 - JAXB 2.0
 - Java Persistence
 - Java Server Faces 1.2
 - Java Server Pages 2.1 (JSP 2.1)
 - Java Server Pages Standard Tag Library (JSTL) 1.2
 - Streaming API for XML (StAX)
 - Web Services Metadata

- Java API for XML based Web Services 2.0 (JAX-WS 2.0)
- Common Annotations for the Java Platform 1.0 (CAJ 1.0)
- Java Servlet 2.5

一連のすべての Java EE 5 プラットフォームテクノロジーについて、このノートで後述します。

- **Web サービス相互運用性テクノロジー (WSIT) サポート** – Sun は Microsoft と密接に連携して、メッセージ最適化、高信頼性メッセージング、およびセキュリティーなどの Web サービスエンタープライズテクノロジーの相互運用性を実現しています。WSIT の初期リリースはこの協力の成果です。WSIT は、エンタープライズ機能をサポートする多くのオープンな Web サービス仕様を実装したものです。メッセージ最適化、信頼できるメッセージング、およびセキュリティーに加えて、WSIT にはブートストラップと設定のテクノロジーも含まれています。現在 Java プラットフォームに組み込まれているコア XML サポートを基本にして、WSIT は既存の機能を使用または拡張し、相互運用可能な Web サービスのための新しいサポートを追加します。それらには、次のサポートが含まれます。
 - ブートストラップおよび設定
 - メッセージ最適化テクノロジー
 - 高信頼性メッセージングテクノロジー
 - セキュリティーテクノロジー

Application Server 9.1 での WSIT 統合の詳細については、この章の [26 ページ](#) の「[WSIT 統合の詳細について](#)」で後述します。

- **Java Business Integration (JBI) サポート** – JBI はビジネス統合 Service Provider Interfaces (SPI) によって Java EE を拡張します。これらの SPI を使用することにより、開発者は WSCI、BPEL4WS、および W3C Choreography Working Group などの仕様のための Java ビジネス統合環境を作成または実装できます。JBI 実装は Application Server インストーラにより通常は AS_HOME/jbi ディレクトリにインストールされます。Application Server JVM で JBI フレームワークを開始する lifecycle モジュールを含む、JBI 用のすべての共通 JAR ファイルとシステムコンポーネントがこのディレクトリに含まれています。
- **インメモリーレプリケーションサポート** – ほかのサーバー上でインメモリーレプリケーションを実行することにより、HADB などの別個のデータベースを入手しなくてもセッション状態データの軽量ストレージを用意できます。このタイプのレプリケーションは、ほかのサーバー上のメモリーを使用して HTTP セッションとステートフルセッション Bean データの高可用性ストレージを実現します。クラスタ化されたサーバーインスタンスはセッション状態をリングトポロジで複製します。各バックアップインスタンスは複製されたデータをメモリーに格納します。セッション状態データを他のサーバー上のメモリーに複製することによって、セッションを分散することが可能になります。インメモリーレプリケーションを使用するには、Group Management Service (GMS) を有効にする必要があります。デフォルトで有効です。

- 使用状況プロファイル-どの管理ドメインもそのドメインの能力を示す使用状況プロファイルと関連付けられています。Application Server 9.1には次のプロファイルが用意されています。
 - 開発者-ドメインを開発環境で実行している場合や、アプリケーションに負荷分散、高可用性、セッションレプリケーションなどのクラスタ化機能が必要でない場合に、このプロファイルを使用します。このプロファイルの実際の名前は「developer」で、大小文字を区別することに注意してください。
 - クラスタ-アプリケーションサーバーインスタンスのクラスタを作成して、配備された Java EE アプリケーションにスケーラビリティと高可用性を付与する場合は、このプロファイルを使用します。アプリケーションの状態はインメモリーのままとなります。このプロファイルの実際の名前は「cluster」で、大小文字を区別することに注意してください。
 - エンタープライズ-HADB と NSSが必要な場合に、このプロファイルを使用します。このプロファイルは、HADB と NSSが別々にインストールされていたり、Application Server が Java Enterprise System (Java ES) の一部としてインストールされていたりする場合には使用できません。
- 負荷分散の拡張機能-負荷分散プラグインにいくつかの拡張機能が追加されました。簡潔にまとめると、次の機能が含まれます。
 - 重み付けラウンドロビン-オプション属性 weight がインスタンス Loadbalancer.xml ファイルに追加されています。このオプションを使用することにより、ロードバランサプラグインは重みに応じて要求を経路指定できるようになります。たとえば、500 の要求のうち、100 が instance1 へ、400 が instance2 へと毎回振り分けられるようになります。デフォルトの重みは 100 です。重みは管理コンソールまたはコマンドラインから各インスタンスに対して割り当てられ、サーバー domain.xml にはインスタンスごとの重みを示す属性があります。
 - ユーザー定義によるロードバランサの方法指定-ユーザーが負荷分散のカスタムロジックを定義できます。たとえば、ユーザー ID に基づくリダイレクトや MIME ベースの負荷分散があります。この機能は、ロードバランサがロードするユーザー定義の共用ライブラリによって実装されます。このカスタム共用ライブラリは、appserver_install_dir /lib/install/templates/にある loadbalancer.h に定義されているとおりにインタフェースを実装します。
 - 管理拡張機能-以前のバージョンの Application Server では、loadbalancer.xml ファイルを特定のサーバーの config ディレクトリに手でコピーすることが必要でした。Application Server 9.1には、このようなコピーを Web サーバーと Application Server の間で実行するプッシュ自動化が組み込まれています。ロードバランサ自体はサーバーの domain.xml ファイルで設定されます。
- オープンソースおよび GlassFish コミュニティー-2005年6月、Sun は、Java EE 5 プラットフォームの最新機能とそれに関連したエンタープライズテクノロジーを実装した、無償でオープンソースの商用レベルのアプリケーションサーバーを開発することを目標として、GlassFish (<http://java.sun.com/javaee/glassfish/>) コ

コミュニティを立ち上げました。Application Server Enterprise Edition 9.1 は、Sun の技術者と GlassFish コミュニティが開発したソースコードを基盤に作成されています。

- アップデータセンターサポート - Application Server 9.1 アップデータセンターにより、Application Server の更新が自動化され、追加コンポーネントが簡単に利用できるようになりました。詳細は、25 ページの「Application Server アップデータセンターの使用」を参照してください。

ハードウェアとソフトウェアの要件

この節では、Sun Java System Application Server 9.1 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。

- 12 ページの「対応プラットフォーム」
- 13 ページの「重要パッチ情報」
- 13 ページの「JDK バージョン」
- 14 ページの「JDBC ドライバとデータベース」
- 15 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「Web サーバー」
- 19 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」
- 21 ページの「Sun Java System Application Server のアップグレード」
- 21 ページの「その他の要件」

対応プラットフォーム

次の表に、Sun Java System Application Server Enterprise Edition 9.1 製品と互換性があるオペレーティングシステムを示します。

表 2-1 サポートされているオペレーティングシステム

オペレーティングシステム	最小メモリー	推奨するメモリー	最小ディスク容量	推奨するディスク容量	JVM
Sun Solaris 9、10 (SPARC)	512M バイト	512M バイト	250M バイトの空き領域	500M バイトの空き領域	J2SE 5.0 Java SE 6
Solaris 9、10 (x86)					
64 ビット版 Sun Solaris 10 (SPARC、x86)	512M バイト	512M バイト	250M バイトの空き領域	500M バイトの空き領域	J2SE 5.0 Java SE 6

表 2-1 サポートされているオペレーティングシステム (続き)

オペレーティングシステム	最小メモリ	推奨するメモリ	最小ディスク容量	推奨するディスク容量	JVM
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 5.0 Java SE 6
SuSE Linux Enterprise Server 10 SP1	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 5.0 Java SE 6
Windows Server 2000 SP4+ Windows 2000 Advanced Server SP4+ Windows Server 2003 Windows XP Pro SP1+ Windows Vista	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 5.0 Java SE 6

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。

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 推奨パッチクラスタ」をインストールすることをお勧めします。このパッチクラスタは [SunSolve \(http://sunsolve.sun.com/pub-cgi/show.pl?target=patchpage\)](http://sunsolve.sun.com/pub-cgi/show.pl?target=patchpage) & セキュリティーパッチ」から入手できます。

JDK バージョン

Application Server 9.1 に必要な JDK の動作が保証されている最小バージョンは **1.5.0_13** です。JDK 1.5.0_13 は <http://java.sun.com/javaee/downloads/?intcmp=1282> からダウンロードできます。

JDBC ドライバとデータベース

表 2-2 に、Java EE の互換性要件を満たすデータベースとドライバを示します。サポートされるすべての Sun Java System Application Server Enterprise Edition の構成では、バンドル版 Java DB データベースとドライバなどの、この表に基づくデータベースとドライバの組み合わせを少なくとも 1 つ含める必要があります。さらに、Application Server は、対応する JDBC ドライバを使用してほかのどの DBMS に対する JDBC 接続もサポートするように設計されています。

表 2-2 Java EE 互換の JDBC ドライバ

JDBC ドライバベンダー	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
Derby Network Client	Type 4	Derby 10.2
DataDirect 3.60.29	Type 4	Oracle 10g
DataDirect 3.6	Type 4	DB2 9.1 DB2 8.2
Oracle Thin	Type 4	Oracle 10g
PostGres	Type 4	8.1
MySQL Connector/J Driver 3.1	Type 4	MySQL 5

表 2-3 に、Application Server FCS リリースでサポートを予定しているその他のデータベースとドライバをすべて示します。

表 2-3 サポートされる JDBC ドライバとデータベース (Application Server FCS)

JDBC ドライバベンダー	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
Oracle OCI	Type 2	Oracle 9i
IBM DB2	Type 2	DB2 9.1
Oracle 10G	Type 4	-RAC
DataDirect 3.60.29	Type 4	Sybase ASE 15 MS SQL 2005
MySQL Connector/J Driver 3.1	Type 4	MySQL 5

通常、Application Server Enterprise Edition 9.1 は、Java EE 仕様に適合するすべての JDBC ドライバをサポートします。

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 9.1 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。Java DB は、[Apache Derby データベース](#)を基盤としています。

- 15 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」
- 16 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 9.1 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。

- Java DB Network Server のインスタンスを開始するには、`asadmin start-database` コマンドを使用します。

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

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 データベースの格納場所です。デフォルトの `path` は、`appserver_install_dir/javadb` です。

- Java DB Network Server の実行中のインスタンスを停止するには、次のように `asadmin stop-database` コマンドを使用します。

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

Java DB ユーティリティースクリプト

注 - このスクリプトは Application Server 9.1 では推奨されていません。

Application Server 9.1 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>/javadb/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>/javadb 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ーの詳細については、Derby のツール (<http://db.apache.org/derby/docs/10.1/tools/>) および Admin (<http://db.apache.org/derby/docs/10.1/aminguide/>) 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、コマンドの Pointbase 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 sampledb.sql;
SQL> unload database sampledb.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。
- プール値 `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。

Web サーバー

この節では、Sun Java System Application Server 9.1 向けにサポートされている Web サーバーを示します。

表 2-4 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	6.1+	Solaris SPARC 8、9、10
		Solaris x86 9、10
		Red Hat Enterprise Linux 2.1 Update 2、3.0 Update 1

表 2-4 サポートされている Web サーバー (続き)

Web サーバー	バージョン	オペレーティングシステム
Apache Web Server	1.3+, 1.4, 2.0	Solaris SPARC 9、10 Solaris x86 10 Red Hat Enterprise Linux 2.1 Update 2、3.0 Update 1
Microsoft IIS™	5.0+	Windows Server 2003 Windows 2000 Advanced Server Service Pack 4+

Windows や HP-UX をはじめとして、その他にもサポートされるプラットフォームが今後追加される予定です。

ブラウザ

この節では、Sun Java System Application Server 9.1 でサポートされるブラウザを示します。

表 2-5 サポートされている Web ブラウザ

ブラウザ	バージョン
Mozilla	1.7.12
Internet Explorer	6.0 Service Pack 2
Firefox	1.0.7
Safari	2
Netscape	8.0.4

HADB の要件とサポートされているプラットフォーム

12 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。

- 20 ページの「対応プラットフォーム」
- 20 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 - HADB は Application Server 9.1 Enterprise プロファイルにのみバンドルされています。HADB は *** からダウンロードできます。システムの Java コンポーネントは JDK 5 を使用して構築され、JDK 1.5 上でテストされています。

対応プラットフォーム

- **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、[37 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 320M 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 512M 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

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手順については、『Sun Java System Application Server 9.1 Update 1 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。ほかの方法を使用して削除すると、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ドメインとして示します。デフォルトでは、
 - HTTP については 8080、HTTPS については 8181
 - IIOP については 3700、IIOP/SSL については 3820、相互認証の IIOP/SSL については 3920
 - Administration Server については 4848 (HTTP)
 - pure JMX Client については 8686 (RMI) ポートが初期デフォルトポートです。

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 9.1 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) —**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、『Sun Java System Application Server 9.1 Update 1 Upgrade and Migration Guide』を参照してください。
- **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 9.1 Update 1 Upgrade and Migration Guide』を参照してください。

Java EE 5 プラットフォームの API

Sun Java System Application Server Enterprise Edition 9.1 は、Java EE 5 プラットフォームをサポートしています。Java EE 5 プラットフォームで使用できる拡張 API を、次の表に示します。

表 2-6 Java EE 5 プラットフォームの主要な API の変更点

API	JSR
Java EE 5	
Java Platform, Enterprise Edition 5 (http://java.sun.com/javaee/5/docs/api/)	JSR 244 (http://jcp.org/aboutJava/communityprocess/pr/jsr244/)
Web サービステクノロジー	
エンタープライズ Web サービスの実装	JSR 109 (http://jcp.org/en/jsr/detail?id=109)
Java API for XML-Based Web Services (JAX-WS) 2.0 (https://jax-ws.dev.java.net/)	JSR 224 (http://jcp.org/en/jsr/detail?id=224)
Java API for XML-Based RPC (JAX-RPC) 1.1 (https://jax-rpc.dev.java.net/)	JSR 101 (http://jcp.org/en/jsr/detail?id=101)
Java Architecture for XML Binding (JAXB) 2.0 (https://jaxb.dev.java.net/)	JSR 222 (http://jcp.org/en/jsr/detail?id=222)
SOAP with Attachments API for Java (SAAJ) (https://saaj.dev.java.net/)	JSR 67 (http://jcp.org/en/jsr/detail?id=67)
Streaming API for XML (http://java.sun.com/webservices/docs/1.6/tutorial/doc/SJSXP.html)	JSR 173 (http://jcp.org/en/jsr/detail?id=173)
Java プラットフォーム用 Web サービスメタデータ	JSR 181 (http://jcp.org/en/jsr/detail?id=181)
コンポーネントモデルテクノロジー	
Enterprise JavaBeans 3.0 (http://java.sun.com/products/ejb/)	JSR 220 (http://jcp.org/en/jsr/detail?id=220)
J2EE Connector Architecture 1.5 (http://java.sun.com/j2ee/connector/)	JSR 112 (http://jcp.org/en/jsr/detail?id=112)
Java Servlet 2.5 (http://java.sun.com/products/servlet/)	JSR 154 (http://jcp.org/en/jsr/detail?id=154)
JavaServer Faces 1.2 (http://java.sun.com/j2ee/javaserverfaces/)	JSR 252 (http://jcp.org/en/jsr/detail?id=252)
JavaServer Pages 2.1 (http://java.sun.com/products/jsp/)	JSR 245 (http://jcp.org/en/jsr/detail?id=245)

表 2-6 Java EE 5 プラットフォームの主要な API の変更点 (続き)

API	JSR
JavaServer Pages Standard Tag Library 1.2 (http://java.sun.com/products/jsp/jstl/)	JSR 52 (http://jcp.org/en/jsr/detail?id=52)
管理テクノロジー	
J2EE Management (http://java.sun.com/j2ee/tools/management/)	JSR 77 (http://jcp.org/en/jsr/detail?id=77)
J2EE Application Deployment (http://java.sun.com/j2ee/tools/deployment/)	JSR 88 (http://jcp.org/en/jsr/detail?id=88)
Java Authorization Contract for Containers (http://java.sun.com/j2ee/javaacc/)	JSR 115 (http://jcp.org/en/jsr/detail?id=115)
その他の Java EE テクノロジー	
Common Annotations for the Java Platform	JSR 250 (http://jcp.org/en/jsr/detail?id=250)
Java Transaction API (JTA) (http://java.sun.com/products/jta/)	JSR 907 (http://jcp.org/en/jsr/detail?id=907)
JavaBeans Activation Framework (JAF) 1.1 (http://java.sun.com/products/javabeans/glasgow/jaf.html)	JSR 925 (http://jcp.org/en/jsr/detail?id=925)
JavaMail (http://java.sun.com/products/javamail/)	JSR 919 (http://jcp.org/en/jsr/detail?id=919)
Java Message Service API (http://java.sun.com/products/jms/)	JSR 914 (http://www.jcp.org/en/jsr/detail?id=914)
Java Persistence API (http://java.sun.com/j2ee/persistence/faq.html)	JSR 220 (http://www.jcp.org/en/jsr/detail?id=220)

Java EE 5 SDK

Sun Java System Application Server 9.1 は、Java EE 5 SDK の一部として使用できます。

Java EE 5 SDK には、次の 2 つのバージョンがあります。

- Java Application Platform SDK (<http://java.sun.com/javaee/downloads/>)
- Java EE 5 SDK (<http://java.sun.com/javaee/downloads/index.jsp>)

さらに、これらの SDK 配布を JDK または NetBeans 5.5.1 ツールとともにダウンロードすることもできます。詳細については、<http://java.sun.com/javaee/downloads/index.jsp> のダウンロードページにアクセスしてください。

注 - GlassFish V2 と Application Server 9.1 は、Java EE アプリケーションの開発では NetBeans 5.5 と連携していません。NetBeans 5.5.1 を使用する必要があります。

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 9.1 では、基盤となる JVM として J2SE 5.0 以上が必要です。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)

注 - 新しい Java VM でドメインを作成したあとに以前の Java バージョンにダウングレードすることは推奨されていません。JVM をダウングレードする必要がある場合は、ドメインごとにそうすることをおすすめします。次の手順はその方法を説明したものです。

▼ 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、希望する **Java SDK (JRE ではない)** をダウンロードしてシステムに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 **JDK** を変更するドメインを開始します。

```
as_install/bin/asadmin start-domain domain_name
```

- 3 管理コンソールにログインして、ドメインの **JVM** 属性を変更します。

特に、ドメインの「JVM 設定」ページの `JAVA_HOME` 変数を変更することが必要です。

または、`asadmin` コマンドを使用することもできます。

```
as_install/bin/asadmin set "server.java-config.java-home= Path-To-Java-Home"
```

既知の Java ES 5 互換性の問題

Application Server 9.1 と Java ES 5 (JES5) の間には 2 つの既知の互換性の問題があります。

1. Application Server 9.1 では JSF 1.2 バージョンが提供されるため、JES5 Service Registry は Application Server 9.1 と互換性がありません。Application Server を 9.1 にアップグレードする前に、Service Registry を JES5u1 にアップグレードする必要があります。この制限については Service Registry のマニュアルにも記載されています。

- Application Server 9.1 では JSF 1.2 バージョンが提供されるため、JES5 Portal Server は Application Server 9.1 と互換性がありません。Application Server を 9.1 にアップグレードする前に、Portal Server を JES5u1 にアップグレードする必要があります。

Application Server 8.2 上で Portal Server を実行する Java ES 5 設定環境で、Application Server 8.2 を 9.1 にアップグレードすると、Portal Server が使用できなくなります。

Application Server 9.1 は JSF 1.2 を使用しますが、リリース 5 Portal Server の JSF-Portlet ブリッジが JSF 1.2 をサポートしていません。Solaris または Linux の場合、Portal Server を JavaES5 Update 1 にアップグレードする必要があります。

Windows の場合、Portal Server は Windows 上の Java ES 5 Update 1 ではサポートされていないため、Release 5 Portal Server を引き続き使用するのであれば、Application Server を 9.1 にアップグレードしないでください。

Application Server アップデートセンターの使用

Application Server 9.1 アップデータセンターにより、Application Server の更新が自動化され、追加コンポーネントが簡単に利用できるようになりました。

アップデートセンターを有効にすると、自動でソフトウェアの更新を実行します。この自動更新プロセス中に、アップデートセンターは次のデータを収集して Sun Microsystems または購入先に転送します。

- 固有インストール ID (GUID)
- IP アドレス
- オペレーティングシステム情報 (名前、バージョン、アーキテクチャー、ロケール)
- JDK バージョン
- モジュールダウンロード情報 (モジュール名、日付、時刻、状況、ダウンロード時間、ダウンロード済みバイト数)

個人情報は追跡しません。どの個人情報も他のデータと関連付けられておらず、レポート用に使用されません。

▼ アップデートセンターを使用するには

自動アップデートへの同意を明示するため、アップデートセンターはデフォルトでは無効になっています。アップデートセンターを有効にして、定期チェックと自動アップデートを実行するには、次の手順に従います。

- 1 アップデートセンターを開始します。
 - UNIX の場合: <installdir>/updatecenter/bin/updatetool
 - Windows の場合: <installdir>\updatecenter\bin\updatetool.bat
- 2 「プリファレンス」タブをクリックします。

- 3 「Update Scheduling」ウィンドウで、「Check for Updates」ドロップダウンボックスの値を「Never (Manual)」から希望する値に変更します。一例として、毎日や毎週などを指定できます。
- 4 アップデートを希望する曜日と時刻を指定します。
- 5 変更を保存するには、「保存」ボタンを選択します。
これで、アップデートセンターは指定したスケジュールにしたがって、Glassfish または Application Server コンポーネントのアップデートを自動的に確認します。アップデートが利用可能である場合には、アップデートセンターが起動して更新可能なコンポーネントを通知します。

WSIT 統合の詳細について

WSIT ステータスの詳細については、[WSIT Status Notes \(https://wsit.dev.java.net/source/browse/*checkout*/wsit/wsit/status-notes-1-0-FCS.html\)](https://wsit.dev.java.net/source/browse/*checkout*/wsit/wsit/status-notes-1-0-FCS.html) ページを参照してください。Application Server での WSIT の使用について、『The WSIT Tutorial』も参照してください。

既知の問題点と制限事項

ここでは、Sun Java System Application Server 9.1 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方法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。この内容は次の項目から構成されています。

- 27 ページの「管理」
- 33 ページの「Apache とロードバランサプラグイン」
- 34 ページの「アプリケーションクライアント」
- 35 ページの「付属の Sun JDBC ドライバ」
- 37 ページの「マニュアル」
- 37 ページの「高可用性」
- 47 ページの「インストール」
- 50 ページの「Java EE Tutorial」
- 51 ページの「Java Persistence」
- 51 ページの「ライフサイクル管理」
- 53 ページの「ロギング」
- 53 ページの「メッセージキュー」
- 54 ページの「監視」
- 55 ページの「サンプル」
- 57 ページの「セキュリティ」
- 58 ページの「アップグレードユーティリティ」
- 65 ページの「Web コンテナ」
- 69 ページの「Web サーバー」
- 69 ページの「Web サービス」

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。(6171458)

説明

デフォルトでは、\$INSTALL/lib/package-appclient.xml に、asenv.conf によってポイントされる domain1 の AS_ACC_CONFIG 変数用にハードコードされた値があります。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

次のいずれかを実行します。

1. 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。
2. domain1 を削除し、\$INSTALL/lib/package-appclient.xml で domain1 用にハードコードされた値を、新たな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。

バックアップしたドメインを別の名前では復元できない(6196993)

説明

同一の Application Server 上でのドメインのミラー化は、backup-domain および restore-domain コマンドでは行えません。asadmin restore-domain コマンドにはドメインをリネームするオプションがありますが、元の名前とは違う名前ではドメインを復元することはできないからです。バックアップ取得したドメインのリネームは正常に行われたように見えますが、そのリネーム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 コマンドで使用したドメイン名と同じである必要があります。Application Server 8.1 での backup-domain および restore-domain コマンドは、同一マシン上の同一ドメインのバックアップおよび復元についてだけ有効です。

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(6200011)

説明

Application Server では、J2SE 1.4.x または 5.0 以降を設定できます。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からサーバーを起動すると、Application Server VM 内に新たな 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 の管理コンソールやコマンド行インタフェース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 JMXService の URL に接続し、資格を正常に指定したあと、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 9.1 Administration Guide』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 (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「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(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。

`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(6288893)

解決方法

1. 既存の `<as_install>/bin/asant` スクリプトの名前を `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(6315957)

説明

Application Server のマニュアルに `.asadmintruststore` ファイルが記述されていません。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。

JMS ブローカに到達するまでにタイムアウトになるため、クラスタ化されたインスタンスが起動できない (6523663)

説明

Application Server クラスティンスタンスのデフォルト MQ 統合モードは `LOCAL` です。Application Server がインストールされている場所の `PATH` が長い、つまり短くない場合、クラスティンスタンス起動時に `imqbrokersvc.exe` で障害が発生します。`imqbrokersvc` のメモリー割り当てに問題があります。

解決方法

クラスティンスタンスの JMS サービスタイプを、デフォルトの `LOCAL` から `REMOTE` に変更する必要があります。この設定では、すべてのインスタンスは `DAS` ブローカを指します。次の手順に従って、`REMOTE` モードでクラスタを設定してください。

注 - `REMOTE` モード使用時には、すべてのインスタンスが 1 つのブローカ (`DAS`) を使用しているため、Application Server クラスタが起動してもブローカクラスタは作成されません。詳細については、<http://www.glassfishwiki.org/gfwiki/attach/OnePagersOrFunctionalSpecs/as-mq-integration-gfv2.txt> にある `One Pager` の 4.1 節、iii 項「`Auto-clustering`」を参照してください。それによると、上記の機能は将来使用できなくなります。

▼ コマンド行の使用

始める前に ご使用の環境に合わせて、ポートおよびパスワードファイルを変更してください。次に示す手順では、クラスタ名が `racluster`、`DAS` 管理ポートが `5858`、および `DAS` `JMS` ポートが `7676` になっています。

- 1 **JMS** タイプを `REMOTE` に変更して、クラスタ設定を変更します。

```
$AS91_HOME/bin/asadmin.bat set --port 5858 --user admin --passwordfile \ $AS91_HOME/bin/password_file  
racluster.jms-service.type=REMOTE
```

2 DAS JMS ホストに対応する JMS ホストを作成します。

```
$AS91_HOME/bin/asadmin.bat create-jms-host --port 5858 --user admin --passwordfile \ $AS91_HOME/bin/
password_file --target racluster --mqhost localhost --mqport 7676 \ --mquser admin --mqpassword admin
dashost
```

3 デフォルトの JMS ホストが前のステップで作成した DAS JMS ホストになるように設定します。

```
$AS91_HOME/bin/asadmin.bat set --port 5858 --user admin --passwordfile \ $AS91_HOME/bin/password_file
racluster.jms-service.default-jms-host=dashost
```

▼ 管理 GUI の使用

1 「設定」->「cluster_name-config」->「Java メッセージサービス」->「JMS ホスト」の順に移動します。

2 「新規」をクリックして新規 JMS ホストを作成し、dashost という名前を付けます。

3 DAS の JMS サービスに対応する設定を入力します。デフォルト設定は次のとおりです。

- ホスト名: localhost
- ポート: 7676
- 管理者ユーザー: admin
- パスワード: admin

これらの設定をご使用の DAS JMS サービスに適した値に変更してください。

4 「Java メッセージサービス」タブに戻って、JMS サービスタイプを REMOTE に変更します (デフォルトは LOCAL)。

5 default-jms-host ドロップダウンリストから dashost を選択します。

6 変更を保存してから、ノードエージェントまたはクラスタを起動します。

Netscape 8.1.3、Mozilla 1.7、および Safari 2.0.4 ブラウザで jmaki チャートが表示できない (6543014)

説明

「ログ統計の監視」ページから一部のサポートされていないブラウザを使用してチャートを表示しようとすると、次のエラーがスローされます。

```
Error loading jmaki.widgets.jmaki.charting.line.Widget : id=form1:jmaki_chart11 Script: http://easqelx5.
red.iplanet.com:4848/resources/jmaki/charting/ \ line/component.js (line:5437). Message: area.initialize
is not a function
```


解決方法

サポートされているブラウザを使用します。Application Server 9.1 でサポートされているブラウザについては、[19 ページの「ブラウザ」](#)を参照してください。

AS のメジャーリリースのたびに、デフォルトポートが変わっている (6566481)

説明

Application Server の過去 3 回のメジャーリリースのたびに、デフォルト管理ポートが変わっていました。具体的には、7.x、8.x、および 9.x のデフォルト管理ポートは次のようになっていました。

- AS 7.x: 4848
- AS 8.x: 4849
- AS 9.x: 4848

解決方法

これはバグではありませんが、このことに注意してください。デフォルト管理ポートは推奨値に過ぎません。今後の Application Server リリースでは、デフォルトに 4848 ポートが継続して使用されます。

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。

『[High-Availability Administration Guide](#)』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 ([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。

『[High-Availability Administration Guide](#)』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 ([6307976](#))

Apache のセキュリティーを実行するには、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) 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 (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あと、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

```
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。

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 (6193556)

説明

クライアント JAR (たとえば `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。

ACC が常に `localhost:3700` に接続しようとする (6527987)

説明

アプリケーションクライアントが常に `localhost:3700` に接続しようとしています。問題は、クライアントノードを呼び出す前に、一部のシステムプロパティが読み取られていなければならないことにあります。

解決方法

次をシステムプロパティとして (JAVA_CMD に -D を指定して) 設定します。これらを applclient コードで設定しないでください。

```
org.omg.CORBA.ORBInitialHost = server_instance_host org.omg.CORBA.ORBInitialPort = server_instance_port
```

sunpkcs11.jar がいないため、ドメインを開始できません (6571044)

説明

64 ビット Linux 上で実行している場合、ドメイン開始時次の例外があります。問題は jdk1.5.0_11/jre/lib/ext/ の下に sunpkcs11.jar がいないことです。

解決方法

これは 64 ビット Linux で既知の JDK バグであり、JDK 1.5.0_13 で修正されます。

ASQuickStartup が SelectionKey の代わりに **NULL** を返して、SocketChannel.keyFor() をブレイクする (Issue Tracker 3027)

説明

複数のセレクトタに SocketChannel が登録されている場合、socketChannel.keyFor(lastRegisteredSelector) を実行すると SelectionKey の代わりに NULL が返されます。

解決方法

これは JDK バグ 6562829 に関連しており、6.0 U3 で修正される予定です。回避方法は Application Server 9.1 に組み込まれており、keyFor API を読み込む前はセレクトタがラップ解除されています。これにより、JDK バグが修正されるまで、keyFor の正常な動作が可能になっています。

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。

PreparedStatement エラーが発生する (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.
```

解決方法 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については、『Sun Java System Application Server 9.1 Administration Guide』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.
```

解決方法 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ガアップすることがあります。

解決方法 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 9.1 Administration Guide』を参照してください。

マシンのリブート後または **Application Server** 開始後に、**Java DB** が起動しない (6515124)

説明

ホストシステムまたは Solaris ゾーンのリブート後、または Application Server 起動後に、付属の Java DB データベースが自動的に再起動しません。これはバグではなく、付属または他社製のアプリケーションで所定の動作です。問題は、Application Server インスタンスの前に Java DB を起動する必要があるということです。

解決方法

ホストマシンまたは Solaris ゾーンのリブート後、必ず Application Server が開始する前に Java DB が起動するようにしてください。一例として、次のようにします。

```
/opt/SUNWappserver/appserver/bin/asadmin start-database
```

asadmin コマンドのオプションについては、『Sun Java System Application Server 9.1 Quick Start Guide』の「Application Server Administration Tools」を参照してください。

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.CONNECTOR_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 例外がスローされる (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。

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。
- 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)

説明

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ふ、次のエラーで失敗することがあります。

```
HADB-E-21054: System resource is unavailable: HADB-S-05512: Attaching shared memory segment with key "xxxx" 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。

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 (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の空きスペースが十分で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 (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 (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、`createdomain` 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 (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 (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 (6262824)

説明

パスワードが `hadb` に格納される時に、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 (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 (6271063)

説明

HADBc 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、そのあと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あとに 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 (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 (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

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。

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 (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 (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 (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態で、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。

パスが / である別の **cookie** をブラウザが取ると、セッション状態が保持されない
(6553415)

説明

cookie のパスが「/」と等価であった場合、持続タイプとしてインメモリーレプリケーションを使用する「/」以外のコンテキストルートに配備された高可用性 Web アプリケーションの cookie と干渉し、高可用性 Web アプリケーションが HTTP セッション状態をまったく保持できなくなります。この状況になる可能性のある 1 つの一般的なシナリオは、同じブラウザを使用して「/」に配備されている管理 GUI と高可用性 Web アプリケーションとの両方にアクセスしている場合です。

解決方法

別のブラウザから「/」に配備されている Web アプリケーションにアクセスします。

<appserver-install> /lib の下に SASL32.DLL と ZLIB.DLL がいないため、LB が IIS 6 と連動しない (6572184)

説明

ロードバランサが Windows IIS 6 と連動するには、SASL32.DLL と ZLIB.DLL のファイルが必要です。これらのファイルは現在 <appserver-install> /lib に含まれていません。

解決方法

この 2 つの DLL ファイルを手動で <appserver-install> /lib にコピーします。これらのファイルは次の場所からダウンロードできます。

<http://download.java.net/javaee5/external/<OS>/aslb/jars/aslb-9.1-MS4-b5.jar>

<OS> は希望するプラットフォームを表し、次の値のいずれかになります。

- SunOS
- SunOS_X86
- Linux
- WINNT

大域ゾーンで DAS 作成/起動および HA パッケージ伝播に問題がある (6573511)

説明

大域ゾーンで Application Server を高可用性パッケージとともにインストールまたはアンインストールすると、次の 2 つの問題が起きます。

1. HA パッケージがすべてのゾーンにインストールされます。これは望ましくありません。
2. アンインストール時に、HA、MQ、JDK パッケージがすべてのゾーンから削除されます。これも望ましくありません。

ルートローカルゾーンからインストールまたはアンインストールする場合には、この問題は起きません。

解決方法

グローバルゾーンではなくローカルルートゾーンからインストールまたはアンインストールを実行します。

「/」に配備された高可用性 Web アプリケーションがインメモリーレプリケーションで複製された HTTP セッションを再開できない (Issue Tracker 2972)

説明

「/」に配備された高可用性 Web アプリケーションは、持続タイプとしてインメモリーレプリケーションを使用している場合、HTTP セッションを保持できません。

解決方法

インメモリーレプリケーションを持続タイプとして使用する高可用性 Web アプリケーションを「/」以外のコンテキストルートに配備します。このような Web アプリケーションを「/」で利用できるようにするには、Web アプリケーションが配備されている仮想サーバーの default-web-module として指定します。

ASLB インストーラが /usr/lib/mps パスを apachectl LD_LIBRARY_PATH に書き込まないため、Apache SSL を起動できない (6591878)

説明

Solaris 上で Apache 用の Application Server ロードバランサをインストール中に、インストーラは apachectl スクリプトの LD_LIBRARY_PATH を更新します。しかし、インストーラは /usr/lib/mps パスを正しく書き込みません。Solaris の場合、LD_LIBRARY_PATH にこのパスがないと、Apache セキュリティーインスタンスは開始しません。

解決方法

この問題は Solaris プラットフォームでのみ起きます。この問題を回避するために、LD_LIBRARY_PATH に /opt/SUNWappserver/appserver/lib/lbplugin/lib を追加します。

インスタンスまたはクラスタの「ロードバランサの有効化/無効化」が正しい状態を示すようにする必要がある (6595113)

説明

domain.xml に保存されている内容にかかわらず、「クラスタ/インスタンス」一般情報ページで「ロードバランサの有効化」ボタンが常にオンになっています。

解決方法

- クラスタ化されたインスタンスの場合、「インスタンス」タブを選択し、表プルダウンから「休止」アクションをクリックします。
- スタンドアロンインスタンスの場合、インスタンスが実行中であることを確認してから、インスタンスの「一般」画面で「休止」ボタンをクリックします。

AS9.1 EE IFR b58f/JES5 UR1。「不完全な」HAが検出されたため、レジストリサーバーをインストールできない。(6602508)

説明

(Solarisのみ) HADB 導入済みの SPARC Solaris 10 に Application Server 9.1 をインストール後に、Application Server を開始して JES 5 UR1 をレジストリサーバーとともにインストールしようとする、次のエラーを受け取ることがあります。

Dependency Error: Installation can not proceed because the version of HA Session Store 4.4.3 detected on this host is incomplete , and a compatible version is required by Service Registry Deployment Support.

解決方法

Application Server 9.1 IFR が Solaris マシンにインストールされた状態で、JES 5 UR1 からレジストリサーバーをインストールすることは不可能です。次の JES5 UR1 配布ディレクトリから、pkgadd コマンドを使用して、手動でレジストリサーバーパッケージをインストールする必要があります。

```
<path>/<OS>/Products/registry-svr/Packages
```

IE 6.0 ブラウザ固有:ロードバランサ設定ファイルをエクスポートすると、エラーがスローされる (6516068)

説明

(Internet Explorer 6のみ) ロードバランサ設定ファイル (loadbalancer.xml) を Internet Explorer 6 からエクスポートしようとする、sun-loadbalancer_1_2.dtd DTD ファイルが見つからないことを示すエラーメッセージがブラウザに表示されます。

解決方法

設定ファイルを保存するには、次の回避方法を使用します。

1. Internet Explorer の「ロードバランサ」ページで、「エクスポート」をクリックします。
「XML page cannot be displayed」メッセージが表示されます。
2. エラーフレームをクリックしてから、Internet Explorer で「ファイル」->「名前を付けて保存」を選択します。
3. loadbalancer.xml ファイルを選択したディレクトリに保存します。

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 (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。

インストールプログラムの最後の画面で「完了」ボタンをクリックすると、インストールプログラムは製品の「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 (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
```

例として以下があります。

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

Windows Vista に SDK をインストール中に見せかけの警告が出される (6533646)

説明

Windows Vista で付属の SDK をインストール中に、エラー「Unsupported Installation Platform Detected.」が発生することがあります。しかし、インストールは何の問題もなく成功します。

解決方法

これは実際には問題ではありません。Application Server は Windows Vista 上で実行するので、この誤ったメッセージは将来のバージョンで削除されます。

アンインストール時に productregistry ファイルが正しく更新されない。サイレントモードを使用してインストールできない (6571598)

説明

Application Server productregistry ファイルに共用コンポーネント設定が含まれている場合、Application Server のアンインストール処理で productregistry ファイルが正しく更新されないため、productregistry ファイルを名前変更または削除しないなら、それ以降のインストールでサイレントモードを使用できなくなります。productregistry ファイル内の共用コンポーネントエントリを変更せずに残しておくことは意図的ですが、そのためにそれ以後のインストールで混乱が生じます。

解決方法

アンインストールが正常に完了したことがアンインストールログファイルによって報告されたあとに、続けてインストールを実行する前に productregistry ファイルを削除します。以前のアンインストールが正常に完了したことを確認するには、<install_dir>に appserv_uninstall.class ファイルがあるかどうかを調べます。アンインストールが正常に完了した場合には、このファイルはありません。

注-インストールが正常に完了していない場合は、productregistry ファイルを削除しないでください。

productregistry ファイルは、Solaris では /var/sadm/install に、Linux では /var/tmp にあります。

IFR: ASを疎ローカルゾーンにインストールできませんでした。**MQ**パッケージに問題があります。(6555578)

説明

疎ローカルゾーンに Application Server をインストールする場合、Message Queue (MQ) が先にインストールされていないとインストールが失敗します。インストーラは MQ をインストールしようとしませんが、インストール全体が失敗します。

解決方法

Application Server を疎ローカルゾーンにインストールする前に、MQ をグローバルゾーンに手動でインストールする必要があります。この問題には2つの回避方法があります。

1. 最新の MQ パッケージを入手するために、Application Server 9.1 IFR インストールが収録されている同一のメディアから、MQ 4.1 を手動でグローバルゾーンにインストールします。
 - a. ご使用のプラットフォームに対応するインストーラを使用してください。

mq4_1-installer-SunOS.zip mq4_1-installer-SunOS_X86.zip mq4_1-installer-Linux_X86.zip mq4_1-installer-WINNT.zip

- b. 圧縮ファイルを解凍して、インストーラを実行します。

インストーラはmq4_1-installerディレクトリにあります。
2. IFR インストールのすべてのコンポーネントをグローバルゾーンにインストールします。この処理で、GZ の MQ のバージョンをチェックし、必要に応じて Application Server 9.1 IFR に付属のバージョンにアップグレードします。サンプルアプリケーションコンポーネントを選択してインストールするだけでも、MQ は IFR バージョンにアップグレードされます。
 - a. グローバルゾーンで Application Server インストールを実行しますが、サンプルコンポーネントのみを選択します。

サンプルコンポーネントインストールでは、MQ と Application Server 共用コンポーネントもすべてのゾーンにインストールされます。
 - b. Application Server インストールを再度実行しますが、今回は疎ローカルゾーンで実行します。

何の問題もなくインストールが完了するはずです。

IFR インストールの `-console` プロンプト「**Do you want to upgrade from previous Application?**」を削除する必要がある (6592454)

説明

Application Server 9.1 IFR インストーラで `-console` オプションを選択して実行すると (コマンド行モード)、次のプロンプトが表示されます。

```
Do you want to upgrade from previous Application Server version?
```

残念ながら、IFR インストーラではこのようなアップグレードをサポートしていないため、このプロンプトは誤りです。プロンプトに「yes」と応答すると、インストールは正常に続行しますが、アップグレードされないばかりか、インストールが完全に実行されたことが表示されません。

解決方法

Application Server インストールのアップグレードを希望する場合は、アップグレードツールを使用してください。

Java EE Tutorial

Sun Java System Application Server 9.1 で J2EE 1.4 Tutorial を実行するには、次の作業を行います。

- 「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。

注 - Application Server 9.1 のデフォルト管理ポートは再び 4848 になっています。詳細は、33 ページの「AS のメジャーリリースのたびに、デフォルトポートが変わっている (6566481)」を参照してください。

- 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。

Java EE Tutorial の例で MDB が失敗する (6591307)

説明

『The Java EE 5 Tutorial』の第 32 章「Java EE Examples Using the JMS API」に『The Java EE 5 Tutorial』の「An Application Example That Consumes Messages from a Remote Server」という節がありますが、この例は有効ではなくなっています。MDB はメッセージの受信に失敗します。2つのシステムの間でメッセージを送信する他の2つの例(『The Java EE 5 Tutorial』の「Running JMS Client Programs on Multiple Systems」および『The Java EE 5 Tutorial』の「An Application Example That Deploys a Message-Driven Bean on Two Servers」)は引き続き正しく動作します。

解決方法

今後の Application Server ビルドで修正されます。

Java Persistence

TopLink は Collection フィールド/プロパティがクローン可能であることを想定する (**Issue Tracker 556**)

説明

Object[] から Collection への変換に java.util.Arrays.asList() API を使用している場合、JDK はクローン可能ではない java.util.ArrayList の実装を返します。結果として次の例外が発生します。

```
The method invocation of the method [protected native java.lang.Object java.lang.Object.clone()] throws
java.lang.CloneNotSupportedException] on the object [[pkg.A id = xxx]], of class [class java.util.Arrays$
ArrayList], triggered an exception. Internal Exception: java.lang.reflect.InvocationTargetException Target
Invocation Exception: java.lang.CloneNotSupportedException: java.util.Arrays$ArrayList
```

この問題は https://glassfish.dev.java.net/issues/show_bug.cgi?id=556 で追跡されています。

解決方法

別のコレクションをそのコンストラクタを使用して作成します。一例として、次のようにします。

```
myCollection = new ArrayList(java.util.Arrays.asList(a))
```

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 プロパティを設定すると set コマンドが失敗する (**6193449**)

説明

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、set コマンドが失敗します。次のエラーメッセージが表示されます。

```
[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。`redelivery-interval-in-millis` の値が `minimum-delivery-interval-in-millis` の値よりも大きいことを確認する Application Server でのチェックに問題があります。

解決方法

次のように、これらの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。

DAS 以外の config 内の JMS 物理送信先を一覧表示すると、エラーがスローされる (6532532)

説明

`default-config` を使用して JMS 物理送信先を表示しようとする、エラーメッセージが表示されます。

解決方法

これは予想どおりの動作です。Application Server 9.1 では、`default-config` は設定情報のテンプレートであるため、JMS 操作 (`list` や `create` など) を `default-config` に対して実行できません。しかし、これらの JMS 操作をクラスタまたはスタンドアロンインスタンスの設定に対して実行することは可能です。

Win2003 のみ: ページング不可能なプールでメモリーリークが発生し、TCP スタックおよび RichAccess テストが中断する (6575349)

説明

(Windows 2003 のみ) Windows 2003 では、リッチアクセス機能実行時にメモリーリークが発生します。Win32 ページング不可能プールが増大し続け、ついには TCP/IP スタック全体が停止してしまうために起きる問題です。一度この障害が発生すると、TCP/IP スタックは回復可能状態のままになり、TCP/IP スタックを復元するには Windows 2003 システムをリブートするしかありません。

解決方法

これは Microsoft の問題 (サポート 案件番号: SRX070906600011) で、修正プログラムが利用可能です。詳細は、[Microsoft サポート](#)に連絡してください。

回避方法

この問題にも、上記の修正プログラムに加えて、2つの回避方法があります。

1. domain.xml http-listener 属性 blocking-enabled="true" を設定して Grizzly ブロックモードを使用するか、または次の http-listener プロパティを追加します。

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

2. Windows Vista または Windows XP を使用します。

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 (**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。

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

Linux 上にクラスタプロファイルがあると MQ ブローカが起動に失敗する (6524871)

説明

ドメインを作成しクラスタプロファイルを Linux システム上に配置すると、`java.lang.OutOfMemoryError: Java heap space` エラーが発生する場合があります、MQ ブローカが起動しないためにサーバーインスタンスが再起動できないことがあります。この状況になったシステムが回復することはありません。問題は `/etc/hosts` ファイルの設定ミスです。特に、サーバーホスト名がループバックアドレス `127.0.0.1` を指している場合です。

解決方法

設計上、ネットワークデバイスがループバックアドレスを指すように設定された状態で MQ ブローカクラスタを開始することはできません。これはバグではありません。回避方法は、Application Server ホストの `/etc/hosts` ファイルが `127.0.0.1` を指さないようにすることです。

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。

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 (6174518)

説明

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)

解決方法

これらの監視情報は将来のリリースで削除され、より適切な情報で置き換えられる予定です。

管理 GUI から JNDI ブラウザを開くと、server.log に大量の例外がダンプされる (6591734)

説明

管理 GUI から JNDI ブラウザを開くと、多くの例外がスローされます。

解決方法

現時点ではありません。

サンプル

ここでは、Application Server 9.1 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。

Message Queue 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、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 (6198003)

説明

asadmin の配備手順にしたがって、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、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。

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、**Web サービス/セキュリティ** 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 (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

次のいずれかを実行します。

- 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
```


アップグレード後に、**AS9.1** サンプルと **JES5** ポータルサンプルが **derby** ポート 1527 で競合する (**6574563**)

説明

Windows で、Application Server 9.1 にアップグレード後に、そのサンプルと JES5 ポータルサンプルが Derby ポート 1527 で競合します。特に、Application Server 9.1 は 0.0.0.0:1527 (APP:APP 設定) で自動的に JavaDB を起動しますが、JES5 ポータル JavaDB は hostnameIP:1527 (portal:portal 設定) へのバインドを希望します。

このバグは JES 5 ですでに示されている問題 (バグ 6472173) を記述しています。バグ 6472173 の回避方法については、『Sun Java Enterprise System 5 Installation Guide for Microsoft Windows』で説明されています。

解決方法

次のコマンドを使用して Derby データベースを開始します。

```
<JES installation dir>\appserver\bin\asadmin start-database --dbhome <JES installation dir>\portal\data\derby
```

セキュリティ

ここでは、Application Server と Web アプリケーションのセキュリティ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。

SSL 終了が機能しない (**6269102**)

説明

SSL 終了が機能しません。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。

SSL でソケット接続リークが発生する (**6492477**)

説明

JVM バグのため、HTTP リスナで security-enabled を true に設定すると、一部の JDK バージョンでリークの問題が起きます。具体的には、このバグを再現する手順は次のようになります。

1. HTTP リスナで security-enabled を true に設定します。

```
<http-listener acceptor-threads="1" address="0.0.0.0" blocking-enabled="false" default-virtual-server="server" enabled="true" family="inet" id=" http-listener-1" port="8080" security-enabled="true" server-name="" xpowered-by="true">
```

- クイックルックテストの末尾にあるドメイン停止をコメントにします。
- クイックルックテストを実行します。
- ソケット使用状況を確認します。

```
netstat -an | grep 8080
```

使用状況が次のように表示されます。

```
*.8080 *.* 0 0 49152 0 LISTEN *.8080 *.* 0 0 49152 0 BOUND
```

この問題は Glassfish サイトの https://glassfish.dev.java.net/issues/show_bug.cgi?id=849 で追跡されています。

解決方法

最新の JDK バージョンにアップグレードします。

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。

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(6165528)

説明

Enterprise Edition 8 から Application Server Enterprise Edition 8.1 にアップグレードするときに、*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。

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

この問題はさまざまな 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、Red Hat ディストリビューションでも発生しています。

インストールプログラムの最後の画面で「アップグレードツールの起動」ボタンをクリックすると、そのインストールプログラム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用し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。

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用し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-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。

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あと、自己署名付き証明書が信頼されない (6296105)

解決方法

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>
```

アップグレードツールがインスタンス(クラスタのメンバ)の変更済み index.html の内容を上書きする (6386451)

説明

アップグレードツールは、どのサーバーインスタンスのどの既存の index.html ファイルでも上書きします。

解決方法

アップグレードツールを実行する前に既存の index.html ファイルのバックアップを取ってから、後でそれらのファイルを復元します。

(sbs-manual、sbs-installer) 「サーバーインスタンスサーバーに null という名前のシステムコネクタがない」というエラーをスローする (6545145)

説明

Application Server 8.0PE から 9.1 へのアップグレード時に、サーバーに null という名前のシステムコネクタがないというエラーがスローされ、sbs-manual に示されているような無効なユーザー情報が表示されます。ハードコードされた値を変更しても、同じエラーメッセージが表示されます。この問題が起きるのは、8.0 と 9.1 で domain.xml が変更されているためです。

解決方法

8.0 PE から 9.1 にアップグレードした場合にのみ、このバグに遭遇します。回避方法は、8.1、8.2、または 9.0 に一度アップグレードしてから、さらに 9.1 にアップグレードすることです。

異なるビルドの組み合わせを使用すると、アップグレード中に異なるドメインが失われる (6546130)

説明

インプレースアップグレード実行時に、ソースに複数のドメインがある場合、プロセスが中止になっても、インストーラはアップグレードツールを呼び出します。GUI モードで呼び出した場合にそうなります。

解決方法

1. インプレースをCLIモードでインストールし、インストール処理の最後にインストーラがアップグレードツールを選択するプロンプトを表示したときに終了します。この操作を行っても、ドメインディレクトリにあるどのドメインも削除されません。アップグレードツールはbinディレクトリから手動で呼び出す必要があります。
2. GUIモードでインプレースをインストールする場合は、処理中にドメインが失われることがないようにするために、ドメインルートにあるドメインのバックアップを取ります。インストール処理の最後に、インストーラがアップグレードツールを選択するプロンプトを表示したときに終了します。失われているドメインがあれば、ドメインバックアップしたドメインをドメインディレクトリにコピーします。アップグレードツールを起動して、手動でアップグレードを実行します。

Solaris: アップグレードツールの入力済みマスタパスワードに **AS8.2** ではなく **AS9.1** のパスワードが使用される (6565825)

説明

AS8.2から9.1へのアップグレード時に、8.2インストール環境のマスタパスワードが9.1インストール環境に継承されない。これが原因となって、次の管理者ログイン時に認証エラーとなります。

解決方法

Application Server 9.1のデフォルト管理パスワードは `changeit` です。8.2からのアップグレード後に9.1サーバーにログインするときの問題を回避するために、次の3つのどれか1つを行ってください。

- アップグレードを実行する前に、8.2管理者パスワードを `changeit` に変更する。
- アップグレード処理時にデフォルトの管理者パスワードを受け入れず、使用するパスワードを明示的に入力する。
- デフォルトパスワードで9.1にログインし、そのあと直ちにパスワードを変更する。

AS9.1 IFR アップグレードツールが **JES5U1 MQ** ストアを新規 **var home** に移行しない (6573635)

説明

アップグレードツールはどのような形式のデータベースまたはテーブルのアップグレードを扱っておらず、将来サポートする予定もありません。リソース参照設定は転送されているので、Application Serverが元のデータベースおよびテーブルを引き続き処理できるようにする必要があります。データベースを変更するか、またはデータベーステーブルを転送する場合、使用中のデータベースを処理するツールを使用します。

解決方法

次のステップを行って、MQストアを移行します。

注 - AS 8.2 をシャットダウンして、AS 9.1 アップグレードツールを実行したあと、AS 9.1 をはじめて起動する前に次のステップを行います。IFR インストール/アップグレード後にすでに AS 9.1 を開始している場合は、MQ メッセージストアが不安定になる可能性があるため、次のステップを行わないでください。

1. domains/domain1/imq サブディレクトリ全体を AS 8.x domains ディレクトリから AS 9.1 domains ディレクトリにコピーします。
2. ディレクトリとファイルの所有者が Application Server を実行するユーザーと同じであることを確認してください。

上記のステップを実行したあと、Application Server 9.1 を開始することができ、MQ は Application Server 9.1 に格納します。domains ディレクトリ内の MQ ストアは JES5 U1 形式から MQ 4.1 形式に移行されます。この手順では、または AS 9.1 による起動時に MQ 4.1 によっては、AS 8.2 の下にある元の JES5 U1 MQ ストアは保存され変更されないことに注意してください。

JES5 (AS 8.2) を AS 9.1 にアップグレードすると、PS コミュニティーサンプルが壊れて、javax.faces エラーが多数発生する (6576700)

説明

JES5 (Application Server 8.2) から Application Server 9.1 にアップグレードすると、Portal Server コミュニティーサンプルは機能しなくなり、多数の javax.faces.application.ApplicationFactory エラーをスローします。

解決方法

Application Server 8.2 が JES5 Portal Server とともにインストールされている場合、Application Server 8.2 から 9.1 へのアップグレードはサポートされていません。Application Server を 9.1 にアップグレードする前に、Portal Server を Java ES 5 Update 1 にアップグレードする必要があります。

AS91 IFR b58, Linux: インストーラで新規 Java をインストールしたあとに /usr/jdk/entsys-j2se リンクが更新されない (6591697)

説明

Linux プラットフォーム上で IFR インストーラを使用し「JDKをインストール」オプションを選択して Application Server 8.2 から 9.1 にアップグレードすると、インストーラは正常に完了しても、ほとんどの JES コンポーネントが動作を停止します。

解決方法

この問題は、Linux プラットフォームでの Application Server 9.1 の IFR インストールにのみ、また「JDKをインストール」オプションを選択した場合にのみ影響が出ます。この問題を回避するには、インストール直後に、`/usr/jdk/entsys-j2se` を `/usr/java/jdk1.5.0_12` ディレクトリに手動でリンクします。

AS9.1 IFR Win インプレースバックアップは `asupdate.bat` フォーム値検査と統合されない。パスの誤り (6596412)

説明

Windows 上で Application Server 9.1 IFR アップグレードを実行すると、インプレースバックアップは `asupdate.bat` フォーム値と正しく統合されません。具体的には、`ASupdate.bat` GUI 画面に間違った情報を入力してから「次へ」をクリックすると、アップグレードインストーラはインプレースアップグレードなのかどうかを検出しようとしています。インプレースアップグレードである場合には、アップグレードの前に `domain1` がバックアップディレクトリに移動されます。アップグレードが進行すると、間違った情報の結果としてエラーメッセージが表示されます。直ちにエラーを訂正しようとする、`domain1` がすでに移動しているので、パスエラーがスローされます。

解決方法

ソースディレクトリを `{current source path}/backup` の `domain1_{timestamp}` ディレクトリに変更するか、または「取消し」ボタンでインストーラを終了して再起動します。

AS9.1 IFR Windows、インラインアップグレードで、`asupgrade.bat` を短いパスと長いパスで使用すると、**Domain1** が削除される (6596394)

説明

(Windows のみ) 以前のバージョンの Application Server が特殊文字または DOS 形式のショートネームを使用してプログラムディレクトリパスにインストールされている場合、同じディレクトリパス名を使用すると、Application Server 9.1 へのインプレースアップグレードが失敗します。

たとえば、Application Server 8.2 が次のディレクトリのどちらかにインストールされているとします。

```
C:\Program Files (x86)\dirs\appserver c:\progra~2\dirs\appserver
```

9.1 へのインプレースアップグレードを実行しようとする、インストーラがショートネームまたは特殊文字を必須のロングネーム形式に変換できないために失敗します。

解決方法

特殊文字や DOS 形式のショートネーム短縮形 (progra-2 など) は、以後のアップグレードインストールの妨げになるため、使用しないことが強く勧められています。このようなインストール環境が存在する場合、アップグレードする前に長いパス名を使用してそれを削除するか、または全く新しいディレクトリに新規バージョンの Application Server をインストールします。

AS 9.1 IFR: Web UI 認証が機能しない (6569813)

説明

Application Server アップグレード後に、<jsp:forward> タグが Authenticate.jsp で予期されているように機能しません。<jsp:forward> を呼び出すとサーバーログにエラーが生成され、Web UI に空白ページが表示されます。問題は、Authenticate.jsp 内の <jsp:forward> に <jsp:forward page="{redirectPage}"/> のようなページ属性が必要なのに、渡される値が /registry/thin/{pagename}.jsp のような相対パスであるため、この場合は Authenticate.jsp が純粋な JSP ページであっても機能しません。

解決方法

Application Server アップグレードを完了したあと、asadmin ツールを使用して次のコマンドを実行し、domain.xml に <auth-realm> を設定します。

1. <appserver9.1-install-dir>/bin に移動して、次のコマンドを実行します。

```
./asadmin delete-auth-realm --host localhost --port 6489 certificate
```

これにより、古い auth-realm 証明書があれば削除されます。

2. 次のコマンドを実行します。

```
./asadmin create-auth-realm --terse=false --echo=true --interactive=true \ --user admin --host localhost --port 6489 --classname \ com.sun.enterprise.security.auth.realm.certificate.CertificateRealm \ --property assign-groups=have.client.cert certificate
```

これにより、新規の <auth-realm> が assign-groups プロパティを指定して作成されます。

3. Application Server registry ドメインを停止して再起動します。

asupgrade GUI のローカライズされたオンラインヘルプが存在しない (6610170)

説明

英語以外の言語で asupgrade GUI を実行すると、この GUI のオンラインヘルプが選択した英語以外の言語にローカライズされていません。

解決方法

現時点ではありません。オンラインヘルプは英語以外のすべての目標言語にローカライズされる予定です。

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、アプリケーションの **JAR** ファイルがロックされ、そのあとの配備取り消しや再配備に失敗する (**5004315**)

説明

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、期待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、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` ユーティリティー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とするとう失敗します。

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とする、`asadmin` ユーティリティー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、そのあと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` を使用して **WAR** を配備できない (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 (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せん。

```
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(Default
CompilerAdapter.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 (6188932)

説明

Sun Java System Application Server 9.1 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、`ProductName;9.1` 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 サーバード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 サーバード経由ではなく、要求を直接受信したかのように見えます。

Application Server 9.1 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 9.1 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 9.1 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 サーバード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

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。

Web サーバー

AS 9.1 b50e: Linux: AS LB インストール後に WS インストールを開始できない: libjvm.so: cannot open shared エラーになる (6572654)

説明

この問題は、Linux システムで Sun Java System Web Server を Application Server 9.1 およびロードバランサとともに使用している場合にのみ起きます。このような場合、Application Server とロードバランサのインストール後に、libicui18n.so.2 と libicuuc.so.2 が競合するため Web Server が起動に失敗することがあります。これらのライブラリは /opt/sun/private/lib と /opt/sun/appserver/lib の両方にあります。

解決方法

使用すべき正しいライブラリは /opt/sun/appserver/lib にあるライブラリです。このライブラリに対して lbplugin が構築されているからです。/opt/sun/private/lib からこの 2 つのライブラリを削除すると、Web Server はエラーを出さずに起動します。

または、/opt/sun/private/lib からライブラリを削除したくない場合は、Web Server startserv スクリプトの LD_LIBRARY_PATH で /opt/sun/private/lib の前に /opt/sun/appserver/lib を配置することもできます。つまり、

```
# Add instance-specific information to LD_LIBRARY_PATH for Solaris and Linux LD_LIBRARY_PATH="${SERVER_LIB_PATH}:${SERVER_JVM_LIBPATH}:${LD_LIBRARY_PATH}:/opt/sun/appserver/lib:/opt/sun/appserver/lbplugin/lib";
export LD_LIBRARY_PATH
```

これを、次の項目で置き換えます。

```
# Add instance-specific information to LD_LIBRARY_PATH for Solaris and Linux LD_LIBRARY_PATH="/opt/sun/appserver/lib:/opt/sun/appserver/lbplugin/lib: ${SERVER_LIB_PATH}:${SERVER_JVM_LIBPATH}:${LD_LIBRARY_PATH}"; export LD_LIBRARY_PATH
```

Web サービス

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。

Ant タスク `wsimport` が **Java EE SDK b33d** (**JDK 1.6** を使用) で `NoClassDefFoundError` を出して失敗する(**6527842**)

説明

Java EE SDK b33d に組み込まれている JDK 1.6 で JAX-WS テストを実行中に問題に遭遇します。テストは次のメッセージを出して直ちに異常終了します。

```
[wsimport] Exception in thread "main" java.lang.NoClassDefFoundError: \ com/sun/tools/ws/WsImport
```

`webservicetools.jar` に `com/sun/tools/ws/WsImport.class`、`com/sun/tools/ws/ant/WsImport.class`、および `com/sun/tools/ws/ant/WsImport2.class` が含まれていても、このエラーは起きます。さらに、1.5.0-10 JDK を使用すると、同じテスト作業空間が問題なく機能します。

解決方法

JAX-WS テストを実行する前に、`webservicetools.jar` を `$JAVA_HOME/jre/lib/endorsed` にコピーします。

IFR EE ビルドで `publish-to-registry` コマンドが失敗する (**6602046**)

説明

JAXR は SAAJ を使用して SOAP メッセージをレジストリに送信します。IFR 以外では、SAAJ impl クラスは `lib/webservicetools.jar` の下にあります。IFR の場合、SAAJ クラスは引き続き `lib/webservicetools.jar` の下にあります。また、`saaj-impl.jar` は `/usr/share/lib` ディレクトリにあります。この jar ファイルは Application Server で格上げされ、`webservicetools.jar` のクラスよりも優先されています。この jar ファイルには、SOAP メッセージを Web サービスレジストリに送信するために必要なセキュリティ権限がありません。`/usr/share/lib` ディレクトリの下に jar に権限を付与するか、または `/usr/share/lib` の jar に依存しないように、パッケージを変更する必要があります。

解決方法

`server.policy` ファイルに次の内容を追加します。

```
grant codeBase "file:/usr/share/lib/saaj-impl.jar" { permission java.security.AllPermission; };
```