

[MSDN Library](#) > [SQL Server](#) > [Documentación de SQL Server 2005](#) > [Libros en pantalla de SQL Server](#) > [SQL Server Service Broker](#) > [Enrutamiento y redes de Service Broker](#) > Protocolos de comunicación de Service Broker
 [Collapse All](#)

Protocolos de comunicación de Service Broker

Service Broker utiliza un protocolo específico del broker para comunicarse con los brokers remotos. El broker administra las conexiones independientemente del grupo normal de conexiones de cliente. Para que dos instancias de SQL Server intercambien mensajes de Service Broker, cada instancia debe poder enviar tráfico TCP/IP al puerto que utiliza la otra instancia para las comunicaciones de Service Broker. Por convención, Service Broker suele utilizar el puerto 4022 para la comunicación entre brokers. Sin embargo, el puerto exacto se especifica cuando se crea el extremo.

[Capas del protocolo](#)

Service Broker establece un método por capas para la comunicación. Cada capa se genera sobre la capa subyacente para asegurar una entrega confiable. Este método permite a una aplicación operar sin saber la ubicación del servicio remoto o el transporte físico que utiliza el broker para comunicarse. En la mayoría de los casos, estos protocolos resultan transparentes para una aplicación. Sin embargo, resulta útil conocer el papel que desempeña cada capa del protocolo para solucionar los problemas con una aplicación.

El protocolo del nivel más alto que utiliza el broker es el *protocolo de diálogo*. La capa del protocolo de diálogo controla la transmisión de mensajes confiable y secuencial. Esta capa genera números de secuencia para los mensajes, genera mensajes de reconocimiento, entrega mensajes a las colas adecuadas y fragmenta y vuelve a componer los mensajes. El protocolo de diálogo controla la autenticación y el cifrado de un diálogo.

El protocolo de diálogo utiliza el *protocolo del broker adyacente* para transmitir fragmentos de mensaje. El protocolo del broker adyacente controla las transmisiones de red intercambiadas entre dos instancias de broker.

El protocolo del broker adyacente utiliza un *protocolo de transporte*, como TCP/IP, para mover mensajes de broker a broker.

Protocolo de diálogo

El protocolo de diálogo aplica el patrón de entrega exactamente una vez por orden (EOIO) en los mensajes de una conversación. Este protocolo no describe el formato que utilizan los mensajes de Service Broker en la red. En su lugar, especifica los pasos lógicos necesarios para una conversación confiable. El protocolo de diálogo controla las tareas necesarias para la entrega confiable, incluida la generación y el procesamiento de mensajes de reconocimiento.

Cada extremo de una conversación es un extremo en la capa del protocolo de diálogo. La vista de catálogo sys.conversation_endpoints muestra información sobre los extremos del protocolo de diálogo. Un extremo de una conversación existe mientras existe la conversación.

Protocolo del broker adyacente

El protocolo del broker adyacente controla los mecanismos de comunicación entre dos instancias de SQL Server. Esta capa codifica cada fragmento de mensaje en un formato estándar adecuado para la transmisión a través de la red. A diferencia de la capa del protocolo de diálogo, la capa del protocolo adyacente reconoce el transporte de red utilizado y da formato a los fragmentos del mensaje de forma adecuada. De hecho, la capa del protocolo del broker adyacente proporciona una capa de abstracción entre la capa del protocolo de diálogo y la capa del protocolo de transporte.

Cada conexión de red de Service Broker es un extremo de la capa del protocolo adyacente. La vista de administración dinámica `sys.dm_broker_connections` muestra información sobre las conexiones de red de Service Broker. Service Broker mantiene la conexión de red mientras que los mensajes se intercambian de forma activa. Cuando no se han enviado ni recibido mensajes a través de la conexión de red durante un período de tiempo breve, la cierra.

Protocolo de transporte

La capa del protocolo de transporte controla la transmisión de red real. Esta capa está fuera de Service Broker. Por ejemplo, los mensajes a un broker que se ejecuta en una instancia distinta de SQL Server utilizan TCP/IP como capa del protocolo de transporte.

Los extremos de Service Broker establecen opciones para el protocolo de transporte. De forma predeterminada, SQL Server no contiene extremos de Service Broker. Para obtener más información sobre cómo crear un extremo de Service Broker, vea [Cómo activar la comunicación en red de Service Broker \(Transact-SQL\)](#).

[Comunicación de Service Broker](#)

Service Broker utiliza dos categorías de mensaje distintas. Un *mensaje en secuencia* es un mensaje que debe entregarse a una aplicación exactamente una vez, en orden. Un *mensaje sin secuencia* es un mensaje que puede procesarse de forma inmediata, independientemente de la secuencia en la que llega el mensaje.

Service Broker utiliza mensajes en secuencia para todos los tipos de mensajes definidos por el usuario, los mensajes de fin de diálogo y los mensajes de error creados por una aplicación. Cada mensaje en secuencia tiene un número de secuencia. La instancia que origina el mensaje crea el número de secuencia de mensaje y lo asigna al mensaje. El broker receptor utiliza el número de secuencia de mensaje para ordenar los mensajes que proporciona a la aplicación. Para un diálogo dado, la aplicación siempre recibe en primer lugar el mensaje con el número de secuencia más bajo. Service Broker también utiliza el número de secuencia de mensaje para detectar mensajes duplicados. Cuando una capa del protocolo de diálogo recibe dos mensajes en el mismo diálogo con el mismo número de secuencia, considera que los mensajes están duplicados y descarta uno.

Service Broker utiliza mensajes sin secuencia en los mensajes de reconocimiento dedicados y los mensajes de error creados por Service Broker. Service Broker no toma precauciones especiales para entregar un mensaje sin secuencia. Sin embargo, tenga en cuenta que Service Broker crea mensajes sin secuencia como respuesta a mensajes entrantes. Por tanto, si el mensaje sin

secuencia se pierde, el remitente volverá a intentar enviar el mensaje original; a continuación, el destinatario generará otro mensaje sin secuencia.

Fragmentación de mensajes

Service Broker divide los mensajes salientes en fragmentos y combina los fragmentos entrantes en el mensaje original. En el caso de mensajes pequeños, el mensaje completo se incluye en un fragmento. En los mensajes grandes, Service Broker crea muchos fragmentos.

La fragmentación de mensajes tiene varias ventajas. El envío de un mensaje grande en pequeños fragmentos mejora la velocidad y la confiabilidad globales en la comunicación a través de redes relativamente lentas y poco confiables como las redes de área extensa (WAN). En caso de que se pierda algún fragmento del mensaje, el protocolo vuelve a transmitir sólo un fragmento en lugar del mensaje completo. La fragmentación de los mensajes grandes también reduce la cantidad de tiempo necesario para que un mensaje pequeño llegue a su destino. Service Broker puede enviar un fragmento que contiene un mensaje pequeño completo entre los fragmentos de un mensaje grande. Esto ralentiza ligeramente el mensaje grande para reducir la cantidad de tiempo que el mensaje pequeño espera para ser transmitido.

Mientras un mensaje está en proceso de volver a componerse, el mensaje parcial se almacena en la cola de destino, o en la cola de transmisión de la base de datos que aloja la cola de destino si ésta no está disponible. Una aplicación no puede recibir un mensaje parcial. La columna de estado de un mensaje parcial se establece en 2 (deshabilitada). Este valor también se utiliza para los mensajes recibidos sin orden.

Reconocimiento de mensajes

Service Broker reconoce cada mensaje recibido. Un reconocimiento puede reconocer uno o varios fragmentos de mensajes. Si es posible, se incluye un reconocimiento en el encabezado de un mensaje devuelto en la misma conversación. Si no hay ningún otro mensaje listo para enviar, Service Broker devuelve un mensaje de reconocimiento dedicado. El reconocimiento de mensajes está controlado íntegramente por Service Broker; las aplicaciones que utilizan Service Broker no reciben estos mensajes.

Un remitente retiene los fragmentos del mensaje que el destinatario no ha reconocido. Si no se recibe ningún reconocimiento en un tiempo de espera definido por el sistema, el remitente vuelve a enviar el fragmento del mensaje. Si no se recibe ningún reconocimiento en el tiempo de espera, Service Broker incrementa de forma exponencial la cantidad de tiempo antes del siguiente reintento, hasta un tiempo de espera máximo. El tiempo de espera inicial para un reintento es de algunos segundos. El tiempo de espera máximo es de aproximadamente un minuto. Observe que no está previsto que el tiempo de espera sea preciso; dependiendo del tráfico de la red y de las demás actividades en la instancia de SQL Server, puede que se vuelva a intentar enviar el fragmento de mensaje unos segundos después de que finalice el tiempo de espera.

Si un reconocimiento se pierde o se retrasa, el destinatario puede recibir mensajes duplicados. En este caso, el destinatario confirma la recepción del mensaje duplicado pero no entrega el mensaje duplicado a la cola.

Service Broker utiliza el reconocimiento de mensajes para proporcionar mensajería confiable sin transacciones distribuidas. Un destinatario envía un reconocimiento sólo después de agregar el mensaje o el fragmento de mensaje a la cola. El remitente aloja el mensaje en la cola de transmisión hasta que llega su reconocimiento. Aunque el remitente y el destinatario nunca comparten una transacción, el protocolo garantiza que el remitente no quita el mensaje de la cola de transmisión hasta que el destinatario ha recibido el mensaje correctamente.

Comprobación de integridad del mensaje

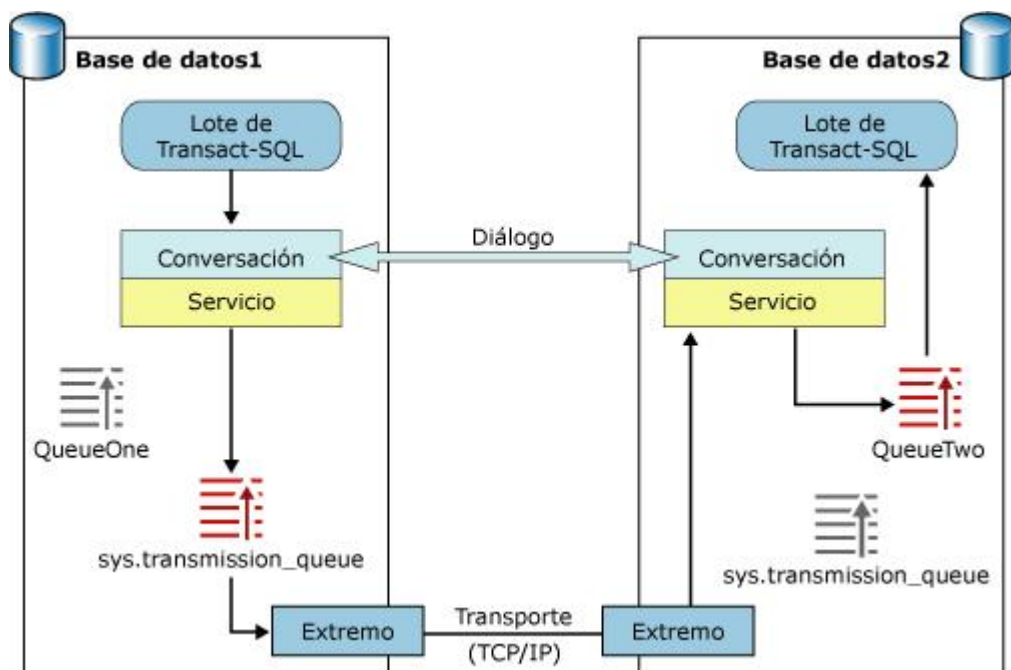
El formato que utiliza Service Broker para transmitir mensajes incluye una comprobación de integridad del mensaje para determinar si un mensaje dado se ha modificado o dañado durante el transporte.

La comprobación de integridad del mensaje es una firma MD5 del contenido del mensaje. SQL Server cifra la firma con la clave de sesión para el mensaje e la incluye en los encabezados del mensaje.

El destino del mensaje descifra el mensaje y, a continuación, compara la firma del mensaje con una nueva firma calculada en función del contenido real recibido. Si las firmas no coinciden, el mensaje se habrá dañado o modificado durante la transmisión. El mensaje genera un error de comprobación de integridad de mensajes. SQL Server descarta el mensaje y no envía un reconocimiento del mensaje al remitente. La clase de evento Broker:Corrupted Message proporciona información cuando un mensaje genera un error en la comprobación de integridad de mensajes.

Flujo de comunicación de red

En la siguiente ilustración se presenta una vista general de la comunicación de red de Service Broker entre dos instancias de SQL Server.



Observe que la conversación es una conexión lógica y persistente. La conversación puede tener lugar a lo largo de un período de tiempo; durante ese período de tiempo, la conversación puede utilizar cualquier número de conexiones de red.

Las conexiones de red se producen entre dos extremos de Service Broker. Estas conexiones utilizan TCP/IP. Si la conexión está inactiva durante un período de tiempo breve, SQL Server cierra la conexión de red.

Para entregar un mensaje, Service Broker aloja el mensaje en la cola de transmisión de la base de datos que ha enviado el mensaje. El destinatario entrega el mensaje directamente a la cola para el servicio de destino. Si esa cola está desactivada, el mensaje se mantiene temporalmente en la cola de transmisión de la base de datos de recepción. Observe que la cola del servicio remitente no está implicada en la operación y que la cola de transmisión de la base de datos que aloja el servicio receptor sólo está implicada si la cola de destino está desactivada.

 [Vea también](#)

Conceptos

[Enrutamiento de Service Broker](#)

Ayuda e información

[Obtener ayuda sobre SQL Server 2005](#)