



Sun GlassFish Communications Server 2.0 リリースノート



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

Part No: 821-1050-10
2009 年 10 月

Sun Microsystems, Inc. (以下米国 Sun Microsystems 社とします) は、本書に記述されている製品に含まれる技術に関連する知的財産権を所有します。特に、この知的財産権はひとつかそれ以上の米国における特許、あるいは米国およびその他の国において申請中の特許を含んでいることがありますが、それらに限定されるものではありません。

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.

この配布には、第三者によって開発された素材を含んでいることがあります。

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いることがあります。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他の国における登録商標です。

un、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、Java および Solaris は、米国およびその他の国における米国 Sun Microsystems 社の商標、登録商標もしくは、サービスマークです。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・ユーザインタ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 のグラフィカル・ユーザインタフェースを実装するか、またはその他の方法で米国 Sun Microsystems 社との書面によるライセンス契約を遵守する、米国 Sun Microsystems 社のライセンス実施権者にも適用されます。

本書で言及されている製品や含まれている情報は、米国輸出規制法で規制されるものであり、その他の国の輸出入に関する法律の対象となることがあります。核、ミサイル、化学あるいは生物兵器、原子力の海洋輸送手段への使用は、直接および間接を問わず厳しく禁止されています。米国が禁輸の対象としている国や、限定はされませんが、取引禁止顧客や特別指定国民のリストを含む米国輸出排除リストで指定されているものへの輸出および再輸出は厳しく禁止されています。

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えないものとします。

概要

Sun GlassFish™ Communications Server は、Sun GlassFish Enterprise Server に Session Initiation Protocol (SIP) Servlet テクノロジーの機能を追加します。

Sun GlassFish Communications Server は開発、配備、および再頒布の用途に無償で使用できます。再頒布を予定している場合は、再頒布ライセンスについて [Sun OEM](#) 販売にお問い合わせください。

Sun GlassFish Communications Server は、Web アプリケーションと Web サービスの開発および配信を行うための、Java™ Platform, Enterprise Edition (Java EE) および SIP テクノロジーに基づいた、使いやすく高速な業界を代表するアプリケーションサーバーです。Sun GlassFish Communications Server では、JSR 289 に準拠した SIP アプリケーションおよび融合アプリケーションを構築できます。ソフトウェア障害やハードウェア障害が発生した場合にも稼働し続けるスケーラブルなキャリアグレードのサービスを実現するために、優れたパフォーマンス、クラスタリング、および高可用性機能を提供します。

- [3 ページの「このリリースノートについて」](#)
- [4 ページの「アクセシビリティ機能」](#)
- [4 ページの「関連マニュアル」](#)
- [5 ページの「問題の報告とフィードバックの方法」](#)
- [6 ページの「このマニュアルに関するコメント」](#)
- [6 ページの「その他の情報」](#)

このリリースノートについて

このリリースノートには、Sun GlassFish Communications Server 2.0 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。Communications Server 2.0 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、Communications Server [ドキュメント Web サイト](#) (<http://docs.sun.com/coll/1343.10>)から入手できます。ソフトウェアをインス

ツールおよび設定する前だけでなく、それ以降も定期的にこの Web サイトをチェックして、最新のリリースノートと製品マニュアルを確認してください。

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

リリースノート改訂履歴

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

表 1-1 リリースノート改訂履歴

改訂日付	説明
2009 年 10 月	Communications Server 2.0 製品の FCS リリース。

アクセシビリティ機能

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

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

関連マニュアル

表 1-2 Communications Server のドキュメントセットの内容

ドキュメント名	説明
『Documentation Center』	タスクや主題ごとに整理された Communications Server のドキュメントのトピック。

表 1-2 Communications Server のドキュメントセットの内容 (続き)

ドキュメント名	説明
『リリースノート』	ソフトウェアとドキュメントに関する最新情報。サポートされているハードウェア、オペレーティングシステム、Java Development Kit (JDK™)、およびデータベースドライバの包括的な表ベースの概要を含みます。
『クイックスタートガイド』	Communications Server 製品の使用を開始するための手順。
『Installation Guide』	ソフトウェアとそのコンポーネントのインストール。
『アプリケーション配備ガイド』	アプリケーションおよびアプリケーションコンポーネントの Communications Server への配備。配備記述子に関する情報を含みます。
『開発者ガイド』	Communications Server 上で動作することを目的とし、Java EE 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 (Java EE プラットフォーム) アプリケーションの作成と実装。開発者ツール、セキュリティ、デバッグ、ライフサイクルモジュールの作成に関する情報を含みます。
Java EE 5 Tutorial	Java EE 5 プラットフォームテクノロジーと API を使用した Java EE アプリケーションの開発。
『Java WSIT Tutorial』	Web サービス相互運用性テクノロジー (WSIT) を使用した Web アプリケーションの開発。WSIT テクノロジーを使用する方法、時期、および理由と、各テクノロジーがサポートする機能およびオプションについて説明します。
『管理ガイド』	設定、監視、セキュリティ、資源管理、および Web サービス管理を含む Communications Server のシステム管理。
『高可用性 (HA) 管理ガイド』	クラスタの設定、ノードエージェントの操作、およびロードバランサの使用法。
『Administration Reference』	Communications Server 設定ファイル domain.xml の編集。
『パフォーマンスチューニングガイド』	パフォーマンスを向上させるための Communications Server の調整。
『Reference Manual』	Communications 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みます。

問題の報告とフィードバックの方法

Sun GlassFish Communications Server に問題が発生した場合は、次のいずれかの方法で Sun にお問い合わせください。

- [SailFin メーリングリスト \(https://sailfin.dev.java.net/servlets/ProjectMailingListList\)](https://sailfin.dev.java.net/servlets/ProjectMailingListList) — 多くの情報やフィードバックを得られる、さまざまな SailFin コミュニティーのメーリングリスト
- [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) — バグを表示または提出するには、Java Developer Connection Bug Parade を使用します。
- [SailFin Issue Tracker \(https://sailfin.dev.java.net/servlets/ProjectIssues\)](https://sailfin.dev.java.net/servlets/ProjectIssues)

- [Java Technology Forums \(http://forum.java.sun.com/\)](http://forum.java.sun.com/) — Java テクノロジおよびプログラミング技術に関する情報や質問を共有するための、対話型のメッセージボード。
- [Sun Software Support Services \(http://www.sun.com/service/sunone/software\)](http://www.sun.com/service/sunone/software) - ナレッジベース、オンラインサポートセンター、Product Tracker、および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
- 保守契約を結んでいるお客様の場合は、専用ダイヤルをご利用ください。最善の問題解決のため、サポートに連絡する際には次の情報をご用意ください。
- 問題が発生した箇所や動作への影響など、問題の具体的な説明
- マシン機種、OS バージョン、および製品のバージョン (問題に関係するパッチおよびその他のソフトウェアを含む)
- 問題を再現するための具体的な手順の説明
- エラーログやコアダンプ

このマニュアルに関するコメント

弊社では、ドキュメントの改善に努めており、お客様からのコメントおよびご忠告をお受けしております。

コメントを共有するには、<http://docs.sun.com> にアクセスして、「Feedback」をクリックします。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、ドキュメント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、このドキュメントのタイトルは『Sun GlassFish Communications Server 2.0 リリースノート』、Part No. は 821-1050-10 です。

その他の情報

以下の場所から有益な情報を入手することができます。

- [Communications Server 製品情報 \(http://www.sun.com/software/products/appsrvr/home_appsrvr.html\)](http://www.sun.com/software/products/appsrvr/home_appsrvr.html)
- [プロフェッショナルサービス \(http://www.sun.com/service/sunps/sunone\)](http://www.sun.com/service/sunps/sunone)
- [ソフトウェア製品とサービス \(http://www.sun.com/software\)](http://www.sun.com/software)
- [ソフトウェアサポートサービス \(http://www.sun.com/service/sunone/software\)](http://www.sun.com/service/sunone/software)
- [サポートとナレッジベース \(http://www.sun.com/service/support/software\)](http://www.sun.com/service/support/software)
- [Sun Support およびトレーニングサービス \(http://training.sun.com\)](http://training.sun.com)
- [コンサルティングと Professional Services \(http://www.sun.com/service/sunps/sunone\)](http://www.sun.com/service/sunps/sunone)
- [開発者向け情報 \(http://developers.sun.com\)](http://developers.sun.com)
- [Sun Developer Support Services \(http://www.sun.com/developers/support\)](http://www.sun.com/developers/support)
- [ソフトウェアトレーニング \(http://www.sun.com/software/training\)](http://www.sun.com/software/training)
- [Sun ソフトウェアデータシート \(http://www.sun.com/software\)](http://www.sun.com/software)

- Sun Microsystems 製品ドキュメント (<http://docs.sun.com/>)

Sun GlassFish Communications Server 2.0 について

Sun GlassFish Communications Server は、大規模な本稼働環境での Web、SIP、および融合アプリケーションと、Java テクノロジベースの Web サービスの開発と配備に利用できる Java EE 5 プラットフォーム互換サーバーです。

この章で説明する内容は、次のとおりです。

- 9 ページの「[Sun GlassFish Communications Server の機能](#)」
- 12 ページの「[ハードウェアとソフトウェアの要件](#)」
- 19 ページの「[Update Center の使用法](#)」

Sun GlassFish Communications Server の機能

Communications Server は、Sun GlassFish Enterprise Server で使用できるすべての機能をサポートし、さらに次のような新しい機能も備えています。Sun GlassFish Enterprise Server の機能一覧については、『Sun GlassFish Enterprise Server リリースノート』を参照してください。

- **JSR 289 への準拠** — Communications Server には、RFC 3261、3262、3265、3311、3515、3903 などに準拠する SIP スタックが含まれます。また、JSR 289 で定義された SIP Servlet Java API も公開されます。
- **アップグレード** — Communications Server は、Communications Server 1.5 から Communications Server 2.0 へのアップグレードをサポートします。
- **SIP 融合および HTTP 融合セッション** — Communications Server は、JSR 289 で定義されているように、HTTP 融合セッションおよび SIP 融合セッション機能をサポートします。
- **DCR プラグインのサポート** — Communications Server は、Java クラスの形式のプラグインを使用して、データ中心ルールの設定をサポートします。
- **過負荷保護** — Communications Server は、CPU ベースおよびメモリーベースの過負荷保護をサポートします。
- **SIP セッションレプリケーション** — Communications Server は、SIP セッションの信頼できるフェイルオーバーを提供します。

- 順次アップグレード — Communications Server は、アプリケーションまたはサーバーのアップグレードをサポートし、サービスやセッションの損失を最小限に抑えます。
- マルチホームのサポート — Communications Server は、2 つ以上の異なるサブネット間の SIP 通信をサポートし、同一サブネット内での冗長ネットワーク接続を提供します。Communications Server は、複数の NIC を持つ環境で 사용할 수 있습니다。
- 統合された **SIP Servlet** コンテナ — SIP Servlet コンテナでは、次の機能を利用できます。
 - ネットワークエンドポイントで SIP リクエストを待機できます。
 - 環境で SIP Servlet のライフサイクルをホストおよび管理できます。

SIP Servlet コンテナは、SSA (SIP Servlet API) 仕様で定義されているように、Web コンテナとともに動作して融合アプリケーションをサポートします。

- 管理の簡単さ — Communications Server では、CLI および GUI を使用して、SIP コンテナと配備された SIP アプリケーションを簡単に管理できます。
- セキュリティ — HTTP および SIP プロトコルの両方で、ダイジェスト認証がサポートされます。SIP プロトコルでは、P-Asserted-Identity がサポートされます。SIP コンテナは SIPS over TLS をサポートします。
- 融合ロードバランサ — 融合ロードバランサ (CLB) は、ソフトウェア負荷分散コンポーネントです。アプリケーションの要求を複数のサーバーインスタンスで構成されたクラスタ内で分散することにより、融合アプリケーションの高可用性をサポートします。CLB には次の機能があります。
 - SIP および HTTP リクエストを、複数のサーバーインスタンスで構成されるクラスタに転送します。
 - 矛盾のないハッシュおよびラウンドロビンポリシーを使用して、リクエストを負荷分散します。
 - クラスタ内のサーバーインスタンスを有効または無効にします。
 - サーバーインスタンスの状態を監視します。
- **SIP** アプリケーションおよび融合アプリケーションの配備のサポート — Communications Server は、GlassFish の拡張モジュールメカニズムを使用して、単独の SIP アプリケーションおよび融合アプリケーションの配備をサポートします。また、Sun 独自の SIP アプリケーションの配備記述子として、`sun-sip.xml` もサポートします。
- **NetBeans** ツール — Communications Server では、NetBeans 6.0 上で JSR 289 に基づく SIP Servlet アプリケーションがサポートされます。テストエージェントを使用して簡単な SIP クライアントを作成し、SIP Servlet をテストできます。このプラグインは、`as-install-dir/lib/tools` にあります。

- アプリケーションルーターの配備 — Communications Server は拡張モジュールメカニズムを使用して、アプリケーションルーターアーカイブ (JAR ファイル) のホット配備をサポートします。デフォルトでは、簡単なアルファベットによるルールに従ってリクエストを経路指定するアプリケーションルーターが用意されています。
- パッケージ化とファイルのレイアウト — Communications Server は、ファイルベースのインストール可能ファイルとして配信されます。
- **Sun Connection** への登録 - インストーラ、管理コンソール GUI、または Update Center を使用して、製品を [Sun Connection](http://www.sun.com/service/sunconnection/index.jsp) (<http://www.sun.com/service/sunconnection/index.jsp>) へ登録できます。Sun Connection で Communications Server を登録すると、次の利点を受けることができます。
 - パッチ情報とバグ更新
 - スクリーンキャストとチュートリアル
 - ニュースとイベント
 - サポートとトレーニングの提供
- **SuSe Linux 64** ビット版のサポート
- **Java EE 5** プラットフォームのサポート — Sun GlassFish Communications Server では、Java EE 5 仕様を実装して、次世代のエンタープライズアプリケーションおよび Web サービスにもっとも適したアプリケーションランタイムの 1 つを実現します。Communications Server は、次の Java EE 標準を実装しています。
 - Enterprise Java Beans 3.0
 - JAXB 2.0
 - Java Persistence
 - Java Server Faces 1.2
 - Java Server Pages 2.1 (JSP 2.1)
 - Java Server Pages Standard Tag Library (JSTL) 1.2
 - Streaming API for XML (StAX)
 - Web Services Metadata
 - Java API for XML based Web Services 2.0 (JAX-WS 2.0)
 - Common Annotations for the Java Platform 1.0 (CAJ 1.0)
 - Java Servlet 2.5

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

- **Web サービス相互運用性テクノロジー (WSIT)** サポート – Sun は Microsoft と密接に連携して、メッセージ最適化、高信頼性メッセージング、およびセキュリティなどの Web サービスエンタープライズテクノロジーの相互運用性を実現しています。WSIT の初期リリースはこの協力の成果です。WSIT は、エンタープライズ機能をサポートする多くのオープンな Web サービス仕様を実装したものです。メッセージ最適化、信頼できるメッセージング、およびセキュリティに加えて、WSIT にはブートストラップと設定のテクノロジーも含まれています。現在 Java プラットフォームに組み込まれているコア XML サポートを基本にし

て、WSIT は既存の機能を使用または拡張し、相互運用可能な Web サービスのための新しいサポートを追加します。それらには、次のサポートが含まれます。

- ブートストラップおよび設定
- メッセージ最適化テクノロジー
- 高信頼性メッセージングテクノロジー
- セキュリティーテクノロジー
- **JBISupport** – JBI はビジネス統合 Service Provider Interfaces (SPI) によって Java EE を拡張します。これらの SPI を使用することにより、開発者は WSCI、BPEL4WS、および W3C Choreography Working Group などの仕様に合わせ Java ビジネス統合環境を作成または実装できます。JBI 実装は直接 Communications Server インストーラにより通常は *as-install/jbi* ディレクトリにインストールされます。Communications Server JVM で JBI フレームワークを開始する lifecycle モジュールを含む、JBI 用のすべての共通 JAR ファイルとシステムコンポーネントがこのディレクトリに含まれています。
- **使用状況プロファイル** – どの管理ドメインもそのドメインの能力を示す使用状況プロファイルと関連付けられています。Communications Server には次のプロファイルが用意されています。
 - **開発者** – ドメインを開発環境で実行している場合や、アプリケーションに負荷分散、高可用性、セッションレプリケーションなどのクラスタ化機能が必要でない場合に、このプロファイルを使用します。このプロファイルの実際の名前は「developer」で、大小文字を区別することに注意してください。
 - **クラスタ** – アプリケーションサーバーインスタンスのクラスタを作成して、配備された Java EE アプリケーションにスケラビリティと高可用性を付与する場合は、このプロファイルを使用します。アプリケーションの状態はインメモリーのままとなります。このプロファイルの実際の名前は「cluster」で、大小文字を区別することに注意してください。
- **オープンソースおよび SailFin コミュニティー** – [SailFin \(http://sailfin.dev.java.net\)](http://sailfin.dev.java.net) コミュニティーは、Java EE 5 プラットフォームの最新機能とそれに関連したエンタープライズテクノロジーを実装した、無償でオープンソースの商用レベルのコミュニケーションサーバーを開発することを目標としています。
- **Update Center サポート** – Communications Server Update Center により、Communications Server の更新が自動化され、追加コンポーネントが簡単に利用できるようになりました。

ハードウェアとソフトウェアの要件

この節では、Sun GlassFish Communications Server 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。

- [13 ページの「サポートされるプラットフォーム」](#)
- [14 ページの「重要パッチ情報」](#)
- [14 ページの「JDK バージョン」](#)

サポートされるプラットフォーム

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

表2-1 サポートされるオペレーティングシステム

オペレーティングシステム	最小メモリー	推奨メモリー	最小ディスク容量	推奨ディスク容量	JVM	配備のタイプ
Sun Solaris 10 (x86) (32 ビットおよび 64 ビット)	512M バイト	512M バイト	250M バイトの空き領域	500M バイトの空き領域	J2SE 5.0 - 32 ビット Java SE 6 - 32 ビットおよび 64 ビット	本稼働
Sun Solaris 10 (SPARC) (32 ビットおよび 64 ビット)	512M バイト	512M バイト	250M バイトの空き領域	500M バイトの空き領域	J2SE 5.0 - 32 ビット Java SE 6 - 32 ビットおよび 64 ビット	本稼働
Red Hat Enterprise Linux 3.0 アップデート 1、4.0、および 5.x (32 ビットおよび 64 ビット)	512M バイト	1G バイト	250M バイトの空き領域	500M バイトの空き領域	J2SE 5.0 - 32 ビット Java SE 6 - 32 ビットおよび 64 ビット	本稼働
SuSE Linux Enterprise Server 10 (32 ビットおよび 64 ビット) (SP1 および SP2 もサポートされています)	512M バイト	1G バイト	250M バイトの空き領域	500M バイトの空き領域	J2SE 5.0 - 32 ビット Java SE 6 - 32 ビットおよび 64 ビット	本稼働

表 2-1 サポートされるオペレーティングシステム (続き)

オペレーティングシステム	最小メモリ	推奨メモリ	最小ディスク容量	推奨ディスク容量	JVM	配備のタイプ
Windows XP Windows Vista Windows 2008	1G バイト	2G バイト	500M バイトの空き領域	1G バイトの空き領域	J2SE 5.0 - 32 ビット Java SE 6 - 32 ビットおよび 64 ビット	開発
Windows 7	1G バイト	2G バイト	500M バイトの空き領域	1G バイトの空き領域	J2SE 5.0 - 32 ビット Java SE 6 - 32 ビットおよび 64 ビット	開発
Macintosh OS 10.4、10.5	512M バイト	1G バイト	250M バイトの空き領域	500M バイトの空き領域	J2SE 5.0 - 32 ビット Java SE 6 - 32 ビット	開発
OpenSolaris 2009.06	512M バイト	1G バイト	250M バイトの空き領域	500M バイトの空き領域	J2SE 5.0 - 32 ビット Java SE 6 - 32 ビットおよび 64 ビット	開発

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

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

重要パッチ情報

Solaris のパッチ要件

Solaris 10 (x86 または SPARC) を使用する場合には、「Sun 推奨パッチクラスタ」をインストールすることをお勧めします。このパッチクラスタは、[SunSolve \(http://sunsolve.sun.com/pub-cgi/show.pl?target=patchpage\)](http://sunsolve.sun.com/pub-cgi/show.pl?target=patchpage) Web サイトの「推奨 & セキュリティーパッチ」で入手できます。

JDK バージョン

Communications Server で必要な (動作が保証されている) JDK の最小バージョンは、**1.6.0_7** です。

▼ サポートされている **Java SE** バージョンへ切り換える

次の手順で `asenv file` を編集して、サポートされている Java SE バージョンに切り替えることができます。

- 1 システムに新しい **Java SE** バージョンをインストールします (インストールしていない場合)。
Java SE SDK は <http://java.sun.com/javase> からダウンロードできます。
- 2 **Communications Server** ドメインを停止します。
- 3 `as-install/config/asenv.conf` ファイル (**Windows** では `asenv.bat`) を編集して、新しい **Java** 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、`com.sun.aas.javaRoot...` で始まる行を、新しい **Java** ホームディレクトリを参照するように変更します。
- 5 **Communications Server** を再起動します。

Ant のバージョン

SailFin JAR インストールにバンドルされた Ant のバージョンは **1.6.5** です。カスタムのアプリケーションルーターアプリケーションを構築するには、**Ant 1.7** が必要です。このバージョンは、別途ダウンロードする必要があります。

JDBC ドライバとデータベース

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

表 2-2 Java EE 互換の JDBC ドライバ

JDBC ドライバベンダー	JDBC ドライバのタイプ	サポートされるデータベースサーバー
Derby Network Client	Type 4	Derby 10.2

表 2-2 Java EE 互換の JDBC ドライバ (続き)

JDBC ドライバベンダー	JDBC ドライバのタイプ	サポートされるデータベースサーバー
DataDirect 3.6.x、3.7.x	Type 4	Oracle 10g Oracle 9i Sybase ASE 12.5、15 MS SQL 2000、2005 DB2 9.1 DB2 8.1、8.2
MySQL Connector/J Driver 5.1.6	Type 4	MySQL 5.0
Oracle 10g、11g	Type 4	Oracle 10g、11g -RAC
PostGres	Type 4	8.2.x

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

Message Queue のバージョン

ファイルベースのインストールでは、Message Queue 4.4 が Communications Server と一緒にパッケージされています。

Communications Server の一部として提供される埋め込みの Sun GlassFish Message Queue コードは、一般的に同等のバージョンの Message Queue ブローカに対してのみ、テストおよび認証されています。つまり、提供される埋め込みの Message Queue コードを使用して、別のバージョンの Message Queue コードを実行しているリモート (Communications Server で管理されていない) Message Queue ブローカに接続することは、サポートされません。

ブラウザ

ここでは、Sun GlassFish Communications Server 2.0 でサポートされているブラウザを示します。

表 2-3 サポートされている Web ブラウザ

ブラウザ	バージョン
Mozilla	1.7.12
Internet Explorer	6.0 Service Pack 2、7.0

表 2-3 サポートされている Web ブラウザ (続き)

ブラウザ	バージョン
Firefox	2.x、3.x
Safari	3.x、4.x
Netscape	8.0.4、8.1、9.0、9.0.x

その他の要件

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

- **空き領域:** ユーザーの一時ディレクトリには、Sun GlassFish Communications Server インストールの場合、35MB 以上の空き領域が必要です。
- **アンインストールプログラムの使用:** システムから Communications Server を削除する必要がある場合は、ソフトウェアに付属のアンインストールプログラムを使用することが重要です。ほかの方法を使用して削除すると、同じバージョンの再インストールや新しいバージョンのインストールを行うときに問題が発生する可能性があります。
- **空きポート:** 利用可能な未使用ポートが7つ必要です。インストールプログラムは、使用されているポートを自動的に検出し、その時点で未使用のポートをデフォルトのドメインとして示します。デフォルトの初期ポートは次のとおりです。
 - HTTP については 8080、HTTPS については 8181
 - IIOP については 3700、IIOP/SSL については 3820、相互認証の IIOP/SSL については 3920
 - Administration Server については 4848 (HTTP)
 - pure JMX Client については 8686 (RMI) ポート
- **ファイアウォールのシャットダウン (Microsoft Windows) —** Sun GlassFish Communications Server ソフトウェアをインストールする前に、すべてのファイアウォールソフトウェアを停止する必要があります。一部のファイアウォールソフトウェアはデフォルトですべてのポートを無効にするからです。インストールプログラムは、使用可能なポートを正確に検出する必要があります。

Java EE 5 プラットフォームの API

Sun GlassFish Communications Server 2.0 は、Java EE 5 プラットフォームをサポートしています。Java EE 5 プラットフォームで利用できる拡張 API を、次の表に示します。

表 2-4 Java EE 5 プラットフォームの主要な API の変更点

API	JSR
Java EE 5	
Java Platform, Enterprise Edition 5 (http://java.sun.com/javaee/5/docs/api/)	JSR 244 (http://jcp.org/aboutJava/communityprocess/pr/jsr244/)
Web サービステクノロジー	
エンタープライズ Web サービスの実装	JSR 109 (http://jcp.org/en/jsr/detail?id=109)
Java API for XML-Based Web Services (JAX-WS) 2.0 (https://jax-ws.dev.java.net/)	JSR 224 (http://jcp.org/en/jsr/detail?id=224)
Java API for XML-Based RPC (JAX-RPC) 1.1 (https://jax-rpc.dev.java.net/)	JSR 101 (http://jcp.org/en/jsr/detail?id=101)
Java Architecture for XML Binding (JAXB) 2.0 (https://jaxb.dev.java.net/)	JSR 222 (http://jcp.org/en/jsr/detail?id=222)
SOAP with Attachments API for Java (SAAJ) (https://saaj.dev.java.net/)	JSR 67 (http://jcp.org/en/jsr/detail?id=67)
Streaming API for XML (http://java.sun.com/webservices/docs/1.6/tutorial/doc/SJSXP.html)	JSR 173 (http://jcp.org/en/jsr/detail?id=173)
Java プラットフォーム用 Web サービスメタデータ	JSR 181 (http://jcp.org/en/jsr/detail?id=181)
コンポーネントモデルテクノロジー	
Enterprise JavaBeans 3.0 (http://java.sun.com/products/ejb/)	JSR 220 (http://jcp.org/en/jsr/detail?id=220)
J2EE Connector Architecture 1.5 (http://java.sun.com/j2ee/connector/)	JSR 112 (http://jcp.org/en/jsr/detail?id=112)
Java Servlet 2.5 (http://java.sun.com/products/servlet/)	JSR 154 (http://jcp.org/en/jsr/detail?id=154)
JavaServer Faces 1.2 (http://java.sun.com/j2ee/javaserverfaces/)	JSR 252 (http://jcp.org/en/jsr/detail?id=252)
JavaServer Pages 2.1 (http://java.sun.com/products/jsp/)	JSR 245 (http://jcp.org/en/jsr/detail?id=245)
JavaServer Pages Standard Tag Library 1.2 (http://java.sun.com/products/jsp/jstl/)	JSR 52 (http://jcp.org/en/jsr/detail?id=52)
管理テクノロジー	
J2EE Management (http://java.sun.com/j2ee/tools/management/)	JSR 77 (http://jcp.org/en/jsr/detail?id=77)

表 2-4 Java EE 5 プラットフォームの主要な API の変更点 (続き)

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

Update Center の使用法

Update Center により、Communications Server の更新が自動化され、追加コンポーネントが簡単に利用できるようになりました。

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

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

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

▼ Update Center を使用するには

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

- 1 Update Center を開始します。
 - UNIX の場合: <installdir>/updatecenter/bin/updatetool
 - Windows の場合: <installdir>\updatecenter\bin\updatetool.bat
- 2 「Preferences」 タブをクリックします。
- 3 「Update Scheduling」 ウィンドウで、「Check for Updates」 ドロップダウンボックスの値を「Never (Manual)」から希望する値に変更します。一例として、毎日や毎週などを指定できます。
- 4 アップデートを希望する曜日と時刻を指定します。
- 5 変更を保存するには、「Save」 ボタンを選択します。

これで、Update Center は指定したスケジュールに従って、SailFin または Communications Server コンポーネントのアップデートを自動的に確認します。アップデートが利用可能である場合には、Update Center が起動して更新可能なコンポーネントを通知します。

WSIT 統合の詳細について

WSIT ステータスの詳細は、[WSIT Status Notes \(https://wsit.dev.java.net/source/browse/*checkout*/wsit/wsit/status-notes/status-notes-1-0-1-FCS.html\)](https://wsit.dev.java.net/source/browse/*checkout*/wsit/wsit/status-notes/status-notes-1-0-1-FCS.html) ページを参照してください。

Sun GlassFish Communications Server の既知の問題点と制限事項

ここでは、Communications Server ソフトウェアに関する既知の問題とそれに関連する解決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。

- 21 ページの「Communications Server の管理」
- 24 ページの「融合ロードバランサ」
- 24 ページの「インストール」
- 25 ページの「セキュリティー」
- 26 ページの「SIP コンテナ」
- 29 ページの「SIP セッションレプリケーション」

Communications Server の管理

Communications Server がクラスタのハートビートポートの競合を検出しない (Issue Number 1967)

説明

クラスタを作成するときに、Communications Server はハートビートポートを 1026 - 45556 の間でランダムに割り当てます。default-cluster (Communications Server のインストールが作成するデフォルトのクラスタ) では、0 - 45556 のランダムな数字が選択されます。クラスタの作成プロセスは、ハートビートポートがすでに別のサービスで使用されているかどうかを正確に検出しません。

解決方法

自動クラスタ作成の設定で選択されるハートビートポートが、すでにそのポートを使用している別のサービスと競合する場合は、クラスタのハートビートポートをシステムが使用していないポートに更新します。

クラスタのハートビートポートを変更するには、次の `asadmin` コマンドを使用します。

```
asadmin set cluster-name.heartbeat-port= newportnumber
```

64 ビット Linux で動作している NFS サーバーで、ドメインの作成が停止する (Issue Number 1961)

説明

64 ビット Linux で動作している NFS サーバーで、NFS (ネットワークファイルシステム) マウント済みファイルシステム上にドメインを作成しているときに、`asadmin create-domain` コマンドが失敗する場合があります。

解決方法

現在のところ解決策はありません。

トラフィックが非常に少ないかまったくない場合に、CPU 使用量が高い (Issue Number 1966)

説明

CPU 過負荷保護が有効になっている場合に、トラフィックが非常に少ないかまったくないのに、Communications Server のインスタンスが高い CPU 使用量を示すことがあります。この問題は、JDK バグ [6693490](http://bugs.sun.com/bugdatabase/view_bug.do?bug_id=6693490) (http://bugs.sun.com/bugdatabase/view_bug.do?bug_id=6693490) が原因で発生します。このバグは JDK 6 Update 18 で解決されています。

解決方法

&ProductName で JDK 6 Update 18 を使用してください。

SIP/SIPS ポートがバインドされなくても Communications Server インスタンスが起動する (Issue Number 998)

説明

SIP または SIPS ポートをバインドできない場合でも、Communications Server インスタンスが起動します。

解決方法

サーバーインスタンスを起動する前に、ポートが開放されていることを確認します。ログファイル (server.log) で、起動時に SIP コンテナのエラーや例外が発生していないことを確認します。

Communications Server が、—javahome オプションで指定した JDK を使用しない (Issue Number 789)

説明

—javahome オプションを使用すると、デフォルトのバージョンの代わりにインストール済みの JDK をインストールで使用できます。デフォルトでは、Communications Server は *as-install/jdk* の JDK バージョンを使用します。

解決方法

asenv.conf ファイルの *AS_JAVA* 変数は、常に *as-install/jdk* をポイントします。別のバージョンの JDK を使用する場合は、*asenv.conf* ファイルを手動で更新し、*AS_JAVA* の値を変更します。

3.5G バイトの Java ヒープを使用すると、トラフィックが存在する間にインスタンスが再起動する (Issue 1169)

説明

JVM ヒープサイズが 3.5G バイトに設定されている場合、トラフィックを受信するとインスタンスでエラーが発生し再起動します。

解決方法

最大の JVM ヒープサイズが 3.0G バイト以下に設定されていることを確認します。

マルチコアシステムで 1 つのコアだけを使用しているときに、Communications Server による CPU 使用状況のレポートが正しくない (Issue 1344)

説明

Solaris プラットフォームでは、Communications Server は CPU 使用状況を、利用可能なプロセッサの数とコアごとの CPU 使用状況に基づいて計算します。このとき、Communications Server は JVM が使用しているコアの数ではなく、コアの数の静的な値を使用して計算を行います。

解決方法

マシンのすべてのコアが使用されていない場合は、CPU のしきい値を再計算してください。

融合ロードバランサ

アプリケーション配備のあとに融合ロードバランサの動的再構成によって、ログに **SEVERE** メッセージが記録される (**Issue 1161**)

説明

ターゲットで融合ロードバランサの構成を変更してアプリケーションを再配備する場合、インスタンスのログに **SEVERE** メッセージが記録されます。

解決方法

これらのメッセージは、融合ロードバランサやその他のインスタンスの動作に影響しません。メッセージを無視してください。

完全 **URI** を使用すると、**Contact** ヘッダーの **BEKey** パラメータが正しくエスケープされない (**Issue 1466**)

説明

融合ロードバランサで、**BEKey** パラメータに完全な **URI** を返すデータ中心ルールファイルを使用すると、**Contact** ヘッダーの **BEKey** パラメータが正しくエスケープされません。RFC 3261 で説明されているように、「:」文字は正しくエスケープされません。

解決方法

現在のところ解決策はありません。

インストール

Communications Server のファイルベースインストーラが **Basic3pcc** サンプルアプリケーションをインストールしない (バグ番号 **6894932**)

説明

Communications Server のファイルベースインストーラは **Basic3pcc** サンプルアプリケーションをインストールしません。このアプリケーションは **JAR** インストーラで利用可能になります。

解決方法

現在のところ解決策はありません。

Communications Server インストーラが **Linux** でクラッシュする (**6739013**)

説明

この問題は、環境変数 **MALLOC_CHECK_** に 2 を設定して **Linux** を実行しているシステムで観察されています。

解決方法

環境変数 `MALLOC_CHECK_` を 0 に設定してください。次のいずれかのコマンドを実行します。

- Bourne シェルの場合

```
MALLOC_CHECK_=0;
export MALLOC_CHECK_
```

- bash シェルの場合

```
export MALLOC_CHECK_=0
```

- csh、tcsh シェルの場合

```
setenv MALLOC_CHECK_ 0
```

64 ビット JDK によるインストールが失敗する (6796171)

説明

64 ビット JDK をインストールしている 64 ビットシステムでは、インストーラが 64 ビット JDK の使用を試みるため、インストールが失敗します。

解決方法

Sun GlassFish Communications Server を 64 ビットシステムにインストールする場合は、使用している 64 ビットマシンで 32 ビット JDK をダウンロードし、この JDK を使用して Sun GlassFish Communications Server をインストールしてください。次のコマンドを使用する必要があります。 `./distribution_filename -javahome path to 32-bit JDK location`

インストールのあと、Sun GlassFish Communications Server で 64 ビット JDK を使用するには、`asenv.conf` ファイルの `AS_JAVA` 変数を、64 ビット JDK のインストールをポイントするように編集します。

セキュリティ

trust-auth-realm-ref プロパティが `sun-sip.xml` に指定されていない場合、**Communications Server** が例外をスローする (CR 6786131)

説明

`sun-sip.xml` で `P-Asserted-Identity` が設定されているときに、レルムが設定されていないと、**Communications Server** が Null ポインタの例外をスローします。

解決方法

`sun-sip.xml` で `trust-auth-realm-ref` プロパティを使用してレルムを設定します。

SIP コンテナ

SIP コンテナが 100 の応答を送信したときに CANCEL を処理できない (Issue 712)

説明

100 の応答が送信された場合、SIP コンテナは CANCEL 要求を処理できません。

解決法

アプリケーションはプロビジョナルな応答 (1xx など) を送信して、リモート側が INVITE 要求に対して CANCEL を実行できるようにする必要があります。

SIP セッションおよび HTTP セッションに、同じセッション有効期限モデルが適用されない (Issue 1180)

説明

SIP セッションのセッション有効期限モデルは、HTTP の有効期限ロジックと異なります。HTTP の場合、HTTP セッション内で新しい HTTP 要求を受信すると、アプリケーションによる制御を必要とせずに、自動的にセッションが延長されます。

SIP セッションの場合、SIP コンテナによって許可されれば、アプリケーションが `SipApplicationSession (SAS)` の期間を管理します。アプリケーションは `setExpires` メソッドを使用して、SAS の有効期限を指定できます。`setExpires` は、`setExpires` メソッドが呼び出された時点からの相対時間で有効期限を定義します。コンテナは、`setExpires` で指定された期限を変更、拒否、または承諾することができます。セッションが無効になっていない場合は、`setExpires` で定義された時間に、`sessionExpired` のコールバックが実行されます。このコールバックで、再度コンテナによって変更、拒否、または承諾されれば、アプリケーションは新しい `setExpires` を呼び出して、SAS の期限の延長を試みることができます。

このため、HTTP セッション上で同じ有効期限の `SipApplicationSession (SAS)` で開始される融合アプリケーションは、HTTP セッションが新しいリクエストを受信した場合、HTTP セッションの前に SAS の有効期限が切れることを通知されます。

解決方法

SIP セッションおよび HTTP セッションを処理する有効期限が異なる場合は、アプリケーションセッションが存続すると予想される (いくつかの HTTP リクエストを含む)、十分な長さの SAS 有効期限を指定して開始することが最善の方法です。特に、`invalidateWhenReady` セマンティクスが使用されている場合は、SAS の有効期限を無期限に設定することもできます。この場合、最後のプロトコルの子セッションが無効になるときに、`SipApplicationSession` が無効になります。SAS の初期有効期限は、配備記述子で設定できます。

最大の総期間を前もって予測できて、追加のコードが必要ない場合は、SAS の期限が切れるときに SIP セッションと HTTP セッションの両方を無効にするのが適切です。最大期間を予測できない場合は、次のコードスニペットのように、SipApplicationSession を延長できます。

SipApplicationSessionListener の実装で、次のような処理を行えます。

```
public void sessionExpired(SipApplicationSessionEvent sasEvent) {
    // check if the SAS needs to be extended first, if so:
    int granted = sasEvent.getApplicationSession().setExpires(2);
    if (granted <= 0) {
        System.out.println("extension rejected");
    } else if (granted < 2) {
        System.out.println("extension granted with lower value " + granted);
    } // else allowed
}
```

コンテナが **sessionExpired** にコールバックしたあとに SIP セッションが存続する (Issue 1265)

説明

これはときどき発生する問題です。セッションが削除されていることを示す NOTIFY に対する 200 と、NOTIFY を受信したときにクライアントが送信する SUBSCRIBE に競合状態がある場合、SIP コンテナは、「481 Call/Transaction does not exist」メッセージの代わりに「500 Server internal error」メッセージを返すことがあります。

解決方法

クライアントはサブスクリプションが期限切れになる前に、SUBSCRIBE を更新する必要があります。

Communications Server が最初に **UAS** として、続いてプロキシとして動作し、**NOP** を生成する (Issue 1432)

説明

Communications Server は INVITE リクエストを受信すると、最初に UAS として動作し、このリクエストに 1XX で応答します。次に、この INVITE リクエストを別のインスタンスにプロキシ処理し、このインスタンスが 200 OK で応答します。1xx は内部仮想ブランチを作成しますが、200 メッセージは実ブランチを作成します。B から 200 OK を受信したら、内部仮想ブランチは取り消されるべきです。

解決方法

この例外トレースは、仮想プロキシブランチの動作に影響しません。

getLastAccessedTime メソッドが正確な結果を返さない (Issue 1351)**説明**

SIP セッションの `getLastAccessedTime` メソッドが、正確な結果を返しません。

解決方法

`lastAccessedTime` を正確に追跡する必要があるアプリケーションは、自身でこれを `SipApplicationSession` に保存する必要があります。

```
synchronized (sas) {
    Long last = (Long) sas.getAttribute("myLastAccessedTime");
    if (last == null) {last = 0};
    // do something with the last one
    // and...
    // set the new one.
    sas.setAttribute("myLastAccessedTime", System.currentTimeMillis());
}
```

削除した SIP リスナーが一定期間アクティブなままになる (Issue 1249)**説明**

TCP および UDP リクエストに対して設定した SIP リスナーを削除したあと、リスナーが一定期間アクティブなままになります。リスナーに送信された UDP リクエストが、リスナーからの応答を受信する場合があります。

解決方法

現在のところ解決策はありません。SIP リスナーは一定期間のあと、UDP リクエストの待機を終了します。この問題は TCP リクエストには影響しません。

「<>」のない Contact ヘッダーを受信すると、Communications Server が例外をスローする (Issue 1489)**説明**

Communications Server は、「<>」がない Contact ヘッダーを受信したときに例外をスローします。SIP RFC 3261 では、アドレスでの「<>」の使用は必須ではありません。これにより、ほかの SIP 互換デバイスで相互運用性の問題が発生する場合があります。

解決法

Contact ヘッダーでは「<>」を使用してください。

無効な UUID 値で **Communications Server** が例外をスローする (Issue 1494)

説明

Communications Server は、無効な UUID 値で 400 Bad Request を返す代わりに例外をスローします。UUID 値は SIP Contact ヘッダーの sip.instance 値にあります。

解決方法

現在のところ解決策はありません。

Windows: Communications Server で UDP メッセージが受信されない (ID なし)

説明

この問題は Windows のみで、ときどき発生します。UDP メッセージが Communications Server で受信されません。

解決方法

JVM オプションを次のように設定して、Communications Server を再起動します。

```
org.jvnet.glassfish.comms.disableUDPSourcePort=true
```

SIP セッションレプリケーション

融合アプリケーションが **SAS** オブジェクトを同期ロックとして使用すると、デッドロックの可能性がある (Issue Number 1954)

説明

HTTP および SIP サーブレットを持つ融合アプリケーションが sipApplicationSession オブジェクトをロックとして使用して、SIP および HTTP ワークスレッド間のアクセスを同期させている場合、デッドロックが起こります。

解決方法

sipApplicationSession を同期ロックとして使用しないでください。その代わりに、Serializable オブジェクトを sipApplicationSession で属性として設定して、ロックとして使用してください。

Sun GlassFish Enterprise Server の既知の問題と制限事項

この章では、Sun GlassFish Enterprise Server 2.1.1 ソフトウェアに関する既知の問題とそれに関連する解決策について説明します。これらの問題は、Sun GlassFish Enterprise Server 2.1 上に構築された Communications Server にも影響を与えます。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適用されます。この節は次の項目から構成されています。

- 31 ページの「管理」
- 38 ページの「アプリケーションクライアント」
- 39 ページの「付属の Sun JDBC ドライバ」
- 41 ページの「配備」
- 41 ページの「マニュアル」
- 42 ページの「EJB」
- 42 ページの「Java EE Tutorial」
- 43 ページの「Java Persistence」
- 43 ページの「ライフサイクル管理」
- 45 ページの「ロギング」
- 45 ページの「メッセージキュー」
- 47 ページの「監視」
- 48 ページの「セキュリティ」
- 49 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。

Communications Server がクラスタのハートビートポートの競合を検出しない (Issue Number 1967)

説明

クラスタを作成するときに、Communications Server はハートビートポートを 1026-45556 の間でランダムに割り当てます。default-cluster (Communications Server のイン

ストールが作成するデフォルトのクラスタ)では、0 - 45556 のランダムな数字が選択されます。クラスタの作成プロセスは、ハートビートポートがすでに別のサービスで使用されているかどうかを正確に検出しません。

解決方法

自動クラスタ作成の設定で選択されるハートビートポートが、すでにそのポートを使用している別のサービスと競合する場合は、クラスタのハートビートポートをシステムが使用していないポートに更新します。

クラスタのハートビートポートを変更するには、次の `asadmin` コマンドを使用します。

```
asadmin set cluster-name.heartbeat-port= newportnumber
```

64 ビット Linux で動作している NFS サーバーで、ドメインの作成が停止する (Issue Number 1961)

説明

64 ビット Linux で動作している NFS サーバーで、NFS (ネットワークファイルシステム) マウント済みファイルシステム上にドメインを作成しているときに、`asadmin create-domain` コマンドが失敗する場合があります。

解決方法

現在のところ解決策はありません。

サイズの大きなログファイルをローテーションするときにパフォーマンスが低下する (6718611)

説明

サイズの大きなログファイルをローテーションするときに、応答時間にわずかな増加が見られます。

解決方法

「ロガー」設定で「ファイルローテーション制限」と「ファイルローテーションの制限時間」の値を変更することで、パフォーマンスの低下を最小化できます。これらのプロパティの値は、アプリケーションおよび環境に応じて決定してください。

IBM MQ に対して Generic RA Resource Adapter を配備できない (Issue 6605)

説明

IBM Message Queue 製品に対する Generic RA アダプタの配備が失敗します。server.policy ファイルで付与される権限は、次のようになります。


```
grant {
    permission java.util.logging.LoggingPermission "control";
    permission java.util.PropertyPermission "*", "read,write";
}
```

解決方法

server.policy ファイル中の権限を、次のように変更します。

```
grant codeBase
"file:${com.sun.aas.installRoot}/lib/install/applications/adminapp/-" {
    permission java.util.logging.LoggingPermission "control";
};
```

スタンドアロンのインスタンスがほかのインスタンスからファイルを取得することがある (6698604)

説明

一部の環境で、特定のインスタンスと同期する目的で DAS にインストールしたファイルが、実際には別のインスタンスに送信されます。

解決方法

現在のところ解決策はありません。

start-cluster コマンドからの起動メッセージが冗長である (6728317)

説明

asadmin start-cluster コマンドが、起動中に重要でないコンポーネントのエラーでも多量のメッセージを表示します。重要でない(クラスタ内のインスタンスに関連する)要素でエラーが発生した場合の、コマンド出力の例を示します。

```
./asadmin start-cluster --port 9898 cluster1
Please enter the admin user name>admin
Please enter the admin password>
The clustered instance, instance2, was successfully started.
error 0 [#|2008-07-17T14:58:16.496+0200|WARNING|sun-appserver9.1|javax.jms|
_ThreadID=10;_ThreadName=main;
_RequestID=90bbbe3a-d654-4480-b295-7e317d945a4a;|[C4003]:
Error occurred on connection creation [localhost:37676]. - cause:
java.net.ConnectException: Connection refused|#]
```

```
error 1 [#|2008-07-17T14:58:17.517+0200|WARNING|sun-appserver9.1|javax.jms|
_ThreadID=10;_ThreadName=main;
_RequestID=90bbbe3a-d654-4480-b295-7e317d945a4a;|[C4003]:
```

```
Error occurred on connection creation [localhost:37676]. - cause:
java.net.ConnectException: Connection refused|#]
```

```
error 2 [#|2008-07-17T14:58:30.596+0200|WARNING|sun-appserver9.1|
javax.enterprise.system.container.ejb|
_ThreadID=13; ThreadName=pool-1-thread-4;TimerBean;
_RequestID=5954a044-df06-4a3e-902a-0c40b4b6cddb;
|EJB5108:Unable to initialize EJB Timer Service.
The likely cause is the database has not been
started or the timer database table has not been created.|#]
```

```
error 3 [#|2008-07-17T14:58:32.512+0200|WARNING|sun-appserver9.1|
javax.enterprise.resource.resourceadapter|_ThreadID=10;_ThreadName=main;
__CallFlowPool;_RequestID=90bbbe3a-d654-4480-b295-7e317d945a4a;|
RAR5005:Error in accessing XA resource with JNDI name [__CallFlowPool] for recovery|#]
```

```
The clustered instance, instance1, was successfully started.
error 0 [#|2008-07-17T14:58:21.117+0200|WARNING|sun-appserver9.1|
javax.enterprise.system.container.ejb|
_ThreadID=13; ThreadName=pool-1-thread-4;TimerBean;
_RequestID=30827d9a-72ac-4854-b216-06494b6a9fb5;
|EJB5108:Unable to initialize EJB Timer Service. The likely cause is the database has
not been started or the timer database table has not been created.|#]
```

```
error 1 [#|2008-07-17T14:58:23.106+0200|WARNING|sun-appserver9.1|
javax.enterprise.resource.resourceadapter|
_ThreadID=10;_ThreadName=main;__CallFlowPool;
_RequestID=b41d76fa-0203-49f7-a2ae-83bf242d3e7a;
|RAR5005:Error in accessing XA resource with JNDI name [__CallFlowPool] for recovery|#]
```

Command start-cluster executed successfully.

解決方法

現在のところ解決策はありません。これらの例外メッセージは無視できます。

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

説明

デフォルトでは、*as-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。

1. domain1 はそのままにしておき、その前後に別のドメインを作成します。
2. domain1 を削除し、*as-install/lib/package-appclient.xml* 内のハードコードされた domain1 の値を、新しいドメイン名で置き換えます。

domain1 がない場合、新たなドメインが作成されるたびにこれを行う必要があります。

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

説明

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

次に例を示します。

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

JMX プロパティを設定してからサーバーを起動すると、Application Server VM 内に新たな jmx-connector サーバーが起動します。この場合は、望ましくない副作用の 1 つとして、管理機能が悪影響を受け、Application Server の管理コンソールやコマンド行インタフェースで予期しない結果が発生することがあります。問題は、組み込みの jmx-connector サーバーと新たな jmx-connector サーバーとの間で衝突が発生することにあります。

解決方法

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

サーバーの起動時に、次に示すような行が *server.log* に記録されます。ここで指定されている JMXService の URL に接続し、資格を正常に指定した後、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.[#]

マニュアルに .asadmintruststore ファイルが記述されていない (6315957)

説明

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

解決方法

- 可能であれば、そのサーバーをインストールしたユーザーが `asadmin start-domain domain1` コマンドを実行してください。
- そのユーザーがこのコマンドを実行できない場合は、.asadmintruststore を、インストールしたユーザーの home ディレクトリから実行中のユーザーの home ディレクトリに移動またはコピーしてください。
- このファイルをインストールユーザーの home ディレクトリから実行中のユーザーの home ディレクトリに(コピーではなく)移動した場合、アップグレードまたはインストールしたユーザーのホームディレクトリ (Java ES では、通常 root) に .asadmintruststore ファイルが存在しなくなるため、バグ 6309079、6310428、および 6312869 で説明されているような、アプリケーションのアップグレードに関する問題が発生する可能性があります。

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

説明

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

解決方法

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

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

▼ コマンド行の使用

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

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

```
as-install/bin/asadmin.bat set --port 5858 --user admin --passwordfile \
as-install/bin/password_file raccluster.jms-service.type=REMOTE
```

- 2 **DAS JMS** ホストに対応する **JMS** ホストを作成します。

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

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

```
as-install/bin/asadmin.bat set --port 5858 --user admin --passwordfile \
as-install/bin/password_file raccluster.jms-service.default-jms-host=dashost
```

▼ 管理 GUI の使用

- 1 「設定」->「*cluster-name-config*」->「Java メッセージサービス」->「**JMS** ホスト」の順に移動します。
- 2 「新規」をクリックして新規 **JMS** ホストを作成し、`dashost` という名前を付けます。
- 3 **DAS** の **JMS** サービスに対応する設定を入力します。デフォルト設定は次のとおりです。
 - ホスト名: `localhost`
 - ポート: `7676`
 - 管理者ユーザー: `admin`
 - パスワード: `admin`

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

- 4 「Java メッセージサービス」タブに戻って、JMS サービスタイプを REMOTE に変更します(デフォルトは LOCAL)。
- 5 default-jms-host ドロップダウンリストから dashost を選択します。
- 6 変更を保存してから、ノードエージェントまたはクラスタを起動します。

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

説明

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

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

解決方法

サポートされているブラウザを使用します。Communications Server でサポートされるブラウザのリストについては、「[16 ページの「ブラウザ」](#)」を参照してください。

AIX でカスタムマスターパスワードを使用すると create-domain コマンドが失敗する(6628170)

説明

AIX オペレーティングシステムで、カスタムマスターパスワードを使用してドメインを作成しようとすると、次のエラーが発生して失敗します。

```
keytool error (likely untranslated): java.lang.NullPointerException
Enter keystore password: New keystore password:
```

アプリケーションクライアント

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

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

説明

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

解決方法

現時点ではありません。

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

説明

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

解決方法

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

```
org.omg.CORBA.ORBInitialHost = server-instance-host
org.omg.CORBA.ORBInitialPort = server-instance-port
```

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

説明

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

解決方法

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

付属の Sun JDBC ドライバ

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

PreparedStatement エラーが発生する (6170432)

説明 1

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

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

解決法 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
```

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。

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

説明

ホストシステムまたは Solaris ゾーンのリポート後、または Communications Server 起動後に、付属の Java DB データベースが自動的に再起動しません。これはバグではな

く、付属または他社製のアプリケーションで所定の動作です。問題は、Communications Server インスタンスの前に Java DB を起動する必要があるということです。

解決方法

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

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

配備

クラスタで自動配備に失敗することがある (6610527)

説明

タイミングの問題によって、クラスタをサポートするように設定されているドメインで自動配備に失敗することがあります。この問題は、クラスタをサポートしないドメインでは発生しません。

解決方法

次の解決方法のいずれかを使用します。

- 次のように自動配備を使用します。
 - 個々のアプリケーションを順次自動的に配備します。
 - 個々のアプリケーションの自動配備の間に遅延を設定します。
- 管理コンソール GUI またはコマンド行のいずれかを使用して、アプリケーションを手動で配備します。

アプリケーションに固有のクラスローダーが JSP のコンパイルで使用されない (6693246)

説明

アプリケーションに固有のクラスローダー (applibs または --libraries) が、JSP のコンパイルで使用されません。結果として、これらの JAR を参照する JSP がコンパイルされません。

解決方法

現在のところ解決策はありません。

マニュアル

ここでは、マニュアル上の既知の問題とその解決方法を示します。

Javadoc に矛盾がある。(さまざまな ID)

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

- NumConnAcquired および NumConnReleased 統計情報の取得メソッドが ConnectorConnectionPoolStats および AltJDBCConnectionPoolStats から抜けている。これらの取得メソッドは、将来のリリースで `getNumConnAcquired()` および `getNumConnReleased()` として追加される予定。
- EJBCacheStats 内でメソッド `getPassivationSuccesses()`、`getExpiredSessionsRemoved()`、`getPassivationErrors()`、`getPassi` 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` は、スペルに誤りがあります ("NNN")。これは将来のリリースで訂正される予定。

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

説明

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

解決方法

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

EJB

リソースインジェクションが **HandlerChain** で動作しない (6750245)

説明

EJB の初期化順が原因で、リソースインジェクションが **HandlerChain** で機能しません。

解決方法

現在のところ解決策はありません。

Java EE Tutorial

管理コンソールを使用して何らかのリソースを作成する場合には、「ターゲット」タブを使用してサーバーをターゲットとして指定します。コマンド行または `asant` ターゲットを使用する場合、サーバーがデフォルトのターゲットになるため、特別な処置は必要ありません。

Java Persistence

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

説明

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

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

この問題は、https://glassfish.dev.java.net/issues/show_bug.cgi?id=556 で説明されています。

解決方法

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

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

GenerationType.IDENTITY および **SyBase** とともに使用する **DataDirect** ドライバ (**Issue Tracker 2431**)

説明

DataDirect ドライバを SyBase とともに使用すると、GenerationType.IDENTITY を使用するエンティティの挿入に失敗します。この失敗は、パラメータ化された PreparedStatement ごとに DataDirect ドライバがストアードプロシージャを作成するために発生します。

解決方法

domain.xml ファイルで、対応するデータソースのプロパティ PrepareMethod=direct を設定します。

ライフサイクル管理

ここでは、ライフサイクル管理に関する既知の問題とその解決方法を示します。

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

説明

ejb-timer-service プロパティの minimum-delivery-interval を 9000 に設定したあと、ejb-timer-service プロパティの redelivery-interval-in-millis を 7000 に設定しようとする、set コマンドが次のエラーにより失敗します。

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

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

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

minimum-delivery-interval-in-millis を、ejb-timer-service プロパティの redelivery-interval-in-millis 以上の値に設定する必要があります。redelivery-interval-in-millis の値が minimum-delivery-interval-in-millis の値よりも大きいことを確認する Application Server でのチェックに問題があります。

解決方法

次のように、これらプロパティのデフォルト値を使用します。

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

これらデフォルト以外の値を指定するとエラーが発生します。

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

説明

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

解決方法

これは予想どおりの動作です。Communications Server 2.0 では、default-config は設定情報のテンプレートであるため、JMS 操作 (list や create など) を default-config に対して実行できません。しかし、これらの JMS 操作をクラスタまたはスタンドアロンインスタンスの設定に対して実行することは可能です。

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

説明

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

回避策

この問題には 2 つの回避方法があります。

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

```
<property name="blocking" value="true"/>
```
- Windows Vista または Windows XP を使用します。

ロギング

この節では、ログに関する既知の問題とその解決方法を示します。

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

説明

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

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

解決方法

現時点ではありません。このフラグは設定しないでください。

メッセージキュー

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

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

説明

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

解決方法

これらの問題は、次の方法で回避できます。

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

MQ ブローカが起動していないと、サーバーが起動しない (6740797)

説明

JMS を REMOTE モードとして設定している場合、MQ ブローカが起動していないと Enterprise Server が起動に失敗します。

解決方法

次のように JVM オプションを設定します。com.sun.enterprise.jms.CONNECT_MQ_LAZILY=true。この JVM オプションを設定したあとは、MQ ブローカが起動していない場合でも Communications Server を起動できます。ただし、サーバーを起動する前に、MQ を起動することをお勧めします。

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

説明

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

解決方法

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

アップグレードの前に **imqjmsra.jar** を読み込む際、旧バージョンと新バージョンのクラスが一致しない (6740794)

説明

Application Server の起動中に、サーバーは Message Queue のバージョンをチェックします。Message Queue のバージョンが正しくない場合、サーバーは **imqjmsra.jar** を使用してアップグレードを行います。このアップグレード用の JAR とそのクラスは、次に Application Server が再起動されるまでサーバーから利用できません。この状況は、Message Queue が単独でアップグレードされる場合、または Application Server のみにパッチが適用される場合に発生します。この状況の副次的な影響として、Application Server が起動しない場合があります。

解決方法

Message Queue と Application Server の両方を同じパッチレベルに維持する必要があります。または、Application Server を再起動してください。

監視

ここでは、監視上の既知の問題とその解決方法を示します。

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

説明

HTTP サービスの特定要素の監視統計を表示しているときに、表示される一部の値が現在の値に対応していないか、常に 0 になります。特に、次の HTTP サービスの統計は、Application Server に適用できる情報を示していません。これらの値は無視してください。

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

解決方法

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

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

説明

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

解決方法

現時点ではありません。

セキュリティ

ここでは、Communications 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。

負荷が非常に高い場合に SSL のシナリオで **OutOfMemory** エラーが発生する (JDK 6 Issue 23)

説明

非常に負荷の高い状況で特定の SSL シナリオを実行すると、JDK6 Sun PKCS11 Provider の JDK バグ (https://jdk6.dev.java.net/issues/show_bug.cgi?id=23 を参照) により、OutOfMemory エラーが発生する場合があります。

解決方法

この問題が発生した場合は、JRE インストールの `java.security` ファイルから `sun.security.pkcs11.SunPKCS11` プロバイダを削除してください。

SSL 終了が機能しない (6269102)

説明

SSL 終了が機能しない。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 Communications Server がプロトコルを `https` から `http` に変更する。

解決方法

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

SSL でソケット接続リークが発生する (6492477)

説明

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

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

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

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

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

使用状況が次のように表示されます。

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

この問題は、GlassFish サイトの https://glassfish.dev.java.net/issues/show_bug.cgi?id=849 で説明されています。

解決方法

最新の JDK バージョンにアップグレードします。

Web コンテナ

ここでは、Web コンテナに関する既知の問題とその解決方法を示します。

Windows で、`--precompilejsp=true` を使用してアプリケーションを配備すると、アプリケーションの **JAR** ファイルがロックされ、そのあとの配備取り消しや再配備に失敗する (**5004315**)

説明

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

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

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

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

診断

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

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

解決方法

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

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

空の `<load-on-startup>` 要素を持つ **Servlet 2.4** ベースの `web.xml` を使用して **WAR** を配備できない (**6172006**)

説明

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

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

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

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

解決方法

Servlet 2.4 ベースの web.xml を使用する場合

は、`<load-on-startup>0</load-on-startup>` を指定して、サーブレットの読み込み順序が問題にならないことを示します。

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

説明

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

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

解決方法

JSP のコンパイルスイッチを「fork」から「false」に設定します。

これは、次のいずれかの方法で行えます。

- グローバルに行うには、次のように、`domain-dir/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。

Web サービス

ここでは、Web コンテナに関する既知の問題とその解決方法を示します。

JDK6 u4 b3 で `wscompile` は「**package javax.xml.rpc does not exist**」エラーが発生する (**6638567**)

説明

`wscompile` Ant タスクは、JDK 6 Update 4 では失敗します。JAX-RPC API クラスごとに、次のエラーメッセージが表示されます。

```
package package-name does not exist
```

解決方法

`wscompile` Ant タスクを実行する前に `j2ee.jar` ではなく `javaee.jar` がクラスパスに指定されていることを確認します。