

# Sun Java™ System Application Server Enterprise Edition リリースノート

バージョン 8.1 2005Q1

Part No. 819-2026

---

Sun Java™ System Application Server Enterprise Edition 8.1 2005Q1 製品は、Web サービスアプリケーションの構築と管理を容易にします。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、優れたパフォーマンス、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。Application Server を使って 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。

このリリースノートには、Sun Java System Application Server 8.1 2005Q1 のリリースの時点での重要な情報を記載しています。新たな機能と拡張機能、既知の問題と制限事項、およびその他の情報を扱っています。Application Server 8.1 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版は、Sun Java System のマニュアルサイト

<http://docs.sun.com/app/docs/prod/entsys?l=ja> 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。

このリリースノートには、次の内容が含まれています。

- [Application Server Enterprise Edition 8.1 2005Q1 の概要](#)
- [既知の問題と制限事項](#)
- [問題の報告およびフィードバックの方法](#)
- [コメントの送付先](#)
- [補足情報](#)

このリリースノートでは、関連情報が提供されているサードパーティの URL も示しています。

---

## 注

Sun は、このマニュアルに記載されているサードパーティの Web サイトの利用可能性について責任を負いません。これらのサイトまたはリソースから得られるコンテンツ、広告、製品、またはその他の資料について、Sun はなんら保証するものではなく、いかなる責任も負いません。こうしたサイトまたはリソースから取得したコンテンツ、商品、サービスを利用または信頼することに起因する損害または損失について、Sun は一切の責任を負いません。

---

---

# Application Server Enterprise Edition 8.1 2005Q1 の概要

Sun Java System Application Server Enterprise Edition 8.1 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。

詳細は次のとおりです。

- [この 8.1 リリースでの新機能](#)
- [ハードウェアとソフトウェアの要件](#)
- [スタンドアロンバージョン](#)
- [関連マニュアル](#)

## この 8.1 リリースでの新機能

以降の節で説明するように、Sun Java System Application Server Enterprise Edition 8.1 2005Q1 には、数多くの新機能と拡張機能が実装されています。

- [このリリースでの拡張機能](#)
- [J2EE のサポート](#)
- [高パフォーマンス](#)
- [スケーラビリティ](#)
- [高可用性](#)
- [JavaServer Faces 1.1 のサポート](#)

## このリリースでの拡張機能

Application Server Enterprise Edition 8.1 は、次の点が拡張されています。

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

- **プラットフォームサポートの拡張** - サポートするオペレーティングシステム、データベース、ロケール、およびハードウェアが追加されています。
- **Sun Java Enterprise System** - Sun Java Enterprise System の主要な構成要素として、Application Server はポータルサービスおよびネットワークアイデンティティサービスと緊密に統合されています。
- **移行ツールとアップグレードツール** - これらのツールにより、J2EE アプリケーションの規格合致性および移植性を保証し、別の J2EE Application Server (JBoss、WebLogic、WebSphere) からの移行に役立ち、以前のバージョンの Sun ONE Application Server や iPlanet Application Server からのアップグレードを支援することが可能になります。
- **Java 2 Standard Edition 5.0 のサポート** - Application Server は、管理および監視のための拡張機能とパフォーマンスおよびスケーラビリティについての改善点を数多く含む Java 2 Standard Edition 5.0 をサポートしています。
- **JDBC ドライバ** - Application Server には、Sun の JDBC ドライバが付属しています。
- **Web サービスセキュリティ** - これらのコンテナメッセージのセキュリティメカニズムは、SOAP Web サービス呼び出しのメッセージレベルでの認証 (たとえば XML 電子署名および暗号化) を、OASIS WS-Security 規格の X509 およびユーザー名 / パスワードのプロファイルによって実装しています。
- **WS-I Basic Profile 1.1** - J2EE 1.4 の仕様で規定されているように、このリリースは Web サービスアプリケーションの相互運用を可能にするための Web Services Interoperability (WS-I) Basic Profile 1.1 を実装しています。
- **iWay アダプタによるバックエンド接続** - Sun Microsystems は、現時点で 22 の iWay アダプタを、主要なバックエンドシステム (SAP、Siebel、Oracle、CICS、および IBM MQ Series) 向けに再販およびサポートしています。これは、Application Server 環境内から既存の IT アプリケーションを利用できるようにするためです。これらのアダプタは、J2EE Connector Architecture 1.5 仕様と Web サービス (SOAP) 規格をサポートしており、バックエンドアプリケーションに接続するための時間を節減する開発者ツールを含んでいます。
- **最新の HADB 管理システム** - UNIX® プラットフォームは、新しい高可用性データベース (HADB) 管理システム (HADB バージョン 4.4) を備えています。これにより、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 の要件と制限の詳細は、『Sun Java System Application Server Enterprise Edition 8.1 Installation Guide』を参照してください。

## J2EE のサポート

Sun Java System Application Server 8.1 2005Q1 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 1 J2EE 1.4 プラットフォームで変更になった主要な API

API	説明
<b>コンポーネント</b>	
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	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
<b>Web サービス</b>	
Java Web Services Developer Pack 1.5	XML アプリケーション、Web サービス、および Web アプリケーションを構築、テスト、および配備するための統合ツールキット
Java API for XML-based Remote Procedure Calls (JAX-RPC) 1.1	WSDL と Java テクノロジーのマッピングと、Web サービスクライアントおよびエンドポイントの開発のサポート
WS-I Basic Profile 1.0	WSDL と SOAP による相互運用の有効化要素
SOAP with attachment API for Java (SAAJ) 1.2	SOAP ベースのメッセージングのための API で、ファイルの添付された SOAP メッセージの作成を促進します
Java APIs for XML Registries (JAXR) 1.0	Universal Description Discovery and Integration (UDDI および ebXML) などの XML レジストリにアクセスするための統一された標準 API
<b>その他</b>	
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
Java API for XML Processing (JAXP) 1.2	XML ドキュメントを解析および変換するための API で、XML スキーマの処理もサポートします

表 1 J2EE 1.4 プラットフォームで変更になった主要な API (続き)

API	説明
JMS 1.1	J2EE アプリケーションコンポーネントがメッセージを作成、送信、受信、および読み取りできるためのメッセージング標準で、キューとトピックのための統一された API もサポートします
JavaMail 1.3	メールシステムをモデル化する抽象クラスの集合で、API の小規模なアップデートも含まれます

## 高パフォーマンス

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

## スケーラビリティ

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

## 高可用性

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

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

## JavaServer Faces 1.1 のサポート

Sun Java System Application Server Enterprise Edition 8.1 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ー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って、カスタム UI コンポーネントを追加することもできます。

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により、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。

## 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 8.1 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。

- [プラットフォーム要件](#)
- [重要パッチ情報](#)
- [JDBC ドライバとデータベース](#)
- [Oracle の設定](#)
- [PointBase の設定](#)
- [Web サーバー](#)
- [ブラウザ](#)
- [高可用性の要件と制限事項](#)
- [HADB のファイルシステムサポート](#)
- [Sun Java System Application Server のアップグレード](#)
- [J2SE 5.0 への切り替え](#)
- [その他の要件](#)

## プラットフォーム要件

次の表に、Sun Java System Application Server Enterprise Edition 8.1 2005Q1 製品がサポートするオペレーティングシステムを示します。また、Application Server をインストールして実行するための最小および推奨のメモリ要件も示します。

表 2 Sun Java System Application Server 8.1 2005Q1 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	推奨メモリ	最小ディスク容量	推奨ディスク容量	JVM
Sun Solaris 9、10 (SPARC) Solaris 9、10 (x86)	512M バイト	1G バイト	250M バイト の空き領域	500M バイト の空き領域	J2SE 1.4.2_06 J2SE 5.0
Redhat Enterprise Linux 2.1 Update 2、3.0 Update 1	512M バイト	1G バイト	220M バイト の空き領域	300M バイト の空き領域	J2SE 1.4.2_06 J2SE 5.0
Microsoft Windows 2000® Advanced Server Service Pack 4+	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 1.4.2_06 J2SE 5.0
Microsoft Windows Server® 2003	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 1.4.2_06 J2SE 5.0

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。

### 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、次の URL の「推奨 & セキュリティパッチ」セクションから入手できます。

<http://sunsolve.sun.com/>

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

インストールプログラムをはじめとして、この製品のネイティブコンポーネントを実行するには、標準の RedHat Enterprise Linux 3.0 ディストリビューションには含まれていないパッケージ `compat-libstdc++-7.3-2.96.118.i386.rpm` をインストールする必要があります。

このパッケージは、次の URL からダウンロードできます。

<http://mirrors.kernel.org/redhat/redhat/linux/9/en/os/i386/RedHat/RPMS/compat-libstdc++-7.3-2.96.118.i386.rpm>

## 重要パッチ情報

現時点で Sun Java System Application Server Enterprise Edition 8.1 用に必要なパッチについては、<http://sunsolve.sun.com> にアクセスして「パッチ」または「パッチ・サポート・ポータル」を選択してください。Sun Java System Application Server Enterprise Edition 8.1 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が変わって Java Enterprise System コンポーネントに対するパッチが利用可能になると、最初は推奨パッチクラスタとして SunSolve からアップデートが入手可能になります。

## JDBC ドライバとデータベース

Sun Java System Application Server は、対応する JDBC ドライバを持つ任意の 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 3 J2EE 準拠の JDBC ドライバ

JDBC ベンダー	JDBC ドライバのタイプ	サポートされるデータベースサーバー
i-net Software	Type 4	Oracle (R) 8.1.7、9i、9.2.0.3 Sybase ASE 12.5.2 Microsoft SQL Server 2000 4.0 Service Pack 1
IBM	Type 2	IBM DB2 8.1 Service Pack 3+
PointBase	Type 4	PointBase Network Server 4.8
DataDirect	Type 4	Oracle (R) 8.1.7、9i、9.2.0.3 Sybase ASE 12.5.2 Microsoft SQL Server IBM DB2 8.1 Service Pack 3+
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

i-net Software については、次の URL を参照してください。

<http://www.inetsoftware.de/>

DataDirect Technologies については、次の URL を参照してください。

<http://www.datadirect.com/>

## Oracle の設定

J2EE 1.4 準拠になるよう、Oracle JDBC ドライバを適切に設定する必要があります。Type 2 および Type 4 のドライバを次のように設定してください。

1. 9.2.0.3 以降の JDBC ドライバを使用します。
2. Oracle データベースのパラメータファイル (init.ora) に `compatible=9.0.0.0.0` またはそれ以上の値を加えます。
3. `ojdbc14.jar` ファイルを使用します。
4. Application Server の設定を変更して JVM のプロパティを次のように定義します。

```
-Doracle.jdbc.J2EE13Compliant=true
```

さらに、Type-2 のドライバについては、変数 `ORACLE_HOME` および `LD_LIBRARY_PATH` (ここには `$ORACLE_HOME/lib` が含まれているべき) を、Application Server が起動される環境に定義する必要があります。たとえば、これらを `asenv.conf` ファイルに追加し、必ずエクスポートされるようにします。

## PointBase の設定

多くのサンプルアプリケーションは、Application Server に付属の PointBase データベースサーバーを使用しています。Application Server Enterprise Edition を使用する場合、事前に PointBase データベースサーバーを設定する必要があります。Application Server で PointBase を使用する前に、サポートされている組み合わせを確認してください。

表 4 J2SE と PointBase のサポートされている組み合わせ

Application Server	PointBase
サポート済み	
J2SE 1.4	J2SE 1.4
J2SE 5.0	J2SE 1.4
未サポート	
J2SE 5.0	J2SE 5.0

PointBase を設定するには、次の 2 つの方法があります。

- 環境変数 `JAVA_HOME` を、J2SE の場所に設定します。Application Server 8.1 に含まれる PointBase 実装は、J2SE 1.4.2 でだけサポートされます。

- Application Server の PointBase 設定ファイルを編集します。

第 1 の方法を使うには、次の手順を実行してください。

1. 使用する J2SE がインストールされていることを確認します。  
まだインストールされていない場合には、J2SE 1.4.2 をダウンロードします。
2. オペレーティングシステムおよびシェルの適切なコマンドを使って、環境変数 JAVA\_HOME を J2SE がインストールされているディレクトリに設定します。次に例を示します。  

```
% setenv JAVA_HOME "/opt/SUNWappserver/jdk"
```

第 2 の方法を使う場合、その手順はオペレーティングシステムによって異なります。

## Solaris および Linux

設定ファイル `install_dir/pointbase/tools/serveroption/pbenv.conf` で、次の行を編集します。

```
PB_JAVA=%%%PB_JAVA%%%
```

この行を、次のように変更します。

```
PB_JAVA=J2SE_location
```

ここで `J2SE_location` は、J2SE がインストールされているディレクトリです。Application Server と一緒に J2SE をインストールした場合、J2SE はデフォルトで `install_dir/jdk` にインストールされます。この変更を加えたあとで、`startserver` スクリプトを使って PointBase を起動できます。

## Microsoft Windows

設定ファイル `install_dir\pointbase\tools\serveroption\pbenv.bat` で、次の行を編集します。

```
set PB_JAVA=%%%PB_JAVA%%%
```

この行を、次のように変更します。

```
set PB_JAVA=J2SE_location
```

ここで `J2SE_location` は、J2SE がインストールされているディレクトリです。Application Server と一緒に J2SE をインストールした場合、J2SE はデフォルトで `install_dir\j2se1.4` にインストールされます。この変更を加えたあとで、`startserver.bat` を実行することによって PointBase を起動できます。

## Web サーバー

ここでは、Sun Java System Application Server Enterprise Edition 8.1 2005Q1 向けにサポートされている Web サーバーを示します。

表 5 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
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+	Microsoft Windows Server 2003 Microsoft Windows 2000 Advanced Server Service Pack 4+

Microsoft 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

ここでは、Sun Java System Application Server Enterprise Edition 8.1 2005Q1 でサポートされているブラウザを示します。

表 6 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

## 高可用性の要件と制限事項

Sun Java System Application Server の高可用性コンポーネントを設定する前に、次の高可用性要件を満たしている必要があります。

- HADB が Application Server で正常に動作するには、最小で 512M バイトのメモリ、推奨で 1G バイトのメモリが必要になる。
- HADB は IPv4 しかサポートしない。
- 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る。
- create domain、extend domain、hadbm create、または hadbm addnodes コマンドで使用されるホストには、動的 IP アドレス (DHCP) を使ってはならない。

- Red Hat Linux 3.0 上で HADB を実行する場合、オペレーティングシステムによる過度なスワッピングの問題を回避するために Update 4 をインストールする必要がある。Red Hat Update 4 では AS をテストしていない点に注意が必要である。

## HADB のファイルシステムサポート

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

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

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

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

これらの設定は、Application Server 8.1 ではテストされていません。

Application Server 8.1 ソフトウェアでの HADB のインストールおよび設定の詳細は、『Sun Java System Application Server Enterprise Edition 8.1 Installation Guide』を参照してください。

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

古いバージョンの Application Server から Sun Java System Application Server Enterprise Edition 8.1 2005Q1 へのアップグレードに関する詳細な解説は、『Installation Guide』を参照してください。

## J2SE 5.0 への切り替え

Sun Java System Application Server 8.1 2005Q1 は、基盤を成す JVM として J2SE 5.0 をサポートしていますが、付属の PointBase データベースは J2SE 5.0 をサポートしていません。付属している J2SE 1.4.2 ではなく J2SE 5.0 を使って Application Server を実行するには、(Microsoft Windows の場合も Unix の場合も) 次の手順を実行してください。

1. 必要に応じて、J2SE 5.0 SDK (JRE ではない) をダウンロードしてインストールします。  
J2SE 5.0 SDK は、<http://java.sun.com/j2se/1.5.0/> からダウンロードできます。
2. Application Server を完全に停止します。  
次のコマンド行が使えます。  
`as-install/bin/asadmin stop-domain`  
または、管理コンソールの GUI で次のようにします。
  - 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 (Microsoft Windows では `asenv.bat`) で、J2SE 5.0 のホームディレクトリを指すように `AS_JAVA` の値を変更します。
4. `as-install/samples/common.properties` ファイルで、J2SE 5.0 のホームディレクトリを指すように「`com.sun.aas.javaRoot...`」で始まる行を編集します。
5. Application Server を再起動します。  
`as-install/bin/asadmin start-domain`

---

**注** 付属しているバージョン (JDK 1.4.2\_06) よりも前の JDK バージョンからアップグレードする場合には、前述した手順だけでは J2SE 5.0 にアップグレードできません。具体的には、前述の手順に加えて、既存のドメインをすべて削除して再作成する必要があります。

---

## 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、次の要件も満たしている必要があります。

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

- インストールプログラムは、使用されているポートを検出し、それ以外の2つを割り当てようとしています。それは、Sun Java™ System Message Queue (デフォルトで7676)と、IIOP (デフォルトで、IIOPは3700、IIOP/SSLは1060と1061)です。これらのデフォルトポート番号が使用されている場合、インストールプログラムは動的ポートの範囲から無作為にポート番号を割り当てようとしています (これは利用可能な次のポート番号とはかぎらない点に注意)。
- **インストール済みサーバーの起動 (UNIX):** すでにインストールされているサーバーを置き換える場合を除いて、Sun Java System Application Server 8.1 のインストール処理を開始する前に、インストール済みのサーバーを起動する必要があります。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換 (UNIX):** 古いバージョンの Sun Java System Application Server をインストール済みで、最新の Application Server で置き換える場合には、新しいサーバーをインストールする前に古いバージョンのサーバーを停止する必要があります。インストールプログラムのアップグレードウィザードを使ってサーバーをアップグレードしてください。
- **ファイアウォールのシャットダウン (Microsoft Windows):** Sun Java System Application Server ソフトウェアをインストールする前に、すべてのファイアウォールソフトウェアを停止する必要があります。一部のファイアウォールソフトウェアはデフォルトですべてのポートを無効にするからです。インストールプログラムは、使用可能なポートを正確に検出できる必要があります。

その他の互換性情報については、次の URL にある『Upgrade and Migration Guide』を参照してください。

<http://docs.sun.com/app/docs/doc/819-0083>

## スタンドアロンバージョン

スタンドアロンバージョンの Sun Java System Application Server Enterprise Edition 8.1 は、さまざまな点で Java ES Enterprise Edition バージョンと異なります。主な相違点は次のとおりです。

- 8.1 スタンドアロン製品はどのユーザーでもインストールできますが、Java ES は root ユーザーとしてしかインストールできません。
- HADB コンポーネントは、スタンドアロンバージョンではサブコンポーネントとして表示されますが、Java ES では共有コンポーネントになります。
- スタンドアロンバージョンでは Application Server に必要なすべての共有コンポーネントが1つのインストールディレクトリの下にインストールされますが、JES ではこれらのコンポーネントが個別のディレクトリにインストールされます。
- スタンドアロンバージョンのインストールプログラムは Application Server の製品ファイル、ドメイン、および設定データをデフォルトで単一のディレクトリ中に格納しますが、Java ES のインストールプログラムはこれらを複数ディレクトリに格納します。

- スタンドアロンバージョンでは、同一バージョンまたは別バージョンの Application Server がすでにインストールされているシステム上にも Application Server をインストールできます。既存のものをアンインストールする必要はありません。これは、バージョンごとに、または同一バージョンのインスタンスごとに個別のインストールディレクトリを使用することによって実現できます。
- スタンドアロンバージョンは、既存の Sun Java System Application Server Platform Edition 8.0 または Sun Java System Application Server Platform Edition 8.1 から Sun Java System Application Server Enterprise Edition 8.1 への「同じ場所へのアップグレード」をサポートしています。

## 関連マニュアル

これらのリリースノート以外に、Application Server 製品にはあらゆるマニュアルがそろっています。次の URL を参照してください。

[http://docs.sun.com/app/docs/coll/ApplicationServer8\\_ee\\_04q4?l=ja](http://docs.sun.com/app/docs/coll/ApplicationServer8_ee_04q4?l=ja)

次の表に、Application Server コアアプリケーション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 7 マニュアルセットの内容

マニュアルのタイトル	説明
『クイックスタートガイド』	Sun Java System Application Server 製品をはじめて使われる方のための入門。
『Installation Guide』	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。
『Deployment Planning Guide』	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。
『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 スタンダー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、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全般的な情報も提供します。
『J2EE 1.4 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。
『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。

表 7 マニュアルセットの内容 ( 続き )

マニュアルのタイトル	説明
『High Availability Administration Guide』	高可用性データベースのための、インストール後の設定と管理に関する解説。
『Administration Reference』	Sun Java System Application Server 設定ファイル domain.xml の編集。
『Upgrade and Migration Guide』	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への移行 ( 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。
『Performance Tuning Guide』	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。
『Troubleshooting Guide』	Sun Java System Application Server におけるトラブルの解決。
『Error Message Reference』	Sun Java System Application Server のエラーメッセージの解決。
『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。

## 既知の問題と制限事項

ここでは、Sun Java System Application Server Enterprise Edition 8.1 2005Q1 に関する既知の問題とその回避策を説明します。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。

- [管理](#)
- [アプリケーションクライアント](#)
- [付属の Sun JDBC ドライバ](#)
- [コネクタ](#)
- [コンテナ管理による持続性](#)
- [Deploytool](#)
- [マニュアル](#)
- [高可用性](#)
- [インストール](#)
- [J2EE Tutorial](#)

- ライフサイクル管理
- ログ
- メッセージキュー
- 監視
- PointBase
- サンプル
- セキュリティ
- アップグレードユーティリティ
- Web コンテナ

## 管理

ここでは、管理上の既知の問題とその解決方法を示します。

### domain1 がない場合に package-appclient スクリプトが機能しない。(ID 6171458)

デフォルトでは、`$INSTALL/lib/package-appclient.xml` に、`asenv.conf` から参照される `domain1` の `AS_ACC_CONFIG` 変数の値がハードコードされています。`domain1` を削除して新たなドメインを作成しても、変数 `AS_ACC_CONFIG` 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。

#### 解決法

次のいずれかを行います。

- `domain1` はそのままにしておき、その前後に別のドメインを作成します。
- `domain1` を削除し、`$INSTALL/lib/package-appclient.xml` 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。

### バックアップ取得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 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も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ってパスを設定するからです。

## 解決法

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 エージェントを伴う Application Server の起動がサポートされていない。(ID 6200011)

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。これの望ましくない副作用として、管理機能が悪影響を受け、Application Server の管理 GUI と CLI から予想できない結果が生じる場合があります。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、資格が正常に認識されたあとは、同様の管理および設定を行うことができます。

```
[#|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.|#]
```

詳細は、『Sun Java System Application Server 8.1 管理ガイド』を参照してください。

## Application Server の start および stop スクリプトに対する実行権に過度な制限がかけられている (UNIX のみ) (ID 6206176)。

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/*
```

## 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 (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。

### 解決法

現時点ではありません。

## 付属の Sun JDBC ドライバ

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

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

希望の遮断レベルを接続に対して設定するには、対応する接続プールをその遮断レベルで作成する必要があります。接続プールの設定については、『Application Server 8.1 2005Q1 管理ガイド』を参照してください。

### 解決法

ありません。

## 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、『Application Server 8.1 2005Q1 管理ガイド』を参照してください。

### 説明 2

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

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

### 解決法 2

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

### 説明 3

遮断レベル TRANSACTION\_SERIALIZABLE。

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

### 解決法 3

希望の遮断レベルを接続に対して設定するには、対応する接続プールをその遮断レベルで作成する必要があります。詳しくは、『Application Server 8.1 2005Q1 管理ガイド』を参照してください。

**Application Server Enterprise Edition 8.1 で、付属の DB2 Sun JDBC ドライバがデフォルトの設定で動作しない。これは、Sun の DB2 JDBC ドライバクラスが明示的な charsetProvider RuntimePermission を要求するため。(ID 6183492)**

### 解決法

JDBC ドライバを使用する配備済みアプリケーションに次の権限を与えるように、server.policy ファイルを変更します。

```
grant codeBase "file:${DEPLOYED_APPLICATION_DIR}" { permission
  java.lang.RuntimePermission "charsetProvider";
};
```

これは、付属の DB2 Sun JDBC ドライバについてだけ必要です。

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

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

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

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

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

## 解決法

ありません。

## コネクタ

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

### jdbc-connection-pools で connection-validation を動的に再設定できない。(ID 4930792)

JDBC 接続プールを生成したあとは、その `is-connection-validation-required` 属性を動的に再設定できません。したがって、生成済みのプール接続については、その場で `validation` を `on` または `off` に切り替えることができません。これは、プールの `validation-method` 属性についても同様です。

## 解決法

稼動しているアプリケーションへの影響が少ない順に挙げると、次の3つの回避策があります。

1. `validation` を `on` にして `jdbc-connection-pools` を生成します。
2. `jdbc-connection-pool` を削除し、`validation` を `on` にして再生成します。  
これは、特定のプールに依存する配備済みアプリケーションにだけ影響します。
3. `validation` プロパティを変更して `Application Server` を再起動します。  
再起動が行われるため、これは配備済みのすべてのアプリケーションに影響します。

### 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.sys
tem
.core|_ThreadID=14;|CORE5023: Error while unloading application [foo]|#]
```

## 解決法

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

## コンテナ管理による持続性

ここでは、コンテナ管理による持続性について、既知の問題とその解決方法を示します。

**where 節に OR 演算子と単一値の cmr ナビゲーションが含まれる場合、EJBQL クエリーに一致するすべての結果が含まれないことがある。(ID 6184864)**

EJBQL クエリーの where 節に OR 演算子と単一値の cmr ナビゲーションが含まれる場合、そのナビゲーションパスが別の OR 節にあっても、ナビゲーションパスが null になっている行の結果はクエリーの結果に含まれません。

たとえば、Employee、Department、および Insurance から構成されるスキーマについて考えます。Employee は Department と 1 対多の関係にあり、Insurance と 1 対 1 の関係にあります。

```
select Distinct Object(e) from Employee e
       where e.name = 'John' OR e.department.name = 'Engineering'
```

このクエリーは、名前 (name) が John であって、かつ、どの部門 (department) にも属さない従業員 (employee) は返しません。

```
select Distinct Object(e) from Employee e
       where e.department.name = 'Engineering' OR e.insurance.name
= 'xyz'
```

このクエリーは、保険 (insurance) の名前が xyz であって、かつ、どの部門にも属さない従業員は返しません。また、部門名が Engineering であって、かつ、どの保険にも加入していない従業員も返しません。

## 解決法

各 OR 条件のクエリーを個別に実行し、その結果を結合してください。

# Deploytool

この節では、deploytool の既知の問題とその解決方法を示します。

## 次の Sun 配備記述子には、deploytool が message-destination 要素を作成しないことがある (ID 6197393)

- sun-application-client.xml
- sun-ejb-jar.xml
- sun-web.xml

「メッセージ送信先」タブで JNDI 名として指定された JMS 送信先リソースは、Sun 記述子に保存されないことがあります。送信先名 (たとえば create-jmsdest で作成される物理送信先である PhysicalQueue) を指定して Enter キーを押すと、「表示名」の下に「送信先名」が表示され、「プロデューサ」リストにクライアントまたは Bean の名前が表示されます。「Sun 固有の JNDI 名」テキストフィールドに「jms/Queue」と入力して Enter キーを押したあとは、アプリケーションはタイトルバーに「(変更されています)」と表示せず、~/deploytool/logfile にエラーが書き出されます。アプリケーションを保存して「メッセージ送信先」タブに戻ると、「JNDI 名」フィールドは再度空白になっています。「ツール」>「記述子ビューア」>「アプリケーションサーバー記述子」によって Sun 記述子を表示すると、<jndi-name> 要素内に <message-destination> 要素は作成されていません。

問題は、deploytool セッションの間、最初にメッセージ送信先 JNDI 名として値を入力した時点でその値が Sun 記述子中で正しいように見えても、

org.netbeans.modules.schema2beans.BeanProp.setElement() から IllegalArgumentException がスローされることにあります。それ以降、同一アプリケーションまたは別のアプリケーションでメッセージ送信先 JNDI 名を変更または追加しても、それは Sun 記述子に追加されません。

## 解決法

メッセージ送信先の既存の JNDI 名を編集するには、次の手順を実行します。

1. 「JNDI 名」テキストフィールドが空白の状態ですべて Enter キーを押すことによって、既存の JNDI 名を削除します。
2. 新たな JNDI 名を入力して Enter キーを押します。
3. 「ツール」>「記述子ビューア」>「アプリケーションサーバー記述子」をクリックして Sun 記述子を確認します。
4. 「ファイル」>「保存」をクリックしてアプリケーションを保存します。

JNDI 名が Sun 記述子に保存されない場合には、次の手順を実行します。

1. deploytool を再起動します。
2. 「メッセージ送信先」タブで、メッセージ送信先を選択するか、新たなメッセージ送信先を追加します。

3. 「Sun 固有の JNDI 名」テキストフィールドにメッセージ送信先の JNDI 名を入力し、Enter キーを押します。
4. 「ツール」>「記述子ビューア」>「アプリケーションサーバー記述子」をクリックして Sun 記述子を確認します。
5. 「ファイル」>「保存」をクリックしてアプリケーションを保存します。

deploytool セッションの間に「JNDI 名」テキストフィールドにはじめて値を入力する場合を除いて、「メッセージ送信先」タブの「Sun 固有の JNDI 名」に値を入力する必要があるたびに、上記の手順を繰り返します。

## 新規 Web サービスウィザードでパネルが繋がらない (ID 6198981)

この問題は、次の 2 つの兆候を伴って現れます。

- 「WSDL ファイル」パネルで「次へ」をクリックしても次のパネルに移動せず、次の例外がスローされます。  

```
java.lang.NoClassDefFoundError: com/sun/org/apache/xpath/internal/XPathAPI
```
- 「パッケージ化されたエンドポイントの作成」パネルで「Web サービスエンドポイントの作成とパッケージ化 (R)」を選択します。「次の手順」パネルで「完了」をクリックしてもエンドポイントモジュールが出力ディレクトリに作成されず、次の例外がスローされます。

```
java.lang.NoClassDefFoundError: com/sun/org/apache/xpath/internal/XPathAPI
```

問題は、XPathAPI.class が含まれる xalan.jar が、CLASSPATH に見つからないことにあります。この問題は、JDK 5.0 では発生しません。

### 解決法

`install_dir/lib/processLauncher.xml` の `s1as-deploytool` プロセス用の `CLASSPATH includes` に、`xalan.jar` を追加します。次に例を示します。

```
includes="appserv-assemblytool.jar,activation.jar,appserv-admin.jar,appserv-cmp.jar,appserv-rt.jar,j2ee.jar,jaxrpc-impl.jar,appserv-ext.jar,deployhelp.jar,admin-cli.jar,dom.jar,xercesImpl.jar, xalan.jar"
```

## 簡体字中国語の deploytool で「home」が「installation directory」として翻訳されている。(ID 6203658)

deploytool で Enterprise Bean を作成し、その Bean ノードの「トランザクション」または「セキュリティ」タブに移動すると、ラベル「Local Home」および「Remote Home」が「Local Installation Directory」および「Remote Installation Directory」として間違って翻訳されています。

## マニュアル

ここでは、マニュアル上の既知の問題とその解決方法を示します。

### index.html および QuickStart.html に誤りがある。(ID 6193749)

Application Server 8.1 マニュアルセットの index.html ファイルと docs-ee/QuickStart.html ファイルに誤りがあります。

1. 各ドメインの docroot ディレクトリにコピーされたデフォルトの index.html ページが、誤ったパスを示している。

```
as_instance/domains/domain_name/docroot/
```

となるべきところで、次のようになっている。

```
install_dir/domains/domain_name/docroot/index.html
```

2. 『クイックスタートガイド』が、ロードバランサと Web サーバーのインストールについて、誤った情報を提供している。最新の情報については、次の URL にあるオンラインの QuickStart Guide を参照。

<http://docs.sun.com/app/docs/doc/819-1261?l=ja>

これは、Java ES Enterprise Edition バージョンの Application Server 8.1 にのみ影響します。スタンドアロンバージョンに影響はありません。

### - asadmin create-domain --help コマンドを実行すると、誤った使用方法と無効なオプションが表示される (--admin.jmxport)。(ID 6207862)

asadmin create-domain のヘルプコマンドで、無効なオプションである --admin.jmxport の説明が表示されます。

#### 解決法

asadmin create-domain コマンドでは --admin.jmxport を使用できません。

### 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()、または 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。

## 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。

『管理ガイド』の付録、「Apache Web サーバーのコンパイルと設定」では Apache を Sun Java System Application Server で使用するための要件が記されているが、これはすでに内容が古い。

Apache Web サーバーを HADB とともに使用するためのソフトウェア要件は、次のとおりです。

- Apache 1.3 の場合の要件
  - openssl-0.9.7e (ソース)
  - mod\_ssl-2.8.16-1.3.33 (ソース)
  - apache\_1.3.33 (ソース)
  - gcc-3.3-sol9-sparc-local パッケージ (Solaris 9 SPARC/ x86 の場合)
  - gcc-3.3-sol9-intel-local パッケージ (Solaris 9 x86 の場合)
  - flex-2.5.4a-sol9-sparc-local パッケージ (Solaris 9 SPARC の場合)
  - flex-2.5.4a-sol9-intel-local パッケージ (Solaris 9 x86 の場合)
- Apache 2.0 の場合の要件
  - openssl-0.9.7e (ソース)
  - httpd-2.0.49 (ソース)
  - gcc-3.3-sol9-sparc-local パッケージ (Solaris 9 SPARC の場合)
  - gcc-3.3-sol9-intel-local パッケージ (Solaris 9 x86 の場合)
  - flex-2.5.4a-sol9-sparc-local パッケージ (Solaris 9 SPARC の場合)
  - flex-2.5.4a-sol9-intel-local パッケージ (Solaris 9 x86 の場合)

コンパイルの前に実行すべき追加のステップもあります。Solaris 10 プラットフォームでは、OpenSSL の make を行う前に、次のコマンドを実行する必要があります。

```
/usr/local/lib/gcc-lib/sparc-sun-solaris2.9/3.3/install-tools/mkheaders
```

最後に、インストールが不完全に終わった場合には、『管理ガイド』中の「ロードバランスとフェイルオーバーの設定」に詳述している変更を Apache Web サーバーに加える必要があります。

### すべてのプラットフォームについて

1. `apache_install_dir` の下に `sec_db_files` という名前のディレクトリを作成します。
2. `domain/config/*.db` を `apache_install_dir/sec_db_files` にコピーします。

### Solaris プラットフォームについて

- `apache_install_dir/bin/apachectl` スクリプトの `LD_LIBRARY_PATH` に、パス `/usr/lib/mps/secv1` を追加します。このパスは、`/usr/lib/mps` よりも前に追加する必要があります。

### Linux プラットフォームについて

- `apache_install_dir/bin/apachectl` スクリプトの `LD_LIBRARY_PATH` に、パス `/opt/sun/private/lib` を追加します。このパスは、`/usr/lib` よりも前に追加する必要があります。

### ダブルネットワークでの HADB 設定 (ID なし)

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

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

### 新規ノードが追加されたあとに作成された新規テーブルは、追加ノード上でフラグメント化されない。(ID 5042351)

データベースインスタンスを作成し、それにノードを追加すると、データベース作成以後に追加されたノード上で、新しく作成されたテーブルがフラグメント化されません。`hadbm addnodes` が再断片化を行うと、`addnodes` コマンドの実行以降に作成されたテーブルは、追加されたノードを使用できません。これは、`hadbm create` の実行時に作成される `sysnode` ノードグループを `create table` が使用するからです。

## 解決法

新規テーブルが追加されたあとで `hadbm refragment` を実行するか、ノードグループ `all_nodes` で新規テーブルを作成します。

## 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、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/install11 --hosts europa11
Package successfully registered.
hadbm registerpackage test --packagepath=/var/install12 --hosts europa12
hadbm:Error 22171: A software package has already been registered with the
package name test.
```

## 解決法

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全域について同一にしてください。

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

`hadbm set` を使っ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。

## co-locate モードの RedHat AS 3.0 上で、負荷がかかったときに問題が発生する。(ID 6158393)

HADB が AS と RedHat 3.0 上で `co-located` モードで稼働している場合、トランザクションが異常終了したり、パフォーマンスに影響を及ぼしたりすることがあります。これは、オペレーティングシステムによって行われる過度なスワッピングに起因します。

## 解決法

この問題は、Red Hat EL 3.0 Update 4 で解決しています。RedHat 3.0 update 4 で HADB がテストされ、オペレーティングシステムによる過度なスワッピングが解消することが確認されています。Application Server 8.1 は update 4 でテストされていない点に注意してください。

## configure-ha-cluster コマンドがハングアップすることがある。(ID 6159633)

複数ホスト上に高可用性クラスタを作成または設定するために `asadmin configure-ha-cluster` コマンドを使用すると、このコマンドがハングアップすることがあります。HADB 管理エージェントまたは Application Server から例外はスローされません。

### 解決法

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と設定ディレクトリは、すべての参加ホスト全域について同一にしてください。再度コマンドを実行する前に、必ずリポジトリディレクトリをクリアしてください。

## HADB を使用する Application Server のパフォーマンスが低下する。(ID 6172589)

すべてのプラットフォームについて、HADB を使用するように設定された Application Server インスタンスのパフォーマンスは、HADB が使用する JDBC ドライバに加えられた変更により、以前のリリースよりも低下します。

### 解決法

ご購入先に連絡してください。

## 再断片化時に 2 回目の `addnodes` が失敗する。(ID 6175436)

2 回目 (およびそれ以降) の `addnodes` コマンドは、再断片化の間に次のエラーで失敗することがあります。

```
hadbm:Error 22042: Database could not be refragmented. Please retry with hadbm
refragment command to refragment the database. Caused by: HADB-E-11747:
Nodegroup all_nodes exists already
```

### 解決法

`hadbm refragment` を使って手作業でテーブルを再断片化してください。

## Microsoft Windows 上で 2G バイトよりも大きなデータデバイスを作成できない。(ID 6181845)

--NumberOfDataDevices=1 (デフォルト) および 2G バイトを超える値の --devicesize を伴う `hadbm create` または `hadbm set` を使用すると、次のエラーが発生します。

```
DEVINIT-ERROR: out of space, wrote -2147479552 B of -2036330496 B
An attempt was made to move the file pointer before the beginning of the file.
```

## 解決法

Microsoft Windows で 2G バイトよりも大きなデータデバイスを作成する必要がある場合には、`devizesize` を 2G バイトで割り、必要となるデバイスの数を求めます。この計算結果をもとに、`--NumberOfDataDevices` オプションを使ってデータデバイスを多数作成します。たとえば、5G バイトのデータデバイスが必要な場合、次のようになります。

$$5 \div 2 = 2.5$$

これを切り上げて、`--NumberOfDataDevices=3` を指定します。

## hadbm ヘルプの情報が古い (ID 6190702)

hadbm のヘルプシステムの情報は、一部が古くなっています。

## 解決法

最新の情報は、『Application Server 8.1 管理ガイド』の HADB に関する章を参照してください。

## addnodes コマンドが table not found エラーで失敗する (ID 6214601)

このシナリオでは、`hadbm refragment` コマンドが次のエラーで失敗します。

```
hadbm:Error 22042: Database could not be refragmented. Please retry with hadbm
refragment command to refragment the database.. Caused by: HADB-E-11701:
*Table singlesignon not found*
```

## 解決法

回避策は、次のように `clusql` コマンドを使うことにより、手動で App Srv テーブルを再断片化することです。

```
> clusql <server:port list> system+<dbpassword specified at database create>
SQL: set autocommit on;
SQL: set schema haschema;
SQL: alter table sessionattribute nodegroup all_nodes;
SQL: alter table singlesignon nodegroup all_nodes;
SQL: alter table statefulsessionbean nodegroup all_nodes;
SQL: alter table sessionheader nodegroup all_nodes;
SQL: alter table blobsessions nodegroup all_nodes;
SQL: quit;
```

## 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。

### インストールプログラムおよびアンインストールプログラムの開始画面上の「次へ」ボタンのレンダリングが断続的に失敗する。この問題は、製品のスタンダードバージョンにのみ影響する。(ID 4977191)

この問題は、Solaris x86 プラットフォームについて継続的に報告されてきましたが、Solaris SPARC および Linux プラットフォームにも影響が出る可能性があります。

これは、インストールプログラムやアンインストールプログラムの最初の画面で、テキストと「ヘルプ」および「キャンセル」ボタンは正常に表示されるものの、次の画面に進むための「次へ」ボタンが表示されない、という問題です。ボタンは表示されませんが、その領域はアクティブになっており、そこをユーザーがクリックすると次の画面に正常に進みます。この問題は、J2SE GUI の再描画が断続的であることが原因で起こります。

#### 解決法

回避策の 1 つは、「ヘルプ」ボタンのすぐ左にある「次へ」ボタン領域をクリックすることです。別の回避策として、インストールプログラムのウィンドウを少しサイズ変更するか、いったん最小化してから元のサイズに戻すことにより、画面を意図的に再描画させるという方法もあります。再描画させると、見えなかった「次へ」ボタンが表示されます。

### 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)

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 で最も一般的に見られますが、RedHat 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。

## Linux 上のインストールラッパーで、J2SE の検出およびブートストラップの問題が断続的に発生する。(6172980)

Linux のインストールプログラムを起動する setup プログラムは、ハングアップする場合があります。J2SE の場所を解決してインストールウィザードを起動する代わりに、ラッパーがハングアップして次のメッセージを返します。

```
Chcking available disk space....
Checking Java(TM) 2 Runtime Environment....
Extracting Java(TM) 2 Runtime Environment....
Deleting temporary files.....
```

この問題は Linux の一部のバージョンでだけ見られ、特に変数 `JAVA_HOME` の存在など、環境の設定に依存します。

### 解決法

この問題を回避するには、次の手順を実行します。

1. 使用しているシェルに応じて `unset` または `unsetenv` を実行することにより、変数 `JAVA_HOME` の設定を解除します。
2. インストールプログラムが使用する `JAVA_HOME` を指定する `-javahome` オプションを付けて `setup` を実行します。

## インストールの間に `imq` ディレクトリを作成する必要がある (Microsoft Windows のみ)。(ID 6199697)

Microsoft Windows に Application Server EE をインストールした直後に、ディレクトリ `drive:\$asomainsomain1\imq` が存在しない旨のメッセージを出して IMQ 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。

### 解決法

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:\$asomainsomain1\imq
```

## J2EE Tutorial

Sun Java System Application Server Enterprise Edition 8.1 2005Q1 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って何らかのリソースを作成する場合には、「ターゲット」タブを使っ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` プロパティの `minimum-delivery-interval` を 9000 に設定したあとで `ejb-timer-service` プロパティの `redelivery-interval-in-millis` を 7000 に設定しようとすると、次のエラーで `set` コマンドが失敗する (ID 6193449)

```
[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。`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 についてデバッグ文を設定すると、Application Server の起動でハングアップする。(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、6208728、6198481、6199510、6199510)

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。しかし、一般には次の方法で問題を回避できます。

- 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 で変更になった (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 VM` はデーモンとして存在します。この動作は、ACC で非同期受信を行う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つから、メインスレッドを終了します。

## Message Broker のログに「unable to deliver」というメッセージが含まれる。(ID 6204180)

クラスタ化された環境でブローカのメモリが少なくなると、ブローカのログに次のエラーメッセージが現れることがあります。

```
Internal error, unable to deliver .....: java.lang.NullPointerException
```

持続コンシューマでは、コンシューマがメッセージの確認応答をした場合でも、あとでそのメッセージを (redeliver フラグを設定して) 再配信できます。

このエラーが発生するのは、システム上のメモリが十分でなくなったときに、システムがまだ必要としていたデータが不適切に解放されたからです。このエラーは、このブローカに接続されたコンシューマ向けに配信されてクラスタ内の別のブローカに送られたメッセージについてのみ発生します。

## 解決法

Message Broker プロセスの最小 Java ヒープサイズを増やす (-Xmx) ことにより、システムのメモリが不足することを回避します。

## クラスタのブローカが、失敗後の再起動のあとにメモリ不足になる。(ID 6205463)

クラスタの一部として Message Broker が動作している場合、再起動を実行するとメモリー不足になります。この問題を引き起こす可能性のある問題点が 2 つあります。どちらかまたは両方が関係しているかを判断するには、クラスタの Application Server および Message Broker のログファイルを調査してください。

- Application Server のログファイルに次の形式の警告メッセージがある場合、

```
MQRA:OMR:Endpoint Unavailable:Marking OMR as invalid-omrId=1  
for:ActvationSpec configuration ...
```

「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、6208728、6198481、6199510、6199510)」で説明しているメモリの問題が発生しています。

- クラスタのその他のブローカ上のブローカログファイルに次の形式のメッセージが現れる場合、

```
ERROR Internal Error: received ack twice on ...
```

「失敗のあとでクラスタを再起動すると、メッセージブローカにメモリが集中する。(ID 6208621)」で説明しているメモリの問題が発生しています。

## 解決法

どちらが問題の原因になっているかに応じて、6208621 または 6208728 に示している解決策を参照してください。

## 失敗のあとでクラスタを再起動すると、メッセージブローカにメモリが集中する。(ID 6208621)

失敗のあとでクラスタ内の Message Broker を再起動すると、トピック送信先の非持続 MDB の状態が適切に送られないため、そのブローカにメモリが集中することがあります。

アクティブなクラスタの一部として Message Broker を起動したあとは、ブローカログとアプリケーションサーバーログの両方に例外が見られます。その後、再起動されたブローカは、メモリが不足し始めます。このような場合、クラスタのアプリケーションサーバーのログは、ブローカの再起動のあとに次のような警告メッセージをスローします。

```
[#|2004-12-03T17:45:12.821-0800|WARNING|sun-appserver-ee8.1|javax.enterprise.  
system.stream.err|_ThreadID=12;|com.sun.messaging.jms.JMSException: [C4000]:  
Packet acknowledge failed. user=admin, broker=<brokername>....
```

通常、Message Broker のログにも関連するメッセージが見られます。次の形式になっています。

```
ERROR Internal Error: received ack twice on ...
```

問題は、Message Broker がアクティブなクラスタに接続する場合、すべてのアクティブなコンシューマに関する情報がクラスタ内のその他のブローカから転送されることにあります。MDB がリモートブローカの 1 つに非持続サブスクライバを持つ場合、そのブローカはコンシューマ情報を転送するときに誤った情報を送信することがあります。再起動されたブローカが無効なコンシューマ情報を受信すると、そのブローカは、その他のブローカに対してメッセージの余計なコピーを配信します。

これが発生すると、リモートコンシューマは Application Server ログに「[C4000]: Packet acknowledge failed」というメッセージを記録し、リモートブローカには「double ack」エラーが記録されます。ブローカログに「double ack」メッセージが作成されても、そのメッセージは生成側のブローカによって適切に確認応答されません。やがて、生成側のブローカは、メモリ不足またはメモリ枯渇の状態になります。

## 解決法

この問題が発生している場合には、システム全体にわたって MDB の配備取り消しを行い、再配備することにより、内部の情報を一掃できます。正常稼働時にこの問題がブローカの処理に影響を及ぼさないようにするには、次の手順を実行します。

- パフォーマンスに問題がない場合、非持続サブスクライバの代わりに永続サブスクライバを使用するように MDB を変更します。
- パフォーマンスに問題がある場合、一定期間内に確認応答を受信できなければ「漏えい」メッセージを除去するように、システムを設定します。これは、次のいずれかの方法で行えます。
  - REMOVE\_OLDEST の limitBehavior と、MDB が使用する送信先のメッセージ制限を設定します。次のようにします。

```
imqcmd update dst -n <dest name> -t t -o limitBehavior=REMOVE_OLDEST -o  
maxNumMsgs=<some limit>
```

- そのトピックに送られるメッセージについて、「生存時間」の設定を変更します。

## 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。

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

HTTP サービスの一部の要素の監視統計を調べる場合、提示される値のいくつかは、現時点での値を反映していないか、常に 0 になっています。具体的には、次に挙げる HTTP サービス統計は Application Server 8.1 に適用できる情報を提示しないため、無視する必要があります。

- http-service
  - load1MinuteAverage
  - load5MinuteAverage
  - load15MinuteAverage
  - rateBytesTransmitted
  - rateBytesReceived
- pwc-thread-pool (要素)

### 解決法

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。

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

次に例を示します。

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

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

### 診断

asadmin list -m "server.applications" を実行すると、次の出力が得られます。

```

server.applications.MEjbApp
server.applications.__ejb_container_timer_app
server.applications.adminapp
server.applications.admingui
server.applications.com_sun_web_ui
server.applications._export_install_nov-11_domains_domain1_applications_j2ee-
modules_sqe_ejb_s1_01

```

次のようにして統計を調べることができます。

```

bin/asadmin list -m "server.applications._expo
rt_install_nov-11_domains_domain1_applications_j2ee-modules_sqe_ejb_s1_01"
server.applications._export_install_nov-11_domains_domain1_applications_j2ee-
mod
ules_sqe_ejb_s1_01.SQEMessage
server.applications._export_install_nov-11_domains_domain1_applications_j2ee-
mod
ules_sqe_ejb_s1_01.TheGreeter

```

いったん配備を取り消します。

```
_export_install_nov-11_domains_domain1_applications_j2ee-modules_sqe_ejb_s1_01
```

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

```

asadmin list -m "server.applications"
server.applications.MEjbApp
server.applications.__ejb_container_timer_app
server.applications._export_install_nov-11_domains_domain1_applications_j2ee-
mod
ules_sqe_ejb_s1_01
server.applications.adminapp
server.applications.admingui
server.applications.com_sun_web_ui

```

しかし、何の監視統計も含まれていません。

```

asadmin list -m "server.applications._expo
rt_install_nov-11_domains_domain1_applications_j2ee-modules_sqe_ejb_s1_01"
Nothing to list at
server.applications.-export-install-nov-11-domains-domain1-ap
plications-j2ee-modules-sqe-ejb-s1-01.

```

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

## 解決法

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

## PointBase

ここでは、PointBase に関する既知の問題とその解決方法を示します。

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

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

#### 解決法

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

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

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

#### 解決法

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

## サンプル

ここでは、Application Server 8.1 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。

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

たとえば、このエラーを再現するには、

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

- コンソール 1  

```
cd install_dir\samples\ee-samples asant start-mq-master-broker1
```
- コンソール 2

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

- コンソール 3

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

- コンソール 4

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

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

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

しかし、このあとシステムはハングアップします。

### 解決法

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

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

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` コマンドを使っ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。

### 解決法

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

## Linux で、webservices/security のサンプルにおける証明書の作成時に実行時エラーが発生する。(ID 6198239)

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。

### 解決法

次のいずれかを行います。

- `LD_LIBRARY_PATH=/opt/sun/private/lib` を設定します。
- 次の行を `install_dir/bin/asant` スクリプトに追加します。

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

- `ee samples asadmin deploy` コマンドのマニュアルで、アプリケーションを配備するための `availabilityenabled=true` オプション (これにより、サンプルアプリケーションがデフォルトで HADB 対応になる) が抜けている。(ID 6198796)

### 解決法

次のように `asadmin` コマンドを使用します。

1. 次の例のようにして、サンプルディレクトリのルートに移動します。

```
cd install_dir/samples/ee-samples/failover/apps/dukesbookstore
```

2. `asadmin deploy` を実行して、ローカルの Application Server インスタンスにアプリケーションを配備します。次のようにします。

```
install_dir/bin/asadmin deploy -user adminuser -password adminpassword --host
DAShost --port DASport --availabilityenabled=true --target cluster-name
dukesbookstore.ear
```

その他のすべての EE サンプル (`install_dir/samples/ee-samples/failover/apps/mqfailover` を除く) についても、同様の `asadmin deploy` コマンドを実行します。MQ は HADB を使用しません。

## dukesbookstore EE サンプルについて、failover テストを asant スクリプトで実行できない。(ID 6199076)

マシン 2 台のクラスタをセットアップすると、dukesbookstore failover サンプルでエラーが発生します。ここでは、クラスタごとに 1 つのデータベースを設けることを想定しています。現時点では、2 台の別々のマシン上で動作するインスタンスを持つクラスタ上に DAS からサンプルを配備する場合、スクリプトは PointBase ホストを localhost として使用します。EE サンプルを配備すると、JDBC リソースは両方のインスタンス上の localhost として PointBase ホストとともに配備されます。したがって、両方のインスタンス上での localhost:9092 は、両方のマシン上で PointBase が動作していると想定します。

問題は、1 つのクラスタに属する 2 つのインスタンスが別々のデータベースを使用できないことにあります。この問題を回避するために、`database.properties` にある localhost をホスト名で置き換えれば、クラスタの両方のインスタンスがデータベースにアクセスできます。1 つは localhost 経由で、もう一方は指定されたホスト名経由でアクセスします。

### 解決法

PointBase サーバーが動作しているホスト上の `install_dir/samples/database.properties` ファイルを編集して、`pointbase.server` の値を localhost ではなく PointBase が動作しているホスト名に設定します。

現在の設定、つまり `pointbase.server=localhost` は、マシン 1 台のクラスタについては有効ですが、第 2 インスタンスの localhost で、PointBase が動作しない可能性のある 2 台構成のクラスタについては有効ではありません。

## MQ-failover サンプルアプリケーションのクラスタ名が、Ant セットアップターゲットの 1 つで「cluster1」としてハードコードされている。(ID 6202363)

MQ-failover サンプルアプリケーションでは、Ant セットアップターゲットの 1 つで cluster1 がハードコードされています。そのため、`ee-samples` 内の `cluster.properties` で `cluster-name` を cluster1 から別の名前に変更すると、cluster1 にデフォルトホストを設定しようとしてサンプルアプリケーションが失敗します。

```
set-default-jms-host-to-broker1:
[echo] Setting default JMS host to samplesbroker1
[exec] No object matches the specified name
"cluster1-config.jms-service.default-jms-host"
[exec] CLI137 Command set failed.
```

`install_dir/samples/ee-samples/build.xml` ファイルでは、`install_dir/samples/ee-samples/cluster.properties` の `${cluster.name}` を使用すべきところを、`cluster1` への参照としてハードコードされています。

## 解決法

ハードコードされている文字列 `cluster1` の代わりに、`${cluster.name}` を使用する必要があります。`install_dir/samples/ee-samples/build.xml` を修正して、`cluster1` の `set-default-jms-host-to-broker1` を、`${cluster.name}` か、`cluster.properties` で指定されている `customer` クラスタ名に変更します。

## 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。

### java-method によるターゲットメッセージの指定が client-side message-security-binding 要素で有効でない。(ID 6155080)

たとえば、`client-side message-security-binding` 要素のターゲットメッセージが `service-ref` 要素内の `port-info` 要素にある `java-method` によって指定される場合に、この問題が発生します。

```
<!ELEMENT service-ref ( service-ref-name, port-info*, call-property*,
wsdl-override?, service-impl-class?, service-qname? )>
<!ELEMENT port-info ( service-endpoint-interface?, wsdl-port?, stub-property*,
call-property*, message-security-binding? )>
<!ELEMENT message-security-binding ( message-security* )>
<!ELEMENT message-security ( message+, request-protection?,
response-protection? )>
<!ELEMENT message ( java-method? | operation-name? )>
```

ここで `message-security-binding` 要素は、Web サービスエンドポイントの特定メソッドのメッセージ保護ポリシーを定義するために使用されます。

## 解決法

`message` 要素内の `operation-name` 要素を使って、`message-security` 要素で定義された保護ポリシーが適用されるメッセージを WSDL オペレーション名によって指定します。

## DAS マシンとリモートのノードエージェントマシンのクロックが同期しない場合に「CertificateNotYetValidException」エラーになる。(ID 6181989)

DAS サーバーとノードエージェントが異なるマシン上にインストールされていて、それらのマシンのクロックが同期していない場合に、`asadmin --start remote-node-agent` コマンドを実行しようとするると、「CertificateNotYetValidException」エラーで失敗します。

### 解決法

DAS サーバーマシンとリモートのすべてのノードエージェントマシンのクロックを同期します。

## EE 上の WebServiceSecurity アプリケーションを J2SE 5.0 で実行できない。(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

J2SE チームは、このバグについて「CR 6190389: Add support for the RSA-PKCS1 and RSA-OAEP wrap/unwrap mechanisms」をファイルしています。

### 解決法

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。

## jms-servicSSL に mq-scheme と mq-service が設定されると、MQ との SSL 通信が機能しない。また、jms-service に mq-scheme と mq-service が設定されると、MQ との通信が機能しない。(ID 6202606)

使用されるべき情報が、リソースアダプタから生成される接続ファクトリから利用できるようにリソースアダプタ中に設定されません。そのため、リソースアダプタから生成される接続ファクトリには SSL 情報がありません。

### 解決法

Application Server と MQ との間で SSL 通信が必要な場合には、SSL 構文で明示的に `addresslist` を指定して接続ファクトリを生成します。たとえば、接続ファクトリの `addresslist` は次のようなものになります。

```
mq://mqserver-1:7676/ssljms,mq://mqserver-2:7676/ssljms
```

MQ と Application Server が異なる場所にあつて、それらの間のネットワーク接続が侵入者によって攻撃される可能性がある場合には、Application Server と MQ の間で SSL 通信が必要になることがあります。

## ポート番号なしで指定された https リスナーへの URL が、ポート 80 の http リスナーにリダイレクトされる (ID 6207297)

デフォルトポート (443) で SSL リスナーを有効にした場合、このセキュリティ保護されたポートへの URL をポート番号なしでブラウザで指定すると、ブラウザはセキュリティ保護されていない (http) リスナーのポート 80 にリダイレクトします。

次に例を示します。

1. ポート 443 に SSL リスナーを作成し、Application Server を再起動します。
2. ブラウザから <https://servername:443> にアクセスします。  
該当するページが適切にロードされます。
3. ブラウザから <https://servername> (ポート番号なし) にアクセスします。  
ブラウザは、<https://servername> ではなく、<http://servername:80> をロードします。

この問題は、SSL リスナーがデフォルト (443) 以外のポートにある場合には発生しません。

### 解決法

次のいずれかの解決法を実行します。

- https ポート 443 の SSL リスナーにアクセスする場合には、必ずポート番号を指定するようサイト訪問者に指示するとともに、そのようにすべてのリンクをコーディングします。
- SSL リスナーにはポート 443 を使用しません。

## アップグレードユーティリティ

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

### Application Server Enterprise Edition 8 から Application Server Enterprise Edition 8.1 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。(ID 6165528)

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。

## J2EE 1.4 SDK から Application Server EE 8.1 にアップグレードするときに、付属の J2SE が適切にアップグレードされない。この問題は、スタンドアロンバージョンにのみ影響する。(ID 6196741)

この問題は、J2EE 1.4 SDK から Application Server 8.1 のスタンドアロンバージョンにアップグレードするときにだけ発生します。このアップグレードの過程で、付属の J2SE 1.4.2 を J2SE 5.0 にアップグレードする必要があります。しかし、その結果としての J2SE 5.0 中の JAR ファイルが不適切にアップグレードされ、インストールされた J2SE 5.0 が破壊されます。

インストールプログラムからはアップグレードに成功した旨が報告され、アップグレードツールの実行中に何らかの問題が発生することは予想できません。しかし、それ以降、アップグレードされた Application Server を起動しようとする、次の例外で失敗します。

```
Exception in thread "main"
[#|2004-11-17T18:12:24.033-0800|WARNING|sun-appserver-ee8.1|javax.enterprise.
system.stream.err|_ThreadID=10;|java.lang.NoClassDefFoundError:
javax/net/ssl/TrustManager at
com.sun.enterprise.security.SecurityLifecycle.onInitialization(SecurityLifecy
cle.java:59) at
com.sun.enterprise.server.ApplicationServer.onInitialization(ApplicationServe
r.java:215) at com.sun.enterprise.server.PEMain.run(PEMain.java:277) at
com.sun.enterprise.server.PEMain.main(PEMain.java:219)
```

### 解決法

次のように、この問題の回避策はいくつかあります。

- サポートされている適切なバージョンのスタンドアロン J2SE をインストールします。Application Server のアップグレード時に、Application Server のインストールプログラムの「Java 設定」画面で、Java 2 SDK 5.0 をインストールするデフォルトのオプションではなく、既存の Java 2 SDK を再利用するオプションを選択します。スタンドアロン J2SE インストールのパスを指定します。
- アップグレードを実行する前に、既存の J2EE 1.4 SDK の *install\_dir*/jdk サブディレクトリを削除または名前を変更します。そのあとは、Application Server 8.1 のアップグレード時に J2SE 5.0 が適切にインストールされます。このディレクトリを削除または名前を変更するのは、インストールプログラムのディレクトリ選択画面をナビゲートし、既存のインストールの「代替アップグレード」を許可するプロンプトが表示されたあとです。
- アップグレードを実行したあとでサーバーの起動で問題が発生した場合には、サポートされている適切なバージョンのスタンドアロン J2SE をインストールしてから、*install\_dir*/config/asenv.conf ファイル (Linux および Solaris の場合) または *install\_dir*¥config¥asenv.bat ファイル (Microsoft Windows の場合) の AS\_JAVA 変数を変更します。この変数の値は、新たにインストールされたスタンドアロン J2SE の場所を指している必要があります。

## 一部の Linux システムで、「代替アップグレード」を実行しているインストールプログラムが、「アップグレードウィザードの起動」ボタンのクリック後にアップグレードツールの起動に失敗する。(6207337)

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 で最も一般的ですが、RedHat ディストリビューションでも発生しています。

インストー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ガアップします。

### 解決法

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

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

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

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

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

[file://install\\_dir/docs-ee/about.html](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。

## 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。

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

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

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

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

同様の理由から、配備取り消しをせずにアプリケーションを再配備しよう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
```

## 解決法

アプリケーションを配備するときに `--precompilejsps=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 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>` はサポートされていません。つまり、Servlet 2.4 ベースの `web.xml` を使用する場合には、整数の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` の Servlet 2.4 スキーマに対する妥当性検証で `web.xml` が失敗するため、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。

### 解決法

Servlet 2.4 ベースの `web.xml` を使用する場合、サーブレットをロードする順序が問題にならないことを示すには、`<load-on-startup>0</load-on-startup>` を指定します。

## AMX API を使用すると、サーバーから J2EE アプリケーション参照を削除することによってアプリケーションを削除しても、そのアプリケーションには依然としてアクセスできる。(ID 6173248)

AMX API を使用している場合、明示的にアプリケーションを終了することなくアプリケーションへの参照を削除すると、そのアプリケーションには依然としてアクセスできます。これは設計どおりの動作で、マニュアルには記載されていません。

### 解決法

二度とアクセスできないようにアプリケーションを削除するには、次の手順を実行します。

1. アプリケーションを終了します。
2. そのアプリケーションへの参照を削除します。
3. そのアプリケーションの配備取り消しを行います。

## 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す。

```
at
org.apache.tools.ant.taskdefs.Execute$Java13CommandLauncher.exec(Execute.java
:655) at org.apache.tools.ant.taskdefs.Execute.launch(Execute.java:416) at
org.apache.tools.ant.taskdefs.Execute.execute(Execute.java:427) at
org.apache.tools.ant.taskdefs.compilers.DefaultCompilerAdapter.executeExternalCompile(DefaultCompilerAdapter.java:448) at
```

```
org.apache.tools.ant.taskdefs.compilers.JavacExternal.execute(JavacExternal.java:81) at org.apache.tools.ant.taskdefs.Javac.compile(Javac.java:842) at org.apache.tools.ant.taskdefs.Javac.execute(Javac.java:682) at org.apache.jasper.compiler.Compiler.generateClass(Compiler.java:396)
```

## 解決法

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。

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

```
<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。

---

## 問題の報告および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** - 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は、次の URL にあります。

<http://java.sun.com/docs/forms/J2EE14SubmittalForm.html>

- **J2EE-INTEREST リスト** - J2EE に関する質疑を扱うメーリングリストには、次の URL でアクセスできます。

<http://archives.java.sun.com/archives/j2ee-interest.html>

- **Java Developer Connection のバグデータベース** - バグを閲覧または報告するには、次の URL にある Java Developer Connection Bug Parade をご利用ください。

<http://developer.java.sun.com/servlet/SessionServlet?url=/developer/bugParade/index.jshtml>

## コメントの送付先

- **Java Technology Forums** -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 8.1 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。

<http://forum.java.sun.com/>

- **Sun Software Support Services** - 次の URL にてオンラインでご利用になれます。

<http://www.sun.com/service/sunone/software>

このサイトには、**Knowledge Base**、オンラインサポートセンター、**ProductTracker** へのリンクと、保守プログラムおよびサポートに関する問い合わせ先電話番号が記載されています。

- 保守契約を結んでいるお客様の場合は、専用ダイヤルをご利用ください。

最善の問題解決のため、サポートに連絡する際には次の情報をご用意ください。

- 問題が発生した箇所や動作への影響など、問題の具体的な説明
- マシン機種、OS バージョン、および、問題の原因と思われるパッチやそのほかのソフトウェアなどの製品バージョン
- 問題を再現するための具体的な手順の説明
- エラーログやコアダンプ

---

## 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。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、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、このマニュアルの場合、タイトルは『Sun Java System Application Server Enterprise Edition 8.1 2005Q1 リリースノート』、**Part No.** は 819-2026 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://wwws.sun.com/software/products/appsrvr/home\\_appsrvr.html](http://wwws.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?l=ja#hic/>
- Sun Java System マニュアル  
<http://docs.sun.com/app/docs/prod/entsys?l=ja#hic>
- 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 Sun 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://jp.sunsolve.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://wwws.sun.com/software/training/>
- Sun Software Data Sheets  
<http://wwws.sun.com/software>
- Sun Microsystems の製品マニュアル  
<http://docs.sun.com/>

---

Copyright © 2004-2005 Sun Microsystems, Inc. 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、<http://www.sun.com/patents> の一覧に示される米国特許、および米国をはじめとする他の国々で取得された、または申請中の特許などが含まれています。

SUN PROPRIETARY/CONFIDENTIAL.

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.

使用は、使用許諾契約の条項に従うものとします。

本製品には、サードパーティ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、カリフォルニア大学からライセンスされている Berkeley BSD システムに基づいて開発されている場合があります。

Sun、Sun Microsystems、Sun ロゴ、Java、および Solaris は、米国およびその他の国における Sun Microsystems, Inc. の商標または登録商標です。すべての SPARC 商標は、米国 SPARC International, Inc. のライセンスを受けて使用されている、米国および他の国々における同社の商標または登録商標です。