

Sun Java™ System Message Queue

Notas de la versión

Versión 3.5 SP1

Número de pieza 817-7199-10

Estas notas de la versión contienen información importante disponible en el momento del lanzamiento de la versión 3.5 SP1 de Sun Java™ System Message Queue (antes conocido como Sun™ ONE Message Queue). Este documento también incluye el contenido de las notas de la versión de Message Queue 3.5 para aquellos usuarios que deseen actualizar desde versiones anteriores a la 3.5. Aquí se tratan nuevas funciones y mejoras, limitaciones y problemas conocidos, notas técnicas e información de otro tipo acerca de Message Queue 3.5.

La versión más actualizada de estas notas de la versión se encuentra en el sitio Web de documentación de Sun Java System: http://docs.sun.com/coll/MessageQueue_35_SP1. Consulte el sitio Web antes de instalar y configurar el software y, después, visítelo de forma periódica para ver los manuales y las notas de la versión más actualizados.

En estas notas de la versión se incluyen los siguientes apartados:

- [Historial de revisiones](#)
- [Acerca de Message Queue 3.5 SP1](#)
- [Errores solucionados](#)
- [Información importante](#)
- [Limitaciones y problemas conocidos](#)
- [Archivos que se pueden distribuir](#)
- [Información sobre problemas y respuestas de los clientes](#)
- [Recursos adicionales de Sun](#)

Historial de revisiones

Tabla 1 Historial de revisiones

Fecha	Descripción de los cambios
12 de marzo de 2004	Actualización de la información sobre los errores. Actualización del apartado “Limitaciones y problemas conocidos” . Adición del apartado “Archivos que se pueden distribuir” . Actualización del apartado “Actualizaciones de la documentación” . Actualización del apartado “Problemas de compatibilidad” . Actualización del apartado “Sun Java System Información” .
9 de enero de 2004	Actualización de la información sobre la compatibilidad con PointBase 4.8; actualización de la información acerca de las funciones de las API escritas en C.

Acerca de Message Queue 3.5 SP1

Message Queue 3.5 SP1 es una actualización de Message Queue 3.5 que incluye todas las nuevas funciones de Message Queue 3.5. Además, Message Queue 3.5 SP1 cuenta con algunos errores solucionados y con un nuevo nombre comercial. Este producto pertenece ahora a la familia de productos Sun Java™ System.

Se ha certificado que Message Queue 3.5 SP1 es compatible con la especificación Java™ Message Service (JMS) 1.1: ha superado el Conjunto de pruebas de compatibilidad (Compatibility Test Suite, CTS) de JMS 1.1.

Este apartado describe los cambios efectuados en Message Queue 3.5 SP1 y aquellos que se realizaron en la versión anterior, Message Queue 3.5.

Message Queue 3.5 SP1

Message Queue 3.5 SP1 cuenta con errores solucionados y un nuevo nombre comercial para el producto y la documentación.

Message Queue 3.5

Message Queue 3.5 incluía numerosas funciones nuevas:

- “Compatibilidad con clientes en C (Enterprise Edition)” en la página 3
- “Recuperación ante fallos de conexión con clientes Java (Enterprise Edition)” en la página 4
- “Control del flujo de mensajes mejorado en el agente” en la página 5
- “Control del flujo mejorado del cliente Java” en la página 6
- “Nuevas métricas de destino” en la página 7
- “API de control remoto (Enterprise Edition)” en la página 8
- “Adaptador de recursos de Message Queue para JMS (compatible con Application Server J2EE)” en la página 8
- “Confirmación personalizada de los mensajes” en la página 8
- “Mejora de las políticas de entrega para colas (Enterprise Edition)” en la página 9
- “Mejora del rendimiento del clúster (Enterprise Edition)” en la página 10
- “Destinos locales (Enterprise Edition)” en la página 11
- “Clústeres de agentes más seguros (Enterprise Edition)” en la página 11
- “Mejora del rendimiento del almacenamiento persistente” en la página 11
- “Autenticación y autorización específica de la instancia” en la página 13
- “Instalaciones Linux basadas en RPM” en la página 13
- “Soporte para el sistema operativo Solaris, X86 Platform Edition” en la página 13

Dichas funciones se describen en los subapartados siguientes.

Compatibilidad con clientes en C (Enterprise Edition)

Message Queue 3.5 incluye una interfaz de programación de aplicaciones (API) escrita en C, así como soporte durante el tiempo de ejecución en C (en adelante, se hace referencia a esta función como “cliente en C”). La función de cliente en C se puede utilizar para integrar sistemas anteriores con el sistema de mensajería Message Queue. Se trata de una implementación casi completa de la especificación JMS. Admite todas las funciones JMS con excepción de: ciertos tipos de cuerpos (mapa, flujo y objeto), funciones de examinación de colas y funciones de servidor de aplicaciones de J2EE (como las transacciones distribuidas y los objetos `ConnectionConsumer`).

El soporte de la función de cliente en C lo proporciona un conjunto de bibliotecas que se instalan por separado y se habilita sólo si se dispone de una licencia Enterprise Edition. En consecuencia, para actualizar de Platform Edition a Enterprise Edition, es necesario instalar el archivo de licencia de Enterprise Edition y las bibliotecas C.

Los usuarios de Platform Edition que activen el periodo de 90 días de prueba de la licencia de Enterprise Edition pueden usar la función de cliente en C si se ponen en contacto con Sun usando el alias imq-feedback@sun.com para solicitar el kit de desarrollo de software (SDK) de la interfaz de programación de aplicaciones (API) escrita en C. Los ingenieros son los encargados de responder a estas solicitudes y poner el SDK de la API escrita en C en el sitio FTP anónimo. Una vez transcurridos los 90 días del periodo de prueba, la licencia caduca y los usuarios podrán seguir creando clientes en C, pero no podrán conectarlos a su agente de Platform Edition.

La función de cliente en C requiere versiones específicas del compilador en las diferentes plataformas de los sistemas operativos, lo que supone agregar nuevos requisitos a Enterprise Edition (consulte *Message Queue Installation Guide* para obtener más información). La función de cliente en C también depende de las bibliotecas Netscape Portable Runtime (NSPR) y Network Security Service (NSS). (En Message Queue 3.5, la función de cliente en C se probó con éxito en Linux Red Hat Advanced Server 2.1. Las versiones de las bibliotecas NSPR y NSS en las que se probó no se han certificado para esta edición de Linux.)

Actualmente, la API en C no admite el tipo de autenticación básica. Si configura el agente para que use este tipo de autenticación, fallará la llamada a la función `MQCreateConnection` y se mostrará el resultado `MQ_UNSUPPORTED_AUTH_TYPE`.

La documentación de la función del cliente en C incluye material de referencia, información sobre la programación y ejemplos de clientes API en C. Para obtener más información, consulte *Message Queue C Client Developer's Guide*.

Recuperación ante fallos de conexión con clientes Java (Enterprise Edition)

Message Queue 3.5 dispone de una función de reconexión automática mediante la cual se puede restablecer una conexión con un agente, ya sea el original u otro distinto (recuperación ante fallos de conexión con clientes). La reconexión se realiza con el servicio de mensajería en lugar de con una instancia de agente específica. Para que esta función se pueda llevar a cabo, es necesario configurar el objeto administrado mediante la fábrica de conexiones (Message Queue 3.5 cuenta con un nuevo esquema de especificación de direcciones para el servicio de mensajería). Para ello, debe especificar un conjunto de direcciones de agentes (`imqAddressList`). Cuando el tiempo de ejecución del cliente debe establecer (o reestablecer) una conexión con un servicio de mensajería, intentará conectarse a los agentes según el orden de prioridad de la lista, hasta que encuentre uno que esté disponible o no encuentre ninguno. Puede especificar el número de intentos de conexión (`imqAddressListIterations`) que se puede efectuar para conectarse a cada agente, así como el intervalo de tiempo que debe transcurrir entre los intentos (`imqAddressListInterval`).

Si la reconexión automática debe efectuarse a una instancia del agente que es distinta de la original, se pueden perder tanto los mensajes persistentes como las informaciones de estado que albergue el agente que ha fallado (o que se ha desconectado). Esto se debe a que las distintas instancias del agente presentes en un clúster no usan un almacén persistente compartido con una alta disponibilidad. No obstante, la capacidad del tiempo de ejecución del cliente para volver a conectarse automáticamente a una instancia del agente diferente le permite crear escenarios de recuperación mediante los que un agente de respaldo o un clúster de agentes se pueden usar como sistema de protección ante pérdidas de datos (aunque no sea de forma completa).

También, si la reconexión automática está habilitada, Message Queue 3.5 conservará los destinos *temporales* cuando falle la conexión asociada, debido al hecho de que los clientes se pueden volver a conectar y acceder a ellos de nuevo. Los destinos temporales tendrán el mismo tratamiento que otros destinos físicos, lo que puede requerir que analice regularmente los agentes de los destinos temporales que no se usen.

Para obtener más información, consulte *Message Queue Java Client Developer's Guide*.

Message Queue disponía anteriormente de una función de reconexión automática mediante la cual el tiempo de ejecución del cliente se podía reconectar a un agente si la conexión fallaba, excepto en situaciones en las que el estado del lado del cliente no se podía restablecer por completo en el agente al realizar la reconexión (por ejemplo, al usar sesiones de transacciones o destinos temporales, que existen sólo mientras dura la conexión).

Control del flujo de mensajes mejorado en el agente

Se han introducido mejoras en el agente para optimizar el control del flujo de mensajes a los destinos y para evitar situaciones en las que la producción de mensajes es mucho mayor que la de su consumo. Además, otras nuevas funciones de Message Queue 3.5 le pueden ayudar a eliminar los cuellos de botella en los flujos de mensajes *fuera* de los destinos. Consulte [“Control del flujo mejorado del cliente Java” en la página 6](#) y [“Mejora de las políticas de entrega para colas \(Enterprise Edition\)” en la página 9](#).

Entre las mejoras del flujo de mensajes del agente se incluyen las siguientes:

- Limitación del número de productores asociados a un destino. Los destinos disponen ahora de un nuevo atributo `maxNumProducers`. Cuando se alcanza este límite, no se podrán crear más productores para el destino en cuestión.
- Establecimiento de límites nuevos configurables, así como la definición de qué acciones se deben realizar cuando se alcancen los límites de `maxTotalMsgBytes` y `maxNumMsgs`. En concreto, se han implementado los siguientes cambios:
 - Ampliación de los atributos de destino `maxTotalMsgBytes` y `maxNumMsgs` a destinos de tema (anteriormente se aplicaban sólo a destinos de colas)
 - Activación de las preferencias `maxTotalMsgBytes` y `maxNumMsgs` para destinos *creados automáticamente*

- (Enterprise Edition) Posibilidad de que un administrador elija qué acción se debe realizar cuando se alcance alguno de los límites indicados. Puede especificar las siguientes acciones: ralentizar los productores (FLOW_CONTROL), eliminar los mensajes más antiguos (REMOVE_OLDEST), eliminar los mensajes de menor prioridad según su antigüedad (REMOVE_LOW_PRIORITY) y rechazar los mensajes nuevos (REJECT_NEWEST).
- Implementación del control del flujo de productores (FLOW_CONTROL) *por cada productor* en lugar de hacerlo por conexiones. La implementación anterior cerraba todos los productores de una conexión cuando se recibían demasiados mensajes por destino mediante la conexión. El control del flujo por productor cierra sólo los productores cuya conexión esté asociada al destino que se ha desbordado, lo que permite que otros productores de la conexión puedan seguir enviando mensajes a otros destinos.
- Posibilidad de que un administrador detenga en pausa y reanude un destino concreto. Puede pausar el envío de mensajes de los productores a los destinos, de los destinos a los consumidores o ambos. Esto se realiza mediante dos subcomandos nuevos de `imqcmd`: `pause` y `resume`, tal y como se muestra a continuación:
 - `imqcmd pause dst -n myQueue -t q -pst PRODUCERS`
 - `imqcmd resume dst -n myQueue -t q`

Para obtener más información, consulte *Message Queue Administration Guide*.

Control del flujo mejorado del cliente Java

El tiempo de ejecución del cliente Message Queue 3.5 gestiona el flujo de mensajes *por consumidor* y por conexión. Puede limitar el número de mensajes que se almacena en la memoria búfer por consumidor para evitar que un consumidor se vea saturado por otros consumidores. Esta función supone también que, en el caso de que haya entregas para colas dirigidas a varios consumidores, se puede equilibrar mejor el envío de mensajes a dichos usuarios. Esto también ayuda a gestionar los recursos de memoria en el tiempo de ejecución del cliente Message Queue.

Un nuevo atributo de fábrica de conexiones, `imqConsumerFlowLimit`, permite limitar el número de mensajes que se almacenan en la memoria búfer *por consumidor* para todos los consumidores que comparten una conexión común. Cuando el número de mensajes de la memoria búfer de un consumidor es inferior al porcentaje definido (`imqConsumerFlowThreshold`) de `imqConsumerFlowLimit`, el agente puede enviar otro lote de mensajes al tiempo de ejecución del cliente para uso de dicho consumidor. Si el número total de mensajes almacenados en búfer para todos los consumidores de una conexión excede el valor de `imqConnectionFlowLimit`, entonces la entrega de mensajes de la conexión cesará hasta que el número sea inferior al valor especificado.

La implementación anterior del control del flujo del tiempo de ejecución del cliente le permitía limitar el número de mensajes almacenados en búfer en el tiempo de ejecución del cliente en espera de ser consumidos (`imqConnectionFlowLimit`). El propósito de esta función era limitar la cantidad de memoria usada por el cliente para almacenar en búfer los mensajes de forma que los clientes que consumían más espacio no se quedaran sin memoria. Esta función se implementó en el nivel de conexión, lo que significa que si una conexión admite numerosos consumidores, una avalancha de mensajes procedentes de un consumidor podría evitar que otros consumidores recibieran mensajes.

Para obtener más información, consulte *Message Queue Java Client Developer's Guide*.

Nuevas métricas de destino

Message Queue 3.5 incluye un sistema mejorado de seguimiento de mensajes y consumidores por destino, lo que permite una mejor supervisión y un mejor control de la memoria y el uso.

Las nuevas métricas aparecen como resultado del nuevo subcomando `imqcmd metrics dst`. Este comando muestra los totales acumulados (desde el inicio del muestreo), los valores actuales, los valores medios (calculados según las muestras recogidas) y los valores de los picos (desde el inicio del muestreo) tanto para métricas de mensajes como de consumidores.

Por ejemplo, el comando `imqcmd metrics dst -m ttl` devuelve la siguiente información:

- Flujo de mensajes
 - Flujo de mensajes entrante: totales acumulados, tasa
 - Flujo de mensajes saliente: totales acumulados, tasa
- Mensajes almacenados en el agente (mensajes no confirmados en la memoria o en almacén persistente)
 - Número de mensajes: actual, pico, media
 - Bytes de mensajes: actual, pico, media
- Mensaje más extenso hasta el momento, bytes

El comando `imqcmd metrics dst -m` con devuelve la siguiente información:

- Número de consumidores activos: actual, pico, media (consulte [“Mejora de las políticas de entrega para colas \(Enterprise Edition\)” en la página 9](#))
- Número de consumidores de respaldo: actual, pico, media (consulte [“Mejora de las políticas de entrega para colas \(Enterprise Edition\)” en la página 9](#))

Para obtener más información, consulte *Message Queue Administration Guide*.

API de control remoto (Enterprise Edition)

Message Queue 3.5 proporciona una API basada en mensajes mediante la cual los clientes remotos JMS (o locales) pueden gestionar y analizar fácilmente las métricas del agente. La API está basada en la capacidad del agente para producir mensajes que contienen información sobre las métricas del agente, la máquina virtual Java y los destinos individuales (consulte [“Nuevas métricas de destino” en la página 7](#)). Los mensajes se envían a destinos de temas específicos, en función de la entidad que se esté supervisando, cada vez que uno o varios consumidores se suscriben a los destinos. El cliente consumidor puede recuperar los mensajes, usar una propiedad de encabezado (`type`) para filtrarlos y, a continuación, extraer la información sobre las métricas que contienen.

Para obtener más información, consulte *Message Queue Administration Guide* y *Message Queue Java Client Developer's Guide*.

(Message Queue antes sólo admitía el inicio de sesión local de los datos de las métricas del agente y las consultas remotas de la información sobre las métricas usando las utilidades de administración de Message Queue. Estas funciones, aunque proporcionaban datos de métricas importantes, no facilitaban el análisis de este tipo de datos.)

Adaptador de recursos de Message Queue para JMS (compatible con Application Server J2EE)

Message Queue 3.5 incluye un adaptador de recursos JMS para conectar el servicio de mensajería Message Queue JMS a cualquier servidor de aplicaciones compatible con J2EE.

Un adaptador de recursos es un modo estándar de agregar funciones adicionales a un servidor de aplicaciones J2EE (mediante la conexión a un EIS, un sistema de mensajería, etc.) ajustándose a la especificación J2EE Connector Architecture (JCA 1.5). Esta arquitectura hace posible que cualquier servidor de aplicaciones J2EE admita, por ejemplo, un sistema de mensajería JMS con sólo establecer una conexión con un proveedor JMS que incorpore la especificación JCA 1.5: los componentes J2EE implementados y ejecutados en el entorno del servidor de aplicaciones pueden intercambiar mensajes JMS usando el proveedor JMS conectado (servidor y tiempo de ejecución del cliente).

Para obtener más información, consulte *Message Queue Administration Guide*.

Confirmación personalizada de los mensajes

Message Queue actualmente admite el modo de confirmación del cliente JMS `CLIENT_ACKNOWLEDGE` mediante el cual un cliente JMS confirma explícitamente el consumo de mensajes. En el modo `CLIENT_ACKNOWLEDGE`, el cliente ejecuta el método `acknowledge()` de un objeto de mensaje, lo que provoca la confirmación de todos los mensajes consumidos por la sesión desde la última ejecución del método.

Message Queue 3.5 mejora este procedimiento al permitirle que sea usted quien confirme los mensajes *individualmente*. Es decir, puede confirmar sólo un mensaje concreto, en lugar de confirmar por lotes todos los mensajes consumidos hasta ese momento. Esto se logra con la conversión del objeto de mensaje en un tipo de mensaje especial de Message Queue mediante el que se puede ejecutar un nuevo método `acknowledge()`. Esto le permite desviarse del estándar JMS para gestionar necesidades específicas de la aplicación.

Para obtener más información, consulte *Message Queue Java Client Developer's Guide*.

Mejora de las políticas de entrega para colas (Enterprise Edition)

Se han realizado cambios en la implementación de las entregas para colas dirigidas a varios consumidores, que antes estaba implementada como tres políticas de entrega para colas distintas (colas de recepción individual, de conmutación por error y de operación por turnos). Message Queue 3.5 usa un enfoque más generalizado en el que las entregas se equilibran entre un número de consumidores activos y de respaldo que se puede configurar. La implementación de Message Queue 3.5 está basada en los siguientes atributos de destino nuevos:

- `maxNumActiveConsumers`: Especifica el número de consumidores (uno o varios) activos en la entrega para colas con equilibrio de carga
- `maxNumBackupConsumers`: Especifica el número de consumidores de respaldo (ninguno o varios) que pueden ocupar el lugar de los consumidores activos en caso de que alguno de éstos falle.

(Los nuevos consumidores serán rechazados si su número supera la cantidad que resulta de la suma de estos dos atributos.)

Message Queue Platform Edition admite la entrega para colas con equilibrio de carga hasta para dos consumidores, mientras que Enterprise Edition admite un número ilimitado de consumidores.

El nuevo mecanismo de equilibrado de carga tiene en cuenta la tasa de consumo de mensajes de los diferentes consumidores. Funciona de esta forma:

- Un número inicial de mensajes puestos en cola en un destino se dirigen a los consumidores activos disponibles (según el orden en que estén registrados en el destino) en lotes del tamaño indicado (atributo `consumerFlowLimit` del destino). Una vez que se hayan entregado dichos mensajes, los otros mensajes que lleguen se dirigen a los consumidores uno a uno, a medida que los consumidores van estando disponibles (es decir, conforme los consumidores confirman que han recibido todos los mensajes que se les han enviado previamente). Si un consumidor activo falla, entonces el primer consumidor de respaldo pasa a estar activo y se hace cargo del trabajo del consumidor que ha fallado.

- En un entorno de clúster de agentes, el mecanismo de entrega se puede definir de forma que dé prioridad a los consumidores locales. Un nuevo atributo de destino, `localDeliveryPreferred`, le permite especificar qué mensajes se deben entregar a los consumidores remotos en caso de que no haya consumidores en el agente local (el agente en el que se creó el destino). Esto incrementa el rendimiento en situaciones en las que la distribución a clientes remotos (a través de sus agentes locales respectivos) puede causar retrasos en la producción. Este atributo requiere que el ámbito del destino no esté restringido sólo a las entregas locales. Consulte [“Mejora del rendimiento del clúster \(Enterprise Edition\)” en la página 10](#).

Para obtener más información, consulte *Message Queue Administration Guide*.

Mejora del rendimiento del clúster (Enterprise Edition)

En un entorno de clúster de agentes, los destinos se replican en todos los agentes y todos los mensajes entregados a dichos destinos se reenvían a todos los agentes que tienen consumidores registrados en los destinos en cuestión, incluso si sólo un pequeño porcentaje de los mensajes se va a entregar a un consumidor en concreto (por ejemplo, en el caso de un suscriptor duradero que usa criterios de selección o un receptor de colas involucrado en una entrega para colas con equilibrio de carga). Este tráfico de agente a agente puede causar avalanchas de mensajes, especialmente cuando un consumidor nuevo pasa a estar activo. Para reducir un tráfico excesivo entre agentes en un clúster, Message Queue 3.5 incluye las siguientes mejoras:

- Adopción de nuevos mecanismos de control de flujo que regulan la entrega de mensajes en una conexión de consumidor. En otras palabras, el *consumidor* regulará la entrega de mensajes (desde el destino al tiempo de ejecución del cliente), lo que impide el paso de mensajes innecesarios de agente a agente. Estos mecanismos también ayudan a evitar que se sature el tiempo de ejecución del cliente. Consulte [“Control del flujo mejorado del cliente Java” en la página 6](#).
- Cambio de la implementación de la entrega para colas destinada a varios consumidores (consulte [“Mejora de las políticas de entrega para colas \(Enterprise Edition\)” en la página 9](#)) con objeto de reducir el paso de mensajes innecesarios de agente a agente. Esta implementación incluye un nuevo atributo de destino de cola, `localDeliveryPreferred`, que le permite especificar que los consumidores locales deben tener prioridad con respecto a los consumidores remotos (consulte [“Mejora de las políticas de entrega para colas \(Enterprise Edition\)” en la página 9](#)).

Para obtener más información, consulte *Message Queue Administration Guide*.

Destinos locales (Enterprise Edition)

Un nuevo atributo de destino, `isLocalOnly`, le permite especificar un destino para que envíe mensajes sólo a los consumidores locales (aquellos que están conectados al agente en el que se crea el destino) en contraposición a los consumidores conectados a otros agentes del clúster. Del mismo modo, se puede configurar para que sólo reciba los mensajes enviados desde los productores locales. Puede usar esta propiedad para crear destinos independientes, destinos no interactivos con el mismo nombre en agentes diferentes de un clúster y para definir situaciones de recuperación ante fallos mediante las cuales un mensaje se envía a dos destinos, por si uno de ellos falla.

Para obtener más información, consulte *Message Queue Administration Guide*.

Clústeres de agentes más seguros (Enterprise Edition)

En situaciones en la que la seguridad es un imperativo, la entrega de mensajes entre el cliente y el servidor de mensajes debe ser cifrada. Para ello, Message Queue 3.5 es compatible ahora con la entrega segura de mensajes entre los agentes de un clúster. Para conseguir una entrega de mensajes segura y cifrada en un clúster, es necesario configurar el servicio interno de conexión del clúster para que sea posible usar un protocolo de transporte basado en SSL.

Para obtener más información, consulte *Message Queue Administration Guide*.

Mejora del rendimiento del almacenamiento persistente

La implementación del almacén de datos de archivos sin formato de Message Queue y el almacén de datos compatibles con JDBC se ha modificado en Message Queue 3.5 para mejorar el rendimiento. Dichas mejoras se describen en los dos apartados siguientes. Para obtener más información, consulte *Message Queue Administration Guide*.

Persistencia incorporada (almacén de datos de archivos sin formato)

La mejora del rendimiento del almacén de datos de archivos sin formato de Message Queue conlleva cambios internos en el formato de los datos que sólo son evidentes al actualizar de Message Queue 3.01 (o anterior) a Message Queue 3.5.

La migración del almacén de archivos se gestiona automáticamente cuando se inicia por primera vez una instancia de agente de Message Queue 3.5 y hace referencia a una versión anterior del almacén de archivos. Una copia del almacén de archivos anterior se guarda en el directorio de instancias y deberá eliminarla manualmente una vez que la migración se haya completado. Para eliminar automáticamente el almacén de archivos anterior, en caso de que no haya espacio suficiente en disco para dos copias del almacén, puede iniciar la versión 3.5 del agente con una opción agregada, tal y como se muestra en el siguiente comando:

```
imqbrokerd -upgrade-store-nobackup
```

(No hay espacios en `upgrade-store-nobackup`.)

La raíz del nuevo almacén de datos de archivos sin formato se cambia de:

```
.../instances/nombre_instancia/filestore/
```

a:

```
.../instances/nombre_instancia/fs350/.
```

Además, la utilidad Command de Message Queue (`imqcmd`) también se ha perfeccionado para proporcionar métricas sobre el almacén de archivos:

```
imqcmd metrics dst -n nombre_destino -t tipo -m dsk
```

Además, `imqcmd` incluirá un nuevo comando para comprimir:

```
imqcmd compact dst -n nombre_destino -t tipo
```

Persistencia conectada (almacén de datos compatible con JDBC™)

El almacén de datos compatible con JDBC de Message Queue ha cambiado en Message Queue 3.5 para dar cabida a mejoras realizadas en la gestión de la memoria del agente (consulte [“Control del flujo de mensajes mejorado en el agente” en la página 5](#) y [“Nuevas métricas de destino” en la página 7](#)) y para incluir una gama mayor de proveedores de bases de datos. Las modificaciones para hacer posible las mejoras de gestión de la memoria (categorizar los mensajes por destino) son cambios de esquema que se gestionan de forma transparente, pero el soporte de proveedores de bases de datos adicionales lleva consigo cambios en la forma de configurar la persistencia conectada.

La migración del almacén de archivos compatible con JDBC se gestiona automáticamente cuando se inicia por primera vez una instancia de agente de Message Queue 3.5 con una versión anterior del almacén persistente. Sin embargo, las tablas antiguas se conservarán intactas y será necesario eliminarlas manualmente con el nuevo comando `imqdbmgr delete oldtbl` una vez que la migración se haya completado. Para eliminar las tablas antiguas de forma automática, en el caso en que no disponga de espacio suficiente para dos copias del almacén, puede iniciar el agente con la nueva opción: `imqbrokerd -upgrade-store-nobackup`.

Al igual que para el soporte de base de datos adicional, no se podían personalizar las instrucciones SQL de la base de datos compatible con JDBC que deseaba conectar (las instrucciones SQL se generaban internamente). Message Queue 3.5 incluye ahora nuevas propiedades de configuración de instancias que le permiten personalizar el código SQL que crea el esquema de base de datos de Message Queue. Existe una propiedad configurable para cada tabla de la base de datos: la propiedad es el código SQL que crea la tabla. Dichas propiedades son necesarias para especificar correctamente los tipos de datos usados por la base de datos conectada. Se proporcionan ejemplos basados en la base de datos incorporada de PointBase, en lugar de en la base de datos anterior de Cloudscape.

Autenticación y autorización específica de la instancia

De forma predeterminada, con Message Queue 3.5, cada instancia tiene su propio repositorio de usuario basado en archivos y su propio archivo de control de acceso, ambos colocados en una ubicación estándar: `.../instances/nombre_instancia/etc/`. Estos dos archivos se crearán cuando la instancia del agente se inicie por primera vez. Si el agente encuentra estos archivos en la ubicación antigua (lo que suele darse cuando se actualiza desde una versión anterior), copiará los archivos en la ubicación específica de la instancia. Si el agente no los encuentra en la ubicación antigua (que es el caso de las instalaciones nuevas), colocará las versiones predeterminadas de los archivos en la ubicación específica de la instancia.

Para dar soporte a repositorios de usuario específicos de la instancia, se ha agregado una opción `-i nombre_instancia` a la utilidad User Manager (`mqusermgr`) para determinar el repositorio de usuario específico de la instancia al que se aplica cada comando `mqusermgr`.

Para obtener más información, consulte *Message Queue Administration Guide*.

Anteriormente, de forma predeterminada, todas las instancias del agente en un mismo ordenador compartían el mismo repositorio de usuario basado en archivos y, en consecuencia, la misma contraseña de inicio de sesión del cliente, y el mismo archivo de control de acceso. No obstante, se podía configurar cada instancia del agente para que usara una ubicación del repositorio de usuario LDAP específico o para que usara un archivo de control de acceso específico (ambos especificados en el archivo de configuración de la instancia).

Instalaciones Linux basadas en RPM

La instalación de Message Queue 3.5 en Linux se realiza usando Red Hat Package Manager (RPM), un sistema de gestión de paquetes mediante la línea de comandos que puede instalar, desinstalar, verificar, consultar y actualizar paquetes de software (RPM).

Además, la estructura de directorio instalada para Message Queue en Linux ha cambiado para ajustarse a las ubicaciones estándar usadas para los productos de Linux que no se distribuyen en paquetes. Las plataformas Solaris™ y Linux disponen de estándares que dependen de si el producto se distribuye o no en paquetes junto con el sistema operativo. En concreto, ya no existe un directorio de instalación raíz de Message Queue en Linux, al igual que ocurre con Solaris.

Para obtener más información, consulte *Message Queue Installation Guide*.

Soporte para el sistema operativo Solaris, X86 Platform Edition

En Solaris 9, Message Queue 3.5 es compatible con procesadores X86, además de con los procesadores SPARC.

Requisitos de hardware y software

El hardware y el software requeridos para esta versión y los productos y plataformas admitidos se describen detalladamente en *Message Queue Installation Guide*.

Errores solucionados

Este apartado incluye breves descripciones de los errores solucionados:

- La [Tabla 2 en la página 15](#) describe los errores solucionados en Message Queue 3.5 SP1.
- La [Tabla 3 en la página 16](#) describe los errores solucionados en Message Queue 3.5.

Si desea acceder a listas anteriores de errores solucionados, consulte lo siguiente:

- Para Message Queue 3.0.1 Service Pack 2, consulte *Message Queue 3.0.1 Service Pack 2 Release Notes* en:
http://docs.sun.com/coll/S1_MessageQueue_301_SP2
- Para Message Queue 3.0.1, consulte *Message Queue 3.0.1 Release Notes* en:
http://docs.sun.com/coll/S1_MessageQueue_301
- Para Message Queue 3.0, consulte *Message Queue 3.0 Release Notes* en:
http://docs.sun.com/coll/S1_MessageQueue_30

Para obtener información más detallada sobre los errores solucionados, puede consultar el informe completo en el sitio de Java Developer Connection:

<http://developer.java.sun.com/developer/bugParade>

Errores solucionados en Message Queue 3.5 SP1

La [Tabla 2](#) describe los errores solucionados en Message Queue 3.5 SP1. (La [Tabla 3 en la página 16](#) describe los errores solucionados en Message Queue 3.5).

Tabla 2 Errores solucionados en Message Queue 3.5 SP1

Número de error	Descripción
4942723	Puede que el agente se quede sin memoria al enviar mensajes extensos con la opción de grupo de subprocesos compartidos.
4944894	El agente puede generar ocasionalmente una excepción <code>CancelledKeyException</code> al usar grupos de subprocesos compartidos.
4947239	La creación y el cierre repetido de los productores provocan una pequeña cantidad de crecimiento en la memoria del cliente.
4947993	No se puede destruir un destino o duración con una suscripción duradera.
4948525	Se pueden mostrar números negativos en el resultado de las métricas para los bytes de mensajes de entrada y salida. Esto ocurre cuando se envían más de 2.143.510.810 bytes.
4948563	Conversión de paquetes: mensaje de información que se muestra cada vez que se envía un mensaje 2.0 SP1 a un agente 3.5. Cada vez que un cliente 2.0 SP1 envía un mensaje a un agente 3.5, se muestra el siguiente mensaje de información: [04/Nov/2003:10:34:16 PST] Internal Error: Unknown ProducerUID 0
4949781	Al iniciar el agente se muestra un mensaje acerca de la imposibilidad de usar el multidifusor del clúster.
4952332	Puede que los mensajes se envíen sin orden previa en caso de que el consumidor principal falle durante una copia de respaldo del consumidor en la misma conexión.
4956748	No se puede usar un agente maestro con la base de datos de Oracle.
4964703	API en C: el encabezado <code>MESSAGE_ID</code> que devuelve la función <code>MQGetMessageHeaders()</code> no está precedido de "ID:".
4964712	API en C: el encabezado <code>MESSAGE_ID</code> definido por la función <code>MQSetMessageHeaders()</code> no se omite en el mensaje que se envía.
4969583	API en C: la misma gestión del mensaje debe ser capaz de ejecutar <code>MQAcknowledgeMessages()</code> más de una vez.
4983150	El indicador <code>JMSRedelivered</code> no se define cuando un agente se reinicia y vuelve a enviar un mensaje.
4983699	El agente pierde las excepciones iniciadas por el almacén cuando falla al persistir un mensaje.

Errores solucionados en Message Queue 3.5

La [Tabla 3](#) describe los errores solucionados en Message Queue 3.5.

Tabla 3 Errores solucionados en Message Queue 3.5

Número de error	Descripción
4449354	En casos poco frecuentes, si se llama a los métodos <code>Connection.stop</code> , <code>Connection.start</code> y <code>Connection.close</code> al mismo tiempo que se llama a <code>Session.recover</code> y <code>Session.rollback</code> (en subprocesos independientes), se puede provocar una petición de reenvío de mensajes inesperada.
4630183	La destrucción de un destino deja suscripciones duraderas en el agente.
4753010	Crecimiento ilimitado del segmento del montón nativo del proceso Java con VM de servidor.
4761626	La creación y destrucción excesiva de consumidores con colas de creación automática pueden provocar la pérdida de mensajes.
4855307	El agente no se puede autenticar en un repositorio LDAP porque la configuración predeterminada usa un nombre de propiedad antigua (<code>bindDN</code>).
4883126	La función de reconexión automática no funciona correctamente.
4888270	La transmisión de nuevo de un mensaje enviado originalmente en una transacción provoca un error de agente.
4431924	<code>imgadmin</code> : los diálogos modales pueden provocar una situación de interbloqueo. La consola de administración (<code>imgadmin</code>) utiliza diálogos modales para las aplicaciones. La mayoría de estos diálogos aparecen de forma explícita al interactuar con la interfaz gráfica de usuario; por ejemplo, si selecciona la opción de menú “Agregar agentes”. Sin embargo, algunos diálogos pueden aparecer también a causa de la pérdida de la conexión de agente. Al abrirse varios diálogos, la consola de administración se bloquea. No podrá cerrar los diálogos modales con el botón “Cerrar”.
4703406	<code>QueueBrowser</code> debería funcionar sin llamar primero a <code>connection.start()</code> . Se debe llamar a <code>Connection.start()</code> en una conexión antes de que <code>QueueBrowser</code> pueda examinar una cola. Si no puede llamar a <code>Connection.start()</code> , la enumeración de <code>QueueBrowser</code> bloqueará <code>nextElement()</code> y acabará devolviendo una excepción <code>java.util. NoSuchElementException</code> .
4866814	En Solaris, el agente no puede registrar mensajes de error ni advertencia usando <code>syslog</code> si éste se ha iniciado con una JVM de 64 bits (el agente se inicia con <code>-vmargs -d64</code>). Esto ocurre porque la versión beta de Message Queue no contiene una versión de 64 bits de nuestra biblioteca, <code>libimqutil.so.1</code> .
4872121	El agente no se inicia en un sistema que no esté en red y que tenga una dirección IP distinta de 127.0.0.1.
4879902	Aumenta la ralentización de la memoria en el agente.
4881968	No se pueden crear nuevos clientes de supervisión si <code>img.autocreate.topic</code> está definido como falso.
4884827	Las pruebas CTS1.3 MDB/EJB CMT fallan con Message Queue 3.5 y con AppServer 7.0

Tabla 3 Errores solucionados en Message Queue 3.5 (*continuación*)

Número de error	Descripción
4885654	Los productores pueden fallar si un nuevo mensaje se publica en un destino creado automáticamente en el mismo momento que el sistema está cerrando el destino.
4887506	Durante la recuperación ante fallos de un único cliente principal a un cliente de respaldo, los mensajes se pueden enviar sin orden previa.
4888939	Los clientes C y Java de un destino con una acción <code>FLOW_CONTROL</code> pueden dejar de recibir mensajes si el tamaño máximo del destino (<code>maxNumMsgs</code>) es muy pequeño (< 5 mensajes).
4889002	La propiedad <code>imq.transaction.autorollback</code> no se admite en la versión 3.5 beta.
4891874	El control del flujo basado en consumidores puede provocar que los mensajes dejen de enviarse a los consumidores. Este problema es más probable que se produzca con 4896133: Beans controlados mediante mensajes de <code>ConnectionConsumers</code> y Sun Java System Application Server 7 .
4895262	Los clientes HTTPS fallan al conectarse al agente mediante <code>HTTPSTunnelServlet</code>
4897500	En un clúster, cuando un cliente ejecuta <code>unsubscribe()</code> para eliminar una suscripción duradera, ésta sólo se elimina del agente al que está conectado el cliente. Esto significa que los mensajes producidos para otros agentes continuarán almacenándose para dicho suscriptor.
4898020	Los agentes de Message Queue 3.0.* y Message Queue 3.5 no se pueden usar juntos en un clúster. El inicio de un clúster mixto generará un error en el agente 3.0.1: discordancia de configuración: anulando la conexión con el agente [...] ya que las siguientes propiedades de configuración no coinciden: <code>null imq.queue.deliverypolicy</code>
4888983	<code>imqcmd list dur</code> no muestra suscriptores duraderos con el mismo nombre duradero.

Información importante

Este apartado incluye la última información que contiene la documentación principal del producto. En este apartado, se describen los siguientes temas:

- [Notas de la instalación](#)
- [Problemas de compatibilidad](#)
- [Actualizaciones de la documentación](#)

Notas de la instalación

Consulte *Message Queue Installation Guide* para obtener más información acerca de los requisitos de sistema; los productos y las plataformas de software; las instrucciones previas a la instalación; los procedimientos de actualización; y otra información relevante para instalar Message Queue en las plataformas Solaris, Linux y Windows.

Problemas de compatibilidad

Este apartado trata sobre los problemas de compatibilidad existentes en Message Queue 3.5 SP1 y en Message Queue 3.5.

Problemas relacionados con la próxima versión principal de Message Queue

A continuación, se muestran los cambios de incompatibilidades que se pueden provocar con la próxima versión principal de Message Queue. La información se proporciona ahora para que pueda adoptar las medidas necesarias ante dichos cambios.

- El cliente Message Queue es compatible con todas las versiones de J2SE 1.3 que salgan al mercado. J2SE 1.4 seguirá siendo admitido.
- Todas las interfaces de comandos de Message Queue se modificarán para eliminar la opción que permite proporcionar una contraseña como argumento de línea de comandos. Por ejemplo:

```
imqbrokerd -ldappassword <contraseña> imqcmd -p <contraseña>
```

Se proporcionarán métodos alternativos para especificar la contraseña.

- El formato del archivo de registro del agente va a sufrir modificaciones. Es posible que las aplicaciones que dependen del formato actual ya no funcionen.
- Puede que cambien las ubicaciones de los archivos individuales instalados como partes de Message Queue. Esto puede afectar a las aplicaciones existentes que dependen de las ubicaciones actuales de ciertos archivos de Message Queue.
- Puede que se elimine el programa `imqkeytool` del producto. J2SE será la `keytool` que se utilice en su lugar.
- Los clientes de Message Queue que usen una versión de Message Queue más antigua que la próxima versión principal es posible que no puedan acceder a las nuevas funciones que se incluirán en dicha nueva versión del producto.
- No se genera ningún error al ejecutar la función `MQAcknowledgeMessages()` desde clientes en C (usando `MQ_CLIENT_ACKNOWLEDGE`) en un mensaje que ya se haya confirmado. Esta conducta está sujeta a cambios.

Problemas de Message Queue 3.5

En líneas generales, Message Queue 3.5 es compatible con Message Queue 3.0 (y las versiones posteriores 3.0.1, 3.0.1 Service Pack 1 y 3.0.1 Service Pack 2). No obstante, los cambios realizados en las propiedades, los objetos administrados, el esquema de persistencia, las ubicaciones de los archivos y las herramientas de administración de los agentes pueden influir en la actualización de las versiones de Message Queue 3.0 a Message Queue 3.5.

La instalación de Message Queue 3.5 no elimina ni sobrescribe el directorio `IMQ_VARHOME` de Message Queue 3.0. Este directorio contiene archivos de configuración y de aspectos relacionados con la seguridad. La mayor parte de estos datos es compatible con Message Queue 3.5 y se pueden conservar usando las instrucciones que figuran en *Message Queue Installation Guide*.

Entre los problemas a los que puede enfrentarse a la hora de actualizar de Message Queue 3.0 a Message Queue 3.5, se incluyen los siguientes:

- [Compatibilidad del agente](#)
- [Cambios en las propiedades y los atributos](#)
- [Ubicación de los archivos públicos .jar](#)

Para obtener información acerca de la compatibilidad con los objetos administrados, con los clientes y con las herramientas de administración, consulte *Message Queue Installation Guide*.

Compatibilidad del agente

Un agente de Message Queue 3.5 funcionará con un agente de Message Queue 3.0, a pesar de los cambios efectuados en las propiedades y el esquema de almacenamiento persistente del agente. Algunos de los datos de Message Queue 3.0 siguen siendo compatibles con Message Queue 3.5. Para obtener más información, consulte *Message Queue Installation Guide*.

Cambios en las propiedades y los atributos

Este apartado contiene un resumen de los cambios realizados en las propiedades, los atributos de destino y los atributos de la fábrica de conexiones del agente en Message Queue 3.5.

Propiedades del agente En las siguientes tablas se enumeran las propiedades nuevas, las obsoletas y los cambios de nombre de propiedades que se han llevado a cabo en Message Queue 3.5. Para obtener más información, consulte el capítulo 2 de *Message Queue Administration Guide*.

Tabla 4 Nuevas propiedades del agente de Message Queue 3.5

Nombre de propiedad	Referencia a la función
<code>imq.persist.file.message.max_record.size</code>	“Mejora del rendimiento del almacenamiento persistente” en la página 11
<code>imq.persist.file.destination.message.filepool.limit</code>	“Mejora del rendimiento del almacenamiento persistente” en la página 11
<code>imq.metrics.topic.enabled</code>	“API de control remoto (Enterprise Edition)” en la página 8
<code>imq.metrics.topic.interval</code>	“API de control remoto (Enterprise Edition)” en la página 8
<code>imq.metrics.topic.persist</code>	“API de control remoto (Enterprise Edition)” en la página 8
<code>imq.metrics.topic.timetolive</code>	“API de control remoto (Enterprise Edition)” en la página 8
<code>imq.autocreate.destination.maxNumMsgs</code>	“Control del flujo de mensajes mejorado en el agente” en la página 5
<code>imq.autocreate.destination.maxTotalMsgBytes</code>	“Control del flujo de mensajes mejorado en el agente” en la página 5
<code>imq.autocreate.destination.maxBytesPerMsg</code>	“Control del flujo de mensajes mejorado en el agente” en la página 5
<code>imq.autocreate.destination.maxNumProducers</code>	“Control del flujo de mensajes mejorado en el agente” en la página 5
<code>imq.autocreate.queue.maxNumActiveConsumers</code>	“Mejora de las políticas de entrega para colas (Enterprise Edition)” en la página 9
<code>imq.autocreate.queue.maxNumBackupConsumers</code>	“Mejora de las políticas de entrega para colas (Enterprise Edition)” en la página 9
<code>imq.autocreate.queue.consumerFlowLimit</code>	“Control del flujo mejorado del cliente Java” en la página 6 y “Mejora de las políticas de entrega para colas (Enterprise Edition)” en la página 9
<code>imq.autocreate.topic.consumerFlowLimit</code>	“Control del flujo mejorado del cliente Java” en la página 6
<code>imq.autocreate.queue.localDeliveryPreferred</code>	“Mejora del rendimiento del clúster (Enterprise Edition)” en la página 10
<code>imq.autocreate.destination.isLocalOnly</code>	“Destinos locales (Enterprise Edition)” en la página 11

Tabla 5 Funciones obsoletas del agente de Message Queue 3.5

Nombre de propiedad
<code>imq.persist.file.message.fdpool.limit</code>
<code>imq.persist.file.message.filepool.limit</code>
<code>imq.redelivered.optimization</code>
<code>imq.queue.deliverypolicy</code>

En la siguiente tabla aparecen los atributos de función nuevos y obsoletos de Message Queue 3.5. Para obtener más información, consulte el capítulo 6 de *Message Queue Administration Guide*.

Tabla 6 Nuevos atributos de destino de Message Queue 3.5

Tipo de destino	Nombre de atributo	Referencia a la función
Cola y tema	<code>maxNumMsgs</code>	“Control del flujo de mensajes mejorado en el agente” en la página 5
Cola y tema	<code>maxTotalMsgBytes</code>	“Control del flujo de mensajes mejorado en el agente” en la página 5
Cola y tema	<code>limitBehavior</code>	“Control del flujo de mensajes mejorado en el agente” en la página 5
Cola y tema	<code>maxBytesPerMsg</code>	“Control del flujo de mensajes mejorado en el agente” en la página 5
Cola y tema	<code>maxNumProducers</code>	“Control del flujo de mensajes mejorado en el agente” en la página 5
Cola sólo	<code>maxNumActiveConsumers</code>	“Mejora de las políticas de entrega para colas (Enterprise Edition)” en la página 9
Cola sólo	<code>maxNumBackupConsumers</code>	“Mejora de las políticas de entrega para colas (Enterprise Edition)” en la página 9
Cola y tema	<code>consumerFlowLimit</code>	“Control del flujo mejorado del cliente Java” en la página 6 y “Mejora de las políticas de entrega para colas (Enterprise Edition)” en la página 9
Cola sólo	<code>localDeliveryPreferred</code>	“Mejora del rendimiento del clúster (Enterprise Edition)” en la página 10
Cola y tema	<code>isLocalOnly</code>	“Destinos locales (Enterprise Edition)” en la página 11

Tabla 7 Atributos de destino obsoletos de Message Queue 3.5

Tipo de destino	Nombre de atributo
Cola	QueueDeliveryPolicy

Atributos de la fábrica de conexiones En la siguiente tabla aparecen los atributos de fábrica de conexiones nuevos de Message Queue 3.5. Para obtener más información, consulte el capítulo 4 de *Message Queue Java Client Developer's Guide*. Tenga en cuenta que los atributos de fábrica de conexiones de Message Queue 3.0 siguen siendo compatibles con Message Queue 3.5 y se seguirán usando hasta la próxima versión principal del producto Message Queue.

Tabla 8 Nuevos atributos de fábrica de conexiones de Message Queue 3.5

Nombre de atributo	Referencia a la función
imqAddressList	“Recuperación ante fallos de conexión con clientes Java (Enterprise Edition)” en la página 4
imqAddressListBehavior	“Recuperación ante fallos de conexión con clientes Java (Enterprise Edition)” en la página 4
imqAddressListIterations	“Recuperación ante fallos de conexión con clientes Java (Enterprise Edition)” en la página 4
imqReconnectEnabled	“Recuperación ante fallos de conexión con clientes Java (Enterprise Edition)” en la página 4
imqReconnectAttempts	“Recuperación ante fallos de conexión con clientes Java (Enterprise Edition)” en la página 4
imqReconnectInterval	“Recuperación ante fallos de conexión con clientes Java (Enterprise Edition)” en la página 4
imqConsumerFlowLimit	“Control del flujo mejorado del cliente Java” en la página 6
imqConsumerFlowThreshold	“Control del flujo mejorado del cliente Java” en la página 6

Tabla 9 Atributos de la fábrica de conexiones con cambio de nombre

Nombre anterior	Nombre en Message Queue 3.5
imqFlowControlCount	imqConnectionFlowCount
imqFlowControlIsLimited	imqConnectionFlowLimitEnabled
imqFlowControlLimit	imqConnectionFlowLimit

Ubicación de los archivos públicos .jar

En la plataforma Solaris, la ubicación de los archivos públicos .jar se ha modificado en Message Queue 3.0.1 con respecto a la ubicación que tenían en Message Queue 3.0 (/usr/share/lib/imq/). La ubicación nueva es: /usr/share/lib/. Se han eliminado los enlaces simbólicos que se introdujeron en Message Queue 3.0.1 en /usr/share/lib/imq/ para los archivos .jar que se habían desplazado.

Esto se aplica a los siguientes archivos .jar:

- jms.jar
- imq.jar
- imqxm.jar
- activation.jar
- saaj-api.jar
- saaj-impl.jar
- mail.jar
- commons-logging.jar
- jaxm-api.jar
- fscontext.jar

Actualizaciones de la documentación

Este apartado describe los cambios efectuados en la documentación de Message Queue en las versiones 3.5 SP1 y 3.5.

Cambios en la versión 3.5 SP1

Los siguientes documentos de Message Queue 3.5 SP1 se han actualizado desde la versión 3.5 del producto:

Guía de instalación

Message Queue Installation Guide se ha actualizado para reflejar la información sobre los cambios de marca y de compatibilidad con la plataforma.

Guía de administración

Message Queue Administration Guide se ha cambiado de nombre (antes se llamaba *Message Queue Administrator's Guide*) para reflejar los cambios en la marca. Este documento también contiene un capítulo ampliado y actualizado donde se describe la supervisión del rendimiento.

Guía del desarrollador del cliente Java

Message Queue Java Client Developer's Guide se ha actualizado para reflejar los cambios de marca.

Guía del desarrollador del cliente C

Message Queue C Client Developer's Guide se ha actualizado para reflejar los cambios de marca.

Cambios en la versión 3.5

Los siguientes documentos de Message Queue 3.5 se han actualizado de la versión 3.0.1 del producto. Estos documentos actualizados se encuentran en el sitio Web de documentación de Message Queue 3.5: http://docs.sun.com/coll/S1_MessageQueue_35.

Guía de instalación

El producto Message Queue 3.5 incluye una actualización de *Message Queue Installation Guide*. En dicha guía se describen los nuevos requisitos de software, los cambios en las instrucciones de instalación de Solaris, el nuevo procedimiento de instalación en Linux usando Red Hat Package Manager (RPM) y la estructura de directorio instalada, así como otros cambios menores relacionados con la instalación en Windows.

Corrección: con respecto al soporte de la persistencia conectada, la tabla 1-2 especifica que Message Queue admite la versión 4.5 de PointBase. En realidad, la versión admitida de PointBase es la 4.8.

Guía del administrador

Message Queue Administration Guide se ha actualizado para incluir los cambios realizados en Message Queue 3.5 (consulte “Este apartado describe los cambios efectuados en Message Queue 3.5 SP1 y aquellos que se realizaron en la versión anterior, Message Queue 3.5.” en la página 2).

Guía del desarrollador del cliente Java

Message Queue Java Client Developer's Guide contiene la mayor parte del contenido de la guía anterior, *Message Queue Developer's Guide*, y se ha actualizado para incluir los cambios de Message Queue 3.5 (consulte “Este apartado describe los cambios efectuados en Message Queue 3.5 SP1 y aquellos que se realizaron en la versión anterior, Message Queue 3.5.” en la página 2).

Guía del desarrollador del cliente C

Message Queue C Client Developer's Guide es un documento nuevo que se ha agregado a la documentación de Message Queue en el que se describe la forma de crear aplicaciones de Message Queue con clientes en C.

Limitaciones y problemas conocidos

Este apartado describe las limitaciones y los problemas conocidos, así como los errores de Message Queue 3.5 SP1 Message Queue 3.5. Dado que la versión 3.5 SP1 es únicamente una versión en la que se incluyeron el cambio de marca de Message Queue 3.5, la información que aquí se presenta es aplicable a ambas versiones.

Para obtener una lista de los errores actuales, su estado y las soluciones, los miembros de Java Developer Connection™ deben consultar la página “Bug Parade” en el sitio Web de Java Developer Connection. Compruebe la página antes de informar de un nuevo error. No encontrará aquí todos los errores de Message Queue, se trata simplemente de un buen punto de partida para saber si se ha notificado un error.

La página en cuestión es:

<http://developer.java.sun.com/developer/bugParade>

NOTA	Para ser miembro de Java Developer Connection hay que registrarse, aunque es gratuito. Puede encontrar información detallada sobre cómo hacerse miembro de Java Developer Connection en la página Web “For Developers” de Sun.
-------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Para informar de un nuevo error o enviar una solicitud de función, envíe un mensaje de correo electrónico a mq-feedback@sun.com.

Problemas conocidos

En este apartado se describen los errores conocidos de Message Queue 3.5 SP1. Algunos de estos errores se introdujeron con la versión Message Queue 3.5. Las limitaciones que se muestran en este apartado están agrupadas según hagan referencia a Enterprise Edition y a Platform Edition de Message Queue o sólo a Enterprise Edition.

En Enterprise Edition y Platform Edition

- Debido al cambio de marca, las API que anteriormente devolvían la cadena:

"Sun ONE Message Queue, Sun Microsystems, Inc."

devuelven ahora la cadena:

"Sun Java(tm) System Message Queue"

- Debido al cambio de marca, los programas con clientes en C compilados con MQ 3.5 FCS que realizan una comparación exacta del valor asociado a `MQ_NAME_PROPERTY` desde `MQGetMetaData()` fallarán cuando se use la biblioteca compartida de 3.5 SP1 `mqcrt` en el tiempo de ejecución.
- Las plataformas Windows limitan el número de conexiones que se pueden iniciar simultáneamente por TCP/IP con un agente, en función del valor máximo del tamaño del registro de seguridad. El registro de seguridad es el búfer para las conexiones de la pila TCP; el número de inicios de conexiones TCP simultáneas no puede superar el tamaño de este registro. Por ejemplo, Windows 2000 Professional limita el registro de seguridad a 5 y Windows 2000 Server a 200.
- Si está ejecutando Windows XP, existe un límite para el número de conexiones *entrantes*. En el caso de Windows XP Professional, el número máximo de equipos que puede conectar simultáneamente mediante una red es de 10. Este límite incluye todos los protocolos compartidos de recursos y transportes combinados. En el caso de Windows XP Home Edition, el número máximo de equipos que puede conectar simultáneamente mediante una red es de 5. Esta limitación afectará al número de clientes que puede conectar al agente que se ejecuta en Windows XP.

Cualquier archivo, impresión, conducción con nombre o sesión de correo que no tenga ninguna actividad se desconecta automáticamente una vez que haya transcurrido el tiempo especificado en `AutoDisconnect`. El valor predeterminado para `AutoDisconnect` es de 15 minutos. Cuando se desconecta la sesión, 1 de cada 10 conexiones pasa a estado disponible para que otro usuario pueda conectarse al sistema Windows XP. En consecuencia, disminuir el valor de `AutoDisconnect` puede ayudar a reducir algunos de los problemas relacionados con el límite de 10 o 5 conexiones en un sistema que no se use demasiado para funciones de servidor. Para obtener más información, consulte:

<http://support.microsoft.com/default.aspx?scid=kb;EN-US;314882>

- No puede editar un archivo de configuración de la instancia del agente sin haber iniciado dicha instancia al menos una vez. Esto se debe a que el archivo `config.properties` no existe hasta que se inicia por primera vez la instancia del agente. Si desea configurar un agente para que utilice la persistencia de conexión o para definir otras propiedades de configuración, inicie el agente una vez (con el nombre de instancia que se debe utilizar para crear el agente) para crear el archivo `config.properties`:

Plataforma	Ubicación
Solaris	<code>/var/imq/instances/<i>nombre_instancia</i>/props/config.properties</code>
Linux	<code>/var/opt/imq/instances/<i>nombre_instancia</i>/props/config.properties</code>
Windows	<code>IMQ_VARHOME\instances\<i>nombre_instancia</i>\props\config.properties</code>

Una vez creado el archivo `config.properties`, edítelo para añadir todos los valores de propiedad de configuración y, a continuación, reinicie el agente.

Sólo en Enterprise Edition

- Sólo se admiten en esta versión los clústeres de agentes totalmente conectados. Esto significa que todos los agentes de un clúster deben establecer comunicación directamente con todos los demás agentes del clúster. Si conecta agentes mediante el argumento de línea de comandos `imqbrokerd -cluster`, asegúrese de incluir todos los agentes del clúster.
- Si no se utiliza un agente maestro en un clúster, la información persistente almacenada por un agente que se ha añadido al clúster no se distribuirá a los demás agentes del clúster.
- El servicio de conexión mediante SSL sólo admite actualmente certificados de servidor firmados automáticamente, es decir, en el modo de host de confianza.
- Cuando un cliente JMS con transporte HTTP finaliza repentinamente (por ejemplo, mediante `Ctrl-C`), el agente espera aproximadamente un minuto antes de liberar la conexión de cliente y todos los recursos asociados.

Si se inicia otra instancia de cliente durante ese período de un minuto y si ésta intenta utilizar el mismo `ClientID`, la misma suscripción duradera o la misma cola, recibirá una excepción que indica que el ID de cliente está todavía en uso. No se trata de un problema real, es simplemente un efecto secundario del proceso de finalización descrito anteriormente. Si el cliente se inicia después de un retraso de, aproximadamente, un minuto, todo debería funcionar correctamente.

Errores conocidos

En la [Tabla 10](#) se enumeran los errores más importantes de Message Queue 3.5 SP1.

Tabla 10 Errores conocidos de Message Queue 3.5

Número de error	Detalles
4683029	<p>La opción <code>-javahome</code> de todas las secuencias de comandos de Solaris/Win no funciona si el valor incluye un espacio en blanco.</p> <p>Los comandos y utilidades de Message Queue utilizan la opción <code>-javahome</code> para especificar un tiempo de ejecución alternativo, compatible con Java 2. Sin embargo, la ruta al tiempo de ejecución Java alternativo no debe contener espacios en blanco.</p> <p>Ejemplos de rutas que contienen espacios en blanco:</p> <p>Windows:</p> <p><code>C:\jdk 1.4</code> (En Windows, la ruta puede contener espacios en blanco si se usan comillas, como por ejemplo, <code>"C:\jdk 1.4"</code>)</p> <p>Solaris:</p> <p><code>/work/java 1.4</code></p> <p>Solución del problema: instale el tiempo de ejecución Java en una ubicación o ruta que no contenga espacios en blanco.</p>
4939923	<p>El agente puede generar una excepción <code>NullPointerException</code> cuando se usa un grupo de subprocesos compartidos y la JVM del agente se está ejecutando con poca memoria.</p> <p>Solución del problema: ninguna. Este error está solucionado en J2SE 1.4.2_03.</p>
4941058	<p>Los destinos en los que está activado el control de flujo puede que no alcancen el límite máximo. Hay situaciones en las que a los productores se les impide que envíen mensajes a un destino antes de que dicho destino alcance el límite máximo configurado.</p> <p>Solución del problema: ninguna.</p>
4941066	<p>Los destinos pueden superar levemente el límite de bytes asignado.</p> <p>Solución del problema: ninguna.</p>
4941127	<p>El destino no se cargará por completo si el mensaje supera el límite del tamaño del mensaje individual. Si el límite de tamaño de los mensajes permitidos en un destino se cambia después de almacenar un mensaje grande, el destino no se cargará correctamente.</p> <p>Solución del problema: aumente el límite de tamaño de los mensajes hasta que se consuma el mensaje grande y, a continuación, establezca un límite inferior. Se podría pausar la producción del destino durante este tiempo para impedir que se acepten otros mensajes grandes.</p>
4946531	<p>La excepción inofensiva <code>NullPointerException</code> se puede producir en raras ocasiones al crear mensajes.</p> <p>Solución del problema: ninguna. La excepción de puntero nulo se puede omitir sin problemas.</p>

Tabla 10 Errores conocidos de Message Queue 3.5 (*continuación*)

Número de error	Detalles
4949398	<p><code>imqcmd query dst</code> registra valores incorrectos en cuanto al número de mensajes y al total de bytes del mensaje cuando se carga el destino correspondiente. Los valores registrados son correctos antes de cargar el destino y después de hacerlo.</p> <p>Solución del problema: el problema sólo se produce mientras se carga el destino. Una vez que está cargado el destino, los valores que se devuelven son correctos.</p>
4950166	<p>Errores aleatorios en el agente cuando se ejecuta en sistemas <code>jdk1.4.2_02</code> y <code>x86</code>. Para obtener más información, consulte J2SE, error número 4947404.</p> <p>Solución del problema: inicie el agente con <code>-XX:UseSSE=0</code>, por ejemplo</p> <pre>imqbrokerd -tty -vmargs -XX:UseSSE=0</pre>
4950601	<p><code>imqcmd metrics dst</code> genera un error interno del agente cuando se usa el almacén persistente JDBC.</p> <p>La información sobre las métricas de uso del espacio se aplica sólo al almacén de archivos. No obstante, cuando se recupera la información sobre las métricas, el agente intentará obtener la información sobre el uso del disco, independientemente del tipo de almacén. Si se usa una base de datos en lugar de un almacén de archivos, el agente mostrará el siguiente mensaje de error:</p> <pre>06/Nov/2003:22:57:36 PST] ERROR [B3100]: Unexpected Broker Internal Error : [unable to disk usage for destinationT:topic1] : com.sun.messaging.jmq.jmsserver.util.BrokerException: La operación no se aplica al almacén persistente conectado.</pre> <p>Solución del problema: ninguna.</p>
4951010	<p>En un clúster de agentes, un agente puede poner en cola mensajes para una conexión remota que es posible que no esté iniciada.</p> <p>Solución del problema: el consumidor recibirá los mensajes una vez que se inicie la conexión. Los mensajes se enviarán a otro usuario si la conexión del consumidor está cerrada.</p>
4953348	<p>En ocasiones, <code>HTTPS createQueueConnection</code> genera excepciones en Windows 2000.</p> <p>Solución del problema: intente establecer una nueva conexión.</p>
4953354	<p>El agente pasa a estar inaccesible cuando un almacén persistente abre demasiados destinos.</p> <p>Solución del problema: este hecho está provocado por el agente que alcanza el límite del descriptor de archivos abiertos del sistema. En Solaris y en Linux use el comando <code>ulimit</code> para aumentar el límite del descriptor de archivos.</p>
4954974	<p>La instalación no se inicia automáticamente al usar el CD de soporte en Windows XP.</p> <p>Solución del problema: en el explorador de Windows, haga doble clic en la carpeta del CD y, después, vuelva a hacer clic en el archivo <code>imq3_5-ent-win.exe</code> para ejecutar el instalador.</p>
4983525	<p>En Linux Red Hat Advanced Server 3.0, es posible que falle la creación de un productor de mensajes para un destino creado automáticamente.</p> <p>Solución del problema: vuelva a crear el productor. Puede que al segundo intento, lo logre. De lo contrario, use un destino creado administrativamente.</p>

Tabla 10 Errores conocidos de Message Queue 3.5 (continuación)

Número de error	Detalles
4986318	<p>El cliente puede generar de forma inesperada un mensaje ACKNOWLEDGE_REPLY:</p> <pre>***** Packet: ACKNOWLEDGE_REPLY(25):26-192.18.86.227-42976-1075458056557 Magic/Version: 469754818/301Size: 97 Type: ACKNOWLEDGE_REPLY(25) Expiration: 0 Timestamp: 1075458056557 Source IP: 192.18.86.227 Source Port: 42976Sequence: 26</pre> <p>Solución del problema: ninguna. Se trata de una condición de temporización extraña en el agente que provoca la generación de este error por parte del cliente. Este error se puede omitir. No se perderá ningún mensaje.</p>
4991257	<p>El envío de mensajes persistentes extensos a suscriptores duraderos de un clúster de agentes en los que el almacén persistente está basado en JDBC puede provocar que el agente se bloquee o que genere errores.</p> <p>Solución del problema: aumente el tiempo de espera del protocolo de bloqueo del agente mediante la siguiente propiedad:</p> <pre>imq.cluster.timeout=<tiempo_espera_en_segundos></pre> <p>El valor predeterminado es 60. Si los mensajes persistentes extensos son lentos, puede que deba ajustar la base de datos del almacén persistente o cambiar a otro almacén distinto.</p>
5006686	<p>El ejemplo ARGS de imqbrokerd.conf no es correcto.</p> <p>Solución del problema: los valores no se deben utilizar tal y como aparecen en el ejemplo:</p> <pre>ARGS="-name newbroker -port 8888"</pre> <p>Los valores deben ser:</p> <pre>ARGS=-name newbroker -port 8888</pre>

Archivos que se pueden distribuir

Sun Java System Message Queue 3.5 SP1 contiene el siguiente grupo de archivos que usted puede distribuir en formato binario:

```
jms.jar  
imq.jar  
imqxm.jar  
fscontext.jar  
providerutil.jar  
jndi.jar  
ldap.jar  
ldapbp.jar  
jaas.jar  
jsse.jar  
jnet.jar  
jcert.jar
```

Además, también puede distribuir los archivos de LICENCIA y COPYRIGHT.

Información sobre problemas y respuestas de los clientes

Para informar de un problema, envíe un mensaje de correo electrónico a imq-feedback@sun.com.

Si dispone de un contrato de asistencia técnica y tiene problemas con Message Queue, póngase en contacto con el servicio de asistencia al cliente utilizando uno de los siguientes sistemas:

- Servicios de atención al cliente de Sun Software en línea en <http://www.sun.com/service/sunone/software>

Este sitio dispone de enlaces a la base de datos de soluciones, al centro de asistencia en línea y al rastreador de productos, así como a programas de mantenimiento y números de contacto de asistencia.

- El número de teléfono del distribuidor asociado al contrato de mantenimiento.

Para que podamos ayudarle de forma óptima en la resolución de problemas, tenga a mano la siguiente información cuando se ponga en contacto con el servicio de asistencia:

- Descripción del problema, incluida la situación en la que éste se produce y la forma en que afecta al funcionamiento
- Tipo de máquina, versión del sistema operativo y versión del producto, incluida cualquier revisión del producto y otro software que pudiera influir en el problema
- Pasos detallados de los métodos que haya usado para reproducir el problema
- Cualquier registro de errores o volcados del núcleo

Sun valora sus comentarios

Sun tiene interés en mejorar su documentación y valora sus comentarios y sugerencias. Para enviar sus comentarios a Sun, use el formulario basado en Web que aparece en:

<http://www.sun.com/hwdocs/feedback/>

Indíquenos el título completo de la documentación y el número de pieza en los campos pertinentes. Puede hallar el número de pieza en la página del título del libro o en la parte superior del documento. Normalmente, es un número que consta de siete o nueve dígitos. Por ejemplo, el número de pieza de este documento, *Message Queue 3.5 SP1 Notas de la versión*, es 817-7199-10.

Recursos adicionales de Sun

Además de la documentación de Message Queue, puede encontrar información adicional tal y como se indica a continuación.

Foros de discusión

Sun Java System Foros sobre software

Puede encontrar un foro de Sun Java System Message Queue en la siguiente dirección:

<http://softwareforum.sun.com/NASApp/jive/forum.jsp?forum=24>

Le agradecemos su participación.

Foro de tecnología de Java

Existe un foro de JMS en los foros de tecnología de Java que puede interesarle.

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

Base de datos de soluciones de SunSolve

Hay disponible información en línea acerca de Sun Java System Message Queue en la base de datos de soluciones de SunSolve, que está ubicada en:

<http://sunsolve.Sun.COM/pub-cgi/search.pl?mode=advanced>

Seleccione el enlace que le permite acceder a la documentación gratuita: se trata de “All Free Collections”. Después debe buscar “Message Queue”.

Sun Java System Información

Puede encontrar información útil de Sun Java System en las siguientes direcciones de Internet:

- Página del producto Message Queue
http://www.sun.com/software/products/message_queue/index.html
- Documentación para Message Queue
http://docs.sun.com/coll/MessageQueue_35_SP1
- Documentación sobre Sun
<http://docs.sun.com/>
- Servicios y productos de software de Sun Java System
<http://www.sun.com/software>
- Servicios de asistencia al cliente de software de Sun
<http://www.sun.com/service/sunone/software>
- Base de datos de soluciones y asistencia al cliente de Sun
<http://sunsolve.sun.com>
- Servicios de formación y asistencia al cliente de Sun
<http://www.sun.com/supporttraining>
- Información para desarrolladores de Sun
<http://developers.sun.com/>
- Servicios de asistencia de desarrolladores de Sun
<http://www.sun.com/developers/support>
- Hojas de datos de software de Sun
<http://www.sun.com/software>

Copyright © 2004 Sun Microsystems, Inc. Todos los derechos reservados.

Derechos del gobierno de Estados Unidos: Software comercial. Los usuarios gubernamentales están sujetos al acuerdo de licencia estándar de Sun Microsystems, Inc. y a las disposiciones aplicables de la regulación FAR y sus suplementos. El uso está sujeto a las condiciones de la licencia. Esta distribución puede incluir materiales desarrollados por terceras partes.

Sun, Sun Microsystems, el logotipo de Sun, Java, Solaris y Sun[tm] ONE son marcas comerciales o marcas comerciales registradas de Sun Microsystems, Inc. en Estados Unidos y otros países.

Todas las marcas comerciales de SPARC se utilizan bajo licencia y son marcas comerciales o marcas registradas de SPARC International, Inc. en EE.UU. y en otros países.

UNIX es una marca comercial registrada en EE.UU. y en otros países, cuya licencia se otorga exclusivamente a través de X/Open Company, Ltd.

