



# Notes de version de Sun GlassFish Communications Server 2.0



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

Référence : 821-1049-10  
Octobre 2009

Copyright 2009 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. Tous droits réservés.

Sun Microsystems, Inc. détient les droits de propriété intellectuelle de la technologie utilisée par le produit décrit dans le présent document. Notamment, mais non exclusivement, ces droits de propriété intellectuelle peuvent inclure un ou plusieurs brevets des États-Unis ou des demandes de brevet en attente aux États-Unis et dans d'autres pays.

Droits énoncés par le gouvernement américain – Logiciel commercial. Les utilisateurs du gouvernement sont soumis au contrat de licence standard de Sun Microsystems, Inc. ainsi qu'aux dispositions applicables du FAR et de ses suppléments.

Cette distribution peut intégrer des éléments conçus par des tiers.

Il est possible que des parties du produit soient dérivées des systèmes Berkeley BSD, concédés en licence par la University of California. UNIX est une marque déposée aux États-Unis et dans d'autres pays, exclusivement concédée en licence par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, le logo Solaris, le logo Java Coffee Cup, docs.sun.com, Java, Java et Solaris sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc. aux États-Unis et dans d'autres pays. Toutes les marques déposées SPARC sont utilisées sous licence et sont des marques commerciales ou déposées de SPARC International, Inc. aux États-Unis et dans d'autres pays. Les produits portant les marques déposées SPARC sont constitués selon une architecture développée par Sun Microsystems, Inc.

OPEN LOOK et l'interface graphique utilisateur Sun<sup>TM</sup> sont développés par Sun Microsystems, Inc. pour ses utilisateurs et ses concessionnaires. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces utilisateur visuelles ou graphiques pour l'industrie de l'informatique. Sun est sous licence non-exclusive de Xerox pour Xerox Graphical User Interface, dont la licence couvre également les détenteurs de licence Sun qui implémentent OPEN LOOK GUIs en accord avec les contrats de licence écrits de Sun.

Les produits couverts et les informations contenues dans cette publication sont contrôlés par les lois régissant les exportations aux États-Unis et peuvent être soumises aux lois régissant les exportations ou les importations dans d'autres pays. L'utilisation d'armes nucléaires, de missiles, d'armes biologiques et chimiques ou d'armes nucléaires maritimes, qu'elle soit directe ou indirecte, est strictement interdite. Son exportation ou sa réexportation vers des pays soumis à l'embargo américain ou à des entités exclues des listes d'exportation américaines, notamment mais pas exclusivement, les personnes et pays figurant sur des listes noires, est strictement interdite.

LA DOCUMENTATION EST FOURNIE « EN L'ÉTAT » ET TOUTES LES AUTRES CONDITIONS, DÉCLARATIONS ET GARANTIES EXPRESSES OU TACITES SONT FORMELLEMENT EXCLUES, DANS LA MESURE AUTORISÉE PAR LA LOI APPLICABLE, Y COMPRIS NOTAMMENT TOUTE GARANTIE IMPLICITE RELATIVE À LA QUALITÉ MARCHANDE, À L'APTITUDE À UNE UTILISATION PARTICULIÈRE OU À L'ABSENCE DE CONTREFAÇON.

# Présentation

---

Sun GlassFish™ Communications Server ajoute les capacités de la technologie de Servlet Session Initiation Protocol (SIP) à Sun GlassFish Enterprise Server.

Sun GlassFish Communications Server est sans frais dans le cadre d'activités de développement, de déploiement et de redistribution. Si vous vous intéressez à la redistribution, contactez le service des ventes sur [de l'OEM Sun](#) pour obtenir une licence de redistribution.

Sun GlassFish Communications Server est un serveur d'applications facile, rapide et leader sur le marché, basé sur la plate-forme Java™, Enterprise Edition (Java EE), et la technologie SIP pour le développement et la distribution d'applications et de services Web. Sun GlassFish Communications Server permet de générer des applications SIP et des applications convergentes compatibles JSR 289. Il permet d'accroître les performances tout en offrant des fonctions de clustering et de disponibilité élevée aux services évolutifs qui sont capables de fonctionner malgré une défaillance matérielle ou logicielle.

- “À propos de ces notes” à la page 3
- “Fonctions d'accessibilité” à la page 4
- “Documentation connexe” à la page 4
- “Comment signaler des problèmes et apporter des commentaires” à la page 5
- “Vos commentaires sont les bienvenus” à la page 6
- “Ressources Sun supplémentaires” à la page 6

## À propos de ces notes

Ces notes de version contiennent des informations importantes disponibles au moment de la commercialisation de Sun GlassFish Communications Server 2.0. Vous y trouverez notamment des renseignements sur les nouvelles fonctions, les améliorations, les restrictions et problèmes connus, etc. Lisez ce document avant d'utiliser Communications Server 2.0.

Vous trouverez la dernière version de ces notes de version sur le [site Web de la documentation \(http://docs.sun.com/coll/1343.10\)](http://docs.sun.com/coll/1343.10) de Communications Server. Consultez ce site Web avant d'installer et de configurer votre logiciel, puis consultez-le régulièrement pour vous procurer la documentation concernant le produit et les notes de mise à jour les plus récentes.

Des URL de sites tiers, qui renvoient à des informations complémentaires connexes, sont référencés dans ce document.

---

**Remarque** – Sun ne saurait être tenu responsable de la disponibilité des sites Web tiers mentionnés dans ce manuel. Sun décline toute responsabilité quant au contenu, à la publicité, aux produits ou tout autre matériel disponibles dans ou par l'intermédiaire de ces sites ou ressources. Sun ne pourra en aucun cas être tenu responsable, directement ou indirectement, de tous dommages ou pertes, réels ou invoqués, causés par ou liés à l'utilisation des contenus, biens ou services disponibles dans ou par l'intermédiaire de ces sites ou ressources.

---

## Historique de révision des notes de mise à jour

Cette section répertorie les modifications apportées aux présentes notes de version après la publication initiale de Sun GlassFish Communications Server.

TABLEAU 1-1 Historique de révision des notes de mise à jour

Date de révision	Description
Octobre 2009	Version FCS de Communications Server 2.0.

## Fonctions d'accessibilité

Pour obtenir la liste des fonctions d'accessibilité mises à disposition depuis la publication de ce média, consultez les évaluations de produit de la Section 508, disponibles sur demande auprès de Sun, afin de déterminer les versions les mieux adaptées au déploiement des solutions accessibles. Vous trouverez les versions mises à jour des applications sur :

<http://sun.com/software/javaenterprisesystem/get.html>

Pour obtenir des informations sur l'engagement de Sun en matière d'accessibilité, consultez la page Web <http://sun.com/access>.

## Documentation connexe

TABLEAU 1-2 Manuels du jeu de documentation de Communications Server

Titre du manuel	Description
<i>Centre de documentation</i>	Rubriques de la documentation de Communications Server classées par tâche et sujet.
<i>Notes de version</i>	Dernières informations sur le logiciel et la documentation. Comprend un récapitulatif complet sous forme de tableau du matériel, des systèmes d'exploitation, du kit de développement Java (JDK™) et des pilotes de base de données pris en charge.
<i>Guide de démarrage</i>	Procédure de démarrage de Communications Server.

TABLEAU 1-2 Manuels du jeu de documentation de Communications Server (Suite)

Titre du manuel	Description
<i>Guide d'installation</i>	Installation du logiciel et de ses composants.
<i>Guide de déploiement d'application</i>	Déploiement des applications et composants d'application pour Communications Server. Ce manuel comprend des informations sur les descripteurs de déploiement.
<i>Guide du développeur</i>	Création et implémentation d'applications sur la plate-forme Java, Enterprise Edition (plate-forme Java EE) destinées à être utilisées sous Communications Server suivant le modèle des normes Java ouvertes pour les composants EE et les API. Ce manuel comprend des informations sur les outils de développement, la sécurité, le débogage et la création de modules de cycle de vie.
<i>Didacticiel Java EE 5</i>	Utilisation des technologies de la plate-forme Java EE 5 et des API pour le développement d'applications Java EE.
<i>Didacticiel Java WSIT</i>	Développement d'applications Web à l'aide de WSIT (Web Service Interoperability Technologies). Ce manuel explique comment, quand et pourquoi utiliser les technologies WSIT et présente les fonctionnalités et options que chaque technologie prend en charge.
<i>Guide d'administration</i>	Administration système pour Communications Server, notamment configuration, contrôle, sécurité, gestion des ressources et gestion des services Web.
<i>Guide d'administration de la haute disponibilité</i>	Configuration de clusters, gestion des agents de noeuds et utilisation d'équilibrateurs de charge.
<i>Guide de référence d'administration</i>	Modification du fichier de configuration de Communications Server, <code>domain.xml</code> .
<i>Guide de réglage des performances</i>	Réglage de Communications Server pour optimiser ses performances.
<i>Manuel de référence</i>	Commandes d'utilitaire disponibles avec Communications Server ; elles sont rédigées comme des pages de manuel. Ce manuel décrit également l'interface de ligne de commande <code>asadmin</code> .

## Comment signaler des problèmes et apporter des commentaires

Si vous rencontrez des problèmes avec Sun GlassFish Communications Server, contactez Sun de l'une des manières suivantes :

- [Listes de diffusion SailFin \(https://sailfin.dev.java.net/servlets/ProjectMailingListList\)](https://sailfin.dev.java.net/servlets/ProjectMailingListList) — Plusieurs listes de diffusion de la communauté SailFin pour divers intérêts et commentaires
- [Base de données de bogues sur Java Developer Connection \(http://developer.java.sun.com/servlet/SessionServlet?url=/developer/bugParade/index.jshtml\)](http://developer.java.sun.com/servlet/SessionServlet?url=/developer/bugParade/index.jshtml) — Pour afficher des bogues ou en soumettre en, utilisez Bug Parade de Java Developer Connection
- [Outil de suivi des problèmes SailFin \(https://sailfin.dev.java.net/servlets/ProjectIssues\)](https://sailfin.dev.java.net/servlets/ProjectIssues)

- **Forums sur la technologie Java** (<http://forum.java.sun.com/>) — Système de message interactif pour partager des informations et poser des questions sur les technologies et techniques de programmation Java.
- **Services de support technique des logiciels Sun** (<http://www.sun.com/service/sunone/software>) — Liens vers la base de connaissances, le centre de support en ligne et le service de téléchargement Product Tracker, ainsi que vers les programmes de maintenance et les coordonnées du support technique
- Le numéro de téléphone indiqué sur votre contrat de maintenance.  
Afin que nous puissions vous aider au mieux à résoudre vos problèmes, munissez-vous des informations suivantes lorsque vous contactez le support :
- Description du problème, notamment les conditions dans lesquelles le problème se produit et sa répercussion sur l'opération effectuée.
- Le type de machine, les versions du système d'exploitation et du produit, y compris les patches et autres logiciels pouvant avoir un lien avec le problème.
- Étapes détaillées des méthodes utilisées pour reproduire le problème.
- Journaux des erreurs ou core dumps éventuels.

## Vos commentaires sont les bienvenus

Dans le souci d'améliorer notre documentation, nous vous invitons à nous faire parvenir vos commentaires et vos suggestions.

Pour faire part de vos commentaires, rendez-vous sur <http://docs.sun.com> et cliquez sur Commentaires. Dans le formulaire en ligne, indiquez le titre et le numéro de référence du document. La référence est un numéro composé de sept ou neuf chiffres figurant sur la page de garde du manuel ou en haut du document. Par exemple, le titre de ce manuel est *Sun GlassFish Communications Server 2.0 Notes de version*, et son numéro de référence 820-4276.

## Ressources Sun supplémentaires

Vous pouvez obtenir des informations utiles sur les sites suivants:

- **Informations sur le produit Communications Server** ([http://www.sun.com/software/products/appsrvr/home\\_appsrvr.html](http://www.sun.com/software/products/appsrvr/home_appsrvr.html))
- **Services professionnels** (<http://www.sun.com/service/sunps/sunone>)
- **Produits et services logiciels** (<http://www.sun.com/software>)
- **Services de support logiciel** (<http://www.sun.com/service/sunone/software>)
- **Support et base de connaissances** (<http://www.sun.com/service/support/software>)
- **Services de support et de formation de Sun** (<http://training.sun.com>)
- **Services de conseil et services professionnels** (<http://www.sun.com/service/sunps/sunone>)
- **Informations à l'attention des développeurs** (<http://developers.sun.com>)
- **Services de support à l'attention des développeurs Sun** (<http://www.sun.com/developers/support>)

- Formation aux logiciels (<http://www.sun.com/software/training>)
- Fiches de présentation des logiciels Sun (<http://www.sun.com/software>)
- Documentation produit Sun Microsystems (<http://docs.sun.com/>)





# À propos de Sun GlassFish Communications Server 2.0

---

Communications Server est un serveur compatible avec la plate-forme Java EE 5, utilisé pour le développement et le déploiement d'applications Web, SIP et convergentes, et de services Web basés sur la technologie Java dans des environnements de production à grande échelle.

Cette section aborde les sujets suivants :

- “Fonctionnalités de Sun GlassFish Communications Server” à la page 9
- “Configurations matérielle et logicielle requises” à la page 12
- “Utilisation du centre de mise à jour” à la page 18

## Fonctionnalités de Sun GlassFish Communications Server

Communications Server prend en charge toutes les fonctionnalités offertes par Sun GlassFish Enterprise Server, mais intègre également nombre de nouvelles fonctionnalités, répertoriées ci-après. Pour obtenir la liste des fonctionnalités de Sun GlassFish Enterprise Server, reportez-vous aux notes de version de Sun GlassFish Enterprise Server.

- **Conforme aux spécifications JSR 289** — Communications Server contient une pile SIP conforme aux spécifications RFC 3261 3262 326, 3311 3515 390, etc. Il permet également d'exposer les API SIP Servlets Java définies par JSR 289.
- **Mise à niveau** — Communications Server prend en charge la mise à niveau à partir de Communications Server 1.5 vers Communications Server 2.0.
- **Sessions SIP et HTTP convergentes** — Communications Server prend en charge la fonctionnalité de sessions HTTP et SIP convergentes, telle que définie par les spécifications JSR 289.
- **Prise en charge du plug-in du DCR** — Communications Server prend en charge la configuration des règles centrées de données utilisant un plug-in sous forme de classe Java.
- **Protection contre les surcharges** — Communications Server prend en charge la protection contre les surcharges de la CPU et de la mémoire.
- **Réplication de session SIP** — Communications Server offre une fonctionnalité de basculement fiable des sessions SIP.

- **Mise à niveau progressive** — Communications Server prend en charge la mise à niveau d'une application ou du serveur avec un minimum de perte du service et des sessions.
- **Multi-home pris en charge** — Communications Server prend en charge la communication SIP entre les deux ou plusieurs sous-réseaux différents et fournit des connexions réseau redondantes au sein d'un même sous-réseau. Communications Server peut être utilisé dans les environnements qui comportent plusieurs cartes réseau.
- **Conteneur servlet SIP intégré** — Le conteneur servlet SIP offre les fonctionnalités suivantes :
  - fournit un point final réseau afin d'écouter les requêtes SIP ;
  - crée un environnement afin d'héberger et de gérer le cycle de vie des servlets SIP.

Le conteneur servlet SIP fonctionne avec le conteneur Web pour assurer la prise en charge des applications convergentes telles que définies par la spécification SIP Servlet API (SSA).

- **Administration simple** — Communications Server permet une administration aisée du conteneur SIP et des applications SIP déployées via l'interface de ligne de commande et l'interface graphique.
- **Sécurité** — L'authentification Digest est prise en charge pour les protocoles HTTP et SIP. P-Asserted-Identity prise en charge pour le protocole SIP. Le conteneur SIP prend en charge SIPs sur TLS.
- **Équilibreur de charge convergent** — L'équilibreur de charge convergent (CLB, Converged Load Balancer) est un composant logiciel d'équilibrage de charge qui facilite la haute disponibilité des applications convergentes en répartissant les requêtes des applications d'un cluster d'instances de serveur. Le CLB fournit des fonctionnalités suivantes :
  - transfert des requêtes SIP(S) et HTTP(S) à un cluster d'instances de serveur ;
  - requêtes d'équilibrage de charge via une stratégie de hachage/circulaire cohérente ;
  - activation et désactivation d'instances de serveur dans un cluster ;
  - contrôle de la maintenance des instances de serveur.
- **Prise en charge du déploiement d'applications SIP et convergentes** — Communications Server prend en charge le déploiement d'applications SIP pures et d'applications convergentes à l'aide du mécanisme de module d'extension de GlassFish. Il prend également en charge `sun-sip.xml` en tant que descripteur de déploiement spécifique à Sun pour les applications SIP.
- **Outils NetBeans** — Communications Server permet la prise en charge d'applications servlet SIP basées sur JSR 289 à partir de NetBeans 6.0. Un simple client SIP peut être créé à l'aide de l'agent d'essai pour tester les servlets SIP. Ce plug-in est disponible sous `as-install-dir/lib/tools`.
- **Déploiement d'Application Router** — Communications Server prend en charge le déploiement à chaud d'archives Application Router (fichiers JAR) en utilisant le mécanisme de module d'extension. Un routeur d'application qui suit une simple règle alphabétique pour l'acheminement des requêtes est fourni par défaut.

- **Format de fichier et conditionnement** — Communications Server est fourni sous la forme de fichiers exécutables.
- **Enregistrement sur Sun Connection** — vous pouvez utiliser le programme d'installation, l'IG de la console d'administration ou le centre de mise à jour pour enregistrer le produit sur [Sun Connection \(http://www.sun.com/service/sunconnection/index.jsp\)](http://www.sun.com/service/sunconnection/index.jsp). En enregistrant Communications Server sur Sun Connection, vous pouvez bénéficier des éléments suivants :
  - informations de patch et mises à jour de bogue ;
  - Screencasts et didacticiels
  - actualités et événements ;
  - support et offres de formation.
- **Prise en charge de SuSe Linux 64 bits**
- **Prise en charge de la plate-forme Java EE 5** : Sun GlassFish Communications Server implémente la spécification Java EE 5 pour offrir l'une des meilleures exécutions d'application possibles pour les applications et services Web de nouvelle génération d'une entreprise. Communications Server implémente les composants standard Java EE suivants :
  - 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

Vous trouverez par la suite la liste complète des technologies de la plate-forme Java EE 5.

- **Prise en charge de WSIT (Web Services Interoperability Technologies)** : Sun travaille étroitement avec Microsoft en vue d'assurer l'interopérabilité des technologies d'entreprise de services Web, telles que l'optimisation des messages, la fiabilité des services de messagerie et la sécurité. La version initiale de WSIT est le fruit de cette collaboration. WSIT est une implémentation de nombreuses spécifications de services Web ouverts visant à prendre en charge des fonctions d'entreprise. Outre l'optimisation des messages, la messagerie fiable et la sécurité, WSIT inclut une technologie d'initialisation et de configuration. Reprenant les composants XML de base actuellement intégrés à la plate-forme Java, WSIT utilise ou étend les fonctions existantes, et ajoute un nouveau support pour les services Web interopérables, notamment :
  - l'initialisation et la configuration ;
  - la technologie d'optimisation des messages ;
  - la technologie de messagerie fiable ;

- la technologie de sécurité.
- **Prise en charge de JBI** : JBI étend Java EE avec des interfaces de fournisseurs de services d'intégration d'entreprise (SPI). Ces SPI permettent aux développeurs de créer ou d'implémenter un environnement Java d'intégration à l'échelle de l'entreprise pour des spécifications telles que WSCI, BPEL4WS et W3C Choreography Working Group. Une implémentation JBI est directement installée par le programme d'installation d'Communications Server, généralement dans le répertoire *as-install/jbi*. Ce répertoire contient tous les fichiers JAR et composants système communs pour JBI, notamment un module *lifecycle* permettant de démarrer la structure JBI dans la JVM d'Communications Server.
- **Profils d'utilisation** : chaque domaine administratif est associé à un profil d'utilisation, permettant d'identifier les capacités de ce domaine. Communications Server inclut les profils suivants :
  - *Developer (Développeur)* : utilisez ce profil si vous exécutez votre domaine dans un environnement de développement et si vos applications ne nécessitent aucune fonctionnalité de clustering, telles que l'équilibrage de charge, la haute disponibilité et la réplication de session. Notez que le nom réel de ce profil est « *developer* » (sensible à la casse).
  - *Cluster* : utilisez ce profil si vous souhaitez créer des clusters d'instances de serveur d'application destinées à transmettre l'évolutivité et la haute disponibilité aux applications Java EE déployées. L'état de ces applications est *en mémoire* persistante. Notez que le nom réel de ce profil est « *cluster* » (sensible à la casse).
- **Open Source et communauté SailFin** — La communauté [SailFin \(http://sailfin.dev.java.net\)](http://sailfin.dev.java.net) vise à développer un serveur de communications gratuit, open source, de qualité commerciale permettant d'implémenter les dernières fonctionnalités de la plate-forme Java EE 5 ainsi que les technologies entreprise associées.
- **Prise en charge du centre de mise à jour** : le centre de mise à jour d'Communications Server fournit des mises à jour automatiques du produit et un accès rapide aux nouveaux composants.

## Configurations matérielle et logicielle requises

Cette section présente la configuration requise pour pouvoir installer Sun GlassFish Communications Server.

- “Plates-formes prises en charge” à la page 12
- “Informations importantes concernant les patches” à la page 14
- “Version de JDK” à la page 14

### Plates-formes prises en charge

Le tableau suivant répertorie les systèmes d'exploitation compatibles avec Sun GlassFish Communications Server.

TABLEAU 2-1 Systèmes d'exploitation pris en charge

Système d'exploitation	Mémoire minimum	Mémoire recommandée	Espace disque minimum	Espace disque recommandé	JVM	Type de déploiement
Sun Solaris 10 (x86) (32 bits et 64 bits)	512Mo	512Mo	250 Mo disponibles	500 Mo disponibles	J2SE 5.0 - 32 bits Java SE 6 - 32 bits et 64 bits	Production
Sun Solaris 10 (SPARC) (32 bits et 64 bits)	512Mo	512Mo	250 Mo disponibles	500 Mo disponibles	J2SE 5.0 - 32 bits Java SE 6 - 32 bits et 64 bits	Production
Red Hat Enterprise Linux 3.0 Update 1, 4.0 et 5.x (32 bits et 64 bits)	512Mo	1 Go	250 Mo disponibles	500 Mo disponibles	J2SE 5.0 - 32 bits Java SE 6 - 32 bits et 64 bits	Production
SUSE Linux Enterprise Server 10 (32 bits et 64 bits)  (SP1 et SP2 sont également pris en charge)	512Mo	1 Go	250 Mo disponibles	500 Mo disponibles	J2SE 5.0 - 32 bits Java SE 6 - 32 bits et 64 bits	Production
Windows XP  Windows Vista  Windows 2008	1 Go	2 Go	500 Mo disponibles	1 Go disponible	J2SE 5.0 - 32 bits Java SE 6 - 32 bits et 64 bits	Développement
Windows 7	1 Go	2 Go	500 Mo disponibles	1 Go disponible	J2SE 5.0 - 32 bits Java SE 6 - 32 bits et 64 bits	Développement

TABLEAU 2-1 Systèmes d'exploitation pris en charge (Suite)

Système d'exploitation	Mémoire minimum	Mémoire recommandée	Espace disque minimum	Espace disque recommandé	JVM	Type de déploiement
Mac OS 10.4, 10.5	512Mo	1 Go	250 Mo disponibles	500 Mo disponibles	J2SE 5.0 - 32 bits Java SE 6 - 32 bits	Développement
OpenSolaris 2009.06	512Mo	1 Go	250 Mo disponibles	500 Mo disponibles	J2SE 5.0 - 32 bits Java SE 6 - 32 bits et 64 bits	Développement

Sous UNIX™, vous pouvez vérifier la version du système d'exploitation en utilisant la commande `uname` et l'espace disque en utilisant la commande `df`.

**Remarque** – Nous vous recommandons d'utiliser le système de fichiers NTFS plutôt que FAT ou FAT32 pour exécuter Communications Server sur n'importe quelle plate-forme Windows.

### Informations importantes concernant les patches

#### Patches requis pour Solaris

Sun conseille aux utilisateurs de Solaris 10 (x86, SPARC) d'installer le groupe de patches recommandés. Ce cluster de patches est disponible sous "Patches recommandés et de sécurité" sur le site Web SunSolve (<http://sunsolve.sun.com/pub-cgi/show.pl?target=patchpage>).

#### Version de JDK

La version minimale (et certifiée) du JDK pour Communications Server est **1.6.0\_7**.

### ▼ Pour passer à la version Java SE prise en charge

Vous pouvez passer à une version Java SE prise en charge en modifiant le fichier `asenv` tel qu'indiqué ici.

#### 1 Si vous ne l'avez pas encore fait, installez la nouvelle version Java SE sur votre système.

Le kit Java SE SDK peut être téléchargé à l'adresse <http://java.sun.com/javase>

#### 2 Stoppez les domaines de Communications Server.

#### 3 Modifiez le fichier `as-install/config/asenv.conf` (`asenv.bat` sous Windows), en remplaçant la valeur `AS_JAVA` de sorte qu'elle pointe sur le nouveau répertoire de base de Java :

- 4 **Éditez le fichier `as-install/samples/common.properties`, en modifiant la ligne commençant par `com.sun.aas.javaRoot..` pour faire référence au nouveau répertoire de base Java.**

- 5 **Redémarrez Communications Server.**

### Version Ant

La version Ant intégrée à l'installation JAR SailFin est **1.6.5**. Pour créer une application Application Router personnalisée, vous avez besoin d'Ant **1.7**, que vous devrez télécharger séparément.

### Pilotes JDBC et bases de données

Tableau 2-2 répertorie les bases de données et les pilotes compatibles avec Java EE. Toutes les configurations prises en charge de Sun GlassFish Communications Server doivent au minimum comprendre une combinaison base de données/pilote de ce tableau, telle que la base de données Java DB intégrée et son pilote. En outre, Communications Server est conçu pour prendre en charge la connectivité JDBC à tout DBMS supplémentaire avec un pilote JDBC correspondant.

TABLEAU 2-2 Pilotes JDBC compatibles Java EE

Fournisseur de pilote JDBC	Type de pilote JDBC	Serveur de base de données pris en charge
Derby Network Client	Type 4	Derby 10.2
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

En règle générale, Communications Server prend en charge tous les pilotes JDBC répondant à la spécification Java EE.

### Versions de Message Queue

Pour les installations basées sur des fichiers, Message Queue 4.4 est intégré à Communications Server.

Le code imbriqué Sun GlassFish Message Queue fourni avec Communications Server est uniquement testé et certifié (en général) par rapport à la version équivalente du courtier Message Queue. Ceci signifie que l'utilisation du code Message Queue imbriqué fourni pour se connecter à un courtier Message Queue distant (non géré par Communications Server) exécutant une version différente du code Message Queue n'est pas prise en charge.

## Navigateurs

Cette section répertorie les navigateurs pris en charge par Sun GlassFish Communications Server 2.0.

TABLEAU 2-3 Navigateurs Web pris en charge

Navigateur	Version
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

## Autres exigences

Avant d'installer le logiciel Sun GlassFish Communications Server, vous devez également veiller à ce que les autres exigences ci-dessous soient satisfaites.

- **Espace disponible** : le répertoire temporaire doit disposer d'au moins 35 Mo disponibles pour l'installation de Sun GlassFish Communications Server.
- **Utilisation du programme de désinstallation** : si vous souhaitez supprimer Communications Server de votre système, veillez à utiliser le programme de désinstallation fourni avec le logiciel. Si vous utilisez une autre méthode, des problèmes peuvent se produire lors de la réinstallation de cette version ou de l'installation d'une nouvelle version.
- **Ports libres** : vous devez disposer de sept ports non utilisés. Le programme d'installation détecte automatiquement les ports utilisés et suggère des ports disponibles pour le domaine par défaut. Les ports par défaut initiaux sont :
  - 8080 pour HTTP, 8181 pour HTTPS ;
  - 3700 pour IIOP, 3820 pour IIOP/SSL et 3920 pour IIOP/SSL avec authentification mutuelle ;
  - 4848 (HTTP) pour le serveur d'administration ;
  - 8686 (RMI) pour les clients JMX.



- **Arrêt du pare-feu** (Microsoft Windows) : vous devez arrêter votre pare-feu avant d'installer le logiciel Sun GlassFish Communications Server. À défaut, tous les ports par défaut risquent d'être désactivés. Le programme d'installation doit être capable de déterminer, avec précision, les ports qui sont disponibles.

## API de la plate-forme Java EE 5

Sun GlassFish Communications Server 2.0 prend en charge la plate-forme Java EE 5. Le tableau suivant répertorie les API améliorées disponibles sur la plate-forme Java EE 5.

TABLEAU 2-4 Principales modifications des API sur la plate-forme Java EE 5

API	JSR
<b>Java EE 5</b>	
plate-forme Java, Enterprise Edition 5 ( <a href="http://java.sun.com/javae/5/docs/api/">http://java.sun.com/javae/5/docs/api/</a> )	JSR 244 ( <a href="http://jcp.org/aboutJava/communityprocess/pr/jsr244/">http://jcp.org/aboutJava/communityprocess/pr/jsr244/</a> )
<b>Technologies de services Web</b>	
Implémentation de Enterprise Web Services	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> )
Métadonnées de service Web pour la plate-forme Java	JSR 181 ( <a href="http://jcp.org/en/jsr/detail?id=181">http://jcp.org/en/jsr/detail?id=181</a> )
<b>Technologies de modèle de composant</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> )
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> )

TABLEAU 2-4 Principales modifications des API sur la plate-forme Java EE 5 (Suite)

API	JSR
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>Technologies de gestion</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> )
Déploiement de l'application J2EE ( <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> )
Contrat d'autorisation Java pour les conteneurs ( <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>Autres technologies Java EE</b>	
Annotations communes pour la plate-forme Java	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> )

## Utilisation du centre de mise à jour

Le centre de mise à jour fournit des mises à jour automatiques de Communications Server et un accès rapide aux nouveaux composants.

Lorsque celui-ci est activé, il effectue une mise à jour automatique du logiciel. Lors de ce processus, le centre de mise à jour collecte et transmet les données suivantes à Sun Microsystems (ou à son fournisseur de services) :

- ID d'installation unique (GUID) ;

- Adresse IP
- informations sur le système d'exploitation (nom, version, architecture, langue) ;
- version du JDK ;
- informations sur le téléchargement du module (nom du module, date, heure, état, temps de téléchargement, nombre d'octets téléchargés).

Aucune information personnelle identifiable ne fait l'objet d'un suivi. Aucune information personnelle identifiable n'est associée à d'autres données ou utilisée dans des rapports.

## ▼ Utilisation du centre de mise à jour

Pour garantir un accord explicite sur les mises à jour automatiques, le centre de mise à jour est, par défaut, désactivé. Pour permettre au centre de mise à jour d'effectuer des vérifications périodiques et des mises à jour automatiques :

### 1 Démarrez le centre de mise à jour.

- Sous Unix : `<installdir>/updatecenter/bin/updatetool`
- Sous Windows : `<installdir>\updatecenter\bin\updatetool.bat`

### 2 Sélectionnez l'onglet Préférences.

### 3 Dans la fenêtre de planification des mises à jour, modifiez la valeur Rechercher les mises à jour de Jamais (manuelle) sur la valeur désirée. Par exemple, choisissez une planification quotidienne ou hebdomadaire.

### 4 Spécifiez le jour et l'heure choisis pour la mise à jour.

### 5 Cliquez sur le bouton Enregistrer pour enregistrer vos modifications.

Le centre de mise à jour recherchera ainsi automatiquement les mises à jour des composants SailFin ou Communications Server suivant la planification spécifiée. Lorsqu'une mise à jour est disponible, le centre de mise à jour vous indiquera le composant prêt à être mis à jour.

## Informations supplémentaires sur l'intégration de WSIT

Pour obtenir des informations détaillées sur le statut WSIT, reportez-vous à la page [Notes de statut WSIT \(https://wsit.dev.java.net/source/browse/\\*checkout\\*/wsit/wsit/status-notes/status-notes-1-0-1-FCS.html\)](https://wsit.dev.java.net/source/browse/*checkout*/wsit/wsit/status-notes/status-notes-1-0-1-FCS.html).



# Problèmes connus et restrictions de Sun GlassFish Communications Server

---

Ce chapitre décrit les problèmes connus relatifs au logiciel Communications Server et les solutions associées. Si aucune plate-forme particulière n'est spécifiée, le problème s'applique à toutes les plates-formes. Voici les sujets abordés :

- “Administration de Communications Server” à la page 21
- “Équilibreur de charge convergent” à la page 23
- “Installation” à la page 24
- “Sécurité” à la page 25
- “Conteneur SIP” à la page 25
- “Réplication de session SIP” à la page 29

## Administration de Communications Server

**Communications Server ne détecte pas de conflit avec le port de pulsation d'un cluster (numéro de problème 1967).**

### Description

Lorsqu'un cluster est créé, Communications Server assigne de manière aléatoire un port de pulsation compris entre 1026 à 45556. Pour le cluster par défaut, à savoir le cluster créé par une installation de Communications Server, un nombre aléatoire compris entre 0 et 45556 est sélectionné. Le processus de création d'un cluster ne détermine pas avec précision si le port de pulsation est déjà utilisé par un autre service.

### Solution

Si la configuration automatique de la création d'un cluster sélectionne un port de pulsation qui est en conflit avec un autre service qui utilise déjà ce port, mettez à jour le port de pulsation du cluster vers un port non utilisé par le système.

Pour modifier le port de pulsation d'un cluster, utilisez la commande `asadmin` suivante :

```
asadmin set nom-cluster.heartbeat-port= nouveau-numéro-port
```

### **La création de domaine s'arrête sur un serveur NFS exécutant Linux 64 bits (numéro de problème 1961).**

#### **Description**

La commande `asadmin create-domain` risque d'échouer lors de la tentative de création d'un domaine sur un système de fichiers NFS (Network File System) avec le serveur NFS sous Linux 64 bits.

#### **Solution**

Aucune solution connue.

### **Utilisation élevée de la CPU lorsqu'il y a peu ou pas de trafic (Problème 1966)**

#### **Description**

Les instances de Communications Server affichent parfois une utilisation importante de la CPU même avec peu ou pas de trafic lorsque la protection de la CPU contre les surcharges est activé. Ce problème découle du bogue JDK 6693490 ([http://bugs.sun.com/bugdatabase/view\\_bug.do?bug\\_id=6693490](http://bugs.sun.com/bugdatabase/view_bug.do?bug_id=6693490)). Ce bogue est résolu dans JDK 6 Update 18.

#### **Solution**

Utilisez JDK 6 Update 18 avec Communications Server.

### **Les instances de Communications Server démarrent même si les ports SIP/SIPS ne sont pas liés (Problème 998)**

#### **Description**

Les instances de Communications Server démarrent même si elles ne peuvent pas être liées à un port SIPS ou SIP.

#### **Solution**

Assurez-vous que les ports sont libres avant de démarrer les instances de serveur. Consultez les fichiers journaux (`server.log`) pour vous assurer qu'il n'y a pas eu d'erreur ou d'exception au niveau du conteneur SIP au démarrage.

### **Communications Server n'utilise pas le JDK spécifié par l'option `—javahome` (Problème 789)**

#### **Description**

Vous pouvez utiliser un JDK préinstallé au lieu de la version par défaut pour l'installation via l'option `—javahome`. Par défaut, Communications Server utilise la version JDK disponible sous `as-install/jdk`.

### Solution

La variable `AS_JAVA` du fichier `asenv.conf` pointe toujours vers `as-install/jdk`. Si vous souhaitez utiliser une autre version de JDK, mettez à jour le fichier `asenv.conf` manuellement et modifiez la valeur de `AS_JAVA`.

### L'utilisation du tas Java 3,5 Go entraîne le redémarrage des instances alors que du trafic est en cours (Problème 1169)

#### Description

Lorsque la taille du tas JVM est définie sur 3,5 Go, les instances de Communications Server ne fonctionnent pas et redémarrent lorsqu'elles reçoivent du trafic.

#### Solution

Assurez-vous que la taille maximale du tas JVM est définie sur 3,0 Go ou moins.

### Communications Server indique l'utilisation de la CPU de manière incorrecte lorsque vous utilisez uniquement l'un des cœurs du système multi-cœurs (Problème 1344)

#### Description

Sur les plates-formes Solaris, Communications Server calcule l'utilisation de la CPU en fonction du nombre de processeurs disponibles et de l'utilisation de la CPU par cœur. Cependant, Communications Server prend en compte la valeur statique du nombre de cœurs et non pas le nombre de cœurs utilisés par le tas JVM.

#### Solution

Recalculez les valeurs de seuil de la CPU si vous n'utilisez pas tous les cœurs de la machine.

## Équilibreur de charge convergent

### Messages GRAVES dans les journaux en raison de la reconfiguration dynamique de l'équilibreur de charge convergent après le déploiement de l'application (Problème 1161)

#### Description

Si vous modifiez la configuration de l'équilibreur de charge convergent sur une cible et redéployez les applications sur cette cible, les journaux d'instance vous afficheront des messages GRAVES.

#### Solution

Ces messages n'ont pas d'incidence sur le fonctionnement de l'équilibreur de charge convergent ou sur les instances. Ne tenez pas compte de ces messages.

### **Lorsque l'URI entier est utilisé, le paramètre BEKey de l'en-tête Contact n'est pas échappé correctement (Problème 1466)**

#### **Description**

Lorsque vous utilisez un équilibreur de charge convergent avec une règle centrée de données qui renvoie un URI complet pour le paramètre BEKey, le paramètre BEKey dans l'en-tête Contact n'est pas échappé correctement. Le caractère « : » n'est pas correctement échappé comme indiqué dans la spécification RFC 3261.

#### **Solution**

Aucune solution connue.

## **Installation**

### **Le programme d'installation basé sur des fichiers de Communications Server n'installe pas l'exemple d'application Basic3pcc (Bogue 6894932)**

#### **Description**

Le programme d'installation basé sur des fichiers de Communications Server n'installe pas l'exemple d'application Basic3pcc. Cette application est disponible avec le programme d'installation JAR.

#### **Solution**

Aucune solution connue.

### **Le programme d'installation de Communications Server provoque une panne système sous Linux (6739013)**

#### **Description**

Ce problème a été observé sur des systèmes exécutés sous Linux avec la variable d'environnement, `MALLOC_CHECK_`, défini à 2.

#### **Solution**

Définissez la variable d'environnement, `MALLOC_CHECK_` à 0. Exécutez l'une des commandes suivantes :

- Pour le bourne shell :

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

- Pour le bash shell:

```
export MALLOC_CHECK_=0
```



- Pour le csh, tcsh shell :

```
setenv MALLOC_CHECK_ 0
```

### Échec de l'installation avec le JDK 64 bits (6796171)

#### Description

Échec de l'installation sur les systèmes 64 bits qui disposent du JDK 64 bits, car le programme d'installation tente d'utiliser le JDK 64 bits.

#### Solution

Si vous installez Sun GlassFish Communications Server sur un système 64 bits, téléchargez le JDK 32 bits et utilisez-le pour installer Sun GlassFish Communications Server sur votre machine 64 bits. Vous devez utiliser la commande suivante : `. /nom-fichier-distribution -javahome chemin d'accès à l'emplacement du JDK 32 bits`

Après l'installation, pour vous assurer que Sun GlassFish Communications Server utilise un JDK 64 bits, modifiez la valeur de la variable `AS_JAVA` dans le fichier `asenv.conf` pour qu'elle pointe vers l'installation du JDK 64 bits.

## Sécurité

### Communications Server renvoie une exception lorsque la propriété `trust-auth-realm-ref` n'est pas spécifiée dans `sun-sip.xml` (CR 6786131)

#### Description

Communications Server renvoie une exception `Pointeur nul`, domaine non configuré lorsque l'authentification `P-Asserted-Identity` est configurée dans `sun-sip.xml`.

#### Solution

Configurez le domaine à l'aide de la propriété `trust-auth-realm-ref` dans `sun-sip.xml`.

## Conteneur SIP

### Le conteneur SIP ne parvient pas à traiter une requête d'annulation lorsqu'il a renvoyé une réponse 100 (Problème 712)

#### Description

Le conteneur SIP ne peut pas traiter une requête d'annulation lorsqu'une réponse 100 a été envoyée.

#### Solution

L'application doit envoyer une réponse provisoire (telle que `1xx`), de manière à ce que la partie distante puisse annuler la requête d'invitation.

## Les sessions SIP et les sessions HTTP n'appliquent pas le même modèle de délai d'expiration de session (Problème 1180)

### Description

Le modèle d'expiration de session des sessions SIP est différent de la logique de délai d'expiration HTTP. Dans le modèle HTTP, la session est automatiquement étendue, hors contrôle de l'application, chaque fois qu'une nouvelle requête HTTP est reçue dans cette session HTTP.

Avec les sessions SIP, l'application contrôle la durée de la session `SipApplicationSession` (SAS), sujette à l'approbation du conteneur SIP. Les applications peuvent utiliser la méthode `setExpires` pour indiquer que cette SAS devrait expirer. `setExpires` définit un délai d'expiration par rapport au moment où la méthode `setExpires` est appelée. Le conteneur peut modifier, rejeter ou accepter la durée indiquée dans `setExpires`. Si la session n'est pas invalidée, alors le rappel `sessionExpired` est réalisé au moment défini par `setExpires`. Dans ce rappel, l'application peut tenter d'étendre la durée de la SAS en appelant un nouveau `setExpires`, également sujet à modification, rejet ou acceptation par le conteneur.

Pour cette raison, les applications convergentes qui commencent avec la même heure d'expiration que la session `SipApplicationSession` (SAS) et la session HTTP noteront que la session SAS expire avant la session HTTP si de nouvelles requêtes ont été reçues sur la session HTTP.

### Solution

La meilleure manière de gérer les différents délais d'expiration des sessions SIP et HTTP est de commencer avec un délai d'expiration SAS suffisamment long, qui représente la durée totale prévue de la session d'application (avec plusieurs requêtes HTTP). La durée de vie d'une session SAS pourrait même être réglée sur l'infini, en particulier si la sémantique `invalidateWhenReady` est utilisée, auquel cas la session `SipApplicationSession` est invalidée lorsque la dernière session enfant du protocole est invalidée. Le premier délai d'expiration pour le SAS peut être configuré dans le descripteur de déploiement.

Si la durée totale maximale peut être estimée à l'avance, aucun autre code est nécessaire, puisqu'il convient donc d'invalider la session SIP et la session HTTP lorsque le SAS expire. Si la durée maximale ne peut être estimée à l'avance, la session `SipApplicationSession` peut être étendue lorsqu'elle arrive à expiration, comme illustré dans le fragment de code ci-dessous.

Dans l'implémentation `SipApplicationSessionListener`, vous pouvez effectuer une action, comme suit :

```
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
}

```

### La session SIP se poursuit après le rappel du conteneur pour sessionExpired (Problème 1265)

#### Description

Il s'agit d'un problème intermittent. Le conteneur SIP répond par intermittence avec un message d'erreur interne 500 Serveur au lieu d'une erreur 481 Appel/Transaction inexistant lorsqu'il y a une condition de compétitivité entre l'erreur 200 de notification indiquant que la session a été supprimée et la requête d'inscription envoyée par le client lorsqu'il reçoit cette notification.

#### Solution

Le client doit actualiser la requête d'inscription avant que l'abonnement expire.

### Communications Server agit d'abord comme un UAS, puis en tant que proxy, puis génère un NOP (Problème 1432)

#### Description

Lorsqu'il reçoit une requête d'invitation, Communications Server agit d'abord comme un UAS, répond à cette requête avec 1XX, puis transmet cette demande d'invitation à une autre instance, qui envoie une réponse 200 OK. Le 1xx crée une branche virtuelle interne tandis que le message 200 crée une branche réelle. Une fois le message 200 OK de B reçu, la branche virtuelle interne doit être annulée.

#### Solution

Cette trace d'exception n'affecte pas la fonctionnalité de la branche de proxy virtuelle.

### La méthode getLastAccessedTime ne permet pas de fournir des résultats précis (Problème 1351)

#### Description

La méthode getLastAccessedTime d'une session SIP ne permet pas de fournir des résultats précis.

#### Solution

Les applications nécessitant de conserver une trace précise de lastAccessedTime doivent être stockées dans SipApplicationSession.

```

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());
}
```

### **Le listener SIP reste actif un certain temps après sa suppression (Problème 1294)**

#### **Description**

Après la suppression d'un listener SIP configuré pour des requêtes TCP et UDP, il reste actif pendant quelques instants. Les requêtes UDP envoyées au listener peuvent recevoir une réponse du listener.

#### **Solution**

Aucune solution connue. Le listener SIP arrête l'écoute des requêtes UDP après un certain temps. Ce problème n'a aucun impact sur les requêtes TCP.

### **Communications Server génère une exception lors de la réception d'un en-tête Contact sans « <> » (Problème 1489)**

#### **Description**

Communications Server génère une exception lors de la réception d'un en-tête Contact sans « <> ». Conformément à la spécification SIP RFC 3261, il n'est pas obligatoire d'indiquer « <> » dans l'adresse. Cela peut entraîner des problèmes d'interopérabilité avec d'autres périphériques non compatibles SIP.

#### **Solution**

Utilisez « <> » dans l'en-tête Contact.

### **Communications Server génère une exception sur une valeur UUID incorrecte (Problème 1494)**

#### **Description**

Communications Server génère une exception sur une valeur UUID incorrecte au lieu de renvoyer une erreur 400, Requête erronée. La valeur UUID réside dans la valeur `sip.instance` de l'en-tête du contact SIP.

#### **Solution**

Aucune solution connue.

### **Windows : parfois Communications Server ne reçoit pas les messages UDP (pas d'ID)**

#### **Description**

Ce problème n'est rencontré que par intermittence et sous Windows. Communications Server ne reçoit pas les messages UDP.

**Solution**

Définissez l'option JVM suivante comme suit et redémarrez Communications Server.

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

**Réplication de session SIP**

**Interblocage possible si une application convergente utilise un objet SAS comme verrou de synchronisation (Problème 1954)**

**Description**

Si une application convergente dotée de servlets HTTP et SIP utilise un objet `sipApplicationSession` comme verrou pour synchroniser l'accès entre les flux de travail SIP et HTTP, un interblocage est observé.

**Solution**

N'utilisez pas `sipApplicationSession` comme verrou de synchronisation. Utilisez plutôt un objet sérialisable défini comme un attribut dans `sipApplicationSession` comme verrou.



# Problèmes connus et restrictions de Sun GlassFish Enterprise Server

---

Ce chapitre décrit les problèmes connus relatifs au logiciel Sun GlassFish Enterprise Server 2.1.1 et les solutions associées. Ces problèmes affectent aussi spécifiquement Communications Server, qui a été créé à partir de Sun GlassFish Enterprise Server 2.1. Si le récapitulatif ne mentionne aucune plate-forme particulière, cela signifie que le problème s'applique à toutes les plates-formes. Voici les sujets abordés :

- “Administration” à la page 31
- “Client d'application” à la page 38
- “Pilotes Sun JDBC intégrés” à la page 39
- “Déploiement” à la page 41
- “Documentation” à la page 41
- “EJB” à la page 42
- “Didacticiel de Java EE” à la page 42
- “Java Persistence” à la page 42
- “Gestion du cycle de vie” à la page 43
- “Enregistrement” à la page 45
- “Message Queue” à la page 45
- “Contrôle” à la page 47
- “Sécurité” à la page 48
- “Conteneur Web” à la page 49

## Administration

Cette section traite des problèmes connus liés à l'administration et les solutions associées.

**Communications Server ne détecte pas de conflit avec le port de pulsation d'un cluster (numéro de problème 1967).**

### Description

Lorsqu'un cluster est créé, Communications Server assigne de manière aléatoire un port de pulsation compris entre 1026 à 45556. Pour le cluster par défaut, à savoir le cluster créé par une

installation de Communications Server, un nombre aléatoire compris entre 0 et 45556 est sélectionné. Le processus de création d'un cluster ne détermine pas avec précision si le port de pulsation est déjà utilisé par un autre service.

### **Solution**

Si la configuration automatique de la création d'un cluster sélectionne un port de pulsation qui est en conflit avec un autre service qui utilise déjà ce port, mettez à jour le port de pulsation du cluster vers un port non utilisé par le système.

Pour modifier le port de pulsation d'un cluster, utilisez la commande `asadmin` suivante :

```
asadmin set nom-cluster.heartbeat-port= nouveau-numéro-port
```

### **La création de domaine s'arrête sur un serveur NFS exécutant Linux 64 bits (numéro de problème 1961).**

#### **Description**

La commande `asadmin create-domain` risque d'échouer lors de la tentative de création d'un domaine sur un système de fichiers NFS (Network File System) avec le serveur NFS sous Linux 64 bits.

#### **Solution**

Aucune solution connue.

### **Une dégradation des performances est observée lorsqu'un fichier journal volumineux fait l'objet d'une rotation (6718611)**

#### **Description**

Lorsqu'un fichier journal volumineux fait l'objet d'une rotation, le temps de réponse augmente légèrement.

#### **Solution**

Il est possible de diminuer la perte de performances en modifiant les valeurs des paramètres Limite de rotation du fichier et Temps limite de rotation du fichier dans le Journal. Les valeurs de ces propriétés dépendent de votre application et de votre environnement.

### **Échec du déploiement d'adaptateur de ressources RA contre IBM MQ (problème 6605)**

#### **Description**

Déploiement d'un adaptateur RA générique contre les défaillances produit d'IBM Message Queue. Les autorisations accordées dans le fichier `server.policy` sont les suivantes.



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

## Solution

Modifiez les autorisations du fichier `server.policy` comme suit :

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

## Des instances autonomes obtiennent parfois des fichiers de la part d'autres instances (6698604)

### Description

Dans certains cas, les fichiers installés sur le DAS devant être synchronisés avec une instance spécifique sont en fait envoyés à d'autres instances.

### Solution

Aucune solution connue.

## Le message de démarrage de la commande `start-cluster` est trop détaillé (6728317)

### Description

La commande `asadmin start-cluster` indique trop de messages même en cas de défaillance de composants non essentiels au démarrage. Consultez l'exemple suivant de sortie de commandes en cas de défaillance d'éléments non essentiels (par rapport aux instances du cluster) :

```
./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.

## Solution

Aucune solution connue. Ces messages (exceptions) peuvent être ignorés.

## Le script package-appclient ne fonctionne pas si le fichier domain1 est manquant (6171458)

### Description

Par défaut, *as-install/lib/package-appclient.xml* comporte une valeur codée en dur pour la variable `AS_ACC_CONFIG` de `domain1`, pointée par `asenv.conf`. Si `domain1` est supprimé et qu'un autre domaine est créé, la variable `AS_ACC_CONFIG` n'est pas mise à jour avec le nouveau nom de domaine, ce qui provoque l'échec du script `package-appclient`.

## Solution

Effectuez l'une des opérations suivantes :

1. Laissez `domain1` intact et créez vos propres domaines en tenant compte de celui-ci.
2. Supprimez `domain1` et remplacez la valeur codée en dur de `domain1` dans `as-install/lib/package-applient.xml` par le nouveau nom de domaine.

Cette procédure devra être effectuée à chaque création d'un nouveau domaine si `domain1` n'est pas présent.

## Le démarrage d'Application Server avec un agent JMX supplémentaire n'est pas pris en charge (6200011)

### Description

La version J2SE1.4.x, 5.0 ou version ultérieure peut être configurée sur Application Server. La fonction de démarrage d'un agent JMX est intégrée à la plate-forme J2SE 5.0. Un agent est activé lorsque vous définissez explicitement les propriétés système lors du démarrage du serveur.

Voici quelques exemples de valeurs:

```
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"
```

Une fois les propriétés JMX configurées et le serveur démarré, un nouveau serveur `jmx-connector` est démarré dans la machine virtuelle d'Application Server. L'un des effets secondaires non désirés de cette opération est que celle-ci nuit aux fonctions d'administration ; la console d'administration d'Application Server et l'interface de ligne de commande peuvent alors produire des résultats inattendus. Le problème provient du fait qu'il existe des conflits entre le serveur `jmx connector` intégré et le nouveau serveur `jmx-connector`.

### Solution

Si vous utilisez `jconsole` (ou tout autre client JMX-compliant), pensez à réutiliser le serveur de connecteurs JMX standard lancé au démarrage d'Application Server.

Lorsque le serveur démarre, une ligne similaire à celle indiquée ci-dessous s'affiche dans le journal du serveur. Vous pouvez vous connecter à l'URL `JMXService` spécifié et exécutez les mêmes opérations de configuration/gestion après avoir fourni des données d'authentification correctes ; par exemple :

```
[#|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. [#]

### **Le fichier .asadmintruststore n'est pas décrit dans la documentation (6315957)**

#### **Description**

Le fichier .asadmintruststore n'est pas présenté dans la documentation. Si ce fichier n'existe pas dans le répertoire home de l'administrateur, vous pouvez rencontrer de graves bogues lors de la mise à niveau d'applications hébergées sur le serveur.

#### **Solution**

- Si possible, la commande `asadmin start-domain domain1` doit être exécutée par l'utilisateur qui a installé le serveur.
- Dans le cas contraire, le fichier .asadmintruststore doit être déplacé ou copié du répertoire home de l'utilisateur qui a procédé à l'installation dans le répertoire home de l'utilisateur qui l'exécute.
- Notez que si le fichier est déplacé (et non copié) du répertoire home de l'utilisateur "installateur" dans le répertoire home de l'utilisateur "exécuteur", vous pouvez rencontrer des problèmes de mise à niveau de l'application, tels que décrits dans les bogues 6309079, 6310428 et 6312869 car l'utilisateur de mise à niveau/installation (généralement root dans Java ES) ne disposera plus du fichier .asadmintruststore dans son répertoire home.

### **Échec du démarrage des instances clusterisées suite à l'expiration du délai d'accès au courtier JMS (6523663)**

#### **Description**

Le mode d'intégration MQ par défaut d'une instance clusterisée d'Communications Server est LOCAL. Lorsqu'Communications Server est installé dans un emplacement (PATH) long (c'est-à-dire « non court »), `imqbrokersvc.exe` s'arrête brutalement au démarrage de l'instance clusterisée. Le problème provient d'une allocation de mémoire incorrecte dans `imqbrokersvc`.

#### **Solution**

Il est nécessaire de modifier le type de service JMS pour l'instance clusterisée, de la valeur par défaut LOCAL sur REMOTE. Avec cette configuration, toutes les instances pointent vers le courtier DAS. Suivez les instructions ci-dessous pour configurer un cluster en mode REMOTE.

---

**Remarque** – Avec le mode REMOTE, toutes les instances n'utilisent qu'un seul courtier (DAS), par conséquent aucun cluster de courtier n'est créé au démarrage du cluster d'Communications Server. Consultez "Clustering automatique" à la section 4.1, division iii de la page <http://www.glassfishwiki.org/gfwiki/attach/OnePagersOrFunctionalSpecs/as-mq-integration-gfv2.txt> pour obtenir plus d'informations. La fonctionnalité susmentionnée ne sera pas disponible.

---

## ▼ Utilisation de la ligne de commande

### Avant de commencer

Modifiez le port et le fichier de mots de passe selon votre environnement. Notez que dans les instructions ci-dessous, le nom du cluster est racluster, le port d'administration DAS est 5858 et le port JMS DAS 7676 .

#### 1 Modifiez la configuration du cluster, en changeant le type JMS sur REMOTE .

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

#### 2 Créez un hôte JMS correspondant à l'hôte JMS DAS.

```
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 Définissez l'hôte JMS DAS créé à l'étape précédente comme le nouvel hôte JMS par défaut.

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

## ▼ Utilisation de l'IG d'administration

#### 1 Accédez à Configurations->cluster-name-config->Java Message Service->Hôtes JMS.

#### 2 Cliquez sur *Nouveau* pour créer un nouvel hôte JMS ; nommez-le dashost.

#### 3 Entrez les paramètres de configuration correspondant au service JMS pour le DAS ; les valeurs par défaut sont les suivantes :

- Nom d'hôte : hôte local
- Port : 7676
- Utilisateur administrateur : admin
- Password: admin

Modifiez ces paramètres au besoin pour votre service JMS DAS.

#### 4 Retournez à l'onglet Java Message Service, puis modifiez le type de service JMS sur REMOTE (défini sur LOCAL par défaut).

- 5 Choisissez `dashost` à partir de la liste déroulante `default-jms-host`.
- 6 Enregistrez vos modifications, puis démarrez votre agent de nœud ou votre cluster.

### Impossible d'afficher le diagramme jmaki sous les navigateurs Netscape 8.1.3, Mozilla 1.7 et Safari 2.0.4 (6543014)

#### Description

Lorsque vous essayez d'afficher un diagramme de la page de contrôle des statistiques de journalisation à l'aide d'un navigateur non pris en charge, le message d'erreur suivant peut s'afficher :

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

#### Solution

Utilisez un navigateur pris en charge. Reportez-vous à la section “[Navigateurs](#)” à la page 16 pour obtenir la liste des navigateurs pris en charge par Communications Server.

### La commande `create-domain` échoue avec le mot de passe principal personnalisé sous AIX (6628170)

#### Description

Sous le système d'exploitation AIX, toute tentative de création de domaine avec un mot de passe principal personnalisé échoue avec l'erreur suivante :

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

## Client d'application

Cette section décrit les problèmes connus des clients d'application et les solutions associées.

### Le JAR de bibliothèque contenu dans l'archive de clients d'application écrase le fichier MANIFEST (6193556)

#### Description

Si vous possédez un fichier JAR de niveau supérieur dans votre JAR client (dans notre cas, `reporter.jar`), le fichier manifeste de ce JAR écrase celui du JAR client lorsque vous déployez ce dernier.

#### Solution

Aucune pour l'instant.

**Le CCA essaie toujours de se connecter à localhost:3700 (6527987)****Description**

Le client d'application essaie toujours de se connecter à localhost:3700. Le problème vient du fait que plusieurs propriétés système doivent être lues avant que le code client ne soit invoqué.

**Solution**

Définissez les éléments suivants en tant que propriétés système (-D dans votre JAVA\_CMD). Ne les définissez *pas* dans votre code client d'application :

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

ASQuickStartup **rompt** SocketChannel.keyFor(), **retournant une valeur null au lieu de** SelectionKey (**Issue Tracker 3027**)

**Description**

Lorsqu'un SocketChannel est enregistré sur plusieurs sélecteurs, socketChannel.keyFor(lastRegisteredSelector) retourne une valeur null au lieu de SelectionKey.

**Solution**

Ce problème est associé au bogue JDK 6562829 et devrait être résolu dans 6.0 U3. Une solution a été incluse dans Communications Server pour que le sélecteur soit déployé avant que l'API keyFor ne soit appelée. Cela permet à keyFor de fonctionner jusqu'à ce que le bogue JDK soit résolu.

**Pilotes Sun JDBC intégrés**

Cette section décrit les problèmes connus du pilote Sun JDBC intégré et les solutions associées.

**Erreurs PreparedStatement (6170432)****Description 1**

Si une application génère plus de 3000 objets PreparedStatement dans une transaction, l'erreur suivante peut se produire avec DB2 :

```
[sunm][DB2 JDBC Driver] Plus d'instruction disponible. Veuillez recréer votre package avec une valeur dynamicSections supérieure.
```

**Solution 1**

Ajoutez les propriétés suivantes à la définition de pool de connexions afin que le pilote puisse rééditer les liens des packages DB2 avec une valeur dynamicSections supérieure :

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

## Description 2

En liaison avec l'erreur `PreparedStatement` mentionnée ci-dessus, le message d'erreur suivant peut également être généré :

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

## Solution 2

Augmentez la valeur du paramètre de configuration `APPLHEAPSZ` pour le serveur DB2. 4096 constitue une valeur correcte.

## Description 3

Niveau d'isolement `TRANSACTION_SERIALIZABLE`. Si votre application utilise le niveau d'isolement `TRANSACTION_SERIALIZABLE` avec l'un des paramètres indiqués ci-dessus, elle peut rester bloquée en tentant d'obtenir la connexion.

## Solution 3

Pour définir le niveau d'isolement d'une connexion, le pool de connexions correspondant doit être créé sur le même niveau d'isolement.

## Java DB n'est pas démarrée après le redémarrage de la machine ou le démarrage d'Application Server (6515124)

### Description

La base de données Java DB intégrée n'est pas automatiquement redémarrée après le redémarrage d'un système hôte ou d'une zone Solaris, ou le démarrage d'Communications Server. Il ne s'agit pas d'un bogue mais du comportement attendu pour toute application tiers ou intégrée. Le problème réside dans le fait que Java DB doit être démarrée avant l'instance d'Communications Server.

### Solution

Après le redémarrage de la machine hôte ou de la zone Solaris, assurez-vous de démarrer Java DB *avant* Communications Server ; par exemple :

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



## Déploiement

### Le déploiement automatique échoue parfois sur un cluster (6610527)

#### Description

Les problèmes de temporisation peuvent provoquer l'échec du déploiement automatique dans les domaines configurés pour prendre les clusters en charge. Ce problème ne survient pas dans les domaines ne prenant pas les clusters en charge.

#### Solution

Utilisez l'une des solutions suivantes :

- Utilisez le déploiement automatique comme suit :
  - Déployez automatiquement les applications individuelles de manière séquentielle.
  - Définissez un délai entre les divers déploiements automatiques des applications individuelles.
- Déployez les applications manuellement à l'aide de l'IG de la console d'administration ou à l'aide de la ligne de commande.

### Classloader spécifique à l'application non utilisé par la compilation JSP (6693246)

#### Description

Le classloader spécifique à l'application (applibs ou --libraries) n'est pas utilisé par la compilation JSP. En conséquence, les JSP associés à ces JAR ne seront pas compilés.

#### Solution

Aucune solution connue.

## Documentation

Cette section décrit les problèmes détectés dans la documentation et les solutions associées.

### Incohérences Javadoc (plusieurs ID)

Une documentation Javadoc est absente ou incorrecte pour plusieurs interfaces et méthodes AMX :

- Les méthodes liées au mécanisme d'obtention des statistiques `NumConnAcquired` et `NumConnReleased` ne figurent pas dans `ConnectorConnectionPoolStats` et `AltJDBCConnectionPoolStats`. Ces méthodes vont être ajoutées dans une version ultérieure en tant que `getNumConnAcquired()` et `getNumConnReleased()`.

- L'appel des méthodes suivantes dans `EJBCacheStats` renvoie une exception : `getPassivationSuccesses()`, `getExpiredSessionsRemoved()`, `getPassivationErrors()`, `getPassivations()`. Ce problème sera résolu dans une version ultérieure.
- Une fois le serveur démarré, les MBeans AMX nécessitent plusieurs secondes avant d'être tous enregistrés et disponibles. Il vous sera bientôt possible, dans une version ultérieure, de déterminer le moment où les MBeans AMX seront complètement chargés.
- La constante `XTypes.CONNECTOR_CONNECTION_POOL_MONITOR` est mal orthographiée ("NNN"). Ce problème sera corrigé dans une version ultérieure.

#### **L'outil ANT intégré génère l'exception `java.lang.NoClassDefFoundError` (6265624)**

##### **Description**

L'exception suivante est générée dans le thread principal "main"

```
java.lang.NoClassDefFoundError: org/apache/tools/ant/launch/Launcher.
```

##### **Solution**

Il est conseillé de ne pas utiliser l'outil ANT intégré ailleurs que dans Communications Server.

## **EJB**

#### **Resource Injection ne fonctionne pas avec HandlerChain (6750245)**

##### **Description**

Resource Injection ne fonctionne pas avec HandlerChain à cause de l'ordre d'initialisation d'EJB.

##### **Solution**

Aucune solution connue.

## **Didacticiel de Java EE**

Lorsque vous créez des ressources à l'aide de la console d'administration, utilisez l'onglet Cibles pour indiquer que le serveur est la cible. Si vous utilisez la ligne de commande ou une cible asant, le serveur représente la cible par défaut et aucune autre action n'est requise.

## **Java Persistence**

#### **TopLink suppose que la propriété/le champ `myCollection` est clonable (Issue Tracker 556)**

##### **Description**

Si l'API `java.util.Arrays.asList()` est utilisée pour convertir un `Object[]` en une `Collection`, JDK retourne une implémentation de `java.util.ArrayList` non clonable.

L'exception suivante est alors générée :

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

Ce problème fait l'objet d'un suivi sur [https://glassfish.dev.java.net/issues/show\\_bug.cgi?id=556](https://glassfish.dev.java.net/issues/show_bug.cgi?id=556).

### Solution

Créez une autre collection à l'aide de son constructeur ; par exemple :

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

### GenerationType.IDENTITY et DataDirect avec SyBase (Issue Tracker 2431)

#### Description

La tentative d'insertion d'une entité utilisant GenerationType.IDENTITY échoue lorsque le pilote DataDirect est utilisé avec SyBase. Cette tentative échoue car le pilote DataDirect crée une procédure stockée pour chaque instruction préparée et paramétrée.

#### Solution

Dans le fichier domain.xml, définissez la propriété PrepareMethod=direct sur la source de données correspondante.

## Gestion du cycle de vie

Cette section décrit les problèmes connus de gestion du cycle de vie et les solutions associées.

### La définition de la propriété ejb-timer-service entraîne l'échec de la commande set (6193449)

#### Description

Après avoir défini minimum-delivery-interval de la propriété ejb-timer-service sur 9000, la tentative de définition de redelivery-interval-in-millis de la propriété ejb-timer-service sur 7000 entraîne l'échec de la commande set avec l'erreur suivante :

```
[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` correspond à l'intervalle de temps minimal entre chaque distribution d'une même horloge.
- `redelivery-interval-in-millis` indique le délai pendant lequel le service d'horloge attend avant d'effectuer une nouvelle tentative de distribution suite à l'expiration de la valeur `ejbTimeout`.

La relation entre la propriété de l'intervalle de redistribution et celle de l'intervalle de livraison minimal n'étant pas logique, il vous est impossible d'utiliser l'interface graphique (IG) ou l'interface de ligne de commande (CLI) pour définir un intervalle de livraison minimal supérieur à celui de redistribution.

`minimum-delivery-interval-in-millis` doit toujours être égal ou supérieur à `redelivery-interval-in-millis` pour la propriété `ejb-timer-service`. Cependant, un contrôle de validité erroné s'effectue dans Application Server pour vérifier que la valeur de `redelivery-interval-in-millis` est supérieure à la valeur de `minimum-delivery-interval-in-millis`.

### Solution

Utilisez les valeurs par défaut suivantes :

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

Toute autre valeur provoquera une erreur.

### Erreur générée lors de l'affichage des destinations physiques JMS dans la configuration non DAS (6532532)

#### Description

Si vous essayez d'afficher les destinations physiques JMS à l'aide de `default-config`, un message d'erreur apparaîtra.

#### Solution

Ce comportement est normal. Dans Communications Server 2.0, `default-config` est un modèle d'informations de configuration, par conséquent, les opérations JMS (telles que `list` et `create`) ne peuvent pas être exécutées pour le fichier `default-config`. En revanche, ces opérations peuvent être exécutées pour la configuration de vos instances autonomes ou clusterisées.

## Win2003 uniquement : fuite de mémoire du pool non paginé, arrêt de la pile TCP et test richaccess (6575349)

### Description

(*Windows 2003 uniquement*) Des fuites de mémoire se produisent sur les systèmes Windows 2003 lors de l'exécution de fonctions richaccess. Ce problème est dû à la croissance continue du pool non paginé Win32, qui peut provoquer une panne de la pile TCP/IP complète. Une fois l'erreur générée, la pile TCP/IP demeure dans un état récupérable, et la seule manière de la restaurer est de redémarrer le système Windows 2003.

### Solution

Il existe deux solutions à ce problème.

- Utilisez le mode de blocage Grizzly en configurant l'attribut `http-listener` de `domain.xml`, `blocking-enabled="true"` ou ajoutez la propriété `http-listener` suivante :  

```
<property name="blocking" value="true"/>
```
- Utilisez Windows Vista ou Windows XP.

## Enregistrement

Cette section décrit les problèmes connus de consignation et les solutions.

### La définition de l'instruction de débogage `access, failure` interrompt le démarrage d'Application Server (6180095)

#### Description

Le paramétrage de l'option `java.security.debug` pour JVM entraîne un blocage du démarrage de l'instance du serveur. Ce problème se produit, par exemple, lorsque vous définissez les paramètres ci-dessous dans le fichier `domain.xml`.

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

#### Solution

Aucune pour l'instant. Évitez de paramétrer cet indicateur.

## Message Queue

Cette section décrit les problèmes connus liés aux files d'attente des messages Java et les solutions associées.

## **La reconnexion de JMS n'est pas complète dans certains cas dépendant des délais d'expiration (6173308, 6189645, 6198481, 6199510, 6208728)**

### **Description**

Dans des scénarios faisant appel à la synchronisation, plusieurs causes peuvent être à l'origine de ce problème.

### **Solution**

Pour contourner ces problèmes :

- Redémarrez les courtiers concernés.
- Redémarrez les instances d'Application Server concernées.

## **Server ne démarre pas si le courtier MQ n'est pas démarré (6740797)**

### **Description**

Si vous configurez JMS de façon à être DISTANT, Enterprise Server échoue à démarrer si le courtier MQ n'est pas démarré.

### **Solution**

Définissez l'option JVM suivante comme suit :

`com.sun.enterprise.jms.CONNECT_MQ_LAZILY=true` . Après la configuration de cette option JVM, vous pouvez démarrer Communications Server si le courtier MQ n'est pas lancé. Cependant, il est recommandé de démarrer MQ avant le serveur.

## **Le courtier MQ ne parvient pas à démarrer avec un profil de cluster sous Linux (6524871)**

### **Description**

Après avoir créé un domaine avec un profil de cluster sur un système Linux, il se peut que vous rencontriez une erreur `java.lang.OutOfMemoryError: Java heap space` et que l'instance du serveur ne parvienne pas à redémarrer suite au blocage du démarrage du courtier MQ. Il est impossible de récupérer le système après cette condition. Ce problème provient d'une configuration incorrecte du fichier `/etc/hosts` ; de manière plus spécifique, le nom d'hôte du serveur pointe vers l'adresse de loopback `127.0.0.1`.

### **Solution**

Un cluster du courtier MQ ne peut pas démarrer lorsque le périphérique réseau est configuré pour pointer vers l'adresse de loopback. Il ne s'agit pas d'un bogue. La solution est de vous assurer que le fichier `/etc/hosts` de l'hôte d'Communications Server ne pointe pas vers `127.0.0.1`.

## Une incompatibilité entre les anciennes et les nouvelles classes est créée lorsque `imqjmsra.jar` est chargé avant la mise à niveau (6740794)

### Description

Lors du démarrage du serveur, ce dernier vérifie la version de Message Queue. Si la version Message Queue est incorrecte, le serveur effectue alors sa mise à niveau à l'aide de `imqjmsra.jar`. Ceci met JAR à niveau et ses classes ne seront pas accessibles au serveur avant le prochain redémarrage d'Application Server. Cette situation se produit uniquement si Message Queue est mis à niveau seul, ou si l'on applique un patch au serveur d'application seul. Cette situation peut parfois entraîner l'effet secondaire suivant : Application Server ne démarre pas.

### Solution

Message Queue et Application Server doivent être maintenus au même niveau de patch, ou bien il faut redémarrer Application Server.

## Contrôle

Cette section décrit les problèmes connus liés au contrôle et les solutions associées.

### Certaines statistiques de contrôle du service HTTP ne présentent aucune information utile et doivent être ignorées (6174518)

#### Description

Lorsqu'on affiche les statistiques de contrôle de certains éléments du service HTTP, certaines valeurs présentées ne correspondent pas aux valeurs actuelles ou sont toujours à 0. En particulier, les statistiques du service HTTP suivantes ne présentent pas d'informations applicables à Application Server, et doivent être ignorées :

- `http-service`
  - `load1MinuteAverage`
  - `load5MinuteAverage`
  - `load15MinuteAverage`
  - `rateBytesTransmitted`
  - `rateBytesReceived`
- `pwc-thread-pool (the element)`

#### Solution

Ces contrôles seront supprimés dans les versions ultérieures et remplacés par des informations mieux adaptées.

### **L'ouverture de la navigation JNDI à partir de l'IG d'administration génère plusieurs exceptions dans le fichier `server.log` (6591734)**

#### **Description**

De nombreuses exceptions sont générées lorsque le navigateur JNDI est ouvert à partir de l'IG d'administration.

#### **Solution**

Aucune pour l'instant.

## **Sécurité**

Cette section décrit les problèmes connus liés aux certificats et à la sécurité des applications Web sous Communications Server ainsi que les solutions associées.

### **Erreur de mémoire saturée dans les scénarios SSL lors d'un stress important (JDK 6 Problème 23)**

#### **Description**

Un bogue JDK (Voir : [https://jdk6.dev.java.net/issues/show\\_bug.cgi?id=23](https://jdk6.dev.java.net/issues/show_bug.cgi?id=23)) dans JDK6 Sun PKCS11 Provider pourrait entraîner une erreur de mémoire saturée (`OutOfMemoryError`) lors de l'exécution de certains scénarios SSL en situation de stress important.

#### **Solution**

Si vous rencontrez ce problème, supprimez le fournisseur `sun.security.pkcs11.SunPKCS11` du fichier `java.security` dans votre installation JRE.

### **La terminaison SSL ne fonctionne pas (6269102)**

#### **Description**

La terminaison SSL ne fonctionne pas ; lorsque l'équilibreur de charge (matériel) est configuré pour la terminaison SSL, Communications Server passe du protocole `https` à `http` au cours de la redirection.

#### **Solution**

Ajoutez un équilibreur de charge logiciel entre l'équilibreur de charge matériel et Communications Server.

### **Fuites sur la connexion du socket avec SSL (6492477)**

#### **Description**

À cause d'un bogue JVM, un problème de fuite survient avec certaines versions JDK lorsque `security-enabled` est défini sur `true` sur un listener HTTP. Les étapes de reproduction de ce bogue sont les suivantes :



1. Définissez `security-enabled` sur `true` sur le listener HTTP :

```
<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. Commentez l'arrêt du domaine à la fin des tests quicklook.
3. Exécutez les tests quicklook.
4. Vérifiez l'utilisation du socket :

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

Les éléments suivants doivent être utilisés :

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

Ce problème fait l'objet d'un suivi sur le site GlassFish [https://glassfish.dev.java.net/issues/show\\_bug.cgi?id=849](https://glassfish.dev.java.net/issues/show_bug.cgi?id=849).

### Solution

Procédez à une mise à niveau vers la dernière version de JDK.

## Conteneur Web

Cette section décrit les problèmes connus liés au conteneur Web et les solutions associées.

**Sous Windows, le déploiement d'une application à l'aide de `--precompilejsp=true` risque de verrouiller les fichiers JAR de celle-ci, entraînant ainsi l'échec de l'annulation du déploiement ou du redéploiement ultérieur (5004315)**

### Description

Si vous devez effectuer une précompilation des pages JSP lors du déploiement d'une application sous Windows, les tentatives ultérieures de redéploiement ou d'annulation de déploiement de cette application (ou de toute autre application contenant le même ID de module) ne fonctionneront pas comme prévu. L'origine de ce problème provient du fait que la précompilation des pages JSP ouvre les fichiers JAR dans votre application, mais ne les referme pas. Windows empêche alors que le processus d'annulation du déploiement ne supprime ces fichiers ou que le processus de redéploiement ne les écrase.

Il est à noter que l'annulation du déploiement réussit partiellement dans la mesure où l'application est supprimée d'Application Server. Notez aussi qu'aucun message d'erreur n'est retourné par l'utilitaire `asadmin`, mais que le répertoire de l'application et les fichiers jar verrouillés ne sont pas supprimés du serveur. Le fichier journal du serveur contient les messages décrivant l'échec de la suppression des fichiers et du répertoire de l'application.

Toute tentative de redéploiement de l'application suite à l'annulation du déploiement échoue, car le serveur essaie en vain de supprimer le répertoire et les fichiers existants. Cela peut se produire si vous essayez de déployer une application qui utilise le même ID de module que celui de l'application initialement déployée. En effet, le serveur utilise cet ID de module lors de la sélection d'un répertoire destiné à contenir les fichiers de l'application.

Les tentatives de redéploiement de l'application sans annulation préalable du déploiement échouent pour les mêmes raisons.

### Diagnosics

Si vous essayez de redéployer l'application ou de la déployer après avoir annulé son déploiement, l'utilitaire `asadmin` renvoie une erreur similaire à l'erreur ci-dessous.

```
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.
```

### Solution

Ce problème ne se produit pas si vous définissez le paramètre par défaut `--precompilejsp=false` lors du déploiement d'une application. Lors de sa première utilisation, l'application déclenche la compilation des pages JSP. C'est pour cette raison que le temps de réponse de la première requête est supérieur à celui des requêtes suivantes.

Notez également qu'en cas de précompilation, vous devez arrêter et redémarrer le serveur avant d'annuler le déploiement de l'application ou de redéployer cette dernière. L'arrêt du serveur permet de libérer les fichiers JAR qui étaient verrouillés et d'effectuer correctement les opérations d'annulation du déploiement ou de redéploiement de l'application après le redémarrage.

### Impossible de déployer les archives WAR avec le fichier `web.xml` basé sur Servlet 2.4, contenant un élément `<load-on-startup>` vide (6172006)

#### Description

L'élément facultatif `load-on-startup` inclus dans le fichier `web.xml` indique que le servlet correspondant doit être chargé et initialisé au démarrage de l'application Web à laquelle il appartient.

Le contenu facultatif de cet élément est un nombre entier précisant en quelle position le servlet doit être chargé et initialisé par rapport aux autres servlets de l'application Web. Lorsque l'élément `<load-on-startup>` est vide, l'ordre de démarrage du servlet est inutile tant que celui-ci est chargé et initialisé au cours du démarrage de l'application Web dont il dépend.

Le schéma Servlet 2.4 du fichier `web.xml` ne prend plus en charge les éléments `<load-on-startup>` vides, ce qui signifie que vous devez obligatoirement indiquer un nombre entier lorsque vous utilisez un fichier `web.xml` basé sur le composant Servlet 2.4. Si vous laissez l'élément `<load-on-startup>` vide, tel que `<load-on-startup/>`, le fichier `web.xml` ne parvient pas à valider le schéma Servlet 2.4 pour `web.xml`, provoquant l'échec du déploiement de l'application Web.

Problème de compatibilité ascendante : Vous pouvez néanmoins laisser l'élément `<load-on-startup>` vide pour un fichier `web.xml` basé sur le composant Servlet 2.3.

### Solution

Définissez la valeur `<load-on-startup>0</load-on-startup>` lors de l'utilisation d'un fichier `web.xml` basé sur Servlet 2.4 afin d'indiquer que l'ordre de chargement du servlet n'est pas important.

## Impossible de compiler la page JSP sur des serveurs limités en ressources (6184122)

### Description

La page JSP est accessible mais ne peut pas être compilée. Le journal du serveur contient le message d'erreur "Impossible d'exécuter la commande" avec le suivi de pile suivant :

```
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)
```

### Solution

Définissez le commutateur de compilation JSP `fork` sur `false`.

Vous pouvez effectuer cette opération de l'une des deux manières suivantes:

- De façon globale, en définissant le paramètre `fork` init de `JspServlet` inclus dans le fichier `domain-dir/config/default-web.xml` sur `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>
```

- De façon ponctuelle, en définissant la propriété de configuration JSP fork incluse dans le fichier `sun-web.xml` sur `false` :

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

Les deux paramétrages empêcheront ant de générer un nouveau processus pour la compilation `javac` .

## Services Web

Cette section décrit les problèmes connus liés au conteneur Web et les solutions associées.

### Échec de la commande `wscompile` avec l'erreur "**package javax.xml.rpc does not exist**" sous JDK6 u4 b3 (6638567)

#### Description

La tâche `ant wscompile` échoue pour JDK 6 Mise à jour 4. Pour chaque classe JAX-RPC API, le message d'erreur suivant s'affiche :

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

#### Solution

Avant d'exécuter la tâche `ant wscompile`, assurez-vous que `javaee.jar` est spécifié dans le chemin de classe, et non *pas* `j2ee.jar`.