



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



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

Part No: 820-1802
2007年2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を続けられるスケーラブルなサービスなので、パフォーマンスの改善、クラスタリング、および高可用性を実現する機能を提供します。

- 3 ページの「このリリースノートについて」
- 4 ページの「アクセシビリティ機能」
- 4 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「コメントの送付先」
- 7 ページの「補足情報」

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 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 \(http://docs.sun.com/db/prod/slappsrv#hic/\)](http://docs.sun.com/db/prod/slappsrv#hic/) 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。

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購入するには、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。

アクセシビリティに対する Sun の取り組みについては、<http://sun.com/access> を参照してください。

関連マニュアル

Application Server Enterprise Edition 8.2 製品には、<http://docs.sun.com/app/docs/prod/sjs.asse#hic> に掲載されているマニュアルセット全体が含まれています。

次の表に、Application Server Enterprise Edition 8.2 のマニュアルセットに含まれるものを要約して示します。

表 1-1 マニュアルセットの内容

マニュアル名	説明
Sun Java System Application Server Enterprise Edition 8.2 Documentation Center	1 か所で Application Server のすべてのトピックにアクセスできる場所。
『Sun Java System Application Server Enterprise Edition 8.2 Quick Start Guide』	Sun Java System Application Server 製品を使い始めるための入門ガイド。

表 1-1 マニュアルセットの内容 (続き)	
マニュアル名	説明
『Sun Java System Application Server Enterprise Edition 8.2 Installation Guide』 ¹	Sun Java System Application Server ソフトウェアとそのコンポーネントのインストール。
『Sun Java System Application Server Enterprise Edition 8.2 配備計画ガイド』	最適な方法で確実に Sun Java System 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。Application Server を配備する場合に注意する必要がある一般的な問題と懸案事項についても解説します。
『Sun Java System Application Server Enterprise Edition 8.2 Developer's Guide』	Sun Java System Application Server 上で動作することを目的とし、J2EE コンポーネントおよび API のオープン Java スタンドモデルに準拠した、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などについての一般的な情報も提供します。
『Sun Java System Application Server Enterprise Edition 8.2 J2EE Tutorial』	J2EE アプリケーションを開発するための J2EE 1.4 プラットフォームテクノロジおよび API の使用と、それらアプリケーションの Sun Java System Application Server への配備。
『Sun Java System Application Server Enterprise Edition 8.2 管理ガイド』	Sun Java System Application Server サブシステムおよびコンポーネントの、管理コンソールからの設定、管理、および配備。
『Sun Java System Application Server Enterprise Edition 8.2 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。
『Sun Java System Application Server Enterprise Edition 8.2 Administration Reference』	Sun Java System Application Server 設定ファイル domain.xml の編集。
『Sun Java System Application Server Enterprise Edition 8.2 アップグレードと移行』	新しい Sun Java System 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 および互換バージョンおよび 7 からの移行)。このマニュアルでは、隣接する製品リリース間の相違点と、製品仕様との不一致を招く可能性のある設定オプションについても説明します。
『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』	パフォーマンスを改善するための Sun Java System Application Server のチューニング。
『Sun Java System Application Server Enterprise Edition 8.2 トラブルシューティングガイド』	Sun Java System Application Server におけるトラブルの解決。
『Sun Java System Application Server Enterprise Edition 8.2 Error Message Reference』	Sun Java System Application Server のエラーメッセージの解決。

表 1-1 マニュアルセットの内容 (続き)

マニュアル名	説明
『Sun Java System Application Server Enterprise Edition 8.2 Reference Manual』	Sun Java System 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。
1 『Sun Java System Application Server Enterprise Edition 8.2 Installation Guide』は、Application Server のスタンドアロンインストールのために用意されています。	

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)
(<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)
(<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/servlet/SessionServlet?url=/developer/bugParade/index.jshtml) (<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、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。
- 保守契約先に電話連絡してください。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

コメントの送付先

Sun では、マニュアルの改善に努めており、お客様のご意見、ご提案をお待ちしております。

<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。オンラインフォームでは、マニュアルのタイトルと Part No. をご記入ください。Part No. は、7桁か9桁の番号で、マニュアルのタイトルページまたは最初のページに記載されています。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 Enterprise Edition 8.2 リリースノート』、Part No. は 820-1802 です。

補足情報

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)
- Application Server の製品マニュアル (<http://docs.sun.com/db/prod/slappsrv#hic/>)
- 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 Enterprise Edition 8.2 について

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。

この章では、次の内容について説明します。

- 9 ページの「8.2 Release 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」
- 17 ページの「Enterprise Edition 8.2 リリースで修正されたバグ」
- 19 ページの「HADB に関する追加情報」
- 22 ページの「Application Server 製品リリース」
- 22 ページの「互換性について」
- 24 ページの「J2EE のサポート」
- 25 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」
- 26 ページの「高パフォーマンス」
- 26 ページの「スケーラビリティ」
- 26 ページの「JavaServer Faces 1.1 のサポート」

8.2 Release 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、セキュリティー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うことができます。また、セキュリティー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のメッセージブローカが含まれています。

- **Message Queue 3.7 UR 1** – Application Server に MQ 3.7 UR 1 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適合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 8.1 PE または 8.1 EE 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 データベースを使用しますが、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 – Application Server には、Sun の JDBC ドライバが付属していません。
- **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 4.4.2-7) 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、cclusql (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。
- 動的コンテンツテクノロジーのサポートの中止 - CGI 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「重要パッチ情報」
- 12 ページの「JDBC ドライバとデータベース」
- 13 ページの「バンドルされている Java DB データベースの使用」
- 15 ページの「Web サーバー」
- 15 ページの「ブラウザ」
- 15 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」

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ー要件も示します。

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
Windows 2000 Advanced Server SP4	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 1.4.2_10、J2SE 5_06
Windows XP SP2					
Windows 2003 Enterprise Server SP1 (32 bit)					
Windows 2003 Enterprise Server SP1 (64 bit)					

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 - 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

現時点で Sun Java System Application Server Enterprise Edition 8.2 に必要なパッチについては、<http://sunsolve.sun.com> にアクセスして「app server 8.1 patch」を検索してください。Sun Java System Application Server Enterprise Edition 8.2 のリンクをたどります。オペレーティングシステムのパッチ要件に変更があり、Java Enterprise System コンポーネントのパッチが利用可能になった場合、更新は SunSolveSM Web サイトから入手できます。最初は、推奨パッチクラスタの形式で提供されます。

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 互換 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® 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

表 2-2 J2EE 互換 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
IBM	Type 2	IBM DB2 8.1 Service Pack 3+
Java DB	Type 4	Apache Derby 10.1.2.1
PointBase	Type 4	PointBase Network Server 5.2
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。

- 13 ページの「Java DB データベースの起動と停止」
- 14 ページの「Java DB ユーティリティースクリプト」

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 (省略可能) 次のプロパティも指定できます。
 - `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。
 - `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。

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	7	Windows 2000 Advanced Server SP4 Windows XP SP2 Windows 2003 Enterprise Server SP1 (32-bit) Windows 2003 Enterprise Server SP1 (64-bit)

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、***
Internet Explorer	5.5 Service Pack 2、6.0
Firefox	***

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。

- 16 ページの「サポートされるプラットフォーム」

- 16 ページの「HADB サーバーホストの要件」
- 16 ページの「HADB 管理ホストの要件」
- 16 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、システムは JDK 1.5 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

- **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。このほかに、データデバイス用にもディスク容量が必要で、テストインストール用にノードあたり 512M バイトが必要です。
- 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。このほかに、データデバイス用にもディスク容量が必要で、テストインストール用にノードあたり 1200M バイトが必要です。

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

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
4887079	配備と配備の取り消し、および配備されているアプリケーションの検索を実行するためのプログラミング API。
4911462	ポートが使用可能な範囲を外れているときの正しくないメッセージ。
4918535	<code>sun-appserv-deploy()</code> が、 <code>createAndDropTables()</code> をサポートするフラグを取らない。
4939749	<code>xml:()lang()</code> 値は、Deploytoolによって自動的に挿入されるべきではない。
4946914	クラスタに対する配備サポート。
4979136	ディレクトリベースの配備によって、アプリケーションがバックアップディレクトリにコピーされる。
4987274	Beanのリモートインタフェースに <code>Util()</code> という名前が付いていると、配備が失敗する。
4988818	J2SE 1.5 が使用されていると、透過的持続性ランタイムテストが失敗する。
4992295	コマンド行インタフェースでのシステムコンポーネントの配備は成功するが、サーバーのログファイルにエラーが記録される。
4994790	<code>precompilejsp=true</code> を使用して配備された JSP が、 <code>sun-web.xml</code> でコンパイラフラグを使用しない。
4996876	ベリファイアと、 <code>verify=true</code> による配備のレポートが異なる。
5003356	最新の <code>server.policy</code> ファイルアップデートが Upgradetoolによって処理されない。
5006854	<code>asadmin deploy --virtualservers</code> が配備に失敗する。
5007309	HTTP リスナーアクセプタスレッドに対する不適切なデフォルト値。
5008941	アプリケーションが配備取り消しの後で再度配備されたとき、JSR88の起動操作が失敗する。
5016848	Windows で、JDK JAR ファイルキャッシュと開いているファイルによって、一部の再配備が妨げられる。
5017956	JAR モジュールレベルでの <code>list -m</code> によって EJB が表示されない。
5030425	<code>deploydir</code> コマンドで <code>security-role-mapping</code> の変更が無視される。

バグ番号	説明
5041343	servlet-mapping url-pattern- -directory が常に / で囲まれているかどうかチェックされない。
5046120	大規模アプリケーションの配備中に SEVERE ログメッセージが出力される。
6041268	HTTP TRACE を無効にする機構がない。
6062410	地域対応化されたマシンで、Upgradetool が英語で起動される。
6067341	Web アプリケーションで ejb-refs を使用して deploydir コマンドを実行すると、リモートインタフェースの rmic で失敗する。
6152752	SPEC J2004 テストの実行中に outofbound 例外がログに記録される。
6154949	接続検証が機能しない。
6157310	関係管理中に、ランタイムで「Collection」フィールドが再読み込みされる。
6165491	ドメインがデフォルトドメインとは別のパスに作成されていると、ドメインの起動に失敗する。
6171667	ライフサイクルモジュールのプロパティ要素が domain.xml に作成されない。
6171729	MDB の配備中に、文字列でない RA ActivationSpec プロパティで IllegalArgumentException が発生する。
6172178	OSS/JTT TCK で、リモートアプリケーションサーバーからの JMS 接続ファクトリの取得に失敗する。
6172589	セキュリティーマネージャーの呼び出しの最適化。
6183492	[DataDirect] DB2: 一部の透過的持続性アプリケーションサーバーテストで、EJB の呼び出し中に例外がスローされて失敗する。
6184864	OR 演算子を使用した EJBQL クエリーで結果が返されない。式の単一値 CMR は null となる。
6197393	Deploytool で、配備記述子に message-destination 要素が作成されないことがよくある。
6198796	アプリケーションを配備する場合、EE サンプルの asadmin コマンドに availabilityenabled=true() オプションを含める必要がある。
6198981	クラスパスに xalan.jar ファイルがないと、ドロップダウンが空になり、Web サービスウィザードが失敗する。
6199076	asant スクリプトを使用して、Duke 書店サンプルのフェイルオーバーテストを実行できない。

バグ番号	説明
6202363	mq-failover サンプルアプリケーション内の1つのantターゲットで、クラスタ名がハードコードされている。
6202606	JMSとMessage Queueの間のSSL JMSにJMSサービス設定を使用できない。
6206176	Application Server 8.1で、startserv および stopserv にアクセス権 755が必要である。
6207297	デフォルトのSSLポート番号(443)を使用しないでApplication Serverにアクセスすると動作しない。
6207862	asadmin create-domain --help によって生成された一部が正しくない。
6363339	managementws サンプルで、MANIFEST.MFの参照をcastor-0.9.3.9-xml.jarからcastor-0.9.9.1.jarに更新する必要がある。
6372759	AS v8.2 Startupによって、特定のJavaシステムプロパティが適切に処理されない。

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。

- [19 ページの「HADBの拡張機能」](#)
- [20 ページの「既知のSQLの制限事項」](#)
- [21 ページの「高可用性負荷分散」](#)

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、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ありません。
- 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、パスワードをコマンド行オプションとして受け入れるすべての `hadbm` コマンドに適用されます。 `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

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、パスワードファイルまたは対話型出力を使用します。これは、コマンド行のパスワードオプションを受け入れるすべての `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。

既知のSQLの制限事項

- テーブルで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。
- すべてのテーブルは主キーを指定して作成する必要があります。すなわち、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。
- リソースに新規所有者を割り当てることができます。ただし、この変更を行う場合、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 NOT EXISTS サブクエリーが入れ子にされていて、各サブクエリーがクエリーの外部レベルと直接関連していないクエリーはサポートされません。
- 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

高可用性負荷分散

Application Server には、以下のためのロードバランサ機能が組み込まれています。

- HTTP、IIOP、および JMS クライアント
- HTTP セッションのフェイルオーバーサポート
- EJB クラスタリングおよびフェイルオーバーサポート
- 高可用性 EJB タイマー
- 分散トランザクションの回復
- 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
- J2EE アプリケーションの一時的な状態を格納する高可用性データベース

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。

Application Server 製品リリース

Application Server 製品は、さまざまな方法で提供されます。次の表に、製品リリース別の配布方法を示します。

Application Server 製品リリース	配布方法
Sun Java Enterprise System 内の Application Server Enterprise Edition コンポーネント	ファイルベースの配布 SunSolve 経由でパッチのインストールが必要
Application Server Standard および Enterprise Edition スタンドアロン製品	ファイルベースおよびパッケージベースの配布

互換性について

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。この詳細および移行の計画については、<http://www.netbeans.org/kb/41/j2ee-tut/index.html> 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 to してください。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	
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。添付ファイル付きの SOAP メッセージの作成を可能にします。
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規約の定義

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

API	説明
Java API for XML Processing (JAXP) 1.2	XML ドキュメントの解析および変換を可能にする API。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によるメッセージの作成、送信、受信、読み取りを可能にするメッセージング標準。キューおよびトピック用の統一された 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。API へのマイナー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 では、基盤となる JVM として J2SE 5.0 または互換バージョンが必要です。Java のバージョンを切り替える場合は、Windows OS と UNIX OS の両方で次の手順を実行します。

▼ 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、**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。
`ApplicationServer-base\bin\asadmin.bat 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、新しい **J2SE** の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。

```
ApplicationServer-base\bin\asadmin.bat 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負荷分散の要求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大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/IOP 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 テクノロジーをサポートしています。JavaServer Faces テクノロジーは、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 テクノロジーで開発している間、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。

- 27 ページの「管理」
- 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」
- 36 ページの「コネクタ」
- 37 ページの「関連文書」
- 39 ページの「高可用性」
- 43 ページの「インストール」
- 46 ページの「J2EE Tutorial」
- 47 ページの「ライフサイクル管理」
- 48 ページの「ロギング」
- 48 ページの「メッセージキュー」
- 49 ページの「監視」
- 50 ページの「サンプル」
- 52 ページの「セキュリティ」
- 52 ページの「アップグレードユーティリティ」
- 53 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。

Application Server の「インストール時に自動的に設定」オプションでロードバランサ機能がサポートされていない (6463858)

Application Server の「インストール時に自動的に設定」オプションでは、ロードバランサ機能はサポートされていません。

対処方法:ロードバランサ機能は Application Server のインストール後に設定できません。

注 - ロードバランサ機能を設定するには、Application Server と Web Server をシステムにインストールしておく必要があります。

ロードバランサ機能を設定するには、次の手順を実行します。

1. レジストリ HKEY_LOCAL_MACHINE -> Sun Microsystems -> EntSys -> Installer -> Application Server で、IS_LB の値を true、Cfgr_LB の値を false に設定します。
2. setup ディレクトリに移動します。
cd JavaES-Install-Dir\setup\
3. ASConfigure.bat バッチファイルを実行します。
4. 指示に従って、適切な値を指定します。

注 - AS_LB プラグインの場合、Sun Java System Web Server [必須] と入力します。これが Java ES 5 でサポートされている唯一のプラグインであるためです。

5. システムをリブートします。

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

デフォルトでは、asenv.conf によって参照される domain1 の AS_ACC_CONFIG 変数のハードコードされた値が `JavaES-Install-Dir\lib\lib\package-appclient.xml` にあります。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、*JavaES-Install-Dir\lib\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の手順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)

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 インストールで使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。

Java ES 2 Application Server 7 と比べると、Java ES 3 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 および互換バージョンと比較して、Application Server 8.2 では asadmin コマンドにいくつかの変更が加えられました。たとえば、Application Server 7 および互換バージョンでは、サーバーインスタンスを起動するコマンドは次のようになります。

```
asadmin start-instance
```

バージョン 8.2 では、次のコマンドを使用します。

```
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)

Java ES 2 Application Server 7 および互換バージョンから Java ES 5 Application Server 8.2 にアップグレードする場合、デフォルトポートが変更されているために、非互換性に関する問題またはエラーが発生することがあります。

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)

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るにもかかわらず、元の名前ではない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)

J2SE 1.4.x、5.0 および互換バージョンを 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。

以下に値の例を示します。

```
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。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が原因です。

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。

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)

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)

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するには、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、変更時には注意が必要です。`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)

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 `asant` を `asant.bak` に変更します。
2. `as_install/lib/install/templates/ee` にある SE/EE バージョンの `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` ファイルを編集して、`%CONFIG_HOME%` トークンを `as_install/config` 値に置き換えます。
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` ファイルに結合します。

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)

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 home ディレクトリに `.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ります。ことに注意してください。

create-domain 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)

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)

セキュリティ保護された http-listener を作成し、lbplugin をインストールしたあとで、`webserver_instance_dir/config` ディレクトリの下での `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。

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)

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

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)

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

JSP および Web サービス技術を代わりに使用します。

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。

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がハングアップする (ID 6165970)

この問題は、2つの並行トランザクションが実行中で、そのうちの1つがロールバックされた場合、準備済みの更新文を使ったときに発生することがあります。

解決方法

接続の遮断レベルを設定するには、対応する接続プールを同じ遮断レベルで作成します。接続プールの設定の詳細については、『Sun Java System Application Server Enterprise Edition 8.2 管理ガイド』を参照してください。

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。

説明 2

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。たとえば、*4096* を使用します。

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

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。

コネクタ

この節では、J2EE のコネクタアーキテクチャー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)

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ます。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)

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。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。

関連文書

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 に不整合がある

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。
- `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。
- 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。
- 定数 `XTypes.CONNNECTOR_CONNECTION_POOL_MONITOR` のスペルが間違っている ("NNN" の部分)。

付属の **ANT** によって

`java.lang.NoClassDefFoundError` 例外がスローされる (ID 6265624)

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 外部のタスクの実行に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、「HTTP File Cache」 in 『Sun Java System Application Server Enterprise Edition 8.2 Performance Tuning Guide』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。

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。

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)

`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、データベースを作成したり、ノードを追加したりする際にはリソースが利用可能かどうかをチェックします。しかし、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)

同一のソフトウェアパッケージを、異なるホストの別の場所に同じ名前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)

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。このクリーンアップは、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)

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と、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。

clu_trans_srv を中断できない (ID 6249685)

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 コマンドが機能せず、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)

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。

セッションオブジェクトがタイムアウトし、管理エージェントで削除されたとき、hadbm/ma は不正なエラーメッセージを出力する (ID 6275103)

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。

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)

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)

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断し、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場所で頻繁に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)

マシンに負荷がかかっていると、マスク機構が機能せず、入力したパスワードの一部が表示されることがあります。パスワードが表示されることにより、軽度のセキュリティリスクが生じます。パスワードは常にマスクされる必要があります。

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 (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。

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。

Java Enterprise System 5 インストーラによる Application Server 8.x ロードバランサの最小インストールが適切でない (ID 6478047)

Apache および IIS は Java ES 5 インストーラでは設定できません。Apache および IIS は Windows プラットフォーム上で手動で設定する必要があります。

解決方法

ロードバランサ Apache または IIS を設定するには、次の手順を実行します。

Apache 2.x を設定するには

1. Apache 2.x をインストールします。
Apache は APDIR=C:\Apache2\Apache2 ディレクトリにインストールされます。
2. JES5 を最小インストールでインストールします。
ロードバランサ以外のすべてのコンポーネントを選択解除します。Java ES 5 は JES5DIR=C:\Program Files\Sun\JavaES5 ディレクトリにインストールされます。
3. Apache2 ディレクトリに resource および errorpages ディレクトリを作成します。

```
mkdir %APDIR%\modules\resource  
mkdir %APDIR%\modules\errorpages
```
4. リソースファイルを resource ディレクトリにコピーします。

```
cd %APDIR%\modules\resource  
copy %JES5DIR%\appserver\lib\webserver-plugin\windows\apache2\LBPlugin*.res  
.
```
5. ロードバランサ DLL を modules ディレクトリにコピーします。

```
cd %APDIR%\modules  
copy %JES5DIR%\appserver\lib\webserver-plugin\windows\apache2\mod_loadbalancer.dll .
```
6. テンプレート errorpages を errorpages ディレクトリにコピーします。

```
cd %APDIR%\modules\errprpages  
copy %JES5DIR%\appserver\lib\webserver-plugin\windows\iws\errorpages .
```
7. ロードバランサテンプレートとその他の DTD を Apache config ディレクトリにコピーします。

```
cd %APDIR%\config
copy %JES5DIR%\appserver\lib\install\templates\loadbalancer.xml.template .
copy %JES5DIR%\appserver\lib\dtds\sun-loadbalancer* .
```

8. httpd.conf ファイルのバックアップを作成します。

```
cd %APDIR%\config
copy httpd.conf httpd.conf.orig
```

9. httpd.conf ファイルを編集します。

httpd.conf ファイルに次の行を追加します。

```
##BEGIN EE LB Plugin Parameters
LoadModule apachelbplugin_module modules/mod_loadbalancer.dll
<IfModule mod_apache2lbplugin.cpp>
    config-file "C:\Apache2\Apache2\conf\loadbalancer.xml"
    locale en
</IfModule>
<VirtualHost 10.12.8.107>
DocumentRoot "C:\Apache2\Apache2\htdocs"
ServerName vm07
</VirtualHost>
##END EE LB Plugin Parameters
```

10. C:\Apache2\Apache2 を実際の %APDIR% ディレクトリに置き換えます。

IP、ServerName、および DocumentRoot ディレクトリも置き換えます。

11. %APDIR% に新しい sec_db_files ディレクトリを作成します。

```
cd %APDIR%
mkdir sec_db_files
```

12. NSS キーストアを %APDIR%\sec_db_files ディレクトリにコピーします。

```
cd %APDIR%\sec_db_files
copy %JES5DIR%\appserver\lib\webserver-plugin\windows\iis\*.db .
```

13. 必要なライブラリが含まれるように PATH を設定します。

以下の追加パスを先頭に付加します。

```
PATH %JES5DIR%\share\lib;%JES5DIR%\appserver\lib;%JES5DIR%\appserver\bin
```

14. %JES5DIR% を実際の Java ES 5 ディレクトリに置き換えます。

15. システム環境変数に、値 1 の NSPR_NATIVE_THREADS_ONLY 変数を追加します。

16. リポートして Apache 2 をテストします (loadbalancer.xml の設定後)。

IIS LBPlugin を設定するには

1. c:\inetpub\wwwroot ディレクトリに sun-passthrough ディレクトリを作成します。

```
cd c:\inetpub\wwwroot  
mkdir sun-passthrough
```
2. c:\inetpub\wwwroot\sun-passthrough ディレクトリに errorpages、resource および sec_db_files ディレクトリを作成します。

```
cd c:\inetpub\wwwroot\sun-passthrough  
mkdir errorpages  
mkdir resources  
mkdir sec_db_files
```
3. DLL ファイルを sun-passthrough ディレクトリにコピーします。

```
copy <as_install_dir>/appserver/lib/webserver-plugin/iis/*.dll  
c:\inetpub\wwwroot\sun-passthrough\
```
4. DTD を sun-passthrough ディレクトリにコピーします。

```
copy <as_install_dir>/appserver/lib/dtds/sun-loadbalancer*.dtd  
c:\inetpub\wwwroot\sun-passthrough\
```
5. sun-passthrough.properties ファイルを sun-passthrough ディレクトリにコピーします。

```
copy <as_install_dir>/appserver/lib/webserver-plugin/iis  
c:\inetpub\wwwroot\sun-passthrough\
```
6. セキュリティー DB ファイルを sun-passthrough ディレクトリにコピーします。

```
copy <as_install_dir>/appserver/lib/webserver-plugin/iis/*.db  
c:\inetpub\wwwroot\sun-passthrough\sec_db_files\
```
7. リソースファイルを sun-passthrough ディレクトリにコピーします。

```
copy <as_install_dir>/appserver/lib/webserver-plugin/iws/*.res  
c:\inetpub\wwwroot\sun-passthrough\resource\
```
8. エラーページを sun-passthrough ディレクトリにコピーします。

```
copy <as_install_dir>/appserver/lib/webserver-plugin/iws/errorpages/*.html  
c:\inetpub\wwwroot\sun-passthrough\errorpages\
```
9. loadbalancer.xml.example テンプレートを sun-passthrough ディレクトリにコピーします。

```
copy  
<as_install_dir>/appserver/lib/install/templates/loadbalancer.xml.example  
c:\inetpub\wwwroot\sun-passthrough\
```
10. sun-passthrough.properties ファイルを編集します。

```
##BEGIN EE LB Plugin Parameters
log-file = C:\InetPub\wwwroot\sun-passthrough\lb.log
### The valid options for different logging levels are FATAL, SEVERE, WARNING, INFO and FINE.
log-level = INFO
lb-config-file = C:\InetPub\wwwroot\sun-passthrough\loadbalancer.xml
##END EE LB Plugin Parameters
```

注 - IIS6 を構成している場合、必ず AS82 マニュアルの説明に従って権限を設定し追加手順を実行してください。また、IIS6 遮断モードを IIS5 互換に設定する必要があります。

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)

Windows プラットフォームでは、Application Server Enterprise Edition をインストールした直後に、Message Queue ブローカが起動に失敗します。ディレクトリ `drive:\as\domains\domain1\imq` が存在しないことを示すメッセージが表示されます。

`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
```

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 プロパティを変更するとエラーが発生する (ID 6193449)

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を使っ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**)

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

ありません。このフラグは設定しないでください。

Java ES 3 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、バージョン7および互換バージョンと比べて、デフォルトのログの場所やサーバーインスタンスの場所が変更されています。

詳細については、『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**)

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)

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。

Application Server への監視フレームワーク統合 (6469302)

Application Server のベータリリースにおいては、デフォルトでは監視フレームワークはサポートされていません。

解決方法

監視フレームワークを Application Server に統合するには、次の手順を実行します。

1. <Install_dir>\appserver\lib\install\templates\ee\com.sun.cmm.as.xml ファイルを編集します。
\${InstalledDate} を Application Server のインストール場所で更新して、
\${InstalledDate} を現在日付にします。
2. <Install_dir>\appserver\lib\install\templates\ee\com.sun.cmm.as.xml ファイルを <Install_dir>\appserver\lib にコピーします。

3. <MFWK_Install_location>\bin\mfwksetup.bat -r
<Install_dir>\appserver\lib\com.sun.cmm.as.xml コマンドを実行します。

注 - \${InstalledLocation} の値は Application Server インストールの場所、
c:\Sun\JavaES5\appserver です。\$InstalledDate にはミリ秒で時間を入力する必要があります
(1970 年から現在時間 (ミリ秒) を計算)。

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 がハングアップする (ID 6195092)

Windows プラットフォームでは、mqfailover コマンドが表示されたら Ctrl+C を押し
てハングアップしたプロセスを終了させて、setup-one-machine-cluster プロセスを
再実行する必要があります。

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
```

ほかの Enterprise Edition サンプルで `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` を実行します。そうでない場合には `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

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 で終了してから再実行することです。

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 (ID 6198003)

asadmin の配備手順を終了して、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後、次のエラーメッセージが表示されます。

```
/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が、マニュアルに明記されていません。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も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。

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。

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)

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

- 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

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)

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。

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)

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 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。

管理ユーザー名またはパスワードでセミicolon (;) 文字を無効化できない (ID 6473341)

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。

ロードバランサプラグインで **Apache** および **IIS** がサポートされていない

Sun Java ES 5 Application Server ではロードバランサプラグインで Apache および IIS (Sun 以外の Web コンテナ) がサポートされていません。Sun Java ES ではロードバランサプラグインの設定用に Sun Java System Web Server がインストールされます。

--precompilejsp=true を使用してアプリケーションを配備すると **JAR** ファイルがロックされることがある (ID 5004315)

Windows プラットフォームで Microsoft Windows 上の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。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、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が Microsoft Windows によって許可されません。

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)

`web.xml` ファイルのオプションの `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>` 要素は、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` ファイル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)

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

```
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。

次のいずれかの方法でこの設定を行うことができます。

- 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)

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、次の制約のある 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。

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必要があります。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)

--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。