



IBM Systems - iSeries  
Base de datos  
Control de compromiso

*Versión 5 Release 4*







IBM Systems - iSeries  
Base de datos  
Control de compromiso

*Versión 5 Release 4*

**Nota**

Antes de utilizar esta información y el producto al que hace referencia, lea la información que figura en: "Avisos", en la página 119.

**Cuarta edición (febrero de 2006)**

Esta edición atañe a la versión 5, release 4, modificación 0 de IBM i5/OS (producto número 5722-SS1) y a todos los releases y modificaciones ulteriores hasta que se indique lo contrario en nuevas ediciones. Esta versión no se ejecuta en todos los modelos de sistema con conjunto reducido de instrucciones (RISC) ni tampoco se ejecuta en los modelos CISC.

© Copyright International Business Machines Corporation 2003, 2006. Reservados todos los derechos.

---

# Contenido

<b>Control de compromiso</b> . . . . .	<b>1</b>
Novedades de la V5R4 . . . . .	1
Archivo PDF imprimible . . . . .	1
Conceptos sobre el control de compromiso . . . . .	2
Cómo funciona el control de compromiso. . . . .	2
Cómo funcionan las operaciones de compromiso y retrotracción . . . . .	3
Definición de compromiso. . . . .	5
Cómo funciona el control de compromiso con los objetos . . . . .	12
Control de compromiso y agrupaciones de discos independientes . . . . .	22
Consideraciones y restricciones sobre el control de compromiso . . . . .	25
Control de compromiso en las aplicaciones por lotes. . . . .	27
Control de compromiso de dos fases . . . . .	27
Soporte de transacciones XA para el control de compromiso . . . . .	45
Modalidad de servidor SQL y transacciones con ámbito de hebra para el control de compromiso . . . . .	50
Iniciar el control de compromiso . . . . .	51
Objeto de notificación de compromiso . . . . .	52
Nivel de bloqueo de compromiso . . . . .	54
Finalizar control de compromiso . . . . .	57
Finalización iniciada por sistema del control de compromiso . . . . .	58
Control de compromiso durante la finalización de un grupo de activación . . . . .	58
Operaciones de compromiso y retrotracción implícitas . . . . .	59
Control de compromiso durante la finalización normal de un paso de direccionamiento . . . . .	63
Control de compromiso durante la finalización anómala del sistema o del trabajo . . . . .	63
Actualizaciones del objeto de notificación . . . . .	65

Recuperación de control de compromiso durante la carga del programa inicial tras una finalización anómala . . . . .	67
Gestionar transacciones y control de compromiso. . . . .	68
Visualizar información de control de compromiso . . . . .	68
Optimizar el rendimiento para el control de compromiso . . . . .	70
Casos prácticos y ejemplos: control de compromiso . . . . .	76
Caso práctico: control de compromiso . . . . .	76
Problema de práctica de control de compromiso . . . . .	79
Ejemplo: utilizar un archivo de anotaciones de transacciones para iniciar una aplicación. . . . .	88
Ejemplo: utilizar un objeto de notificación para iniciar una aplicación . . . . .	92
Ejemplo: utilizar un programa de proceso estándar para iniciar una aplicación . . . . .	98
Resolución de problemas relacionados con las transacciones y el control de compromiso . . . . .	105
Errores de control de compromiso . . . . .	105
Detectar puntos muertos. . . . .	112
Recuperar transacciones tras una anomalía de comunicaciones. . . . .	113
Cuándo forzar operaciones de compromiso y retrotracción y cuándo cancelar la resincronización . . . . .	114
Finalizar una retrotracción de larga duración . . . . .	116
Información relacionada con el control de compromiso . . . . .	117
Información de licencia de código y declaración de limitación de responsabilidad . . . . .	118

<b>Apéndice. Avisos</b> . . . . .	<b>119</b>
Información de la interfaz de programación . . . . .	121
Marcas registradas. . . . .	121
Términos y condiciones . . . . .	121



---

## Control de compromiso

El control de compromiso es una función que garantiza la integridad de los datos. Permite definir y procesar un grupo de cambios que se realizan en recursos (por ejemplo, archivos o tablas de base de datos) como una transacción.

El control de compromiso garantiza que o bien se efectúa todo el grupo de cambios individuales en todos los sistemas participantes o bien no se realiza ninguno de los cambios. IBM DB2 Universal Database para iSeries utiliza la función de control de compromiso para comprometer y retrotraer las transacciones de base de datos que se ejecutan con un nivel de aislamiento distinto de \*NONE (Ningún compromiso).

Puede emplear el control de compromiso para diseñar una aplicación de tal manera que el sistema pueda reiniciarla si un trabajo, un grupo de activación de un trabajo o el sistema finaliza de forma anómala. Con el control de compromiso, puede estar seguro de que cuando la aplicación vuelva a iniciarse, no habrá ninguna actualización parcial en la base de datos debido a transacciones incompletas como consecuencia de una anomalía anterior.

| **Nota:** Si utiliza los ejemplos de código, indica que acepta los términos que figuran en: “Información de  
| licencia de código y declaración de limitación de responsabilidad” en la página 118.

---

## Novedades de la V5R4



En este tema se resaltan los cambios realizados en el control de compromiso en la V4R4 de IBM i5/OS.

### | **Novedades del control de compromiso:**

- | • Mejoras realizadas en la funcionalidad XA (encontrará los detalles en Soporte de transacciones XA para el control de compromiso):
  - | – Soporte de base de datos relacional (RDB) remota para transacciones XA que tienen bloqueos con ámbito de transacción
  - | – Compartimiento de bloqueos opcional para ramas de transacciones débilmente acopladas
  - | – Elección de hebra de control XA
  - | – Soporte de acceso a base de datos tradicional dentro de transacciones XA mediante la gestión de datos distribuidos (DDM)
- | • Compromiso leve

## Cómo se sabe lo que es nuevo o lo que ha cambiado

Para que le resulte más fácil ver dónde se han hecho cambios técnicos, se utiliza:

- La imagen de  para marcar dónde empieza la información nueva y la que ha cambiado.
- La imagen de  para marcar dónde acaba la información nueva y la que ha cambiado.

Para obtener más información sobre las novedades o los cambios de este release, vea el Memorándum para los usuarios.

---

## Archivo PDF imprimible

Utilice esta función para ver e imprimir un PDF de esta información.


Para ver o descargar la versión PDF de este documento, seleccione Control de compromiso (alrededor de 1413 KB).

## Cómo guardar los archivos PDF

Si desea guardar un archivo PDF en su estación de trabajo para verlo o imprimirlo:

1. Pulse el PDF con el botón derecho del ratón en el navegador (pulse el enlace anterior con el botón derecho del ratón).
2. Pulse la opción que guarda el PDF localmente.
3. Navegue hasta el directorio en el que desea guardar el PDF.
4. Pulse **Guardar**.

## Cómo descargar Adobe Reader

1. Para poder ver o imprimir archivos PDF, debe instalar Adobe Reader en su sistema. Puede descargar una copia gratuita desde el sitio Web de Adobe ([www.adobe.com/products/acrobat/readstep.html](http://www.adobe.com/products/acrobat/readstep.html)) .

---

## Conceptos sobre el control de compromiso

En este tema se proporciona información que pretende ayudarle a comprender cómo funciona el control de compromiso, cómo interacciona con el sistema y cómo interacciona con otros sistemas de la red.

### Cómo funciona el control de compromiso

El control de compromiso es una función que permite definir y procesar un grupo de cambios que se realizan en recursos (por ejemplo, archivos o tablas de base de datos) como una transacción.

El control de compromiso garantiza que o bien se efectúa todo el grupo de cambios individuales en todos los sistemas participantes o bien no se realiza ninguno de los cambios. Por ejemplo, al transferir fondos de una cuenta de ahorros a una cuenta corriente, se efectúa más de un cambio como un grupo. A la vista del usuario, esta transferencia aparece como un único cambio. Sin embargo, se realiza más de un cambio en la base de datos ya que se actualiza tanto la cuenta de ahorros como la cuenta corriente. Para que ambas cuentas permanezcan exactas, o bien deben efectuarse todos los cambios o bien no debe efectuarse ninguno de los cambios en la cuenta de ahorros y la cuenta corriente.

El control de compromiso permite:

- Asegurarse de que se efectúan todos los cambios de una transacción para todos los recursos afectados.
- Asegurarse de que se eliminan todos los cambios de una transacción si se interrumpe el proceso.
- Eliminar los cambios efectuados durante una transacción cuando la aplicación determina que una transacción es anómala.

También puede diseñar una aplicación de modo que el control de compromiso pueda reiniciar la aplicación si un trabajo, un grupo de activación de un trabajo o el sistema finaliza de forma anómala. Con el control de compromiso, puede estar seguro de que cuando la aplicación vuelva a iniciarse, no habrá ninguna actualización parcial en la base de datos debido a transacciones incompletas como consecuencia de una anomalía anterior.

### Transacción

Una transacción es un grupo de cambios individuales en objetos del sistema que se muestran al usuario como un solo cambio atómico.

**Nota:** En iSeries Navigator se utiliza el término *transacción*, mientras que en la interfaz basada en caracteres se utiliza el término *unidad de trabajo lógica (LUW)*. Ambos términos son intercambiables. En este tema, salvo cuando se refiera específicamente a la interfaz basada en caracteres, se emplea el término transacción.



Una transacción puede representar cualquiera de estas situaciones:

- Consultas en las que no se produce ningún cambio de archivo de base de datos.
- Transacciones simples que cambian un archivo de base de datos.
- Transacciones complejas que cambian uno o varios archivos de base de datos.
- Transacciones complejas que cambian uno o varios archivos de base de datos, pero estos cambios representan únicamente una parte de un grupo lógico de transacciones.
- Transacciones simples o complejas en las que intervienen archivos de base de datos de más de una ubicación. Los archivos de base de datos pueden estar:
  - En un único sistema remoto.
  - En el sistema local y uno o varios sistemas remotos.
  - Asignados a más de un diario del sistema local. Cada diario puede describirse de forma general como una *ubicación local*.
- Transacciones simples o complejas en el sistema local en las que intervienen objetos distintos de archivos de base de datos.

## Cómo funcionan las operaciones de compromiso y retroacción

Las operaciones de compromiso y retroacción afectan a los cambios que se realicen bajo control de compromiso.

Los lenguajes de programación y las interfaces de programación de aplicaciones (API) que soportan las operaciones de compromiso y retroacción son los siguientes.

Lenguaje o API	Compromiso	Retroacción
CL	Mandato COMMIT	Mandato ROLLBACK
IBM Integrated Language Environment (ILE) RPG/400	Código de operación COMIT	Código de operación ROLBK
ILE COBOL/400	Verbo COMMIT	Verbo ROLLBACK
ILE C/400	Función _Rcommit	Función _Rrollback
PL/I	Subrutina PLICOMMIT	Subrutina PLIROLLBACK
SQL	Sentencia COMMIT	Sentencia ROLLBACK
Interfaz a nivel de llamada (CLI) SQL	Función SQLTransact() (se utiliza para comprometer y retrotraer una transacción)	
Interfaces API XA	Las API xa_commit() y db2xa_commit()	Las API xa_rollback() y db2xa_rollback()

### Conceptos relacionados

Interfaz a nivel de llamada (CLI) SQL

Programación de bases de datos

### Información relacionada

COBOL/400 Guía del usuario PDF

RPG/400 Guía del usuario PDF

WebSphere Development Studio: ILE C/C++ Guía del programador PDF

Programación CL

Interfaces de programación de aplicaciones (API)

## Operación de compromiso

Una operación de compromiso establece como permanentes todos los cambios efectuados bajo control de compromiso desde la operación de compromiso o retroacción anterior. El sistema también libera todos los bloqueos relacionados con la transacción.

El sistema lleva a cabo los pasos siguientes cuando recibe una petición de compromiso:

- El sistema salva la identificación de compromiso, si se proporciona alguna, para utilizarla en el momento de la recuperación.
- El sistema graba registros en el archivo antes de efectuar la operación de compromiso si se dan las dos circunstancias siguientes:
  - Se han añadido registros a un archivo de base de datos local o remoto bajo control de compromiso.
  - Se ha especificado SEQONLY(\*YES) al abrir el archivo de modo que el sistema emplea la información de E/S de bloques y existe un bloque de registros parcial.

De lo contrario, el área de información de E/S y los almacenamientos intermedios de E/S no se cambian.

- El sistema efectúa una llamada al programa de salida de compromiso y retroacción para cada recurso de compromiso de API que esté presente en la definición de compromiso. Si una ubicación tiene más de un programa de salida registrado, el sistema llama a los programas de salida de esa ubicación en el orden en que se han registrado.
- Si se han efectuado cambios de registros en los recursos asignados a un diario, el sistema graba una entrada de diario C CM en cada diario local asociado a la definición de compromiso. En Secuencia de entradas de diario bajo control de compromiso se muestran las entradas que normalmente se graban mientras una definición de compromiso está activa.
- El sistema establece como permanentes los cambios a nivel de objeto que están pendientes.
- El sistema desbloquea los bloqueos de registros y objetos adquiridos y mantenidos con fines de control de compromiso. Estos recursos quedan disponibles para otros usuarios.
- El sistema cambia la información de la definición de compromiso para mostrar que la transacción actual se ha finalizado.

El sistema debe llevar a cabo todos los pasos anteriores correctamente para que la operación de compromiso sea satisfactoria.

#### **Conceptos relacionados**

“Definición de compromiso” en la página 5

Se crea una definición de compromiso cuando se utiliza el mandato Arrancar control de compromiso (STRCMTCTL) para iniciar el control de compromiso en el sistema. Además, DB2 Universal Database (UDB) para iSeries crea automáticamente una definición de compromiso cuando el nivel de aislamiento es distinto de Ningún compromiso.

“Secuencia de entradas de diario bajo control de compromiso” en la página 17

Esta tabla muestra la secuencia de las entradas que normalmente se escriben mientras una definición de compromiso está activa. Puede utilizar el buscador de información de entradas de diario para obtener más detalles sobre el contenido de las entradas de diario.

#### **Operación de retroacción**

La operación de retroacción elimina todos los cambios efectuados desde la operación de compromiso o retroacción anterior. El sistema también libera todos los bloqueos relacionados con la transacción.

El sistema lleva a cabo los pasos siguientes cuando recibe una petición de retroacción:

- El sistema borra los registros del almacenamiento intermedio de E/S si se cumplen estas dos condiciones:
  - Se han añadido registros a un archivo de base de datos local o remoto bajo control de compromiso.
  - Se ha especificado SEQONLY(\*YES) al abrir el archivo, por lo que el sistema emplea la E/S agrupada en bloques, y existe un bloque de registros parcial que todavía no se ha escrito en la base de datos.

De lo contrario, el área de información de E/S y los almacenamientos intermedios de E/S no cambian.

- El sistema efectúa una llamada al programa de salida de compromiso o retrotracción para cada recurso de compromiso de API que esté presente en la definición de compromiso. Si una ubicación tiene más de un programa de salida registrado, el sistema llama a los programas de salida de esa ubicación en el orden inverso al que se han registrado.
- Si se había suprimido un registro de un archivo, el sistema añade el registro de nuevo al archivo.
- El sistema elimina los cambios efectuados en los registros que se han efectuado durante esta transacción y coloca los registros originales (las imágenes anteriores) de nuevo en el archivo.
- Si en esta transacción se han añadido registros al archivo, permanecen en el archivo como archivos suprimidos.
- Si se han realizado cambios de registro en los recursos asignados a un diario durante la transacción, el sistema añade una entrada C RB al diario para indicar que se ha producido una operación de retrotracción. El diario también contiene imágenes de los cambios de registros retrotraídos. Antes de que se solicitara la operación de retrotracción, se colocaron en el diario imágenes anteriores y posteriores de los registros cambiados. El sistema también escribe una entrada C RB en el diario predeterminado si se asignan recursos comprometibles a ese diario.
- El sistema coloca los archivos abiertos bajo control de compromiso en una de las posiciones siguientes:
  - En el último registro al que se ha accedido en la transacción anterior.
  - En la posición de apertura si no se ha efectuado ninguna operación de compromiso para el archivo con esta definición de compromiso.

Esta consideración es importante si se efectúa un proceso secuencial.

- El sistema no retrotrae los cambios que no se pueden comprometer para los archivos de base de datos. Por ejemplo, los archivos abiertos no se cierran y los archivos borrados no se restauran. El sistema no vuelve a abrir ni vuelve a colocar los archivos cerrados durante esta transacción.
- El sistema desbloquea los bloqueos de registros adquiridos con fines de control de compromiso y establece estos registros como disponibles para otros usuarios.
- La identificación de compromiso salvada por el sistema en este momento sigue siendo la misma que la identificación de compromiso proporcionada con la última operación de compromiso para la misma definición de compromiso.
- El sistema invierte o retrotrae los cambios comprometibles a nivel de objeto efectuados durante esta transacción.
- Los bloqueos de objetos adquiridos con fines de control de compromiso se desbloquean y esos objetos pasan a estar disponibles para otros usuarios.
- El sistema establece el límite de compromiso anterior como el límite de compromiso actual.
- El sistema cambia la información de la definición de compromiso para mostrar que la transacción actual se ha finalizado.

El sistema debe llevar a cabo todos los pasos anteriores correctamente para que la operación de retrotracción sea satisfactoria.

## Definición de compromiso

Se crea una definición de compromiso cuando se utiliza el mandato Arrancar control de compromiso (STRCMTCTL) para iniciar el control de compromiso en el sistema. Además, DB2 Universal Database (UDB) para iSeries crea automáticamente una definición de compromiso cuando el nivel de aislamiento es distinto de Ningún compromiso.

La definición de compromiso contiene información relacionada con los recursos que se cambian bajo control de compromiso en ese trabajo. El sistema mantiene la información de control de compromiso en la definición de compromiso a medida que cambian los recursos de compromiso, hasta que se finaliza la definición de compromiso. Cada transacción activa del sistema se representa mediante una definición de compromiso. Una transacción posterior puede reutilizar una definición de compromiso tras cada compromiso o retrotracción de una transacción activa.

Una definición de compromiso normalmente incluye:

- Los parámetros del mandato STRCMTCTL.
- El estado actual de la definición de compromiso.
- Información sobre los archivos de base de datos y otros recursos que pueden comprometerse que contienen cambios efectuados durante la transacción actual.

En el caso de las definiciones de compromiso con bloqueos cuyo ámbito se circunscribe al trabajo, únicamente el trabajo que inicia el control de compromiso conoce esa definición de compromiso. Ningún otro trabajo conoce esa definición de compromiso.

Los programas pueden iniciar y emplear varias definiciones de compromiso. Cada definición de compromiso de un trabajo identifica una transacción aparte que tiene recursos comprometibles asociados a ella. Estas transacciones se pueden comprometer o retrotraer con independencia de las transacciones que están asociadas a otras definiciones de compromiso iniciadas para el trabajo.

#### **Conceptos relacionados**

“Operación de compromiso” en la página 3

Una operación de compromiso establece como permanentes todos los cambios efectuados bajo control de compromiso desde la operación de compromiso o retroacción anterior. El sistema también libera todos los bloqueos relacionados con la transacción.

“Control de compromiso y agrupaciones de discos independientes” en la página 22

Cada agrupación de discos independiente y grupo de agrupaciones de discos independientes puede tener una base de datos SQL i5/OS aparte. Se puede emplear el control de compromiso con estas bases de datos.

“Consideraciones sobre las agrupaciones de discos independientes en las definiciones de compromiso” en la página 23

Debe tener presentes las consideraciones sobre definiciones de compromiso presentadas en este tema a la hora de utilizar agrupaciones de discos independientes.

### **Ámbito de una definición de compromiso**

El *ámbito* de una definición de compromiso determina qué programas utilizarán esa definición de compromiso y cuál será el ámbito de los bloqueos adquiridos en las transacciones.

La interfaz que inicia la definición de compromiso determina el ámbito de la definición de compromiso. Hay cuatro ámbitos posibles para las definiciones de compromiso, que se dividen en dos categorías generales:

#### **Definiciones de compromiso con bloqueos cuyo ámbito se circunscribe al trabajo**

- Definición de compromiso a nivel de grupo de activación
- Definición de compromiso a nivel de trabajo
- Definición de compromiso denominada explícitamente

#### **Definiciones de compromiso con bloqueos cuyo ámbito se circunscribe a la transacción**

- Definición de compromiso con ámbito de transacción

Solo los programas que se ejecutan en el trabajo que ha iniciado las definiciones de compromiso pueden emplear las definiciones de compromiso con bloqueos cuyo ámbito se circunscribe al trabajo. En cambio, varios trabajos pueden emplear las definiciones de compromiso con bloqueos cuyo ámbito se circunscribe a la transacción.

Las aplicaciones normalmente utilizan definiciones de compromiso a nivel de grupo de activación o a nivel de trabajo. Estas definiciones de compromiso se crean explícitamente con el mandato Arrancar control de compromiso (STRCMTCTL) o puede crearlas implícitamente el sistema cuando una aplicación SQL se ejecuta con un nivel de aislamiento distinto de \*NONE.

## Definición de compromiso a nivel de grupo de activación

El ámbito más habitual es el de grupo de activación. La definición de compromiso a nivel de grupo de activación es el ámbito predeterminado cuando el mandato STRCMTCTL inicia explícitamente la definición de compromiso o cuando una aplicación SQL que se ejecuta con un nivel de aislamiento distinto de Ningún compromiso inicia implícitamente la definición de compromiso. Solo los programas que se ejecutan en ese grupo de activación utilizan esa definición de compromiso. Puede haber muchas definiciones de compromiso a nivel de grupo de activación activas para un trabajo a la vez. Sin embargo, cada definición de compromiso a nivel de grupo de activación solo puede estar asociada a un único grupo de activación. Los programas que se ejecutan en ese grupo de activación solo pueden asociar sus cambios para comprometer a esa definición de compromiso a nivel de grupo de activación.

Cuando iSeries Navigator, el mandato Trabajar con definiciones de compromiso (WRKCMTDFN), el mandato Visualizar trabajo (DSPJOB) o el mandato Trabajar con trabajo (WRKJOB) visualiza una definición de compromiso a nivel de grupo de activación, estos campos muestran la siguiente información:

- El campo de definición de compromiso visualiza el nombre del grupo de activación. Muestra el valor especial \*DFTACTGRP para indicar el grupo de activación predeterminado.
- El campo de grupo de activación muestra el número de grupo de activación.
- El campo de trabajo muestra el trabajo que ha iniciado la definición de compromiso.
- El campo de hebra visualiza \*NONE.

## Definición de compromiso a nivel de trabajo

El ámbito de una definición de compromiso solo se puede circunscribir al trabajo si se emite STRCMTCTL CMTSCOPE(\*JOB). Cualquier programa que se ejecute en un grupo de activación que no tenga iniciada una definición de compromiso a nivel de grupo de activación utilizará la definición de compromiso a nivel de trabajo, si otro programa ya la ha iniciado para el trabajo. Solo puede iniciar una única definición de compromiso a nivel de trabajo para un trabajo.

Cuando iSeries Navigator, el mandato Trabajar con definiciones de compromiso (WRKCMTDFN), el mandato Visualizar trabajo (DSPJOB) o el mandato Trabajar con trabajo (WRKJOB) visualiza una definición de compromiso a nivel de trabajo, estos campos muestran la siguiente información:

- El campo de definición de compromiso visualiza el valor especial \*JOB.
- El campo de grupo de activación muestra un espacio en blanco.
- El campo de trabajo muestra el trabajo que ha iniciado la definición de compromiso.
- El campo de hebra visualiza \*NONE.

Para un grupo de activación determinado, los programas que se ejecutan en ese grupo de activación solo pueden emplear una definición de compromiso. Por lo tanto, los programas que se ejecutan en un grupo de activación pueden emplear la definición de compromiso a nivel de trabajo o a nivel de grupo de activación, pero no ambas a la vez. En un trabajo multihebra que no utiliza la modalidad de servidor SQL, el ámbito del trabajo transaccional de un programa se circunscribirá a la definición de compromiso adecuada según el grupo de activación del programa, independientemente de la hebra que lo lleve a cabo. Si varias hebras utilizan el mismo grupo de activación, estas deben cooperar para llevar a cabo el trabajo transaccional y asegurarse de que las operaciones de compromiso y retroacción se producen en el momento correcto.

Aun cuando la definición de compromiso a nivel de trabajo está activa para el trabajo, un programa puede iniciar la definición de compromiso a nivel de grupo de activación si ningún trabajo en ejecución en ese grupo de activación ha llevado a cabo ninguna petición ni operación de control de compromiso para la definición de compromiso a nivel de trabajo. De lo contrario, deberá finalizar primero la definición de compromiso a nivel de trabajo antes de poder iniciar la definición de compromiso a nivel

de grupo de activación. Las peticiones u operaciones de control de compromiso para la definición de compromiso a nivel de trabajo que pueden impedir que se inicie la definición de compromiso a nivel de grupo de activación son las siguientes:

- Abrir (en modalidad completa o compartida) un archivo de base de datos bajo control de compromiso.
- Utilizar la API Añadir recurso de compromiso (QTNADDCR) para añadir un recurso de compromiso de API.
- Comprometer una transacción.
- Retrotraer una transacción.
- Añadir un recurso remoto bajo control de compromiso.
- Utilizar la API Cambiar opciones de compromiso (QTNCHGCO) para cambiar las opciones de compromiso.
- Establecer la definición de compromiso en un estado de retrotracción necesaria mediante la API Retrotracción necesaria (QTNRBRQD).
- Enviar una entrada de diario de usuario que incluya el identificador de ciclo de compromiso actual empleando la API Enviar entrada de diario (QJOSJRNE) con el parámetro Incluir identificador de ciclo de compromiso.

Del mismo modo, si los programas de un grupo de activación en este momento utilizan la definición de compromiso a nivel de grupo de activación, primero debe finalizarse la definición de compromiso para que los programas que se ejecutan en ese mismo grupo de activación puedan emplear la definición de compromiso a nivel de trabajo.

Al abrir un archivo de base de datos, el ámbito de apertura del archivo abierto puede circunscribirse al grupo de activación o al trabajo con una restricción: si un programa abre un archivo bajo control de compromiso y el ámbito del archivo se circunscribe al trabajo, el programa que realiza la petición de apertura debe utilizar la definición de compromiso a nivel de trabajo.

## **Definición de compromiso denominada explícitamente**

El sistema inicia definiciones de compromiso denominadas explícitamente cuando necesita llevar a cabo sus propias transacciones de control de compromiso sin afectar a las transacciones empleadas por una aplicación. Las anotaciones de problemas son un ejemplo de una función que inicia estos tipos de definiciones de compromiso. Una aplicación no puede iniciar definiciones de compromiso denominadas explícitamente.

Cuando iSeries Navigator, el mandato Trabajar con definiciones de compromiso (WRKCMTDFN), el mandato Visualizar trabajo (DSPJOB) o el mandato Trabajar con trabajo (WRKJOB) visualiza una definición de compromiso denominada explícitamente, estos campos muestran la siguiente información:

- El campo de definición de compromiso visualiza el nombre que le ha asignado el sistema.
- El campo de grupo de activación muestra un espacio en blanco.
- El campo de trabajo muestra el trabajo que ha iniciado la definición de compromiso.
- El campo de hebra visualiza \*NONE.

## **Definiciones de compromiso con ámbito de transacción**

Las definiciones de compromiso con ámbito de transacción se inician con las API XA para los bloqueos con ámbito de transacción.

Estas API utilizan protocolos de control de compromiso basados en hebras o en conexiones SQL, y no en grupos de activación. Dicho de otro modo, las API se utilizan para asociar la definición de compromiso a una hebra o a una conexión SQL determinada mientras se lleva a cabo el trabajo transaccional, así como

para comprometer o retrotraer las transacciones. El sistema asocia estas definiciones de compromiso a las hebras que llevan a cabo el trabajo transaccional, respecto a los protocolos de las API. Pueden ser utilizadas por hebras de distintos trabajos.

Cuando iSeries Navigator, el mandato Trabajar con definiciones de compromiso (WRKCMTDFN), el mandato Visualizar trabajo (DSPJOB) o el mandato Trabajar con trabajo (WRKJOB) visualiza una definición de compromiso con ámbito de transacción, estos campos muestran la siguiente información:

- El campo de definición de compromiso visualiza el valor especial \*TNSOBJ.
- El campo de grupo de activación muestra un espacio en blanco.
- El campo de trabajo muestra el trabajo que ha iniciado la definición de compromiso. O bien, si la definición de compromiso está unida actualmente a una hebra, se visualiza el trabajo de la hebra.
- El campo de hebra visualiza la hebra a la que está asociada la definición de compromiso (o \*NONE si en este momento la definición de compromiso no está asociada a ninguna hebra).

#### Referencia relacionada

Interfaces API XA

### Nombres de definiciones de compromiso

El sistema da nombres a todas las definiciones de compromiso iniciadas para un trabajo.

La tabla siguiente muestra diversas definiciones de compromiso y los nombres que tienen asociados para un trabajo determinado.

Grupo de activación	Ámbito de compromiso	Nombre de definición de compromiso
Cualquiera	Trabajo	*JOB
Grupo de activación por omisión	Grupo de activación	*DFACTGRP
Grupo de activación denominado por el usuario	Grupo de activación	Nombre de grupo de activación (por ejemplo, PAYROLL)
Grupo de activación denominado por el sistema	Grupo de activación	Número de grupo de activación (por ejemplo, 000000145)
Ninguno	Denominado explícitamente	QDIR001 (ejemplo de una definición de compromiso definida por el sistema para uso exclusivo del sistema). Los nombres de definiciones de compromiso definidos por el sistema empiezan por Q.
Ninguno	Transacción	*TNSOBJ

Solo los programas IBM Integrated Language Environment (ILE) compilados pueden iniciar el control de compromiso para grupos de activación distintos del grupo de activación predeterminado. Por consiguiente, un trabajo solo puede emplear múltiples definiciones de compromiso si ejecuta uno o varios programas compilados ILE.

Los programas de modelo de programa original (OPM) se ejecutan en el grupo de activación predeterminado y utilizan por omisión la definición de compromiso \*DFACTGRP. En un entorno OPM e ILE combinado, los trabajos deben emplear la definición de compromiso a nivel de trabajo si todos los cambios comprometibles efectuados por todos los programas deben comprometerse o retrotraerse juntos.

Un archivo de base de datos abierto cuyo ámbito se circunscribe a un grupo de activación se puede asociar a una definición de compromiso a nivel de grupo de activación o a nivel de trabajo. Un archivo de base de datos abierto cuyo ámbito se circunscribe al trabajo se puede asociar únicamente a la definición de compromiso a nivel de trabajo. Por lo tanto, cualquier programa, OPM o ILE, que abra un

archivo de base de datos bajo control de compromiso cuyo ámbito se circunscriba al trabajo deberá utilizar la definición de compromiso a nivel de trabajo.

Los programas de aplicación no utilizan el nombre de definición de compromiso para identificar una definición de compromiso determinada al efectuar una petición de control de compromiso. Los nombres de definiciones de compromiso se utilizan principalmente en los mensajes para identificar una definición de compromiso específica para un trabajo.

En el caso de las definiciones de compromiso a nivel de grupo de activación, el sistema determina qué definición de compromiso hay que utilizar, basándose en qué grupo de activación se ejecuta el programa peticionario. Esto es posible dado que los programas que se ejecutan en un grupo de activación en todo momento solo pueden emplear una única definición de compromiso.

En el caso de las transacciones con bloqueos cuyo ámbito se circunscribe a la transacción, las API XA y los atributos relacionados con las transacciones añadidos a la interfaz CLI determinan qué definición de compromiso utiliza la hebra de invocación.

#### **Información relacionada**

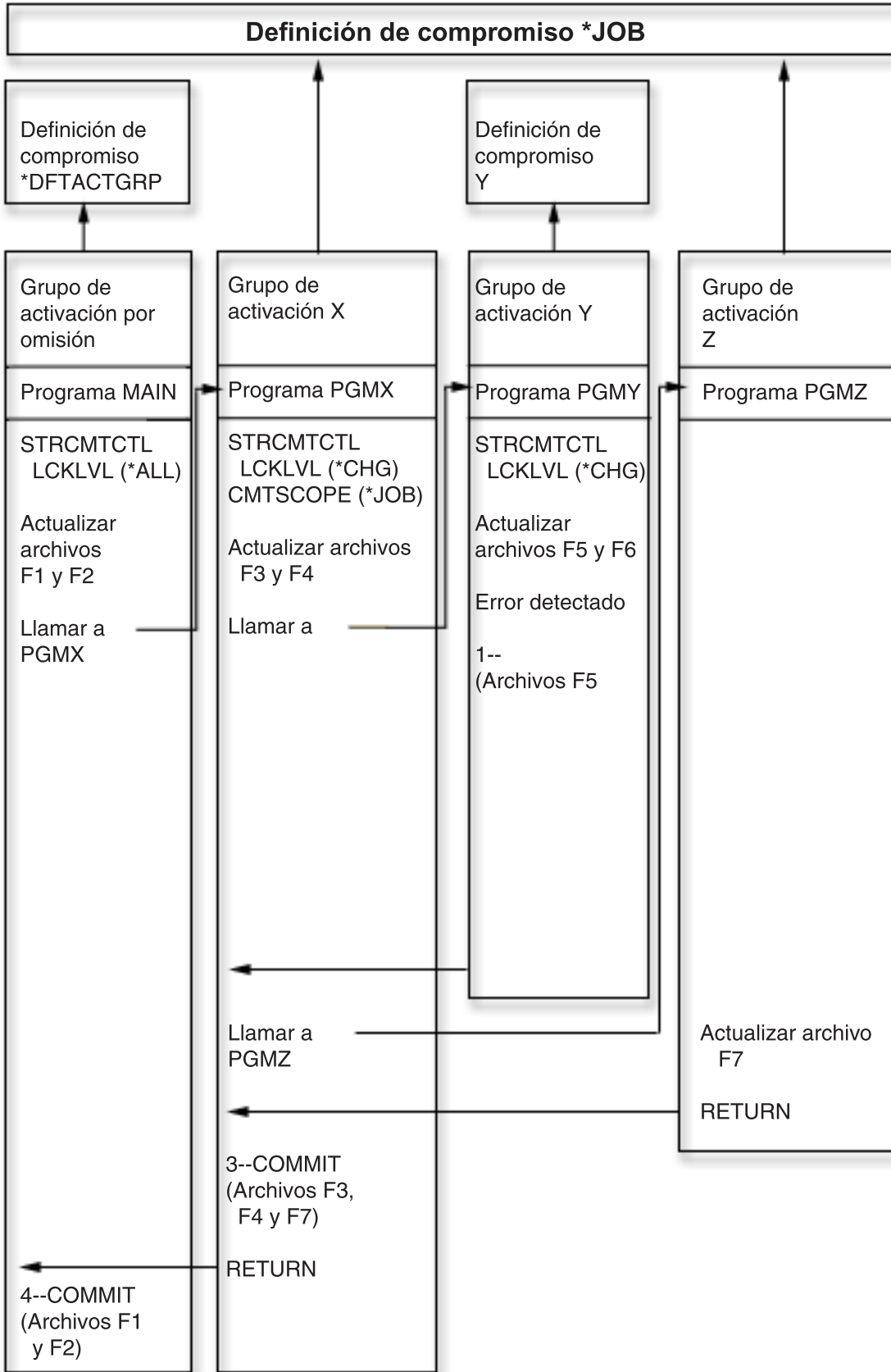
PDF de Conceptos sobre ILE

### **Ejemplo: trabajos y definiciones de compromiso**

Esta figura muestra un ejemplo de un trabajo que utiliza múltiples definiciones de compromiso.

Indica qué actualizaciones de archivos se comprometen o retrotraen en cada nivel de grupo de activación. El ejemplo supone que todas las actualizaciones que efectúan en los archivos de base de datos todos los programas se realizan bajo control de compromiso.





La siguiente tabla muestra cómo se comprometen o retrotraen los archivos si cambia el caso práctico de la figura anterior.

### Ejemplos adicionales de múltiples definiciones de compromiso de un trabajo

Cambio en el caso práctico	Efecto de los cambios en estos archivos:			
	F1 y F2	F3 y F4	F5 y F6	F7
PGMX lleva a cabo una operación de retrotracción en lugar de una operación de compromiso (3= =COMMIT pasa a ser ROLLBACK).	Siguen pendientes	Retrotraídos	Ya comprometidos	Retrotraídos
PGMZ lleva a cabo una operación de compromiso antes de volver a PGMX.	Siguen pendientes	Comprometidos por PGMZ	Ya comprometidos	Comprometido
PGMZ intenta iniciar el control de compromiso especificando CMTSCOPE(*ACTGRP) después de actualizar el archivo F7. El intento resulta anómalo ya que hay cambios pendientes que utilizan la definición de compromiso a nivel de trabajo.	Siguen pendientes	Siguen pendientes	Ya comprometidos	Siguen pendientes
PGMX no inicia el control de compromiso ni abre los archivos F3 y F4 con COMMIT(*YES). PGMZ intenta abrir el archivo F7 con COMMIT(*YES).	Siguen pendientes	No bajo control de compromiso	Ya comprometidos	El archivo F7 no se puede abrir porque no existe ninguna definición de compromiso *JOB (PGMX no la ha creado).

### Cómo funciona el control de compromiso con los objetos

Un objeto, cuando se coloca bajo control de compromiso, pasa a ser un recurso comprometible. Se registra en la definición de compromiso. Participa en cada operación de compromiso y retrotracción que se lleve a cabo para esa definición de compromiso.

Los temas siguientes describen estos atributos de un recurso comprometible:

- Tipo de recurso
- Ubicación
- Protocolo de compromiso
- Propósito de acceso

## Tipos de recursos comprometibles

En este tema encontrará una tabla con los distintos tipos de recursos comprometibles.

En la tabla se muestran:

- Los tipos de recursos comprometibles.
- Cómo se colocan bajo control de compromiso.
- Cómo se eliminan del control de compromiso.
- Las restricciones que se aplican al tipo de recurso.

Tipo de recurso	Cómo colocarlo bajo control de compromiso	Cómo eliminarlo del control de compromiso	Qué tipos de cambios son comprometibles	Restricciones
FILE - Archivos de base de datos locales	Abriéndolo bajo control de compromiso <sup>1</sup> .	Cerrando el archivo, si no hay cambios pendientes.  Si hay cambios pendientes al cerrar el archivo, tras efectuar la siguiente operación de compromiso o retroacción.	Cambios a nivel de registro	No se pueden bloquear más de 500 000 000 de registros de una sola transacción <sup>2</sup> .
DDL - Cambios a nivel de objeto en tablas SQL y colecciones SQL locales.	Ejecutando SQL bajo control de compromiso.	Realizando una operación de compromiso o retroacción tras el cambio a nivel de objeto.	Cambios a nivel de objeto, como los de: <ul style="list-style-type: none"> <li>• Crear paquete SQL</li> <li>• Crear tabla SQL</li> <li>• Eliminar tabla SQL</li> </ul>	Solo los cambios a nivel de objeto efectuados utilizando SQL están bajo control de compromiso.
DDM - Archivo de gestión de datos distribuidos (DDM) remoto	Abriéndolo bajo control de compromiso. En la información sobre el soporte de control de compromiso para DDM encontrará más detalles sobre el control de compromiso y la gestión de datos distribuidos (DDM).	Cerrando el archivo, si no hay cambios pendientes.  Si hay cambios pendientes al cerrar el archivo, tras efectuar la siguiente operación de compromiso o retroacción.	Cambios a nivel de registro	
LU 6.2 - Conversación protegida	Iniciando la conversación <sup>3</sup> .	Finalizando la conversación.		
DRDA - base de datos relacional distribuida	Utilizando la sentencia SQL CONNECT.	Finalizando la conexión.		

Tipo de recurso	Cómo colocarlo bajo control de compromiso	Cómo eliminarlo del control de compromiso	Qué tipos de cambios son comprometibles	Restricciones
API - Recurso de compromiso de API local	API Añadir recurso de compromiso (QTNADDCR)	API Eliminar recurso de compromiso (QTNRMVCR)	El programa del usuario determina esta cuestión. El programa del usuario podría escribir entradas de diario mediante la API Enviar entrada de diario (QJOSJRNE) para ayudar a hacer un seguimiento de estos cambios.	La aplicación debe proporcionar un programa de salida al que se llamará durante las operaciones de compromiso, retrotracción o resincronización.
TCP - Conexión TCP/IP	Utilizando la sentencia SQL CONNECT en una RDB definida para utilizar las conexiones TCP/IP, o abriendo un archivo DDM definido con una ubicación TCP/IP.	Finalizando la conexión SQL o cerrando el archivo DDM si no hay cambios pendientes. Si el archivo DDM se cierra con cambios pendientes, la conexión se cierra tras realizar la siguiente operación de compromiso o retrotracción.		

**Notas:**

<sup>1</sup> Para obtener información detallada sobre cómo colocar un archivo de base de datos bajo control de compromiso, vea el manual de consulta del lenguaje pertinente. En Información relacionada con el control de compromiso, hay enlaces que llevan a manuales de lenguaje que le interesarán.

<sup>2</sup> Puede utilizar un archivo QAQQINI para reducir el límite de 500 000 000. Encontrará instrucciones en: "Gestionar el tamaño de las transacciones" en la página 74.

<sup>3</sup> Cuando se inicia una conexión DDM, el archivo DDM especifica PTCCNV(\*YES) y el archivo DDM se define con una ubicación remota SNA; se añade un recurso LU 6.2 con el recurso DDM.

Cuando se inicia una conexión DRDA, se añade un recurso LU 6.2 con el recurso DRDA si se cumplen estas dos condiciones:

- El programa utiliza los protocolos de conexión de unidad de trabajo distribuida.
- La conexión se establece con una base de datos relacional (RDB) definida con una ubicación remota SNA. Hallará

más información sobre cómo iniciar conversaciones protegidos en la publicación APPC Programming  .

**Conceptos relacionados**

Soporte de control de compromiso para DDM

"Actualizaciones del objeto de notificación" en la página 65

En este tema figuran los cambios no comprometidos en relación con el objeto de notificación.

**Referencia relacionada**

API Añadir recurso de compromiso (QTNADDCR)

Mandato Eliminar recurso de compromiso (QTNRMVCR)

Mandato Enviar entrada de diario (QJOSJRNE)

**Recursos comprometibles locales y remotos**

Un recurso comprometible puede ser un recurso local o un recurso remoto.

## Recurso comprometible local

El recurso comprometible local está en el mismo sistema que la aplicación. Cada diario asociado a los recursos bajo control de compromiso se puede describir de forma general como una ubicación local. Todos los recursos que se registran sin un diario (de forma opcional en el caso de los recursos DDL y los recursos de API) pueden considerarse una ubicación local aparte.

Si un recurso comprometible está en una agrupación de discos independiente, el recurso no se considera local si la definición de compromiso está en una agrupación de discos distinta.

## Recursos comprometibles remotos

Un recurso comprometible remoto está en un sistema distinto del de la aplicación. Existe una ubicación remota para cada conversación exclusiva con un sistema remoto. Una definición de compromiso puede tener una o varias ubicaciones remotas en uno o varios sistemas remotos.

Si coloca un recurso local bajo control de compromiso para la agrupación de discos del sistema, o para cualquier agrupación de discos independiente, debe utilizar DRDA para acceder a los recursos bajo control de compromiso en cualquier otra agrupación de discos independiente.

La siguiente tabla muestra los tipos de recursos para acceder a los recursos comprometibles y sus ubicaciones.

Tipo de recurso	Ubicación
API	Local
DDL	Local
DDM	Remota
DRDA	Local o remota
FILE	Local
LU62	Remota
TCP	Remota

### Conceptos relacionados

“Control de compromiso y agrupaciones de discos independientes” en la página 22

Cada agrupación de discos independiente y grupo de agrupaciones de discos independientes puede tener una base de datos SQL i5/OS aparte. Se puede emplear el control de compromiso con estas bases de datos.

## Propósito de acceso de un recurso comprometible

El propósito de acceso determina cómo participan juntos los recursos en una transacción.

Cuando se coloca un recurso bajo control de compromiso, el gestor de recursos indica cómo se accederá al recurso:

- Actualización
- Solo lectura
- Indeterminado

La siguiente tabla muestra los propósitos de acceso posibles para un tipo de recurso concreto y cómo determina el sistema el propósito de acceso de un recurso cuando se registra.

Tipo de recurso	Propósitos de acceso posibles	Cómo se determina el propósito de acceso
FILE	Actualización, solo lectura	En función de cómo se ha abierto el archivo
DDL	Actualización	Actualización siempre
API	Actualización	Actualización siempre
DDM	Actualización, solo lectura	En función de cómo se ha abierto el archivo
LU62	Indeterminado	Indeterminado siempre
DRDA	Actualización, solo lectura, indeterminado	Para DRDA Nivel 1, el propósito de acceso siempre es de actualización si no hay otros recursos remotos registrados. De lo contrario, el propósito de acceso es solo de lectura. Para DRDA Nivel 2, el propósito de acceso siempre indeterminado.
TCP	Indeterminado	Indeterminado siempre

El propósito de acceso de los recursos que ya están registrados determina si se puede registrar un nuevo recurso. Se aplican las normas siguientes:

- Un recurso de una fase cuyo propósito de acceso es de actualización no se puede registrar cuando se cumple alguna de estas condiciones:
  - Ya hay recursos cuyo propósito de acceso es de actualización registrados en otras ubicaciones.
  - Ya hay recursos cuyo propósito de acceso es indeterminado registrados en otras ubicaciones.
  - Ya hay recursos cuyo propósito de acceso es indeterminado registrados en la misma ubicación y los recursos han cambiado durante la transacción actual.
- Un recurso de dos fases cuyo propósito de acceso es de actualización no se puede registrar cuando ya hay registrado un recurso de una fase cuyo propósito de acceso sea de actualización.

### Protocolo de compromiso de un recurso comprometible

*Protocolo de compromiso* es la posibilidad que tiene un recurso de participar en el proceso de compromiso de una fase o de dos fases. Los recursos locales, excepto los recursos comprometibles de API, siempre son recursos de dos fases.

Si un recurso comprometible reside en una agrupación de discos independiente y la definición de comprometible reside en una agrupación de discos distinta, el recurso no se considera local i de dos fases.

Un recurso de dos fases también se denomina *recurso protegido*. Los recursos remotos y los recursos comprometibles de API se deben registrar como recursos de una fase o recursos de dos fases cuando se colocan bajo control de compromiso. La tabla siguiente muestra los tipos de recursos comprometibles que pueden coexistir en una definición de compromiso con un recurso de una fase.

Tipo de recurso	Puede coexistir con
Recurso de API de una fase.	Otros recursos locales. Ningún recurso remoto.
Recurso remoto de una fase.	Otros recursos de una fase en la misma ubicación. Ningún recurso local.

### Conceptos relacionados

“Control de compromiso y agrupaciones de discos independientes” en la página 22

Cada agrupación de discos independiente y grupo de agrupaciones de discos independientes puede tener una base de datos SQL i5/OS aparte. Se puede emplear el control de compromiso con estas bases de datos.

## **Archivos registrados por diario y control de compromiso**

Debe registrar por diario (anotar) un archivo de base de datos (cuyo tipo de recurso es FILE o DDM) antes de que este pueda abrirse para salida bajo control de compromiso o de que una aplicación SQL que utilice un nivel de aislamiento distinto de Ningún compromiso pueda hacer referencia al mismo. No es necesario registrar por diario un archivo a fin de abrirlo para entrada únicamente bajo control de compromiso.

Se produce un error si:

- Se intenta abrir un archivo de base de datos para salida bajo control de compromiso, pero el archivo actualmente no está registrado por diario.
- No hay iniciada ninguna definición de compromiso que pueda emplear el archivo que se abre bajo control de compromiso.

Si solo se registran por diario las imágenes posteriores de un archivo de base de datos cuando se abre ese archivo bajo control de compromiso, el sistema automáticamente empieza a registrar por diario las imágenes anteriores y posteriores. Las imágenes anteriores se escriben únicamente para los cambios efectuados en el archivo que se producen bajo control de compromiso. Si al mismo tiempo en el archivo se producen otros cambios que no están bajo control de compromiso, solo se escriben las imágenes posteriores de esos cambios.

El sistema escribe automáticamente los cambios comprometibles a nivel de registro y los cambios comprometibles a nivel de objeto en un diario. En el caso de los cambios a nivel de registro, posteriormente el sistema utiliza las entradas de diario, si es necesario, con fines de recuperación; el sistema no emplea las entradas de los cambios comprometibles a nivel de objeto con vistas a la recuperación. Además, el sistema no escribe automáticamente entradas de diario para los recursos de compromiso de API. Sin embargo, el programa de salida del recurso de API puede emplear la API Enviar entrada de diario (QJOSJRNE) para escribir entradas de diario a fin de proporcionar un seguimiento de auditoría o contribuir a la recuperación. El programa de salida de usuario controla el contenido de estas entradas.

El sistema emplea una técnica distinta de un diario para efectuar la recuperación de los recursos de compromiso a nivel de objeto. La recuperación de los recursos de compromiso de API se lleva a cabo llamando al programa de salida de compromiso y retroacción asociado a cada recurso de compromiso de API específico. El programa de salida es el encargado de llevar a cabo la recuperación real que se necesita para la situación.

### **Conceptos relacionados**

Gestión de diarios

## **Secuencia de entradas de diario bajo control de compromiso**

Esta tabla muestra la secuencia de las entradas que normalmente se escriben mientras una definición de compromiso está activa. Puede utilizar el buscador de información de entradas de diario para obtener más detalles sobre el contenido de las entradas de diario.

Las entradas de control de compromiso se escriben en un diario local si se cumple al menos una de las siguientes condiciones:

- El diario se especifica como diario predeterminado en el mandato Arrancar control de compromiso (STRCMTCTL).
- Se abre al menos un archivo registrado por diario en el diario bajo control de compromiso.
- Se registra bajo control de compromiso al menos un recurso de compromiso de API asociado al diario.

Tipo de entrada	Descripción	Dónde se escribe	Cuándo se escribe
C BC	Empezar control de compromiso	En el diario predeterminado, si se especifica alguno en el mandato STRCMTCTL.	Cuando se utiliza el mandato STRCMTCTL.
		En el diario.	Cuando se abre el primer archivo registrado en un diario o cuando se registra un recurso de API para un diario.
C SC	Iniciar ciclo de compromiso	En el diario.	Cuando se produce el primer cambio de registro para la transacción de un archivo registrado en este diario <sup>1</sup> .
		En el diario de un recurso de API.	Cuando se utiliza por primera vez la API QJOSJRNE con la clave <i>Incluir identificador de ciclo de compromiso</i> .
Códigos de diario D y F	Entradas DDL a nivel de objeto.	En el diario asociado al objeto que se actualiza. Solo las entradas de diario que contienen un identificador de ciclo de compromiso representan un cambio DDL a nivel de objeto que forma parte de la transacción.	Cuando se producen actualizaciones.
Código de diario R	Entradas a nivel de registro.	En el diario asociado al archivo que se actualiza.	Cuando se producen actualizaciones.
Código de diario U	Entradas creadas por el usuario.	En el diario asociado a un recurso de API.	Si el programa de aplicación utiliza por primera vez la API QJOSJRNE con la clave <i>Incluir identificador de ciclo de compromiso</i> .
C CM	Compromiso	En el diario.	Cuando el compromiso ha finalizado satisfactoriamente.
		En el diario predeterminado.	Si hay recursos comprometibles asociados al diario.
C RB	Retrotracción	En el diario.	Una vez finalizada la operación de retrotracción.
		En el diario predeterminado.	Si hay recursos comprometibles asociados al diario.



Tipo de entrada	Descripción	Dónde se escribe	Cuándo se escribe
C LW	Finalizar transacción.	En el diario predeterminado, si se especifica alguno en el mandato STRCMTCTL. El sistema escribe un registro de cabecera LW y uno o varios registros de detalle. Estas entradas solo se escriben si se ha especificado OMTJRNE(*NONE) en el mandato STRCMTCTL o se produce un error del sistema.	Cuando ha finalizado la operación de compromiso o retrotracción.
C EC	Finalizar control de compromiso	En el diario.	Cuando se completa el mandato Finalizar control de compromiso (ENDCMTCTL).
		En un diario local que no sea el diario predeterminado.	Cuando hay establecido un límite de compromiso, tras el punto en que se han eliminado del control de compromiso todos los recursos comprometibles asociados a ese diario.
C SB	Inicio de punto de salvar o ciclo de compromiso anidado	En el diario.	Cuando la aplicación crea un SQL SAVEPOINT o cuando el sistema crea un ciclo de compromiso anidado interno para manejar una serie de funciones de base de datos como una sola operación.
C SQ	Liberación del punto de salvar o compromiso del ciclo de compromiso anidado	En el diario.	Cuando la aplicación libera un SQL SAVEPOINT o cuando el sistema compromete un ciclo de compromiso anidado interno <sup>2</sup> .
C SU	Retrotracción del punto de salvar o ciclo de compromiso anidado	En el diario.	Cuando la aplicación retrotrae un SQL SAVEPOINT o cuando el sistema retrotrae un ciclo de compromiso anidado interno <sup>2</sup> .

Tipo de entrada	Descripción	Dónde se escribe	Cuándo se escribe
<b>Notas:</b>			
<p><sup>1</sup> Puede indicar que la parte de longitud fija de la entrada de diario incluya información de la transacción, especificando el valor de unidad lógica de trabajo (*LUW) en el parámetro Datos de longitud fija (FIXLENDTA) del mandato Crear diario (CRTJRN) o Cambiar diario (CHGJRN). Si especifica el parámetro FIXLENDTA (*LUW), la parte de longitud fija de cada entrada de diario C SC contendrá el ID de unidad lógica de trabajo (LUWID) de la transacción actual. Igual que sucede en el caso de las transacciones XA, si especifica el parámetro FIXLENDTA(*XID), la parte de longitud fija de cada entrada de diario C SC contendrá el XID de la transacción actual. El LUWID o XID puede ayudarle a encontrar todos los ciclos de compromiso de una transacción determinada si en la transacción intervienen varios diarios o sistemas.</p> <p><sup>2</sup> Estas entradas solo se envían si establece que la variable de entorno QTN_JRNSAVPT_MYLIB_MYJRN sea igual a *YES, donde MYJRN es el diario que se utiliza y MYLIB es la biblioteca en la que se almacena el diario. Se puede utilizar el valor especial *ALL en los valores de MYLIB y MYJRN. Puede establecer estas variables a nivel de todo el sistema o para un trabajo concreto. Para hacer que se envíen entradas al diario MYLIB/MYJRN solamente para un trabajo, utilice este mandato en ese trabajo:</p> <ul style="list-style-type: none"> <li>• ADDENVVAR ENVVAR(QTN_JRNSAVPT_MYLIB_MYJRN) VALUE(*YES)</li> </ul> <p>Para que se envíen entradas de todos los diarios de todos los trabajos, utilice este mandato:</p> <ul style="list-style-type: none"> <li>• ADDENVVAR ENVVAR('QTN_JRNSAVPT_*ALL_*ALL') VALUE(*YES) LEVEL(*SYS)</li> </ul> <p>Debe establecer las variables de entorno antes de iniciar el control de compromiso.</p>			

### Conceptos relacionados

“Operación de compromiso” en la página 3

Una operación de compromiso establece como permanentes todos los cambios efectuados bajo control de compromiso desde la operación de compromiso o retroacción anterior. El sistema también libera todos los bloqueos relacionados con la transacción.

Buscador de información de entradas de diario

### Referencia relacionada

Mandato Finalizar control de compromiso (ENDCMTCTL)

## Identificador de ciclo de compromiso

Un *ciclo de compromiso* es el tiempo que separa un límite de compromiso del siguiente. El sistema asigna un *identificador de ciclo de compromiso* para asociar de forma conjunta todas las entradas de diario de un ciclo de compromiso específico. Cada diario que participa en una transacción tiene su propio ciclo de compromiso y su propio identificador de ciclo de compromiso.

El identificador de ciclo de compromiso es el número de secuencia de diario de la entrada de diario C SC grabada para el ciclo de compromiso. El identificador de ciclo de compromiso se coloca en cada entrada de diario grabada durante el ciclo de compromiso. Si se emplea más de un diario durante el ciclo de compromiso, el identificador de ciclo de compromiso para cada diario es diferente.

Puede indicar que la parte de longitud fija de la entrada de diario incluya información de la transacción especificando el valor de unidad lógica de trabajo (\*LUW) para el parámetro Datos de longitud fija (FIXLENDTA) del mandato Crear diario (CRTJRN) o Cambiar diario (CHGJRN). Si especifica el parámetro FIXLENDTA (\*LUW), la parte de longitud fija de cada entrada de diario C SC contendrá el ID de unidad lógica de trabajo (LUWID) de la transacción actual. Igual que sucede en el caso de las transacciones XA, si especifica el parámetro FIXLENDTA(\*XID), la parte de longitud fija de cada entrada de diario C SC contendrá el XID de la transacción actual. El LUWID o XID puede ayudarle a encontrar todos los ciclos de compromiso de una transacción determinada si en la transacción intervienen varios diarios o sistemas.

Puede emplear la API Enviar entrada de diario (QJOSJRNE) para grabar entradas de diario para los recursos de API. Tiene la opción de incluir el identificador de ciclo de compromiso en esas entradas de diario.

Puede emplear el identificador de ciclo de compromiso para aplicar o eliminar los cambios registrados por diario hasta un límite de compromiso mediante el mandato Aplicar cambios registrados por diario (APYJRNCHG) o el mandato Eliminar cambios registrados por diario (RMVJRNCHG). Se aplican estas limitaciones:

- La mayoría de los cambios a nivel de objeto efectuados bajo control de compromiso se graban en el diario pero no se aplican ni se eliminan con los mandatos APYJRNCHG y RMVJRNCHG.
- La API QJOSJRNE graba las entradas de diario creadas por el usuario con el código de diario U. Estas entradas no se pueden aplicar ni eliminar con los mandatos APYJRNCHG y RMVJRNCHG. Deben aplicarse o eliminarse con un programa escrito por el usuario.

## Bloqueo de registros

Cuando un trabajo retiene un bloqueo de registro y otro trabajo intenta recuperar ese registro para actualización, el trabajo peticionario espera y se elimina del proceso activo.

El trabajo peticionario estará activo hasta que se produzca uno de estos eventos:

- Se libera el bloqueo del registro.
- Finaliza el tiempo de espera especificado.

Varios trabajos pueden solicitar un registro bloqueado por otro trabajo. Cuando se libera el bloqueo del registro, el primer trabajo en solicitar el registro recibirá ese registro. Al esperar un registro bloqueado, especifique el tiempo de espera en el parámetro WAITRCD de los siguientes mandatos de creación, cambio o alteración temporal:

- Crear archivo físico (CRTPF)
- Crear archivo lógico (CRTLFL)
- Crear archivo físico fuente (CRTSRCPF)
- Cambiar archivo físico (CHGPF)
- Cambiar archivo lógico (CHGLFL)
- Cambiar archivo físico fuente (CHGSRCPF)
- Alterar temporalmente archivo de base de datos (OVRDBF)

Al especificar el tiempo de espera, tenga en cuenta la siguiente información:

- Si no especifica ningún valor, el programa esperará el tiempo de espera por omisión del proceso.
- En el caso de las definiciones de compromiso con bloqueos cuyo ámbito se circunscribe a la transacción únicamente, el tiempo de espera por omisión del trabajo puede verse alterado temporalmente por un tiempo de espera de bloqueo de transacción que se puede especificar en:
  - La API xa\_open.
  - Una interfaz JDBC o JTA. En la información sobre transacciones distribuidas figuran estas API.
- Si no se puede asignar el registro en el tiempo especificado, se envía un mensaje de notificación al programa de lenguaje de alto nivel.
- Si se supera el tiempo de espera de un registro, el mensaje que se envía a las anotaciones de trabajo proporciona el nombre del trabajo que retiene el registro bloqueado que ha originado la espera del trabajo solicitante. Si sufre excepciones de bloqueos de registros, puede emplear las anotaciones de trabajo a modo de ayuda para determinar qué programas deben modificarse a fin de que no retengan bloqueos durante largos períodos.

Los programas conservan bloqueos de registros durante largos períodos por uno de los motivos siguientes:

- El registro permanece bloqueado mientras el usuario de la estación de trabajo se plantea la posibilidad de efectuar un cambio.
- El bloqueo del registro forma parte de una transacción de compromiso larga. Piense en la conveniencia de efectuar transacciones más pequeñas para que pueda llevarse a cabo una operación de compromiso con más frecuencia.
- Se ha producido un bloqueo no deseado. Por ejemplo, suponga que se define un archivo como archivo de actualización con claves exclusivas y que el programa actualiza y añade registros adicionales al archivo. Si el usuario de la estación de trabajo desea añadir un registro al archivo, el programa podría tratar de acceder al registro para determinar si la clave ya existe. Si ese es el caso, el programa informa al usuario de la estación de trabajo de que la petición efectuada no es válida. Cuando el registro se recupera del archivo, se bloquea hasta que otra operación de lectura efectuada en el mismo archivo lo libera implícitamente o hasta que se libera explícitamente.

**Nota:** Para obtener más información sobre cómo utilizar cada interfaz de lenguaje de alto nivel para liberar bloqueos de registros, vea el manual de consulta del lenguaje de alto nivel pertinente. En Información relacionada con el control de compromiso, hay enlaces con manuales de lenguaje de alto nivel que se pueden utilizar con el control de compromiso.

La duración del bloqueo es muy superior si se especifica LCKLVL(\*ALL) ya que el registro recuperado del archivo se bloquea hasta la siguiente operación de compromiso o retroacción. Otra operación de lectura no lo libera implícitamente y no es posible liberarlo explícitamente.

Otra función que puede establecer un bloqueo en un archivo es la función de salvar mientras activo.

#### **Conceptos relacionados**

Transacciones distribuidas

Salvar el servidor mientras está activo

#### **Referencia relacionada**

“Información relacionada con el control de compromiso” en la página 117

Aquí se proporciona una lista de los manuales del producto y de los libros rojos de IBM (IBM Redbooks), en formato PDF, de los sitios Web y de los temas de Information Center que guardan relación con el control de compromiso. Puede ver o imprimir cualquiera de los PDF.

## **Control de compromiso y agrupaciones de discos independientes**

Cada agrupación de discos independiente y grupo de agrupaciones de discos independientes puede tener una base de datos SQL i5/OS aparte. Se puede emplear el control de compromiso con estas bases de datos.

Sin embargo, dado que cada agrupación de discos independiente o grupo de agrupaciones de discos independientes tiene una base de datos SQL aparte, debe tener en cuenta algunas consideraciones.

#### **Conceptos relacionados**

“Definición de compromiso” en la página 5

Se crea una definición de compromiso cuando se utiliza el mandato Arrancar control de compromiso (STRCMTCTL) para iniciar el control de compromiso en el sistema. Además, DB2 Universal Database (UDB) para iSeries crea automáticamente una definición de compromiso cuando el nivel de aislamiento es distinto de Ningún compromiso.

“Recursos comprometibles locales y remotos” en la página 14

Un recurso comprometible puede ser un recurso local o un recurso remoto.

“Protocolo de compromiso de un recurso comprometible” en la página 16

*Protocolo de compromiso* es la posibilidad que tiene un recurso de participar en el proceso de compromiso de una fase o de dos fases. Los recursos locales, excepto los recursos comprometibles de API, siempre son recursos de dos fases.

## **Consideraciones sobre las agrupaciones de discos independientes en las definiciones de compromiso**

Debe tener presentes las consideraciones sobre definiciones de compromiso presentadas en este tema a la hora de utilizar agrupaciones de discos independientes.

## **Consideraciones sobre la biblioteca QRECOVERY**

Al iniciar el control de compromiso, se crea la definición de compromiso en la biblioteca QRECOVERY. Cada agrupación de discos independiente o grupo de agrupaciones de discos independientes tiene su propia versión de una biblioteca QRECOVERY. En una agrupación de discos independiente, el nombre de la biblioteca QRECOVERY es QRCYxxxx, donde xxxx es el número de la agrupación de discos independiente. Por ejemplo, el nombre de la biblioteca QRECOVERY para la agrupación de discos independiente 39 es QRCY00039. Además, si la agrupación de discos independiente forma parte de un grupo de agrupaciones de discos independientes, solo la agrupación de discos primaria tiene una biblioteca QRCYxxxx.

Al iniciar el control de compromiso, se crea la definición de compromiso en la biblioteca QRECOVERY de la agrupación de discos independiente asociada al trabajo, con lo que se activa el control de compromiso en la agrupación de discos independiente.

## **Consideraciones sobre Establecer grupo de ASP**

La utilización del mandato Establecer grupo de ASP (SETASPGRP) mientras el control de compromiso está activo en una agrupación de discos independiente tiene los efectos siguientes:

- Si conmuta desde una agrupación de discos independiente, y hay recursos registrados con control de compromiso en la agrupación de discos, el mandato SETASPGRP falla con el mensaje CPDB8EC, código de razón 2, La hebra tiene una transacción no comprometida. A continuación de este mensaje se muestra el mensaje CPF8E9.
- Si conmuta desde una agrupación de discos independiente y no hay ningún recurso registrado con control de compromiso, las definiciones de compromiso se mueven a la agrupación de discos independiente a la que conmuta.
- Si conmuta desde la agrupación de discos del sistema (grupo de ASP \*NONE), el control de compromiso no se ve afectado. Las definiciones de compromiso permanecen en la agrupación de discos del sistema.
- Si utiliza un objeto de notificación, el objeto de notificación debe residir en la misma agrupación de discos independiente o el mismo grupo de agrupaciones de discos independientes que la definición de compromiso.
- Si mueve la definición de compromiso a otra agrupación de discos independiente u otro grupo de agrupaciones de discos independientes, el objeto de notificación también debe residir en esa otra agrupación de discos independiente o en ese otro grupo de agrupaciones de discos independientes. El objeto de notificación de la otra agrupación de discos independiente o del otro grupo de agrupaciones de discos independientes se actualiza si la definición de compromiso finaliza de forma anómala. Si el objeto de notificación no se encuentra en la otra agrupación de discos independiente o el otro grupo de agrupaciones de discos independientes, la actualización falla con el mensaje CPF8358.

## **Consideraciones sobre el diario predeterminado**

Debe tener presentes las siguientes consideraciones sobre el diario predeterminado:

- Si utiliza el diario predeterminado, este debe residir en la misma agrupación de discos independiente o grupo de agrupaciones de discos independientes que la definición de compromiso.
- Si el diario predeterminado no se encuentra en la otra agrupación de discos independiente o grupo de agrupaciones de discos independientes cuando se inicia el control de compromiso, el inicio del control de compromiso finaliza de forma anómala con el mensaje CPF9873.

- Si mueve la definición de compromiso a otra agrupación de discos independiente u otro grupo de agrupaciones de discos independientes, el diario predeterminado también debe residir en esa otra agrupación de discos independiente o en ese otro grupo de agrupaciones de discos independientes. Si el diario no se encuentra en la otra agrupación de discos independiente o grupo de agrupaciones de discos independientes, la definición de compromiso se mueve, pero a partir de este momento deja de utilizarse el diario predeterminado.

## Consideraciones sobre la IPL y la desactivación

Debe tener presentes las siguientes consideraciones sobre la IPL y la desactivación:

- La recuperación de las definiciones de compromiso que residen en una agrupación de discos independiente se realiza durante el proceso de activación de la agrupación de discos independiente y es similar a la recuperación de IPL.
- Las definiciones de compromiso de una agrupación de discos independiente no se recuperan durante la IPL del sistema.
- La desactivación de una agrupación de discos independiente tiene los siguientes efectos en las definiciones de compromiso:
  - Los trabajos asociados a la agrupación de discos independiente finalizan.
  - No se permite crear nuevas definiciones de compromiso en la agrupación de discos independiente.
  - Las definiciones de compromiso que residen en la agrupación de discos independiente quedan inutilizables.
  - Las definiciones de compromiso que residen en la agrupación de discos independiente, pero que no están conectadas a un trabajo, liberan los bloqueos cuyo ámbito se circunscribe a la transacción

## Consideraciones sobre la base de datos remota

Debe tener presentes las siguientes consideraciones sobre la base de datos remota:

- No puede utilizar una conexión LU6.2 SNA (conversaciones protegidas o unidad de trabajo distribuida (DUW)) para conectar a una base de datos remota desde una base de datos de agrupación de discos independiente. Puede emplear conversaciones SNA no protegidas para conectar desde una base de datos de agrupación de discos independiente a una base de datos remota.
- Cuando el control de compromiso está activo para un trabajo o una hebra, el acceso a los datos externos a la agrupación de discos independiente o al grupo de agrupaciones de discos independientes al que pertenece la definición de compromiso solo es posible de forma remota, como si fueran datos que residieran en otro sistema. Cuando se emite una sentencia SQL CONNECT para conectar a la base de datos relacional (RDB) de la agrupación de discos independiente, el sistema establece una conexión remota.
- La agrupación de discos del sistema y las agrupaciones de discos básicas no requieren una conexión remota para el acceso solo de lectura a los datos que residen en una agrupación de discos independiente. Del mismo modo, una agrupación de discos independiente no requiere una conexión remota para el acceso solo de lectura a los datos que residen en la agrupación de discos del sistema o una agrupación de discos básica.

### Conceptos relacionados

“Definición de compromiso” en la página 5

Se crea una definición de compromiso cuando se utiliza el mandato Arrancar control de compromiso (STRCMTCTL) para iniciar el control de compromiso en el sistema. Además, DB2 Universal Database (UDB) para iSeries crea automáticamente una definición de compromiso cuando el nivel de aislamiento es distinto de Ningún compromiso.

## Consideraciones sobre las transacciones XA

En el entorno XA, cada base de datos se considera un gestor de recursos aparte. Si un gestor de transacciones pretende acceder a dos bases de datos en la misma transacción, debe emplear los protocolos XA para llevar a cabo el compromiso de dos fases con los dos gestores de recursos.

Dado que cada agrupación de discos independiente es una base de datos SQL aparte, en el entorno XA cada agrupación de discos independiente también se considera un gestor de recursos aparte. Para que un servidor de aplicaciones pueda llevar a cabo una transacción que utiliza dos agrupaciones de discos independientes distintas, el gestor de transacciones debe emplear también un protocolo de compromiso de dos fases.

#### **Conceptos relacionados**

“Soporte de transacciones XA para el control de compromiso” en la página 45

DB2 Universal Database (UDB) para iSeries puede participar en las transacciones globales X/Open.

Agrupaciones de discos independientes

## **Consideraciones y restricciones sobre el control de compromiso**

En este tema se proporcionan diversas consideraciones y restricciones relacionadas con el control de compromiso.

### **Consideraciones sobre archivos de base de datos**

- Si especifica que un archivo compartido debe abrirse bajo control de compromiso, todos los usos posteriores de ese archivo deben abrirse bajo control de compromiso.
- Si se especifica SEQONLY(\*YES) para el archivo abierto solo para lectura con LCKLVL(\*ALL)(ya sea implícitamente o mediante un programa de lenguaje de alto nivel, o explícitamente mediante el mandato Alterar temporalmente con archivo de base de datos (OVRDBF)), el parámetro SEQONLY(\*YES) no se tiene en cuenta y se utiliza SEQONLY(\*NO).
- Los cambios a nivel de registro efectuados bajo control de compromiso se graban en un diario. Estos cambios se pueden aplicar a la base de datos o eliminarse de ella con el mandato Aplicar cambios registrados por diario (APYJRNCHG) o el mandato Eliminar cambios registrados por diario (RMVJRNCHG).
- Las imágenes anteriores y posteriores de los archivos se registran por diario bajo control de compromiso. Si especifica registrar por diario únicamente las imágenes posteriores de los archivos, el sistema también registrará por diario automáticamente la imagen anterior de los cambios de archivo que se hayan producido bajo control de compromiso. Sin embargo, puesto que no se capturan las imágenes anteriores para todos los cambios efectuados en los archivos, no se puede emplear el mandato RMVJRNCHG para estos archivos.

### **Consideraciones sobre los cambios a nivel de objeto y a nivel de registro**

- Los cambios a nivel de objeto y a nivel de registro realizados bajo control de compromiso mediante SQL utilizarán la definición de compromiso que esté activa en este momento para el grupo de activación en el que se ejecuta el programa peticionario. Si no está activa la definición de compromiso a nivel de trabajo ni la definición de compromiso a nivel de grupo de activación, SQL iniciará una definición de compromiso a nivel de grupo de activación.

### **Consideraciones sobre el compromiso de una fase y de dos fases**

- Mientras esté establecida una conversación o conexión remota de una fase, las conversaciones o conexiones remotas con otras ubicaciones no están permitidas. Si se establece un límite de compromiso y se eliminan todos los recursos, la ubicación se puede cambiar.
- Si utiliza el compromiso de dos fases, no es necesario que utilice el mandato Someter mandato remoto (SBMRMTCMD) para iniciar el control de compromiso ni para realizar cualquier otra operación de control de compromiso en las ubicaciones remotas. El sistema realiza estas funciones automáticamente.
- En el caso de una ubicación remota de una fase, los mandatos CL COMMIT y ROLLBACK fallarán si SQL está en la pila de llamadas y la base de datos relacional remota no está en un sistema. Si SQL no está en la pila de llamadas, los mandatos COMMIT y ROLLBACK no fallarán.
- Para una ubicación remota de una fase, el control de compromiso debe iniciarse en el sistema origen antes de efectuar cambios que se puedan comprometer en los recursos remotos. El sistema inicia automáticamente el control de compromiso para SQL de base de datos distribuido en el sistema origen

en el momento de la conexión si el programa SQL se ejecuta con una opción de control de compromiso distinta de \*NONE. Cuando se coloca el primer recurso remoto bajo control de compromiso, el sistema inicia el control de compromiso en el sistema destino.

## Consideración sobre la operación de salvar

Se impide la realización de una operación de salvar si el trabajo que lleva a cabo la operación de salvar tiene una o varias definiciones de compromiso activas con alguno de los siguientes tipos de cambios comprometibles:

- Un cambio de registro en un archivo que reside en la biblioteca que se salva. Para los archivos lógicos, se comprueban todos los archivos físicos relacionados.
- Cualquier cambio a nivel de objeto en una biblioteca que se salva.
- Cualquier recurso de API añadido con la API Añadir recurso de compromiso (QTNADDCR) y con el campo de permitir proceso de salvar normal establecido en el valor predeterminado N.

Esto impide que las operaciones de salvar salven en los medios de salvar los cambios que se deban a una transacción parcial.

**Nota:** Si utiliza la nueva función de salvar con transacciones parciales, el objeto se puede salvar sin finalizar una definición de compromiso.

Los bloqueos de objetos y los bloqueos de registros impiden que se salven en los medios de salvar los cambios pendientes de definiciones de compromiso de otros trabajos. Esto solo se cumple para los recursos de compromiso de API si se obtienen bloqueos cuando se efectúan cambios en los objetos asociados al recurso de compromiso de API.

## Consideraciones y restricciones varias

- Antes de actualizar el sistema a una nueva versión, todas las resincronizaciones pendientes deben completarse o cancelarse.
- Los valores COMMIT y ROLLBACK se muestran en el campo de función de WRKACTJOB durante un compromiso o una retroacción. Si la función sigue siendo COMMIT o ROLLBACK durante mucho tiempo, puede haberse producido una de las situaciones siguientes:
  - Una anomalía de recurso durante el compromiso o la retroacción exige la resincronización. El control no volverá a la aplicación hasta que la resincronización se complete o se cancele.
  - Este sistema ha emitido un voto de solo lectura durante el compromiso. El control no volverá a la aplicación hasta que el sistema que ha iniciado el compromiso envíe datos a este sistema.
  - Este sistema ha emitido un voto de aceptar exclusión durante el compromiso. El control no volverá a la aplicación hasta que el sistema que ha iniciado el compromiso envíe datos a este sistema.

### Conceptos relacionados

Asegurar la integridad del compromiso de dos fases

“Nivel de bloqueo de compromiso” en la página 54

El valor que especifique para el parámetro LCKLVL en el mandato Arrancar control de compromiso (STRCMTCTL) se convertirá en el nivel de bloqueo de registros predeterminado para los archivos de base de datos que se abran y se coloquen bajo control de compromiso para la definición de compromiso.

### Referencia relacionada

Mandato Alterar temporalmente con archivo de base de datos (OVRDBF)

Mandato Aplicar cambios registrados por diario (APYJRNCHG)

Mandato Eliminar cambios registrados por diario (RMVJRNCHG)

Programación SQL

Mandato Someter mandato remoto (SBMRMTCMD)



Mandato COMMIT

Mandato ROLLBACK

API Añadir recurso de compromiso (QTNADDCR)

## Control de compromiso en las aplicaciones por lotes

Las aplicaciones por lotes pueden necesitar o no el control de compromiso. En algunos casos, una aplicación por lotes puede llevar a cabo una sola función de leer un archivo de entrada y actualizar un archivo maestro. Sin embargo, puede emplear el control de compromiso para este tipo de aplicación si es importante volver a iniciarla tras una finalización anómala.

El archivo de entrada es un archivo de actualización con un código en los registros para indicar que se ha procesado un registro. Este archivo y los archivos actualizados se colocan bajo control de compromiso. Cuando el código está presente en el archivo de entrada, representa una transacción completada. El programa lee el archivo de entrada y elude los registros con el código de finalización. Esto permite emplear la misma lógica de programa para las condiciones normales y de reinicio.

Si la aplicación por lotes contiene registros de entrada que dependen unos de otros y contiene conmutadores o totales, se puede emplear un objeto de notificación para proporcionar información sobre el reinicio. Los valores del objeto de notificación se utilizan para volver a iniciar el proceso desde la última transacción comprometida del archivo de entrada.

Si los registros de entrada dependen unos de otros, pueden procesarse como una transacción. Un trabajo por lotes puede bloquear un máximo de 500 000 000 de registros. Puede reducir este límite utilizando un archivo de opciones de consulta (QAQQINI). Utilice el parámetro QRYOPLIB del mandato Cambiar atributos de consulta (CHGQRYA) para especificar un archivo de opciones de consulta que deberá utilizar un trabajo. Utilice el valor COMMITMENT\_CONTROL\_LOCK\_LEVEL del archivo de opciones de consulta como límite de bloqueos para el trabajo.

Los ciclos de compromiso que superen los 2000 bloqueos ralentizarán probablemente el rendimiento del sistema de forma notable. Estas mismas consideraciones sobre bloqueos también atañen a las aplicaciones interactivas, pero la cantidad de tiempo que los registros están bloqueados en una aplicación por lotes podría ser menos importante que en las aplicaciones interactivas.

### Conceptos relacionados

“Objeto de notificación de compromiso” en la página 52

*Objeto de notificación* es una cola de mensajes, un área de datos o un archivo de base de datos que contiene información que identifica la última transacción satisfactoria llevada a cabo para una definición de compromiso determinada si esa definición de compromiso no ha finalizado con normalidad.

“Gestionar el tamaño de las transacciones” en la página 74

Otra manera de minimizar los bloqueos de registros consiste en gestionar el tamaño de la transacción.

### Referencia relacionada

Mandato Cambiar atributos de consulta (CHGQRYA)

## Control de compromiso de dos fases

El control de compromiso de dos fases garantiza que los recursos comprometibles de múltiples sistemas permanezcan sincronizados.

El programa bajo licencia i5/OS soporta el compromiso de dos fases en conformidad con la arquitectura SNA LU 6.2. Para obtener información más detallada sobre los protocolos internos empleados por el sistema para el compromiso de dos fases, consulte *SNA Transaction Programmer's Reference for LU Type 6.2, GC30-3084-05*. Todos los releases soportados de i5/OS permiten utilizar los protocolos Presumed Nothing de SNA LU 6.2 y los protocolos Presumed Abort de SNA LU 6.2.

El compromiso de dos fases también está soportado mediante TCP/IP como protocolo de unidad de trabajo distribuida (DUW) *Distributed Relational Database Architecture (DRDA)*. Para utilizar conexiones de DUW TCP/IP, todos los sistemas (el peticionario de aplicación y el servidor de aplicaciones) deben utilizar la versión V5R1M0 o posterior. Hallará más información sobre DRDA en Open Group Technical Standard, *DRDA V2 Vol. 1: Distributed Relational Database Architecture*, en el sitio Web de Open Group.

Bajo el compromiso de dos fases, el sistema efectúa la operación de compromiso en dos ondas:

- Durante la *onda de preparación*, un gestor de recursos emite una petición de compromiso a su gestor de transacciones. El gestor de transacciones informa a los demás recursos que gestiona y a los demás gestores de transacciones de que la transacción está preparada para comprometerse. Todos los gestores de recursos deben responder que están preparados para comprometer. Esto es lo que se denomina el *voto*.
- Durante la *onda comprometida*, el gestor de transacciones que inicia la petición de compromiso decide qué hacer, según el resultado de la onda de preparación. Si la onda de preparación se completa satisfactoriamente y todos los participantes votan que están preparados, el gestor de transacciones indica a todos los recursos que gestiona y a los demás gestores de transacciones que comprometan la transacción. Si la onda de preparación no se completa satisfactoriamente, se indica a todos los gestores de transacciones y gestores de recursos que retrotraigan la transacción.

## Operaciones de compromiso y retrotracción con recursos remotos

Cuando hay recursos remotos bajo control de compromiso, el iniciador envía una petición de compromiso a todos los agentes remotos. La petición se envía por la red del programa de transacciones. Cada agente responde con el resultado de la operación de compromiso.

Si se producen errores en la onda de preparación, el iniciador envía una petición de retrotracción a todos los agentes. Si se producen errores en la onda comprometida, el sistema intenta establecer en estado comprometido el máximo número de ubicaciones posible. Estos intentos podrían provocar un estado heurístico mixto. Consulte Estados de la transacción para el control de compromiso de dos fases para obtener más información sobre los estados posibles.

Los errores que se produzcan se envían al iniciador, donde se señalan para el usuario. Si se ha especificado un diario predeterminado en el mandato Arrancar control de compromiso (STRCMTCTL), se escriben entradas C LW. Si se producen errores, estas entradas se escriben aunque se haya especificado OMTJRNE(\*LUWID). Puede emplear estas entradas, junto con los mensajes de error y la información de estado de la definición de compromiso, para intentar sincronizar manualmente los recursos comprometibles.

Cuando hay recursos remotos bajo control de compromiso, el iniciador envía una petición de retrotracción a todos los agentes remotos. La petición se envía por la red de programas de transacciones. Cada agente responde con el resultado de la operación de retrotracción.

### Conceptos relacionados

Sitio Web de Open Group

### Referencia relacionada

Mandato Arrancar control de compromiso (STRCMTCTL)

## Roles del proceso de compromiso

Si en una operación de compromiso de una transacción interviene más de un gestor de recursos, cada gestor de recursos asume un rol en la transacción. Un gestor de recursos es el encargado de comprometer o retrotraer los cambios efectuados durante la transacción.

Los gestores de recursos por tipo de recurso son:

**FILE** Gestor de base de datos

**DDM** Gestor de base de datos

DDL Gestor de base de datos

DRDA

Programa de transacciones de comunicaciones

LU62 Programa de transacciones de comunicaciones

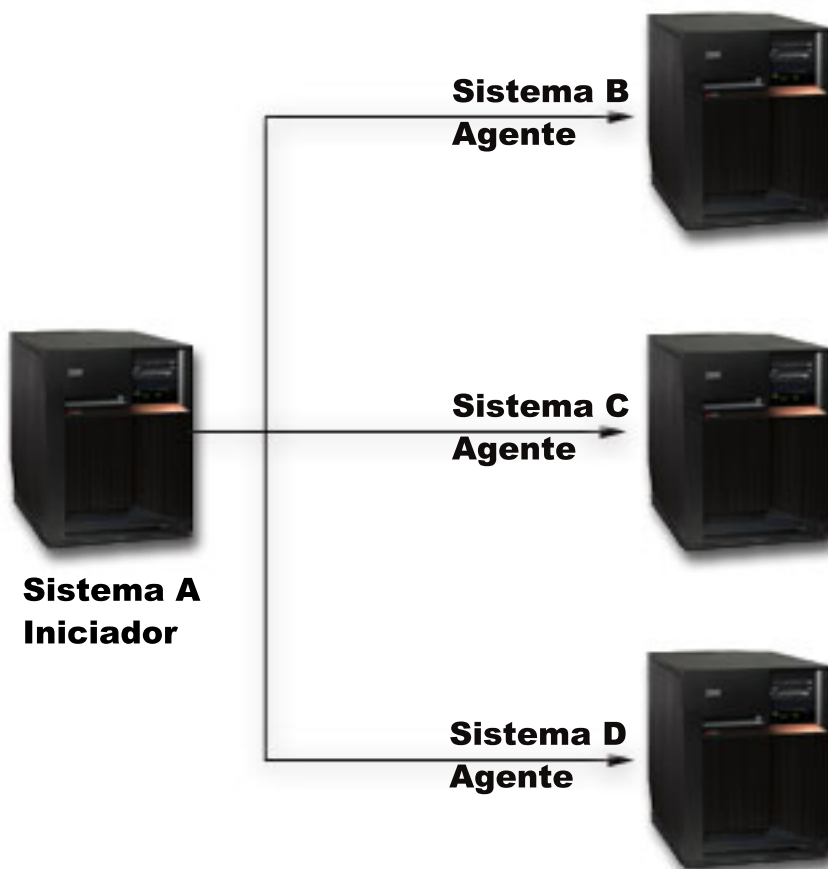
API Programa de salida de API

Las figuras siguientes muestran los roles básicos de una transacción. La estructura que ilustran las figuras se denomina *red de programas de transacciones*. La estructura puede tener la forma de un árbol de un solo nivel o de un árbol multinivel.

### Roles del proceso de compromiso de dos fases: árbol de un solo nivel

Cuando una aplicación del sistema A emite una petición de compromiso, el gestor de recursos del sistema A se convierte en el *iniciador*. En el caso de las conexiones de unidad de trabajo distribuida DRDA por TCP/IP, el iniciador se llama *coordinador*.

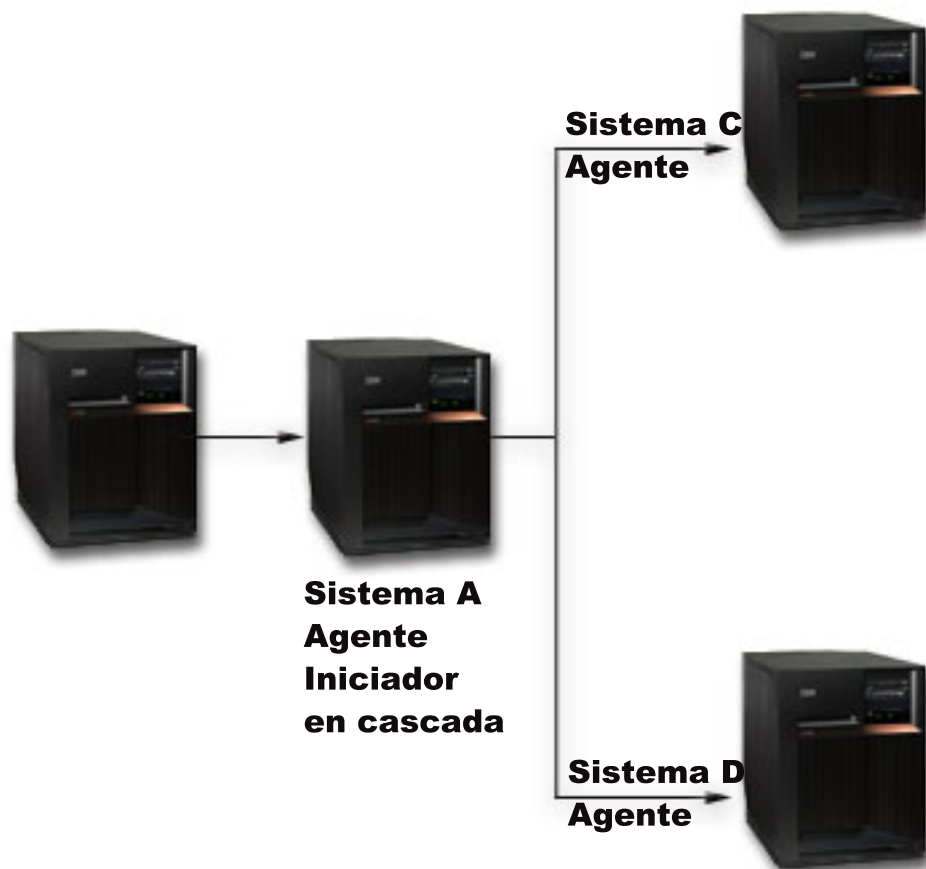
Los gestores de recursos de los tres sistemas restantes (B, C y D) pasan a ser los *agentes* de esta transacción. En el caso de las conexiones de unidad de trabajo distribuida DRDA por TCP/IP, los agentes se llaman a veces *participantes*.



## Roles del proceso de compromiso de dos fases: árbol multinivel

Si la aplicación utiliza las comunicaciones APPC para llevar a cabo el compromiso de dos fases, la relación entre los sistemas puede cambiar de una transacción a la próxima. La siguiente figura muestra los mismos sistemas cuando una aplicación del sistema B emite la petición de compromiso. Esta configuración se representa en un árbol multinivel.

Los roles de esta figura no son válidos para las conexiones de unidad de trabajo distribuida DRDA por TCP/IP, porque en ellas no se pueden utilizar árboles de transacciones multinivel.



La red del programa de transacciones tiene otro nivel porque el sistema B no se comunica directamente con los sistemas C y D. Ahora, el gestor de recursos del sistema A tiene el rol de agente y el rol de iniciador en cascada.

Para mejorar el rendimiento de las transacciones de dos fases LU 6.2, el iniciador podría asignar el rol de último agente a uno de los agentes. El último agente no participa en la onda de preparación. En la onda comprometida, el último agente es el que compromete en primer lugar. Si el último agente no compromete satisfactoriamente, el iniciador indica a los otros agentes que retrotraigan.

En el caso de las conexiones de unidad de trabajo distribuida DRDA por TCP/IP, el coordinador podría asignar el rol de servidor de resincronización a un participante. El servidor de resincronización es el encargado de resincronizar los demás participantes en el caso de producirse una anomalía de comunicaciones con el coordinador, o si el coordinador tiene una anomalía del sistema.

### Conceptos relacionados

“Definición de compromiso para el compromiso de dos fases: permitir voto de solo lectura” en la página 34

Normalmente un gestor de transacciones participa en las dos fases del proceso de compromiso. Para mejorar el rendimiento del proceso de compromiso, puede configurar algunas o la totalidad de las ubicaciones de una transacción a fin de permitir al gestor de transacciones que emita un voto de solo lectura.

### Estados de la transacción para el control de compromiso de dos fases

Se establece una definición de compromiso en cada ubicación que forma parte de la red de programas de transacciones. Para cada definición de compromiso, el sistema mantiene un seguimiento del estado de la transacción actual y la transacción anterior.

El sistema emplea el estado para decidir si hay que comprometer o retrotraer una transacción si esta se interrumpe como consecuencia de una anomalía de las comunicaciones o del sistema. Si son múltiples las ubicaciones que participan en una transacción, los estados de las transacciones de cada ubicación se pueden comparar para determinar la acción correcta (compromiso o retrotracción). Este proceso de comunicación entre las ubicaciones para determinar la acción correcta se llama **resincronización**.

La tabla siguiente muestra:

- Los estados básicos que pueden producirse durante una transacción.
- Los estados adicionales que podrían tener lugar.
- Si un estado requiere resincronización en el caso de que se interrumpa la transacción como consecuencia de una anomalía de las comunicaciones o del sistema. Los valores posibles son:

**No es necesaria**

Cada ubicación puede tomar la decisión correcta de forma independiente.

**Puede ser necesaria**

Cada ubicación puede tomar la decisión correcta, pero podría ser necesario informar al iniciador de la decisión.

**Es necesaria**

Para poder tomar la decisión correcta, primero hay que determinar el estado de cada ubicación.

- Acción tomada debido a una anomalía de las comunicaciones o del sistema.

Nombre del estado	Descripción	Resincronización si se interrumpe la transacción	Acción tomada debido a una anomalía de comunicaciones o del sistema
<b>Estados básicos durante el proceso de compromiso de dos fases:</b>			
Restablecer (RST)	Desde el límite de compromiso hasta que un programa emite una petición de compromiso o retrotracción.	No es necesaria.	Los cambios pendientes se retrotraen.
Preparar en curso (PIP)	El iniciador ha iniciado la onda de preparación. Todavía no han votado todas las ubicaciones.	Puede ser necesaria.	Los cambios pendientes se retrotraen.
Preparado (PRP)	Esta ubicación y todas las ubicaciones situadas más abajo en la red de programa de transacciones han votado comprometer. Esta ubicación todavía no ha recibido una notificación del iniciador para comprometer.	Es necesaria.	Pendiente. Depende del resultado del proceso de resincronización.

Nombre del estado	Descripción	Resincronización si se interrumpe la transacción	Acción tomada debido a una anomalía de comunicaciones o del sistema
Comprometer en curso (CIP)	Todas las ubicaciones han votado comprometer. El iniciador ha iniciado la onda comprometida.	Es necesaria.	Los cambios pendientes se comprometen. La resincronización se lleva a cabo para garantizar que todas las ubicaciones han comprometido. Si otra ubicación informa de una retrotracción heurística, se notifica un error.
Comprometido (CMT)	Todos los agentes han comprometido y devuelto una respuesta a este nodo.	Puede ser necesaria.	Ninguna.
<b>Estados adicionales durante el proceso de compromiso de dos fases:</b>			
Pendiente de último agente (LAP)	Si se selecciona un último agente, este estado se produce en el iniciador entre el estado PIP y el estado CIP. El iniciador ha indicado al último agente que comprometa y aún no ha recibido una respuesta.	Es necesaria.	Pendiente. Depende del resultado del proceso de resincronización.
Voto de solo lectura (VRO)	Este agente ha respondido a la onda de preparación indicando que no tiene cambios pendientes. Si se permite el estado de voto de solo lectura, este agente no se incluye en la onda comprometida.	Puede ser necesaria.	Ninguna.
Retrotracción necesaria (RBR)	<p>Se ha producido uno de estos eventos:</p> <ul style="list-style-type: none"> <li>• Un agente ha emitido una petición de retrotraer antes de la operación de compromiso.</li> <li>• Se ha producido una anomalía en la transacción.</li> <li>• Se ha utilizado la API QTNRBRQD para colocar la transacción en un estado de retrotracción necesaria.</li> </ul> <p>El programa de transacciones no tiene permiso para efectuar ningún cambio adicional bajo control de compromiso.</p>	Puede ser necesaria.	Los cambios pendientes se retrotraen.
<b>Condiciones que se producen debido a acciones del operador o a errores:</b>			

Nombre del estado	Descripción	Resincronización si se interrumpe la transacción	Acción tomada debido a una anomalía de comunicaciones o del sistema
Retrotracción forzada	Esta ubicación y todas las ubicaciones situadas más abajo en la red del programa de transacciones, excepto el último agente, se han retrotraído mediante la intervención del operador.	Puede ser necesaria.	Los cambios pendientes ya se han retrotraído.
Compromiso forzado	Esta ubicación y todas las ubicaciones situadas más abajo en la red del programa de transacciones, excepto el último agente, han comprometido mediante la intervención del operador.	Puede ser necesaria.	Los cambios pendientes ya se han comprometido.
Heurística mixta (HRM)	Algunos gestores de recursos han comprometido. Otros han retrotraído. Se ha utilizado la intervención del operador o se ha producido un error del sistema. La opción de heurística mixta no aparece como un estado en las pantallas de definición de compromiso. Se envían mensajes de notificación al operador.	Puede ser necesaria.	El operador debe llevar a cabo una operación de restauración en todas las ubicaciones participantes para colocar la base de datos en un estado coherente.

### Conceptos relacionados

“Definición de compromiso para el compromiso de dos fases: permitir voto de solo lectura” en la página 34

Normalmente un gestor de transacciones participa en las dos fases del proceso de compromiso. Para mejorar el rendimiento del proceso de compromiso, puede configurar algunas o la totalidad de las ubicaciones de una transacción a fin de permitir al gestor de transacciones que emita un voto de solo lectura.

“Definición de compromiso para el compromiso de dos fases: no esperar resultado” en la página 36  
 Cuando se produce una anomalía de comunicaciones o del sistema durante una operación de compromiso que exija una resincronización, el valor predeterminado consiste en esperar a que finalice la resincronización antes de que se complete la operación de compromiso.

“Iniciar el control de compromiso” en la página 51

Para iniciar el control de compromiso, utilice el mandato Arrancar control de compromiso (STRCMTCTL).

“Recuperación de control de compromiso durante la carga del programa inicial tras una finalización anómala” en la página 67

Cuando se lleva a cabo una carga del programa inicial (IPL) después de que el sistema finalice de forma anómala, el sistema intenta recuperar todas las definiciones de compromiso que estaban activas al finalizar el sistema.

“Errores de control de compromiso” en la página 105

Al utilizar el control de compromiso, es importante entender qué condiciones pueden ocasionar errores y cuáles no.

## Definiciones de compromiso para el control de compromiso de dos fases

Una vez que haya iniciado el control de compromiso, puede emplear la API Cambiar opciones de compromiso (QTNCHGCO) para cambiar las opciones de compromiso de la transacción.

Según el entorno y las aplicaciones que utilice, al cambiar las opciones de compromiso puede mejorar el rendimiento del sistema.

**Nota:** Si utiliza una conexión de unidad de trabajo distribuida DRDA por TCP/IP, la única opción válida es Permitir voto de solo lectura.

### Referencia relacionada

API Cambiar opciones de compromiso (QTNCHGCO)

### Definición de compromiso para el compromiso de dos fases: permitir voto de solo lectura:

Normalmente un gestor de transacciones participa en las dos fases del proceso de compromiso. Para mejorar el rendimiento del proceso de compromiso, puede configurar algunas o la totalidad de las ubicaciones de una transacción a fin de permitir al gestor de transacciones que emita un voto de solo lectura.

Si la ubicación no tiene cambios para comprometer durante una transacción, el gestor de transacciones emite un voto de solo lectura durante la onda de preparación. La ubicación no participa en la onda comprometida. De este modo se mejora el rendimiento global ya que los flujos de comunicaciones que normalmente se producen durante la onda comprometida se eliminan en las transacciones en las que no se efectúa ninguna actualización en una o varias ubicaciones remotas.

Una vez que inicie el control de compromiso, puede emplear la API Cambiar opciones de compromiso (QTNCHGCO) para cambiar la opción Voto de solo lectura permitido, dándole el valor Y. Puede interesarle hacer este cambio si se dan las siguientes circunstancias:

- Uno o varios sistemas remotos a menudo no tienen cambios que se puedan comprometer para una transacción.
- Una transacción no depende de dónde ha establecido el cursor del archivo (el siguiente registro) la transacción anterior. Cuando una ubicación emite un voto de solo lectura, nunca se informa a la aplicación si la transacción se retrotrae. La ubicación ha comprometido las operaciones de lectura efectuadas en los archivos de base de datos y, por consiguiente, ha movido la posición del cursor. La posición del cursor del archivo acostumbra a ser importante únicamente si se efectúan procesos secuenciales.

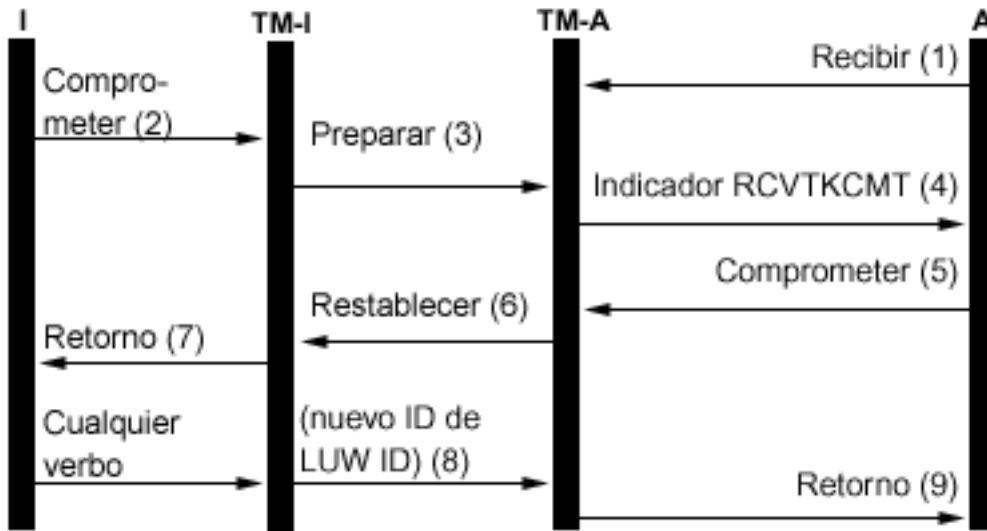
Si la definición de compromiso está configurada para permitir emitir un voto de solo lectura, la aplicación espera al siguiente flujo de mensajes procedente de otra ubicación.

La opción Voto de solo lectura permitido está pensada para las aplicaciones de tipo cliente/servidor. Si la finalidad del programa A solo es satisfacer las peticiones efectuadas por el programa I (no llevar a cabo trabajo independiente), es adecuado utilizar la opción Voto de solo lectura permitido para el programa A.

### Flujo del proceso de compromiso sin optimización del último agente cuando el agente emite un voto de solo lectura

La figura siguiente muestra el flujo de mensajes entre los programas de aplicación y los gestores de transacciones cuando un programa de aplicación emite una instrucción de compromiso sin optimización del último agente cuando el agente emite un voto de solo lectura. Ni el programa de aplicación del iniciador ni los programas de aplicación del agente tienen conocimiento del proceso de compromiso de dos fases. Los números entre paréntesis () de la figura corresponden a los elementos numerados en la descripción que se facilita a continuación.





### Descripción

- I** = Iniciador (la aplicación que inicia la petición de compromiso)
- TM-I** = El gestor de transacciones para el iniciador
- A** = Agente (la aplicación que recibe la petición de compromiso)
- TM-A** = El gestor de transacciones para el agente

La siguiente lista proporciona una descripción de los eventos del proceso normal sin optimización de último agente cuando el agente emite un voto de solo lectura. Aquí se describe un flujo básico. La secuencia de los eventos puede ser mucho más compleja si la red del programa de transacciones tiene varios niveles o se producen errores.

1. El programa de aplicación A recibe una petición en la que se le solicita que indique que está preparado para recibir una petición del programa I.
2. La aplicación del iniciador (I) emite una instrucción de compromiso.
3. El gestor de transacciones del iniciador (TM-I) asume el rol de iniciador para esta transacción. Inicia la onda de preparación enviando un mensaje de preparación a todas las demás ubicaciones que participan en la transacción.
4. Los gestores de transacciones de todas las demás ubicaciones asumen el rol de agente (TM-A). El TM-A notifica al programa de aplicación A que se ha recibido una petición de compromiso. Para los archivos ICF, la notificación tiene el formato del indicador de recepción de toma de compromiso (RCVTKCMT) de ICF que se activa.
5. El programa de aplicación A responde emitiendo una instrucción de compromiso (o una instrucción de retrotracción). Este es el voto del programa de aplicación.
6. Si el programa de aplicación A ha utilizado la API Cambiar opciones de compromiso (QTNCHGCO) para establecer que la opción Voto de solo lectura permitido sea igual a Y, y no se han hecho cambios en el agente durante la transacción, el agente (TM-A) responde al iniciador (TM-I) con un mensaje de restablecimiento. No hay ninguna onda comprometida para el agente.
7. Se envía información de retorno al programa de aplicación (A) para indicar que la transacción se ha completado en el agente TM-A.
8. La próxima vez que el iniciador (TM-I) emita algún mensaje al agente (TM-A) (tanto si es un flujo de datos como si es una instrucción de compromiso), el TM-I hará que su ID de transacción actual se envíe con el mensaje. Esto se debe a que podría haberse generado un nuevo ID de transacción en el TM-I si se ha producido una anomalía de comunicaciones entre el TM-I y otro sistema durante la operación de compromiso.

9. Se envía información de retorno al programa de aplicación (A) para indicar que la transacción se ha completado en el agente TM-A. El retorno se retarda hasta que se recibe el siguiente mensaje ya que debe recibirse un nuevo ID de transacción procedente del TM-I antes de que la aplicación A pueda iniciar la siguiente transacción.

#### **Conceptos relacionados**

“Roles del proceso de compromiso” en la página 28

Si en una operación de compromiso de una transacción interviene más de un gestor de recursos, cada gestor de recursos asume un rol en la transacción. Un gestor de recursos es el encargado de comprometer o retrotraer los cambios efectuados durante la transacción.

“Estados de la transacción para el control de compromiso de dos fases” en la página 31

Se establece una definición de compromiso en cada ubicación que forma parte de la red de programas de transacciones. Para cada definición de compromiso, el sistema mantiene un seguimiento del estado de la transacción actual y la transacción anterior.

“Optimizar el rendimiento para el control de compromiso” en la página 70

Para utilizar el control de compromiso se necesitan recursos que pueden afectar al rendimiento del sistema. Son varios los factores que inciden en el rendimiento del sistema en relación con el control de compromiso.

#### **Referencia relacionada**

API Cambiar opciones de compromiso (QTNCHGCO)

#### **Definición de compromiso para el compromiso de dos fases: no esperar resultado:**

Cuando se produce una anomalía de comunicaciones o del sistema durante una operación de compromiso que exija una resincronización, el valor predeterminado consiste en esperar a que finalice la resincronización antes de que se complete la operación de compromiso.

**Nota:** La opción No esperar resultado no es válida si se utiliza una conexión de unidad de trabajo distribuida DRDA por TCP/IP. Las conexiones de unidad de trabajo distribuida DRDA por TCP/IP nunca esperan el resultado.

Plantéese la posibilidad de cambiar este comportamiento cuando se den las condiciones siguientes:

- Las aplicaciones que intervienen son independientes entre sí.
- La lógica del programa no necesita el resultado de las transacciones anteriores para garantizar que los archivos de base de datos permanecen sincronizados.

Después de iniciar el control de compromiso, puede utilizar la API Cambiar opciones de compromiso (QTNCHGCO) para especificar que la definición de compromiso no espere el resultado de la resincronización. Si especifica N (No) para la opción Esperar resultado, el sistema utiliza un trabajo servidor de base de datos (QDBSRVnn) para manejar la resincronización de manera asíncrona.

**Nota:** Estos trabajos servidores de base de datos se inician durante el proceso de IPL. Si cambia las opciones de control de compromiso, esta modificación no tiene ningún efecto en el número de trabajos que inicia el sistema.

Este tema solo hace referencia a dos valores para la opción Esperar resultado resuelta, Y (Sí) y N (No). En realidad hay dos valores más que puede especificar, L (Sí o heredar del iniciador) y U (No o heredar del iniciador). Si utiliza estos valores, el sistema resuelve en Sí o No el valor utilizado realmente en cada operación de compromiso. En el tema que trata de la API QTNCHGCO (Cambiar opciones de compromiso) hay más detalles sobre estos valores.

**Nota:** El valor del iniciador solo lo puede heredar un agente si ambos, el iniciador y el agente, dan soporte a la terminación anómala que se presupone.

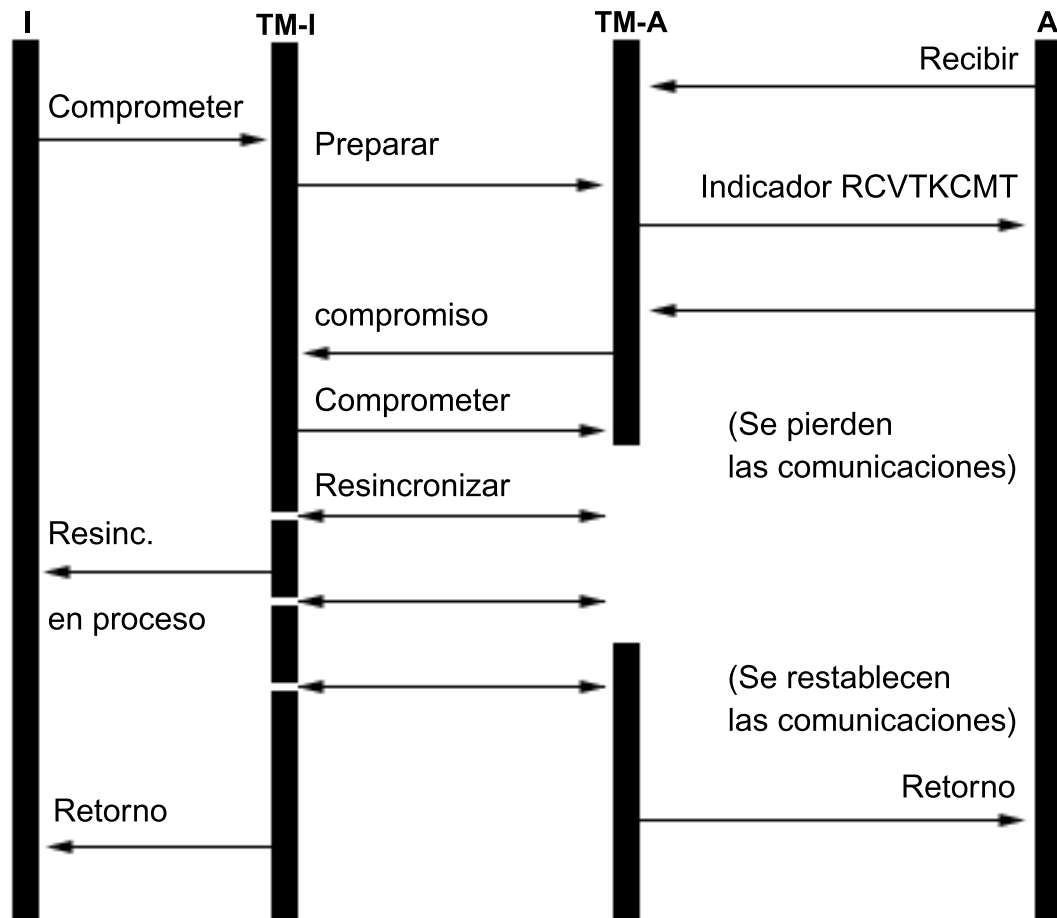
La opción Esperar resultado (WFO) no afecta al proceso de compromiso sin errores normal. Si se produce un error, la opción WFO determina si la aplicación espera o no el resultado de la resincronización, con las condiciones siguientes:

- Si la opción WFO resuelta es Y (Sí), la aplicación espera el resultado de la resincronización.
- Si la opción WFO resuelta es N (No) y se produce una anomalía de comunicaciones durante la onda de preparación o la retrotracción de una ubicación que soporta los protocolos de terminación anómala que se presupone, no se efectúa ninguna resincronización y la definición de compromiso se retrotrae.
- Si la definición de compromiso está pendiente (el estado de la transacción es Preparada o Pendiente del último agente), la aplicación esperará el resultado de la resincronización independientemente del valor de WFO resuelto.
- Si la opción WFO resuelta es N, y no se cumple la segunda condición ni la tercera, el sistema intenta resincronizar una vez. Si no es satisfactoria, el sistema emite el mensaje de estado CPF83E6 a la aplicación para indicar que la resincronización está en curso.

Dado que el mensaje CPF83E6 está relacionado con el estado, solo tiene efecto si la aplicación lo supervisa por si aparece. Normalmente, la aplicación puede tratar este mensaje como informativo. Los sistemas que participan en la transacción intentan resincronizar la transacción hasta que se corrige la anomalía. Estos intentos de resincronización posteriores se llevan a cabo en los trabajos servidores de base de datos. Si un intento de resincronización posterior realizado en un trabajo servidor de base de datos falla, se envía el mensaje CPI83D0 a QSYSOPR.

### **Valor de Esperar resultado: Sí**

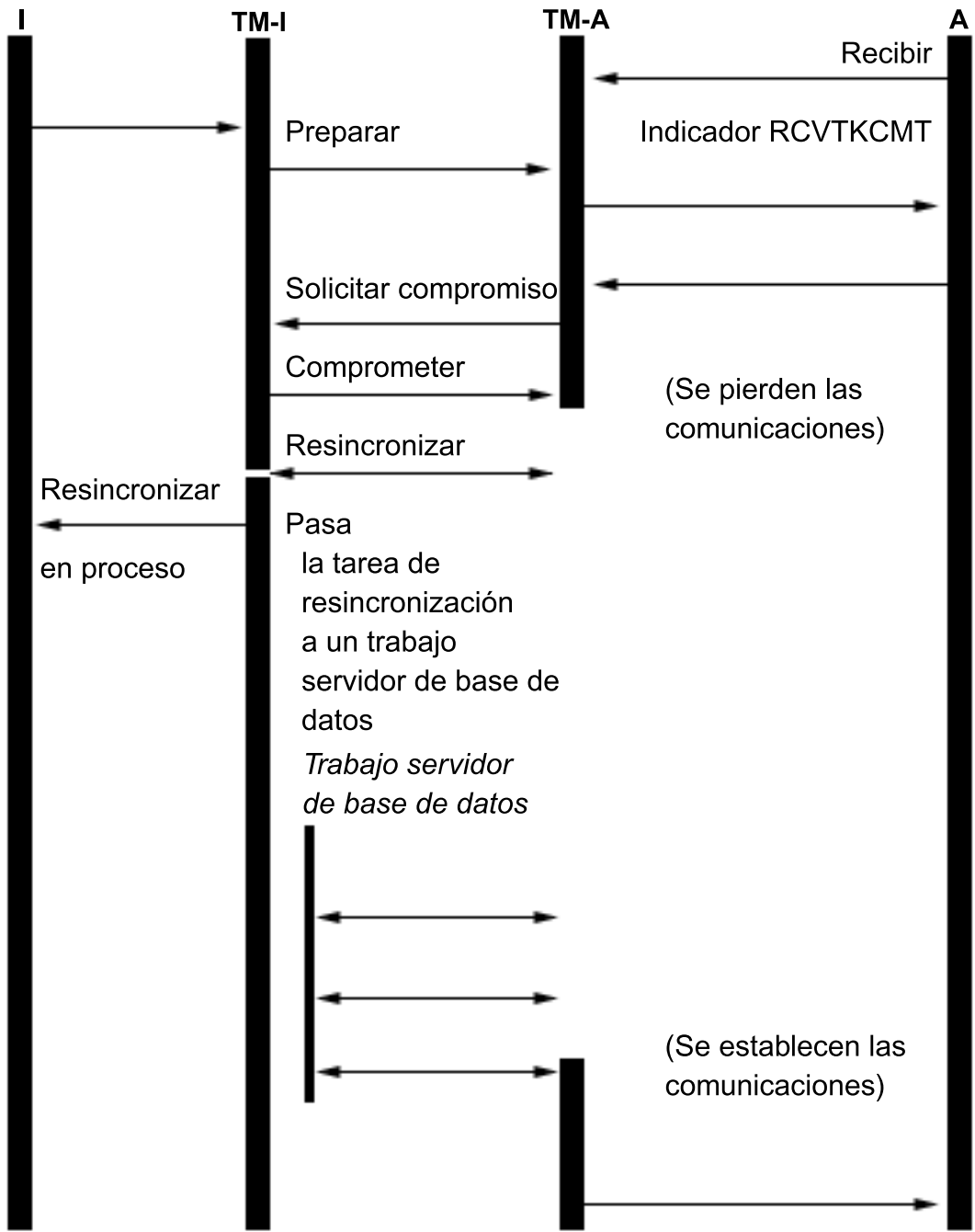
En la figura siguiente, la definición de compromiso del iniciador (I) utiliza el valor predeterminado Y (Sí) para la opción Esperar resultado. Si se pierden las comunicaciones entre el TM-I y el TM-A, las aplicaciones A e I esperan a que se resincronice la transacción.



### Valor de Esperar resultado: No

En la figura siguiente, la definición de compromiso del iniciador tiene la opción WFO resuelta establecida en N (No). El TM-A cumple la condición 3 de la lista anterior, mientras que el TM-I cumple la condición 4. El control se devuelve a la aplicación I después de un intento de resincronizar con el TM-A. Un trabajo servidor de base de datos intenta llevar a cabo la resincronización. La aplicación I nunca recibe el indicador de retorno cuando la petición de compromiso se ha completado satisfactoriamente. El control no se devuelve a la aplicación del agente (A) hasta que se vuelven a establecer las comunicaciones. Esto depende del momento en que se produce la anomalía. En este caso, la anomalía de comunicaciones se produce antes de que se reciba el mensaje de compromiso del iniciador, con lo que el TM-A queda pendiente de si se debe comprometer o retrotraer. Cuando el gestor de transacciones está pendiente, conserva el control hasta que se ha completado la resincronización, independientemente del valor de WFO resuelto en ese sistema.

Si desea que las aplicaciones de todos los sistemas continúen antes de que finalice la resincronización, debe cambiar la opción WFO resuelta a N (No) en todos los sistemas o establecer el iniciador en N y el resto de los sistemas en U (No o heredar del iniciador). Sin embargo, recuerde que la opción WFO resuelta no se tiene en cuenta cuando el gestor de transacciones está pendiente de si se debe comprometer o retrotraer y siempre espera a que finalice la resincronización antes de devolver el control.



Cuando se establece una conexión con una base de datos relacional remota y todavía no se ha iniciado ninguna conversación protegida, el sistema cambia implícitamente el valor de Esperar resultado, dándole el valor N. Esto se debe a que el rendimiento de las operaciones de compromiso mejora cuando el valor de Esperar resultado es N y el sistema remoto soporta la terminación anómala que se presupone. Este cambio implícito del valor de Esperar resultado solo se realiza para las aplicaciones DRDA y DDM. Las aplicaciones APPC utilizan el valor predeterminado de Esperar resultado, que es Y, a menos que llamen a la API QTNCHGCO para cambiarlo.

**Conceptos relacionados**

“Estados de la transacción para el control de compromiso de dos fases” en la página 31  
Se establece una definición de compromiso en cada ubicación que forma parte de la red de programas de transacciones. Para cada definición de compromiso, el sistema mantiene un seguimiento del estado de la transacción actual y la transacción anterior.

“Errores de control de compromiso” en la página 105

Al utilizar el control de compromiso, es importante entender qué condiciones pueden ocasionar errores y cuáles no.

#### **Referencia relacionada**

API Cambiar opciones de compromiso (QTNCHGCO)

#### **Definición de compromiso para el compromiso de dos fases: indicar aceptar exclusión:**

Normalmente, el gestor de transacciones de cada ubicación de la red de programas de transacciones participa en cada operación de compromiso o retrotracción. Para mejorar el rendimiento, puede configurar algunas o la totalidad de las ubicaciones de una transacción a fin de permitir al gestor de transacciones que indique aceptar exclusión.

**Nota:** La opción Indicar aceptar exclusión no es válida si se utiliza una conexión de unidad de trabajo distribuida DRDA por TCP/IP.

Si no se envía ningún flujo de comunicaciones a la ubicación durante una transacción, la ubicación se excluye cuando se efectúa una operación de compromiso o retrotracción. De este modo se mejora el rendimiento global ya que los flujos de comunicaciones que normalmente se producen durante el compromiso o la retrotracción se eliminan en las transacciones en las que no se envía ninguna información a una o varias ubicaciones remotas.

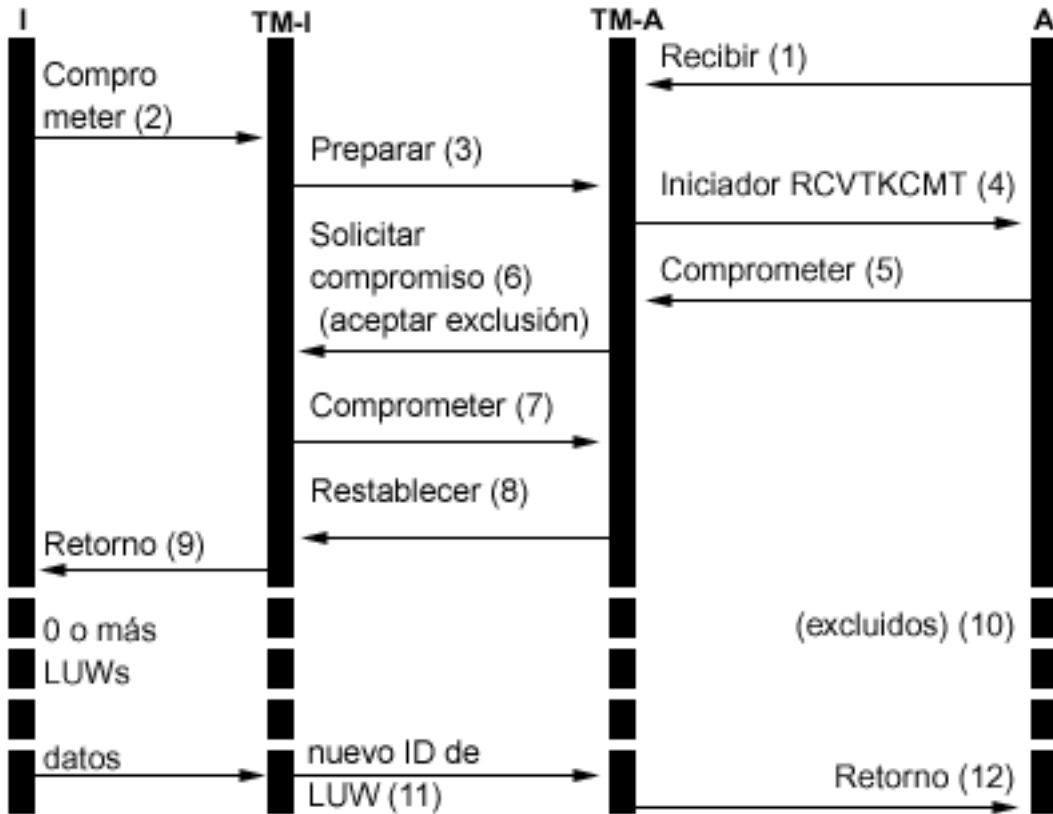
Después de iniciar el control de compromiso, puede emplear la API Cambiar opciones de compromiso (QTNCHGCO) para cambiar la opción Aceptar exclusión, dándole el valor Y (Sí). Puede interesarle hacer esta operación si uno o varios sistemas remotos no intervienen en una transacción.

Si la definición de compromiso está configurada para indicar aceptar exclusión, la aplicación espera al siguiente flujo de mensajes procedente de otra ubicación.

La opción de aceptar exclusión está pensada para las aplicaciones de tipo cliente/servidor. Si la única finalidad del programa A es satisfacer las peticiones procedentes del programa I y no llevar a cabo ningún trabajo independiente, es adecuado permitir la opción de aceptar exclusión para el programa A.

#### **Flujo del proceso de compromiso sin optimización de último agente cuando el agente emite un voto de aceptar exclusión**

La figura siguiente muestra el flujo de mensajes entre los programas de aplicación y los gestores de transacciones cuando un programa de aplicación emite una instrucción de compromiso sin optimización del último agente cuando el agente indica aceptar exclusión. Ni el programa de aplicación del iniciador ni los programas de aplicación de agente tienen conocimiento del proceso de compromiso de dos fases. Los números entre paréntesis () de la figura corresponden a los elementos numerados en la descripción que se facilita a continuación.



**Descripción**

- I** = Iniciador (la aplicación que inicia la petición de compromiso)
- TM-I** = El gestor de transacciones para el iniciador
- A** = Agente (la aplicación que recibe la petición de compromiso)
- TM-A** = El gestor de transacciones para el agente

Esta es una descripción de los eventos del proceso normal sin optimización de último agente cuando el agente emite un voto de aceptar exclusión. Aquí se describe un flujo básico. La secuencia de los eventos puede ser mucho más compleja si la red del programa de transacciones tiene varios niveles o se producen errores.

1. El programa de aplicación A recibe una petición en la que se le solicita que indique que está preparado para recibir una petición del programa I.
2. La aplicación del iniciador (I) emite una instrucción de compromiso.
3. El gestor de transacciones del iniciador (TM-I) asume el rol de iniciador para esta transacción. Inicia la onda de preparación enviando un mensaje de preparación a todas las demás ubicaciones que participan en la transacción.
4. Los gestores de transacciones de todas las demás ubicaciones asumen el rol de agente (TM-A). El TM-A notifica al programa de aplicación A que se ha recibido una petición de compromiso. Para los archivos ICF, la notificación tiene el formato del indicador de recepción de toma de compromiso (RCVTKCMT) de ICF que se activa.
5. El programa de aplicación A responde emitiendo una instrucción de compromiso (o una instrucción de retrotracción). Este es el voto del programa de aplicación.
6. Si el programa de aplicación A ha utilizado la API Cambiar opciones de compromiso (QTNCHGCO) para establecer que la opción de aceptar exclusión sea igual a Y, se envía un indicador cuando el agente (TM-A) responde al iniciador (TM-I) con un mensaje de petición de compromiso.

**Nota:** Los cambios efectuados en la opción de compromiso Aceptar exclusión no entran en vigor hasta la siguiente operación de compromiso satisfactoria.

7. Cuando el iniciador (TM-I) recibe todos los votos, el TM-I envía un mensaje de compromiso. De este modo se inicia la onda comprometida.
8. Cada agente (TM-A) compromete y responde con un mensaje de restablecimiento.
9. Se envía información de retorno al programa de aplicación (I) para indicar que la transacción se ha completado en el iniciador.
10. En el TM-I pueden producirse cualquier número de transacciones, ninguna de las cuales requiere cambios en el TM-A ni en los datos del TM-A. El TM-A no se incluye en estas transacciones.
11. La próxima vez que el iniciador (TM-I) emita un mensaje al agente (A), se enviará un nuevo ID de transacción con el mensaje. Si el iniciador lleva a cabo alguna operación de compromiso o retrotracción antes de enviar un mensaje al agente, no se envía ningún mensaje al agente durante estas operaciones (el agente se excluye de estas operaciones de compromiso o retrotracción). Dado que, mientras el agente estaba excluido, pueden haberse comprometido o retrotraído una o varias transacciones en el iniciador, este debe comunicar su ID de transacción actual cuando se envíe el siguiente mensaje al agente.
12. Se envía información de retorno al programa de aplicación (A) para indicar que el compromiso original se ha completado y que participa en la transacción actual.

#### **Conceptos relacionados**

“Optimizar el rendimiento para el control de compromiso” en la página 70

Para utilizar el control de compromiso se necesitan recursos que pueden afectar al rendimiento del sistema. Son varios los factores que inciden en el rendimiento del sistema en relación con el control de compromiso.

#### **Definición de compromiso para el compromiso de dos fases: no seleccionar un último agente:**

Por omisión, el gestor de transacciones del iniciador tiene libertad para seleccionar cualquier agente como último agente durante una operación de compromiso.

**Nota:** La opción No seleccionar un último agente no es válida si se utiliza una conexión de unidad de trabajo distribuida DRDA por TCP/IP.

En el caso de un árbol de varios niveles, cualquier agente seleccionado como último agente por su iniciador también tiene libertad para seleccionar un último agente propio. El rendimiento mejora cuando se selecciona un último agente durante la operación de compromiso ya que se eliminan dos flujos de comunicaciones entre un iniciador y su último agente (la fase de preparación se elimina para estos sistemas).

No obstante, cuando el iniciador envía la petición de compromiso a su último agente, debe esperar a recibir el voto del último agente para continuar. Este proceso es independiente del valor de Esperar resultado de la definición de compromiso. En el proceso de compromiso sin errores normal esto no es un problema. Sin embargo, si se produce un error a lo largo de esta ventana, el iniciador no puede continuar hasta que se completa la resincronización. Si la aplicación del iniciador maneja peticiones de un usuario en un terminal, puede ser una consideración de uso.

Plantéese si el rendimiento mejorado durante las operaciones de compromiso normales es más importante que el impacto en el uso cuando se produce un error de este tipo. Tenga en cuenta que si se produce el error antes de que se envíe la petición de compromiso al último agente, la LUW llevará a cabo la retrotracción de inmediato y el iniciador no esperará. Por consiguiente, la ventana que un error puede hacer que el iniciador espere es bastante reducida, por lo que un error de este tipo es poco habitual.



Si decide que el impacto en el uso no vale el rendimiento mejorado, puede cambiar las definiciones de compromiso para no seleccionar un último agente. Una vez que inicie el control de compromiso, puede emplear la API Cambiar opciones de compromiso (QTNCHGCO) para cambiar la opción último agente permitido, dándole el valor N.

#### Referencia relacionada

API Cambiar opciones de compromiso (QTNCHGCO)

#### Efecto del voto fiable en el flujo del proceso de compromiso:

El voto fiable es una optimización que mejora el rendimiento al volver antes a la aplicación del iniciador tras una operación de compromiso y eliminar un mensaje durante una operación de compromiso.

No existe una optimización de voto fiable explícita para las conexiones de unidad de trabajo distribuida DRDA por TCP/IP. Sin embargo, i5/OS nunca solicita una confirmación de restablecimiento (olvido) para las conexiones TCP/IP. Por lo tanto, siempre hay un restablecimiento (olvido) implícito para las conexiones TCP/IP.

Después de iniciar el control de compromiso, puede utilizar la API Cambiar opciones de compromiso (QTNCHGCO) para cambiar la opción Aceptar voto fiable, dándole el valor Y.

Puede definirse el voto fiable como una promesa efectuada por un agente a su iniciador de que no se tomará ninguna decisión heurística en el agente si se produce una anomalía de comunicaciones mientras el agente está pendiente. Un agente que utiliza la optimización de voto fiable envía un indicador al iniciador durante la onda de preparación del compromiso. Si el iniciador también emplea la optimización de voto fiable, a continuación envía un indicador al agente para señalar que no es necesario el restablecimiento en respuesta al mensaje de compromiso. De este modo se elimina el mensaje de restablecimiento y se permite al gestor de transacciones volver a la aplicación del iniciador tan pronto como se envía el mensaje de compromiso.

Plantéese la posibilidad de utilizar la optimización de voto fiable si se dan las condiciones siguientes:

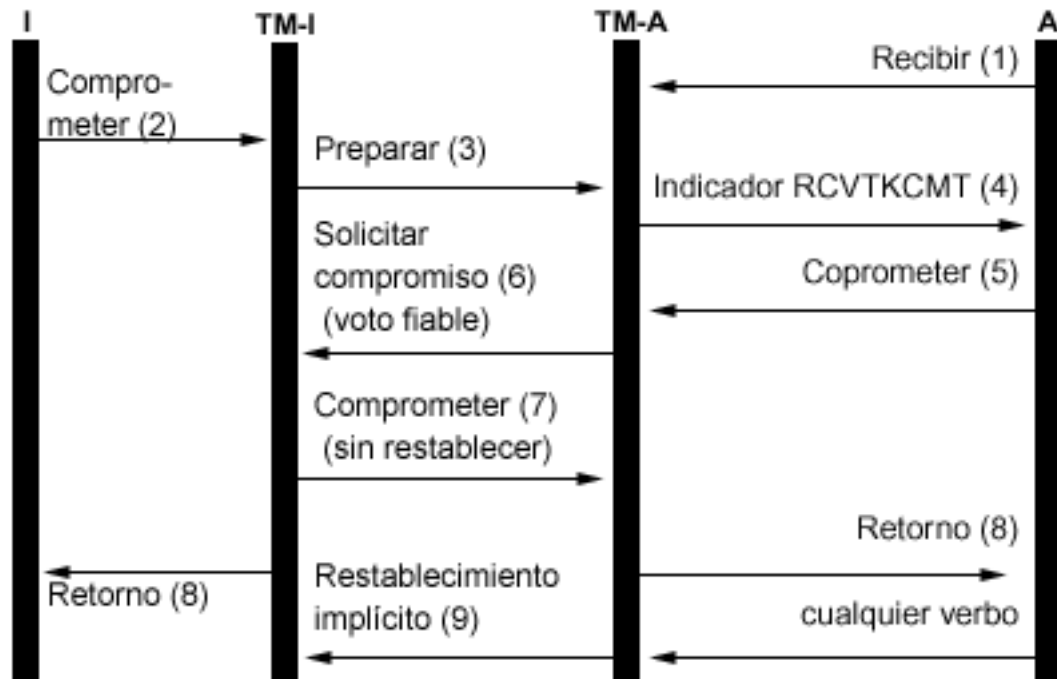
- Es poco probable que se tome una decisión heurística en un agente pendiente en el caso de que se produzca un error del sistema o de las comunicaciones salvo que la anomalía no pueda repararse.
- La lógica del programa no necesita el resultado de las transacciones anteriores para garantizar que los archivos de base de datos permanecen sincronizados.

La optimización de voto fiable solo se puede usar en i5/OS si se cumplen *todas* estas condiciones:

- Las ubicaciones del iniciador y el agente soportan el nivel de terminación anómala que se presupone de control de compromiso.
- La ubicación del iniciador acepta la indicación de voto fiable del agente. En los iniciadores i5/OS, esto depende del valor de dos opciones de compromiso:
  - El valor de la opción de compromiso Esperar resultado debe ser No (el valor predeterminado es Sí).
  - El valor de la opción de compromiso Aceptar voto fiable debe ser Sí (el valor predeterminado es Sí).
- La ubicación del agente emite un voto fiable durante la onda de preparación. Los agentes i5/OS siempre emiten un voto fiable. Esto se debe a que solo se pueden tomar decisiones heurísticas mediante un procedimiento manual que avise de los posibles efectos colaterales negativos de una toma de decisión heurística.

#### Flujo del proceso de compromiso con optimización de voto fiable

La figura siguiente muestra el flujo de mensajes entre los programas de aplicación y los gestores de transacciones cuando se utiliza la optimización del voto fiable. Ni el programa de aplicación del iniciador ni los programas de aplicación de agente tienen conocimiento del proceso de compromiso de dos fases. Los números entre paréntesis () de la figura corresponden a los elementos numerados en la descripción que se facilita a continuación.



#### Legend

- I** = Iniciador (la aplicación que inicia la petición de compromiso)
- TM-I** = El gestor de transacciones para el iniciador
- A** = Agente (la aplicación que recibe la petición de compromiso)
- TM-A** = El gestor de transacciones para el agente

La siguiente lista describe los eventos del proceso normal sin optimización de último agente cuando el agente emite un voto fiable. Aquí se describe un flujo básico. La secuencia de los eventos puede ser mucho más compleja si la red del programa de transacciones tiene varios niveles o se producen errores.

1. El programa de aplicación A recibe una petición en la que se le solicita que indique que está preparado para recibir una petición del programa I.
2. La aplicación del iniciador (I) emite una instrucción de compromiso.
3. El gestor de transacciones del iniciador (TM-I) asume el rol de iniciador para esta transacción. Inicia la onda de preparación enviando un mensaje de preparación a todas las demás ubicaciones que participan en la transacción.
4. Los gestores de transacciones de todas las demás ubicaciones asumen el rol de agente (TM-A). El TM-A notifica al programa de aplicación A que se ha recibido una petición de compromiso. Para los archivos ICF, la notificación tiene el formato del indicador de recepción de toma de compromiso (RCVTKCMT) de ICF que se activa.
5. El programa de aplicación A responde emitiendo una instrucción de compromiso (o una instrucción de retrotracción). Este es el voto del programa de aplicación.
6. El agente (TM-A) responde al iniciador (TM-I) con un mensaje de petición de compromiso. Los sistemas i5/OS envían un indicador de voto fiable con la petición de compromiso.
7. Cuando el iniciador (TM-I) recibe todos los votos, el TM-I envía un mensaje de compromiso. Si la opción de compromiso Esperar resultado es N (No) y la opción de compromiso Aceptar voto fiable es Y (Sí), se envía un indicador de no restablecimiento con el mensaje de compromiso. De este modo se indica al agente que no es necesario ningún mensaje de restablecimiento en respuesta al compromiso.
8. La transacción ha finalizado. Se envía información de retorno a los programas de aplicación (I y A). Esta información de retorno indica que la operación de compromiso se ha ejecutado.

satisfactoriamente. Si se produce un daño heurístico en el sistema A por tomar una decisión heurística antes de recibir el mensaje comprometido, no se informa de ello a la aplicación I. En cambio, el mensaje se envía a la cola de mensajes de QSYSOPR. Sin embargo, la aplicación A sí recibe la indicación del daño heurístico.

9. La próxima vez que el agente (TM-A) envíe algún mensaje al iniciador (TM-I) (tanto si es un flujo de datos como si es una instrucción de compromiso), con el mensaje se enviará un indicador de restablecimiento implícito para informar al TM-I de que el TM-A ha completado el compromiso satisfactoriamente. Esto se debe a que el TM-I debe conservar la información sobre la transacción completada hasta que haya confirmado que el TM-A ha recibido satisfactoriamente el mensaje de compromiso del paso 7 en la página 44

#### **Referencia relacionada**

API Cambiar opciones de compromiso (QTNCHGCO)

## **Soporte de transacciones XA para el control de compromiso**

DB2 Universal Database (UDB) para iSeries puede participar en las transacciones globales X/Open.

The Open Group ha definido un modelo estándar del sector para el trabajo transaccional que hace posible que los cambios realizados en recursos no relacionados formen parte de una única transacción global. Ejemplo de ello son los cambios realizados en bases de datos proporcionadas por dos proveedores independientes. Este es el llamado modelo de *proceso de transacciones distribuidas X/Open*.

Las publicaciones siguientes describen con detalle el modelo de proceso de transacciones distribuidas X/Open:

- *X/Open Guide*, febrero de 1996, "Distributed Transaction Processing: Reference Model, Version 3" (ISBN:1-85912-170-5, G504), The Open Group.
- *X/Open CAE Specification*, diciembre de 1991, "Distributed Transaction Processing: The XA Specification" (ISBN:1-872630-24-3, C193 or XO/CAE/91/300), The Open Group.
- *X/Open CAE Specification*, abril de 1995, "Distributed Transaction Processing: The TX (Transaction Demarcation) Specification" (ISBN:1-85912-094-6, C504), The Open Group.

Conviene que se familiarice con la información de estos manuales, especialmente con el de la especificación XA, antes de intentar utilizar el soporte de transacciones XA proporcionado por DB2 UDB para iSeries. Encontrará estos manuales en el sitio Web de The Open Group.

Hay cinco componentes en el modelo DTP:

#### **Programa de aplicación (AP)**

Implementa la función que necesita el usuario, especificando una secuencia de operaciones en la que intervienen recursos como las bases de datos. Define el principio y el final de las transacciones globales, accede a los recursos dentro de los límites de las transacciones y normalmente toma la decisión de si hay que comprometer o retrotraer cada transacción.

#### **Gestor de transacciones (TM)**

Gestiona las transacciones globales y coordina la decisión de iniciarlas y comprometerlas o retrotraerlas para garantizar que se lleve a cabo una transacción atómica. El TM también coordina las actividades de recuperación con los RM después de que haya fallado un componente.

#### **Gestor de recursos (RM)**

Gestiona una parte definida de los recursos compartidos de la máquina, como puede ser un sistema de gestión de bases de datos. El AP utiliza las interfaces definidas por cada RM para realizar el trabajo transaccional. El TM utiliza las interfaces definidas por cada RM para llevar a cabo la conclusión de las transacciones.

### Gestor de recursos de comunicaciones (CRM)

Permite que una instancia del modelo acceda a otra instancia, ya sea dentro o fuera del dominio del TM actual. Los CRM están fuera del ámbito de DB2 UDB para iSeries y no se estudian en este tema.

### Protocolo de comunicaciones

Los protocolos empleados por los CRM para comunicarse unos con otros. Están fuera del ámbito de DB2 UDB para iSeries y no se estudian en este tema.

La especificación XA es la parte del modelo DTP que describe un conjunto de interfaces que emplean los componentes TM y RM del modelo DTP. DB2 UDB para iSeries implementa estas interfaces como un conjunto de API y programas de salida de estilo de la plataforma UNIX. Consulte Interfaces API XA para obtener una documentación detallada de estas API y más información sobre cómo utilizar DB2 UDB para iSeries como un RM.

## iSeries Navigator y las transacciones XA

iSeries Navigator permite gestionar las transacciones XA como *transacciones globales*.

Una transacción global puede contener cambios tanto fuera como dentro de DB2 UDB para iSeries. La transacción global se coordina mediante un gestor de transacciones externo que utilice la arquitectura XA de The Open Group, u otra arquitectura similar. Una aplicación compromete o retrotrae una transacción global utilizando interfaces proporcionadas por el gestor de transacciones. El gestor de transacciones utiliza protocolos de compromiso definidos por la arquitectura XA, u otra arquitectura, para completar la transacción. DB2 UDB para iSeries actúa como gestor de recursos XA al participar en una transacción global. Hay dos tipos de transacciones globales:

- **Bloqueos con ámbito de transacción:** el ámbito de los bloqueos adquiridos en nombre de la transacción se circunscribe a la transacción. La transacción se puede mover de un trabajo a otro o de una hebra a otra.
- **Bloqueos con ámbito de trabajo:** el ámbito de los bloqueos adquiridos en nombre de la transacción se circunscribe al trabajo. La transacción no se puede mover del trabajo que la ha iniciado.

## Consideraciones sobre las transacciones XA

Las API XA de los bloqueos con ámbito de transacción se recomiendan para los nuevos usuarios del soporte de transacciones XA. Las API XA de los bloqueos con ámbito de trabajo se podrán seguir utilizando, pero han dejado de tener ventajas sobre los API XA de los bloqueos con ámbito de transacción. Las API de los bloqueos con ámbito de transacción tienen menos restricciones y más rendimiento en las siguientes situaciones;

- Si se utilizan múltiples conexiones SQL para trabajar en una sola rama de transacción XA.
- Si se utiliza una sola conexión SQL para trabajar en múltiples ramas de transacciones XA concurrentes.

En estas situaciones, hay que iniciar un trabajo aparte para que ejecute ramas de transacciones XA al utilizar las API XA para los bloqueos con ámbito de trabajo.

Conviene que comprenda las siguientes consideraciones y restricciones antes de utilizar DB2 UDB para iSeries como RM. El término *hebra* se refiere a un trabajo que no tenga capacidad para hebras o a una sola hebra de un trabajo con capacidad para hebras.

Las consideraciones siguientes son válidas tanto para las transacciones con bloqueos cuyo ámbito se circunscribe a la transacción como a las transacciones con bloqueos cuyo ámbito se circunscribe al trabajo, salvo que se indique lo contrario.

## Consideraciones sobre DB2 UDB para iSeries

- Las transacciones XA producidas con respecto a una base de datos local se deben realizar en trabajos que se ejecuten en modalidad de servidor SQL, lo que significa que las aplicaciones están limitadas a las interfaces SQL cuando se hacen cambios en DB2 UDB para iSeries durante una transacción XA. En el caso de estas transacciones, si se utiliza la API `xa_open()` o `db2xa_open()` en un trabajo que todavía no se ejecute en modalidad de servidor SQL, esta modalidad se inicia implícitamente.
- Las transacciones XA producidas con respecto a una base de datos remota tienen que utilizar la modalidad de servidor SQL cuando se utilizan las API XA para los bloqueos con ámbito de trabajo. Sin embargo, la modalidad de servidor es opcional para las transacciones XA producidas con respecto a una base de datos remota cuando se utilizan las API XA para los bloqueos con ámbito de transacción. Además, los cambios realizados en archivos DDM mediante los métodos de acceso de base de datos tradicionales de i5/OS están permitidos dentro de las transacciones XA producidas con respecto a una base de datos remota cuando no se utiliza la modalidad de servidor SQL.
- Los errores detectados por DB2 UDB para iSeries durante las invocaciones de las API XA se notifican mediante códigos de retorno según la especificación XA. Los mensajes de diagnóstico se dejan en las anotaciones de trabajo cuando el significado del error puede no estar claro solo a partir del código de retorno.

## Consideraciones sobre SQL incorporado

- Para utilizar una conexión de lenguaje de consulta estructurada (SQL) en las transacciones XA, hay que emplear la interfaz de programación de aplicaciones (API) `xa_open()` o `db2xa_open()` antes de que se establezca la conexión SQL. La base de datos relacional a la que se conectará se debe pasar a la API `xa_open()` o `db2xa_open()` mediante el parámetro `xainfo`. El perfil de usuario y la contraseña que se utilizarán en el trabajo al que se direcciona la conexión se pueden pasar a la API `xa_open()` o `db2xa_open()`. Si no se pasan, el perfil utiliza el valor que se ha especificado o utilizado por omisión durante el intento de conexión.

**Nota:** La siguiente consideración solo es válida para las transacciones con bloqueos cuyo ámbito se circunscribe al trabajo.

- Si se utiliza SQL incorporado para llevar a cabo transacciones XA, el trabajo efectuado para cada conexión se direcciona a un trabajo distinto, aunque las conexiones se efectúen en la misma hebra. Esto es distinto de la modalidad de servidor SQL sin XA, en la que el trabajo efectuado para todas las conexiones en una sola hebra se direcciona al mismo trabajo. Esto se debe a que la especificación XA requiere una llamada de preparación, compromiso o retrotracción aparte para cada instancia del gestor de recursos.

**Nota:** La siguiente consideración solo es válida para las transacciones con bloqueos cuyo ámbito se circunscribe al trabajo.

- Si se utiliza SQL incorporado para llevar a cabo transacciones XA, solo puede efectuarse una conexión por base de datos relacional para cada hebra. Cuando la hebra no esté asociada activamente a una rama de transacción, el trabajo solicitado en una de las conexiones de la hebra hará que el RM utilice el programa de salida `ax_reg()` del TM para determinar si el trabajo debe iniciar, reanudar o unir una rama de transacción.

Si el trabajo debe iniciar una rama de transacción, ello se lleva a cabo en la conexión de esa hebra con la base de datos relacional correspondiente.

Si el trabajo debe unir una rama de transacción, se redirecciona por la conexión con la base de datos relacional correspondiente establecida en la hebra que ha iniciado la rama de transacción. Observe que el sistema no garantiza que el perfil de usuario de esa conexión sea el mismo que el de la conexión de la hebra que efectúa la unión. El TM es el encargado de garantizar que esto no suponga ningún problema de seguridad. Los TM típicos utilizan el mismo perfil de usuario para todas las conexiones. Este perfil de usuario está autorizado para todos los datos gestionados por el TM. Las medidas de seguridad adicionales de acceso a estos datos se gestionan mediante el TM o el AP, en lugar de utilizar las técnicas de seguridad estándar del iSeries.

l **Nota:** La siguiente consideración solo es válida para las transacciones con bloqueos cuyo ámbito se  
l circunscribe al trabajo.

- Si el trabajo debe reanudar una rama de transacción, la conexión que se utilice depende de si la asociación de la rama de transacción suspendida se ha establecido iniciando o uniendo la rama de transacción.

El trabajo ulterior se lleva a cabo por la misma conexión hasta que se emplee la API `db2xa_end()` para suspender o finalizar la asociación de la hebra con esa rama de transacción.

## Consideraciones sobre CLI

- Si se utiliza la interfaz CLI para realizar transacciones XA, podría establecerse más de una conexión en la misma hebra después de utilizar la API `db2xa_open()`. Las conexiones se pueden utilizar en otras hebras para realizar transacciones XA, siempre y cuando esas otras hebras utilicen primero la API `db2xa_open()` con el mismo valor del parámetro `xainfo`.

**Nota:** La siguiente consideración solo es válida para las transacciones con bloqueos cuyo ámbito se circunscribe al trabajo.

- Si se utiliza la interfaz CLI para llevar a cabo transacciones XA, la conexión que se utiliza para iniciar una rama de transacción se debe emplear para todo el trabajo efectuado en esa rama de transacción. Si otra hebra debe unir la rama de transacción, el handle de la conexión empleada para iniciar la rama de transacción se debe pasar a la hebra que efectúa la unión, para que pueda realizar trabajo por esa misma conexión. Del mismo modo, si una hebra debe reanudar la rama de transacción, hay que utilizar la misma conexión.

Dado que no se pueden usar handles de conexión CLI en un trabajo distinto, la función de unión está limitada a las hebras que se ejecutan en el mismo trabajo que ha iniciado la rama de transacción cuando se utiliza la interfaz CLI.

## Consideraciones sobre bases de datos relacionales remotas

**Nota:** Estas consideraciones sobre una base de datos relacional remota solo son válidas para las transacciones con bloqueos cuyo ámbito se circunscribe al trabajo.

- Solo se pueden utilizar conexiones XA con una base de datos relacional remota si la base de datos relacional reside en un sistema que soporte conexiones de unidad de trabajo distribuida (DUW) DRDA. Ello incluye los sistemas iSeries que ejecuten conversaciones DRDA por SNA LU6.2 o que utilicen la V5R1 o posterior al ejecutar DRDA mediante conexiones TCP/IP. Incluye asimismo sistemas no iSeries que soporten DRDA por SNA LU6.2 o que soporten el protocolo XA mediante DRDA por TCP/IP.
- Antes de emplear la función de unión XA, hay que utilizar la API `db2xa_open()` en la hebra que efectúa la unión. Hay que especificar el mismo nombre de base de datos relacional y el mismo RMID en la API `db2xa_open()` tanto en la hebra que ha iniciado la rama de transacción como en la hebra que efectúa la unión. Si la rama de transacción está activa cuando se intenta una unión, la hebra que efectúa la unión se bloquea. La hebra que efectúa la unión permanece bloqueada hasta que la hebra activa suspende o finaliza su asociación a la rama de transacción.

## Consideración sobre la recuperación

- Se puede utilizar el soporte de las operaciones de compromiso y retrotracción heurísticas manuales que se proporciona para todas las definiciones de compromiso si es necesario forzar el compromiso o la retrotracción de una rama de transacción mientras está en un estado preparado.

## Consideraciones sobre las ramas de las transacciones

- La información sobre las ramas de transacción XA se facilita como parte de la información sobre control de compromiso visualizada por iSeries Navigator y los los mandatos Trabajar con trabajo (WRKJOB), Visualizar trabajo (DSPJOB) y Trabajar con definición de compromiso (WRKCMTDFN). Se muestra el nombre del TM, el estado de la rama de transacción, el identificador de la transacción y el calificador de la rama. Las definiciones de compromiso relacionadas con todas las transacciones XA

que están activas en este momento se pueden ver con el mandato WRKCMDFN JOB(\*ALL) STATUS(\*XOPEN) o visualizando la opción **Transacciones globales** en iSeries Navigator.

**Nota:** El siguiente punto solo es válido para las transacciones con bloqueos cuyo ámbito se circunscribe al trabajo.

- Si se suspende o finaliza una asociación entre una hebra y una rama de transacción existente mediante la API db2xa\_end(), la hebra podría iniciar una nueva rama de transacción. Si la conexión empleada para iniciar la nueva rama de transacción se ha empleado anteriormente para iniciar otra rama de transacción, y la asociación de la hebra a esa rama de transacción se ha finalizado o suspendido con la API db2xa\_end(), se podría iniciar un nuevo trabajo servidor SQL. Solo es necesario un nuevo trabajo servidor SQL si la primera rama de transacción todavía no se ha completado mediante la API db2xa\_commit() o db2xa\_rollback(). En este caso, se envía otro mensaje de finalización SQL7908 a las anotaciones de trabajo en el que se identifica el nuevo trabajo servidor SQL, igual que se identificó el trabajo servidor SQL original de la conexión al establecerse la conexión. Todas las peticiones SQL de la nueva rama de transacción se dirigen al nuevo trabajo servidor SQL. Cuando la rama de transacción se completa mediante la API db2xa\_commit() o db2xa\_rollback(), el nuevo trabajo servidor SQL se recicla y retorna a la agrupación de trabajos de prearranque.

| • Una rama de transacción se marca como Solo retroacción en las siguientes situaciones, pero solo para las transacciones XA que tienen bloqueos con ámbito de trabajo:

- | – Una hebra finaliza cuando todavía está asociada a la rama de transacción. Esto incluye el caso de que una hebra finalice como consecuencia de una terminación del proceso.
- | – El sistema falla.

| • En el caso de las transacciones XA que tengan bloqueos con ámbito de transacción, el sistema retrotrae una rama de transacción si todavía hay hebras asociadas a ella y se da alguna de estas situaciones:

- | – Finaliza la conexión relacionada con la rama de transacción.
- | – Finaliza el trabajo que ha iniciado la rama de transacción.
- | – El sistema falla.

| **Nota:** La siguiente consideración solo es válida para las transacciones con bloqueos cuyo ámbito se circunscribe al trabajo.

- Hay una situación en la que el sistema retrotraerá una rama de transacción, con independencia de si todavía hay hebras asociadas. Esto se produce cuando finaliza el trabajo servidor SQL al que se direcciona el trabajo de la conexión. Esta situación solo puede darse cuando se utiliza el mandato CL Finalizar trabajo (ENDJOB) para ese trabajo.
- Una rama de transacción no se verá afectada si ninguna hebra tiene una asociación activa con ella cuando se produce alguna de las situaciones siguientes. El TM puede comprometer o retrotraer la rama de transacción desde cualquier hebra que haya utilizado la API xa\_open() o db2xa\_open() con el mismo valor del parámetro xainfo que el especificado en la hebra que ha iniciado la rama de transacción.
  - Finaliza la conexión relacionada con la rama de transacción.
  - Una hebra o un trabajo que haya llevado a cabo trabajo para la rama de transacción utiliza la API db2xa\_close().
  - El sistema falla. En este caso, la rama de transacción solo no se verá afectada si está en estado preparado. Si está en estado desocupado, el sistema la retrotrae.

| • Cuando el identificador de transacción (XID) de dos ramas de transacción XA tienen el mismo identificador de transacción global (GTRID), pero distintos calificadores de rama (BQUAL), se dice que están *débilmente acopladas*. Por omisión, las ramas de transacción débilmente acopladas no comparten bloqueos. Sin embargo, cuando se utilizan las API XA para bloqueos con ámbito de transacción, existe una opción que permite compartir bloqueos a las transacciones débilmente acopladas.

#### **Conceptos relacionados**

“Consideraciones sobre las transacciones XA” en la página 24

En el entorno XA, cada base de datos se considera un gestor de recursos aparte. Si un gestor de

transacciones pretende acceder a dos bases de datos en la misma transacción, debe emplear los protocolos XA para llevar a cabo el compromiso de dos fases con los dos gestores de recursos.

Sitio Web de Open Group

“Modalidad de servidor SQL y transacciones con ámbito de hebra para el control de compromiso”

En general, las definiciones de compromiso con bloqueos cuyo ámbito se circunscribe al trabajo suelen tener un ámbito de grupo de activación.

#### **Tareas relacionadas**

“Cuándo forzar operaciones de compromiso y retrotracción y cuándo cancelar la resincronización” en la página 114

En este tema se explica cuándo y cómo hay que forzar una operación de compromiso o retrotracción y cuándo hay que cancelar la resincronización.

## **Modalidad de servidor SQL y transacciones con ámbito de hebra para el control de compromiso**

En general, las definiciones de compromiso con bloqueos cuyo ámbito se circunscribe al trabajo suelen tener un ámbito de grupo de activación.

Si un trabajo es multihebra, todas las hebras del trabajo tienen acceso a la definición de compromiso, y los cambios realizados para una transacción concreta pueden quedar repartidos en múltiples hebras. Es decir, todas las hebras cuyos programas se ejecutan en el mismo grupo de activación participan en una sola transacción.

Hay casos en los que es deseable que el ámbito del trabajo transaccional se circunscribe a la hebra y no a un grupo de activación. Dicho de otro modo, cada hebra tiene su propia definición de compromiso, y el trabajo transaccional de cada definición de compromiso es independiente del trabajo realizado en otras hebras.

Esto está soportado en DB2 Universal Database (UDB) para iSeries si se utiliza la API Cambiar trabajo (QWTCHGJB) con el fin de cambiar el trabajo para que se ejecute en modalidad de servidor SQL. Cuando se solicita una conexión SQL en la modalidad de servidor SQL, se direcciona a un trabajo independiente. Todas las operaciones SQL posteriores que se efectúen para esa conexión también se direccionarán a ese trabajo. Una vez establecida la conexión, se envía el mensaje de finalización SQL7908 a las anotaciones de trabajo del trabajo de modalidad de servidor SQL indicando a qué trabajo se direccionan las peticiones SQL. La definición de compromiso es propiedad del trabajo que se indica en este mensaje. Si se producen errores, podría ser necesario examinar las anotaciones de ambos trabajos para entender el origen del problema, porque no se está efectuando ninguna tarea real en el trabajo que ejecuta las sentencias SQL.

Cuando se ejecuta en modalidad de servidor SQL, solo se pueden utilizar interfaces SQL para realizar trabajo bajo control de compromiso. Se puede utilizar SQL incorporado o la interfaz a nivel de llamada (CLI) SQL. Todas las conexiones establecidas mediante SQL incorporado en una sola hebra se direccionan al mismo trabajo de fondo. Esto permite que una sola petición de compromiso comprometa el trabajo de todas las conexiones, igual que se haría en un trabajo que no se ejecutara en modalidad de servidor SQL. Cada conexión establecida mediante la interfaz CLI se direcciona a un trabajo aparte. La interfaz CLI requiere que el trabajo llevado a cabo para cada conexión se comprometa o retrotraiga de manera independiente.

Cuando se ejecuta en modalidad de servidor SQL, no se pueden efectuar estas operaciones bajo control de compromiso:

- Realizar cambios de registros que se efectúen con interfaces que no son interfaces SQL
- Realizar cambios en archivos DDM
- Realizar cambios en recursos de compromiso de API



No puede iniciar el control de compromiso directamente en un trabajo que se ejecute en modalidad de servidor SQL.

#### **Conceptos relacionados**

“Soporte de transacciones XA para el control de compromiso” en la página 45

DB2 Universal Database (UDB) para iSeries puede participar en las transacciones globales X/Open.

Ejecutar CLI DB2 UDB en modalidad de servidor

Iniciar CLI DB2 en modalidad de servidor SQL

Restricciones para ejecutar CLI DB2 UDB en modalidad de servidor

#### **Referencia relacionada**

API Cambiar trabajo (QWTCHGJB)

---

## **Iniciar el control de compromiso**

Para iniciar el control de compromiso, utilice el mandato Arrancar control de compromiso (STRCMTCTL).

**Nota:** No es necesario iniciar el control de compromiso mediante aplicaciones SQL. SQL inicia implícitamente el control de compromiso en el momento de la conexión cuando el nivel de aislamiento de SQL no es \*NONE.

Al utilizar el mandato STRCMTCTL, puede especificar estos parámetros:

#### **Nivel de bloqueo de compromiso**

Especifique el nivel de bloqueo con el parámetro LCKLVL del mandato STRCMTCTL. El nivel que especifique se convertirá en el nivel de bloqueo de registros predeterminado para los archivos de base de datos que se abran y se coloquen bajo control de compromiso para la definición de compromiso.

#### **Objeto de notificación de compromiso**

Utilice el parámetro NTFY para especificar el objeto de notificación. Un objeto de notificación es una cola de mensajes, un área de datos o un archivo de base de datos que contiene información que identifica la última transacción satisfactoria llevada a cabo para una definición de compromiso determinada si esa definición de compromiso no ha finalizado con normalidad.

#### **Parámetro de ámbito de compromiso**

Utilice el parámetro CMTSCOPE para especificar el ámbito de compromiso. Cuando se inicia el control de compromiso, el sistema crea una definición de compromiso. El parámetro de ámbito de compromiso identifica el ámbito de la definición de compromiso. El valor predeterminado establece que el ámbito de la definición de compromiso se circunscriba al grupo de activación del programa que realiza la petición de iniciar el control de compromiso. El ámbito alternativo es el que se circunscribe al trabajo.

#### **Parámetro de diario predeterminado**

Puede especificar un diario predeterminado al iniciar el control de compromiso. Puede emplear un diario predeterminado por estos motivos:

- Desea capturar entradas de diario de las transacciones. Estas entradas pueden ayudarle a analizar el historial de los recursos asociados a una transacción. No se utilizan para aplicar ni eliminar cambios registrados por diario. El parámetro de omitir entradas de diario (OMTJRNE) determina si el sistema escribe entradas de transacciones.
- Desea mejorar el rendimiento de los trabajos que cierran archivos y los vuelven a abrir en un paso de direccionamiento. Si cierra todos los archivos asignados a un diario que no es el diario predeterminado, toda la información del sistema sobre el diario se elimina del paso de direccionamiento. Si más adelante se abre un archivo asignado a ese diario, toda la información sobre el diario debe volverse a crear. El sistema guarda información sobre el diario predeterminado con la definición de compromiso, tanto si los recursos asignados al diario están activos como si no.

### **Parámetro de texto de compromiso**

Utilice el parámetro TEXT para identificar el texto específico que se asociará a una definición de compromiso al visualizar información sobre las definiciones de compromiso iniciadas para un trabajo. Si no se especifica ningún texto, el sistema proporciona un texto descriptivo predeterminado.

### **Parámetro de omitir entradas de diario**

Si especifica un diario predeterminado para mejorar el rendimiento, puede emplear el parámetro OMTJRNE para evitar que el sistema escriba entradas de diario de transacciones. El hecho de que el sistema escriba entradas de transacciones aumenta de forma notable el tamaño del receptor de diario y disminuye el rendimiento en las operaciones de compromiso y retrotracción.

Las entradas de transacciones pueden resultar de utilidad al configurar y probar el entorno de control de compromiso o una aplicación nueva.

Las entradas de transacciones se escriben en el diario predeterminado independientemente del valor del parámetro OMTJRNE en estas condiciones:

- Se produce un error del sistema en una operación de compromiso o retrotracción.
- Se efectúa un cambio manual en un recurso que participaba en una transacción y el cambio ha generado una condición heurística mixta. En Estados de la transacción para el control de compromiso de dos fases, encontrará una descripción de la condición heurística mixta. Este tipo de cambio manual se llama decisión heurística.

Puede emplear la información sobre los recursos que participaban en la transacción para determinar la acción que se debe llevar a cabo en estas situaciones.

Puede utilizar el buscador de información de entradas de diario para ver los diseños de los datos específicos de las entradas de diario (control de compromiso) de las transacciones.

### **Conceptos relacionados**

“Estados de la transacción para el control de compromiso de dos fases” en la página 31

Se establece una definición de compromiso en cada ubicación que forma parte de la red de programas de transacciones. Para cada definición de compromiso, el sistema mantiene un seguimiento del estado de la transacción actual y la transacción anterior.

Buscador de información de entradas de diario

### **Tareas relacionadas**

“Cuándo forzar operaciones de compromiso y retrotracción y cuándo cancelar la resincronización” en la página 114

En este tema se explica cuándo y cómo hay que forzar una operación de compromiso o retrotracción y cuándo hay que cancelar la resincronización.

### **Referencia relacionada**

Mandato Arrancar control de compromiso (STRCMTCTL)

## **Objeto de notificación de compromiso**

*Objeto de notificación* es una cola de mensajes, un área de datos o un archivo de base de datos que contiene información que identifica la última transacción satisfactoria llevada a cabo para una definición de compromiso determinada si esa definición de compromiso no ha finalizado con normalidad.

La información empleada para identificar la última transacción satisfactoria de una definición de compromiso se proporciona mediante la *identificación de compromiso* que asocia una operación de compromiso a un conjunto específico de cambios de recursos comprometibles.

La identificación de compromiso de la última transacción satisfactoria de una definición de compromiso se coloca en el objeto de notificación únicamente si la definición de compromiso no finaliza normalmente. Esta información puede emplearse a modo de ayuda para determinar dónde ha finalizado el proceso de una aplicación a fin de que pueda volverse a iniciar la aplicación.

En el caso de las agrupaciones de discos independientes, el objeto de notificación debe residir en la misma agrupación de discos independiente o el mismo grupo de agrupaciones de discos independientes que la definición de compromiso. Si mueve la definición de compromiso a otra agrupación de discos independiente u otro grupo de agrupaciones de discos independientes, el objeto de notificación también debe residir en esa otra agrupación de discos independiente o en ese otro grupo de agrupaciones de discos independientes. El objeto de notificación de la otra agrupación de discos independiente o del otro grupo de agrupaciones de discos independientes se actualiza si la definición de compromiso finaliza de forma anómala. Si el objeto de notificación no se encuentra en la otra agrupación de discos independiente o el otro grupo de agrupaciones de discos independientes, la actualización falla con el mensaje CPF8358.

Si en la transacción actual participan recursos registrados por diario y se efectúa una operación de compromiso con una identificación de compromiso, la identificación de compromiso se coloca en la entrada de diario de compromiso (código de diario y tipo de entrada C CM) que identifica esa transacción específica como comprometida. Se envía una entrada de diario de compromiso con la identificación de compromiso a cada diario asociado a los recursos que participan en la transacción.

La tabla siguiente muestra cómo se especifica la identificación de compromiso y su tamaño máximo. Si la identificación de compromiso supera este tamaño máximo, se truncará cuando se escriba en el objeto de notificación.

Lenguaje	Operación	Máximo de caracteres de la identificación de compromiso
CL	Mandato COMMIT	3000 <sup>1</sup>
ILE RPG*	Código de operación COMMIT	4000 <sup>1</sup>
PLI	Subrutina PLICOMMIT	4000 <sup>1</sup>
ILE C*	Función _Rcommit	4000 <sup>1</sup>
ILE COBOL*	Verbo COMMIT	No soportado
SQL	Sentencia COMMIT	No soportado
<b>Nota:</b> <sup>1</sup> Si el objeto de notificación es un área de datos, el tamaño máximo es de 2000 caracteres.		

Cuando se actualiza un objeto de notificación con la identificación de compromiso, se actualiza de esta manera:

#### Archivo de base de datos

Si se utiliza un archivo de base de datos como objeto de notificación, la identificación de compromiso se añade al final del archivo. Los registros existentes permanecerán en el archivo. Dado que varios usuarios o trabajos pueden cambiar los registros al mismo tiempo, cada identificación de compromiso del archivo contiene información exclusiva para asociar los datos al trabajo y la definición de compromiso que han fallado. El archivo que da servicio se puede registrar por diario.

#### Área de datos

Si se utiliza un área de datos como objeto de notificación, todo el contenido del área de datos se sustituye cuando se coloca la identificación de compromiso en el área de datos. Si hay más de un usuario o trabajo que utiliza el mismo programa, solo estará en el área de datos la identificación de compromiso de la última definición de compromiso que no ha finalizado normalmente. Por lo tanto, un único objeto de notificación de área de datos podría no generar la información correcta para volver a iniciar los programas de aplicación. A fin de resolver este problema, emplee un área de datos aparte para cada definición de compromiso de cada trabajo o usuario de estación de trabajo.

#### Cola de mensajes

Si se utiliza una cola de mensajes como objeto de notificación, se envía el mensaje CPI8399 a la cola de mensajes. La identificación de compromiso se coloca en el texto de segundo nivel del mensaje CPI8399. Como sucede al utilizar un archivo de base de datos para el objeto de

notificación, el contenido de cada identificación de compromiso identifica de forma exclusiva una definición de compromiso determinada de un trabajo para que pueda volverse a iniciar un programa de aplicación.

### **Conceptos relacionados**

“Control de compromiso en las aplicaciones por lotes” en la página 27

Las aplicaciones por lotes pueden necesitar o no el control de compromiso. En algunos casos, una aplicación por lotes puede llevar a cabo una sola función de leer un archivo de entrada y actualizar un archivo maestro. Sin embargo, puede emplear el control de compromiso para este tipo de aplicación si es importante volver a iniciarla tras una finalización anómala.

“Ejemplo: utilizar un objeto de notificación para iniciar una aplicación” en la página 92

Cuando se inicia un programa tras una finalización anómala, este puede buscar una entrada en el objeto de notificación. Si la entrada existe, el programa puede iniciar de nuevo una transacción. Una vez que transacción se ha vuelto a iniciar, el programa borra el objeto de notificación para impedir que se inicie otra vez la misma transacción.

## **Nivel de bloqueo de compromiso**

El valor que especifique para el parámetro LCKLVL en el mandato Arrancar control de compromiso (STRCMTCTL) se convertirá en el nivel de bloqueo de registros predeterminado para los archivos de base de datos que se abran y se coloquen bajo control de compromiso para la definición de compromiso.

El nivel de bloqueo de registros predeterminado no se puede alterar temporalmente al abrir archivos de base de datos locales. Sin embargo, los archivos de base de datos a los que se accede mediante SQL utilizan el nivel de aislamiento de SQL actual que está en vigor cuando se emite la primera sentencia SQL para el mismo.

El nivel de bloqueo debe especificarse según las necesidades propias, los períodos de espera permitidos y los procedimientos de liberación más utilizados.

Las descripciones siguientes solo hacen referencia a los archivos que se abren bajo control de compromiso:

### **Nivel de bloqueo \*CHG**

Utilice este valor si desea impedir que otros trabajos que se ejecutan al mismo tiempo efectúen cambios en los registros modificados. Para los archivos que se abren para control de compromiso, el bloqueo se retiene mientras dura la transacción. En el caso de los archivos que no se abren bajo control de compromiso, el bloqueo del registro solo se retiene desde el momento en que se lee el registro hasta que se completa la operación de actualización.

### **Nivel de bloqueo \*CS**

Utilice este valor para impedir que otros trabajos que se ejecutan al mismo tiempo efectúen cambios en los registros modificados y los registros recuperados. Los registros recuperados que no se cambian solo se protegen hasta que se liberan, o se recupera otro registro.

El nivel de bloqueo \*CS garantiza que otros trabajos no podrán leer un registro para actualización que este trabajo ha leído. Además, el programa no puede leer registros para actualización que se han bloqueado con el tipo de bloqueo de registros \*UPDATE en otro trabajo hasta que ese trabajo accede a otro registro.

### **Nivel de bloqueo \*ALL**

Utilice este valor para impedir que otros trabajos que se ejecutan bajo control de compromiso al mismo tiempo efectúen cambios en los registros modificados y los registros recuperados que están bajo control de compromiso. Los registros que se recuperan o modifican se protegen hasta la siguiente operación de compromiso o retroacción.

El nivel de bloqueo \*ALL garantiza que otros trabajos no podrán acceder a un registro para actualización que este trabajo ha leído. Es distinto del protocolo de bloqueo normal. Cuando se

especifica el nivel de bloqueo \*ALL, ni tan solo es posible acceder a un registro que no se ha leído para actualización si este está bloqueado con el tipo de bloqueo de registros \*UPDATE en otro trabajo.

La tabla siguiente muestra la duración de los bloqueos de registros para los archivos que están bajo control de compromiso y para los que no lo están.

Petición	Parámetro LCKLVL	Duración del bloqueo	Tipo de bloqueo
Solo lectura	Ningún control de compromiso	Ningún bloqueo	Ninguno
	*CHG	Ningún bloqueo	Ninguno
	*CS	Desde lectura hasta siguiente lectura, compromiso o retrotracción	*READ
	*ALL	Desde lectura hasta compromiso o retrotracción	*READ
Lectura para actualización y luego actualización o supresión <sup>1</sup>	Ningún control de compromiso	Desde lectura hasta actualización o supresión	*UPDATE
		Desde lectura hasta actualización o supresión	*UPDATE
	*CHG	Luego, desde actualización o supresión hasta siguiente compromiso o retrotracción <sup>2</sup>	*UPDATE
		Desde lectura hasta actualización o supresión	*UPDATE
	*CS	Luego, desde actualización o supresión hasta siguiente compromiso o retrotracción <sup>2</sup>	*UPDATE
		Desde lectura hasta actualización o supresión	*UPDATE
	*ALL	Desde lectura hasta actualización o supresión	*UPDATE
		Luego, desde actualización o supresión hasta siguiente compromiso o retrotracción <sup>2</sup>	
Lectura para actualización y luego liberación <sup>1</sup>	Ningún control de compromiso	Desde lectura hasta liberación	*UPDATE
	*CHG	Desde lectura hasta liberación	*UPDATE
	*CS	Desde lectura hasta liberación, compromiso o retrotracción	*UPDATE
		Luego, desde liberación hasta siguiente lectura, compromiso o retrotracción	*UPDATE
	*ALL	Desde lectura hasta liberación, compromiso o retrotracción	*UPDATE
		Luego, desde liberación hasta siguiente compromiso o retrotracción	

Petición	Parámetro LCKLVL	Duración del bloqueo	Tipo de bloqueo
Adición	Ningún control de compromiso	Ningún bloqueo	Ninguno
	*CHG	Desde adición hasta compromiso o retrotracción	*UPDATE
	*CS	Desde adición hasta compromiso o retrotracción	*UPDATE
	*ALL	Desde adición hasta compromiso o retrotracción	*UPDATE
Escritura directa	Ningún control de compromiso	Mientras dura la escritura directa	*UPDATE
	*CHG	Desde escritura directa hasta compromiso o retrotracción	*UPDATE
	*CS	Desde escritura directa hasta compromiso o retrotracción	*UPDATE
	*ALL	Desde escritura directa hasta compromiso o retrotracción	*UPDATE

**Nota:**

<sup>1</sup>Si se realiza una operación de compromiso o retrotracción después de una operación de lectura para actualización, pero antes de que se actualice, suprima o libere el registro, el registro se desbloquea durante la operación de compromiso o retrotracción. La protección del registro se pierde en cuanto finaliza el compromiso o la retrotracción.

<sup>2</sup>Si se suprime un registro pero todavía no se ha emitido el compromiso o la retrotracción para la transacción, el registro suprimido no permanece bloqueado. Si el mismo trabajo u otro distinto intenta leer el registro suprimido por clave, el trabajo recibirá una indicación de registro no encontrado. Sin embargo, si existe una vía de acceso por clave exclusiva sobre el archivo, se impedirá que otro trabajo inserte o actualice un registro con el mismo valor de clave exclusivo que el del registro suprimido hasta que se comprometa la transacción.

El tipo de bloqueo de registros \*READ se obtiene en los registros que no se leen para actualización cuando el nivel de bloqueo es \*CS o \*ALL. Este tipo de bloqueo impide que otros trabajos lean los registros para actualización, pero no impide que se acceda a los registros desde una operación solo de lectura.

El tipo de bloqueo de registros \*UPDATE se obtiene en los registros actualizados, suprimidos, añadidos o leídos para actualización. Este tipo de bloqueo impide que otros trabajos lean los registros para actualización e impide que los trabajos que se ejecutan bajo control de compromiso con el nivel de bloqueo de registros \*CS o \*ALL accedan a los registros, incluso para una operación solo de lectura.

Los programas que no utilizan el control de compromiso pueden leer los registros bloqueados por otro trabajo, pero no pueden leer registros para actualización, independientemente del valor especificado para el parámetro LCKLVL.

El nivel de bloqueo, especificado para una definición de compromiso cuando se inicia el control de compromiso para un grupo de activación o para el trabajo, solo se aplica a las operaciones de apertura asociadas a esa definición de compromiso concreta.

**Nota:** Los valores de nivel de bloqueo \*CS y \*ALL impiden que se recupere un registro que actualmente tiene un cambio pendiente de otro trabajo. Sin embargo, los valores de nivel de bloqueo \*CS y

\*ALL *no* impiden que se recupere un registro utilizando un programa que se ejecuta en un grupo de activación que actualmente tiene un cambio pendiente de un programa que se ejecuta en otro grupo de activación del mismo trabajo.

En el mismo trabajo, un programa puede cambiar un registro que ya se ha cambiado en la transacción actual siempre que se acceda de nuevo al registro utilizando la misma definición de compromiso. Al emplear la definición de compromiso a nivel de trabajo, el acceso al registro modificado puede efectuarse desde un programa que se ejecute en cualquier grupo de activación que utilice la definición de compromiso a nivel de trabajo.

#### **Conceptos relacionados**

“Consideraciones y restricciones sobre el control de compromiso” en la página 25

En este tema se proporcionan diversas consideraciones y restricciones relacionadas con el control de compromiso.

#### **Referencia relacionada**

Mandato Arrancar control de compromiso (STRCMTCTL)

---

## **Finalizar control de compromiso**

En este tema encontrará qué prerequisites hacen falta para finalizar el control de compromiso y cómo finalizar el control de compromiso.

Puede finalizar el control de compromiso para la definición de compromiso a nivel de trabajo o a nivel de grupo de activación con el mandato Finalizar control de compromiso (ENDCMTCTL). Al emitir el mandato ENDCMTCTL se indica al sistema que debe finalizarse la definición de compromiso que utiliza el programa que realiza la petición. El mandato ENDCMTCTL solo finaliza una definición de compromiso del trabajo y todas las demás definiciones de compromiso del trabajo permanecen intactas.

Si se finaliza la definición de compromiso a nivel de grupo de activación, los programas que se ejecuten en ese grupo de activación ya no podrán efectuar cambios bajo control de compromiso, salvo que ya se haya iniciado la definición de compromiso a nivel de trabajo para el trabajo. Si la definición de compromiso a nivel de trabajo está activa, inmediatamente pasa a estar disponible a fin de que puedan utilizarla los programas que se ejecutan en el grupo de activación que acaba de finalizar el control de compromiso.

Si se finaliza la definición de compromiso a nivel de trabajo, los programas que se ejecuten en el trabajo que utilizaba la definición de compromiso a nivel de trabajo ya no podrán efectuar cambios bajo control de compromiso sin volver a iniciar primero el control de compromiso con el mandato STRCMTCTL.

Antes de emitir el mandato ENDCMTCTL, deben cumplirse las condiciones siguientes para la definición de compromiso que hay que finalizar:

- Todos los archivos abiertos bajo control de compromiso para la definición de compromiso que se va a finalizar deben cerrarse previamente. Si se finaliza la definición de compromiso a nivel de trabajo, esta operación incluye todos los archivos abiertos bajo control de compromiso por cualquier programa que se ejecute en cualquier grupo de activación que utilice la definición de compromiso a nivel de trabajo.
- Todos los recursos de compromiso de API de la definición de compromiso que se va a finalizar deben eliminarse previamente con la API QTNRMVCR. Si se finaliza la definición de compromiso a nivel de trabajo, esta operación incluye todos los recursos de compromiso de API añadidos por cualquier programa que se ejecute en cualquier grupo de activación que utilice la definición de compromiso a nivel de trabajo.
- Una base de datos remota asociada a la definición de compromiso que se va a finalizar debe desconectarse.
- Todas las conversaciones protegidas asociadas a la definición de compromiso deben finalizarse normalmente utilizando el nivel de sincronización correcto.

Si se finaliza el control de compromiso en un trabajo interactivo y uno o varios de los recursos comprometibles asociados a la definición de compromiso tienen cambios pendientes, se envía el mensaje de consulta CPA8350 al usuario para preguntarle si desea comprometer los cambios pendientes, retrotraer los cambios pendientes o cancelar la petición ENDCMTCTL.

Si se finaliza el control de compromiso en un trabajo por lotes y uno o varios de los archivos cerrados asociados a la definición de compromiso que se va a finalizar todavía tienen cambios pendientes, los cambios se retrotraen y se envía un mensaje:

- CPF8356 si solo se registran los recursos locales.
- CPF835C si solo se registran los recursos remotos.
- CPF83E4 si se registran los recursos locales y remotos.

Si está definido un objeto de notificación para la definición de compromiso que hay que finalizar, el objeto podría actualizarse.

Cuando se finaliza un grupo de activación que tiene una API registrada como el último agente, se llama al programa de salida de la API para recibir la decisión de compromiso o retrotracción. En este caso, aunque el grupo de activación finalice con normalidad, el programa de salida de la API todavía puede devolver una petición de retrotracción. Por consiguiente, la operación de compromiso implícita podría no realizarse.

Una vez que la definición de compromiso ha finalizado satisfactoriamente, se han llevado a cabo todas las tareas de recuperación necesarias (en caso de haber alguna). No se lleva a cabo ninguna tarea de recuperación adicional para los recursos de compromiso asociados a la definición de compromiso que acaba de finalizar.

Tras finalizarse la definición de compromiso, puede volver a iniciarse la definición de compromiso a nivel de trabajo o a nivel de grupo de activación para los programas que se ejecutan en el grupo de activación. Solo se puede iniciar la definición de compromiso a nivel de trabajo si todavía no se ha iniciado para el trabajo.

Aunque los programas que se ejecutan en un grupo de activación pueden iniciar y finalizar las definiciones de compromiso muchas veces, la cantidad de recursos del sistema que se precisan para las repetidas operaciones de inicio y finalización puede hacer que disminuya el rendimiento del trabajo y el rendimiento global del sistema. Por lo tanto, se recomienda dejar activa una definición de compromiso si un programa al que se llamará más adelante la utilizará.

#### **Conceptos relacionados**

“Actualizaciones del objeto de notificación” en la página 65

En este tema figuran los cambios no comprometidos en relación con el objeto de notificación.

#### **Referencia relacionada**

Mandato Finalizar control de compromiso (ENDCMTCTL)

---

## **Finalización iniciada por sistema del control de compromiso**

El sistema puede finalizar el control de compromiso o llevar a cabo una operación de compromiso o retrotracción implícita. En ocasiones la finalización del control de compromiso iniciada por el sistema es normal. En otras situaciones, el control de compromiso concluye con una finalización anómala del sistema o del trabajo.

## **Control de compromiso durante la finalización de un grupo de activación**

El sistema finaliza automáticamente una definición de compromiso a nivel de grupo de activación cuando se finaliza un grupo de activación.



Si existen cambios pendientes para una definición de compromiso a nivel de grupo de activación y el grupo de activación finaliza normalmente, el sistema lleva a cabo una operación de compromiso implícita para la definición de compromiso antes de que finalice. En caso contrario, se efectúa una operación de retrotracción implícita para la definición de compromiso a nivel de grupo de activación antes de que se finalice si el grupo de activación finaliza de forma anómala o el sistema ha encontrado errores al cerrar los archivos abiertos bajo control de compromiso cuyo ámbito se circunscribe al grupo de activación.

**Nota:** Una operación de compromiso o retrotracción implícita nunca se lleva a cabo durante el proceso de finalización del grupo de activación para las definiciones de compromiso \*JOB o \*DFTACTGRP. Esto se debe a que las definiciones de compromiso \*JOB y \*DFTACTGRP nunca se finalizan como consecuencia de la finalización de un grupo de activación. En su lugar, estas definiciones de compromiso se finalizan explícitamente con un mandato ENDCMTCTL o el sistema las finaliza cuando finaliza el trabajo.

El sistema cierra automáticamente los archivos cuyo ámbito se circunscribe al grupo de activación cuando finaliza el grupo de activación. Esto incluye los archivos de base de datos cuyo ámbito se circunscribe al grupo de activación abiertos bajo control de compromiso. El cierre de estos archivos se produce antes de que pueda llevarse a cabo ninguna operación de compromiso implícita para la definición de compromiso a nivel de grupo de activación. Por consiguiente, primero se fuerza el traslado de los registros que residen en un almacenamiento intermedio de E/S a la base de datos antes de llevarse a cabo ninguna operación de compromiso implícita.

Como parte de la operación de compromiso o retrotracción implícita que se podría realizar, se efectúa una llamada al programa de salida de compromiso y retrotracción de API para cada recurso de compromiso de API asociado a la definición de compromiso a nivel de grupo de activación. El programa de salida debe completar su proceso en 5 minutos. Una vez que se ha llamado al programa de salida de compromiso y retrotracción de API, el sistema elimina el recurso de compromiso de API automáticamente.

Si se efectúa una operación de retrotracción implícita para una definición de compromiso que finaliza debido a la finalización de un grupo de activación, se podría actualizar el objeto de notificación, si hay uno definido para la definición de compromiso.

#### **Conceptos relacionados**

“Actualizaciones del objeto de notificación” en la página 65

En este tema figuran los cambios no comprometidos en relación con el objeto de notificación.

## **Operaciones de compromiso y retrotracción implícitas**

En algunos casos, el sistema inicia una operación de compromiso o retrotracción para una definición de compromiso. Estos tipos de operaciones de compromiso y retrotracción se conocen como *peticiones de compromiso y retrotracción implícitas*.

Normalmente, una operación de compromiso o retrotracción se inicia desde un programa de aplicación utilizando uno de los lenguajes de programación disponibles que da soporte al control de compromiso. Estos tipos de operaciones de compromiso y retrotracción se denominan *peticiones de compromiso y retrotracción explícitas*.

Las dos tablas siguientes muestran lo que hace el sistema cuando se producen determinados eventos en relación con una definición de compromiso que tiene cambios pendientes. Una definición de compromiso tiene cambios pendientes si se da alguna de las condiciones siguientes:

- Se ha actualizado alguno de los recursos comprometibles.
- Se ha leído un archivo de base de datos abierto bajo control de compromiso ya que al leer un archivo se cambia la posición del archivo.

- La definición de compromiso tiene un recurso de API. Como los cambios en los recursos de API se efectúan mediante un programa de usuario, el sistema debe suponer que todos los recursos de API tienen cambios pendientes.

La entrada de diario C CM (operación de compromiso) y la entrada de diario C RB (operación de retrotracción) indican si la operación era explícita o implícita.

La tabla siguiente muestra las acciones que efectúa el sistema cuando un trabajo finaliza, ya sea de forma normal o anómala, basándose en estas situaciones:

- El estado de la transacción.
- El valor de acción si se finaliza el trabajo de la definición de compromiso.
- Si un recurso de API es el último agente.

Estado	API de último agente	Opción de Acción en caso de finalización de trabajo <sup>1</sup>	Operación de compromiso o retrotracción
RST	N/A	N/A	<p>Si la definición de compromiso no está asociada a una transacción global X/Open, se efectúa una retrotracción implícita.</p> <p>Si la definición de compromiso está asociada a una transacción global X/Open, se producen estos eventos:</p> <ul style="list-style-type: none"> <li>• Si el estado de la rama de transacción no es Activo (S1), no se lleva a cabo ninguna acción y la rama de transacción permanece en el mismo estado.</li> <li>• Si el estado de la rama de transacción es Activo (S1), se efectúa una retrotracción implícita.</li> </ul>
PIP	N/A	N/A	<p>Si la definición de compromiso no está asociada a una transacción global X/Open, se efectúa una retrotracción implícita.</p> <p>Si la definición de compromiso está asociada a una transacción global X/Open, la rama de transacción está en estado Desocupado (S2) y permanece en estado Desocupado (S2).</p>
PRP	N/A	WAIT	<p>Si la definición de compromiso no está asociada a una transacción global X/Open<sup>2</sup>, se produce lo siguiente:</p> <ul style="list-style-type: none"> <li>• Se inicia la resincronización para recibir la decisión del iniciador de la operación de compromiso.</li> <li>• Se lleva a cabo la decisión devuelta de comprometer o retrotraer. Se considera una operación explícita.</li> </ul>

Estado	API de último agente	Opción de Acción en caso de finalización de trabajo <sup>1</sup>	Operación de compromiso o retrotracción
PRP	N/A	C	Si la definición de compromiso no está asociada a una transacción global X/Open <sup>2</sup> , se realiza una operación de compromiso implícita.
		R	Si la definición de compromiso no está asociada a una transacción global X/Open, se efectúa una operación de retrotracción implícita.  Si la definición de compromiso está asociada a una transacción global X/Open, se produce lo siguiente: <ul style="list-style-type: none"> <li>• Si el trabajo que inició la transacción finaliza, la transacción permanece en estado preparado hasta que el TM XA la compromete o la retrotrae. El estado de la rama de transacción XA quedará en este caso en estado Preparado (S3).</li> <li>• Si se finaliza el trabajo servidor SQL al que se direcciona el trabajo de la transacción, se efectúa implícitamente una retrotracción forzada. El estado de la rama de transacción XA cambiará en este caso a Completado heurísticamente (S5).</li> </ul>
CIP	N/A	N/A	Se efectúa una operación de compromiso explícita.
LAP	NO	WAIT	1. Se utiliza la resincronización al último agente para recuperar la decisión de comprometer o de retrotraer.
			2. Se lleva a cabo la decisión devuelta de comprometer o retrotraer. Se considera una operación explícita.
LAP	Sí	WAIT	1. Se llama a la API de último agente para recuperar la decisión de comprometer o retrotraer.
			2. Se lleva a cabo la operación de compromiso o retrotracción. Se considera una operación explícita.
LAP	N/A	C	Se efectúa una operación de compromiso implícita.
		R	Se efectúa una operación de retrotracción implícita.
CMT	N/A	N/A	Ya se ha completado una operación de compromiso para esta definición de compromiso y las ubicaciones en sentido descendente. La operación de compromiso ha finalizado.
VRO	N/A	N/A	Los agentes locales y remotos han emitido un voto de solo lectura. Todos los agentes en sentido descendente también deben haber emitido un voto de solo lectura. No es necesario llevar a cabo ninguna acción.

Estado	API de último agente	Opción de Acción en caso de finalización de trabajo <sup>1</sup>	Operación de compromiso o retrotracción
RBR	N/A	N/A	Debe llevarse a cabo una operación de retrotracción. Se efectúa una operación de retrotracción explícita.

**Nota:**

<sup>1</sup> Puede cambiar la opción de Acción en caso de finalización de trabajo con la API Cambiar opciones de compromiso (QTNCHGCO).

<sup>2</sup> Si la definición de compromiso está asociada a una transacción global X/Open, se producen estos eventos:

- Si el trabajo que inició la transacción finaliza, la transacción permanece en estado preparado hasta que el TM XA la compromete o la retrotrae. El estado de la rama de transacción XA quedará en este caso en estado Preparado (S3).
- En el caso de los bloqueos cuyo ámbito se circunscribe a la transacción, si se finaliza el trabajo servidor SQL al que se direcciona el trabajo de la transacción, se efectúa implícitamente una retrotracción forzada. El estado de la rama de transacción XA cambiará en este caso a Completado heurísticamente (S5).

La tabla siguiente muestra las acciones que efectúa el sistema cuando un grupo de activación finaliza y solo hace referencia a las transacciones con bloqueos cuyo ámbito se circunscribe al trabajo. Las acciones del sistema se basan en los factores siguientes:

- El estado de la transacción. (Siempre es Restablecer (RST) cuando un grupo de activación finaliza).
- Cómo finaliza el grupo de activación, de forma normal o anómala.
- Si un recurso de API es el último agente.

**Nota:** Si un recurso de API se registra como último agente, el control de la decisión de compromiso o retrotracción se da al último agente. El resultado de la decisión se considera una operación explícita.

Estado	API de último agente	Tipo de finalización	Operación de compromiso o retrotracción
RST	No	Normal	Se efectúa una operación de compromiso implícita. Si hay conversaciones protegidas, la definición de compromiso se convertirá en el iniciador de raíz de la operación de compromiso.
RST	No	Anómala	Se efectúa una retrotracción implícita.
RST	Sí	Normal	Se llama al programa de salida de API. La API determina la operación de compromiso o retrotracción.
RST	Sí	Anómala	Se llama al programa de salida de API. La API determina la operación de compromiso o retrotracción.

**Conceptos relacionados**

“Control de compromiso durante la finalización anómala del sistema o del trabajo” en la página 63  
Este tema solo hace referencia a las definiciones de compromiso con bloqueos cuyo ámbito se circunscribe al trabajo.

**Referencia relacionada**

API Cambiar opciones de compromiso (QTNCHGCO)

## Control de compromiso durante la finalización normal de un paso de direccionamiento

El sistema finaliza todas las definiciones de compromiso de un trabajo cuando un paso de direccionamiento finaliza con normalidad.

**Nota:** La información siguiente solo atañe a las definiciones de compromiso con bloqueos cuyo ámbito se circunscribe al trabajo.

Un paso de direccionamiento finaliza con normalidad cuando se da una de las siguientes situaciones:

- Una finalización normal de un trabajo por lotes.
- Un fin de sesión normal de un trabajo interactivo.
- El mandato Redireccionar trabajo (RRTJOB), Transferir trabajo (TFRJOB) o Transferir trabajo por lotes (TFRBCHJOB) finaliza el paso de direccionamiento actual e inicia un nuevo paso de direccionamiento.

Cualquier otra finalización de un paso de direccionamiento se considera anómala y se reconoce mediante un código de finalización distinto de cero en el mensaje CPF1164 de finalización del trabajo, en las anotaciones de trabajo.

Antes de finalizar una definición de compromiso durante la finalización de un paso de direccionamiento, el sistema lleva a cabo una operación de retrotracción implícita si la definición de compromiso tiene cambios pendientes. Esto incluye llamar al programa de salida de compromiso y retrotracción de API para cada recurso de compromiso de API asociado a la definición de compromiso. El programa de salida debe completar su proceso en 5 minutos. Una vez que se ha llamado al programa de salida de compromiso y retrotracción de API, el sistema elimina el recurso de compromiso de API automáticamente.

Si se ha definido un objeto de notificación para la definición de compromiso, el objeto se puede actualizar.

### Conceptos relacionados

“Actualizaciones del objeto de notificación” en la página 65

En este tema figuran los cambios no comprometidos en relación con el objeto de notificación.

## Control de compromiso durante la finalización anómala del sistema o del trabajo

Este tema solo hace referencia a las definiciones de compromiso con bloqueos cuyo ámbito se circunscribe al trabajo.

El sistema finaliza todas las definiciones de compromiso de un trabajo cuando el trabajo finaliza de forma anómala. Estas definiciones de compromiso se finalizan durante el proceso de finalización del trabajo. Si el sistema finaliza de forma anómala, el sistema finaliza todas las definiciones de compromiso iniciadas y empleadas por todos los trabajos activos en el momento de la finalización anómala del sistema. Estas definiciones de compromiso se finalizan dentro del proceso de recuperación de la base de datos que se lleva a cabo durante la siguiente IPL tras la finalización anómala del sistema.

**Atención:** La recuperación de las definiciones de compromiso hace referencia a una finalización anómala del sistema o de un trabajo debido a una anomalía de la alimentación, un error de hardware o un error del sistema operativo o el código interno bajo licencia. No utilice el mandato Finalizar trabajo anómalo (ENDJOBABN) para forzar la finalización anómala de un trabajo. La finalización anómala puede ocasionar que los cambios pendientes de las transacciones activas del trabajo que finaliza se comprometan o retrotraigan parcialmente. La siguiente IPL podría intentar llevar a cabo la recuperación de las transacciones parciales del trabajo finalizado con el mandato ENDJOBABN.

El resultado de la recuperación del control de compromiso que efectúa el sistema durante una IPL para un trabajo finalizado con el mandato ENDJOBABN es incierto. Esto se debe a que todos los bloqueos de los recursos de compromiso se liberan cuando el trabajo finaliza de forma anómala. Los cambios pendientes como consecuencia de las transacciones parciales se establecen como disponibles para otros trabajos. Otros programas de aplicación pueden emplear estos cambios pendientes para realizar cambios adicionales erróneos en la base de datos. Del mismo modo, la consiguiente recuperación de IPL que se lleva a cabo posteriormente puede afectar de forma negativa a los cambios efectuados por las aplicaciones tras la finalización anómala del trabajo. Por ejemplo, podría haberse eliminado una tabla SQL durante la recuperación de IPL como acción de retrotracción de una creación de tabla pendiente. Sin embargo, otras aplicaciones podrían haber insertado ya varias filas en la tabla después de que el trabajo haya finalizado de forma anómala.

El sistema efectúa las acciones siguientes para las definiciones de compromiso que finalizan durante una finalización anómala de un trabajo o durante la siguiente IPL tras una finalización anómala del sistema:

- Antes de finalizar una definición de compromiso, el sistema lleva a cabo una operación de retrotracción implícita si la definición de compromiso tiene cambios pendientes, salvo que el proceso de la definición de compromiso se haya interrumpido en medio de una operación de compromiso. Si se finalizase en medio de una operación de compromiso, la transacción se podría retrotraer, resincronizar o comprometer, en función de su estado. El proceso de llevar a cabo la operación de retrotracción implícita o completar la operación de compromiso comprende llamar al programa de salida de compromiso y retrotracción de API para cada recurso de compromiso de API asociado a la definición de compromiso. Una vez que se ha llamado al programa de salida de compromiso y retrotracción de API, el sistema elimina el recurso de compromiso de API automáticamente.

**Atención:** Si se finaliza el trabajo mientras una transacción está pendiente (el estado de la transacción es LAP o PRP), se pueden producir incoherencias en la base de datos (los cambios se podrían comprometer en uno o varios sistemas y retrotraer en otros).

- Si la opción de compromiso de *Acción en caso de finalización de trabajo* es COMMIT, los cambios en este sistema se comprometen si se finaliza el trabajo, independientemente de si los cambios en los demás sistemas que intervienen en la transacción se comprometen o se retrotraen.
- Si la opción de compromiso de *Acción en caso de finalización de trabajo* es ROLLBACK, los cambios en este sistema se retrotraen si se finaliza el trabajo, independientemente de si los cambios en los demás sistemas que intervienen en la transacción se comprometen o se retrotraen.
- Si la opción de compromiso de *Acción en caso de finalización de trabajo* es WAIT, el trabajo no se finalizará hasta que haya terminado la resincronización en el sistema que posee la decisión de compromiso o retrotracción. Para hacer que se finalice el trabajo antes de que se complete la resincronización, debe tomarse una decisión heurística y debe cancelarse la resincronización.

**No** se recomienda finalizar el trabajo o el sistema de forma anómala durante una retrotracción de larga ejecución. En caso de llevar a cabo esta operación, se producirá otra retrotracción al finalizar el trabajo (o durante la siguiente IPL si se finaliza el sistema). La retrotracción consiguiente repetirá el trabajo efectuado por la retrotracción original y tardará mucho más tiempo en ejecutarse.

- Si hay definido un objeto de notificación para la definición de compromiso, se podría actualizar.

Si un proceso finaliza antes de que finalice el control de compromiso y todavía hay activas conversaciones protegidas, la definición de compromiso podría tener que comprometerse o retrotraerse. La acción depende de la opción de estado y la opción de acción si se finaliza el trabajo de la definición de compromiso.

### Conceptos relacionados

“Operaciones de compromiso y retrotracción implícitas” en la página 59

En algunos casos, el sistema inicia una operación de compromiso o retrotracción para una definición de compromiso. Estos tipos de operaciones de compromiso y retrotracción se conocen como *peticiones de compromiso y retrotracción implícitas*.

“Actualizaciones del objeto de notificación”

En este tema figuran los cambios no comprometidos en relación con el objeto de notificación.

## Actualizaciones del objeto de notificación

En este tema figuran los cambios no comprometidos en relación con el objeto de notificación.

- Actualización de un registro efectuada bajo control de compromiso.
- Supresión de un registro bajo control de compromiso.
- Cambio a nivel de objeto efectuado en un objeto DDL local bajo control de compromiso.
- Operación de lectura realizada para un archivo de base de datos abierto bajo control de compromiso. Esto se debe a que la posición del archivo se vuelve a establecer en el último límite de compromiso cuando se efectúa una operación de retrotracción. Si se efectúa una operación de lectura bajo control de compromiso, la posición del archivo se cambia y, por consiguiente, existe un cambio no comprometido para la definición de compromiso.
- Definición de compromiso con uno de los siguientes recursos añadidos siempre se considera que tiene cambios no comprometidos:
  - Recurso de compromiso de API
  - Recurso Distributed Relational Database Architecture (DRDA\*) remoto
  - Recurso de arquitectura de gestión de bases de datos distribuidas (DDM)
  - Recurso LU 6.2

Esto es así porque el sistema no sabe cuando se hace un cambio real en el objeto o en los objetos asociados a estos tipos de recursos. En Tipos de recursos comprometibles encontrará más información sobre cómo añadir y trabajar con estos tipos de recursos.

El sistema realiza actualizaciones en el objeto de notificación basándose en las siguientes maneras de finalizar una definición de compromiso:

- Si un trabajo finaliza con normalidad y no existen cambios no comprometidos, el sistema no coloca la identificación de compromiso de la última operación de compromiso satisfactoria en el objeto de notificación.
- Si se realiza una operación de compromiso implícita para una definición de compromiso a nivel de grupo de activación cuando se finaliza el grupo de activación, el sistema no coloca la identificación de compromiso de la última operación de compromiso satisfactoria en el objeto de notificación.

**Nota:** Nunca se realizan operaciones de compromiso para la definición de compromiso \*DFACTGRP o \*JOB

- Si el sistema, el trabajo o un grupo de activación finaliza de forma anómala antes de la primera operación de compromiso satisfactoria para una definición de compromiso, el sistema no actualiza el objeto de notificación, porque no hay ninguna identificación de último compromiso. Para distinguir entre esta condición y una conclusión normal del programa, este debe actualizar el objeto de notificación con una entrada específica antes de concluir la primera operación de compromiso satisfactoria para la definición de compromiso.
- Si se produce una finalización anómala del trabajo o del sistema tras al menos una operación de compromiso satisfactoria, el sistema coloca la identificación de compromiso de esa operación de

compromiso en el objeto de notificación. Si la última operación de compromiso satisfactoria no especificaba ninguna identificación de compromiso, el objeto de notificación no se actualiza. En el caso de una finalización anómala del trabajo, este proceso del objeto de notificación se efectúa para cada definición de compromiso que estaba activa para el trabajo. En el caso de una finalización anómala del sistema, este proceso del objeto de notificación se efectúa para cada definición de compromiso que estaba activa para todos los trabajos del sistema.

- El sistema actualiza el objeto de notificación con la identificación de compromiso de la última operación de compromiso satisfactoria para esa definición de compromiso si se producen todos estos eventos:
  - Finaliza un grupo de activación.
  - Se efectúa una operación de retrotracción implícita para la definición de compromiso a nivel de grupo de activación.
  - Se ha efectuado al menos una operación de compromiso satisfactoria para esa definición de compromiso.

Si la última operación de compromiso satisfactoria no especificaba ninguna identificación de compromiso, el objeto de notificación no se actualiza. Se efectúa una operación de retrotracción implícita para una definición de compromiso a nivel de grupo de activación si el grupo de activación finaliza de forma anómala o se han producido errores al cerrar los archivos abiertos bajo control de compromiso cuyo ámbito se circunscribía a ese grupo de activación. Para obtener más información sobre cómo puede circunscribir el ámbito de los archivos de base de datos a grupos de activación y cómo pueden finalizarse los grupos de activación, consulte el manual de referencia del lenguaje ILE que utiliza.

- Si existen cambios no comprometidos cuando un trabajo finaliza normalmente y se ha efectuado al menos una operación de compromiso satisfactoria, la identificación de compromiso de la última operación de compromiso satisfactoria se coloca en el objeto de notificación y los cambios no comprometidos se retrotraen. Si la última operación de compromiso satisfactoria no especificaba ninguna identificación de compromiso, el objeto de notificación no se actualiza. Este proceso del objeto de notificación se efectúa para cada definición de compromiso que estaba activa para el trabajo al finalizar el trabajo.
- Si existen cambios no comprometidos cuando se ejecuta el mandato ENDCMTCTL, el objeto de notificación solo se actualiza si la última operación de compromiso satisfactoria especificaba una identificación de compromiso:
  - Para un trabajo por lotes, se retrotraen los cambios no comprometidos y se coloca la identificación de compromiso de la última operación de compromiso satisfactoria en el objeto de notificación.
  - Para un trabajo interactivo, si la respuesta al mensaje de consulta CPA8350 es retrotraer los cambios, se retrotraen los cambios no comprometidos y se coloca la identificación de compromiso de la última operación de compromiso satisfactoria en el objeto de notificación.
  - Para un trabajo interactivo, si la respuesta al mensaje de consulta CPA8350 es comprometer los cambios, el sistema solicita una identificación de compromiso para su uso y los cambios se comprometen. La identificación de compromiso especificada en la pantalla de solicitud se coloca en el objeto de notificación.
  - Para un trabajo interactivo, si la respuesta al mensaje de consulta CPA8350 es cancelar la petición ENDCMTCTL, los cambios pendientes permanecen y el objeto de notificación no se actualiza.

### **Conceptos relacionados**

“Finalizar control de compromiso” en la página 57

En este tema encontrará qué prerequisites hacen falta para finalizar el control de compromiso y cómo finalizar el control de compromiso.

“Control de compromiso durante la finalización de un grupo de activación” en la página 58

El sistema finaliza automáticamente una definición de compromiso a nivel de grupo de activación cuando se finaliza un grupo de activación.



“Control de compromiso durante la finalización normal de un paso de direccionamiento” en la página 63

El sistema finaliza todas las definiciones de compromiso de un trabajo cuando un paso de direccionamiento finaliza con normalidad.

“Control de compromiso durante la finalización anómala del sistema o del trabajo” en la página 63  
Este tema solo hace referencia a las definiciones de compromiso con bloqueos cuyo ámbito se circunscribe al trabajo.

“Tipos de recursos comprometibles” en la página 13

En este tema encontrará una tabla con los distintos tipos de recursos comprometibles.

## **Recuperación de control de compromiso durante la carga del programa inicial tras una finalización anómala**

Cuando se lleva a cabo una carga del programa inicial (IPL) después de que el sistema finalice de forma anómala, el sistema intenta recuperar todas las definiciones de compromiso que estaban activas al finalizar el sistema.

Del mismo modo, cuando se activa una agrupación de discos independiente, el sistema intenta recuperar todas las definiciones de compromiso relacionadas con esa agrupación de discos independiente que estaban activas al desactivarse o finalizar de forma anómala.

Los trabajos servidores de base de datos iniciados por el sistema durante la IPL efectúan la recuperación. El sistema inicia los trabajos servidores de base de datos para manejar las tareas que otros trabajos no pueden o no deben llevar a cabo.

Los trabajos servidores de base de datos se denominan QDBSRVnn, donde nn es un número de dos dígitos. El número de trabajos servidores de base de datos depende del tamaño del sistema. Del mismo modo, el nombre del trabajo servidor de base de datos de una agrupación de discos independiente o un grupo de agrupaciones de discos independientes es QDBSxxxVnn, donde xxx es el número de la agrupación de discos independiente y nn es un número de dos dígitos. Por ejemplo, QDBS035V02 puede ser el nombre del trabajo servidor de base de datos de la agrupación de discos independiente 35.

En Estados de la transacción para el control de compromiso de dos fases se muestran las acciones que efectúa el sistema, en función del estado de la transacción al producirse la anomalía. En el caso de dos estados, PRP y LAP, la acción del sistema queda pendiente.

### **Notas:**

- La siguiente información solo atañe a las definiciones de compromiso con bloqueos cuyo ámbito se circunscribe al trabajo.
- El gestor de transacciones recupera las definiciones de compromiso asociadas a las transacciones XA (tanto si el ámbito de sus bloqueos se circunscribe al trabajo como a la transacción) utilizando las API XA, no el proceso de resincronización descrito en este tema.

El sistema no puede determinar qué acción debe llevar a cabo hasta que efectúa la resincronización con las demás ubicaciones que participaban en la transacción. Esta resincronización se efectúa una vez completada la operación de IPL o activación.

El sistema emplea los trabajos servidores de base de datos para realizar esta resincronización. Las definiciones de compromiso que deben recuperarse están asociadas a los trabajos servidores de base de datos. Durante la IPL, el sistema adquiere todos los bloqueos de registros y otros bloqueos de objetos poseídos por la definición de compromiso antes de que finalizara el sistema. Estos bloqueos son necesarios para proteger los recursos de compromiso locales hasta que se complete la resincronización y se puedan comprometer o retrotraer los recursos.

Se envían mensajes a las anotaciones de trabajo de los trabajos servidores de base de datos para indicar el estado de la resincronización con las ubicaciones remotas. Si la transacción está pendiente, debe completarse la resincronización con la ubicación que posee la decisión para la transacción antes de comprometer o retrotraer los recursos locales.

Cuando se toma la decisión para una transacción, pueden enviarse los siguientes mensajes a las anotaciones del trabajo servidor de base de datos.

**CPI8351**

Se están retro trayendo &1 cambios pendientes

**CPC8355**

Se ha completado la recuperación posterior a la IPL de la definición de compromiso &8 para el trabajo &19/&18/&17.

**CPD835F**

La recuperación de IPL de la definición de compromiso &8 para el trabajo &19/&18/&17 ha fallado.

También se pueden enviar otros mensajes relacionados con la recuperación. Estos mensajes se envían a las anotaciones históricas (QHST). Si se producen errores, también se envían mensajes a la cola de mensajes QSYSOPR.

Puede determinar el progreso de la recuperación utilizando iSeries Navigator, visualizando las anotaciones del trabajo servidor de base de datos o utilizando el mandato Trabajar con definiciones de compromiso (WRKCMTDFN). Aunque iSeries Navigator y la pantalla Trabajar con definiciones de compromiso le permiten forzar un compromiso o una retrotracción del sistema, solo debe emplear esta opción como último recurso. Si prevé que todas las ubicaciones que participaban en la transacción terminarán volviendo a estar operativas, deje que los sistemas se resincronicen. De este modo se garantiza la integridad de las bases de datos.

**Conceptos relacionados**

“Estados de la transacción para el control de compromiso de dos fases” en la página 31

Se establece una definición de compromiso en cada ubicación que forma parte de la red de programas de transacciones. Para cada definición de compromiso, el sistema mantiene un seguimiento del estado de la transacción actual y la transacción anterior.

---

## Gestionar transacciones y control de compromiso

Siga estas instrucciones para gestionar un sistema con control de compromiso.

### Visualizar información de control de compromiso

Puede utilizar iSeries Navigator para visualizar información sobre todas las unidades lógicas de trabajo del sistema. También puede examinar información sobre el trabajo asociado a una transacción (si hay alguno).

**Nota:** Estas operaciones de visualización no muestran el nivel de aislamiento de las aplicaciones SQL.

Para visualizar información, haga lo siguiente:

1. En la ventana **iSeries Navigator**, expanda el servidor que desea utilizar.
2. Expanda **Bases de datos**.
3. Expanda el sistema con el que desea trabajar.
4. Expanda **Transacciones**.

**Nota:** Para ver una transacción que esté asociada a una transacción global X/Open, expanda **Transacciones globales**. Para ver una transacción gestionada por DB2 Universal Database (UDB), expanda **Transacciones de base de datos**.

5. Expanda **Transacciones globales** o **Transacciones de base de datos**.

Esta pantalla muestra la siguiente información:

- ID de unidad de trabajo
- Estado de la unidad de trabajo
- Trabajo
- Usuario
- Número
- Resincronización en curso
- Definición de compromiso

La ayuda en línea proporciona información sobre todas las pantallas de estado y los campos de cada pantalla.

#### **Tareas relacionadas**

“Detectar puntos muertos” en la página 112

Siga estas instrucciones para localizar condiciones de punto muerto.

“Recuperar transacciones tras una anomalía de comunicaciones” en la página 113

Puede utilizar las instrucciones de este tema para manejar las transacciones que llevan a cabo trabajo en un sistema remoto después de que haya fallado la comunicación con ese sistema.

“Cuándo forzar operaciones de compromiso y retrotracción y cuándo cancelar la resincronización” en la página 114

En este tema se explica cuándo y cómo hay que forzar una operación de compromiso o retrotracción y cuándo hay que cancelar la resincronización.

## **Visualizar objetos bloqueados para una transacción**

Puede visualizar los objetos bloqueados para las transacciones globales con bloqueos solo con ámbito de transacción.

Para visualizar los objetos bloqueados para una transacción, siga estos pasos:

1. En la ventana **iSeries Navigator**, expanda el servidor que desea utilizar.
2. Expanda **Bases de datos**.
3. Expanda el sistema con el que desea trabajar.
4. Expanda **Transacciones**.
5. Expanda **Transacciones globales**.
6. Pulse con el botón derecho del ratón la transacción con la que desea trabajar y seleccione **Objetos bloqueados**.

#### **Tareas relacionadas**

“Detectar puntos muertos” en la página 112

Siga estas instrucciones para localizar condiciones de punto muerto.

## **Visualizar trabajos asociados a una transacción**

1. En la ventana **iSeries Navigator**, expanda el servidor que desea utilizar.
2. Expanda **Bases de datos**.
3. Expanda el sistema con el que desea trabajar.
4. Expanda **Transacciones**.
5. Expanda **Transacciones globales** o **Transacciones de base de datos**.

6. Pulse con el botón derecho del ratón la transacción con la que desea trabajar y seleccione **Trabajos**.

Para las transacciones de base de datos y las transacciones globales con bloqueos cuyo ámbito se circunscribe al trabajo, se visualiza una lista de los trabajos asociados a la transacción.

Para las transacciones globales con bloqueos cuyo ámbito se circunscribe a la transacción, se visualiza una lista de los trabajos que tienen este objeto de transacción asociado o que están en espera de que se les asocie este objeto de transacción.

#### **Tareas relacionadas**

“Detectar puntos muertos” en la página 112

Siga estas instrucciones para localizar condiciones de punto muerto.

### **Visualizar el estado de los recursos de una transacción**

1. En la ventana **iSeries Navigator**, expanda el servidor que desea utilizar.
2. Expanda **Bases de datos**.
3. Expanda el sistema con el que desea trabajar.
4. Expanda **Transacciones**.
5. Expanda **Transacciones globales** o **Transacciones de base de datos**.
6. Pulse con el botón derecho del ratón la transacción con la que desea trabajar y seleccione **Estado de recurso**.

### **Visualizar las propiedades de las transacciones**

1. En la ventana **iSeries Navigator**, expanda el servidor que desea utilizar.
2. Expanda **Bases de datos**.
3. Expanda el sistema con el que desea trabajar.
4. Expanda **Transacciones**.
5. Expanda **Transacciones globales** o **Transacciones de base de datos**.
6. Pulse con el botón derecho del ratón la transacción con la que desea trabajar y seleccione **Propiedades**.

## **Optimizar el rendimiento para el control de compromiso**

Para utilizar el control de compromiso se necesitan recursos que pueden afectar al rendimiento del sistema. Son varios los factores que inciden en el rendimiento del sistema en relación con el control de compromiso.

### **Factor que no afecta al rendimiento**

#### **Abrir un archivo**

Si abre un archivo sin especificar la opción de apertura de compromiso, no se utilizará ningún recurso del sistema adicional aunque se haya iniciado una definición de compromiso. Para obtener más información sobre cómo especificar la opción de apertura de compromiso, vea el manual de consulta del lenguaje de alto nivel pertinente.

### **Factores que disminuyen el rendimiento**

#### **Registrar por diario**

Para registrar por diario un archivo se necesitan recursos del sistema. Sin embargo, en la mayoría de los casos, el registro por diario se efectúa mejor con control de compromiso que sin control de compromiso. Si especifica solo imágenes anteriores, el control de compromiso cambia esta opción por la de imágenes tanto anteriores como posteriores mientras está en vigor el control de compromiso. Normalmente esta es una consideración que debe tenerse presente en cuanto al espacio, no al rendimiento.

### **Operación de compromiso**

Si se ha efectuado algún cambio en los recursos registrados por diario durante la transacción, cada compromiso de una transacción añade dos entradas a cada diario relacionado con estos recursos. El número de entradas que se añaden puede aumentar de modo notable para un gran volumen de pequeñas transacciones. Puede interesarle colocar los receptores de diario en una agrupación de discos aparte de los diarios.

### **Operación de retrotracción**

Dado que el control de compromiso debe retrotraer los cambios pendientes registrados en la base de datos, se necesitan más recursos del sistema cada vez que se produce una retrotracción. Además, si hay cambios de registro pendientes, una operación de retrotracción hace que se añadan entradas adicionales al diario.

### **Mandatos Arrancar control de compromiso (STRCMTCTL) y Finalizar control de compromiso (ENDCMTCTL)**

El sistema debe hacer frente a una actividad adicional cada vez que se inicia una definición de compromiso con el mandato STRCMTCTL y cada vez que se finaliza con el mandato ENDCMTCTL. Evite utilizar los mandatos STRCMTCTL y ENDCMTCTL para cada transacción. Utilícelos solo cuando sea necesario. Puede establecer una definición de compromiso al principio de un trabajo interactivo y utilizarla durante todo el trabajo.

### **Utilizar más de un diario para las transacciones de control de compromiso**

Con el compromiso de dos fases, los archivos que se abren bajo control de compromiso pueden registrarse por diario en más de un diario. No obstante, al utilizar más de un diario se emplean recursos del sistema adicionales para gestionar la definición de compromiso. Además, el uso de más de un diario puede hacer que la recuperación sea más complicada.

### **Bloqueo de registros**

El bloqueo de registros puede afectar a otras aplicaciones. El número de registros bloqueados en un trabajo determinado aumenta los recursos globales del sistema empleados para el trabajo. Las aplicaciones que necesitan acceder al mismo registro deben esperar a que finalice la transacción.

### **Petición SEQONLY(\*YES)**

Si solicita la opción SEQONLY(\*YES) (mediante el mandato OVRDBF o el intento implícito del programa de aplicación de utilizar SEQONLY(\*YES)) y el archivo se abre para entrada solo bajo control de compromiso con LCKLVL(\*ALL), la opción cambia a SEQONLY(\*NO). Esta opción puede afectar al rendimiento de los archivos de entrada, porque los registros no se bloquearán.

### **Solicitar un cambio a nivel de registro para un archivo de base de datos cuando está activo el proceso de salvar mientras activo**

La petición de hacer un cambio a nivel de registro bajo control de compromiso para un archivo de base de datos podría retardarse si la definición de compromiso está en un límite de compromiso y se ejecuta una operación de salvar mientras activo en otro trabajo. Esto puede suceder cuando se registra por diario un archivo en el mismo diario que algunos de los objetos de la petición de salvar.

**Nota:** La columna Estado de la pantalla Trabajar con trabajos activos (mandato WRKACTJOB) visualiza CMTW (espera de compromiso) cuando un trabajo está retenido debido a un proceso de punto de control de salvar mientras activo.

### **Comprometer o retrotraer cambios cuando está activo el proceso de salvar mientras activo**

La operación de compromiso o retrotracción se podría retardar en un límite de compromiso cuando se ejecuta una operación de salvar mientras activo en otro trabajo. Esto puede suceder cuando anteriormente se ha añadido un recurso de compromiso de API a la definición de compromiso, salvo que el recurso de API se haya añadido con la API QTNADDCR y el campo *Permitir proceso de salvar normal* tenga el valor Y.

Como el trabajo se retiene durante la petición de compromiso o retrotracción, y dado que solo puede realizarse una petición de compromiso o retrotracción para una única definición de

compromiso cada vez, los trabajos con más de una definición de compromiso con recursos de compromiso de API añadidos pueden impedir que se lleve a cabo una operación de salvar mientras activo.

**Nota:** Si utiliza la nueva función de salvar con transacciones parciales, el objeto se puede salvar sin finalizar una definición de compromiso.

### **Solicitar un cambio a nivel de objeto cuando está activo el proceso de salvar mientras activo**

La petición de hacer un cambio a nivel de objeto bajo control de compromiso se podría retardar si la definición de compromiso está en un límite de compromiso y se ejecuta una operación de salvar mientras activo en otro trabajo. Esto puede suceder cuando se efectúa un cambio a nivel de objeto mientras se ejecuta la operación de salvar mientras activo para la biblioteca en la que se encuentra el objeto. Por ejemplo, la operación de crear tabla SQL bajo control de compromiso para la tabla MYTBL de la biblioteca MYSQLLIB se podría retardar si hay una operación de salvar mientras activo en ejecución para la biblioteca MYSQLLIB.

**Nota:** Si el tiempo de espera es superior a 60 segundos, se envía el mensaje de consulta CPA8351 para preguntar al usuario si se debe seguir esperando o cancelar la operación.

### **Añadir un recurso de API mediante la API QTNADDCR**

La petición de añadir un recurso de compromiso de API mediante la API QTNADDCR se podría retardar si todas las definiciones de compromiso del trabajo están en un límite de compromiso y se ejecuta una operación de salvar mientras activo en otro trabajo.

#### **Notas:**

1. Si el tiempo de espera es superior a 60 segundos, se envía el mensaje de consulta CPA8351 para preguntar al usuario si se debe seguir esperando o cancelar la operación.
2. Esto no se aplica a los recursos de API añadidos mediante la API QTNADDCR si el campo *Permitir proceso de salvar normal* tiene el valor Y.

## **Factores que mejoran el rendimiento**

### **Utilizar un diario predeterminado**

El uso de un diario predeterminado puede mejorar el rendimiento si se cierran y se vuelven a abrir todos los archivos bajo control de compromiso mientras la definición de compromiso está activa. Sin embargo, si se emplea un diario predeterminado con OMTJRNE(\*NONE) disminuye el rendimiento de las operaciones de compromiso y retrotracción.

### **Seleccionar un último agente**

El rendimiento mejora cuando se selecciona un último agente ya que se necesitan menos interacciones entre el sistema y el último agente durante una operación de compromiso. Sin embargo, si se produce una anomalía de comunicaciones durante una operación de compromiso, la operación de compromiso no se completará hasta que finalice la resincronización, sea cuál sea el valor de la opción de esperar resultado. Este tipo de error es poco habitual pero esta opción permite a la persona que escribe la aplicación plantearse el impacto negativo de hacer que el usuario espere de forma indefinida a que se complete la resincronización cuando se produce un error. Esta posibilidad debe sopesarse frente a la mejora del rendimiento que proporciona la optimización del último agente durante las operaciones de compromiso satisfactorias. En general, esta consideración es más significativa para los trabajos interactivos que para los trabajos por lotes.

El valor predeterminado establece que se permite al sistema seleccionar un último agente pero el usuario puede modificar este valor con la API QTNCHGCO.

### **No utilizar la opción Esperar resultado**

Cuando los recursos remotos están bajo control de compromiso, el rendimiento mejora si la opción Esperar resultado tiene el valor N (No) y todos los sistemas remotos soportan la

terminación anómala que se presupone. El sistema establece que la opción Esperar resultado es igual a N para las aplicaciones DRDA y DDM cuando se establece la primera conexión con un sistema remoto. Las aplicaciones APPC deben establecer explícitamente la opción Esperar resultado; de lo contrario, se utilizará el valor predeterminado Y.

### **Seleccionar la opción Aceptar exclusión**

El rendimiento mejora cuando se selecciona la opción Aceptar exclusión.

### **Seleccionar la opción Voto de solo lectura**

El rendimiento mejora cuando se selecciona la opción Voto de solo lectura.

#### **Conceptos relacionados**

Gestión de diarios

“Definición de compromiso para el compromiso de dos fases: indicar aceptar exclusión” en la página 40

Normalmente, el gestor de transacciones de cada ubicación de la red de programas de transacciones participa en cada operación de compromiso o retroacción. Para mejorar el rendimiento, puede configurar algunas o la totalidad de las ubicaciones de una transacción a fin de permitir al gestor de transacciones que indique aceptar exclusión.

“Definición de compromiso para el compromiso de dos fases: permitir voto de solo lectura” en la página 34

Normalmente un gestor de transacciones participa en las dos fases del proceso de compromiso. Para mejorar el rendimiento del proceso de compromiso, puede configurar algunas o la totalidad de las ubicaciones de una transacción a fin de permitir al gestor de transacciones que emita un voto de solo lectura.

### **Minimizar bloqueos**

Un método de uso habitual para minimizar los bloqueos de registros consiste en liberar el bloqueo de registros. (Esta técnica no funciona si se ha especificado LCKLVL(\*ALL)).

Este ejemplo explica cómo minimizar el bloqueo de registros a base de liberarlo. Una aplicación de mantenimiento de un solo archivo suele seguir estos pasos:

1. Visualizar una solicitud que pide una identificación del registro que hay que cambiar.
2. Recuperar el registro solicitado.
3. Visualizar el registro.
4. Permitir que el usuario de la estación de trabajo haga el cambio.
5. Actualizar el registro.

En la mayoría de los casos, el registro se bloquea desde que se accede al registro solicitado hasta que se actualiza. El tiempo de espera del registro podría ser más largo para otro trabajo que esté esperando el registro. Para evitar bloquear un registro mientras el usuario de la estación de trabajo se plantea efectuar un cambio, libere el registro una vez que se recupere de la base de datos (antes de que aparezca la pantalla del registro). A continuación deberá acceder al registro de nuevo antes de actualizarlo. Si el registro ha cambiado entre el momento en que se ha liberado y el momento en que se ha vuelto a acceder a él, informe de esta circunstancia al usuario de la estación de trabajo. El programa puede determinar si el registro ha cambiado salvando uno o varios campos del registro original y comparándolos con los campos del mismo registro después de su recuperación, de la siguiente manera:

- Utilice un campo de recuento de actualizaciones en el registro y añada 1 al campo inmediatamente antes de una actualización. El programa salva el valor original y lo compara con el valor del campo cuando se vuelve a recuperar el registro. Si se ha producido un cambio, se informa del mismo al usuario de la estación de trabajo y el registro vuelve a aparecer. El campo de recuento de actualizaciones solo cambia si se produce una actualización. El registro se libera mientras el usuario de la estación de trabajo se plantea la posibilidad de efectuar un cambio. Si utiliza esta técnica, debe emplearla en todos los programas que actualicen el archivo.

- Salve el contenido de todo el registro de datos y compárelo con el registro la próxima vez que se recupere.

En los dos casos anteriores, la secuencia de operaciones impide el simple uso de datos descritos externamente en RPG donde se utilizan los mismos nombres de campos en el registro maestro y en el archivo de pantalla. El uso de los mismos nombres de campos (en RPG) no funciona porque los cambios del usuario de la estación de trabajo se recubren cuando se vuelve a recuperar el registro. Puede solucionar este problema moviendo los datos del registro a una estructura de datos. O bien, si utiliza la palabra clave DDS RTNDDTA, puede continuar, utilizando datos descritos externamente. La palabra clave RTNDDTA permite al programa volver a leer los datos de la pantalla sin que el sistema operativo tenga que mover los datos de la pantalla al programa. Así el programa podrá:

1. Solicitar la identificación del registro.
2. Recuperar el registro solicitado de la base de datos.
3. Liberar el registro.
4. Salvar los campos empleados para determinar si el registro ha cambiado.
5. Visualizar el registro y esperar a que responda el usuario de la estación de trabajo.

Si el usuario de la estación de trabajo cambia el registro en la pantalla, el programa utiliza la secuencia siguiente:

1. Recuperar de nuevo el registro de la base de datos.
2. Comparar los campos salvados para determinar si el registro de base de datos ha cambiado. Si ha cambiado, el programa libera el registro y envía un mensaje cuando aparece el registro.
3. Recuperar el registro desde la pantalla, ejecutando una operación de lectura con la palabra clave RTNDDTA, y actualizar el registro en la base de datos.
4. Continuar con la siguiente solicitud lógica, porque no hay más registros para liberar si el usuario de la estación de trabajo cancela la petición.

LCKLVL(\*CHG) y LCKLVL(\*CS) funcionan en esta situación. Si se emplea LCKLVL(\*ALL), deberá liberar el bloqueo del registro mediante una operación de compromiso o retrotracción.

#### **Tareas relacionadas**

“Detectar puntos muertos” en la página 112

Siga estas instrucciones para localizar condiciones de punto muerto.

## **Gestionar el tamaño de las transacciones**

Otra manera de minimizar los bloqueos de registros consiste en gestionar el tamaño de la transacción.

En esta información, las transacciones son interactivas. (También se puede emplear el control de compromiso para las aplicaciones por lotes, que a menudo se pueden considerar como una serie de transacciones). Muchas de estas consideraciones son válidas para las aplicaciones por lotes, que se explican en Control de compromiso para aplicaciones por lotes.

Puede bloquear un máximo de 500 000 000 de registros durante una transacción para cada diario asociado a la transacción. Puede reducir este límite empleando un archivo de opciones de consulta (QAQQINI). Utilice el parámetro QRYOPLIB del mandato Cambiar atributos de consulta (CHGQRYA) para especificar un archivo de opciones de consulta que deberá utilizar un trabajo. Utilice el valor COMMITMENT\_CONTROL\_LOCK\_LEVEL del archivo de opciones de consulta como límite de bloqueos para el trabajo.

Al elegir el nivel de bloqueo para los registros, plantéese el tamaño de las transacciones. Utilice el tamaño para determinar cuánto tiempo permanecen bloqueados los registros antes de que finalice una transacción. Tiene que decidir si una operación de compromiso o retrotracción para control de compromiso se limita a un solo uso de la tecla Intro o si la transacción consta de múltiples usos de la tecla Intro.



**Nota:** Cuanto más corta es la transacción, antes puede continuar y completarse el trabajo que espera para iniciar el proceso de punto de control de salvar mientras activo.

Por ejemplo, en el caso de una aplicación de entrada de pedidos, un cliente puede solicitar varios artículos en un solo pedido que requiere una actualización del registro de detalle de pedido y del registro maestro de inventario para cada artículo del pedido. Si la transacción se ha definido como el pedido completo y cada uso de la tecla Intro solicita un artículo, todos los registros que intervienen en el pedido permanecen bloqueados durante todo el pedido. Por consiguiente, podría suceder con frecuencia que los registros utilizados (tales como los registros maestros de inventario) permanecieran bloqueados largos períodos de tiempo, lo que impediría el proceso de otro trabajo. Si todos los artículos se entran con una sola tecla Intro utilizando un subarchivo, se minimizar la duración de los bloqueos para todo el pedido.

En general, se recomienda minimizar la cantidad y la duración de los bloqueos para que varios usuarios de estación de trabajo puedan acceder a los mismos datos sin largos períodos de espera. Para ello, puede hacer que no se retengan bloqueos mientras el usuario entra datos en la pantalla. Algunas aplicaciones podrían no necesitar que hubiera más de un usuario de estación de trabajo que accediera a los mismos datos. Por ejemplo, en una aplicación de asientos de efectivo con muchos registros de artículos abiertos por cliente, el método habitual consiste en bloquear todos los registros y retardarlos hasta que un usuario de estación de trabajo termine de hacer el asiento del efectivo correspondiente a un recibo determinado.

Si el usuario de la estación de trabajo pulsa la tecla Intro varias veces para una transacción, es posible llevar a cabo la transacción en varios segmentos. Por ejemplo:

- El primer segmento es una consulta en la que el usuario de la estación de trabajo solicita la información.
- El segundo segmento es una confirmación de la intención del usuario de la estación de trabajo de efectuar toda la transacción.
- El tercer segmento es la recuperación y actualización de los registros afectados.

Este método permite restringir el bloqueo de registros a un solo uso de la tecla Intro.

Este método de consulta previa se utiliza normalmente en las aplicaciones en las que se toma una decisión a partir de información visualizada. Por ejemplo, en una aplicación de reserva de vuelos, un cliente podría querer informarse sobre los horarios de vuelo, los vuelos de conexión y la disposición de asientos que están disponibles antes de tomar una decisión sobre qué vuelo tomar. Cuando el cliente haya tomado la decisión, se entra la transacción. Si la transacción falla (el vuelo está lleno en ese momento), se puede emplear la función de retroacción y entrar otra petición. Si los registros estuvieran bloqueados desde la primera consulta hasta que se tomara una decisión, otro empleado de reservas estaría esperando hasta que se completara la otra transacción.

### **Conceptos relacionados**

*“Control de compromiso en las aplicaciones por lotes” en la página 27*

Las aplicaciones por lotes pueden necesitar o no el control de compromiso. En algunos casos, una aplicación por lotes puede llevar a cabo una sola función de leer un archivo de entrada y actualizar un archivo maestro. Sin embargo, puede emplear el control de compromiso para este tipo de aplicación si es importante volver a iniciarla tras una finalización anómala.

### **| Compromiso leve**

| El compromiso leve le permite mejorar el rendimiento de las transacciones de base de datos. El compromiso leve limita el número de veces que el sistema escribe entradas de diario asociadas a una transacción en el disco.

| El compromiso leve puede mejorar el rendimiento de las transacciones, pero podría provocar la pérdida de una o varias transacciones si se produjera una anomalía del sistema. El control de compromiso tradicional de DB2 Universal Database para iSeries garantiza la durabilidad de las transacciones; es decir, una transacción que ya se haya comprometido persiste en el sistema. El compromiso leve no proporciona

| esta durabilidad, aunque sí que garantiza la atomicidad de la transacción. Dicho de otra manera, el sistema garantiza un límite de compromiso, pero si se produjera una anomalía del sistema, se podría perder una transacción completa o varias.

| Para utilizar el compromiso leve, tanto si es para un trabajo concreto como si es para todo el sistema, especifique \*NO en la variable de entorno *QIBM\_TN\_COMMIT\_DURABLE*. Puede cambiar esta variable con el mandato Añadir variable de entorno (ADDENVVAR).

| Por ejemplo, para solicitar un compromiso leve desde un trabajo concreto, ejecute este mandato desde el trabajo:

```
| ADDENVVAR ENVVAR (QIBM_TN_COMMIT_DURABLE) VALUE (*NO)
```

| Para solicitar un compromiso leve en todo el sistema, ejecute este mandato:

```
| ADDENVVAR ENVVAR (QIBM_TN_COMMIT_DURABLE) VALUE (*NO) LEVEL (*SYS)
```

| **Nota:** Debe poseer la autorización \*JOBCTL para establecer esta variable de entorno en todo el sistema.

| Si la variable de entorno *QIBM\_TN\_COMMIT\_DURABLE* del caso predeterminado no se ha añadido o si la variable de entorno se ha establecido con un valor distinto de \*NO, el sistema no utiliza el compromiso leve; en su lugar, el sistema utiliza el control de compromiso tradicional para que quede garantizada la durabilidad de las transacciones.

| Puede comprobar la existencia de esta nueva variable de entorno, así como su valor y su nivel, si existe, utilizando el mandato Trabajar con variables de entorno (WRKENVVAR).

| En el caso de algunas transacciones, el sistema operativo opta por no hacer caso de la petición de compromiso leve y, en su lugar, realiza un compromiso tradicional. Esto sucede en algunos entornos complejos, en los que se necesitan múltiples conexiones de base de datos o en los que se desarrollan operaciones DDL. El sistema operativo puede determinar cuándo conviene realizar la petición y cuándo tiene más sentido efectuar una operación de compromiso tradicional. Por lo tanto, no resulta perjudicial solicitar compromisos leves en esos entornos.

---

## Casos prácticos y ejemplos: control de compromiso

Si lee los casos prácticos y ejemplos de este tema, verá cómo configura una empresa el control de compromiso. Lea los ejemplos de código de programas que utilizan el control de compromiso.

El siguiente caso práctico ilustra cómo implementa la compañía JKL Toy el control de compromiso para hacer un seguimiento de las transacciones en su base de datos local.

Los siguientes ejemplos proporcionan código de ejemplo para el control de compromiso. El problema de práctica es un programa RPG que implementa el control de compromiso. Incluye un flujo de lógica que muestra qué sucede en cada paso del recorrido.

### Caso práctico: control de compromiso

La compañía JKL Toy utiliza el control de compromiso para proteger los registros de base de datos de fabricación e inventario. Este caso práctico muestra cómo utiliza la compañía JKL Toy el control de compromiso para transferir una parte del departamento de inventario al departamento de fabricación.

En el caso práctico: gestión de diarios se incluye una descripción del entorno de red de la compañía JKL Toy. El caso práctico que se describe a continuación muestra cómo funciona el control de compromiso en el servidor de producción, JKLPROD.

Este caso práctico pone de manifiesto las ventajas de utilizar el control de compromiso en dos ejemplos. El primer ejemplo muestra cómo el programa de inventario de la compañía, el programa A, puede funcionar sin control de compromiso y los problemas que pueden producirse. El segundo ejemplo muestra cómo funciona el programa con control de compromiso.

La compañía JKL Toy emplea un programa de aplicación de inventario, el programa A, en el servidor JKLPROD. El programa A utiliza dos registros. Un registro hace un seguimiento de los artículos almacenados en el almacén. Otro registro hace un seguimiento de los artículos retirados del almacén y empleados en producción.

## Programa A sin control de compromiso

Suponga que el siguiente programa de aplicación no utiliza el control de compromiso. El sistema bloquea los registros leídos para actualización. El procedimiento siguiente describe cómo el programa de aplicación hace un seguimiento de un diodo al retirarse del almacén y transferirse a una cuenta corriente:

- El programa A bloquea y recupera el registro de almacén. (Esta acción puede requerir una espera si el registro está bloqueado por otro programa).
- El programa A bloquea y recupera el registro de producción. (Esta acción también puede requerir una espera). El programa A ahora tiene bloqueados ambos registros y ningún otro programa puede cambiarlos.
- El programa A actualiza el registro de almacén. Con esta acción se libera el registro, de modo que ahora está disponible para que cualquier otro programa pueda leerlo a fin de actualizarlo.
- El programa A actualiza el registro de producción. Con esta acción se libera el registro, de modo que ahora está disponible para que cualquier otro programa pueda leerlo a fin de actualizarlo.

Si no se utiliza el control de compromiso, es necesario resolver un problema para que este programa funcione correctamente en todas las circunstancias. Por ejemplo, se produce un problema si el programa A no actualiza ambos registros como consecuencia de un error de un trabajo o del sistema. En este caso, los dos archivos no son coherentes (los diodos se eliminan del registro de almacén pero no se añaden al registro de producción). La utilización del control de compromiso le permite asegurarse de que todos los cambios que implica la transacción se llevan a cabo o que los archivos vuelven a su estado original si se interrumpe el proceso de la transacción.

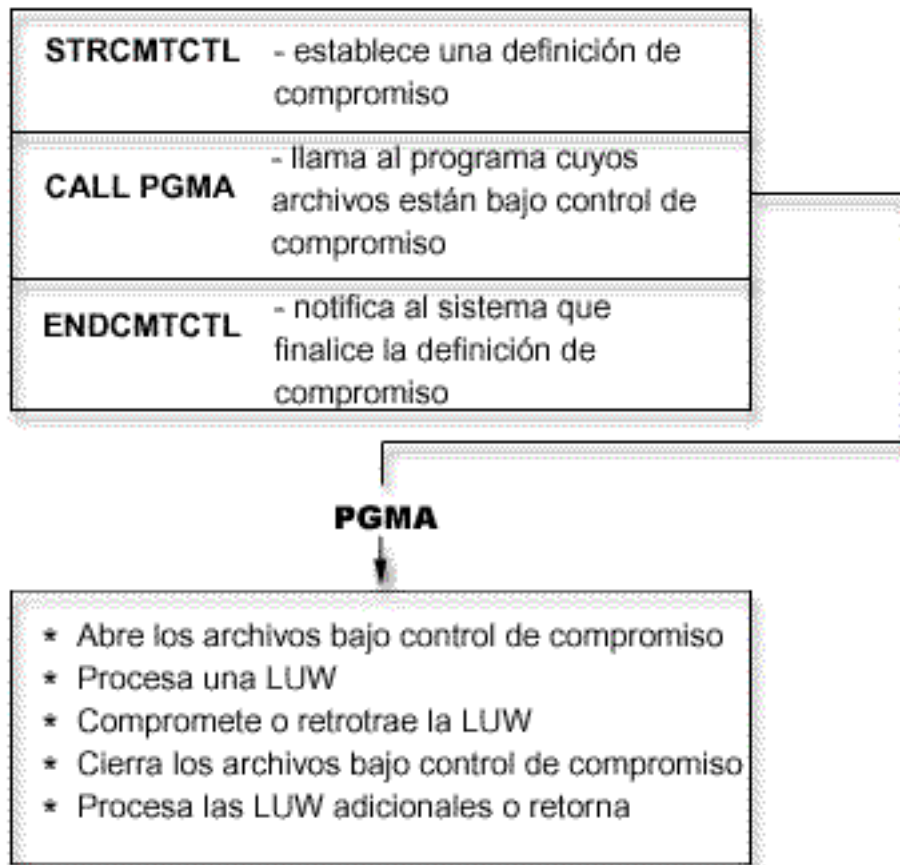
## Programa A con control de compromiso

Si se utiliza el control de compromiso, el ejemplo anterior cambia de la siguiente manera:

1. Se inicia el control de compromiso.
2. El programa A bloquea y recupera el registro de almacén. (Esta acción puede requerir una espera si el registro está bloqueado por otro programa).
3. El programa A bloquea y recupera el registro de producción. (Esta acción también puede requerir una espera). El programa A ahora tiene bloqueados ambos registros y ningún otro programa puede cambiarlos.
4. El programa A actualiza el registro de almacén y el control de compromiso mantiene el bloqueo del registro.
5. El programa A actualiza el registro de producción y el control de compromiso mantiene el bloqueo del registro.
6. El programa A compromete la transacción. Los cambios efectuados en el registro de almacén y el registro de producción se establecen como permanentes en los archivos. Los cambios se graban en el diario, que supone que aparecerán en el disco. El control de compromiso libera los bloqueos de ambos registros. Ahora los registros están disponibles para que cualquier otro programa pueda leerlos a fin de actualizarlos.

Dado que el control de compromiso mantiene los bloqueos de ambos registros hasta que se compromete la transacción, no puede producirse una situación en la que un registro se actualice y el otro no. Si se produce un error de un paso de direccionamiento o del sistema antes de que se comprometa la transacción, el sistema elimina (retrotrae) los cambios efectuados para que los archivos se actualicen al punto en el que se comprometió la última transacción.

Para cada paso de direccionamiento en el que los archivos deben estar bajo control de compromiso, se producen los pasos que se muestran en la figura siguiente:



Las operaciones que se efectúan bajo control de compromiso se registran por diario en el diario. La entrada de diario de inicio del control de compromiso aparece tras la primera entrada de apertura de archivo bajo control de compromiso. Esto es debido a que la primera entrada de apertura de archivo determina qué diario se utiliza para el control de compromiso. La entrada de diario de la primera operación de apertura se emplea para comprobar las operaciones de apertura posteriores con el fin de asegurar que todos los archivos utilizan el mismo diario.

Cuando se produce un error de un trabajo o del sistema, los recursos bajo control de compromiso se actualizan a un límite de compromiso. Si se inicia una transacción pero no se completa antes de que finalice un paso de direccionamiento, el sistema retrotrae esa transacción y esta no aparece en el archivo después de que finalice el paso de direccionamiento. Si el sistema finaliza de forma anómala antes de que se complete una transacción, el sistema retrotrae esa transacción y esta no aparece en el archivo tras una carga del programa inicial (IPL) del código interno bajo licencia satisfactoria posterior. En el momento en que se produce una retrotracción, las entradas que se han invertido se colocan en el diario.

Por ejemplo, suponga que la empresa JKL tiene 100 diodos en almacén. Fabricación retira 20 del almacén, con lo que el saldo nuevo asciende a 80. La actualización de la base de datos genera dos entradas de diario de imagen anterior (100) e imagen posterior (80).

Imagine que el sistema ha finalizado de forma anómala tras registrar por diario las entradas, pero antes de alcanzar el punto de compromiso o el punto de retrotracción. Después de la IPL, el sistema lee la entrada de diario y actualiza el registro de base de datos correspondiente. Esta actualización genera dos entradas de diario que invierten la actualización: la primera entrada es la imagen anterior (80) y la segunda entrada es la imagen posterior (100).

Una vez completada satisfactoriamente la IPL tras la finalización anómala, el sistema elimina (o retrotrae) los cambios de la base de datos que no están comprometidos. En el ejemplo anterior, el sistema elimina los cambios del registro de almacén dado que no existe una operación de compromiso en el diario para esa transacción. En este caso, la imagen anterior del registro de almacén se coloca en el archivo. El diario contiene los cambios retrotraídos y una indicación que señala que se ha producido una operación de retrotracción.

### Conceptos relacionados

Caso práctico: gestión de diarios

## Problema de práctica de control de compromiso

Este problema de práctica pretende ayudarle a entender el control de compromiso y sus requisitos. En estos pasos se presupone que ya está familiarizado con el programa bajo licencia i5/OS y con el programa de utilidad de archivo de datos y que ha leído este temario.

El flujo de lógica le ayudará a comprender mejor este programa de práctica del control de compromiso.

**Nota:** Si utiliza los ejemplos de código, indica que acepta los términos que figuran en: “Información de licencia de código y declaración de limitación de responsabilidad” en la página 118.

Antes de empezar este problema, debe seguir estos pasos:

- Cree una biblioteca especial para este problema de práctica. En las instrucciones, la biblioteca se denomina CMTLIB. Donde vea CMTLIB, en su lugar especifique el nombre de la biblioteca que va a utilizar.
- Cree archivos fuente y una descripción de trabajo.

Para utilizar el control de compromiso, siga estos pasos:

1. Cree un archivo físico denominado ITMP (archivo maestro de artículos). La especificación de descripción de datos (DDS) de este archivo es:

```
10  A  R  ITMR
20  A  ITEM    2
30  A  ONHAND  5  0
40  A  K  ITEM
```

2. Cree un archivo físico denominado TRNP (archivo de transacciones). Este archivo se utiliza como archivo de anotaciones de transacciones. La DDS de este archivo es:

```
10  A  R  TRNR
20  A  QTY    5  0
30  A  ITEM    2
40  A  USER   10
```

3. Cree un archivo lógico denominado TRNL (lógica de transacciones). Este archivo se utiliza para ayudarle a volver a iniciar la aplicación. El campo *USER* es el tipo de secuencia LIFO. Las DDS de este archivo son:

```
10          R  TRNR          LIFO
20  A  R  TRNR          PFILE (TRNP)
30  A  K  USER
```

4. Especifique el mandato STRDFU y cree una aplicación DFU denominada ITMU para el archivo ITMP. Acepte los valores predeterminados que le ofrece DFU en la definición de la aplicación.
5. Escriba el mandato CHGDTA ITMU y especifique los siguientes registros para el archivo ITMP:

Artículo	En existencias
AA	450
BB	375
CC	4000

6. Finalice el programa mediante F3. Esta entrada proporciona algunos datos para los que se ejecutará el programa.
7. Cree el programa CL de proceso de artículos (ITMPCSC) tal como se indica a continuación:

```
PGM
DCL &USER *CHAR LEN(10)
RTVJOBA USER(&USER)
CALL ITMPCS PARM(&USER)
ENDPGM
```

Este es el programa de control que llama al programa ITMPCS. Recupera el nombre de usuario y lo pasa al programa de proceso. Esta aplicación presupone que se utilizan nombres de usuario exclusivos.

8. Cree un archivo de pantalla denominado ITMPCSD a partir de las DDS tal como se indica a continuación.

Hay dos formatos, el primero para la pantalla de solicitud básica y el segundo para permitir al operador revisar la última transacción entrada. El programa ITMPCS utiliza este archivo de pantalla.

SEQNBR \*... .. 1 ... .. 2 ... .. 3 ... .. 4 ... .. 5 ... .. 6 ... .. 7 ..

1.00	A		R	PROMPT			
2.00	A					CA03(93 'Fin del programa')	
3.00	A					CA04(94 'Revisar última')	
4.00	A					SETOFF(64 'No hay ningún registro para revisar')	
5.00	A					1 2'TRANSACCIONES DE INVENTARIO'	
6.00	A					3 2'Cantidad'	
7.00	A			QTY	5 0I	+1	
8.00	A	61				ERRMSG('Cantidad +	
9.00	A					no válida' 61)	
10.00	A					+5'ARTÍCULO'	
11.00	A			ITEM	2 I	+1	
12.00	A	62				ERRMSG('Número de artículo +	
13.00	A					no válido' 62)	
14.00	A	63				ERRMSG('Se ha producido +	
15.00	A					una retrotracción' 63)	
16.00	A	64				24 2'Se ha pulsado CF4 y +	
17.00	A					no hay ninguna +	
18.00	A					transacción para +	
19.00	A					este usuario'	
20.00	A					DSPATR(HI)	
21.00	A					23 2'CF4 Revisar última +	
22.00	A					transacción'	
23.00	A			R	REVV		
24.00	A					1 2'TRANSACCIONES DE INVENTARIO'	
25.00	A					+5'REVISAR ÚLTIMA TRANSACCIÓN'	
26.00	A					3 2'Cantidad'	
27.00	A			QTY	5 0	+1EDTCDE(Z)	
28.00	A					+5'Artículo'	
29.00	A			ITEM	2	+1	

9. Observe el flujo de lógica que se proporciona en el flujo de lógica del programa de práctica de control de compromiso.
10. Especifique el mandato STRSEU y escriba el fuente tal como se indica a continuación:

SEQNBR \*... .. 1 ... .. 2 ... .. 3 ... .. 4 ... .. 5 ... .. 6 ... .. 7 ..

```

1.00 FITMP UF E K DISK
2.00 F* KCOMIT
3.00 FTRNP O E DISK
4.00 F* KCOMIT
5.00 FTRNL IF E K DISK
6.00 F TRNR KRENAMETRNR1
7.00 FITMPCSD CF E WORKSTN
8.00 C* Especificar parámetro con nombre de usuario para archivo -TRNP-
9.00 C *ENTRY PLIST
10.00 C PARM USER 10
11.00 C LOOP TAG
12.00 C EXFMTPROMPT
13.00 C* Buscar CF3 para fin del programa
14.00 C 93 DO Fin del programa
15.00 C SETON LR
16.00 C RETRN
17.00 C END
18.00 C* Buscar CF4 para revisar última transacción
19.00 C 94 DO Revisar última
20.00 C* Comprobar si existe un registro para este usuario en el archivo -TRNL-
21.00 C USER CHAINTRNR1 64 No encontrado
22.00 C 64 GOTO LOOP
23.00 C EXFMTREVV
24.00 C GOTO LOOP
25.00 C END
26.00 C* Acceder al registro del artículo
27.00 C ITEM CHAINITMR 62 No encontrado
28.00 C* Manejar condición de tipo -no encontrado-
29.00 C 62 GOTO LOOP
30.00 C* Existe una cantidad suficiente
31.00 C ONHAND SUB QTY TEST 50 61 Menos
32.00 C* Manejar cantidad insuficiente
33.00 C 61 DO
34.00 C* Liberar registro de artículo bloqueado por cadena para actualización
35.00 C EXCPTRLSITM
36.00 C GOTO LOOP
37.00 C END
38.00 C* Cambiar ONHAND y actualizar registro de artículo
39.00 C Z-ADDTEST ONHAND
40.00 C UPDATITMR
41.00 C* Probar para condiciones de simulación especiales
42.00 C ITEM IFEQ 'CC'
43.00 C* Simular necesidad del programa de retrotracción
44.00 C QTY IFEQ 100
45.00 C SETON 63 Simular retrotracción
46.00 C* ROLBK
47.00 C GOTO LOOP
48.00 C END
49.00 C* Simular cancelación anómala del programa por división por cero
50.00 C* El operador debe responder -C- al mensaje de consulta
51.00 C QTY IFEQ 101
52.00 C Z-ADD0 ZERO 30
53.00 C TESTZ DIV ZERO TESTZ 30 Se produce un mensaje
54.00 C END
55.00 C* Simular cancelación anómala del trabajo por DSPLY.
56.00 C* Operador debe emitir petición del sistema para otro
57.00 C* trabajo y cancelar este con OPTION(*IMMED)
58.00 C QTY IFEQ 102
59.00 C 'CC=102' DSPLY Se produce mensaje
60.00 C END
61.00 C END ITEM=CC
62.00 C* Escribir el archivo -TRNP-
63.00 C WRITETRNR

```

```

64.00    C* Comprometer actualización de -ITMP- y escribir en -TRNP-
65.00    C*                                COMMIT
66.00    C                                GOTO LOOP
67.00    OITMR    E                                RLSITM

```

11. Especifique el mandato CRTRPGPGM para crear el programa ITMPCS a partir del fuente especificado en el paso anterior.
12. Escriba el mandato CALL ITMPCSC, pulse Intro y pulse F4. Aparece un mensaje que le indica que no hay entradas para este operador.
13. Especifique los datos siguientes para ver si el programa se ejecuta correctamente:

Cantidad	Artículo
3	AA
4	BB

14. Pulse F4. Aparece la pantalla de revisión con el artículo BB entrado por última vez. Especifique los datos siguientes:

Cantidad	Artículo
5	FF (Aparecerá un mensaje de número de artículo no válido)
9000	BB (Aparecerá un mensaje de error de cantidad insuficiente)
100	CC (Aparecerá un mensaje de retrotracción)
102	CC (Se producirá la operación RPG DSPLY. Pulse la tecla Intro).
101	CC (El programa visualizará un mensaje de consulta que le indicará que se ha producido una condición de división por cero o finalizará, según el valor del atributo del trabajo INQMSGRPY. Si aparece el mensaje de consulta, entre C para cancelar el programa RPG y, a continuación, C para cancelar el programa CL en la consulta siguiente. Esto simula una condición de error imprevista).

15. Escriba el mandato Visualizar datos DSPDTA ITMP.  
Vea si los registros AA y BB se han actualizado correctamente. Los valores deben ser AA = 447, BB = 371 y CC = 3697. Tenga en cuenta que se han restado cantidades de CC, pero no se han escrito los registros de transacciones.
16. Cree un receptor de diario para el control de compromiso. Utilice el mandato Crear receptor de diario (CRTJRNRCV) para crear un receptor de diario denominado RCVR1 en la biblioteca CMRLIB. Especifique un umbral de al menos 5000 KB. Conviene que utilice un umbral superior, si el sistema tiene espacio suficiente, para maximizar el tiempo transcurrido entre la generación de nuevos receptores de diario y así minimizar la repercusión que los cambios de diarios demasiado frecuentes pueden tener en el rendimiento.
17. Cree un diario para control de compromiso. Utilice el mandato Crear diario (CRTJRN) para crear un diario denominado JRNTEST en la biblioteca CMTLIB. Como este diario solo se utiliza para control de compromiso, especifique MNGRCV(\*SYSTEM) DLTRCV(\*YES). En el parámetro JRNRCV, especifique el receptor de diario que creó en el paso 16.
18. Emplee el mandato Arrancar registro por diario de archivo físico (STRJRNPF) con los parámetros FILE(CMTLIB/ITMP CMTLIB/TRNP) JRN(CMTLIB/JRNTEST) para registrar por diario los archivos que se utilizarán para el control de compromiso.  
El parámetro IMAGES utiliza el valor predeterminado \*AFTER, que significa que solo aparecerán en el diario los cambios de la imagen posterior de los registros. Ahora los archivos ITMP y TRNP han iniciado el registro por diario.



Normalmente se salvan los archivos tras iniciar el registro por diario. No puede aplicar los cambios registrados por diario a un archivo restaurado que no tiene el mismo JID que las entradas de diario. Como en este problema de práctica no es necesario que aplique cambios registrados por diario, puede saltarse el paso de salvar los archivos registrados por diario.

19. Escriba el mandato CALL ITMPCSC y entre las transacciones siguientes:

Cantidad	Artículo
5	AA
6	BB

Finalice el programa pulsando F3.

20. Escriba el mandato Visualizar diario: DSPJRN CMTLIB/JRNTEST.

Observe las entradas que aparecen en el diario. En el diario aparece la misma secuencia de entradas (R UP = actualización de ITMP seguida de R PT = registro añadido a TRNP) que la que ha llevado a cabo el programa. Esto se debe a que hay un archivo lógico definido sobre el archivo físico TRNP y el sistema altera temporalmente el valor predeterminado de RPG. Si no existe ningún archivo lógico, se utiliza el valor SEQONLY(\*YES) que supone RPG y aparece un bloque de entradas PT ya que los registros se conservan en el almacenamiento intermedio de RPG hasta que el bloque esté completo.

21. Cambie el programa CL ITMPCSC, de la siguiente manera (las sentencias nuevas se indican con un asterisco):

```

PGM
DCL &USER *CHAR LEN(10)
RTVJOBA USER(&USER)
* STRCMTCTL LCKLVL(*CHG)
CALL ITMPCS PARM(&USER)
* MONMSG MSGID(RPG9001) EXEC(ROLLBACK)
* ENDCMTCTL
ENDPGM

```

El mandato STRCMTCTL configura el entorno de control de compromiso. La palabra LCKLVL especifica que los registros leídos para actualización pero no actualizados pueden liberarse durante la transacción. El mandato MONMSG maneja los mensajes de escape RPG y lleva a cabo una retrotracción si el programa RPG finaliza de forma anómala. El mandato ENDCMTCTL finaliza el entorno de control de compromiso.

22. Suprima el programa ITMPCSC existente y créelo de nuevo.
23. Cambie el programa RPG para eliminar los símbolos de comentario de las sentencias 2.00, 4.00, 46.00 y 65.00. Ahora el fuente está preparado para el uso con el control de compromiso.
24. Suprima el programa ITMPCS existente y créelo de nuevo. Ahora el programa está preparado para ejecutarse bajo control de compromiso.
25. Escriba el mandato CALL ITMPCSC y las transacciones siguientes:

Cantidad	Artículo
7	AA
8	BB

26. Utilice Petición del Sistema y solicite la opción para visualizar el trabajo actual. Cuando aparezca la pantalla Visualizar trabajo, seleccione la opción 16 para solicitar la visualización del estado de control de compromiso.

Observe los valores de la pantalla. Debe haber dos compromisos ya que se han ejecutado dos sentencias de compromiso en el programa.

27. Pulse F9 para ver una lista de los archivos bajo control de compromiso y la cantidad de actividad de cada archivo.
28. Vuelva al programa y finalícelo pulsando F3.
29. Escriba DSPJRN CMTLIB/JRNTEST y observe las entradas de los archivos y las entradas de diario especiales de control de compromiso.

**Entrada**

C BC  
C SC

C CM

C EC

**Significado**

Se ha ejecutado un mandato STRCMTCTL.

Iniciar ciclo de compromiso. Esto se produce cuando la primera operación de base de datos de la transacción hace que se inserte, actualice o suprima un registro como parte del control de compromiso.

Se ha producido una operación de compromiso.

Se ha ejecutado un mandato ENDCMTCTL.

Las imágenes anteriores y posteriores de control de compromiso (tipos R UB y R UP) se producen automáticamente aunque inicialmente se haya solicitado IMAGES(\*AFTER) para el diario.

30. Escriba el mandato CALL ITMPCSC y las transacciones siguientes:

**Cantidad**

12  
100

**Artículo**

AA  
CC (Esta es la condición para simular la necesidad de que una aplicación utilice la retrotracción. El registro CC del archivo ITMP, actualizado por la sentencia RPG 40.00, se retrotrae).

31. Pulse F4 para determinar la última transacción entrada.

La última transacción comprometida es la entrada del artículo AA.

32. Utilice Petición del Sistema y solicite la opción Visualizar trabajo actual. Cuando aparezca la pantalla Visualizar trabajo, solicite la visualización del estado de control de compromiso.

Observe los valores de la pantalla y las modificaciones que han experimentado como consecuencia de la retrotracción.

33. Vuelva al programa.

34. Vuelva a la pantalla básica de solicitud y finalice el programa pulsando F3.

35. Escriba el mandato DSPJRN CMTLIB/JRNTEST.

Observe las entradas adicionales que aparecen en el diario para el uso de la entrada de retrotracción (entrada C RB). Cuando se retrotrae el registro ITMP, se colocan tres entradas en el diario. Esto se debe a que cualquier cambio efectuado en el archivo de base de datos bajo control de compromiso genera una entrada anterior (R BR) y una entrada posterior (R UR).

36. Visualice las entradas con el código de diario R y estos tipos de entradas: UB, UP, BR y UR. Utilice la opción 5 para visualizar las entradas completas. Como el campo *Cantidad* tiene el formato decimal empaquetado, utilice F11 para solicitar una visualización hexadecimal. Fíjese en las siguientes condiciones:

- El valor en existencias del registro ITMP del registro UB
- Cómo el registro UP reduce el valor en existencias
- Cómo el registro BR es igual que el registro UP
- Cómo el registro UR devuelve el valor tal como se ha visualizado inicialmente para el registro UB

La última entrada es la entrada RB para el final de la retrotracción.

37. Teclee el mandato CALL ITMPCSC; pulse Intro y, después, F4. Fíjese en la última transacción entrada.

38. Escriba las transacciones siguientes:

**Cantidad**

13

**Artículo**

AA

**Cantidad**  
101

**Artículo**

CC (Esta es la condición para simular una condición de error imprevisto, lo que hace que el programa finalice. La simulación se produce dividiendo un campo por 0. El programa visualizará un mensaje de consulta o finalizará, según el valor del atributo del trabajo INQMSGRPY. Si aparece el mensaje de consulta, entre C para finalizar el programa. Como el programa CL se ha modificado para supervisar los errores del programa RPG, la segunda consulta que se producía no tiene lugar).

39. Escriba el mandato DSPJRN CMTLIB/JRNTEST.

Se ha producido el mismo tipo de manejo de retrotracción, pero esta vez la retrotracción se ha debido al parámetro EXEC del mandato MONMSG en el programa CL en lugar del programa RPG. Visualice las dos entradas RB para ver qué programa las ha generado.

40. Escriba el mandato WRKJOB y anote el nombre del trabajo totalmente calificado para utilizarlo más adelante.
41. Escriba el mandato CALL ITMPCSC y entre la transacción siguiente:

**Cantidad**  
14  
102

**Artículo**

AA

CC (Se producirá la operación RPG DSPLY para la cola de mensajes externa. Utilice la tecla PetSis y seleccione la opción 1 del menú de petición del sistema para transferir a un trabajo secundario).

42. Inicie sesión en el segundo trabajo y vuelva a establecer el entorno.
43. Escriba el mandato ENDJOB y especifique el nombre del trabajo totalmente calificado identificado anteriormente y OPTION(\*IMMED). Esto simula una finalización anómala del trabajo o del sistema.
44. Espere unos 30 segundos, escriba el mandato CALL ITMPCSC y pulse F4.  
Observe la última transacción comprometida. Debe ser el artículo AA entrado anteriormente.
45. Vuelva a la pantalla básica de solicitud y finalice el programa pulsando F3.
46. Escriba el mandato DSPJRN CMTLIB/JRNTEST.

Se ha producido el mismo tipo de manejo de retrotracción, pero esta vez es el sistema el que ha originado la retrotracción en lugar de uno de los programas. El programa QWTPITPP, que es el programa de finalización anómala de gestión del trabajo, ha escrito la entrada RB.

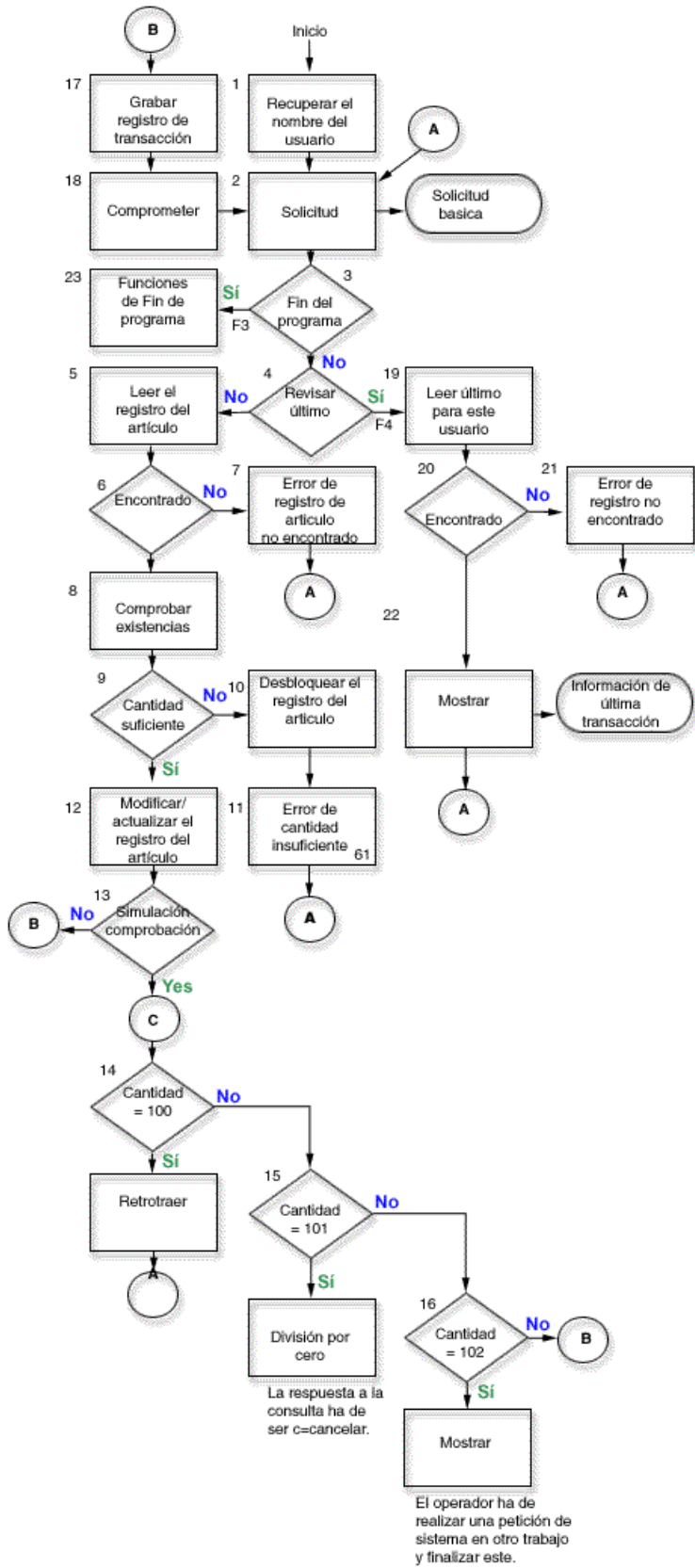
Ya ha utilizado las funciones básicas de control de compromiso. Puede ejecutar el control de compromiso en sus aplicaciones o probar algunas de las demás funciones, como las siguientes:

- Utilizar un objeto de notificación
- Bloquear registros que solo se leen con LCKLVL(\*ALL)
- Bloquear múltiples registros en un mismo archivo con LCKLVL(\*ALL)

## Flujo de lógica del problema de práctica

Esta imagen muestra el flujo del problema de práctica del control de compromiso.

Los detalles de cada uno de los pasos que se ven en la imagen están en: "Pasos asociados al flujo de lógica del programa de práctica" en la página 87.



## Pasos asociados al flujo de lógica del programa de práctica

Estos pasos están asociados al flujo de lógica del problema de práctica.

1. Recupere el nombre de usuario que se ha pasado como parámetro. Este valor se utiliza para escribir en el archivo TRNP y también para recuperar la última transacción entrada por cada operador. Esta aplicación presupone que se utilizan nombres de usuario exclusivos para los operadores.
2. Solicite la pantalla básica mediante el nombre de formato PROMPT.
3. Si se pulsa F3, inicie una función de fin de programa.
4. Si se pulsa F4, inicie una rutina para acceder a la última transacción entrada por el operador.
5. Lea el registro de artículo utilizando el campo *ITEM*. Dado que este es un archivo de actualización, esta petición bloquea el registro.
6. Compruebe si existe una condición de no encontrado en el archivo ITMP.
7. Si no existe ningún registro ITMP, active el indicador 62 para generar el mensaje de error y vuelva al paso 2.
8. Reste la cantidad solicitada (QTY) del saldo de existencias (ONHAND) en un área de trabajo.
9. Compruebe si existe una cantidad suficiente para satisfacer la petición.
10. Si no existe una cantidad suficiente, libere el bloqueo del registro en el archivo ITMP. Este paso es necesario, porque la cantidad es insuficiente.
11. Active el indicador 61 para señalar un mensaje de error de visualización de cantidad insuficiente y vuelva al paso 2.
12. Cambie el campo ONHAND para el saldo nuevo y actualice el registro ITMR.
13. Compruebe si existe una entrada especial en el campo ITEM que se pueda usar para simular condiciones en las que sea necesario retrotraer (ROLLBACK).
14. Compruebe si existe QTY=100. Emita una operación de retrotracción (ROLLBACK). De este modo se simula una condición en la que el programa detecta la necesidad de retrotraer.
15. Compruebe si existe QTY=101. Provoque una excepción en el programa que generará un mensaje de consulta. Utilice dividir por cero para esta función. El operador debe entrar C para cancelar el programa salvo que la opción INQMSGRPH de la descripción de trabajo proporcione una respuesta automática. De este modo se simula una condición en la que se ha producido un error imprevisto y el operador cancela el programa.
16. Compruebe si existe QTY=102. Emita una operación de visualizar con consulta. De este modo se detiene el programa en este paso y se permite el uso de la tecla PetSis para acceder a otro trabajo. Cancele el trabajo de actualización. De este modo se simula una condición en la que se ha producido una finalización anómala del trabajo o del sistema en medio de un límite de compromiso.
17. Escriba el registro de transacción en TRNP.
18. Comprometa los registros para la transacción y vuelva al paso 2.
19. Lea el primer registro de la vía de acceso del archivo TRNL, utilizando USER como clave. Dado que este archivo está en orden LIFO, será el último registro de transacción entrado por este usuario.
20. Compruebe si existe una condición de registro no encontrado en el archivo TRNL, provocada si el archivo no contiene entradas para este usuario.
21. Si no hay ningún registro para este usuario, active el indicador 64 para generar un mensaje de error y vuelva al paso 2.
22. Visualice la última transacción entrada para este usuario. Esta información se puede utilizar si el operador olvida lo que se ha entrado anteriormente, o cuando se reinicia la transacción. Cuando el operador responda, vuelva al paso 2.
23. Lleve a cabo las funciones de fin de programa que hagan falta.

## Ejemplo: utilizar un archivo de anotaciones de transacciones para iniciar una aplicación

Este ejemplo proporciona código de ejemplo e instrucciones acerca de cómo utilizar un archivo de anotaciones de transacciones a fin de iniciar una aplicación tras una finalización anómala.

**Nota:** Si utiliza los ejemplos de código, indica que acepta los términos que figuran en: "Información de licencia de código y declaración de limitación de responsabilidad" en la página 118.

Se emplea un *archivo de anotaciones de transacciones* para volver a iniciar una aplicación tras una anomalía del sistema o del trabajo cuando no se utiliza un objeto de notificación. A menudo se utiliza un archivo de anotaciones de transacciones en las aplicaciones interactivas para resumir los efectos de una transacción.

Por ejemplo, en una aplicación de entrada de pedidos, normalmente se escribe un registro en un archivo de anotaciones de transacciones para cada artículo solicitado. El registro contiene el artículo solicitado, la cantidad y el precio. En una aplicación de cuentas a pagar, se escribe un registro en un archivo de anotaciones de transacciones para cada número de cuenta que debe recibir un cargo. Este registro normalmente contiene información como, por ejemplo, el número de cuenta, el importe cargado y el proveedor.

En muchas de las aplicaciones en las que ya existe un archivo de anotaciones de transacciones, un usuario de estación de trabajo puede solicitar información sobre la última transacción entrada. Añadiendo el control de compromiso a las aplicaciones en las que ya existe un archivo de anotaciones de transacciones, puede:

- Asegurarse de que los archivos de base de datos se actualizan a un límite de compromiso.
- Simplificar el reinicio de la transacción.

Debe poder identificar de forma exclusiva el usuario de la estación de trabajo si emplea un archivo de anotaciones de transacciones para volver a iniciar las aplicaciones bajo control de compromiso. Si se emplean nombres de perfil de usuario exclusivos en el sistema, ese nombre de perfil puede colocarse en un campo del registro de anotaciones de transacciones. Este campo puede utilizarse como la clave del archivo.

Los ejemplos siguientes suponen que se utiliza un archivo de inventario de pedidos para llevar a cabo las transacciones y que ya existe un archivo de anotaciones de transacciones. El programa realiza estas tareas:

1. Solicitar al usuario de la estación de trabajo una cantidad y un número de artículo.
2. Actualizar la cantidad en el archivo maestro de producción (PRDMSTP).
3. Escribir un registro en el archivo de anotaciones de transacciones (ISSLOGL).

Si la cantidad de inventario en existencias es insuficiente, el programa rechaza la transacción. El usuario de la estación de trabajo puede preguntar al programa dónde se ha interrumpido la entrada de datos, dado que se ha escrito la información de número de artículo, descripción, cantidad, nombre de usuario y fecha en el archivo de anotaciones de transacciones.

### DDS para el archivo físico PRDMSTP

SEQNBR \*... .. 1 ... .. 2 ... .. 3 ... .. 4 ... .. 5 ... .. 6 ... .. 7

1.00	A	R PRDMSTR		TEXT('Registro maestro')
2.00	A	PRODCT	3	COLHDG('Número' 'producto')
3.00	A	DESCRP	20	COLHDG('Descripción')
4.00	A	ONHAND	5 0	COLHDG('Cantidad en' 'existencias')
5.00	A			EDTCDE(Z)
6.00	A	K PRODCT		

## DDS para el archivo físico ISSLOGP utilizado en ISSLOGP

```

SEQNBR *... .. 1 ... .. 2 ... .. 3 ... .. 4 ... .. 5 ... .. 6 ... .. 7

1.00      A          R ISSLOGR          TEXT('Registro de anotaciones de producto')
2.00      A          PRODC T           3          COLHDG('Número' 'producto')
3.00      A          DESCRP           20         COLHDG('Descripción')
4.00      A          QTY              3 0        COLHDG('Cantidad')
5.00      A          EDTCDE(Z)
6.00      A          USER            10         COLHDG('Nombre' 'usuario')
7.00      A          DATE              6 0        EDTCDE(Y)
8.00      A          COLHDG('Fecha')

```

## DDS para el archivo lógico ISSLOGL

```

SEQNBR *... .. 1 ... .. 2 ... .. 3 ... .. 4 ... .. 5 ... .. 6 ... .. 7

1.00      A          LIFO
2.00      A          R ISSLOGR        PFILE(ISSLOGP)
3.00      A          K USER

```

## DDS para el archivo de pantalla PRDISSD utilizado en el programa

```

SEQNBR *... .. 1 ... .. 2 ... .. 3 ... .. 4 ... .. 5 ... .. 6 ... .. 7 ..

1.00      A          REF(ISSLOGP)
2.00      A          R PROMPT
3.00      A          CA03(98 'Fin del programa')
4.00      A          CA02(97 'Lugar donde estoy')
5.00      A          1 20'EMITE PROCESO'
6.00      A          3 2'Cantidad'
7.00      A          QTY              R          I          +1
8.00      A 62          ERRMSG('Cantidad +
9.00      A          insuficiente' 62)
10.00     A          +6'Producto'
11.00     A          PRODC T           R          I          +1
12.00     A 61          ERRMSG('Registro de producto +
13.00     A          no encontrado' 62)
14.00     A 55          15 2'No existe ningún registro anterior'
15.00     A          24 2'CF2 Última transacción'
16.00     A          R RESTART
17.00     A          1 20'INFORMACIÓN DE +
18.00     A          ÚLTIMA TRANSACCIÓN'
19.00     A          5 2'Producto'
20.00     A          PRODC T           R          +1
21.00     A          7 2'Descripción'
22.00     A          DESCRP           R          +1
23.00     A          9 2'Cantidad'
24.00     A          QTY              R          +1REFFLD(QTY)

```

Este proceso se representa en el **flujo del programa**.

## Flujo de programa



El código de operación COMMIT RPG se especifica después de que se actualice el archivo PRDMSTP y se escriba el registro en el archivo de anotaciones de transacciones. Dado que cada solicitud presentada al operador representa un límite para una nueva transacción, la transacción se considera que es una transacción de un solo Intro.

El nombre de usuario se pasa al programa cuando se le llama. La vía de acceso del archivo de anotaciones de transacciones se define en la secuencia LIFO (último en entrar, primero en salir) de modo que el programa puede acceder fácilmente al último registro entrado.

El usuario de la estación de trabajo puede volver a iniciar el programa tras una anomalía del sistema o del trabajo empleando la misma función que ha identificado dónde se ha detenido la entrada de datos. No es necesario añadir código adicional al programa. Si en este momento utiliza un archivo de anotaciones de transacciones, pero no lo emplea para averiguar dónde se encuentra usted, añada el nombre de usuario al archivo de anotaciones de transacciones (suponiendo que los nombres de usuario sean exclusivos) y siga este procedimiento en el programa.

El siguiente ejemplo muestra el programa RPG empleado. Las sentencias necesarias para el control de compromiso se marcan con flechas (==>).



## Programa RPG

```

SEQNBR *... 1 ... 2 ... 3 ... 4 ... 5 ... 6 ... 7 ..
=>1.00  FPRDMSTP UP E           K           DISK      KCOMIT
=>2.00  FISSLOGL IF E           K           DISK      KCOMIT
3.00   PRDISSD CP E           WORKSTN
4.00
5.00   *ENTRY   PLIST           USER      10
6.00   C*
7.00   C* Inicializar campos en registro anotaciones transacciones
8.00   C*
9.00   C           MOVE UDATE   DATE
10.00  C*
11.00  C* Bucle básico del proceso
12.00  C*
13.00  C           LOOP      TAG
14.00  C           EXFMTPROMPT
15.00  C 98           GOTO END           Fin del programa
16.00  C 97           DO              Lugar donde estoy
17.00  C           EXSR WHERE
18.00  C           GOTO LOOP
19.00  C           END
20.00  C           PRODC   CHAINPRDMSTR           61      No encontrado
21.00  C 61           GOTO LOOP
22.00  C           ONHAND SUB QTY   TEST   50   62      Inferior a
23.00  C 62           DO              Insuficiente
24.00  C           EXCPTRLSMST           Liberar bloqueo
25.00  C           GOTO LOOP
26.00  C           END
27.00  C*
28.00  C* Actualizar registro maestro y enviar a salida registro anotaciones transacciones
29.00  C*
30.00  C           Z-ADDTEST   ONHAND
31.00  C           UPDATPRDMSTR
32.00  C           WRITEISSLOGR
=>33.00 C           COMMIT
34.00  C           GOTO LOOP
35.00  C*
36.00  C* Proceso de fin del programa
37.00  C*
38.00  C           END      TAG
39.00  C           SETON           LR
40.00  C*
41.00  C* Subrutina WHERE para peticiones de tipo "Lugar donde estoy"
42.00  C*
43.00  C           WHERE   BEGSR
44.00  C           USER   CHAINISSLOGL           55      No encontrado
45.00  C N55           EXFMTRSTART
46.00  C           ENDSR
47.00  C OPRDMSTR E           RLSMST

```

## Programa CL empleado para llamar al programa RPG PRDISS

```

SEQNBR *... 1 ... 2 ... 3 ... 4 ... 5 ... 6 ... 7 ..
1.00   PGM
2.00   DCL           &USER *CHAR LEN(10)
3.00   STRCMTCTL   LCKLVL(*CHG)
4.00   RTVJOBA    USER(&USER)
5.00   CALL       PRDISS PARM(&USER)
6.00   MONMSG     MSGID(RPG9001) EXEC(ROLLBACK)
7.00   ENDCMTCTL
8.00   ENDPGM

```

Para emplear el control de compromiso en este programa, se suele especificar el nivel de bloqueo \*CHG. El cambio bloquea el registro hasta que se ejecuta una operación de compromiso. Observe que si hay una

cantidad de inventario insuficiente, el registro se libera explícitamente. (Si el registro no se liberara explícitamente en el programa, se liberaría cuando el siguiente registro se leyera para actualización en el archivo).

En este ejemplo, no existe ninguna ventaja adicional respecto del uso del nivel de bloqueo \*ALL. Si se utiliza \*ALL, debe utilizarse una operación de compromiso o retroacción para liberar el bloqueo cuando existiera una cantidad insuficiente.

El código anterior es un programa CL que llama al programa RPG PRDISS. Fíjese en el uso de los mandatos STRCMTCTL/ENDCMTCTL. El nombre de usuario exclusivo se recupera (mandato RTVJOBA) y se pasa al programa. El uso del mandato MONMSG para hacer que se lleve a cabo una operación de retroacción se describe en Ejemplo: utilizar un programa de proceso estándar para decidir si hay que reiniciar la aplicación.

### **Conceptos relacionados**

“Ejemplo: utilizar un programa de proceso estándar para iniciar una aplicación” en la página 98  
Programa de proceso estándar es una manera de volver a iniciar la aplicación utilizando un archivo de base de datos como objeto de notificación para todas las aplicaciones. En este enfoque se presupone que los nombres de perfil de usuario son exclusivos de cada usuario en todas las aplicaciones que emplean el programa estándar.

“Ejemplo: un objeto de notificación exclusivo para cada programa” en la página 93  
El uso de un solo objeto de notificación exclusivo para cada trabajo permite emplear una identificación de compromiso descrita externamente, aunque podría haber múltiples usuarios del mismo programa.

## **Ejemplo: utilizar un objeto de notificación para iniciar una aplicación**

Cuando se inicia un programa tras una finalización anómala, este puede buscar una entrada en el objeto de notificación. Si la entrada existe, el programa puede iniciar de nuevo una transacción. Una vez que transacción se ha vuelto a iniciar, el programa borra el objeto de notificación para impedir que se inicie otra vez la misma transacción.

El objeto de notificación se puede usar de distintas maneras:

- Si la identificación de compromiso se coloca en un archivo de base de datos, consulte este archivo para determinar dónde hay que volver a iniciar cada aplicación o trabajo de estación de trabajo.
- Si la identificación de compromiso se coloca en una cola de mensajes para una estación de trabajo determinada, puede enviarse un mensaje a los usuarios de la estación de trabajo cuando inicien la sesión para informarles de la última transacción comprometida.
- Si la identificación de compromiso se coloca en un archivo de base de datos que tiene una clave o un nombre de usuario, el programa puede leer este archivo al iniciarse. Si existe un registro en el archivo, inicie de nuevo el programa. El programa puede enviar un mensaje al usuario de la estación de trabajo en el que se identifique la última transacción comprometida. El programa lleva a cabo la recuperación necesaria. Si existía un registro en el archivo de base de datos, el programa suprime ese registro al final del programa.
- Para una aplicación por lotes, la identificación de compromiso se puede colocar en un área de datos que contiene los totales, los valores de conmutadores y otra información de estado necesaria para volver a iniciar la aplicación. Cuando se inicia la aplicación, esta accede al área de datos y verifica los valores almacenados en ella. Si la aplicación finaliza normalmente, el área de datos se configura para la siguiente ejecución.
- Para una aplicación por lotes, la identificación de compromiso se puede enviar a una cola de mensajes. Un programa que se ejecute cuando se inicie la aplicación puede recuperar los mensajes de la cola y volver a iniciar los programas.

Se pueden usar varias técnicas para volver a iniciar las aplicaciones, en función de las necesidades de las aplicaciones. Al elegir la técnica, tenga en cuenta la siguiente información:

- Cuando hay múltiples usuarios de un programa al mismo tiempo, no se puede utilizar una sola área de datos como objeto de notificación, porque, después de una finalización anómala del sistema, la identificación de compromiso de cada uno de los usuarios recubriría la de los demás en el área de datos.
- El método que utilice para suprimir información del objeto de notificación debe manejar la situación cuando se produce un error inmediatamente después de utilizar la información:
  - Si la información se suprime de inmediato, no existiría si se produjera otra anomalía antes de procesar la transacción interrumpida.
  - Se recomienda no suprimir la información del objeto de notificación hasta que se procese satisfactoriamente la transacción interrumpida. En este caso, existirá más de una entrada en el objeto de notificación si este es un archivo de base de datos o una cola de mensajes.
  - El programa debe acceder al último registro si hay más de una entrada.
- No es posible utilizar un objeto de notificación para proporcionar al usuario de la estación de trabajo la última transacción comprometida ya que el objeto de notificación solo se actualiza si se produce un error del sistema o del trabajo o si existen cambios no comprometidos al finalizar un trabajo normalmente.
- Si se visualiza información al usuario de la estación de trabajo, debe ser significativa. Para ello podría hacer falta que el programa convirtiese los códigos contenidos en el objeto de notificación en información que ayudase al usuario a iniciar de nuevo.
- La información para iniciar de nuevo se debe visualizar si el usuario de la estación de trabajo la necesita. Se necesita lógica adicional en el programa para evitar que se visualice de nuevo información cuando ya no sea significativa.
- Un único objeto de notificación y un programa de proceso estándar pueden proporcionar una función de volver a iniciar si el objeto de notificación es un archivo de base de datos. Los programas que necesitan la capacidad de volver a iniciar llaman a este programa de proceso estándar para minimizar los cambios realizados en cada programa individual.

#### Conceptos relacionados

“Objeto de notificación de compromiso” en la página 52

*Objeto de notificación* es una cola de mensajes, un área de datos o un archivo de base de datos que contiene información que identifica la última transacción satisfactoria llevada a cabo para una definición de compromiso determinada si esa definición de compromiso no ha finalizado con normalidad.

### Ejemplo: un objeto de notificación exclusivo para cada programa

El uso de un solo objeto de notificación exclusivo para cada trabajo permite emplear una identificación de compromiso descrita externamente, aunque podría haber múltiples usuarios del mismo programa.

En los ejemplos siguientes, se emplea un archivo de base de datos como objeto de notificación y solo se utiliza en este programa.

El programa tiene dos archivos de base de datos (PRDMSTP y PRDLOCP) que deben actualizarse en relación con los recibos de inventario. El archivo de pantalla empleado por el programa se denomina PRDRCTD. Como objeto de notificación se utiliza un archivo de base de datos, PRDRCTP. Este objeto de notificación se define en el programa como un archivo y también se utiliza como la definición de una estructura de datos para la función de notificación.

**Nota:** Si utiliza los ejemplos de código, indica que acepta los términos que figuran en: “Información de licencia de código y declaración de limitación de responsabilidad” en la página 118.

### DDS para archivo físico PRDLOCP

SEQNBR \*... .. 1 ... .. 2 ... .. 3 ... .. 4 ... .. 5 ... .. 6 ... .. 7

1.00	A	R PRDLOCR		TEXT('Registro de ubicación')
2.00	A	PRDCT	3	COLHDG('Número' 'producto')

3.00	A	LOCATN	6	COLHDG('Ubicación')
4.00	A	LOCAMT	5 0	COLHDG('Cantidad' 'ubicación')
5.00	A			EDTCDE(Z)
6.00	A	K PRODC		
7.00	A	K LOCATN		

### DDS para archivo de pantalla PRDRCTD

SEQNBR	*...	...	1	...	...	2	...	...	3	...	...	4	...	...	5	...	...	6	...	...	7	..
1.00	A																					REF(PRDMSTP)
2.00	A	R	PROMPT																			
3.00	A																					CA03(98 'Fin del programa')
4.00	A																					SETOFF(71 'REINICIAR')
5.00	A								1	20												'RECIBOS DE PRODUCTOS'
6.00	A								3	2												'Cantidad'
7.00	A		QTY			3	0I															+1
8.00	A																					+6'Producto'
9.00	A		PRODC	R			I															+1
10.00	A	61																				ERRMSG('Ningún registro +
11.00	A																					encontrado en el +
12.00	A																					archivo maestro' 62)
13.00	A																					+6'Ubicación'
14.00	A		LOCATN	R			I															+1REFFLD(LOCATN PRDLOCP)
15.00	A	62																				ERRMSG('Ningún registro +
16.00	A																					encontrado en el +
17.00	A																					archivo de ubicación' 62)
18.00	A								9	2												'Última transacción'
19.00	A	71																				+6'Esta es la información +
20.00	A																					de reinicio'
21.00	A																					DSPATR(HI BL)
22.00	A								12	2												'Cantidad'
23.00	A								12	12												'Producto'
24.00	A								12	23												'Ubicación'
25.00	A								12	35												'Descripción'
26.00	A		LSTPRD	R					14	15												REFFLD(PRODC)
27.00	A		LSTLOC	R					14	26												REFFLD(LOCATN *SRC)
28.00	A		LSTQTY	R					14	5												REFFLD(QTY *SRC)
29.00	A																					EDTCDE(Z)
30.00	A		LSTDSC	R					14	35												REFFLD(DESCRP)

### DDS para objeto de notificación y estructura de datos descrita externamente (PRDRCTP)

SEQNBR	*...	...	1	...	...	2	...	...	3	...	...	4	...	...	5	...	...	6	...	...	7	..
1.00	A																					LIFO
2.00	A																					REF(PRDMSTP)
3.00	A	R	PRDRCTR																			
4.00	A		USER						10													
5.00	A		PRODC	R																		
6.00	A		DESCRP	R																		
7.00	A		QTY						3	0												
8.00	A		LOCATN	R																		REFFLD(LOCATN PRDLOCP)
9.00	A	K	USER																			

El programa procesa el objeto de notificación de la siguiente manera:

- Al principio, el programa procesa el objeto de notificación de forma aleatoria y visualiza un registro si existe para la clave específica:
  - Si existen varios registros, se utiliza el último registro de esta clave ya que el archivo PRDRCTP está en orden LIFO.
  - Si no existe ningún registro, no se ha interrumpido ninguna transacción por lo que no es necesario reiniciar.

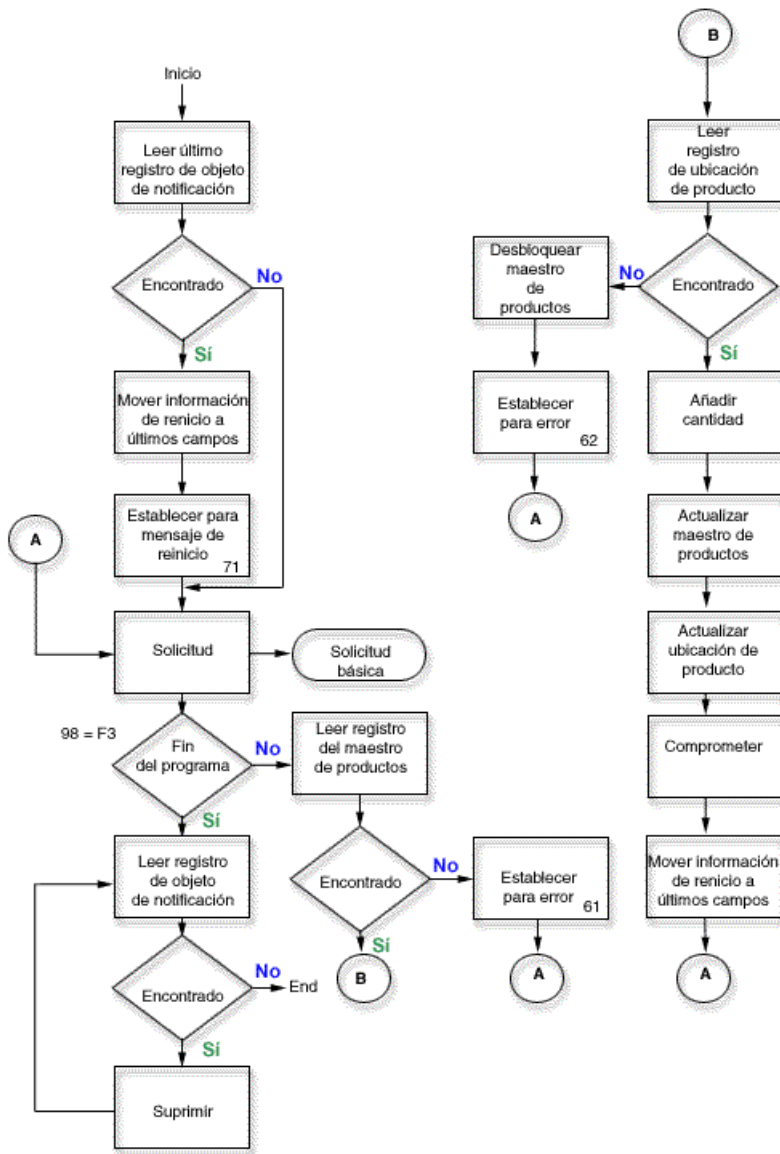
- Si el programa falla antes de la primera operación de compromiso satisfactoria, no considera que sea necesario reiniciar.
- La rutina para borrar el objeto de notificación tiene lugar al final del programa:
  - Si se han producido varias anomalías, la rutina puede manejar la supresión de varios registros en el objeto de notificación.
  - Aunque el sistema coloque la identificación de compromiso en un archivo de base de datos, la identificación de compromiso debe especificarse como una variable en el programa RPG.
  - Como RPG permite describir externamente una estructura de datos, una estructura de datos es un método adecuado para especificar la identificación de compromiso. En este ejemplo, la estructura de datos utiliza la misma descripción externa que el archivo de base de datos empleado como objeto de notificación.

El proceso de este programa solicita al usuario un número de producto, una ubicación y una cantidad:

- Deben actualizarse dos archivos:
  - Archivo maestro de producto (PRDMSTP)
  - Archivo de ubicación de producto (PRDLOCP)
- Debe existir un registro en cada archivo antes de que se actualice alguno de ellos.
- El programa mueve los campos de entrada a los campos últimos correspondientes tras entrarse satisfactoriamente cada transacción. Estos campos últimos se muestran al operador en cada solicitud a modo de información sobre lo que se ha especificado en último lugar.
- Si existe información para reiniciar, esta se mueve a estos campos últimos y en la pantalla aparece un mensaje especial.

Este proceso se muestra en la figura siguiente. El nombre de usuario se pasa al programa para proporcionar un registro exclusivo en el objeto de notificación.

## Flujo de programa



El siguiente ejemplo se refiere al código fuente RPG. El objeto de notificación (archivo PRDRCTP) se utiliza como un archivo normal al principio y al final del programa y también se especifica como objeto de notificación en el CL (mandato STRCMTCTL) antes de llamar al programa.

### Fuente RPG

SEQNBR \*... .. 1 ... .. 2 ... .. 3 ... .. 4 ... .. 5 ... .. 6 ... .. 7 ..

```

1.00   FPRDMSTP UF  E           K       DISK       KCOMIT
2.00   FPRDLOCP UF  E           K       DISK       KCOMIT
3.00   FPRDRCTD CF  E                   WORKSTN
4.00   F*
5.00   F* Archivo siguiente es objeto de notificación específico de este programa.
6.00   F*      Solo se accede a él en una situación de reinicio y al final
7.00   F*      del programa para suprimir los registros. Control de
8.00   F*      compromiso escribe los registros en el objeto de notificación.
9.00   F*
10.00  FPRDRCTP UF  E           K       DISK
11.00  ICMTID   E  DSPRDRCTP

```

```

12.00 C          *ENTRY  PLIST
13.00 C          PARM          USER10 10
14.00 C          MOVE USER10  USER
15.00 C*
16.00 C*  Buscar información de reinicio - obtener último registro por usuario
17.00 C*  Vía de acceso de archivo PRDRCTP está en secuencia LIFO
18.00 C*
19.00 C          USER      CHAINPRDRCTR          20  No encontrado
20.00 C N20          DO          Reiniciar
21.00 C          EXSR MOVLST          Mover a último
22.00 C          SETON          71  Reiniciar
23.00 C          END
24.00 C*
25.00 C*  Bucle básico del proceso
26.00 C*
27.00 C          LOOP      TAG
28.00 C          EXFMTPROMPT
29.00 C 98          GOTO END          Fin del programa
30.00 C          PRODCY      CHAINPRDMSTR          61  No encontrado
31.00 C 61          GOTO LOOP
32.00 C          KEY      KLIST
33.00 C          KFLD          PRODCY
34.00 C          KFLD          LOCATN
35.00 C          KEY      CHAINPRDLOCR          62  No encontrado
36.00 C 62          DO
37.00 C          EXCPTRLSMST          Liberar bloqueo
38.00 C          GOTO LOOP
39.00 C          END
40.00 C          ADD QTY          ONHAND          Añadir
41.00 C          ADD QTY          LOCAMT
42.00 C          UPDATPRDMSTR          Actualizar
43.00 C          UPDATPRDLOCR          Actualizar
44.00 C*
45.00 C*  Comprometer y mover a campos anteriores
46.00 C*
47.00 C          CMTID      COMIT
48.00 C          EXSR MOVLST          Mover a último
49.00 C          GOTO LOOP
50.00 C*
51.00 C*  Proceso de fin del programa
52.00 C*
53.00 C          END      TAG
54.00 C          SETON          LR
55.00 C*56.00 C*  Suprimir los registros del objeto de notificación
57.00 C*
58.00 C          DLTLPL      TAG
59.00 C          USER      CHAINPRDRCTR          20  No encontrado
60.00 C N20          DO
61.00 C          DELETPRDRCTR          Suprimir
62.00 C          GOTO DLTLPL
63.00 C          END
64.00 C*
65.00 C*  Mover a campos de último uso para información del operador
66.00 C*
67.00 C          MOVLST      BEGSR
68.00 C          MOVE PRODCY      LSTPRD
69.00 C          MOVE LOCATN      LSTLOC
70.00 C          MOVE QTY          LSTQTY
71.00 C          MOVE DESCRP      LSTDSC
72.00 C          ENDSR
73.00 C OPRDMSTR E          RLSMST

```

### Conceptos relacionados

“Ejemplo: utilizar un archivo de anotaciones de transacciones para iniciar una aplicación” en la página 88

Este ejemplo proporciona código de ejemplo e instrucciones acerca de cómo utilizar un archivo de anotaciones de transacciones a fin de iniciar una aplicación tras una finalización anómala.

“Ejemplo: un único objeto de notificación para todos los programas”

El uso de un único objeto de notificación para todos los programas es beneficioso dado que toda la información necesaria para reiniciar está en el mismo objeto y puede emplearse un método estándar para el objeto de notificación en todos los programas. En este caso, utilice una combinación exclusiva de identificaciones de usuario y programa para asegurarse de que el programa accede a la información correcta al volver a iniciarse.

### **Ejemplo: un único objeto de notificación para todos los programas**

El uso de un único objeto de notificación para todos los programas es beneficioso dado que toda la información necesaria para reiniciar está en el mismo objeto y puede emplearse un método estándar para el objeto de notificación en todos los programas. En este caso, utilice una combinación exclusiva de identificaciones de usuario y programa para asegurarse de que el programa accede a la información correcta al volver a iniciarse.

Como la información necesaria para reiniciar puede variar de un programa a otro, no emplee una estructura de datos descrita externamente para la identificación de compromiso. Si se emplea un único objeto de notificación, el programa precedente puede describir la estructura de datos en el programa, en lugar de hacerlo externamente. Por ejemplo:

1	10	USER
11	20	PGMNAM
21	23	PRODC
24	29	LOCATN
30	49	DESC
50	51 0	QTY
52	220	DUMMY

En cada programa que utilice ese objeto de notificación, la información especificada para la identificación de compromiso es exclusiva del programa (los nombres de usuario y de programa no son exclusivos). El objeto de notificación debe ser lo suficientemente grande como para contener la máxima información que cualquier programa pueda colocar en la identificación de compromiso.

#### **Conceptos relacionados**

“Ejemplo: un objeto de notificación exclusivo para cada programa” en la página 93

El uso de un solo objeto de notificación exclusivo para cada trabajo permite emplear una identificación de compromiso descrita externamente, aunque podría haber múltiples usuarios del mismo programa.

### **Ejemplo: utilizar un programa de proceso estándar para iniciar una aplicación**

Programa de proceso estándar es una manera de volver a iniciar la aplicación utilizando un archivo de base de datos como objeto de notificación para todas las aplicaciones. En este enfoque se presupone que los nombres de perfil de usuario son exclusivos de cada usuario en todas las aplicaciones que emplean el programa estándar.

**Nota:** Si utiliza el ejemplo de código, indica que acepta los términos que figuran en: “Información de licencia de código y declaración de limitación de responsabilidad” en la página 118.

Para este procedimiento, se utiliza el archivo físico NFYOBJP como objeto de notificación definido como:

Nombre de perfil de usuario exclusivo	10 caracteres
Identificación de programa	10 caracteres
Información para volver a iniciarse	Campo de caracteres (Debe tener el tamaño suficiente para contener la máxima cantidad de información para iniciar de nuevo programas que necesiten información para volver a iniciarse. Este campo es necesario para



los programas de aplicación.  
 En el ejemplo, se presupone  
 que tiene una longitud igual a 200).

El archivo se crea con SHARE(\*YES). Los dos primeros campos del archivo son la clave del archivo. (Este archivo también se puede definir como una estructura de datos en los programas RPG).

### Conceptos relacionados

“Ejemplo: utilizar un archivo de anotaciones de transacciones para iniciar una aplicación” en la página 88

Este ejemplo proporciona código de ejemplo e instrucciones acerca de cómo utilizar un archivo de anotaciones de transacciones a fin de iniciar una aplicación tras una finalización anómala.

### Ejemplo: código para un programa de proceso estándar

Este es un ejemplo de cómo utilizar un programa de proceso estándar.

La aplicación que se muestra en el ejemplo de código siguiente se ejecuta de esta manera:

1. El programa de aplicación recibe el nombre de usuario en un parámetro y lo utiliza con el nombre de programa como identificador exclusivo en el objeto de notificación.
2. El programa de aplicación pasa el código de petición R al programa de proceso de compromiso estándar, que determina si existe un registro en el objeto de notificación.
3. Si el programa de proceso de compromiso estándar devuelve el código 1, se ha encontrado un registro y el programa de aplicación presenta al usuario la información necesaria para volver a empezar.
4. El programa de aplicación lleva a cabo el proceso normal.
5. Cuando se completa una transacción, se salvan los valores para consulta, por lo que el usuario de la estación de trabajo puede ver lo que se ha realizado en la transacción anterior.

El objeto de notificación no proporciona la información salvada ya que el objeto de notificación solo se actualiza si se produce una anomalía del trabajo o del sistema.

**Nota:** Si utiliza los ejemplos de código, indica que acepta los términos que figuran en: “Información de licencia de código y declaración de limitación de responsabilidad” en la página 118.

### Ejemplo de programa de aplicación

```

SEQNBR *... .. 1 ... .. 2 ... .. 3 ... .. 4 ... .. 5 ... .. 6 ... .. 7 ..
1.00      FPRDMSTP UF E          K          DISK          KCOMIT
2.00      FPRDLOCP UF E          K          DISK          KCOMIT
3.00      FPRDRCTD CF E          WORKSTN
4.00      F*
5.00      F* A continuación figura una matriz de tiempo de compilación con
6.00      F* la información de reinicio empleada en el siguiente ejemplo
7.00      F*
8.00      E                      RTXT  50  50  1          Texto de reinicio
9.00      I*
10.00     I* Estructura datos empleada para info. pasada a objeto notificación
11.00     I*
12.00     ICMTID          DS
13.00     I                      1  10 USER
14.00     I                      11  20 PGMNAM
15.00     I                      21  23 PRODC
16.00     I                      24  29 LOCATN
17.00     I                      30  49 DESCRP
18.00     I                      P  50  510QTY
19.00     I                      52  170 DUMMY
20.00     I                      171 220 RSTART
21.00     C          *ENTRY  PLIST
22.00     C          PARM          USER10 10
23.00     C*
24.00     C* Inicializar campos empleados para comunicar con programa estándar
25.00     C*

```

```

26.00 C          MOVE USER10  USER
27.00 C          MOVE 'PRDRC2' PGMNAM
28.00 C          MOVE 'R'      RQSCOD          Petición de lectura
29.00 C          CALL 'STDCMT'
30.00 C          PARM          RQSCOD 1
31.00 C          PARM          RTNCOD 1
32.00 C          PARM          CMTID 220      Estructura de datos
33.00 C          RTNCOD IFEQ '1'            Reiniciar
34.00 C          EXSR MOVLST                Mover a último
35.00 C SETON                               71 Reiniciar
36.00 C          END
37.00 C*
38.00 C* Inicializar campos empleados en objeto de notificación
39.00 C*
40.00 C          MOVEARTXT,1  RSTART          Mover texto
41.00 C*
42.00 C* Bucle de proceso básico
43.00 C*
44.00 C          LOOP      TAG
45.00 C          EXFMT PROMPT
46.00 C 98              GOTO END
47.00 C          PRODC   CHAINPRDMSTR        61 No encontrado
48.00 C 61              GOTO LOOP
49.00 C          KEY      KLIST
50.00 C          KFLD     PRODC
51.00 C          KFLD     LOCATN
SEQNBR *... .. 1 ... .. 2 ... .. 3 ... .. 4 ... .. 5 ... .. 6 ... .. 7 ..

52.00 C          KEY      CHAINPRDLOC   62 No encontrado
53.00 C 62              DO
54.00 C          EXCPTRLSMST            Liberar bloqueo
55.00 C          GOTO LOOP
56.00 C          END
57.00 C          ADD QTY  ONHAND        Añadir
58.00 C          ADD QTY  LOCAMT
59.00 C          UPDATPRDMSTR            Actualizar
60.00 C          UPDATPRDLOC            Actualizar
61.00 C*
62.00 C* Comprometer y mover a campos anteriores
63.00 C*
64.00 C          CMTID   COMIT
65.00 C          EXSR MOVLST            Mover a último
66.00 C          GOTO LOOP
67.00 C* Proceso de fin del programa
68.00 C*
69.00 C          END      TAG
70.00 C          MOVE 'D'  RQSCOD        Petición de supresión
71.00 C          CALL 'STDCMT'
72.00 C          PARM          RQSCOD
73.00 C          PARM          RTNCOD
74.00 C          PARM          CMTID
75.00 C          SETON                               LR
76.00 C*
77.00 C* Mover a campos de último uso para información del operador
78.00 C*
79.00 C          MOVLST  BEGSR
80.00 C          MOVE PRODC  LSTPRD
81.00 C          MOVE LOCATN LSTLOC
82.00 C          MOVE DESCRP LSTDSC
83.00 C          MOVE QTY   LSTQTY
84.00 C          ENDSR
85.00 C          OPRDMSTR E          RLSMST
86.00 ** RTXT          Texto de reinicio
87.00 Menú Inventario - Opción Recibos

```

## Flujo de proceso:

En este tema se muestra el flujo de proceso del programa de proceso.

Las aplicaciones que deben volver a iniciarse llaman al programa estándar. Los programas de aplicación pasan esta lista de parámetros al programa estándar:

- Código de petición
- Código de retorno
- Nombre de la estructura de datos (el contenido del objeto de notificación)

Los códigos de petición realizan las siguientes operaciones:

- R (Lectura)

Recupera el último registro añadido al objeto de notificación con la misma clave. El código de retorno se establece en:

**0** No hay ningún registro disponible (no es necesario volver a iniciar).

**1** Registro devuelto en el campo de información para volver a iniciar (es necesario volver a iniciar).

- WA (Escritura)

Escribe un registro en el archivo. Este código puede utilizarse si se utiliza un objeto de notificación con fines propios. Por ejemplo, si el programa determina que la transacción debe reiniciarse, el programa puede escribir un registro en el objeto de notificación para simular lo que hará el sistema si se produce un error del trabajo o del sistema.

- DE (Supresión)

Suprime todos los registros del objeto de notificación con la misma clave. El código de retorno se establece en:

**0** No existe ningún registro para suprimir.

**1** Se han suprimido uno o varios registros.

- OE (Apertura)

El código de petición O es opcional y se utiliza para evitar tener que iniciar el programa de proceso cada vez que se le llama.

- CA (Cierre)

Una vez utilizado el código de petición de apertura, el uso del código de petición de cierre garantiza el cierre del archivo.

- SA (Búsqueda)

Devuelve el último registro de este usuario. El nombre del programa no se utiliza. Este código se puede emplear en un programa inicial para determinar si es necesario volver a iniciar.

### **Ejemplo: código para un programa de proceso de compromiso estándar**

El programa de proceso de compromiso estándar (STDCMT) lleva a cabo las funciones necesarias para comunicar con un único objeto de notificación empleado por todas las aplicaciones.

Un programa estándar escrito por el usuario debe procesar el objeto de notificación, mientras que la función de control de compromiso escribe automáticamente una entrada en el objeto de notificación. El programa estándar simplifica y estandariza el procedimiento.

El programa se escribe para verificar los parámetros que se pasan y para realizar la acción pertinente, como se indica a continuación:

#### **O=Abrir**

El programa llamador solicita que el archivo de objeto de notificación se mantenga abierto al volver. Como el programa RPG abre implícitamente el objeto de notificación, el programa no debe cerrarlo. El indicador 98 se establece de tal manera que el programa retorna con LR

desactivado para conservar las áreas de trabajo del programa, y deja abierto el objeto de notificación para que se le pueda volver a llamar sin que se produzca una actividad adicional excesiva.

### C=Cerrar

El programa llamador ha determinado que ya no necesita el objeto de notificación y solicita un cierre. El indicador 98 se desactiva para permitir un cierre completo del objeto de notificación.

### R=Leer

El programa llamador solicita que se lea un registro con campos de clave coincidentes y se vuelva a pasar. El programa utiliza los campos de clave pasados para intentar recuperar un registro de NIFYOBJP. Si existen registros duplicados para la misma clave, se devuelve el último registro. El código de retorno se establece en consecuencia y, si existe el registro, se pasa de nuevo en la estructura de datos CMTID.

### W=Escribir

El programa llamador solicita que se escriba un registro en el objeto de notificación para permitir que el programa llamador se vuelva a iniciar la próxima vez que se le llame. El programa escribe el contenido de la información pasada como un registro en NIFYOBJP.

### D=Suprimir

El programa llamador solicita que se supriman los registros de esta clave coincidente. Esta función se utiliza normalmente tras la finalización satisfactoria del programa llamador para eliminar toda información sobre volver a iniciarse. El programa intenta suprimir los registros de los campos de clave pasados. Si no existe ningún registro, se pasa un código de retorno distinto.

### S=Buscar

El programa llamador solicita una búsqueda de un registro de un usuario determinado, con independencia del programa que lo escribió. Esta función se utiliza en el programa de inicio de sesión para indicar que es necesario iniciar de nuevo. El programa solo utiliza el nombre de usuario como clave para ver si existen registros. El código de retorno se establece en consecuencia y el contenido del último registro de esta clave (si existe) se lee y se vuelve a pasar.

El siguiente ejemplo muestra el programa de proceso de compromiso estándar, STDCMT.

## Programa de proceso de compromiso estándar

**Nota:** Si utiliza el ejemplo de código, indica que acepta los términos que figuran en: "Información de licencia de código y declaración de limitación de responsabilidad" en la página 118.

```

SEQNBR *... .. 1 ... .. 2 ... .. 3 ... .. 4 ... .. 5 ... .. 6 ... .. 7 ..

1.00      FNFYOBJP UF  E          K          DISK          A
2.00      ICMTID      DS
3.00      I
4.00      I
5.00      I
6.00      C          *ENTRY  PLIST
7.00      C          PARM      RQSCOD  1
8.00      C          PARM      RTNCOD  1
9.00      C          PARM      CMTID  220
10.00     C          UNQUSR  CABEQ*BLANKS  BADEND      H1 No válido
11.00     C          UNQPGM  CABEQ*BLANKS  BADEND      H2 No válido
12.00     C*
13.00     C*  '0' para Abrir
14.00     C*
15.00     C          RQSCOD  IFEQ '0'          Abrir
16.00     C          SETON          98      Fin de LR
17.00     C          GOTO END
18.00     C          END
19.00     C*
20.00     C*  'C' para Cerrar
21.00     C*

```

```

22.00 C          RQSCOD  IFEQ 'C'          Cerrar
23.00 C          SETOF          98
24.00 C          GOTO END
25.00 C          END
26.00 C*
27.00 C* 'R' para Leer - Obtener el último registro de la clave
28.00 C*
29.00 C          RQSCOD  IFEQ 'R'          Leer
30.00 C          KEY     KLIST
31.00 C          KFLD     UNQUSR
32.00 C          KFLD     UNQPGM
33.00 C          KEY     CHAINNFYOBJR      51  No encontrado
34.00 C  51          MOVE '0'      RTNCOD
35.00 C  51          GOTO END
36.00 C          MOVE '1'      RTNCOD      Encontrado
37.00 C          LOOP1  TAG
38.00 C          KEY     READENFYOBJR      20 EOF
39.00 C  20          GOTO END
40.00 C          GOTO LOOP1
41.00 C          END
42.00 C*
43.00 C* 'W' para Escribir
44.00 C*
45.00 C          RQSCOD  IFEQ 'W'          Escribir
46.00 C          WRITENFYOBJR
47.00 C          GOTO END
48.00 C          END
49.00 C*
50.00 C* 'D' para Suprimir - Suprimir todos los registros de la clave
51.00 C*
52.00 C          RQSCOD  IFEQ 'D'          Suprimir
53.00 C          KEY     CHAINNFYOBJR      51  No encontrado
54.00 C  51          MOVE '0'      RTNCOD
55.00 C  51          GOTO END
56.00 C          MOVE '1'      RTNCOD      Encontrado
57.00 C          LOOP2  TAG
58.00 C          DELETNFYOBJR
59.00 C          KEY     READENFYOBJR      20 EOF
60.00 C  N20        GOTO LOOP2
61.00 C          GOTO END
62.00 C          END
63.00 C*
64.00 C* 'S' para Buscar el último registro de este usuario
65.00 C*      (Omitir la parte de programa de la clave)
66.00 C*
67.00 C          RQSCOD  IFEQ 'S'          Buscar
68.00 C          UNQUSR  SETLLNFYOBJR      20 Si igual
69.00 C  N20        MOVE '0'      RTNCOD
70.00 C  N20        GOTO END
71.00 C          MOVE '1'      RTNCOD      Encontrado
72.00 C          LOOP3  TAG
73.00 C          UNQUSR  READENFYOBJR      20 EOF
74.00 C  N20        GOTO LOOP3
75.00 C          GOTO END
76.00 C          END
77.00 C*
78.00 C* Proceso de código de petición no válido
79.00 C*
80.00 C          SETON          H2  Código de petición incorrecto
81.00 C          GOTO BADEND
82.00 C*
83.00 C* Proceso de fin del programa
84.00 C*
85.00 C          END     TAG

```

```

86.00      C  N98              SETON              LR
87.00      C                   RETRN
88.00      C* Código BADEND se utiliza, retorno de error de ciclo de RPG
89.00      C                   BADEND TAG

```

### Conceptos relacionados

“Ejemplo: utilizar un programa de proceso estándar para decidir si hay que reiniciar la aplicación”  
Este ejemplo proporciona código CL de ejemplo para utilizar un programa de proceso estándar que decida si hay que reiniciar o no una aplicación tras una IPL anómala.

### Ejemplo: utilizar un programa de proceso estándar para decidir si hay que reiniciar la aplicación

Este ejemplo proporciona código CL de ejemplo para utilizar un programa de proceso estándar que decida si hay que reiniciar o no una aplicación tras una IPL anómala.

El programa inicial puede llamar al programa de proceso de compromiso estándar para determinar si es necesario volver a empezar. A continuación el usuario de la estación de trabajo puede decidir si volver a empezar o no.

El programa inicial pasa el código de petición S (búsqueda) al programa estándar, que busca cualquier registro del usuario. Si existe un registro, la información para volver a empezar se pasa al programa inicial y la información se muestra al usuario de la estación de trabajo.

La identificación de compromiso del objeto de notificación contiene información que el programa inicial puede visualizar para identificar qué programa debe reiniciarse. Por ejemplo, los 50 últimos caracteres de la identificación de compromiso pueden estar reservados para incluir esta información. En el programa de aplicación, esta información puede estar en una matriz de tiempo de compilación y trasladarse a la estructura de datos en un paso de inicialización. En Ejemplo: código para un programa de proceso de compromiso estándar se muestra cómo incluir esta información en el programa de aplicación.

El siguiente ejemplo muestra un programa inicial que determina si existe un registro en el objeto de notificación.

### Ejemplo: programa inicial

**Nota:** Si utiliza el ejemplo de código, indica que acepta los términos que figuran en: “Información de licencia de código y declaración de limitación de responsabilidad” en la página 118.

```
SEQNBR *... .. 1 ... .. 2 ... .. 3 ... .. 4 ... .. 5 ... .. 6 ... .. 7
```

```

1.00      PGM
2.00      DCLF      CMTINLD
3.00      DCL      &RQSCOD *CHAR LEN(1) VALUE(S) /* Búsqueda */
4.00      DCL      &RTNCOD *CHAR LEN(1)
5.00      DCL      &CMTID *CHAR LEN(220)
6.00      DCL      &USER *CHAR LEN(10)
7.00      DCL      &INFO *CHAR LEN(50)
8.00      RTVJOBA  USER(&USER)
9.00      CHGVAR   &CMTID (&USER *CAT XX)
10.00     /* El valor XX es necesario para impedir un nombre de programa en blanco */
11.00     CALL     STDCMT PARM(&RQSCOD &RTNCOD &CMTID)
12.00     IF      (&RTNCOD *EQ '1') DO /* REINICIAR PETICIÓN */
13.00     CHGVAR   &INFO %SST(&CMTID 171 50)
14.00     SNDRCVF  RCDfmt(RESTART)
15.00     ENDDO
16.00     /*
17.00     /* Especifique sentencias del programa inicial normales */
18.00     /* o -TFRCTL- para el primer programa de menú */
19.00     /*
20.00     ENDPGM

```

### Conceptos relacionados

“Ejemplo: código para un programa de proceso de compromiso estándar” en la página 101

El programa de proceso de compromiso estándar (STDCMT) lleva a cabo las funciones necesarias para comunicar con un único objeto de notificación empleado por todas las aplicaciones.

---

## Resolución de problemas relacionados con las transacciones y el control de compromiso

Lea esta información cuando necesite resolver problemas relacionados con el control de compromiso.

### Errores de control de compromiso

Al utilizar el control de compromiso, es importante entender qué condiciones pueden ocasionar errores y cuáles no.

En general, los errores se producen cuando se utilizan las funciones de control de compromiso de forma incoherente, como por ejemplo al ejecutar un mandato Finalizar control de compromiso (ENDCMTCTL) cuando todavía hay abiertos archivos que emplean la definición de compromiso.

### Errores durante el proceso de compromiso

Si se produce una anomalía de comunicaciones o del sistema durante una operación de compromiso, podría ser necesario hacer una resincronización para asegurar que los gestores de transacciones mantienen los datos coherentes en todos los sistemas que intervienen en la transacción. El comportamiento de la resincronización y su incidencia en la operación de compromiso dependen de estos factores:

- La opción de compromiso Esperar resultado.
- El estado de la transacción.

Si la anomalía es tan grave que nunca se podrá reparar o si no se puede reparar en un tiempo oportuno, los operadores de los demás sistemas implicados en la transacción deben tomar una decisión heurística. La decisión heurística compromete o retrotrae los cambios efectuados en ese sistema durante la transacción. Si la anomalía se corrige tras esa decisión y la resincronización detecta que la decisión ha ocasionado problemas de integridad de los datos, se enviará el mensaje CPD83D9 o CPD83E9 a la cola de mensajes QSYSOPR.

### Conceptos relacionados

“Definición de compromiso para el compromiso de dos fases: no esperar resultado” en la página 36  
Cuando se produce una anomalía de comunicaciones o del sistema durante una operación de compromiso que exija una resincronización, el valor predeterminado consiste en esperar a que finalice la resincronización antes de que se complete la operación de compromiso.

“Estados de la transacción para el control de compromiso de dos fases” en la página 31

Se establece una definición de compromiso en cada ubicación que forma parte de la red de programas de transacciones. Para cada definición de compromiso, el sistema mantiene un seguimiento del estado de la transacción actual y la transacción anterior.

### Condiciones de error

Si se produce un error, se envía un mensaje de escape que se puede supervisar en un programa.

A continuación se indican algunos de los errores más habituales relacionados con el control de compromiso:

- Se ejecutan mandatos STRCMTCTL consecutivos sin que intervenga ningún mandato ENDCMTCTL.
- Se abren archivos bajo control de compromiso pero no se ha ejecutado ningún mandato STRCMTCTL.

Esta no es una condición de error en el caso de los programas que se ejecutan en un grupo de activación que van a utilizar la definición de compromiso a nivel de trabajo. Únicamente un programa

puede iniciar la definición de compromiso a nivel de trabajo, pero cuando un programa la ha iniciado, cualquier programa que se ejecute en cualquier grupo de activación que no utilice una definición de compromiso a nivel de grupo de activación puede utilizar la definición de compromiso a nivel de trabajo. Los programas que se ejecutan en un grupo de activación que van a utilizar la definición de compromiso a nivel de grupo de activación primero deben iniciar la definición de compromiso a nivel de grupo de activación con el mandato STRCMTCTL.

- Los archivos que se abren para la salida bajo control de compromiso no se registran por diario.
- La primera operación de apertura de un archivo compartido coloca el archivo bajo control de compromiso, pero no así las operaciones de apertura posteriores del mismo archivo compartido.
- La primera operación de apertura de un archivo compartido no coloca el archivo bajo control de compromiso, pero sí las operaciones de apertura posteriores del mismo archivo compartido.
- En una sola transacción se alcanza el límite de bloqueos de registros del trabajo.
- El programa emite una operación de lectura, una operación de compromiso y un cambio para el mismo registro. La operación de lectura debe emitirse de nuevo tras la operación de compromiso ya que la operación de compromiso ha liberado el bloqueo del registro.
- Para una ubicación de una sola fase, los recursos que se colocan bajo control de compromiso no residen en la misma ubicación que los recursos que ya están bajo control de compromiso para la definición de compromiso.
- Hay cambios no comprometidos cuando se emite un mandato ENDCMTCTL.

Esta no es una condición de error en el caso del mandato ENDCMTCTL si todos los archivos están cerrados, la base remota está desconectada y no hay ningún recurso de compromiso de API que todavía esté asociado a la definición de compromiso que debe finalizarse.

- Se ejecuta un mandato de compromiso, retroacción o ENDCMTCTL y no se ha ejecutado ningún mandato STRCMTCTL.

Esta no es una condición de error en el caso de los programas que se ejecutan en un grupo de activación y la definición de compromiso a nivel de trabajo está activa. Únicamente un programa puede iniciar la definición de compromiso a nivel de trabajo, pero cuando un programa la ha iniciado, cualquier programa que se ejecute en cualquier grupo de activación que no utilice una definición de compromiso a nivel de grupo de activación puede utilizar la definición de compromiso a nivel de trabajo. Los programas que se ejecutan en un grupo de activación y van a utilizar la definición de compromiso a nivel de grupo de activación primero deben iniciar la definición de compromiso a nivel de grupo de activación con el mandato STRCMTCTL.

- Se ejecuta un mandato ENDCMTCTL con archivos que todavía están abiertos bajo control de compromiso para la definición de compromiso.
- Un trabajo que lleva a cabo una operación de salvar tiene una o varias definiciones de compromiso que no están en un límite de compromiso.
- Una operación de salvar mientras activo ha finalizado dado que otros trabajos con recursos para comprometer no han alcanzado un límite de compromiso en el plazo de tiempo especificado para el parámetro SAVACTWAIT.
- Un proceso de salvar mientras activo no ha podido continuar debido a la adición de recursos de API para comprometer a más de una definición de compromiso para un solo trabajo.
- Existen más de 1023 definiciones de compromiso para un solo trabajo.
- La conversación con una ubicación remota se pierde debido a una anomalía de recursos. Esto puede ocasionar la retroacción de la transacción.
- Un recurso de una fase abierto para actualización está presente en un nodo que no ha iniciado la operación de compromiso. Debe eliminar el recurso o el nodo que ha iniciado la petición de compromiso.
- Se solicita una operación de compromiso mientras la transacción se encuentra en un estado de retroacción necesaria (RBR). Debe llevarse a cabo una operación de retroacción.
- Un programa de salida de API emite una petición de compromiso o una petición de retroacción.



- Un programa desencadenante emite una petición de compromiso o una petición de retrotracción para la definición de compromiso en la que se ha llamado al programa desencadenante.

El programa desencadenante puede iniciar una definición de compromiso aparte y emitir una petición de compromiso o retrotracción para esa definición.

### **Condiciones en las que no se producen errores**

En este tema se enumeran las situaciones que parecen tener el potencial de provocar un mensaje de error. Sin embargo, el control de compromiso permite estas situaciones en las que no se produce ningún error.

A continuación se indican algunas situaciones de control de compromiso en las que no se producen errores:

- Se ejecuta una operación de compromiso o retrotracción y no hay ningún recurso bajo control de compromiso. Esto le permite incluir operaciones de compromiso o retrotracción en el programa sin plantearse si hay recursos bajo control de compromiso. Asimismo, permite especificar una identificación de compromiso antes de efectuar cambios comprometibles.
- Se ejecuta una operación de compromiso o retrotracción y no hay ningún cambio de recurso no comprometido. Esto le permite incluir operaciones de compromiso y retrotracción en el programa sin plantearse si hay cambios de recursos no comprometidos.
- Se cierra un archivo bajo control de compromiso y existen registros no comprometidos. Esta situación permite llamar a otro programa para llevar a cabo la operación de compromiso o retrotracción. Esto se produce independientemente de si el archivo está compartido o no. Esta función permite a un subprograma efectuar cambios de base de datos que forman parte de una transacción en la que participan varios programas.
- Un trabajo finaliza, de forma normal o anómala, con cambios no comprometidos para una o más definiciones de compromiso. Los cambios de todas las definiciones de compromiso se retrotraen.
- Un grupo de activación finaliza con cambios pendientes para la definición de compromiso a nivel de grupo de activación. Si el grupo de activación finaliza de forma normal y no se encuentran errores al cerrar los archivos abiertos bajo control de compromiso cuyo ámbito se circunscribe al mismo grupo de activación que finaliza, el sistema lleva a cabo un compromiso implícito. De lo contrario, se efectúa una retrotracción implícita.
- Un programa accede de nuevo a un registro modificado que no se ha comprometido. Esto permite a un programa llevar a cabo lo siguiente:
  - Añadir un registro y actualizarlo antes de especificar la operación de compromiso.
  - Actualizar el mismo registro dos veces antes de especificar la operación de compromiso.
  - Añadir un registro y suprimirlo antes de especificar la operación de compromiso.
  - Acceder de nuevo a un registro no comprometido mediante un archivo lógico distinto (bajo control de compromiso).
- Se especifica LCKLVL(\*CHG o \*CS) en el mandato STRCMTCTL y se abre un archivo con una operación de compromiso para solo lectura. En este caso, no se produce ningún bloqueo en la petición. Se trata como si el control de compromiso no estuviera vigente, pero el archivo sí aparece en la opción del menú WRKJOB de los archivos bajo control de compromiso.
- Se emite el mandato STRCMTCTL y no se abre ningún archivo bajo control de compromiso. En esta situación, los cambios a nivel de registro efectuados en los archivos no se realizan bajo control de compromiso.

### **Mensajes de error que hay que supervisar durante el control de compromiso**

Son varios los mensajes de error que pueden devolver las operaciones de compromiso o retrotracción o que se pueden enviar a las anotaciones de trabajo, en función del tipo de mensaje y de cuándo se produjo el error.

Los mensajes de error pueden aparecer durante los procesos siguientes:

- Proceso de compromiso o retrotracción normal

- Proceso de compromiso o retrotracción durante la finalización del proceso de un trabajo
- Proceso de compromiso o retrotracción durante la finalización de un grupo de activación

No podrá supervisar cualquiera de los mensajes siguientes durante la finalización de un grupo de activación o la finalización del proceso de un trabajo. Además, solo podrá supervisar los mensajes CPFxxxx. Los mensajes CPDxxxx siempre se envían como mensajes de diagnóstico, que no es posible supervisar. Los errores que se encuentran al finalizar una definición de compromiso a nivel de grupo de activación durante la finalización de un grupo de activación o al finalizar cualquier definición de compromiso durante la finalización de un trabajo quedan en las anotaciones de trabajo como mensajes de diagnóstico.

Los mensajes de error relacionados con el control de compromiso que puede buscar son los siguientes:

**CPD8351**

Puede haber cambios que no se hayan comprometido.

**CPD8352**

Cambios no comprometidos en la ubicación remota &3.

**CPD8353**

Algunos cambios realizados en la base de datos relacional &1 pueden no haberse comprometido.

**CPD8354**

Algunos cambios realizados en el archivo DDM &1 pueden no haberse comprometido.

**CPD8355**

Algunos cambios realizados en el objeto DDL &1 pueden no haberse comprometido.

**CPD8356**

Podrían haberse comprometido cambios retrotraídos.

**CPD8358**

Algunos cambios realizados en la base de datos relacional &1 pueden no haberse retrotraído.

**CPD8359**

Algunos cambios realizados en el archivo DDM &1 pueden no haberse retrotraído.

**CPD835A**

Algunos cambios realizados en el objeto DDL &3 pueden no haberse retrotraído.

**CPD835C**

Objeto de notificación &1 en &2 no actualizado.

**CPD835D**

El recurso DRDA no permite la retención de cursor SQL.

**CPF835F**

Ha fallado la operación de compromiso o retrotracción.

**CPD8360**

Miembros o archivos, o las dos cosas, ya estaban desasignados.

**CPD8361**

El programa de salida de API &1 ha fallado durante el compromiso.

**CPD8362**

El programa de salida de API &1 ha fallado durante la retrotracción.

**CPD8363**

El programa de salida de API &1 ha finalizado después de &4 minutos durante el proceso de compromiso.

- CPD8364**  
El programa de salida de API &1 ha finalizado después de &4 minutos durante el proceso de retrotracción.
- CPD836F**  
Se ha producido un error de protocolo durante la operación de control de compromiso.
- CPD83D1**  
El recurso de API &4 no puede ser el último agente.
- CPD83D2**  
Recurso incompatible con el control de compromiso.
- CPD83D7**  
La operación de compromiso ha cambiado a retrotracción.
- CPD83D9**  
Se ha producido una condición heurística mixta.
- CPF83DB**  
La operación de compromiso ha supuesto una retrotracción.
- CPD83DC**  
Acción en caso de problemas utilizada para determinar la operación de compromiso o retrotracción; razón &2.
- CPD83DD**  
Conversación finalizada; razón &1.
- CPD83DE**  
La información de retorno no es válida.
- CPD83EC**  
El programa de salida de API &1 ha elegido retrotracción.
- CPD83EF**  
Se ha iniciado la operación de retrotracción para la unidad de trabajo lógica siguiente.
- CPF8350**  
Definición de compromiso no encontrada.
- CPF8355**  
ENDCMTCTL no permitido. Cambios pendientes activos.
- CPF8356**  
El control de compromiso ha finalizado con &1 cambios locales no comprometidos.
- CPF8358**  
Objeto de notificación &1 en &2 no actualizado.
- CPF8359**  
Se han producido errores en la operación de retrotracción.
- CPF835A**  
Se ha cancelado la finalización de la definición de compromiso &1.
- CPF835B**  
Se han producido errores mientras finalizaba el control de compromiso.
- CPF835C**  
El control de compromiso ha finalizado con cambios remotos no comprometidos.
- CPF8363**  
Error en la operación de compromiso.

**CPF8364**

El valor del parámetro de control de compromiso no es válido. Código de razón &3.

**CPF8367**

No puede realizarse una operación de control de compromiso.

**CPF8369**

No puede colocarse un recurso de compromiso de API bajo control de compromiso; código de razón &1.

**CPF83D0**

No se permite la operación de compromiso.

**CPF83D2**

Compromiso realizado == Se ha retornado el estado Resincronización en proceso.

**CPF83D3**

Compromiso realizado == Se ha devuelto el estado Heurística mixta.

**CPF83D4**

No se ha enviado la entrada de diario de unidad de trabajo lógica.

**CPF83E1**

La operación de compromiso ha fallado por una violación de restricción.

**CPF83E2**

Se necesita una operación de retrotracción.

**CPF83E3**

El nivel de anidación solicitado no está activo.

**CPF83E4**

El control de compromiso ha finalizado con recursos no comprometidos.

**CPF83E6**

La operación de control de compromiso se ha completado con la resincronización en proceso.

**CPF83E7**

No se permite el compromiso o la retrotracción de la transacción global X/Open.

**Supervisar errores después de un mandato CALL**

Cuando se llama a un programa que utiliza el control de compromiso, supervise si se producen errores imprevistos y efectúe una operación de retrotracción en caso de que se produzca un error.

Por ejemplo, puede haber registros no comprometidos cuando un programa se encuentra con un error imprevisto, como un error RPG de división por cero.

Según el estado del parámetro de respuesta a mensajes de consulta (INQMSGRPY) de un trabajo, el programa envía un mensaje de consulta o lleva a cabo una acción por omisión. Si la respuesta del operador o la acción predeterminada finaliza el programa, todavía hay registros no comprometidos en espera de una operación de compromiso o retrotracción.

Si se llama a otro programa y este hace que se lleve a cabo una operación de compromiso, se comprometerá la transacción completada parcialmente del programa anterior.

Para evitar que se comprometan transacciones completadas parcialmente, supervise los mensajes de escape tras el mandato CALL. Por ejemplo, si es un programa RPG, utilice el código siguiente:

```
CALL RPGA  
MONMSG MSGID(RPG9001)  
EXEC(ROLLBACK) /*Retrotraer si se cancela el programa*/
```

Si es un programa COBOL:

CALL COBOLA  
 MONMSG MSGID(CBE9001)  
 EXEC(ROLLBACK) /\*Retrotraer si se cancela el programa\*/

## Anomalía del proceso de compromiso o retrotracción normal

Pueden producirse errores en cualquier momento durante el proceso de compromiso o retrotracción.

La tabla siguiente divide el proceso en cuatro situaciones. La columna del medio describe las acciones que lleva a cabo el sistema cuando encuentra errores en cada una de las situaciones. La tercera columna propone las acciones que debe emprender el usuario o la aplicación en respuesta a los mensajes. Estas recomendaciones son coherentes con el modo en que el sistema maneja el proceso del control de compromiso.

Situación	Proceso de compromiso o retrotracción	Acción recomendada
Se produce una anomalía del <b>compromiso</b> de E/S a nivel de registro.	<ul style="list-style-type: none"> <li>• Si el error se produce durante la onda de preparación, la transacción se retrotrae y se envía el mensaje CPF83DB.</li> <li>• Si se produce el error durante la onda comprometida, el proceso de compromiso continúa comprometiendo el máximo número de recursos restantes posible. Al final del proceso de compromiso se envía el mensaje CPF8363.</li> </ul>	Supervise los mensajes y maneje el proceso como quiera.
Durante el compromiso se produce una anomalía del programa de salida de compromiso y retrotracción para el recurso de compromiso de API o a nivel de objeto.	<ul style="list-style-type: none"> <li>• Si el error se produce durante la onda de preparación, la transacción se retrotrae y se envía el mensaje CPF83DB.</li> <li>• Si se produce el error durante la onda comprometida, el proceso continúa comprometiendo o retrotrayendo el máximo número de recursos restantes posible. Se devuelve uno de los mensajes siguientes, según el tipo de recurso de compromiso:               <ul style="list-style-type: none"> <li>– CPD8353</li> <li>– CPD8354</li> <li>– CPD8355</li> <li>– CPD8361</li> </ul> </li> </ul> <p>Al final del proceso de compromiso se envía el mensaje CPF8363.</p>	Supervise los mensajes y maneje el proceso como quiera.
Se produce una anomalía de la <b>retrotracción</b> de E/S a nivel de registro.	<ol style="list-style-type: none"> <li>1. Devuelve CPD8356.</li> <li>2. Intenta seguir con el proceso para retrotraer los recursos de compromiso de API o a nivel de objeto.</li> <li>3. Devuelve CPF8359 al final del proceso.</li> </ol>	Supervise los mensajes y maneje el proceso como quiera.

Situación	Proceso de compromiso o retrotracción	Acción recomendada
Durante la retrotracción se produce una anomalía del programa de salida de compromiso y retrotracción para los recursos de compromiso de API o a nivel de objeto.	<ol style="list-style-type: none"> <li>1. Devuelve uno de los mensajes siguientes, según el tipo de recurso de compromiso: <ul style="list-style-type: none"> <li>• CPD8358</li> <li>• CPD8359</li> <li>• CPD835A</li> <li>• CPD8362</li> </ul> </li> <li>2. Continúa el proceso.</li> <li>3. Devuelve CPF8359 al final del proceso.</li> </ol>	Supervise los mensajes y maneje el proceso como quiera.

## Proceso de compromiso o retrotracción durante la finalización del trabajo

Todas las situaciones descritas en la tabla anterior también son válidas cuando un trabajo finaliza, excepto si se envía uno de los mensajes siguientes:

- CPF8356 si solo se registran recursos locales.
- CPF835C si solo se registran recursos remotos.
- CPF83E4 si se registran recursos locales y remotos.

Además, puede aparecer uno de los dos mensajes específicos de la finalización del trabajo si se ha llamado a un programa de salida de compromiso y retrotracción para un recurso comprometible de API. Si el programa de salida de compromiso y retrotracción no finaliza al cabo de 5 minutos, el programa se cancela; se envía un mensaje de diagnóstico CPD8363 (para el compromiso) o CPD8364 (para la retrotracción), y continúa el resto del proceso de compromiso o retrotracción.

## Proceso de compromiso o retrotracción durante IPL

Todas las situaciones descritas en la tabla anterior también son válidas durante la recuperación de IPL para las definiciones de compromiso, excepto si se envía el mensaje CPF835F, en lugar del mensaje CPF8359 o CPF8363. Los mensajes que se envían para una definición de compromiso concreta pueden aparecer en las anotaciones de uno de los trabajos QDBSRVxx o en las anotaciones QHST. En las anotaciones QHST, el mensaje CPI8356 indica el principio de la recuperación de IPL para una definición de compromiso específica. El mensaje CPC8351 indica el final de la recuperación de IPL para una definición de compromiso específica y los demás mensajes relacionados con la recuperación de esa definición de compromiso se encuentran entre estos dos mensajes.

Puede aparecer uno de los dos mensajes específicos de una definición de compromiso si se ha llamado a un programa de salida de compromiso y retrotracción para un recurso comprometible de API. Si el programa de salida de compromiso y retrotracción no finaliza al cabo de 5 minutos, el programa se cancela; se envía un mensaje de diagnóstico CPD8363 (para el compromiso) o CPD8364 (para la retrotracción), y continúa el resto del proceso de compromiso o retrotracción.

## Detectar puntos muertos

Siga estas instrucciones para localizar condiciones de punto muerto.

Una condición de punto muerto puede producirse cuando un trabajo posee un bloqueo sobre un objeto, el objeto A, y espera obtener un bloqueo sobre otro objeto, el objeto B. Al mismo tiempo, otro trabajo u otra transacción en ese momento posee un bloqueo sobre el objeto B y espera obtener un bloqueo sobre el objeto A.

Lleve a cabo los pasos siguientes para averiguar si se ha producido una condición de punto muerto y, de ser así, corregirla:

1. Localice el trabajo suspendido en la lista de trabajos activos.
2. Fíjese en los objetos que el trabajo espera bloquear.
3. Para todos los objetos que el trabajo espera bloquear, examine la lista de los propietarios de bloqueos (transacciones o trabajos) e intente encontrar un bloqueo conflictivo que se corresponda con el nivel solicitado por el trabajo suspendido.
4. Si una transacción posee un bloqueo conflictivo, visualice los trabajos asociados a esta transacción y vea si uno de ellos está a la espera de establecer un bloqueo.
5. Determine si este trabajo en espera intenta bloquear uno de los objetos bloqueados por el trabajo suspendido inicial. Cuando encuentre el trabajo que intenta bloquear uno de los objetos bloqueados por el trabajo suspendido inicial, podrá identificar los objetos en cuestión como los causantes del problema.
6. Investigue la transacción para determinar la línea de acción adecuada.
  - a. Fíjese en las propiedades de la transacción para averiguar qué aplicación la ha iniciado y, después, fíjese en el código de la aplicación.
  - b. También puede rastrear las acciones de la transacción hasta este punto localizando el ID de ciclo de compromiso en las propiedades de la transacción y, a continuación, buscando en un diario las entradas que contienen este ID. Para ello, puede emplear el mandato Recuperar entrada de diario (RTVJRNE) y especificar el parámetro CMTCYCID.
  - c. Cuando haya obtenido información relevante, puede elegir la opción de forzar una operación de compromiso o de retrotracción.

#### **Tareas relacionadas**

“Minimizar bloqueos” en la página 73

Un método de uso habitual para minimizar los bloqueos de registros consiste en liberar el bloqueo de registros. (Esta técnica no funciona si se ha especificado LCKLVL(\*ALL)).

Determinar el estado de un trabajo

“Visualizar objetos bloqueados para una transacción” en la página 69

“Visualizar trabajos asociados a una transacción” en la página 69

“Visualizar información de control de compromiso” en la página 68

Puede utilizar iSeries Navigator para visualizar información sobre todas las unidades lógicas de trabajo) del sistema. También puede examinar información sobre el trabajo asociado a una transacción (si hay alguno).

“Cuándo forzar operaciones de compromiso y retrotracción y cuándo cancelar la resincronización” en la página 114

En este tema se explica cuándo y cómo hay que forzar una operación de compromiso o retrotracción y cuándo hay que cancelar la resincronización.

#### **Referencia relacionada**

Mandato Recuperar entrada de diario (RTVJRNE)

## **Recuperar transacciones tras una anomalía de comunicaciones**

Puede utilizar las instrucciones de este tema para manejar las transacciones que llevan a cabo trabajo en un sistema remoto después de que haya fallado la comunicación con ese sistema.

Si se produce una anomalía de comunicaciones, normalmente el sistema completa la resincronización con el sistema remoto de forma automática. Sin embargo, si la anomalía es tan grave que nunca volverán a establecerse las comunicaciones con el sistema remoto (por ejemplo, si se corta la línea de comunicaciones), usted tendrá que encargarse de cancelar la resincronización y restaurar las transacciones. Asimismo, las transacciones podrían retener bloqueos que se tienen que liberar.

1. En iSeries Navigator, visualice información de control de compromiso para la transacción con la que se propone trabajar.

2. Localice la transacción en cuestión que intenta llevar a cabo la resincronización con el sistema remoto. El campo **Resincronización en curso** para esa transacción se establece en **Sí**.
3. Localice las transacciones que tenían una conexión con el sistema remoto comprobando el estado de recurso de las transacciones individuales.
4. Después de identificar las transacciones, debe forzar el compromiso o forzar la retrotracción, según sea el estado de la transacción.
5. Puede tomar la decisión de comprometer o retrotraer después de investigar las propiedades de las transacciones.
  - Puede emplear el **ID de unidad de trabajo** para hallar otros elementos de la transacción en otros sistemas.
  - También puede determinar si hay que comprometer o retrotracción a partir del estado de la transacción. Por ejemplo, si una transacción de base de datos realiza un compromiso de dos fases durante una anomalía de comunicaciones y su estado tras la anomalía es "Preparada" o "Pendiente de último agente", puede elegir la opción de forzar el compromiso en la transacción.
6. Tras forzar un compromiso o una retrotracción en las transacciones pendientes, detenga la resincronización en la conexión anómala para las transacciones identificadas.

#### **Tareas relacionadas**

"Visualizar información de control de compromiso" en la página 68

Puede utilizar iSeries Navigator para visualizar información sobre todas las unidades lógicas de trabajo del sistema. También puede examinar información sobre el trabajo asociado a una transacción (si hay alguno).

"Cuándo forzar operaciones de compromiso y retrotracción y cuándo cancelar la resincronización"

En este tema se explica cuándo y cómo hay que forzar una operación de compromiso o retrotracción y cuándo hay que cancelar la resincronización.

## **Cuándo forzar operaciones de compromiso y retrotracción y cuándo cancelar la resincronización**

En este tema se explica cuándo y cómo hay que forzar una operación de compromiso o retrotracción y cuándo hay que cancelar la resincronización.

La decisión de forzar una operación de compromiso o retrotracción o cancelar la resincronización se denomina *decisión heurística*. Una decisión heurística es la acción que se lleva a cabo cuando se fuerza al sistema a comprometer o retrotraer una transacción. Cuando toma una decisión heurística, el estado de la transacción pasa a ser de heurística mixta si la decisión no es coherente con el resultado de las demás ubicaciones de la transacción. Debe determinar la acción que han llevado a cabo todas las demás ubicaciones que han participado en la transacción y resincronizar los registros de la base de datos.

Antes de tomar una decisión heurística, recopile el máximo de información posible sobre la transacción. Visualice los trabajos asociados a la definición de compromiso y efectúe un registro de los diarios y archivos que intervienen. Puede emplear esta información más adelante si necesita visualizar entradas de diario y aplicar o eliminar manualmente cambios registrados por diario.

El mejor lugar que puede examinar para hallar información sobre una transacción es la ubicación del iniciador de esa transacción. Sin embargo, la decisión de comprometer o retrotraer podría ser propiedad de un recurso de API o de un último agente.

Si un recurso de API se ha registrado como un recurso de último agente, la decisión final de si se debe comprometer o retrotraer es propiedad del recurso de API. Debe examinar la información sobre la aplicación y ver cómo utiliza el recurso de API para determinar si se debe comprometer o retrotraer.

Si la transacción tiene seleccionado un último agente, el último agente posee la decisión de comprometer o retrotraer. Examine el estado del último agente para obtener información sobre la transacción.



Si debe tomar decisiones heurísticas o cancelar la resincronización debido a una anomalía del sistema o de comunicaciones que no se puede reparar, puede encontrar todas las transacciones pendientes siguiendo estos pasos:

1. En iSeries Navigator, expanda el sistema con el que desea trabajar.
2. Expanda **Bases de datos** y la base de datos local del sistema.
3. Expanda **Transacciones**.
4. Expanda **Transacciones de base de datos** o **Transacciones globales**.

En estas ventanas de visualización, puede ver la definición de compromiso, el estado de resincronización, el ID de la unidad de trabajo actual y el estado de la unidad de trabajo actual para cada transacción. Busque transacciones con la siguiente información:

- Transacciones con **Estado de unidad lógica de trabajo** establecido en Preparada o Pendiente del último agente.
- Transacciones que muestran el estado **Resincronización en curso** con un valor afirmativo.

Para trabajar con el trabajo que participa en la transacción en este sistema pulse la transacción con el botón derecho del ratón y seleccione **Trabajo**.

Al pulsar la transacción con el botón derecho, también puede seleccionar **Forzar compromiso**, **Forzar retrotracción** o **Cancelar resincronización**.

Antes de tomar una decisión heurística o cancelar la resincronización, es posible que desee comprobar el estado de los trabajos en otros sistemas asociados a la transacción. El hecho de consultar los trabajos en sistemas remotos puede ayudarle a evitar tomar decisiones que generen incoherencias de base de datos entre los sistemas.

1. Pulse con el botón derecho del ratón la transacción con la que desea trabajar.
2. Seleccione **Estado de recurso**.
3. En el diálogo Estado de recurso, seleccione la pestaña **Conversación** para las conexiones SNA; seleccione **Conexión** para las conexiones TCP/IP.

Cada recurso de conversación representa un sistema remoto que participa en la transacción. En los sistemas remotos, puede utilizar iSeries Navigator para ver las transacciones asociadas a la transacción.

La parte base del ID de unidad de trabajo es igual en todos los sistemas. Cuando visualice información de control de compromiso en el sistema remoto, busque la parte base del ID de unidad de trabajo que sea igual en el sistema local.

Por ejemplo, si el ID de unidad de trabajo del sistema local empieza por APPN.RCHASL97.X'112233445566, busque el ID de unidad de trabajo del sistema remoto que también comience por APPN.RCHASL97.X'112233445566.

#### **Conceptos relacionados**

“Soporte de transacciones XA para el control de compromiso” en la página 45

DB2 Universal Database (UDB) para iSeries puede participar en las transacciones globales X/Open.

“Iniciar el control de compromiso” en la página 51

Para iniciar el control de compromiso, utilice el mandato Arrancar control de compromiso (STRCMTCTL).

#### **Tareas relacionadas**

“Detectar puntos muertos” en la página 112

Siga estas instrucciones para localizar condiciones de punto muerto.

“Recuperar transacciones tras una anomalía de comunicaciones” en la página 113

Puede utilizar las instrucciones de este tema para manejar las transacciones que llevan a cabo trabajo en un sistema remoto después de que haya fallado la comunicación con ese sistema.

“Visualizar información de control de compromiso” en la página 68

Puede utilizar iSeries Navigator para visualizar información sobre todas las unidades lógicas de trabajo del sistema. También puede examinar información sobre el trabajo asociado a una transacción (si hay alguno).

## Finalizar una retrotracción de larga duración

En este tema se explica cómo finalizar las retrotracciones de larga duración que consumen un tiempo de procesador valioso, bloquean recursos y ocupan espacio de almacenamiento.

Una operación de retrotracción elimina todos los cambios efectuados en una transacción desde la operación de compromiso o retrotracción anterior. Durante una operación de retrotracción, el sistema también libera los bloqueos relacionados con la transacción. Si el sistema contiene miles de transacciones, puede tardar varias horas en realizar una operación de retrotracción. Estas retrotracciones de larga duración pueden consumir tiempo de procesador crítico, bloquear recursos u ocupar espacio de almacenamiento.

Antes de finalizar una retrotracción de larga duración, es necesario conocer las definiciones de compromiso que se retrotraerán y el estado en el que se encuentran las definiciones de compromiso. El campo Estado de las definiciones de compromiso que se retrotraen se establece en RETROTRACCIÓN EN PROGRESO.

Utilice el mandato Trabajar con definiciones de compromiso (WRKCMTDFN) para conocer el estado de una retrotracción, efectuando estos pasos:

1. Escriba WRKCMTDFN JOB(\*ALL) en la interfaz basada en caracteres.
2. Pulse F11 para visualizar el campo Estado.

Si finaliza una retrotracción de larga duración, los archivos que se modificaron durante las transacciones se dejarán con transacciones parciales. No debe finalizar una retrotracción si los archivos no pueden tener transacciones parciales. Para ver los archivos que se han modificado durante la transacción, elija la opción 5 para visualizar el estado en la lista de WRKCMTDFN. Pulse F6 para visualizar el estado de los recursos y seleccione el nivel de Registro.

Debe tener la autorización especial Todos los objetos (\*ALLOBJ) para finalizar una retrotracción de larga duración. Para finalizar una retrotracción de larga duración, efectúe los siguientes pasos:

1. Escriba WRKCMTDFN JOB(\*ALL) en la interfaz basada en caracteres.
2. Elija la opción 20 (Finalizar retrotracción) en la definición de compromiso que desea finalizar.

Los archivos con transacciones parciales tienen los campos Existen transacciones parciales y Retrotracción finalizada establecidos en \*YES en la salida del mandato Visualizar descripción de archivo (DSPFD). Debe eliminar las transacciones parciales para poder utilizar el archivo. Puede eliminar las transacciones parciales suprimiendo el archivo y restaurándolo a partir de una operación de salvar anterior. Si no ha realizado una operación de salvar anterior, puede utilizar el mandato Cambiar objeto registrado por diario (CHGJRNOBJ) para restablecer el estado de Existen transacciones parciales, y así poder abrir el archivo. Para utilizar CHGJRNOBJ se necesita editar el archivo para llevarlo a un estado coherente. Solo debe utilizar el mandato CHGJRNOBJ si no dispone de una operación de salvar anterior.

## Inhabilitar la capacidad de finalizar una retrotracción de larga duración

Los usuarios con la autorización especial \*ALLOBJ pueden finalizar retrotracciones por omisión. Si desea impedir que los usuarios que tengan la autorización especial \*ALLOBJ puedan finalizar retrotracciones, puede hacerlo creando el área de datos QGPL/QTNNOENDRB.

### Referencia relacionada

Mandato Crear área de datos (CRTDTAARA)

---





## Información relacionada con el control de compromiso

Aquí se proporciona una lista de los manuales del producto y de los libros rojos de IBM (IBM Redbooks), en formato PDF, de los sitios Web y de los temas de Information Center que guardan relación con el control de compromiso. Puede ver o imprimir cualquiera de los PDF.

### Manuales

- Consulte la guía del usuario de COBOL/400  (5939 KB) en el sitio Web de manuales suplementarios de la V5R1.
- Consulte la guía del usuario de RPG/400  (2048 KB) en el sitio Web de manuales suplementarios de la V5R1.

### IBM Redbooks

- Connecting WebSphere to DB2 UDB Server  (5 MB)
- Advanced Functions and Administration on DB2 Universal Database for iSeries  (5529 KB)
- Stored Procedures and Triggers on DB2 Universal Database for iSeries  (5836 KB)
- Striving for Optimal Journal Performance on DB2 Universal Database for iSeries  (3174 KB)

### Sitios Web

- The Open Group ([www.opengroup.org](http://www.opengroup.org)) 

### Más información


- Programación de bases de datos
- Programación SQL
- Interfaces API XA
- Gestión de diarios

### Cómo guardar los archivos PDF

Si desea guardar un archivo PDF en su estación de trabajo para verlo o imprimirlo:

1. Pulse el PDF con el botón derecho del ratón en el navegador (pulse el enlace anterior con el botón derecho del ratón).
2. Pulse la opción que guarda el PDF localmente.
3. Navegue hasta el directorio en el que desea guardar el PDF.
4. Pulse **Guardar**.

### Cómo descargar Adobe Acrobat Reader

1. Para poder ver o imprimir archivos PDF, debe instalar Adobe Reader en su sistema. Puede descargar una copia gratuita desde el sitio Web de Adobe ([www.adobe.com/products/acrobat/readstep.html](http://www.adobe.com/products/acrobat/readstep.html)) .

---

## Información de licencia de código y declaración de limitación de responsabilidad

IBM le otorga una licencia de copyright no exclusiva para utilizar todos los ejemplos de código de programación, a partir de los que puede generar funciones similares adaptadas a sus necesidades específicas.

| SUJETO A LAS GARANTÍAS ESTATUTARIAS QUE NO PUEDAN EXCLUIRSE, IBM, LOS  
| DESARROLLADORES Y LOS SUMINISTRADORES DE PROGRAMAS NO OFRECEN NINGUNA  
| GARANTÍA NI CONDICIÓN, YA SEA IMPLÍCITA O EXPLÍCITA, INCLUIDAS, PERO SIN LIMITARSE A  
| ELLAS, LAS GARANTÍAS O CONDICIONES IMPLÍCITAS DE COMERCIALIZACIÓN, ADECUACIÓN  
| A UN PROPÓSITO DETERMINADO Y NO VULNERACIÓN CON RESPECTO AL PROGRAMA O AL  
| SOPORTE TÉCNICO, SI EXISTE.

| BAJO NINGUNA CIRCUNSTANCIA, IBM, LOS DESARROLLADORES O SUMINISTRADORES DE  
| PROGRAMAS SE HACEN RESPONSABLES DE NINGUNA DE LAS SIGUIENTES SITUACIONES, NI  
| SIQUIERA EN CASO DE HABER SIDO INFORMADOS DE TAL POSIBILIDAD:

- | 1. PÉRDIDA O DAÑO DE LOS DATOS;
- | 2. DAÑOS ESPECIALES, ACCIDENTALES, DIRECTOS O INDIRECTOS, O DAÑOS ECONÓMICOS  
| DERIVADOS;
- | 3. PÉRDIDAS DE BENEFICIOS, COMERCIALES, DE INGRESOS, CLIENTELA O AHORROS  
| ANTICIPADOS.

| ALGUNAS JURISDICCIONES NO PERMITEN LA EXCLUSIÓN O LA LIMITACIÓN DE LOS DAÑOS  
| DIRECTOS, ACCIDENTALES O DERIVADOS, POR LO QUE PARTE DE LAS LIMITACIONES O  
| EXCLUSIONES ANTERIORES, O TODAS ELLAS, PUEDE NO SER PROCEDENTE EN SU CASO.

---

## Apéndice. Avisos

Esta información se ha escrito para productos y servicios ofrecidos en Estados Unidos de América.

Es posible que en otros países IBM no ofrezca los productos, los servicios o las características que se describen en este documento. El representante local de IBM le puede informar acerca de los productos y servicios que actualmente están disponibles en su localidad. Las referencias hechas a productos, programas o servicios de IBM no pretenden afirmar ni dar a entender que únicamente puedan utilizarse dichos productos, programas o servicios de IBM. Puede utilizarse en su lugar cualquier otro producto, programa o servicio funcionalmente equivalente que no vulnere ninguno de los derechos de propiedad intelectual de IBM. No obstante, es responsabilidad del usuario evaluar y verificar el funcionamiento de cualquier producto, programa o servicio que no sea de IBM.

IBM puede tener patentes o solicitudes de patente pendientes de aprobación que cubran los temas descritos en este documento. La posesión de este documento no le otorga ninguna licencia sobre dichas patentes. Puede enviar consultas sobre las licencias, por escrito, a:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
Estados Unidos de América

Para consultas sobre licencias relativas a la información de doble byte (DBCS), póngase en contacto con el departamento de propiedad intelectual de IBM en su país o envíe las consultas, por escrito, a:

IBM World Trade Asia Corporation  
Licensing  
2-31 Roppongi 3-chome, Minato-ku  
Tokyo 106-0032, Japón

**El párrafo siguiente no es de aplicación en el Reino Unido ni en ningún otro país en el que tales disposiciones sean incompatibles con la legislación local:** INTERNATIONAL BUSINESS MACHINES CORPORATION PROPORCIONA ESTA PUBLICACIÓN "TAL CUAL" SIN GARANTÍA DE NINGUNA CLASE, EXPLÍCITA O IMPLÍCITA, INCLUIDAS, PERO SIN LIMITARSE A ELLAS, LAS GARANTÍAS IMPLÍCITAS DE NO VULNERACIÓN Y DE COMERCIALIZACIÓN O IDONEIDAD PARA UN PROPÓSITO DETERMINADO. Algunas legislaciones no contemplan la declaración de limitación de responsabilidad, ni implícitas ni explícitas, en determinadas transacciones, por lo que cabe la posibilidad de que esta declaración no se aplique en su caso.

Esta información puede contener imprecisiones técnicas o errores tipográficos. Periódicamente se efectúan cambios en la información incluida en este documento; estos cambios se incorporarán en nuevas ediciones de la publicación. IBM puede efectuar mejoras y/o cambios en los productos y/o programas descritos en esta publicación en cualquier momento y sin previo aviso.

Las referencias hechas en esta publicación a sitios Web que no son de IBM se proporcionan únicamente por cortesía y de ningún modo deben interpretarse como promoción de dichos sitios Web. Los materiales de estos sitios Web no forman parte de los materiales de IBM para este producto, y el usuario será responsable del uso que se haga de estos sitios Web.

IBM puede utilizar o distribuir la información que usted le suministre del modo que IBM considere conveniente sin incurrir por ello en ninguna obligación para con usted.

Los licenciarios de este programa que deseen obtener información acerca del mismo con el fin de: (i) intercambiar la información entre programas creados independientemente y otros programas (incluido este) y (ii) utilizar mutuamente la información que se ha intercambiado, deben ponerse en contacto con:

IBM Corporation  
Software Interoperability Coordinator, Department YBWA  
3605 Highway 52 N  
Rochester, MN 55901  
Estados Unidos de América

Esta información puede estar disponible, sujeta a los términos y condiciones pertinentes, e incluir en algunos casos el pago de una cantidad.

- | El programa bajo licencia descrito en esta información y todo el material bajo licencia disponible para el mismo, se proporciona bajo los términos del Acuerdo de Cliente IBM, el Acuerdo de Licencia de Programa Internacional IBM, el Acuerdo de Licencia para Código Máquina IBM o cualquier otro acuerdo equivalente entre ambas partes.

Los datos de rendimiento incluidos aquí se determinaron en un entorno controlado. Por lo tanto, los resultados que se obtengan en otros entornos operativos pueden variar significativamente. Algunas mediciones pueden haberse realizado en sistemas que estén en fase de desarrollo y no existe ninguna garantía de que esas mediciones vayan a ser iguales en los sistemas disponibles en el mercado. Además, es posible que algunas mediciones se hayan estimado mediante extrapolación. Los resultados reales pueden variar. Los usuarios de este documento deben verificar los datos aplicables a su entorno específico.

La información concerniente a productos que no son de IBM se ha obtenido de los suministradores de dichos productos, de sus anuncios publicados o de otras fuentes de información pública disponibles. IBM no ha comprobado dichos productos y no puede afirmar la exactitud en cuanto a rendimiento, compatibilidad u otras características relativas a productos no IBM. Las consultas acerca de las prestaciones de los productos que no son de IBM deben dirigirse a los suministradores de tales productos.

Todas las declaraciones relativas a la dirección o intención futura de IBM están sujetas a cambios o anulación sin previo aviso y representan únicamente metas y objetivos.

Esta información contiene ejemplos de datos e informes utilizados en operaciones comerciales diarias. Para ilustrarlas de la forma más completa posible, los ejemplos incluyen nombres de personas, empresas, marcas y productos. Todos estos nombres son ficticios y cualquier parecido con los nombres y direcciones utilizados por una empresa real es mera coincidencia.

#### LICENCIA DE COPYRIGHT:

Esta información contiene programas de aplicación de ejemplo en lenguaje fuente, que muestran técnicas de programación en diversas plataformas operativas. Puede copiar, modificar y distribuir los programas de ejemplo de cualquier forma, sin tener que pagar a IBM, con intención de desarrollar, utilizar, comercializar o distribuir programas de aplicación que estén en conformidad con la interfaz de programación de aplicaciones (API) de la plataforma operativa para la que están escritos los programas de ejemplo. Los ejemplos no se han probado minuciosamente bajo todas las condiciones. Por lo tanto, IBM no puede garantizar ni dar por sentada la fiabilidad, la facilidad de mantenimiento ni el funcionamiento de los programas.

Cada copia o parte de estos programas de ejemplo, así como todo trabajo derivado, debe incluir un aviso de copyright como el siguiente:

© (nombre de su empresa) (año). Algunas partes de este código se derivan de programas de ejemplo de IBM Corp. © Copyright IBM Corp. \_escriba el año o los años\_. Reservados todos los derechos.

Si está viendo esta información en copia software, es posible que las fotografías y las ilustraciones en color no aparezcan.

---

## Información de la interfaz de programación

Esta publicación de Control de compromiso documenta las interfaces de programación cuya finalidad es permitir al cliente escribir programas para obtener los servicios de IBM i5/OS.

---

## Marcas registradas

Los términos siguientes son marcas registradas de International Business Machines Corporation en Estados Unidos y/o en otros países:

C/400  
COBOL/400  
DB2  
DB2 Universal Database  
Distributed Relational Database Architecture  
DRDA  
i5/OS  
IBM  
IBM (logotipo)  
Integrated Language Environment  
iSeries  
Redbooks  
RPG/400  
WebSphere

UNIX es una marca registrada de The Open Group en Estados Unidos y en otros países.

Los demás nombres de compañías, productos o servicios pueden ser marcas registradas o de servicio de terceros.

---

## Términos y condiciones

Los permisos para utilizar estas publicaciones están sujetos a los siguientes términos y condiciones.

**Uso personal:** puede reproducir estas publicaciones para uso personal (no comercial) siempre y cuando incluya una copia de todos los avisos de derechos de autor. No puede distribuir ni visualizar estas publicaciones ni ninguna de sus partes, como tampoco elaborar trabajos que se deriven de ellas, sin el consentimiento explícito de IBM.

**Uso comercial:** puede reproducir, distribuir y visualizar estas publicaciones únicamente dentro de su empresa, siempre y cuando incluya una copia de todos los avisos de derechos de autor. No puede elaborar trabajos que se deriven de estas publicaciones, ni tampoco reproducir, distribuir ni visualizar estas publicaciones ni ninguna de sus partes fuera de su empresa, sin el consentimiento explícito de IBM.

Aparte de la autorización que se concede explícitamente en este permiso, no se otorga ningún otro permiso, licencia ni derecho, ya sea explícito o implícito, sobre las publicaciones, la información, los datos, el software o cualquier otra propiedad intelectual contenida en ellas.

IBM se reserva el derecho de retirar los permisos aquí concedidos siempre que, según el parecer del fabricante, se utilicen las publicaciones en detrimento de sus intereses o cuando, también según el parecer del fabricante, no se sigan debidamente las instrucciones anteriores.

No puede bajar, exportar ni reexportar esta información si no lo hace en plena conformidad con la legislación y normativa vigente, incluidas todas las leyes y normas de exportación de Estados Unidos.

IBM NO PROPORCIONA NINGUNA GARANTÍA SOBRE EL CONTENIDO DE ESTAS PUBLICACIONES. LAS PUBLICACIONES SE PROPORCIONAN "TAL CUAL", SIN GARANTÍA DE NINGUNA CLASE, YA SEA EXPLÍCITA O IMPLÍCITA, INCLUIDAS, PERO SIN LIMITARSE A ELLAS, LAS GARANTÍAS IMPLÍCITAS DE COMERCIALIZACIÓN, NO VULNERACIÓN E IDONEIDAD PARA UN FIN DETERMINADO.







Impreso en España