



# Sun GlassFish Communications Server 2.0 版本說明



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

文件號碼：821-1053-10  
2009 年 10 月

Sun Microsystems, Inc. 對本文件所述產品所採用的技術擁有相關智慧財產權。特別是 (但不僅限於)，這些智慧財產權可能包含一項或多項美國專利，或美國及其他國家/地區的申請中專利。

美國政府權利 – 商業軟體。政府使用者均應遵守 Sun Microsystems, Inc. 的標準授權合約和 FAR 及其增補文件中的適用條款。

本發行物可能包含由協力廠商開發的材料。

本產品中的某些部分可能源自加州大學授權的 Berkeley BSD 系統的開發成果。UNIX 是在美國及其他國家/地區的註冊商標，已獲得 X/Open Company, Ltd. 專屬授權。

Sun、Sun Microsystems、Sun 標誌、Solaris 標誌、Java 咖啡杯標誌、docs.sun.com、Java Java 與 Solaris 是 Sun Microsystems, Inc. 或其子公司在美國及其他國家/地區的商標或註冊商標。所有 SPARC 商標的使用均已獲得許可，它們是 SPARC International, Inc. 在美國和其他國家/地區的商標或註冊商標。凡具有 SPARC 商標的產品都是採用 Sun Microsystems, Inc. 所開發的架構。

OPEN LOOK 與 Sun™ Graphical User Interface (Sun 圖形化使用者介面) 都是由 Sun Microsystems, Inc. 為其使用者與授權者所開發的技術。Sun 感謝 Xerox 公司在研究和開發視覺化或圖形化使用者介面之概念上，為電腦工業所做的開拓性貢獻。Sun 已向 Xerox 公司取得 Xerox 圖形化使用者介面之非獨占性授權，該授權亦適用於使用 OPEN LOOK GUI 並遵守 Sun 書面授權合約的 Sun 公司授權者。

本出版物所涵蓋的產品和包含的資訊受到美國出口控制法規的控制，並可能受到其他國家/地區進出口法規的管轄。嚴禁核武、導彈、生化武器或海上核武等最終用途或一般使用者直接或間接使用本產品。嚴禁向被美國禁運的國家/地區或美國出口除外清單 (包括但不僅限於被拒人清單和特別指定的國家/地區清單) 上標識的實體出口或再出口本產品。

文件以其「原狀」提供，對任何明示或暗示的條件、陳述或擔保，包括對適銷性、特殊用途的適用性或非侵權性的暗示保證，均不承擔任何責任，除非此免責聲明的適用範圍在法律上無效。

# 簡介

---

Sun GlassFish™ Communications Server 在 Sun GlassFish 企業伺服器中新增了通訊啟動協定 (SIP) Servlet 技術功能。

Sun GlassFish Communications Server 可免費用於開發、部署和重新發行。如果您是有意重新發行的客戶，請與 [Sun OEM](#) 銷售人員連絡，以取得重新發行授權。

Sun GlassFish Communications Server 是一種簡易、快捷、領先業界的應用程式伺服器，以 Java™ Platform Enterprise Edition (Java EE) 與 SIP 技術為基礎，適用於開發及提供 Web 應用程式與 Web 服務。Sun GlassFish Communications Server 能讓您建立與 JSR 289 相容的 SIP 應用程式和整合應用程式。這套產品讓可延伸的電信業者等級服務提供優異效能、叢集和高可用性功能，即使出現軟硬體故障仍可繼續作業。

- [第 3 頁的「關於此版本說明」](#)
- [第 4 頁的「為殘障人士提供的無障礙功能」](#)
- [第 4 頁的「相關文件」](#)
- [第 5 頁的「如何報告問題與提供建議」](#)
- [第 6 頁的「Sun 歡迎您提出寶貴意見」](#)
- [第 6 頁的「其他 Sun 資源」](#)

## 關於此版本說明

這些版本說明包含 Sun GlassFish Communications Server 2.0 發行當時提供的重要資訊。此處還提供增強功能、已知問題及其他最新問題。請在開始使用 Communications Server 2.0 之前閱讀此文件。

這些版本說明的最新版本位於 Communications Server [文件網站 \(http://docs.sun.com/coll/1343.10\)](http://docs.sun.com/coll/1343.10)。請在安裝和設定您的軟體前查閱此網站，並在安裝後定期查閱以檢視最新的版本說明和產品文件。

本文件中提供了協力廠商 URL 以供參考，另亦提供其他相關的資訊。

**備註** – Sun 對本文件中提到的協力廠商網站的可用性不承擔任何責任。對於此類網站或資源中的 (或透過它們所取得的) 任何內容、廣告、產品或其他材料，Sun 並不表示認可，也不承擔任何責任。Sun 對於因使用或依靠此類網站或資源中的 (或透過它們所取得的) 任何內容、產品或服務而造成的或連帶產生的任何實際或名義上之損壞或損失，Sun 概不負責，也不承擔任何責任。

## 版本說明修訂歷程記錄

本節列出在 Sun GlassFish Communications Server 產品的初期測試版之後，已在此版本說明中所做的變更。

表 1-1 版本說明修訂歷程記錄

修訂日期	說明
2009 年 10 月	Communications Server 2.0 產品的 FCS 發行版本。

## 為殘障人士提供的無障礙功能

欲獲得此媒體發佈以來已發行的無障礙功能，請向 Sun 索取依據美國「Section 508」法規進行產品評估所得之結果文件，以便決定最適合佈署無障礙功能解決方案的版本。您可以在下列網站找到應用程式的更新版本：<http://sun.com/software/javaenterprisesystem/get.html>

如需有關 Sun 在無障礙功能方面之成果的資訊，請至 <http://sun.com/access>。

## 相關文件

表 1-2 Communications Server 文件集中的書籍

書名	說明
文件中心	依作業及主旨編排的 Communications Server 文件主題。
版本說明	軟體與文件的最新資訊。包含支援硬體、作業系統、Java 開發工具組 (JDK™) 與資料庫驅動程式的完整表格式摘要。
快速入門指南	如何開始使用 Communications Server 產品。
安裝指南	安裝軟體及其元件。
應用程式部署指南	將應用程式及應用程式元件部署到 Communications Server。包含部署描述元的相關資訊。

表 1-2 Communications Server 文件集中的書籍 (續)

書名	說明
開發人員指南	建立與實作要在 Communications Server (遵循適用於 Java EE 元件及 API 的開放式 Java 標準模型) 上執行的 Java Platform, Enterprise Edition (Java EE 平台) 應用程式。其中包括有關開發人員工具、安全性、除錯和建立生命週期模組的資訊。
Java EE 5 指導	使用 Java EE 5 平台技術及 API 來開發 Java EE 應用程式。
Java WSIT 教學課程	使用 Web 服務互通技術 (WSIT) 開發 Web 應用程式。說明如何、何時及為何要使用 WSIT 技術，以及每一項技術所支援的功能和選項。
管理指南	Communications Server 的系統管理，包括配置、監視、安全性、資源管理以及 Web 服務管理。
高可用性管理指南	設定叢集、使用節點代理程式及使用負載平衡器。
管理參照	編輯 Communications Server 配置檔案 domain.xml。
效能調校指南	調校 Communications Server 以提昇效能。
參考手冊	與 Communications Server 一起提供的公用程式指令；以線上手冊樣式編寫。其中包含 asadmin 指令行介面。

## 如何報告問題與提供建議

如果您有 Sun GlassFish Communications Server 方面的問題，請使用以下機制與 Sun 連絡：

- [SailFin 郵件收信人清單 \(https://sailfin.dev.java.net/servlets/ProjectMailingListList\)](https://sailfin.dev.java.net/servlets/ProjectMailingListList)：針對不同主題與意見設立的各種 SailFin 社群郵件收信人清單
  - [Java Developer Connection 上的錯誤資料庫 \(http://developer.java.sun.com/servlet/SessionServlet?url=/developer/bugParade/index.jshtml\)](http://developer.java.sun.com/servlet/SessionServlet?url=/developer/bugParade/index.jshtml)：若要檢視錯誤或提交錯誤，請使用 Java Developer Connection Bug Parade
  - [SailFin 問題追蹤系統 \(https://sailfin.dev.java.net/servlets/ProjectIssues\)](https://sailfin.dev.java.net/servlets/ProjectIssues)
  - [Java 技術論壇 \(http://forum.java.sun.com/\)](http://forum.java.sun.com/)：一個互動留言板，可讓您分享有關 Java 技術及程式設計技巧的知識與問題。
  - [Sun 軟體支援服務 \(http://www.sun.com/service/sunone/software\)](http://www.sun.com/service/sunone/software)：包含知識庫、線上支援中心和 Product Tracker 的連結，以及維護方案和支援服務連絡電話的連結
  - 與維護合約相關的電話派送號碼
- 如此我們將可為您提供最完善的解決問題方法，請在與支援連絡時提供以下資訊：
- 對問題進行說明，包括問題發生的情形以及對作業的影響
  - 機器類型、作業系統版本以及產品版本，包括任何可能對問題造成影響的修補程式及其他軟體
  - 您的詳細操作步驟以重現問題

- 所有的錯誤記錄或核心傾印

## Sun 歡迎您提出寶貴意見

Sun 致力於提高文件品質，因此誠心歡迎您提出意見與建議。

若要分享您的意見，請前往 <http://docs.sun.com>，並按一下 [回]。在線上表單中，請提供文件標題和文件號碼。文件號碼是一個七位或九位的數字，可以在書的標題頁面或文件的頂部找到。例如，本書標題為「Sun GlassFish Communications Server 2.0 版本說明」，文件號碼為 821-1053-10。

## 其他 Sun 資源

可在下列位置找到有用的資訊：

- Communications Server 產品資訊 ([http://www.sun.com/software/products/appsrvr/home\\_appsrvr.html](http://www.sun.com/software/products/appsrvr/home_appsrvr.html))
- 專業服務 (<http://www.sun.com/service/sunps/sunone>)
- 軟體產品與服務 (<http://www.sun.com/software>)
- 軟體支援服務 (<http://www.sun.com/service/sunone/software>)
- 支援與知識庫 (<http://www.sun.com/service/support/software>)
- Sun 支援與教育訓練服務 (<http://training.sun.com>)
- 顧問與專業服務 (<http://www.sun.com/service/sunps/sunone>)
- 開發人員資訊 (<http://developers.sun.com>)
- Sun 開發人員支援服務 (<http://www.sun.com/developers/support>)
- 軟體教育訓練 (<http://www.sun.com/software/training>)
- Sun 軟體資料表 (<http://www.sun.com/software>)
- Sun Microsystems 產品文件 (<http://docs.sun.com/>)

## 關於 Sun GlassFish Communications Server 2.0

---

Communications Server 是與 Java EE 5 平台相容的伺服器，用於在大型生產環境中開發和部署 Web、SIP 和整合應用程式，及以 Java 技術為基礎的 Web 服務。

本章包含以下內容：

- 第 7 頁的「Sun GlassFish Communications Server 功能」
- 第 10 頁的「硬體和軟體需求」
- 第 15 頁的「使用更新中心」

### Sun GlassFish Communications Server 功能

除支援 Sun GlassFish 企業伺服器中的所有可用功能之外，Communications Server 還具有下列新功能。如需在 Sun GlassFish 企業伺服器中可用的功能清單，請參閱「Sun GlassFish 企業伺服器版本說明」。

- **與 JSR 289 相容** — Communications Server 含有與 RFC 3261、3262、3265、3311、3515、3903 等相容的 SIP 堆疊，它也會顯示由 JSR 289 所定義的 SIP Servlet Java API。
- **升級** — Communications Server 支援從 Communications Server 1.5 升級為 Communications Server 2.0。
- **整合的 SIP 與 HTTP 階段作業** — Communications Server 支援依照 JSR 289 定義的整合 HTTP 和 SIP 階段作業功能。
- **DCR 外掛程式支援** — Communications Server 支援使用 Java 類別形式的外掛程式進行資料導向規則的配置。
- **超載保護** — Communications Server 支援以 CPU 為基礎以及以記憶體為基礎的超載保護。
- **SIP 階段作業複製** — Communications Server 提供可靠的 SIP 階段作業容錯移轉。
- **輪流升級** — Communications Server 支援透過對服務和階段作業所造成損失最小的方式升級應用程式或伺服器。

- **多址支援** — Communications Server 支援跨兩個或兩個以上不同子網路進行 SIP 通訊，並提供相同子網路中的備援網路連線。Communications Server 可用於具有多個 NIC 的環境。
- **整合的 SIP Servlet 容器** — SIP Servlet 容器提供下列功能：
  - 提供偵聽 SIP 請求的網路端點。
  - 提供主控與管理 SIP Servlet 生命週期的環境。

SIP Servlet 容器可與 Web 容器共同運作，依照 SIP Servlet API 規格 (SSA) 的定義為整合應用程式提供支援。

- **管理簡單** — Communications Server 能讓您使用 CLI 和 GUI 輕易地管理 SIP 容器以及部署的 SIP 應用程式。
- **安全性** — 支援 HTTP 和 SIP 協定的摘要認證。支援 SIP 協定的 P-Asserted-Identity。SIP 容器支援 SIPS over TLS。
- **整合負載平衡器** — 整合負載平衡器 (CLB) 是軟體負載平衡元件，能在伺服器實例叢集中分配應用程式請求，賦予整合應用程式高可用性。CLB 提供下列功能：
  - 將 SIP 和 HTTP 請求轉送至伺服器實例叢集。
  - 使用一致的雜湊/循環策略讓請求負載平衡。
  - 在叢集中啟用與停用伺服器實例。
  - 監視伺服器實例的運作狀態。
- **支援部署 SIP 和整合應用程式** — Communications Server 支援使用 GlassFish 中的延伸模組機制，進行純 SIP 應用程式和整合應用程式的部署。它也支援使用 sun-sip.xml 做為 SIP 應用程式的 Sun 專用部署描述元。
- **NetBeans 工具製作** — Communications Server 在 NetBeans 6.0 上提供以 JSR 289 為基礎的 SIP Servlet 應用程式支援。您可以使用測試代理程式建立簡單的 SIP 用戶端，以測試 SIP Servlet。此外掛程式可以在 as-install-dir/lib/tools 下取得。
- **應用路由器部署** — Communications Server 支援使用延伸模組機制，線上部署應用路由器歸檔 (JAR 檔案)。預設會提供遵循簡易字母順序規則路由請求的應用路由器。
- **封裝與檔案配置** — Communications Server 是透過以檔案為基礎的可安裝程式型態提供。
- **在 Sun Connection 上註冊**：您可以使用安裝程式、管理主控台 GUI 或更新中心，在 [Sun Connection](http://www.sun.com/service/sunconnection/index.jsp) (<http://www.sun.com/service/sunconnection/index.jsp>) 上註冊產品。在 Sun Connection 上註冊 Communications Server 之後，您可以獲得以下好處：
  - 修補程式資訊與錯誤更新
  - 螢幕展示與教學課程
  - 新聞與活動
  - 支援與教育訓練服務
- **SuSe Linux 64 位元支援**
- **Java EE 5 平台支援** — Sun GlassFish Communications Server 實作 Java EE 5 規格，以便為下一代企業應用程式和 Web 服務提供最佳的應用程式執行階段。Communications Server 實作以下 Java EE 標準：



- Enterprise Java Beans 3.0
- JAXB 2.0
- Java 持續性
- 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

此版本說明稍後會提供完整的 Java EE 5 平台技術清單。

- **Web 服務互通功能技術 (WSIT) 支援：**Sun 與 Microsoft 密切合作，以確保訊息最佳化、可靠訊息傳送和安全性等 Web 服務企業技術具有互通性。WSIT 的初期測試版便是此共同努力的成果。WSIT 實作多種開放式 Web 服務規格，用於支援企業功能。除了訊息最佳化、可靠訊息傳送和安全性之外，WSIT 還包含啟動和配置技術。從 Java 平台目前內建的核心 XML 支援功能開始，WSIT 使用或延伸了現有功能，同時新增了可互通 Web 服務的支援功能，其中包括：
  - 啟動和配置
  - 訊息最佳化技術
  - 可靠訊息傳送技術
  - 安全性技術
- **JBIs 支援：**JBIs 以商業整合服務提供者介面 (SPI) 延伸了 Java EE。這些 SPI 可讓開發人員針對 WSCI、BPEL4WS 和 W3C 流程編排工作群組等規格建立或實作 Java 商務整合環境。JBIs 實作直接由 Communications Server 安裝程式進行安裝，通常位於 *as-install/jbi* 目錄下。此目錄包含 JBIs 的所有共用 JAR 檔案和系統元件，其中包括在 Communications Server JVM 中啟動 JBIs 架構的 lifecycle 模組。
- **用法設定檔：**每個管理網域都與可指定此網域功能的用法設定檔相關聯。Communications Server 提供以下設定檔：
  - **開發人員設定檔：**在以下情況使用此設定檔：網域在開發環境中執行，並且您的應用程式不需要叢集功能 (例如負載平衡、高可用性，以及階段作業複製)。請注意，此設定檔的實際名稱爲「developer」(區分字母大小寫)。
  - **叢集設定檔：**如果您要建立應用程式伺服器實例的叢集，讓已部署的 Java EE 應用程式具有延展性和高可用性，請使用此設定檔。應用程式的狀態持續位於**記憶體中**。請注意，此設定檔的實際名稱爲「cluster」(區分大小寫)。
- **開放原始碼和 SailFin 社群 — SailFin (<http://sailfin.dev.java.net>)** 社群旨在開發實作 Java EE 5 平台之最新功能與相關企業技術的免費開放原始碼商業等級通訊伺服器。
- **更新中心支援：**Communications Server 更新中心提供自動執行的 Communications Server 更新，同時可讓您輕鬆存取其他元件。

# 硬體和軟體需求

本節列出安裝 Sun GlassFish Communications Server 之前必須滿足的需求。

- 第 10 頁的「支援的平台」
- 第 11 頁的「重要修補程式資訊」
- 第 11 頁的「JDK 版本」

## 支援的平台

下表列出與 Sun GlassFish Communications Server 相容的作業系統。

表 2-1 支援的作業系統

作業系統	最小記憶體	建議的記憶體	最小磁碟空間	建議的磁碟空間	JVM	部署類型
Sun Solaris 10 (x86) (32 位元和 64 位元)	512 MB	512 MB	250 MB 可用空間	500 MB 可用空間	J2SE 5.0 - 32 位元  Java SE 6 - 32 位元和 64 位元	生產
Sun Solaris 10 (SPARC)(32 位元和 64 位元)	512 MB	512 MB	250 MB 可用空間	500 MB 可用空間	J2SE 5.0 - 32 位元  Java SE 6 - 32 位元和 64 位元	生產
Red Hat Enterprise Linux 3.0 Update 1、4.0 和 5.x (32 位元和 64 位元)	512 MB	1 GB	250 MB 可用空間	500 MB 可用空間	J2SE 5.0 - 32 位元  Java SE 6 - 32 位元和 64 位元	生產
SUSE Linux Enterprise Server 10 (32 位元和 64 位元)  (也支援 SP1 和 SP2)	512 MB	1 GB	250 MB 可用空間	500 MB 可用空間	J2SE 5.0 - 32 位元  Java SE 6 - 32 位元和 64 位元	生產

表 2-1 支援的作業系統 (續)

作業系統	最小記憶體	建議的記憶體	最小磁碟空間	建議的磁碟空間	JVM	部署類型
Windows XP Windows Vista Windows 2008	1 GB	2 GB	500 MB 可用空間	1 GB 可用空間	J2SE 5.0 - 32 位元  Java SE 6 - 32 位元和 64 位元	開發
Windows 7	1 GB	2 GB	500 MB 可用空間	1 GB 可用空間	J2SE 5.0 - 32 位元  Java SE 6 - 32 位元和 64 位元	開發
Macintosh OS 10.4、10.5	512 MB	1 GB	250 MB 可用空間	500 MB 可用空間	J2SE 5.0 - 32 位元  Java SE 6 - 32 位元	開發
OpenSolaris 2009.06	512 MB	1 GB	250 MB 可用空間	500 MB 可用空間	J2SE 5.0 - 32 位元  Java SE 6 - 32 位元和 64 位元	開發

在 UNIX™ 上，您可以使用 `uname` 指令檢查作業系統版本。可以使用 `df` 指令檢查磁碟空間。

備註 – 建議您在任何 Windows 平台上執行 Communications Server 時，採用 NTFS 檔案系統，而非 FAT 或 FAT32。

## 重要修補程式資訊

### Solaris 修補程式需求

建議 Solaris 10 (x86、SPARC) 使用者安裝「Sun 建議的修補程式叢集」。您可以在 [SunSolve \(http://sunsolve.sun.com/pub-cgi/show.pl?target=patchpage\)](http://sunsolve.sun.com/pub-cgi/show.pl?target=patchpage) 網站上的「建議與安全修補程式」下找到此修補程式叢集。

### JDK 版本

Communications Server 所需的 JDK 最舊 (且經過認證) 版本為 **1.6.0\_7**。

▼ 切換至支援的 Java SE 版本

您可以根據此處所述內容編輯 `asenv` 檔案，來切換至支援的 Java SE 版本。

- 1 如果尚未執行此操作，請在系統上安裝新的 Java SE 版本。  
您可以從 <http://java.sun.com/javase> 下載 Java SE SDK
- 2 停止 **Communications Server** 網域。
- 3 編輯 `as-install/config/asenv.conf` 檔案(在 Windows 中為 `asenv.bat`)，將 `AS_JAVA` 的值變更為指向新的 Java 主目錄：
- 4 編輯 `as-install/samples/common.properties` 檔案，變更以 `com.sun.aas.javaRoot...` 開頭的一行，來參照新的 Java 主目錄。
- 5 重新啟動 **Communications Server**。

Ant 版本

SailFin JAR 安裝隨附的 Ant 版本是 **1.6.5**。若要建立自訂的應用路由器應用程式，您需要使用必須另外下載的 Ant **1.7**。

JDBC 驅動程式和資料庫

表 2-2 列出符合 Java EE 相容性需求的資料庫和驅動程式。所有支援的 Sun GlassFish Communications Server 配置都必須包含此表格中的至少一個資料庫/驅動程式組合，例如隨附的 Java DB 資料庫和驅動程式。此外，Communications Server 旨在支援與其他任何具有相應 JDBC 驅動程式的 DBMS 之 JDBC 連結。

表 2-2 與 Java EE 相容的 JDBC 驅動程式

JDBC 驅動程式供應商	JDBC 驅動程式類型	支援的資料庫伺服器
Derby Network Client	類型 4	Derby 10.2
DataDirect 3.6.x、3.7.x	類型 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	類型 4	MySQL 5.0

表 2-2 與 Java EE 相容的 JDBC 驅動程式 (續)

JDBC 驅動程式供應商	JDBC 驅動程式類型	支援的資料庫伺服器
Oracle 10g、11g	類型 4	Oracle 10g、11g -RAC
PostGres	類型 4	8.2.x

一般來說，Communications Server 支援所有符合 Java EE 規格的 JDBC 驅動程式。

### Message Queue 版本

如果是從檔案安裝，Message Queue 4.4 會與 Communications Server 封裝在一起。

Communications Server 隨附的內嵌 Sun GlassFish Message Queue 程式碼，一般而言僅會針對等效版本的 Message Queue 代理程式測試及認證。這表示目前不支援使用隨附的內嵌 Message Queue 程式碼，連線至執行不同版本 Message Queue 程式碼的遠端 (未由 Communications Server 管理) Message Queue 代理程式。

### 瀏覽器

本節列出 Sun GlassFish Communications Server 2.0 支援的瀏覽器。

表 2-3 支援的 Web 瀏覽器

瀏覽器	版本
Mozilla	1.7.12
Internet Explorer	6.0 Service Pack 2、7.0
Firefox	2.x、3.x
Safari	3.x、4.x
Netscape	8.0.4、8.1、9.0、9.0.x

### 其他需求

在安裝 Sun GlassFish Communications Server 軟體之前，必須滿足以下其他需求。

- **可用空間：**您的暫存目錄必須有最少 35MB 的可用空間供 Sun GlassFish Communications Server 安裝使用。
- **使用解除安裝程式：**如果您需要從系統中移除 Communications Server，則務必使用此軟體包含的解除安裝程式。如果您嘗試使用其他方法，則在嘗試重新安裝相同的版本或安裝新的版本時將會出現問題。
- **可用連接埠：**您必須擁有七個未使用的連接埠可用。安裝程式會自動偵測使用中的連接埠，並為預設網域建議目前未使用的連接埠。依預設，初始預設連接埠為：
  - 8080 (用於 HTTP)、8181 (用於 HTTPS)

- 3700 (用於 IIOP)、3820 (用於 IIOP/SSL) 以及 3920 (用於具有相互認證的 IIOP/SSL)
- 4848 (HTTP)，用於管理伺服器
- 8686 (RMI) 連接埠，用於純 JMX 用戶端
- 關閉防火牆 (Microsoft Windows)：您必須在安裝 Sun GlassFish Communications Server 軟體之前停止所有防火牆，因為依預設，某些此類軟體會停用所有連接埠。安裝程式必須可以確切地決定哪些連接埠是可用的。

## Java EE 5 平台 API

Sun GlassFish Communications Server 2.0 支援 Java EE 5 平台。下表列出 Java EE 5 平台提供的增強型 API。

表 2-4 Java EE 5 平台上的主要 API 變更

API	JSR
<b>Java EE 5</b>	
Java Platform, Enterprise Edition 5 ( <a href="http://java.sun.com/javaee/5/docs/api/">http://java.sun.com/javaee/5/docs/api/</a> )	JSR 244 ( <a href="http://jcp.org/aboutJava/communityprocess/pr/jsr244/">http://jcp.org/aboutJava/communityprocess/pr/jsr244/</a> )
<b>Web 服務技術</b>	
實作企業 Web 服務	JSR 109 ( <a href="http://jcp.org/en/jsr/detail?id=109">http://jcp.org/en/jsr/detail?id=109</a> )
Java API for XML-Based Web Services (JAX-WS) 2.0 ( <a href="https://jax-ws.dev.java.net/">https://jax-ws.dev.java.net/</a> )	JSR 224 ( <a href="http://jcp.org/en/jsr/detail?id=224">http://jcp.org/en/jsr/detail?id=224</a> )
Java API for XML-Based RPC (JAX-RPC) 1.1 ( <a href="https://jax-rpc.dev.java.net/">https://jax-rpc.dev.java.net/</a> )	JSR 101 ( <a href="http://jcp.org/en/jsr/detail?id=101">http://jcp.org/en/jsr/detail?id=101</a> )
Java Architecture for XML Binding (JAXB) 2.0 ( <a href="https://jaxb.dev.java.net/">https://jaxb.dev.java.net/</a> )	JSR 222 ( <a href="http://jcp.org/en/jsr/detail?id=222">http://jcp.org/en/jsr/detail?id=222</a> )
SOAP with Attachments API for Java (SAAJ) ( <a href="https://saaj.dev.java.net/">https://saaj.dev.java.net/</a> )	JSR 67 ( <a href="http://jcp.org/en/jsr/detail?id=67">http://jcp.org/en/jsr/detail?id=67</a> )
Streaming API for XML ( <a href="http://java.sun.com/webservices/docs/1.6/tutorial/doc/SJSXP.html">http://java.sun.com/webservices/docs/1.6/tutorial/doc/SJSXP.html</a> )	JSR 173 ( <a href="http://jcp.org/en/jsr/detail?id=173">http://jcp.org/en/jsr/detail?id=173</a> )
用於 Java 平台的 Web 服務中介資料	JSR 181 ( <a href="http://jcp.org/en/jsr/detail?id=181">http://jcp.org/en/jsr/detail?id=181</a> )
<b>元件模型技術</b>	
Enterprise JavaBeans 3.0 ( <a href="http://java.sun.com/products/ejb/">http://java.sun.com/products/ejb/</a> )	JSR 220 ( <a href="http://jcp.org/en/jsr/detail?id=220">http://jcp.org/en/jsr/detail?id=220</a> )
J2EE Connector Architecture 1.5 ( <a href="http://java.sun.com/j2ee/connector/">http://java.sun.com/j2ee/connector/</a> )	JSR 112 ( <a href="http://jcp.org/en/jsr/detail?id=112">http://jcp.org/en/jsr/detail?id=112</a> )

表 2-4 Java EE 5 平台上的主要 API 變更 (續)

API	JSR
Java Servlet 2.5 ( <a href="http://java.sun.com/products/servlet/">http://java.sun.com/products/servlet/</a> )	JSR 154 ( <a href="http://jcp.org/en/jsr/detail?id=154">http://jcp.org/en/jsr/detail?id=154</a> )
JavaServer Faces 1.2 ( <a href="http://java.sun.com/j2ee/javaserverfaces/">http://java.sun.com/j2ee/javaserverfaces/</a> )	JSR 252 ( <a href="http://jcp.org/en/jsr/detail?id=252">http://jcp.org/en/jsr/detail?id=252</a> )
JavaServer Pages 2.1 ( <a href="http://java.sun.com/products/jsp/">http://java.sun.com/products/jsp/</a> )	JSR 245 ( <a href="http://jcp.org/en/jsr/detail?id=245">http://jcp.org/en/jsr/detail?id=245</a> )
JavaServer Pages Standard Tag Library 1.2 ( <a href="http://java.sun.com/products/jsp/jstl/">http://java.sun.com/products/jsp/jstl/</a> )	JSR 52 ( <a href="http://jcp.org/en/jsr/detail?id=52">http://jcp.org/en/jsr/detail?id=52</a> )
<b>管理技術</b>	
J2EE Management ( <a href="http://java.sun.com/j2ee/tools/management/">http://java.sun.com/j2ee/tools/management/</a> )	JSR 77 ( <a href="http://jcp.org/en/jsr/detail?id=77">http://jcp.org/en/jsr/detail?id=77</a> )
J2EE Application Deployment ( <a href="http://java.sun.com/j2ee/tools/deployment/">http://java.sun.com/j2ee/tools/deployment/</a> )	JSR 88 ( <a href="http://jcp.org/en/jsr/detail?id=88">http://jcp.org/en/jsr/detail?id=88</a> )
Java Authorization Contract for Containers ( <a href="http://java.sun.com/j2ee/javaacc/">http://java.sun.com/j2ee/javaacc/</a> )	JSR 115 ( <a href="http://jcp.org/en/jsr/detail?id=115">http://jcp.org/en/jsr/detail?id=115</a> )
<b>其他 Java EE 技術</b>	
Java 平台的共用附註	JSR 250 ( <a href="http://jcp.org/en/jsr/detail?id=250">http://jcp.org/en/jsr/detail?id=250</a> )
Java Transaction API (JTA) ( <a href="http://java.sun.com/products/jta/">http://java.sun.com/products/jta/</a> )	JSR 907 ( <a href="http://jcp.org/en/jsr/detail?id=907">http://jcp.org/en/jsr/detail?id=907</a> )
JavaBeans Activation Framework (JAF) 1.1 ( <a href="http://java.sun.com/products/javabeans/glasgow/jaf.html">http://java.sun.com/products/javabeans/glasgow/jaf.html</a> )	JSR 925 ( <a href="http://jcp.org/en/jsr/detail?id=925">http://jcp.org/en/jsr/detail?id=925</a> )
JavaMail ( <a href="http://java.sun.com/products/javamail/">http://java.sun.com/products/javamail/</a> )	JSR 919 ( <a href="http://jcp.org/en/jsr/detail?id=919">http://jcp.org/en/jsr/detail?id=919</a> )
Java Message Service API ( <a href="http://java.sun.com/products/jms/">http://java.sun.com/products/jms/</a> )	JSR 914 ( <a href="http://www.jcp.org/en/jsr/detail?id=914">http://www.jcp.org/en/jsr/detail?id=914</a> )
Java Persistence API ( <a href="http://java.sun.com/j2ee/persistence/faq.html">http://java.sun.com/j2ee/persistence/faq.html</a> )	JSR 220 ( <a href="http://www.jcp.org/en/jsr/detail?id=220">http://www.jcp.org/en/jsr/detail?id=220</a> )

# 使用更新中心

更新中心提供自動執行的 Communications Server 更新，同時可讓您輕鬆存取其他元件。

啓用更新中心之後，便會自動執行軟體更新。在此自動更新過程中，更新中心會收集以下資料並將其傳輸至 Sun Microsystems (或其服務提供者)：

- 唯一的安裝 ID (GUID)
- IP 位址
- 作業系統資訊 (名稱、版本、架構、語言環境)
- JDK 版本
- 模組下載資訊 (模組名稱、日期、時間、狀態、下載時間、已下載的位元組數)

我們不會追蹤個人識別資訊，也不會將個人識別資訊與其他任何資料進行關聯，或者用於生成報告。

## ▼ 使用更新中心

為確保明確同意使用自動更新，更新中心預設處於停用狀態。若要啓用更新中心以執行定期檢查和自動更新，請執行以下步驟：

- 1 啓動更新中心。
  - 在 Unix 上：`<installdir>/updatecenter/bin/updatetool`
  - 在 Windows 上：`<installdir>\updatecenter\bin\updatetool.bat`
- 2 選取[喜好設定]標籤。
- 3 在[更新排程]視窗中，將[檢查更新]下拉式方塊的值從[永遠不(手動)]變更為所需的值。例如，每天或每週。
- 4 指定在每週的哪一天和每天的哪一時間執行更新。
- 5 選取[儲存]按鈕以儲存您的變更。

從今以後更新中心將會自動依照指定的排程，檢查 SailFin 或 Communications Server 元件更新。當具有可用更新時，更新中心便會啓動並通知您可以更新元件。

## 有關 WSIT 整合的更多資訊

如需有關 WSIT 狀態的詳細資訊，請參閱「[WSIT 狀態說明 \(https://wsit.dev.java.net/source/browse/\\*checkout\\*/wsit/wsit/status-notes/status-notes-1-0-1-FCS.html\)](https://wsit.dev.java.net/source/browse/*checkout*/wsit/wsit/status-notes/status-notes-1-0-1-FCS.html)」頁。



## Sun GlassFish Communications Server 的已知問題與限制

---

本章說明 Communications Server 軟體的已知問題以及相關的解決方法。如果摘要敘述未指明特定的平台，則所有平台都可能出現此問題。這些資訊按以下章節進行分類：

- 第 17 頁的「Communications Server 管理」
- 第 19 頁的「整合負載平衡器」
- 第 20 頁的「安裝」
- 第 21 頁的「安全性」
- 第 21 頁的「SIP 容器」
- 第 24 頁的「SIP 階段作業複製」

### Communications Server 管理

#### Communications Server 不會偵測與叢集活動訊號連接埠的衝突 (問題編號 1967)

##### 說明

叢集建立時，Communications Server 會為其隨機指定 1026 到 45556 之間的活動訊號連接埠。對於預設叢集 (亦即 Communications Server 安裝所建立的預設叢集)，系統會隨機選取介於 0 至 45556 之間的數字。叢集建立程序不會精確偵測是否已有其他服務使用該活動訊號連接埠。

##### 解決方案

如果自動叢集建立配置選取了與另一服務 (已使用該連接埠) 衝突的活動訊號連接埠，請將叢集活動訊號連接埠更新為系統未使用的連接埠。

要變更叢集的活動訊號連接埠，請使用下列 `asadmin` 指令：

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

## 執行 64 位元 Linux 的 NFS 伺服器停止建立網域 (問題編號 1961)

### 說明

在執行 64 位元 Linux 的 NFS 伺服器上，`asadmin create-domain` 指令在嘗試於掛載 Network File System (NFS) 的檔案系統上建立網域時可能會失敗。

### 解決方案

無已知解決方案。

## 流量極小或沒有流量時，CPU 使用率居高不下 (問題編號 1966)

### 說明

啓用了 CPU 超載保護時，即使在流量極小或沒有流量時，Communications Server 實例有時也會顯示很高的 CPU 使用量。這個問題是由於 JDK 錯誤所造成的：[6693490](http://bugs.sun.com/bugdatabase/view_bug.do?bug_id=6693490) ([http://bugs.sun.com/bugdatabase/view\\_bug.do?bug\\_id=6693490](http://bugs.sun.com/bugdatabase/view_bug.do?bug_id=6693490))。這個錯誤已經在 JDK 6 Update 18 中解決。

### 解決方案

請將 JDK 6 Update 18 配合 Communications Server 使用。

## 在未連結 SIP/SIPS 連接埠的情況下，Communications Server 實例仍會啓動 (問題編號 998)

### 說明

在無法連結到 SIP 或 SIPS 連接埠的情況下，Communications Server 實例仍會啓動。

### 解決方案

請在啓動伺服器實例之前，確認連接埠可以使用。請檢查記錄檔 (`server.log`) 以確認啓動期間沒有發生過任何 SIP 容器錯誤或異常。

## Communications Server 不會使用 —javahome 選項所指定的 JDK (問題編號 789)

### 說明

您可以使用 —javahome 選項，以預先安裝的 JDK 替代安裝的預設版本。依照預設，Communications Server 會使用來自 `as-install/jdk` 的 JDK 版本。

### 解決方案

`asenv.conf` 檔案中的 `AS_JAVA` 變數永遠會指向 `as-install/jdk`。如果您想要使用不同的 JDK 版本，請手動更新 `asenv.conf` 檔案並變更 `AS_JAVA` 的值。

### 使用 3.5 GB Java 堆疊會導致實例在收到流量時重新啟動 (問題 1169)

#### 說明

當 JVM 堆疊大小設為 3.5 GB 時，如果接收到流量，Communications Server 實例會失敗並重新啟動。

#### 解決方案

確認最大 JVM 堆疊大小已設為 3.0 GB 或更低。

### 僅使用多核心系統中的單一核心時，Communications Server 未正確報告 CPU 使用量 (問題 1344)

#### 說明

在 Solaris 平台上，Communications Server 會根據可用的處理器數目和每個核心的 CPU 使用量，來計算 CPU 的使用量。然而，Communications Server 考量的是核心數目的靜態值，而非 JVM 所使用的核心數目。

#### 解決方案

如果您沒有使用機器中的所有核心，請重新計算 CPU 臨界值。

## 整合負載平衡器

### 於應用程式部署後動態重新配置整合負載平衡器導致記錄檔中出現「發生嚴重問題」訊息 (問題 1161)

#### 說明

如果您在目標機器上修改整合負載平衡器的配置，並於該目標機器上重新部署應用程式，則實例記錄檔會顯示「發生嚴重問題」訊息。

#### 解決方案

這些訊息不會影響整合負載平衡器或實例的功能。您可以忽略這些訊息。

### 使用完整 URI 時，連絡標頭中的 BEKey 參數未正確換碼 (問題 1466)

#### 說明

當您將整合負載平衡器配合會為 BEKey 參數傳回完整 URI 的資料導向規則檔案使用時，連絡標頭中的 BEKey 參數沒有正確換碼。「:」字元未依照 RFC 3261 中的指定正確換碼。

#### 解決方案

無已知解決方案。

## 安裝

**Communications Server 以檔案為基礎的安裝程式不會安裝 Basic3pcc 範例應用程式 (錯誤編號 6894932)**

### 說明

Communications Server 以檔案為基礎的安裝程式不會安裝 Basic3pcc 範例應用程式。此應用程式會透過 JAR 安裝程式提供。

### 解決方案

無已知解決方案。

**Communications Server 安裝程式在 Linux 上當機 (6739013)**

### 說明

執行 Linux 的系統將環境變數 `MALLOC_CHECK_` 設定為 2 時會出現此問題。

### 解決方案

將環境變數 `MALLOC_CHECK_` 設定為 0。執行下列其中一個指令：

- 針對 Bourne shell：

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

- 針對 bash shell：

```
export MALLOC_CHECK_=0
```

- 針對 csh、tcsh shell：

```
setenv MALLOC_CHECK_ 0
```

**64 位元 JDK 安裝失敗 (6796171)**

### 說明

在具有 64 位元 JDK 的 64 位元系統上安裝失敗，因為安裝程式嘗試使用 64 位元 JDK。

### 解決方案

如果您想要在 64 位元系統上安裝 Sun GlassFish Communications Server，請下載 32 位元 JDK 以便在您的 64 位元機器上安裝 Sun GlassFish Communications Server。您需要使用下列指令：`./distribution_filename -javahome 32 位元 JDK 位置路徑`

安裝後，如果要確認 Sun GlassFish Communications Server 使用 64 位元 JDK，請在 `asenv.conf` 檔案中將 `AS_JAVA` 變數的值編輯為指向 64 位元 JDK 安裝。

## 安全性

**Communications Server 在 sun-sip.xml 中未指定 trust-auth-realm-ref 屬性時丟出異常 (CR 6786131)**

### 說明

在 sun-sip.xml 中配置 P-Asserted-Identity 認證時，Communications Server 丟出「未配置範圍」的空指標異常。

### 解決方案

請在 sun-sip.xml 中使用 trust-auth-realm-ref 屬性配置範圍。

## SIP 容器

**SIP 容器無法在已傳送 100 回應時處理取消請求 (問題 712)**

### 說明

SIP 容器無法在已傳送 100 回應時處理取消請求。

### 解決方案

應用程式需要傳送暫時性的回應 (例如 1xx)，以便遠端能夠取消此 INVITE 請求。

**SIP 階段作業和 HTTP 階段作業沒有套用相同的階段作業過期時間模型 (問題 1180)**

### 說明

SIP 階段作業的階段作業過期模型和 HTTP 過期時間邏輯不同。在 HTTP 方面，每當在 HTTP 階段作業中接收到新的 HTTP 請求時，該階段作業會自動延伸，並不會受到應用程式的控制。

而在 SIP 階段作業方面，應用程式會在 SIP 容器的許可下，控制 SipApplicationSession (SAS) 的持續時間。應用程式可以使用 setExpires 方法指出 SAS 何時應過期。setExpires 會相對於呼叫 setExpires 方法的時間來定義過期時間。容器可以修改、拒絕或接受 setExpires 中所指出的持續時間。如果階段作業沒有失效，則會在 setExpires 定義的時間執行 sessionExpired 回呼。在此回呼中，應用程式可以嘗試呼叫新的 setExpires 以延伸 SAS 的持續時間，但仍會受到容器的修改、拒絕或接受控制。

因此，使用與 SipApplicationSession (SAS) 相同的過期時間啟動且位於 HTTP 階段作業的整合應用程式，在 HTTP 階段作業中接收到新的請求時，會發現 SAS 在 HTTP 階段作業前逾時。

## 解決方案

對於 SIP 和 HTTP 階段作業兩者不同的過期時間處理，最好的應付方式是啟動階段作業時使用夠長的 SAS 過期時間，此一過期時間即為應用程式階段作業預期存在 (包括數個 HTTP 請求) 的總時間長度。SAS 使用期限甚至可以設為無限，特別是使用 `invalidateWhenReady` 語義時，在這個情況下，`SipApplicationSession` 會在最後的協定子階段作業失效時失效。SAS 的初始過期時間可以在部署描述元中配置。

如果可以事先估計最大總持續時間，就不需要額外的程式碼，因為在 SAS 過期時讓 Sip 階段作業和 HTTP 階段作業失效非常適當。如果無法事先估計最大持續時間，則 `SipApplicationSession` 可以在其過期後延伸，如以下程式碼片段內容所示。

在 `SipApplicationSessionListener` 實作中，您可以進行類似這樣的作業：

```
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
}
```

## 容器回呼 `sessionExpired` 後 SIP 階段作業仍持續 (問題 1265)

### 說明

這是偶爾會發生的問題。當指出階段作業已移除的 200 通知訊息，以及用戶端在收到該通知時所傳送的訂閱之間發生競爭情況時，SIP 容器偶爾會使用 500「伺服器內部錯誤」訊息回應，而非使用 481「呼叫/作業事件不存在」訊息回應。

## 解決方案

用戶端需要在訂閱到期前儘早更新訂閱。

## Communications Server 會先後扮演 UAS 和代理的角色執行動作並產生 NOP (問題 1432)

### 說明

當它接收到邀請請求時，Communications Server 會先扮演 UAS 的角色，以 1XX 回覆此請求，然後會代理此邀請請求傳送給另一個實例，該實例會以 200「確定」訊息回覆。1xx 會建立內部虛擬分支，而 200 訊息會建立實際分支。在從 B 接收到 200「確定」訊息後，內部虛擬分支應會取消。

## 解決方案

此一異常追蹤不會影響虛擬代理分支的功能。

**getLastAccessedTime 方法未提供精確的結果 (問題 1351)****說明**

SIP 階段作業的 `getLastAccessedTime` 方法並不會提供精確的結果。

**解決方案**

需要精確追蹤 `lastAccessedTime` 的應用程式必須自行將其儲存到 `SipApplicationSession` 之中。

```
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());
}
```

**SIP 偵聽程式在刪除後一段期間內仍維持作用中的狀態 (問題 1294)****說明**

在刪除針對 TCP 和 UDP 請求所配置的 SIP 偵聽程式後，偵聽程式在一段期間內仍維持作用中的狀態。傳送給偵聽程式的 UDP 請求可能會接收到來自偵聽程式的回應。

**解決方案**

無已知解決方案。在一段時間之後，SIP 偵聽程式會停止偵聽 UDP 請求。這個問題並不會影響 TCP 請求。

**Communications Server 在接收到不含「<>」的連絡標頭時丟出異常 (問題 1489)****說明**

Communications Server 在接收到不含「<>」的連絡標頭時丟出異常。根據 SIP RFC 3261，位址中並非一定要使用「<>」。這可能會導致與其他 SIP 相容裝置間的互通性問題。

**解決方案**

請在使用連絡標頭中使用「<>」。

**Communications Server 針對無效的 UUID 值丟出異常 (問題 1494)****說明**

Communications Server 會針對無效的 UUID 值丟出異常，而非傳回 400「錯誤的請求」訊息。UUID 值位於 SIP 連絡標頭的 `sip.instance` 值之中。

### 解決方案

無已知解決方案。

**Windows：**有時 **Communications Server** 沒有收到 UDP 訊息 (無 ID)

### 說明

這是 Windows 作業系統上偶爾會發生的問題。Communications Server 沒有收到 UDP 訊息。

### 解決方案

將下列 JVM 選項依照下面的方式設定，並重新啟動 Communications Server。

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

## SIP 階段作業複製

**整合應用程式使用 SAS 物件做為同步化鎖定時可能發生死結情況 (問題編號 1954)**

### 說明

如果具有 HTTP 和 SIP Servlet 的整合應用程式使用 `sipApplicationSession` 物件做為同步化 SIP 和 HTTP 背景工作執行緒之間存取的鎖定，會發生死結情況。

### 解決方案

請不要使用 `sipApplicationSession` 做為同步化鎖定。而是使用 `sipApplicationSession` 中設為屬性的可序列化物件做為鎖定。



## Sun GlassFish 企業伺服器的已知問題與限制

---

本章說明 Sun GlassFish Enterprise Server 2.1.1 軟體的已知問題以及相關的解決方案。這些問題特別會影響建立於 Sun GlassFish Enterprise Server 2.1 上的通訊伺服器。如果沒有看到指出特定平台的摘要聲明，表示這個問題會發生在所有平台上。這些資訊按以下章節進行分類：

- 第 25 頁的「管理」
- 第 31 頁的「應用程式用戶端」
- 第 32 頁的「隨附的 Sun JDBC 驅動程式」
- 第 33 頁的「部署」
- 第 34 頁的「說明文件」
- 第 35 頁的「EJB」
- 第 35 頁的「Java EE Tutorial」
- 第 35 頁的「Java 持續性」
- 第 36 頁的「生命週期管理」
- 第 37 頁的「記錄」
- 第 37 頁的「Message Queue」
- 第 39 頁的「監視」
- 第 40 頁的「安全性」
- 第 41 頁的「Web 容器」

### 管理

本節介紹已知的管理問題以及相關的解決方案。

#### Communications Server 不會偵測與叢集活動訊號連接埠的衝突 (問題編號 1967)

##### 說明

叢集建立時，Communications Server 會為其隨機指定 1026 到 45556 之間的活動訊號連接埠。對於預設叢集 (亦即 Communications Server 安裝所建立的預設叢集)，系統會隨機選取介於 0 至 45556 之間的數字。叢集建立程序不會精確偵測是否已有其他服務使用該活動訊號連接埠。

## 解決方案

如果自動叢集建立配置選取了與另一服務(已使用該連接埠)衝突的活動訊號連接埠，請將叢集活動訊號連接埠更新為系統未使用的連接埠。

要變更叢集的活動訊號連接埠，請使用下列 `asadmin` 指令：

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

## 執行 64 位元 Linux 的 NFS 伺服器停止建立網域(問題編號 1961)

### 說明

在執行 64 位元 Linux 的 NFS 伺服器上，`asadmin create-domain` 指令在嘗試於掛載 Network File System (NFS) 的檔案系統上建立網域時可能會失敗。

## 解決方案

無已知解決方案。

## 自動重建大型記錄檔時發生效能降低情形(6718611)

### 說明

自動重建大型記錄檔時，觀察到回應時間略為增加的情形。

## 解決方案

修改記錄程式設定中「檔案自動重建限制」和「檔案自動重建時間限制」的值，可以將效能降低情形減至最低。這些特性的值會視您的應用程式和環境而定。

## 無法針對 IBM MQ 部署常規 RA 資源配接卡(問題 6605)

### 說明

針對 IBM Message Queue 產品部署常規 RA 配接卡失敗。`server.policy` 檔案授予的權限如下。

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

## 解決方案

變更 `server.policy` 檔案中的權限，如下所示：

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

## 獨立實例有時會從其他實例取得檔案 (6698604)

### 說明

在某些情況下，系統會將安裝在 DAS 上但需與特定實例同步的檔案，真的傳送至其他實例。

### 解決方案

無已知解決方案。

## start-cluster 指令產生的啟動訊息太詳細 (6728317)

### 說明

即使啟動期間失敗的元件並不重要，asadmin start-cluster 指令仍會顯示過多訊息。請參閱下列不重要元素 (與叢集中的實例有關) 失敗時產生的指令輸出範例：

```
./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.

## 解決方案

無已知解決方案。可以略過這些 (異常) 訊息。

**如果 domain1 不存在，則 package-appclient 程序檔無法使用 (6171458)**

## 說明

依預設，在 *as-install/lib/package-appclient.xml* 中，*asenv.conf* 指向的 *domain1* 之 *AS\_ACC\_CONFIG* 變數有一個程序內定值。如果刪除 *domain1* 並建立新網域，則不會使用新網域名稱更新 *AS\_ACC\_CONFIG* 變數，這將導致 *package-appclient* 程序檔失敗。

## 解決方案

執行下列動作之一：

1. 保留 *domain1* 的完整，並在周圍建立其他網域。
2. 移除 *domain1* 並使用新網域名稱替代 *as-install/lib/package-appclient.xml* 中 *domain1* 的程序內定值。

如果沒有 *domain1*，則每次建立新網域時均必須執行此作業。

**不支援同時啟動 Application Server 和其他 JMX 代理程式 (6200011)**

## 說明

在 Application Server 上可以配置 J2SE 1.4.x、5.0 或更高版本。啟動 JMX 代理程式是 J2SE 5.0 平台不可或缺的功能。如果您在伺服器啟動時明確設定系統特性，則會啟動此功能。

範例值包含：

```
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"
```

在配置 JMX 特性並啟動伺服器之後，會在 Application Server Virtual Machine 上啟動新的 jmx-connector 伺服器。此作業會產生不良副作用，即負面影響管理功能，並且 Application Server 管理主控台和指令行介面可能會產生未預期的結果。問題在於在內建 jmx-connector 伺服器和新的 jmx-connector 伺服器之間存在某些衝突。

## 解決方案

在使用 jconsole (或其他任何 JMX-compliant 用戶端) 時，考量重複使用透過 Application Server 啟動程序檔啟動的標準 JMX 連接器伺服器。

伺服器啟動後，server.log 中會顯示與下行類似的文字。您可連接至其中指定的 JMXService URL，並在成功提供憑證之後執行相同的管理/配置作業，例如：

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

.asadmintruststore 檔案未在文件中說明 (6315957)

## 說明

在文件中未說明 .asadmintruststore 檔案。如果伺服器管理員的 home 目錄中不存在此檔案，則升級在此伺服器上代管的某些應用程式時，可能會遇到嚴重錯誤。

## 解決方案

- 如果可能，應由安裝此伺服器的使用者執行 `asadmin start-domain domain1` 指令。
- 如果未由該使用者執行，則應從安裝使用者的 home 目錄中，將 .asadmintruststore 移動或複製到執行使用者的 home 目錄。
- 請注意，如果將此檔案從安裝使用者的 home 目錄移動 (而非複製) 到執行使用者的 home 目錄，您可能會遇到應用程式升級問題 (例如在錯誤 6309079、6310428 和 6312869 中說明的問題)，因為在升級/安裝使用者 (在 Java ES 中一般為 root) 的 home 目錄中將不再包含 .asadmintruststore 檔案。

## 叢集實例由於連接 JMS 代理程式逾時而無法啟動 (6523663)

### 說明

Communications Server 叢集實例的預設 MQ 整合模式為 LOCAL。當 Communications Server 的安裝位置 (PATH) 很長 (顯示為「not short」) 時，imqbrokersvc.exe 便會在叢集實例啟動時當機。此問題屬於 imqbrokersvc 中的記憶體分配問題。

### 解決方案

叢集實例的 JMS 服務類型必須從預設的 LOCAL 變更為 REMOTE。在此配置中，所有實例都指回 DAS 代理程式。請依照以下說明在 REMOTE 模式中配置叢集。

---

**備註**– 當使用 REMOTE 模式時，所有實例均使用一個代理程式 (DAS)，因此在 Communications Server 叢集啟動時，不會建立任何代理程式叢集。如需詳細資訊，請參閱 <http://www.glassfishwiki.org/gfwiki/attach/OnePagersOrFunctionalSpecs/as-mq-integration-gfv2.txt> 單頁短文第 4.1 節第 iii 點的「自動叢集」。上述功能將無法使用！

---

## ▼ 使用指令行

**開始之前** 依據您的環境修改連接埠與密碼檔案。請注意，在以下說明中，叢集名稱為 racluster，DAS 管理連接埠為 5858，DAS JMS 連接埠為 7676。

- 1 修改叢集配置，將 JMS 類型變更為 REMOTE。

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

- 2 建立與 DAS JMS 主機對應的 JMS 主機。

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

- 3 將預設 JMS 主機設定為在上一步驟中建立的 DAS JMS 主機。

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

## ▼ 使用管理 GUI

- 1 移至 [配置] -> [cluster-name-config] -> [Java 訊息服務] -> [JMS 主機]。
- 2 按一下 [新增] 以建立新的 JMS 主機，將其命名為 dashost。
- 3 輸入與 DAS 的 JMS 服務對應的配置設定，預設設定如下所示：

- 主機名稱：localhost
- 連接埠：7676
- 管理使用者：admin
- 密碼：admin

依據您 DAS JMS 服務的需要修改上述設定。

- 4 瀏覽回 [Java 訊息服務] 標籤，然後將 JMS 服務類型變更為 REMOTE (預設為 LOCAL)。
- 5 從 [default-jms-host] 下拉式清單中選擇 [dashost]。
- 6 儲存變更，然後啟動節點代理程式或叢集。

無法在 Netscape 8.1.3、Mozilla 1.7 和 Safari 2.0.4 等瀏覽器中顯示 jmaki 圖表 (6543014)

### 說明

當嘗試使用某些不支援的瀏覽器顯示 [記錄統計監視] 頁面中的圖表時，可能會丟出以下錯誤：

```
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
```

### 解決方案

使用支援的瀏覽器。請參閱第 13 頁的「瀏覽器」，以取得 Communications Server 支援的瀏覽器清單。

在 AIX 中使用自訂主密碼執行 create-domain 指令失敗 (6628170)

### 說明

在 AIX 作業系統上，嘗試使用自訂主密碼建立網域失敗，同時會顯示以下錯誤：

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

## 應用程式用戶端

本節介紹已知的應用程式用戶端問題以及相關的解決方案。

應用程式用戶端歸檔中封裝的程式庫 JAR 會置換清單檔案 (6193556)

### 說明

如果您在用戶端 JAR 內部具有頂層的 JAR 檔案 (這種情況下，為 reporter.jar)，當您部署用戶端 JAR 時，該 JAR 的可取得的檔案將置換用戶端 JAR 的可取得的檔案。

## 解決方案

目前尚無解決方案。

### ACC 一直嘗試連線至 localhost:3700 (6527987)

## 說明

應用程式用戶端始終嘗試連線至 localhost:3700。問題在於呼叫用戶端程式碼之前，需要讀取數個系統特性。

## 解決方案

將以下內容設定為系統特性 (-D 位於 JAVA\_CMD 中)。請勿在應用程式用戶端程式碼中設定它們：

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

### ASQuickStartup 中斷了 SocketChannel.keyFor()，傳回空值，而非 SelectionKey (問題追蹤代碼 3027)

## 說明

在數個選擇器上註冊 SocketChannel 時，執行 socketChannel.keyFor(lastRegisteredSelector) 會傳回 Null，而非 SelectionKey。

## 解決方案

此問題與 JDK 錯誤 6562829 相關，預計在 6.0 U3 中修正。Communications Server 中已經提供了解決方法，以便在呼叫 keyFor API 之前展開選擇器。這樣，將繼續使用 keyFor，直至修正了 JDK 錯誤。

## 隨附的 Sun JDBC 驅動程式

本節介紹已知的附帶的 Sun JDBC 驅動程式問題以及相關的解決方案。

### PreparedStatement 錯誤 (6170432)

#### 描述 1

如果應用程式在一個作業事件中產生 3000 個以上 PreparedStatement 物件，DB2 可能會遇到以下錯誤：

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



### 解決方法 1

將以下特性增加到連線區定義中，以使用更大的動態區段值來使驅動程式重新連結 DB2 封裝：

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

### 描述 2

與上述 `PreparedStatement` 錯誤相關，可能丟出其他錯誤訊息：

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

### 解決方法 2

增大 DB2 伺服器配置參數 `APPLHEAPSZ`。合適的值為 4096。

### 描述 3

隔離層級 `TRANSACTION_SERIALIZABLE`。如果應用程式使用隔離層級 `TRANSACTION_SERIALIZABLE`，並使用上述建議的參數之一，則應用程式在取得連線時可能會掛機。

### 解決方法 3

若要為連線設定所需的隔離層級，必須在此隔離層級建立相應的連線區。

### Java DB 不會在重新啟動機器或啟動 Application Server 之後啟動 (6515124)

#### 說明

隨附的 Java DB 資料庫不會在重新啟動主機系統或 Solaris 區域，或啟動 Communications Server 之後自動重新啟動。這不是錯誤，而是所有隨附應用程式或協力廠商應用程式的預期運作方式。問題在於 Java DB 必須在 Communications Server 實例之前啟動。

#### 解決方案

重新啟動主機電腦或 Solaris 區域之後，務必在啟動 Communications Server 之前啟動 Java DB，例如：

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

## 部署

### 有時在叢集上進行自動部署會失敗 (6610527)

#### 說明

在已配置為支援叢集的網域中，時序問題有時會造成自動部署失敗。不支援叢集的網域尚未發生此問題。

## 解決方案

使用以下解決方案之一：

- 按以下方式使用自動部署：
  - 循序自動部署個別的應用程式。
  - 每個應用程式的自動部署作業之間需加入延遲時間。
- 使用管理主控台 GUI 或指令行以手動方式部署應用程式。

## JSP 編譯未使用應用程式專用類別載入器 (6693246)

### 說明

JSP 編譯未使用應用程式專用類別載入器 (applibs 或 --libraries)。因此參照這些 JAR 的 JSP 將無法進行編譯。

## 解決方案

無已知解決方案。

## 說明文件

本節說明已知的文件問題以及相關的解決方案。

### Javadoc 不一致 (不同的 ID)

用於多個 AMX 介面與方法的 Javadoc 缺漏或不正確：

- ConnectorConnectionPoolStats 和 AltJDBCConnectionPoolStats 中缺少 NumConnAcquired 和 NumConnReleased 統計的獲取方法。這些獲取方法將在未來的發行版本中加入為 `getNumConnAcquired()` 和 `getNumConnReleased()`。
- 在 EJBCacheStats 中呼叫以下方法將丟出異常：  
`getPassivationSuccesses()`、`getExpiredSessionsRemoved()`、`getPassivationErrors()` 和 `getPassivations()`。這將在未來的版本中進行修正。
- 啟動伺服器後，AMX MBeans 可能需要數秒鐘才能完全註冊和使用。未來的版本將可能確定完全載入 AMX MBeans 的時間。
- 常數 `XTypes.CONNNECTOR_CONNECTION_POOL_MONITOR` 拼寫錯誤 ("NNN")。這將在未來的版本中進行校正。

## 隨附的 ANT 丟出 `java.lang.NoClassDefFoundError` (6265624)

### 說明

在「主」執行緒中丟出以下異常 `java.lang.NoClassDefFoundError: org.apache.tools.ant/launch/Launcher`。

## 解決方案

不建議將隨附的 ANT 用於 Communications Server 以外的軟體。

## EJB

### 無法在 HandlerChain 中使用資源注入 (6750245)

#### 說明

由於 EJB 初始化的順序，導致無法在 HandlerChain 中使用資源注入。

#### 解決方案

無已知解決方案。

## Java EE Tutorial

當使用 [管理主控台] 建立任何資源時，請使用 [目標] 標籤將伺服器指定為目標。如果使用指令行或 asant 目標，則伺服器為預設目標，無需其他動作。

## Java 持續性

### TopLink 預期 Collection 欄位/特性可複製 (代碼 556)

#### 說明

如果使用 `java.util.Arrays.asList()` API 將 `Object[]` 轉換為 `Collection`，JDK 便會傳回不可複製的 `java.util.ArrayList` 之實作。這樣會導致出現以下異常：

```
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
```

此問題將以 [https://glassfish.dev.java.net/issues/show\\_bug.cgi?id=556](https://glassfish.dev.java.net/issues/show_bug.cgi?id=556) 持續追蹤。

#### 解決方案

使用其建構子建立另一集合；例如：

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

### GenerationType.IDENTITY 和 DataDirect 驅動程式搭配 SyBase (問題追蹤代碼 2431)

#### 說明

將 DataDirect 驅動程式與 SyBase 配合使用時，嘗試插入使用 `GenerationType.IDENTITY` 的實體會失敗。失敗的原因在於 DataDirect 驅動程式會為參數化預備陳述式建立已儲存程序。

## 解決方案

在 domain.xml 檔案中，設定對應資料來源的特性 PrepareMethod=direct。

## 生命週期管理

本節說明已知的生命週期管理問題以及相關的解決方案。

### 設定 ejb-timer-service 特性導致 set 指令失敗 (6193449)

#### 說明

將 ejb-timer-service 特性 minimum-delivery-interval 設定為 9000 之後，嘗試將 ejb-timer-service 特性 redelivery-interval-in-millis 設定為 7000 的行為會導致 set 指令失敗，並出現以下錯誤：

```
[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 是同一個計時器的兩次傳送之間的最小間隔持續時間。
- redelivery-interval-in-millis 是計時器服務在 ejbTimeout 失敗後嘗試重新傳送之前的等待時間。

問題在於，將重新傳送間隔特性與最小傳送特性相聯繫的邏輯不正確，並阻止您使用 GUI 或 CLI 設定任何最小傳送間隔大於重新傳送間隔的值。

minimum-delivery-interval-in-millis 的設定必須始終等於或高於 ejb-timer-service 特性 redelivery-interval-in-millis 的設定。問題在於，應用程式伺服器具有錯誤的驗證檢查可驗證 redelivery-interval-in-millis 的值是否大於 minimum-delivery-interval-in-millis 的值。

## 解決方案

使用這些特性的預設值，如下所示：

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

這些預設值以外的值將會產生錯誤。

### 當列出非 DAS 配置內的 JMS 實體目標時丟出錯誤 (6532532)

#### 說明

如果您要嘗試檢視使用 default-config 的 JMS 實體目標，便會出現一則錯誤訊息。

## 解決方案

這是預期的運作方式。在 Communications Server 2.0 中，default-config 是配置資訊範本，因此無法針對 default-config 執行 JMS 作業 (如 list 和 create)。但是，可以針對叢集或獨立實例的配置執行 JMS 作業。

**僅適用於 Win2003：非分頁池記憶體發生錯誤，中斷 tcp 堆疊和 richaccess 測試 (6575349)**

## 說明

(僅適用於 Windows 2003) 執行豐富存取功能時，Windows 2003 系統上會出現記憶體錯誤。出現此問題是因為 Win32 非分頁池一直在增長，最終導致整個 TCP/IP 堆疊出現錯誤。一旦出現此故障，TCP/IP 堆疊便處於可回復狀態，復原它的唯一方式是重新啟動 Windows 2003 系統。

## 解決方法

有兩種解決方法可解決這個問題：

- 將 domain.xml 的 http-listener 屬性配置成 blocking-enabled="true" 以使用 Grizzly 封鎖模式，或者增加以下 http-listener 特性：

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

- 使用 Windows Vista 或 Windows XP。

## 記錄

本節說明已知的記錄問題以及解決方案。

**設定 access, failure 的除錯敘述導致 Application Server 啟動當機 (6180095)**

## 說明

設定 JVM 的 java.security.debug 選項將會導致伺服器實例啟動因為死結而凍結；例如，在 domain.xml 中進行以下設定會導致該問題：

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

## 解決方案

目前尚無解決方案。請避免設定此標幟。

## Message Queue

本節說明已知的 Java Message Queue 問題以及相關的解決方案。

在與時間相關的特定情況下，JMS 重新連線無法成功完成  
(6173308、6189645、6198481、6199510、6208728)

### 說明

在與時間相關的方案中重新連線失敗可能是由多個問題導致的。

### 解決方案

您可以透過以下方法解決這些問題：

- 重新啟動涉及的代理程式
- 重新啟動涉及的 Application Server 實例

**如果 MQ 代理程式未啟動，則伺服器無法啟動 (6740797)**

### 說明

如果您將 JMS 配置為 REMOTE，則 MQ 代理程式未啟動時，企業伺服器亦無法啟動。

### 解決方案

設定下列 JVM 選項，如下所

示：`com.sun.enterprise.jms.CONNECT_MQ_LAZILY=true`。設定此 JVM 選項後，如果 MQ 代理程式未啟動，您可以啟動 Communications Server。然而，建議您在啟動伺服器之前啟動 MQ。

**MQ 代理程式無法在帶有叢集設定檔的 Linux 上啟動 (6524871)**

### 說明

在 Linux 系統上建立帶有叢集設定檔的網域之後，可能會遇到 `java.lang.OutOfMemoryError: Java heap space` 錯誤，由於 MQ 代理程式未啟動，伺服器實例也可能無法重新啟動。出現此狀況之後，系統便無法再回復。出現此問題是因為 `/etc/hosts` 檔案配置錯誤；具體的說，伺服器主機名稱將指向迴路位址 `127.0.0.1`。

### 解決方案

依設計，MQ 代理程式叢集不能在網路裝置配置為指向迴路位址的情形下啟動。這不是錯誤。解決方案是確保 Communications Server 主機的 `/etc/hosts` 檔案不指向 `127.0.0.1`。

**在升級前載入 `imqjmsra.jar` 時，會產生舊類別與新類別不相符的情形 (6740794)**

### 說明

應用程式伺服器啟動期間，伺服器會檢查 Message Queue 版本。如果 Message Queue 版本錯誤，伺服器會使用 `imqjmsra.jar` 升級。應用程式伺服器必須重新啟動才可使用此升級 JAR 及其類別。只有當 Message Queue 單獨升級時，或是單獨修補應用程式伺服器時，才會發生此情況。此情況的副作用是有時無法啟動應用程式伺服器。

## 解決方案

您需要將 Message Queue 與應用程式伺服器維持在同一修補層級，或重新啟動應用程式伺服器。

## 監視

本節介紹已知的監視問題和相關的解決方案。

**有些 HTTP 服務監視統計不能提供有用的資訊，應該將其忽略 (6174518)**

### 說明

檢視 HTTP 服務的某些元素的監視統計資料時，出現的某些值與目前值不相符，或始終為 0。具體的說，下列 HTTP 服務統計資料顯示的資訊並不適用於應用程式伺服器，應予以忽略：

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

## 解決方案

在以後的版本中，將移除這些監視，並由更適當的資訊取代。

**從管理 UI 開啓 JNDI 瀏覽會在 server.log 中傾印大量異常 (6591734)**

### 說明

從管理 GUI 開啓 JNDI 瀏覽器時，會丟出許多異常。

## 解決方案

目前尚無解決方案。

**對於繁體中文、簡體中文和西班牙文，無法以 asadmin monitor 指令監視 JVM (7076)**

### 說明

在繁體中文、簡體中文和西班牙文語言環境下執行 asadmin monitor --type jvm server 會發生異常狀況。

## 解決方案

請從【管理主控台】檢視 JVM 資訊，或在英文語言環境中執行 asadmin monitor 指令。

## 安全性

本節說明與 Communications Server 和 Web 應用程式安全性及憑證有關的已知問題和相關解決方案。

### 負荷過大時，SSL 分析藍本出現記憶體不足的錯誤 (JDK 6 問題 23)

#### 說明

在負荷過大的情況下執行某些 SSL 分析藍本時，JDK6 Sun PKCS11 提供者中的 JDK 錯誤 (請參閱：[https://jdk6.dev.java.net/issues/show\\_bug.cgi?id=23](https://jdk6.dev.java.net/issues/show_bug.cgi?id=23)) 可能會導致出現記憶體不足的錯誤。

#### 解決方案

如果您遇到此問題，請從 JRE 安裝內的 `java.security` 檔案中移除 `sun.security.pkcs11.SunPKCS11` 提供者。

### SSL 終止未起作用 (6269102)

#### 說明

SSL 終止無法正常工作；如果為 SSL 終止設定了負載平衡程式 (硬體)，Communications Server 會在重新導向期間將協定從 `https` 變更為 `http`。

#### 解決方案

在硬體負載平衡器與 Communications Server 之間增加軟體負載平衡器。

### SSL 出現通訊端連線錯誤 (6492477)

#### 說明

由於 JVM 錯誤，在 HTTP 偵聽程式上將 `security-enabled` 設定為 `true` 時，某些 JDK 版本會出現錯誤問題。具體的說，重現此錯誤的步驟如下：

1. 在 HTTP 偵聽程式上，將 `security-enabled` 設定為 `true`：

```
<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. 在 quicklook 測試結尾處加入停止網域的註釋。
3. 執行 quicklook 測試。
4. 檢查通訊埠的使用：

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

下面顯示的是正在使用：



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

GlassFish 網站已經追蹤此問題，網址爲：[https://glassfish.dev.java.net/issues/show\\_bug.cgi?id=849](https://glassfish.dev.java.net/issues/show_bug.cgi?id=849)。

## 解決方案

升級到最新的 JDK 版本。

## Web 容器

本節說明已知的 Web 容器問題以及相關的解決方案。

**在 Windows 上使用 `--precompilejsp=true` 部署應用程式可能會鎖定應用程式中的 JAR 檔案，從而導致後面的取消部署或重新部署作業失敗 (5004315)**

### 說明

如果您在 Windows 上部署應用程式時請求 JSP 的預先編譯，則以後無法按預期嘗試取消部署或重新部署該應用程式 (或任何具有相同模組 ID 的應用程式)。問題在於 JSP 預先編譯會開啓應用程式中的 JAR 檔案，但不會關閉它們，同時 Windows 會防止取消部署刪除這些檔案或防止重新部署置換它們。

請注意，取消部署會進行到某個地步，此時會依據邏輯將該應用程式從 Application Server 中移除。還請注意，`asadmin` 公用程式不會傳回任何錯誤訊息，但應用程式的目錄和鎖定的 jar 檔案會保留在伺服器上。伺服器的記錄檔將包含描述無法刪除檔案和應用程式目錄的訊息。

取消部署失敗後會嘗試重新部署應用程式，因為伺服器會嘗試移除現有檔案與目錄，此嘗試仍失敗。如果您嘗試部署使用與原來部署的應用程式具有相同模組 ID 的任何應用程式，便會出現這種情況，因為伺服器使用該模組 ID 選擇目錄名稱以存放應用程式檔案。

基於同樣原因，不先取消部署即嘗試重新部署應用程式將會失敗。

### 診斷

如果您嘗試重新部署應用程式或在取消部署之後再部署該應用程式，`asadmin` 公用程式會傳回一個如下類似錯誤。

```
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.
```

## 解決方案

如果您在部署應用程式時指定 `--precompilejsp=false` (預設的設定)，則不會出現此問題。請注意，第一次使用應用程式將觸發 JSP 編譯，因此第一次請求的回應時間會比以後的請求的回應時間長。

還請注意，如果進行預編譯，應先停止並重新啟動伺服器，然後再取消部署或重新部署應用程式。關機會釋放鎖定的 JAR 檔案，因此重新啟動後才能成功取消部署或重新部署。

## 無法使用基於 Servlet 2.4 的 web.xml (包含空的 `<load-on-startup>` 元素) 部署 WAR (6172006)

### 說明

web.xml 中的選擇性 `load-on-startup` servlet 元素表示要載入相關的 servlet 並將其初始化為宣告該 servlet 的 Web 應用程式啟動的一部分。

該元素的可選內容是一個整數，表示要載入並初始化與 Web 應用程式之其他 servlet 相關的 servlet 的順序。只要在啟動其含有的 Web 應用程式過程中載入並初始化 servlet，空的 `<load-on-startup>` 即表示順序錯誤。

web.xml 的 Servlet 2.4 模式不再支援空的 `<load-on-startup>`，亦即使用基於 Servlet 2.4 的 web.xml 時，必須指定整數。若指定空的 `<load-on-startup>` (如 `<load-on-startup/>`)，web.xml 將無法針對 web.xml 的 Servlet 2.4 模式進行驗證，進而導致部署 Web 應用程式失敗。

返回至相容性問題。指定空的 `<load-on-startup>` 仍可使用基於 Servlet 2.3 的 web.xml。

## 解決方案

使用基於 Servlet 2.4 的 web.xml 時，指定 `<load-on-startup>0</load-on-startup>`，以表示 servlet 載入順序並不重要。

## 無法在受資源約束的伺服器上編譯 JSP 頁面 (6184122)

### 說明

存取 JSP 頁面後無法編譯，且伺服器記錄含有錯誤訊息「Unable to execute command」，以及以下堆疊追蹤：

```
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)
```

## 解決方案

將 JSP 編譯切換「fork」設定為「false」。

有兩種方法可以執行此操作：

- 將 *domain-dir/config/default-web.xml* 中 JspServlet 的 fork init 參數全部設定為 false：

```
<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>
```

- 依據每個 Web 應用程式，將 *sun-web.xml* 中的 fork JSP 配置特性設定為 false：

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

以上任何一種設定都將阻止 ant 產生用於 javac 編譯的新程序。

## Web 服務

本節說明已知的 Web 容器問題以及相關的解決方案。

在 JDK6 u4 b3 上 wscompile 失敗，並出現「**package javax.xml.rpc does not exist**」的訊息 (6638567)

### 說明

對於 JDK 6 Update 4，wscompile ant 作業會失敗。針對每個 JAX-RPC API 類別，會顯示下列錯誤訊息：

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

### 解決方案

在執行 wscompile ant 作業之前，請確認在類別路徑中已指定 *javaee.jar*，而非 *j2ee.jar*。

