



System i  
Banco de Dados  
Controle de Confirmação

*Versão 6 Release 1*







System i  
Banco de Dados  
Controle de Confirmação

*Versão 6 Release 1*

**Nota**

Antes de utilizar estas informações e o produto suportado por elas, leia as informações em “Avisos”, na página 121.

Esta edição se aplica à versão 6, release 1, modificação 0 do IBM i5/OS (número do produto 5761-SS1) e a todos os releases e modificações subsequentes até que seja especificado de outra forma em novas edições. Esta versão não é executada em todos os modelos RISC (Reduced Instruction Set Computer) nem é executada nos modelos CISC.

© Copyright International Business Machines Corporation 2003, 2008. Todos os direitos reservados.

# Índice

## Controle de Confirmação . . . . . 1

O Que Há de Novo no V6R1 . . . . .	1
Arquivo PDF para Controle de Confirmação. . . . .	1
Conceitos do Controle de Confirmação . . . . .	2
Como Funciona o Controle de Confirmação . . . . .	2
Como as Operações de Confirmação e Rollback Funcionam . . . . .	3
Operação de Confirmação . . . . .	3
Operação de Rollback . . . . .	4
Definição de Confirmação . . . . .	5
Escopo de uma Definição de Confirmação . . . . .	6
Nomes de Definição de Confirmação . . . . .	9
Exemplo: Jobs e Definições de Confirmação . . . . .	10
Como o Controle de Confirmação Funciona com Objetos . . . . .	12
Tipos de Recursos Confirmáveis . . . . .	13
Recursos Confirmáveis Locais e Remotos . . . . .	15
Intenção de Acesso de um Recurso Confirmável . . . . .	15
Protocolo de Confirmação de um Recurso Confirmável . . . . .	16
Arquivos Registrados em Diário e Controle de Confirmação . . . . .	17
Seqüência das Entradas no Diário sob o Controle de Confirmação . . . . .	17
Identificador do Ciclo de Confirmação . . . . .	20
Bloqueio de Registro . . . . .	21
Controle de Confirmação e Conjuntos de Discos Independentes . . . . .	22
Considerações do Conjunto de Discos Independentes para Definições de Confirmação . . . . .	22
Considerações para Transações XA . . . . .	24
Considerações e Restrições para o Controle de Confirmação . . . . .	25
Controle de Confirmação para Aplicativos Batch . . . . .	27
Controle de Confirmação de Duas Fases. . . . .	27
Funções no Processamento de Confirmação. . . . .	28
Estados da Transação para o Controle de Confirmação de Duas Fases . . . . .	30
Definições de Confirmação para Controle de Confirmação de Duas Fases . . . . .	33
Definição de Confirmação para Confirmação de Duas Fases: Permitir Votar Somente Leitura . . . . .	33
Definição de Confirmação para Confirmação de Duas Fases: Não Aguardar pelo Resultado . . . . .	36
Definição de Confirmação para Confirmação de Duas Fases: Indicar OK para Omitir . . . . .	40
Definição de Confirmação para Confirmação de Duas Fases: Não Selecionar um Último Agente . . . . .	42
Efeito do Voto Confiável no Fluxo de Processamento de Confirmação. . . . .	43

Suporte de Transação XA para o Controle de Confirmação . . . . .	45
Modo do Servidor SQL e Transações com Escopo de Encadeamento para o Controle de Confirmação . . . . .	50
Iniciando o Controle de Confirmação . . . . .	51
Objeto de Notificação de Confirmação . . . . .	52
Nível de Bloqueio de Confirmação . . . . .	54
Encerrando o Controle de Confirmação . . . . .	56
Finalização do Controle de Confirmação Iniciado pelo Sistema . . . . .	58
Controle de Confirmação Durante Finalização do Grupo de Ativação . . . . .	58
Operações Implícitas de Confirmação e Rollback . . . . .	59
Controle de Confirmação Durante Finalização Normal da Etapa de Roteamento . . . . .	63
Controle de Confirmação Durante Finalização Anormal do Sistema ou Job . . . . .	63
Atualizações Feitas no Objeto de Notificação . . . . .	65
Recuperação do Controle de Confirmação Durante Carregamento Inicial do Programa Após Término Anormal . . . . .	67
Gerenciando Transações e Controle de Confirmação . . . . .	68
Exibindo Informações de Controle de Confirmação . . . . .	68
Exibindo Objetos Bloqueados para uma Transação . . . . .	69
Exibindo Jobs Associados a uma Transação . . . . .	70
Exibindo Satus do Recurso de uma Transação . . . . .	70
Exibindo as Propriedades da Transação . . . . .	70
Otimizando o Desempenho para Controle de Confirmação . . . . .	70
Minimizando os Bloqueios . . . . .	73
Gerenciando o Tamanho da Transação . . . . .	75
Confirmação Temporária . . . . .	76
Cenários e Exemplos: Controle de Confirmação . . . . .	77
Cenário: Controle de Confirmação. . . . .	77
Problema Prático para o Controle de Confirmação . . . . .	80
Fluxo Lógico para o Problema Prático . . . . .	86
Etapas Associadas ao Fluxo Lógico para o Programa Prático . . . . .	88
Exemplo: Utilizando um Arquivo de Log de Transação para Iniciar um Aplicativo . . . . .	89
Exemplo: Utilizando um Objeto de Notificação para Iniciar um Aplicativo . . . . .	93
Exemplo: Objeto de Notificação Exclusivo para Cada Programa . . . . .	95
Exemplo: Objeto de Notificação Simples para Todos os Programas . . . . .	99
Exemplo: Utilizar um Programa de Processamento Padrão para Iniciar um Aplicativo . . . . .	100
Exemplo: Código para um Programa de Processamento Padrão . . . . .	100
Fluxo de Processamento . . . . .	102

Exemplo: Código para um Programa de Processamento de Confirmação Padrão . . .	103	Detectando Conflitos . . . . .	114
Exemplo: Utilizar um Programa de Processamento Padrão para Decidir Reiniciar ou Não o Aplicativo . . . . .	105	Recuperando Transações Após Falha em Comunicações . . . . .	115
Resolvendo Problemas em Transações e Controle de Confirmação . . . . .	106	Quando Forçar as Operações de Confirmação e de Rollback e Quando Cancelar a Ressincronização . . . . .	116
Erros do Controle de Confirmação . . . . .	106	Encerrando um Rollback de Longa Execução	117
Condições de Erro. . . . .	107	Informações Relacionadas para o Controle de Confirmação. . . . .	118
Condições Sem Erro . . . . .	108	<b>Apêndice. Avisos. . . . .</b>	<b>121</b>
Mensagens de Erro a Serem Monitoradas durante o Controle de Confirmação . . . . .	109	Informações da Interface de Programação . . . . .	123
Monitorando Erros Após um Comando CALL	111	Marcas Registradas . . . . .	123
Falha do Processamento Normal de Confirmação ou Rollback . . . . .	112	Termos e Condições . . . . .	123

---

## Controle de Confirmação

O controle de confirmação é uma função que garante a integridade dos dados. Ele define e processa um grupo de alterações para recursos, como arquivos ou tabelas de banco de dados, como uma transação.

O controle de confirmação garante que o grupo inteiro de alterações individuais ocorra em todos os sistemas que participam na transação ou garante que nenhuma das alterações ocorra. O IBM DB2 para i5/OS utiliza a função de controle de confirmação para confirmar e fazer rollback das transações de banco de dados que estejam em execução com um nível de isolamento diferente de \*NONE (sem confirmação).

Você pode utilizar o controle de confirmação para projetar um aplicativo de modo que o sistema possa reiniciá-lo se uma tarefa, um grupo de ativação dentro de uma tarefa ou o sistema for finalizado de modo anormal. Com o controle de confirmação, você pode ter certeza de que quando o aplicativo é iniciado novamente, nenhuma atualização parcial encontra-se no banco de dados devido a transações incompletas de uma falha anterior.

**Nota:** Utilizando os exemplos de código, você estará concordando com os termos das “Informações sobre o Código de Licença e Renúncia” na página 119.

---



## O Que Há de Novo no V6R1

Leia sobre informações novas ou significativamente alteradas para a coleta de tópicos do Controle de Confirmação.

- Suporte para transações de controle de confirmação \*SYSBAS quando o espaço de nome atual do job for uma ASP independente (consulte a seção de considerações Set ASP Group no Independent disk pool considerations for commitment definitions para obter informações detalhadas).
- Suporte para operação de quiesce ASP independente (consulte Otimizando o Desempenho para Controle de Confirmação para obter informações detalhadas).

## Como Saber o Que É Novo ou o Que Foi Alterado

Para ajudar a ver onde as alterações técnicas foram feitas, estas informações utilizam:

- A imagem  marca onde começam as informações novas ou alteradas.
- A imagem  marca onde terminam as informações novas ou alteradas.

Nos arquivos PDF, você poderá ver barras de revisão (|) na margem esquerda das informações novas ou alteradas.

Para localizar outras informações sobre as novidades ou alterações neste release, consulte Memorando para Usuários.

---

## Arquivo PDF para Controle de Confirmação

Você pode visualizar e imprimir um arquivo PDF dessas informações.

Para visualizar ou fazer download da versão PDF deste documento, selecione Controle de Confirmação (aproximadamente 1361 KB).

## Salvando Arquivos PDF

Para salvar um PDF em sua estação de trabalho para exibição ou impressão:

1. Clique com o botão direito do mouse sobre o link do PDF no seu navegador.
2. Clique na opção que salva o PDF localmente.
3. Navegue para o diretório no qual deseja salvar o PDF.
4. Clique em **Salvar**.

## Fazendo Download do Adobe Reader

É necessário ter o Adobe Reader instalado em seu sistema para visualizar ou imprimir esses PDFs. É possível fazer download de uma cópia gratuita no Web site da Adobe

([www.adobe.com/products/acrobat/readstep.html](http://www.adobe.com/products/acrobat/readstep.html)) .

### Referências relacionadas

“Informações Relacionadas para o Controle de Confirmação” na página 118

Manuais do Produto, Publicações do IBM Redbooks, Web Sites e outras coletas de tópico do centro de informações contêm informações relacionadas à coleta de tópico de controle de Confirmação. É possível visualizar ou imprimir quaisquer dos arquivos PDF.

---

## Conceitos do Controle de Confirmação

Esses conceitos de controle de confirmação ajudam a entender como funciona o controle de confirmação, como ele se interage com o seu sistema e como se interage com outros sistemas em sua rede.

## Como Funciona o Controle de Confirmação

O controle de confirmação garante que o grupo inteiro de alterações individuais ocorra em todos os sistemas participantes ou que nenhuma das alterações ocorra.

Por exemplo, quando você transfere fundos de uma poupança para uma conta corrente, mais de uma alteração ocorre como um grupo. Para você, esta transferência parece uma única alteração. Porém, mais de uma alteração ocorre no banco de dados, pois a poupança e a conta corrente são atualizadas. Para manter ambas as contas exatas, todas as alterações ou nenhuma delas devem ocorrer na conta corrente e na poupança.

O controle de confirmação permite concluir as seguintes tarefas:

- Assegurar-se de que todas as alterações dentro de uma transação serão concluídas para todos os recursos afetados.
- Assegurar-se de que todas as alterações dentro de uma transação serão removidas se o processamento for interrompido.
- Remover alterações feitas durante uma transação quando o aplicativo determina que uma transação está em erro.

Você também pode projetar um aplicativo para que o controle de confirmação possa reiniciar o aplicativo se um job, um grupo de ativação dentro de um job ou o sistema for finalizado de modo anormal. Com o controle de confirmação, você pode ter certeza de que quando o aplicativo é iniciado novamente, nenhuma atualização parcial encontra-se no banco de dados devido a transações incompletas de uma falha anterior.

## Transação

Uma transação é um grupo de alterações individuais em objetos no sistema que aparece como uma única alteração atômica para o usuário.



**Nota:** O System i Navigator utiliza o termo *transação*, enquanto a interface baseada em caracteres utiliza o termo *LUW (Logical Unit of Work)*. Os dois termos são intercambiáveis. Este tópico, a menos que esteja referindo-se especificamente à interface baseada em caracteres, utiliza o termo *transação*.

Uma transação pode ser qualquer uma das seguintes situações:

- Indagações nas quais não ocorre nenhuma alteração do arquivo do banco de dados.
- Transações simples que alteram um arquivo do banco de dados.
- Transações complexas que alteram um ou mais arquivos do banco de dados.
- Transações complexas que alteram um ou mais arquivos do banco de dados, mas estas alterações representam apenas uma parte de um grupo lógico de transações.
- Transações simples ou complexas que envolvem arquivos do banco de dados em mais de uma localização. Os arquivos do banco de dados podem estar em uma das seguintes situações:
  - Em um único sistema remoto.
  - No sistema local e em um ou mais sistemas remotos.
  - Atribuídos a mais de um diário no sistema local. Cada diário pode ser considerado como uma *localização local*.
- Transações simples ou complexas no sistema local que envolvem objetos que não sejam arquivos do banco de dados.

## Como as Operações de Confirmação e Rollback Funcionam

As operações de confirmação e rollback afetam as alterações feitas sob o controle de confirmação.

As linguagens de programação e as APIs (Interfaces de Programação de Aplicativo) a seguir suportam operações de confirmação e rollback.

Linguagem ou API	Confirmação	Rollback
CL	Comando COMMIT	Comando ROLLBACK
RPG IBM ILE (Integrated Language Environment)	Código de operação COMMIT	Código de operação ROLBK
ILE COBOL	Verbo COMMIT	Verbo ROLLBACK
ILE C	Função _Rcommit	Função _Rrollback
PLI	Sub-rotina PLICOMMIT	Sub-rotina PLIROLLBACK
SQL	Instrução COMMIT	Instrução ROLLBACK
Call Level Interface (CLI) do SQL	Função SQLTransact() (É utilizado para confirmar e efetuar rollback de uma transação)	
APIs de XA	APIs xa_commit() e db2xa_commit()	APIs xa_rollback() e db2xa_rollback()

### Conceitos relacionados

interface de nível de chamada SQL

Programação de Banco de Dados

### Informações relacionadas



PDF WebSphere Development Studio: ILE C/C++ Programmer's Guide

Programação de CL

Interfaces de Programação de Aplicativo

## Operação de Confirmação

Uma operação confirmar torna permanente todas as alterações feitas sob o controle de confirmação desde a operação anterior de confirmação ou rollback. O sistema também libera todos os bloqueios relacionados à transação.

O sistema executa as seguintes etapas quando recebe um pedido para confirmar.

- O sistema salva a identificação de confirmação, se esta for fornecida, para usar no tempo de recuperação.
- O sistema grava registros no arquivo antes de executar a operação de confirmação, se as duas condições a seguir forem verdadeiras:
  - Os registros foram incluídos em um arquivo de banco de dados local ou remoto sob controle de confirmação.
  - SEQONLY(\*YES) foi especificado quando o arquivo foi aberto de modo que o feedback de E/S bloqueado é utilizado pelo sistema e um bloqueio parcial dos registros existe.

Caso contrário, a área de feedback da E/S e os buffers de E/S não são alterados.

- O sistema faz uma chamada ao programa de saída de confirmação e rollback para cada recurso de confirmação da API que estiver presente na definição de confirmação. Se a localização possui mais de um programa de saída registrado, o sistema chama programas de saída para aquela localização na ordem em que foram registrados.
- Se quaisquer alterações de registro foram feitas em recursos atribuídos a um diário, o sistema grava uma entrada de diário C CM para cada diário local associado à definição de confirmação. A seqüência de entradas do diário sob o controle de confirmação mostra as entradas que são geralmente gravadas enquanto uma definição de confirmação está ativa.
- O sistema torna permanentes as alterações de nível de objeto que estão pendentes.
- O sistema desbloqueia bloqueios de registro e de objetos que foram adquiridos e mantidos para fins de controle de confirmação. Estes recursos são colocados à disposição de outros usuários.
- O sistema altera as informações na definição de confirmação para mostrar que a transação atual foi finalizada.

O sistema deve executar todas as etapas anteriores corretamente para que a operação de confirmação obtenha sucesso.

#### **Conceitos relacionados**

“Definição de Confirmação” na página 5

A definição de confirmação contém informações que pertencem aos recursos sendo alterados sob controle de confirmação durante uma transação.

“Seqüência das Entradas no Diário sob o Controle de Confirmação” na página 17

Esta tabela mostra a seqüência de entradas que são geralmente gravadas enquanto uma definição de confirmação está ativa. Você pode utilizar o localizador de informações de entrada do Diário para obter informações adicionais sobre o conteúdo das entradas do diário.

## **Operação de Rollback**

Uma operação de rollback remove todas as alterações feitas desde a última operação de confirmação ou rollback. O sistema também libera todos os bloqueios relacionados à transação.

O sistema executa as seguintes etapas quando recebe um pedido de rollback:

- O sistema limpará os registros do buffer de E/S se ambas as condições a seguir forem verdadeiras:
  - Se os registros foram incluídos em um arquivo de banco de dados local ou remoto sob o controle de confirmação.
  - Se SEQONLY(\*YES) foi especificado quando o arquivo foi aberto para que a E/S bloqueada seja usada pelo sistema e exista um bloqueio de registros parcial que ainda não tenha sido gravado no banco de dados.

Caso contrário, a área de feedback e os buffers de E/S permanecerão inalterados.

- O sistema faz uma chamada para o programa de saída de confirmação ou rollback para cada recurso de confirmação da API que estiver presente na definição de confirmação. Se uma localização tiver mais de um programa de saída registrado, o sistema os chamará para essa localização em ordem reversa da qual foram registrados.

- Se um registro foi excluído de um arquivo, o sistema o incluirá novamente.
- O sistema remove todas as alterações feitas nos registros durante esta transação e coloca os registros originais (as imagens "antes") de volta para o arquivo.
- Se algum registro foi incluído no arquivo durante esta transação, ele permanecerá no arquivo como registro excluído.
- Se alguma alteração de registro foi feita nos recursos designados a um diário durante a transação, o sistema incluirá uma entrada C RB no diário, indicando a ocorrência de uma operação de rollback. O diário também contém imagens de alterações de registro que foram revertidas. Antes da operação de rollback ser solicitada, as imagens "antes" e "depois" dos registros alterados foram colocadas no diário. O sistema também gravará a entrada C RB no diário padrão se algum recurso confirmável foi atribuído a ele.
- O sistema coloca os arquivos abertos sob o controle de confirmação em uma das posições a seguir:
  - O último registro acessado na transação anterior
  - Na posição aberta se nenhuma operação confirmar foi executada para o arquivo utilizando esta definição de confirmação

Esta consideração será importante caso você esteja fazendo o processamento seqüencial.

- O sistema não reverte alterações que não podem ser confirmadas para arquivos de banco de dados. Por exemplo, arquivos abertos não são fechados e arquivos limpos não são restaurados. O sistema não reabre ou reposiciona nenhum arquivo fechado durante essa transação.
- O sistema desbloqueia registros bloqueados adquiridos para fins de controle de confirmação e torna esses registros disponíveis para outros usuários.
- A identificação de confirmação atualmente salva pelo sistema permanece a mesma que a fornecida com a última operação de confirmação para a mesma definição de confirmação.
- O sistema inverte ou reverte as alterações confirmáveis de nível do objeto feitas durante esta transação.
- Os bloqueios dos objetos adquiridos para fins de controle de confirmação são desbloqueados e esses objetos se tornam disponíveis para outros usuários.
- O sistema estabelece o tempo limite de confirmação anterior como sendo o atual.
- O sistema altera as informações na definição de confirmação para mostrar que a transação atual foi finalizada.

O sistema deve executar todas as etapas anteriores corretamente para que a operação de rollback seja bem-sucedida.

## Definição de Confirmação

A definição de confirmação contém informações que pertencem aos recursos sendo alterados sob controle de confirmação durante uma transação.

Para criar uma definição de confirmação, utilize o comando Start Commitment Control (STRCMTCTL) para iniciar o controle de confirmação em seu sistema. Além disso, o DB2 para i5/OS cria automaticamente uma definição de confirmação quando o nível de isolamento é diferente de \*NONE (sem confirmação).

O sistema mantém as informações de controle de confirmação na definição de confirmação a medida que são alterados os recursos de confirmação, até que seja finalizada a definição de confirmação. Cada transação ativa no sistema é representada por uma definição de confirmação. Uma transação subsequente pode reutilizar uma definição de confirmação após cada confirmação ou rollback de uma transação ativa.

Uma definição de confirmação geralmente inclui as seguintes informações:

- Os parâmetros no comando STRCMTCTL.
- O status atual da definição de confirmação.

- Informações sobre arquivos de banco de dados e outros recursos confirmáveis que contém alterações que são feitas durante a transação atual.

Para definições de confirmação com bloqueios com escopo de job, apenas o job que inicia o controle de confirmação conhece aquela definição de confirmação. Nenhum outro job conhece aquela definição de confirmação.

Os programas podem iniciar e usar várias definições de confirmação. Cada definição de confirmação para um job identifica uma transação separada que possui recursos confirmáveis associados a ela. Estas transações podem ser confirmadas ou revertidas independentemente das transação que estão associadas a outras definições de confirmação iniciadas para o job.

#### **Conceitos relacionados**

“Operação de Confirmação” na página 3

Uma operação confirmar torna permanente todas as alterações feitas sob o controle de confirmação desde a operação anterior de confirmação ou rollback. O sistema também libera todos os bloqueios relacionados à transação.

“Controle de Confirmação e Conjuntos de Discos Independentes” na página 22

Os conjuntos de discos independentes e os grupos de conjuntos de discos independentes podem cada um ter um banco de dados i5/OS separado. Você pode usar o controle de confirmação com estes bancos de dados.

“Considerações do Conjunto de Discos Independentes para Definições de Confirmação” na página 22

Esteja ciente dessas considerações para as definições de confirmação ao utilizar conjuntos de discos independentes.

## **Escopo de uma Definição de Confirmação**

O *escopo* de uma definição de confirmação determina quais programas usam essa definição de confirmação e como os bloqueios adquiridos durante as transações são examinados.

A interface que inicia a definição de confirmação determina o escopo desta. Existem quatro escopos possíveis para uma definição de confirmação, que se classificam em duas categorias gerais:

### **Definições de Confirmação com Bloqueios de Job com Escopo Definido**

- Definição de confirmação de nível do grupo de ativação
- Definição de Confirmação do Nível do Job
- Definição de Confirmação Explícitamente Nomeada

### **Definições de Confirmação com Bloqueios de Transação com Escopo Definido**

- Definição de Confirmação de Transação com Escopo Definido

As definições de confirmação com bloqueios de escopo de job podem ser usadas somente por programas executados no job que as iniciou. Em comparação, mais de um job pode utilizar definições de confirmação com bloqueios de escopo de transação.

Os aplicativos geralmente utilizam o nível de grupo de ativação ou as definições de confirmação do nível do job. Essas definições são criadas explicitamente com o comando Start Commitment Control (STRCMTCTL) ou implicitamente pelo sistema quando um aplicativo SQL é executado com um nível de isolamento diferente de \*NONE.

## **Definição de confirmação de nível do grupo de ativação**

O escopo mais comum é o para o grupo de ativação. A definição de confirmação de nível do grupo de ativação é o escopo padrão quando o comando STRCMTCTL explicitamente inicia a definição de confirmação ou quando um aplicativo SQL que executa com um nível de isolamento diferente de \*NONE (sem confirmação) inicia implicitamente a definição de confirmação. Somente programas executados

dentro desse grupo de ativação utilizam a definição de confirmação. Diversas definições de confirmação do nível do grupo de ativação podem estar ativas para um job de uma vez. Porém, cada uma pode estar associada somente a um único grupo de ativação. Os programas executados dentro do grupo de ativação podem associar as alterações confirmáveis somente com essa definição de confirmação de nível de grupo de ativação.

Quando o System i Navigator, o comando Work with Commitment Definitions (WRKCMTDFN), o comando Display Job (DSPJOB) ou o comando Work with Job (WRKJOB) exibe uma definição de confirmação de nível de grupo de ativação, esses campos exibem as seguintes informações:

- O campo da definição de confirmação exibe o nome do grupo de ativação. Ele mostra o valor especial \*DFACTGRP para indicar o grupo de ativação padrão.
- O campo do grupo de ativação exibe o número do grupo de ativação.
- O campo do job exibe o job que iniciou a definição de confirmação.
- O campo do encadeamento exibe \*NONE.

### **Definição de Confirmação do Nível do Job**

Uma definição de confirmação pode ser examinada no job emitindo somente STRCMTCTL CMTSCOPE(\*JOB). Qualquer programa executando em um grupo de ativação que não tem a definição de confirmação de nível do grupo de ativação iniciada utiliza a definição de confirmação de nível do job, se já foi iniciada por outro programa para o job. Você pode iniciar somente uma única definição de confirmação do nível do job para um job.

Quando o System i Navigator, o comando Work with Commitment Definitions (WRKCMTDFN), o comando Display Job (DSPJOB) ou comandos Work with Job (WRKJOB) exibe uma definição de confirmação de nível de job, esses campos exibem as seguintes informações:

- O campo da definição de confirmação exibe o valor especial \*JOB.
- O campo do grupo de ativação exibe um espaço em branco.
- O campo do job exibe o job que iniciou a definição de confirmação.
- O campo do encadeamento exibe \*NONE.

Para um certo grupo de ativação, os programas executados dentro dele podem utilizar somente uma única definição de confirmação. Portanto, programas executados dentro de um grupo de ativação podem utilizar a definição de confirmação de nível de tarefa ou de grupo de ativação, mas não ambas ao mesmo tempo. Em um job de multitarefas que não utiliza o modo do servidor SQL, o trabalho transacional de um programa será examinado com a definição de confirmação apropriada, com respeito ao grupo de ativação do programa, independente de qual encadeamento a executou. Se vários encadeamentos utilizarem o mesmo grupo de ativação, deverão cooperar para executar o trabalho transacional e garantir que as confirmações e rollbacks ocorram na hora certa.

Mesmo quando a definição de confirmação de nível do job está ativa para o job, um programa ainda pode iniciar a definição de confirmação de nível do grupo de ativação, se nenhum programa executando dentro desse grupo de ativação executou algum pedido ou operação de controle de confirmação para a definição de confirmação de nível do job. Caso contrário, você deverá primeiro encerrá-la antes de iniciar a definição de confirmação de nível do grupo de ativação. Os seguintes pedidos ou operações de controle de confirmação para a definição de confirmação de nível de job pode impedir que a definição de confirmação de nível de grupo de ativação seja iniciada:

- Abertura (completa ou compartilhada) de um arquivo de banco de dados sob o controle de confirmação.
- Uso da API Add Commitment Resource (QTNADDCR) para incluir um recurso de confirmação da API.
- Confirmação de uma transação.
- Rollback de uma transação.

- Inclusão um recurso remoto sob o controle de confirmação.
- Uso da API Change Commitment Options (QTNCHGCO) para alterar as opções de confirmação.
- Condução da definição de confirmação para um estado de pedido de rollback utilizando a API Rollback Required (QTNRBRQD).
- Envio de uma entrada no diário que inclua o identificador do ciclo de confirmação atual utilizando a API Send Journal Entr (QJOSJRNE) com o parâmetro Incluir Identificador do Ciclo de Confirmação.

Igualmente, se os programas dentro de um grupo de ativação estão utilizando atualmente a definição de confirmação do nível do grupo de ativação, a definição deve ser finalizada primeiro, antes que os programas executados dentro desse mesmo grupo de ativação possam usar a definição de confirmação de nível de job.

Ao abrir um arquivo de banco de dados, o escopo aberto para o arquivo aberto pode ser para o grupo de ativação ou job com uma restrição: se um programa estiver abrindo um arquivo sob o controle de confirmação e o arquivo foi examinado no job, o programa que fizer o pedido de abertura deverá usar a definição de confirmação de nível do job.

### **Definição de Confirmação Explicitamente Nomeada**

As definições de confirmação nomeadas explicitamente são iniciadas pelo sistema quando precisa executar suas próprias transações de controle de confirmação sem afetar as transações usadas por um aplicativo. Um exemplo de uma função que inicia esses tipos de definições de confirmação é o log de problemas. Um aplicativo não pode iniciar definições de confirmação explicitamente nomeadas.

Quando o System i Navigator, o comando Work with Commitment Definitions (WRKCMTDFN), o comando Display Job (DSPJOB) ou o comando Work with Job (WRKJOB) exibe uma definição de confirmação explicitamente nomeada, esses campos exibem as seguintes informações:

- O campo da definição de confirmação exibe o nome dado a ele pelo sistema.
- O campo do grupo de ativação exibe um espaço em branco.
- O campo do job exibe o job que iniciou a definição de confirmação.
- O campo do encadeamento exibe \*NONE.

### **Definições de Confirmação de Transação Delimitada**

As definições de confirmação de transação delimitada são iniciadas com as APIs de XA para Bloqueios de Transação Delimitada.

Essas APIs utilizam os protocolos de controle de confirmação que têm base no encadeamento ou na conexão SQL e não no grupo de ativação. Ou seja, as APIs são utilizadas para associar a definição de confirmação com uma determinada conexão do encadeamento ou SQL enquanto o trabalho transacional é executado e para confirmar ou reverter as transações. O sistema anexa essas definições de confirmação aos encadeamentos que executam o trabalho transacional, de acordo com os protocolos da API. Elas podem ser usadas pelos encadeamentos em jobs diferentes.

Quando o System i Navigator, o comando Work with Commitment Definitions (WRKCMTDFN), o comando Display Job (DSPJOB) ou o comando Work with Job (WRKJOB) exibe uma definição de confirmação de escopo de transação, esses campos exibem as seguintes informações:

- O campo da definição de confirmação exibe o valor especial \*TNSOBJ.
- O campo do grupo de ativação exibe um espaço em branco.
- O campo do job exibe o job que iniciou a definição de confirmação. Ou, se a definição de confirmação estiver atualmente conectada a um encadeamento, a tarefa do encadeamento será exibida.
- O campo do encadeamento exibe o encadeamento na qual a definição de confirmação está conectada (ou \*NONE se a definição de confirmação não está conectada atualmente a nenhum encadeamento).

## Referências relacionadas

APIs de XA

## Nomes de Definição de Confirmação

O sistema atribui nomes a todas as definições de confirmação iniciadas para um job.

A tabela a seguir mostra várias definições de confirmação e seus nomes associados para um job determinado.

Grupo de Ativação	Escopo de Confirmação	Nome da Definição de Confirmação
Qualquer	Job	*JOB
Grupo de ativação padrão	Grupo de Ativação	*DFACTGRP
Grupo de ativação nomeado pelo usuário	Grupo de Ativação	Nome do grupo de ativação (por exemplo, PAYROLL)
Grupo de ativação nomeado pelo sistema	Grupo de Ativação	Número do Grupo de Ativação (por exemplo, 000000145)
Nenhuma	Nomeado explicitamente	QDIR001 (exemplo de uma definição de confirmação definida pelo sistema para uso do sistema apenas). Nomes de definição de confirmação definidos pelo sistema começam com Q.
Nenhuma	Transação	*TNSOBJ

Apenas os programas compilados IBM ILE (Integrated Language Environment) podem iniciar o controle de confirmação para grupos de ativação diferentes do grupo de ativação padrão. Portanto, um job pode usar várias definições de confirmação somente se o job estiver sendo executado em um ou mais programas compilados ILE.

Os programas OPM (Original Program Model) são executados no grupo de ativação padrão e, por padrão, utilizam a definição de confirmação \*DFACTGRP. Em um ambiente misto OPM e ILE, os jobs deverão utilizar a definição de confirmação de nível do job, se todas as alterações confirmáveis efetuadas por todos os programas tiverem de ser confirmadas ou revertidas juntas.

Um arquivo de banco de dados aberto com escopo para um grupo de ativação pode ser associado a uma definição de confirmação de nível de grupo de ativação ou de nível de job. Um arquivo de banco de dados aberto com escopo para o job pode ser associado apenas à definição de confirmação de nível de job. Portanto, qualquer programa, OPM ou ILE, que abrir um arquivo de banco de dados sob o controle de confirmação com escopo definido para a tarefa, precisará utilizar a definição de confirmação de nível de tarefa.

Os programas aplicativos não utilizam o nome da definição de confirmação para identificar uma determinada definição de confirmação ao fazer um pedido de controle de confirmação. Os nomes de definição de confirmação são essencialmente utilizados em mensagens para identificar uma determinada definição de confirmação para um job.

Para definições de confirmação de nível de grupo de ativação, o sistema determina a definição de confirmação a ser utilizada com base no grupo de ativação em que o programa solicitante está sendo executado. Isto é possível porque os programas que são executados dentro de um grupo de ativação, em qualquer momento, só podem usar uma única definição de confirmação.

Para transações com bloqueios com escopo de transação, as APIs XA e os atributos relacionados à transação adicionados ao CLI determinam qual definição de confirmação o encadeamento de chamada utiliza.

## Informações relacionadas



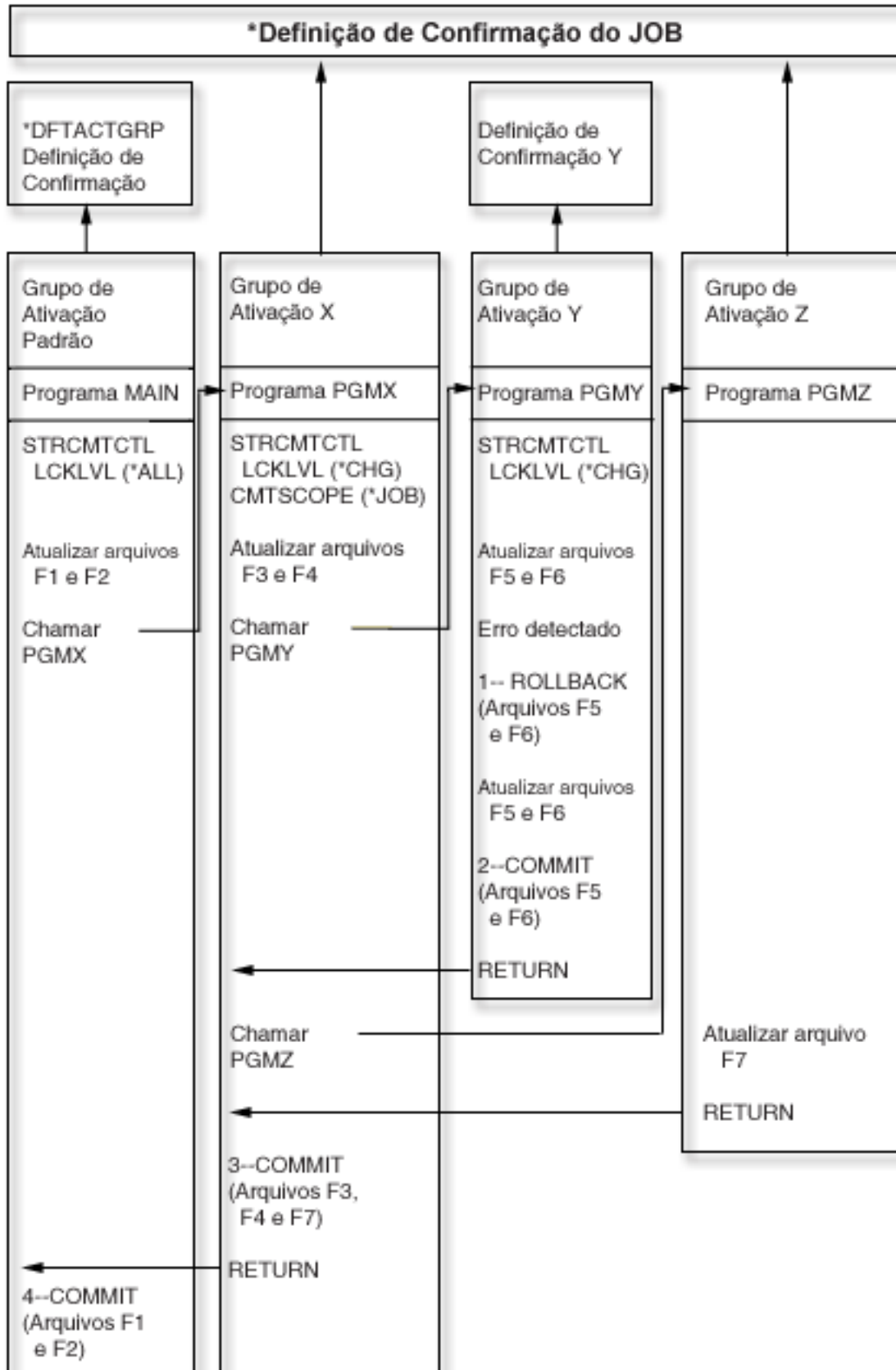
PDF ILE Concepts

### **Exemplo: Jobs e Definições de Confirmação**

Esta figura a seguir mostra um exemplo de uma tarefa que utiliza várias definições de confirmação.

A figura indica quais atualizações do arquivo são confirmadas ou revertidas em cada nível do grupo de ativação. O exemplo supõe que todas as atualizações, que são feitas nos arquivos de banco de dados por todos os programas, são feitas sob controle de confirmação.





A tabela a seguir mostra como os arquivos serão confirmados ou terão o rollback efetuado se o cenário na figura anterior for alterado.

### Exemplos Adicionais de Várias Definições de Confirmação em um Job

Mudança de situação	Efeito em alterações nestes arquivos			
	F1 e F2	F3 e F4	F5 e F6	F7
PGMX executa uma operação de rollback ao invés de uma operação de confirmação (3= =COMMIT torna-se ROLLBACK).	Ainda pendente	Revertido	Já confirmado	Revertido
PGMZ executa uma operação de confirmação antes de retornar ao PGMX.	Ainda pendente	Confirmado por PGMZ	Já confirmado	Confirmado
O PGMZ tenta iniciar o controle de confirmação especificando CMTSCOPE(*ACTGRP) após a atualização do arquivo F7. A tentativa falha porque as alterações estão pendentes utilizando a definição de confirmação de nível de job.	Ainda pendente	Ainda pendente	Já confirmado	Ainda pendente
PGMX não inicia o controle de confirmação e não abre os arquivos F3 e F4 com COMMIT(*YES). PGMZ tenta abrir o arquivo F7 com COMMIT(*YES).	Ainda pendente	Não está sob controle de confirmação	Já confirmado	O arquivo F7 não pode ser aberto porque não existe nenhuma definição de confirmação *JOB (o PGMX não a criou).

### Como o Controle de Confirmação Funciona com Objetos

Quando você coloca um objeto sob o controle de confirmação, ele se torna um recurso confirmável. Ele é registrado com a definição de confirmação. Ele participa de cada operação de confirmação e rollback que ocorre com essa definição de confirmação.

Os tópicos a seguir descrevem os atributos de um recurso confirmável:

- Tipo de recurso
- Localização
- Protocolo de confirmação
- Intenção de acesso

## Tipos de Recursos Confirmáveis

Esta tabela lista os diferentes tipos de recursos confirmáveis, incluindo FILE, DDL (Data Definition Language), DDM (Distributed Data Management), LU (Logical Unit) 6.2, DRDA (Distributed Relational Database Architecture), API e TCP.

A tabela mostra os seguintes itens:

- Os tipos de recursos confirmáveis.
- Como são colocados sob o controle de confirmação.
- Como são removidos do controle de confirmação.
- Restrições que se aplicam ao tipo de recurso.

Tipo de recurso	Como colocá-lo sob o controle de confirmação	Como removê-lo do controle de confirmação	Quais tipos de alterações são confirmáveis	Restrições
FILE- arquivos de banco de dados local	Abrindo sob o controle de confirmação <sup>1</sup>	Fechando o arquivo, se nenhuma alteração estiver pendente.  Se as alterações estiverem pendentes quando o arquivo estiver fechado, depois de executar a operação seguinte de confirmação ou rollback.	Alterações de nível de registro	Não é possível bloquear mais de 500.000.000 registros em uma única transação <sup>2</sup> .
DDL - Alterações de nível de objeto em tabelas locais do SQL e coleções SQL.	Executando o SQL sob o controle de confirmação	Executando uma operação de confirmação ou rollback após alteração do nível do objeto.	Alterações de nível de objeto, tais como: <ul style="list-style-type: none"> <li>• Criar pacote SQL</li> <li>• Criar tabela SQL</li> <li>• Eliminar tabela SQL</li> </ul>	Somente alterações de nível de objeto feitas com o SQL estão sob o controle de confirmação.
DDM- arquivo distributed data management (DDM) remoto	Abrindo sob o controle de confirmação. O suporte ao controle de confirmação para DDM possui informações adicionais sobre controle de confirmação e gerenciamento de dados distribuídos.	Fechando o arquivo, se nenhuma alteração estiver pendente.  Se as alterações estiverem pendentes quando o arquivo estiver fechado, depois de executar a operação seguinte de confirmação ou rollback.	Alterações de nível de registro	
LU 6.2 - conversaç�o protegida	Iniciando a conversaç�o <sup>3</sup>	Finalizando a conversaç�o		
DRDA - banco de dados relacional distribuído	Utilizando a instruç�o SQL CONNECT	Finalizando a conex�o		

Tipo de recurso	Como colocá-lo sob o controle de confirmação	Como removê-lo do controle de confirmação	Quais tipos de alterações são confirmáveis	Restrições
API - recurso de confirmação da API local	API Add Commitment Resource (QTNADDCR)	API Remove Commitment Resource (QTNRMVCR)	O programa do usuário determina isto. As entradas do diário podem ser gravadas pelo programa do usuário utilizando a API Send Journal Entry (QJOSJRNE) para ajudar a rastrear essas alterações.	O aplicativo deve fornecer um programa de saída para ser chamado durante as operações de confirmação, rollback e ressincronização.
TCP - Conexão TCP/IP	Utilizando a instrução SQL CONNECT para um RDB definido para usar as conexões TCP/IP ou abrindo um arquivo DDM definido com uma localização TCP/IP	Finalizando a conexão SQL ou fechando o arquivo DDM se nenhuma alteração estiver pendente. Se o arquivo DDM foi fechado com alterações pendentes, a conexão será fechada depois de executar a próxima operação de confirmação ou rollback.		

**Notas:**

<sup>1</sup>Para obter detalhes sobre como colocar um arquivo de banco de dados sob o controle de confirmação, consulte o manual de referência de linguagem apropriado. Informações relacionadas ao controle de confirmação têm links para manuais de linguagem que você pode utilizar.

<sup>2</sup> Você pode utilizar um arquivo QAQQINI para reduzir o limite de 500.000.000. Consulte “Gerenciando o Tamanho da Transação” na página 75 para obter instruções.

<sup>3</sup>Quando uma conexão DDM é iniciada, o arquivo DDM especifica PTCCNV(\*YES) e esse arquivo é definido com um local remoto SNA; um recurso da LU 6.2 é incluído com o recurso DDM.

Quando uma conexão DRDA for iniciada, um recurso da LU 6.2 será incluído com o recurso DRDA, se ambas as condições a seguir forem verdadeiras:

- O programa está utilizando os protocolos de conexão da unidade de trabalho distribuído.
- A conexão é para um RDB (Rational Database) definido com um local remoto SNA. Para obter informações

adicionais sobre como iniciar conversões protegidas, consulte Programação APPC .

**Conceitos relacionados**

Programação de Bancos de Dados Distribuídos

“Atualizações Feitas no Objeto de Notificação” na página 65

O sistema atualizará o objeto de notificação com a identificação de confirmação da última operação de confirmação bem-sucedida dessa definição de confirmação.

**Referências relacionadas**

API Add Commitment Resource (QTNADDCR)

API Remove Commitment Resource (QTNRMVCR)

API Send Journal Entry (QJOSJRNE)

## Recursos Confirmáveis Locais e Remotos

Um recurso confirmável pode ser um recurso local ou um recurso remoto.

### Recurso Confirmável Local

Um recurso confirmável local fica no mesmo sistema que o aplicativo. Cada diário associado aos recursos sob controle de confirmação pode ser considerado como uma localização local. Todos os recursos que são registrados sem um diário (opcionalmente para recursos DDL e recursos API) podem ser considerados como uma localização local separada.

Se um recurso confirmável estiver em um conjunto de disco independente e a definição de confirmação estiver em um conjunto de discos diferente, o recurso não será considerado local.

### Recursos Confirmáveis Remotos

Um recurso confirmável remoto está em um sistema diferente do aplicativo. Uma localização remota existe para cada conversação exclusiva a um sistema remoto. Uma definição de confirmação pode ter um ou mais locais remotos em um ou mais sistemas remotos.

Quando você coloca um recurso local sob controle de confirmação para o conjunto de discos do sistema, ou quaisquer conjuntos de discos independentes, é necessário utilizar o DRDA (Distributed Relational Database Architecture) para acessar os recursos sob o controle de confirmação em qualquer outro conjunto de discos independentes.

A tabela a seguir mostra os tipos de recursos confirmáveis e seus locais.

Tipo de recurso	Localização
API	Local
DDL	Local
DDM	Remoto
DRDA	Local ou remoto
ARQUIVO	Local
LU62	Remoto
TCP	Remoto

#### Conceitos relacionados

“Controle de Confirmação e Conjuntos de Discos Independentes” na página 22

Os conjuntos de discos independentes e os grupos de conjuntos de discos independentes podem cada um ter um banco de dados i5/OS separado. Você pode usar o controle de confirmação com estes bancos de dados.

### Intenção de Acesso de um Recurso Confirmável

A intenção de acesso determina como os recursos participam juntos de uma transação.

Quando um recurso é colocado sob controle de confirmação, o gerenciador de recursos indica como o recurso é acessado:

- Atualização
- Somente leitura
- Indeterminada

A tabela a seguir mostra quais intenções de acesso são possíveis para um determinado tipo de recurso e como o sistema determina a intenção de acesso para um recurso quando ele é registrado.

Tipo de recurso	Possíveis intenções de acesso	Como é determinada a intenção de acesso
ARQUIVO	Atualização, somente leitura	Com base em como o arquivo foi aberto
DDL	Atualização	Sempre atualizar
API	Atualização	Sempre atualizar
DDM	Atualização, somente leitura	Com base em como o arquivo foi aberto
LU62	Indeterminada	Sempre indeterminada
DRDA	Atualização, somente leitura, indeterminada	Para o DRDA Nível 1, a intenção de acesso é atualização se nenhum outro recurso remoto estiver registrado. Caso contrário, a intenção de acesso é somente leitura. Para o DRDA Nível 2, a intenção de acesso é sempre indeterminada.
TCP	Indeterminada	Sempre indeterminada

A intenção de acesso de recursos que já estão registrados determina se um novo recurso pode ser registrado. As seguintes regras se aplicam:

- Um recurso de uma fase, cuja intenção de acesso é atualização, não pode ser registrado quando uma das seguintes condições é verdadeira:
  - Recursos cuja intenção de acesso é atualização já estão registrados em outras localizações.
  - Recursos cuja intenção de acesso é indeterminada já estão registrados em outras localizações.
  - Recursos cuja intenção de acesso é indeterminada já estão registrados na mesma localização e os recursos foram alterados durante a transação atual.
- Um recurso de duas fases, cuja intenção de acesso é atualização, não pode ser registrado quando um recurso de uma fase, cuja intenção de acesso é atualização, já está registrado.

### Protocolo de Confirmação de um Recurso Confirmável

*Protocolo de confirmação* é a capacidade que um recurso tem de participar em um processamento de confirmação de uma ou duas fases. Os recursos locais, exceto os recursos confirmáveis da API, sempre são recursos de duas fases.

Se um recurso confirmável residir em um conjunto de discos independentes e a definição de confirmação residir em um conjunto de discos diferente, o recurso não será considerado local ou de duas fases.

Um recurso de duas fases também se chama *recurso protegido*. Os recursos remotos e os recursos confirmáveis da API devem ser registrados como recursos de uma ou de duas fases quanto forem colocados sob o controle de confirmação. A tabela a seguir mostra quais tipos de recursos confirmáveis podem coexistir em uma definição de confirmação com um recurso de uma fase.

Tipo de recurso	Pode Coexistir Com
Recurso da API de uma fase	Outros recursos locais. Sem recursos remotos.
Recurso remoto de uma fase	Outros recursos de uma fase na mesma localização. Nenhum recurso local.

### Conceitos relacionados

“Controle de Confirmação e Conjuntos de Discos Independentes” na página 22

Os conjuntos de discos independentes e os grupos de conjuntos de discos independentes podem cada um ter um banco de dados i5/OS separado. Você pode usar o controle de confirmação com estes bancos de dados.

## **Arquivos Registrados em Diário e Controle de Confirmação**

É preciso registrar em diário (registrar) um arquivo de banco de dados (tipo de recurso FILE ou DDM) para que ele possa ser aberto para saída sob controle de confirmação ou referenciado por um aplicativo SQL que utilize um nível de isolamento que não seja \*NONE (sem confirmação). Um arquivo não precisa ser registrado em diário para ser aberto apenas para entrada sob controle de confirmação.

Ocorrerá um erro se alguma das seguintes condições for verdadeira:

- Há uma tentativa de abertura de um arquivo do banco de dados para saída sob controle de confirmação, mas o arquivo não está atualmente registrado em diário.
- Não há definição de confirmação iniciada que possa ser usada pelo arquivo sendo aberto sob controle de confirmação.

Se somente as imagens posteriores estiverem sendo registradas em diário para um arquivo de banco de dados quando este arquivo é aberto sob controle de confirmação, o sistema começa a registrar automaticamente em diário as imagens anteriores e posteriores. As imagens anteriores são gravadas apenas para alterações feitas no arquivo que ocorrem sob controle de confirmação. Se ocorrerem, ao mesmo tempo, outras alterações no arquivo que não estão sob controle de confirmação, apenas as imagens posteriores são gravadas para estas alterações.

O sistema grava automaticamente em um diário alterações confirmáveis de nível de registro e de nível de objeto. Para alterações de nível de registro, o sistema utiliza as entradas no diário, se necessário, para fins de recuperação; ele não utiliza as entradas de alterações confirmáveis de nível de objeto para fins de recuperação. Além disso, o sistema não grava automaticamente entradas no diário para recursos de confirmação da API. Porém, o programa de saída para o recurso da API pode usar a API Send Journal Entry (QJOSJRNE) para gravar entradas no diário para fornecer uma trilha de auditoria ou para ajudar na recuperação. O conteúdo destas entradas é controlado pelo programa de saída do usuário.

O sistema utiliza uma técnica, que não é um diário, para fazer a recuperação para recursos de confirmação de nível de objeto. A recuperação para os recursos de confirmação da API é executada chamando o programa de saída de confirmação e rollback associado a cada determinado recurso de confirmação da API. O programa de saída tem a responsabilidade de execução da verdadeira recuperação necessária à situação.

### **Conceitos relacionados**

Gerenciamento de Diário

## **Seqüência das Entradas no Diário sob o Controle de Confirmação**

Esta tabela mostra a seqüência de entradas que são geralmente gravadas enquanto uma definição de confirmação está ativa. Você pode utilizar o localizador de informações de entrada do Diário para obter informações adicionais sobre o conteúdo das entradas do diário.

As entradas do controle de confirmação serão gravadas em um diário local, se pelo menos uma das seguintes condições for verdadeira:

- O diário é especificado como diário padrão no comando Start Commitment Control (STRCMTCTL).
- Pelo menos um arquivo registrado no diário for aberto sob o controle de confirmação.
- Pelo menos um recurso de confirmação da API associado ao diário é registrado sob o controle de confirmação.

Tipo de Entrada	Descrição	Onde é Gravada	Quando é Gravada
C BC	Inicia o controle de confirmação	No diário padrão, se uma estiver especificada no comando STRCMTCTL.	Quando o comando STRCMTCTL é utilizado.
		No diário.	Quando o primeiro arquivo registrado em um diário é aberto ou quando um recurso da API é registrado em um diário.
C SC	Inicia o ciclo de confirmação	No diário.	Quando ocorre a primeira alteração de registro da transação de um arquivo registrado nesse diário <sup>1</sup> .
		No diário de um recurso da API.	Quando a API QJOSJRNE é utilizada pela primeira vez com a chave <i>Incluir Identificador do Ciclo de Confirmação</i> .
Códigos do diário D e F	Entradas de nível de objeto DDL	No diário associado ao objeto que está sendo atualizado. Somente entradas de diário que contêm um identificador de ciclo de confirmação representam uma alteração de nível de objeto DDL que seja parte da transação.	Quando ocorrem atualizações.
Código do diário R	Entradas de nível de registro	No diário associado ao arquivo que está sendo atualizado.	Quando ocorrem as atualizações.
Código do diário U	Entradas criadas pelo usuário	No diário associado a um recurso da API.	Se o programa aplicativo que utiliza a API QJOSJRNE for utilizado primeiramente com a chave <i>Incluir Identificador do Ciclo de Confirmação</i> .
C CM	Confirmação	No diário.	Quando a confirmação foi concluída com sucesso.
		No diário padrão.	Se algum recurso confirmável estiver associado ao diário.
C RB	Rollback	No diário.	Após a conclusão da operação de rollback.
		No diário padrão.	Se algum recurso confirmável estiver associado ao diário.



Tipo de Entrada	Descrição	Onde é Gravada	Quando é Gravada
C LW	Finalização da transação	No diário padrão, se uma estiver especificada no comando STRCMTCTL. O sistema grava um registro de cabeçalho de LW e um ou mais registros de detalhes. Essas entradas serão gravadas somente se OMTJRNE(*NONE) estiver especificado no comando STRCMTCTL ou se ocorrer um erro do sistema.	Quando a operação de confirmação ou rollback é concluída.
C EC	Finalizar o Controle de Confirmação	No diário.	Quando o comando End Commitment Control (ENDCMTCTL) é concluído.
		Em um diário local que não seja o padrão.	Quando um limite de confirmação é estabelecido, depois do ponto em que todos os recursos confirmáveis associados a esse diário foram removidos do controle de confirmação.
C SB	Início do ciclo de confirmação aninhado ou do ponto de salvamento.	No diário.	Quando o aplicativo cria um SQL SAVEPOINT ou quando o sistema cria um ciclo de confirmação interno aninhado para manipular uma série de funções do banco de dados como uma única operação <sup>2</sup> .
C SQ	Liberação do ponto de salvamento ou confirmação do ciclo de confirmação aninhado.	No diário.	Quando o aplicativo libera um SQL SAVEPOINT ou quando o sistema confirma um ciclo de confirmação interno aninhado <sup>2</sup> .
C SU	Rollback do ponto de salvamento ou do ciclo de confirmação aninhado.	No diário.	Quando o aplicativo efetua rollback de um SQL SAVEPOINT ou quando o sistema efetua rollback de um ciclo de confirmação interno aninhado <sup>2</sup> .

Tipo de Entrada	Descrição	Onde é Gravada	Quando é Gravada
<b>Notas:</b>			
<p><sup>1</sup>Você pode especificar que a parte do comprimento fixo da entrada no diário inclui informações sobre a transação especificando o valor de Unidade Lógica de Trabalho (*LUW) do parâmetro Fixed-Length Data (FIXLENDTA) dos comandos Create Journal (CRTJRN) ou Change Journal (CHGJRN). Ao especificar o parâmetro FIXLENDTA (*LUW), a porção de tamanho fixo de cada entrada do diário C SC irá conter o ID da Unidade Lógica de Trabalho (LUWID) da transação atual. Da mesma forma, para transações XA, se você especificar o parâmetro FIXLENDTA(*XID), a parte de tamanho fixo de cada entrada no diário C SC conterá o XID da transação atual. O LUWID ou o XID podem ajudá-lo a encontrar todos os ciclos de confirmação de uma transação em particular, se vários diários ou sistemas estiverem envolvidos na transação.</p> <p><sup>2</sup> Essas entradas serão enviadas apenas se você configurar a variável de ambiente QTN_JRNSAVPT_MYLIB_MYJRN como *YES, em que MYJRN é o diário que está sendo utilizado e MYLIB é a biblioteca em que o diário está armazenado. O valor especial *ALL é suportado para os valores MYLIB e MYJRN. Essas variáveis podem ser definidas para todo o sistema ou para um job específico. Para que as entradas sejam enviadas para o diário MYLIB/MYJRN, para apenas um job, utilize este comando nesse job:</p> <ul style="list-style-type: none"> <li>• ADDENVVAR ENVVAR(QTN_JRNSAVPT_MYLIB_MYJRN) VALUE(*YES)</li> </ul> <p>Para que as entradas sejam enviadas para todos os diários de todas as tarefas, utilize este comando:</p> <ul style="list-style-type: none"> <li>• ADDENVVAR ENVVAR('QTN_JRNSAVPT_*ALL_*ALL') VALUE(*YES) LEVEL(*SYS)</li> </ul> <p>É necessário definir a variável de ambiente antes de iniciar o controle de confirmação.</p>			

### Conceitos relacionados

“Operação de Confirmação” na página 3

Uma operação confirmar torna permanente todas as alterações feitas sob o controle de confirmação desde a operação anterior de confirmação ou rollback. O sistema também libera todos os bloqueios relacionados à transação.

Localizador de Informações de Entradas do Diário

### Referências relacionadas

Comando End Commitment Control (ENDCMTCTL)

## Identificador do Ciclo de Confirmação

Um *ciclo de confirmação* é o tempo de um limite de confirmação até o próximo. O sistema atribui um *identificador de ciclo de confirmação* para associar todas as entradas do diário para um determinado ciclo de confirmação ao mesmo tempo. Cada diário que participa de uma transação possui seu próprio ciclo de confirmação e seu próprio identificador de ciclo de confirmação.

O identificador de ciclo de confirmação é o número de seqüência do diário da entrada do diário C SC gravado para o ciclo de confirmação. O identificador de ciclo de confirmação é colocado em cada entrada do diário gravada durante o ciclo de confirmação. Se mais de um diário for usado durante o ciclo de confirmação, o identificador para cada diário é diferente.

Você pode especificar que a porção de tamanho fixo da entrada do diário contém informações da transação especificando o valor da Unidade Lógica de Trabalho (\*LUW) para o parâmetro Fixed-Lenght Data (FIXLENDTA) do comando Create Journal (CRTJRN) ou Change Journal (CHGJRN). Ao especificar o parâmetro FIXLENDTA (\*LUW), a porção de tamanho fixo de cada entrada do diário C SC irá conter o ID da Unidade Lógica de Trabalho (LUWID) da transação atual. Da mesma forma, para transações XA, se você especificar o parâmetro FIXLENDTA (\*XID), a parte de tamanho fixo de cada entrada do diário C SC conterá o XID da transação atual. O LUWID ou o XID podem ajudá-lo a encontrar todos os ciclos de confirmação de uma transação em particular, se vários diários ou sistemas estiverem envolvidos na transação.

Você pode usar a API Send Journal Entry (QJOSJRNE) para gravar entradas de diário para recursos da API. Você tem a opção de incluir o identificador do ciclo de confirmação nestas entradas de diário.

Você pode usar o identificador de ciclo de confirmação para aplicar ou remover alterações registradas em diário em um limite de confirmação utilizando o comando Apply Journalized Changes (APYJRNCHG) ou o comando Remove Journalized Changes (RMVJRNCHG). Estas limitações se aplicam:

- A maioria das alterações no nível de objeto feitas sob controle de confirmação são gravadas no diário mas não aplicadas ou removidas utilizando os comandos APYJRNCHG e RMVJRNCHG.
- A API QJOSJRNE grava entradas de diário criadas pelo usuário com um código de diário de U. Estas entradas não podem ser aplicadas ou removidas utilizando os comandos APYJRNCHG e RMVJRNCHG. Elas devem ser aplicadas ou removidas com um programa desenvolvido pelo usuário.

## Bloqueio de Registro

Quando uma tarefa suspende um bloqueio de registro e outra tarefa tenta recuperar esse registro para atualização, a tarefa solicitante aguarda e é removida do processamento ativo.

A tarefa solicitante estará ativa até que ocorra um dos seguintes eventos:

- O bloqueio de registro é liberado.
- O tempo de espera especificado finaliza.

Mais de um job pode solicitar que um registro seja bloqueado por outro job. Quando o bloqueio do registro é liberado, o primeiro job a solicitar recebe o registro. Enquanto aguarda por um registro bloqueado, especifique o tempo de espera no parâmetro WAITRCD nos seguintes comandos create, change ou override:

- Create Physical File (CRTPF)
- Create Logical File (CRTLF)
- Create Source Physical File (CRTSRCPF)
- Change Physical File (CHGPF)
- Change Logical File (CHGLF)
- Change Source Physical File (CHGSRCPF)
- Override Database File (OVRDBF)

Ao especificar o tempo de espera, considere as seguintes informações:

- Se você não especificar um valor, o programa aguardará o tempo de espera padrão do processo.
- Para obter somente definições de confirmação com bloqueios de escopo de transação, é possível substituir o tempo de espera do job padrão por um tempo de espera do bloqueio da transação, que pode ser especificado em:
  - Na API xa\_open.
  - Na interface JDBC ou JTA. As transações distribuídas listam essas APIs.
- Se não for possível alocar o registro dentro do tempo especificado, será enviada uma mensagem de notificação ao programa de linguagem de alto nível.
- Se o tempo de espera de um registro exceder, a mensagem enviada ao log dos jobs informará o nome do job que suspende o registro bloqueado que causou a espera do job solicitante. Se você encontrar exceções de bloqueio de registro, poderá usar o log dos jobs para determinar quais programas irá alterar para que não suspendam bloqueios por períodos longos.

Os programas mantêm bloqueios de registros por longos períodos por um dos seguintes motivos:

- O registro permanecerá bloqueado enquanto o usuário da estação de trabalho estiver considerando uma alteração.
- O bloqueio de registro faz parte de uma longa transação de confirmação. Faça transações menores de modo que uma operação de confirmação possa ser executada com mais frequência.

- Ocorreu um bloqueio indesejado. Por exemplo, suponha que um arquivo esteja definido como um arquivo de atualização com chaves exclusivas e o programa atualize e inclua registros adicionais no arquivo. Se o usuário da estação de trabalho desejar incluir um registro no arquivo, o programa poderá tentar acessar o registro para determinar se a chave já existe. Se existir, o programa informará ao usuário da estação de trabalho que o pedido feito não é válido. Quando o registro é recuperado do arquivo, ele é bloqueado até ser liberado implicitamente por outra operação de leitura no mesmo arquivo ou até ser liberado explicitamente.

**Nota:** Para obter informações adicionais sobre como utilizar cada interface da linguagem de alto nível para liberar bloqueios de registros, consulte o manual de referência da linguagem de alto nível apropriado. Informações relacionadas ao controle de confirmação têm links para manuais de linguagem de alto nível que podem ser utilizados com o controle de confirmação.

A duração do bloqueio será maior se LCKLVL(\*ALL) estiver especificado porque o registro recuperado do arquivo estará bloqueado até a próxima operação de confirmação ou rollback. Não há liberação implícita feita por outra operação de leitura e não é possível que seja explícita.

Outra função que pode colocar um bloqueio no arquivo é a função salvar enquanto ativo.

#### **Conceitos relacionados**

Transações Distribuídas JDBC

Função Save-while-active

#### **Referências relacionadas**

“Informações Relacionadas para o Controle de Confirmação” na página 118

Manuais do Produto, Publicações do IBM Redbooks, Web Sites e outras coletas de tópico do centro de informações contêm informações relacionadas à coleta de tópico de controle de Confirmação. É possível visualizar ou imprimir quaisquer dos arquivos PDF.

## **Controle de Confirmação e Conjuntos de Discos Independentes**

Os conjuntos de discos independentes e os grupos de conjuntos de discos independentes podem cada um ter um banco de dados i5/OS separado. Você pode usar o controle de confirmação com estes bancos de dados.

No entanto, como cada conjunto de discos independentes ou grupo de conjuntos de discos independentes possui um banco de dados SQL separado, você deve estar ciente de algumas considerações.

#### **Conceitos relacionados**

“Definição de Confirmação” na página 5

A definição de confirmação contém informações que pertencem aos recursos sendo alterados sob controle de confirmação durante uma transação.

“Recursos Confirmáveis Locais e Remotos” na página 15

Um recurso confirmável pode ser um recurso local ou um recurso remoto.

“Protocolo de Confirmação de um Recurso Confirmável” na página 16

*Protocolo de confirmação* é a capacidade que um recurso tem de participar em um processamento de confirmação de uma ou duas fases. Os recursos locais, exceto os recursos confirmáveis da API, sempre são recursos de duas fases.

## **Considerações do Conjunto de Discos Independentes para Definições de Confirmação**

Esteja ciente dessas considerações para as definições de confirmação ao utilizar conjuntos de discos independentes.

## **Considerações sobre a Biblioteca QRECOVERY**

Quando você inicia o controle de confirmação, a definição de confirmação é criada na biblioteca QRECOVERY. Cada conjunto de discos independente ou grupo de conjuntos de discos independentes

possui sua própria versão de uma biblioteca QRECOVERY. Em um conjunto de discos independentes, o nome da biblioteca QRECOVERY é QRCYxxxxx, onde xxxxx é o número do conjunto de discos independente. Por exemplo, o nome da biblioteca QRECOVERY para o conjunto de discos independente 39 é QRCY00039. Além disso, se o conjunto de discos independentes fizer parte de um grupo de conjuntos de discos, somente o conjunto de discos primário possui uma biblioteca QRCYxxxxx.

Quando você inicia o controle de confirmação, a definição de confirmação é criada na biblioteca QRECOVERY do conjunto de discos independentes associado àquele job, tornando o controle de confirmação ativo no conjunto de discos independentes.

## Considerações sobre Set ASP Group

O uso do comando Set ASP Group (SETASPGRP), enquanto o controle de confirmação está ativo em um conjunto de discos independentes, tem os seguintes efeitos:

- Se você comutar de um conjunto de discos independentes e os recursos estiverem registrados com o controle de confirmação do conjunto de discos, o comando SETASPGRP falhará com a mensagem CPDB8EC, código de razão 2, 0 encadeamento possui uma transação não confirmada. Essa mensagem é seguida pela mensagem CPFB8E9.
- Se você muda de um conjunto de discos independentes e nenhum recurso está registrado com o controle de confirmação, as definições de confirmação são movidas para o conjunto de discos independentes para o qual você está mudando.
- Se você muda do conjunto de discos do sistema (grupo ASP \*NONE), o controle de confirmação não é afetado. As definições de confirmação permanecem no conjunto de discos do sistema.
- Quando você utiliza um objeto de notificação, este deve residir no mesmo conjunto de discos independentes ou grupo de conjuntos de discos independentes que a definição de confirmação.
- Se você mover a definição de confirmação para outro conjunto ou grupo de discos independentes, o objeto de notificação também deverá residir neles. O objeto de notificação no outro conjunto de discos independentes ou grupo de conjuntos de discos independentes será atualizado se a definição de confirmação finalizar de forma anormal. Se o objeto de notificação não for encontrado no outro conjunto ou grupo de discos independentes, a atualização falhará a mensagem CPF8358.

| O espaço de nome atual do job determina em qual ASP independente a definição de confirmação é criada. É possível abrir um arquivo sob o controle de confirmação para a definição de confirmação sob as seguintes condições:

- | • Quando o arquivo estiver no mesmo ASP independente da definição de confirmação.
- | • Quando o arquivo estiver no \*SYSBAS ASP e a definição de confirmação não tiver recursos registrados no ASP independente.

| Se você abrir um arquivo que esteja no \*SYSBAS ASP enquanto a definição de confirmação estiver em um ASP independente, a definição de confirmação será movida para o ASP \*SYSBAS e nenhum recurso no ASP independente pode ser registrado.

## Considerações sobre Diário Padrão

Você deve estar ciente das seguintes considerações sobre diário padrão:

- Se você utilizar o diário padrão, o diário deverá residir no mesmo conjunto de discos independentes ou no grupo de conjuntos de discos independentes da definição de confirmação.
- Se, quando o controle de confirmação for iniciado, o diário padrão não for encontrado em outro conjunto de discos independentes ou grupo de conjuntos de discos independentes, o início do controle de confirmação falhará com a mensagem CPF9873.
- Se você mover a definição de confirmação para outro conjunto de discos independentes ou grupo de conjuntos de discos independentes, o diário padrão também deverá residir nesse outro conjunto de discos independentes ou grupo de conjuntos de discos independentes. Se o diário não for encontrado

em outro conjunto de discos independentes ou grupo de conjuntos de discos independentes, a definição de confirmação será movida, mas nenhum diário padrão será utilizado desse ponto em diante.

## Considerações sobre o IPL (Initial Program Load) e a Desativação

Você deve estar ciente das seguintes considerações sobre IPL e Desativação:

- A recuperação das definições de confirmação, que residem em um conjunto de discos independentes, é desempenhada durante o processamento de ativação do conjunto de discos independentes e é semelhante à recuperação IPL.
- As definições de confirmação em um conjunto de discos independentes não são recuperadas durante o IPL do sistema.
- A desativação de um conjunto de discos independentes tem os seguintes efeitos nas definições de confirmação:
  - O jobs associados ao conjunto de discos independentes são finalizados.
  - Nenhuma definição de confirmação nova pode ser criada no conjunto de discos independente.
  - As definições de confirmação que residem no conjunto de discos independentes tornam-se inutilizáveis.
  - As definições de confirmação que residem no conjunto de discos independentes, mas não anexadas a um job, liberam bloqueios com escopo de transação.

## Considerações sobre Banco de Dados Remoto

Você deve estar ciente das seguintes considerações sobre banco de dados remoto:

- Você não pode usar uma conexão SNA LU 6.2 (conversações protegidas ou Unidade de Trabalho Distribuída (DUW)) para conectar-se a um banco de dados remoto a partir de um banco de dados do conjunto de discos independentes. Você pode usar conversações SNA desprotegidas para conectar-se de um banco de dados do conjunto de discos independentes para um banco de dados remoto.
- Quando o controle de confirmação está ativo para um job ou encadeamento, o acesso aos dados fora do conjunto de discos independentes ou grupo de conjuntos de discos, ao qual pertence a definição de confirmação, só é possível remotamente, como se fossem dados que residem em outro sistema. Quando você emite uma instrução SQL CONNECT para conectar-se ao RDB (Relational Database) no conjunto de discos independentes, o sistema transforma a conexão em uma conexão remota.
- O conjunto de discos do sistema e os conjuntos de discos básicos não exigem uma conexão remota para acesso somente leitura aos dados que residem em um conjunto de discos independentes. Da mesma forma, um conjunto de discos independentes não exige uma conexão remota para acesso somente leitura aos dados que residem no conjunto de discos do sistema ou no conjunto de discos básico.

### Conceitos relacionados

“Definição de Confirmação” na página 5

A definição de confirmação contém informações que pertencem aos recursos sendo alterados sob controle de confirmação durante uma transação.

“Objeto de Notificação de Confirmação” na página 52

Um *objeto de notificação* é uma fila de mensagens, área de dados ou arquivo de banco de dados contendo informações que identificam a última transação concluída com sucesso para uma determinada definição de confirmação, se ela não foi finalizada normalmente.

## Considerações para Transações XA

No ambiente XA, cada banco de dados é considerado um gerenciador de recursos separado. Quando um gerenciador de transações deseja acessar dois bancos de dados sob a mesma transação, ele deve usar os protocolos XA para executar a confirmação de duas fases com os dois gerenciadores de recursos.

Como cada conjunto de discos independentes é um banco de dados SQL separado, no ambiente XA, cada conjunto de discos independentes também é considerado um gerenciador de recursos separado. Para que um servidor de aplicativos execute uma transação que tem como alvo dois conjuntos de discos independentes diferentes, o gerenciador de transações também deve usar um protocolo de confirmação de duas fases.

#### Conceitos relacionados

“Suporte de Transação XA para o Controle de Confirmação” na página 45  
O DB2 para i5/OS pode participar nas transações globais do X/Open.

Exemplos do Conjunto de Discos Independentes

## Considerações e Restrições para o Controle de Confirmação

Você precisa estar ciente dessas considerações e restrições para o controle de confirmação.

### Considerações sobre Arquivo de Banco de Dados

- Se você especificar que um arquivo compartilhado será aberto sob o controle de confirmação, todos os usos seguintes desse arquivo deverão ser abertos sob o controle.
- Se SEQONLY(\*YES) for especificado para o arquivo aberto no modo de leitura com LCKLVL(\*ALL) (implicitamente ou por um programa de linguagem de alto nível ou explicitamente pelo comando Override with Database File (OVRDBF)), SEQONLY(\*YES) será ignorado e SEQONLY(\*NO) utilizado.
- As alterações de nível de registro feitas sob o controle de confirmação são gravadas em um diário. Essas alterações podem ser aplicadas ou removidas do banco de dados com o comando Apply Journalled Changes (APYJRNCHG) ou o comando Remove Journalled Changes (RMVJRNCHG).
- As imagens antes e depois dos arquivos são registradas em diário sob o controle de confirmação. Se você especificar somente registrar em diário as imagens de depois dos arquivos, o sistema também registrará em diário automaticamente a imagem de antes das alterações do arquivo ocorridas sob o controle de confirmação. Contudo, como as imagens de antes não são capturadas para todas as alterações feitas nos arquivos, não será possível utilizar o comando RMVJRNCHG para esses arquivos.

### Considerações para Alterações no Nível de Objeto e no Nível de Registro

- As alterações de nível de objeto e de registro feitas sob o controle de confirmação com SQL utilizarão a definição de confirmação atualmente ativa para o grupo de ativação em que o programa solicitante está sendo executado. Se as definições de confirmação de nível de tarefa e de nível de grupo de ativação não estiverem ativadas, o SQL iniciará uma definição de confirmação de nível de grupo de ativação.

### Considerações sobre Confirmação de Uma Fase e de Duas Fases

- Durante o estabelecimento de uma conversão remota ou conexão de uma fase, elas não são permitidas em outras localidades. Se um limite de confirmação estiver estabelecido e todos os recursos forem removidos, a localização poderá ser alterada.
- Se você estiver utilizando a confirmação de duas fases, não será necessário utilizar o comando Submit Remote Command (SBMRMTCMD) para iniciar o controle de confirmação ou desempenhar qualquer outra operação de controle de confirmação nos locais remotos. O sistema executa essas funções para você.
- Para um local remoto de uma fase, os comandos de CL COMMIT e ROLLBACK falharão se o SQL estiver na pilha de chamada e o banco de dados relacional remoto não estiver em um sistema. Se o SQL não estiver na pilha de chamada, os comandos COMMIT e ROLLBACK não falharão.
- Para uma localização remota de uma fase, o controle de confirmação deve ser iniciado no sistema-fonte antes que sejam feitas alterações confirmáveis nos recursos remotos. O sistema inicia automaticamente o controle de confirmação para o SQL do banco de dados distribuído no sistema-fonte na hora da conexão se o programa SQL estiver sendo executado com uma opção de controle de confirmação diferente de \*NONE. Quando o primeiro recurso remoto for colocado sob o controle de confirmação, o sistema iniciará o controle de confirmação no sistema de destino.

## Consideração sobre Salvamento

Uma operação de salvamento será evitada se o job que executar o salvamento possuir uma ou mais definições de confirmação ativas com qualquer um dos tipos de alterações confirmáveis a seguir:

- Uma alteração de registro num arquivo que resida na biblioteca sendo salva. Para arquivos lógicos, todos os arquivos físicos relacionados serão verificados.
- Quaisquer alterações de nível de objeto dentro de uma biblioteca que está sendo salva.
- Qualquer recurso da API incluído utilizando a API Add Commitment Resource (QTNADDCR) e com o campo Permitir processamento de salvamento normal configurado com o valor padrão N.

Isto impede que as operações de salvamento salvem na mídia alterações que sejam devidas uma transação parcial.

**Nota:** Se você utilizar o novo salvamento com o recurso de transações parciais, o objeto poderá ser salvo sem finalizar uma definição de confirmação.

Bloqueios de objeto e registro impedem que alterações pendentes de definições de confirmação em outros jobs sejam salvas na mídia de salvamento. Isto será verdadeiro somente para recursos de confirmação da API se os bloqueios forem adquiridos quando as alterações forem feitas nos objetos ou objetos associados ao recurso de confirmação da API.

## Considerações e Restrições Diversas

- Antes de atualizar seu sistema para um release novo, todas as ressincronizações pendentes deverão ser concluídas ou canceladas.
- Os valores COMMIT e ROLLBACK são mostrados no campo Função WRKACTJOB durante uma confirmação ou um rollback. Se a Função permanecer COMMIT ou ROLLBACK por um período longo, poderá ter ocorrido um dos seguintes eventos:
  - Uma falha de recurso durante a confirmação ou o rollback requer ressincronização. O controle não retornará ao aplicativo até que a ressincronização seja concluída ou cancelada.
  - Esse sistema votou em durante a confirmação. O controle não retornará ao aplicativo até que o sistema que iniciou a confirmação envie dados para esse sistema.
  - Esse sistema votou em OK para ficar de fora durante a confirmação. O controle não retornará ao aplicativo até que o sistema que iniciou a confirmação envie dados para esse sistema.

### Conceitos relacionados

Assegurando a Integridade de Confirmação de Duas Fases

“Nível de Bloqueio de Confirmação” na página 54

O valor especificado para o parâmetro LCKLVL no comando Start Commitment Control (STRCMTCTL) torna-se o nível padrão de bloqueio de registro para os arquivos de banco de dados que são abertos e colocados sob controle de confirmação para a definição de confirmação.

### Referências relacionadas

Comando Override with Database File (OVRDBF)

Comando Apply Journaled Changes (APYJRNCHG)

Comando Remove Journaled Changes (RMVJRNCHG)

Programação de SQL

Comando Submit Remote Command (SBMRMTCMD)

Comando Commit (COMMIT)

Comando Rollback (ROLLBACK)

API Add Commitment Resource (QTNADDCR)



## Controle de Confirmação para Aplicativos Batch

As aplicações em batch podem ou não precisar de controle de confirmação. Em alguns casos, um aplicativo batch pode executar uma única função de leitura de um arquivo de entrada e atualização de um arquivo master. Porém, você pode usar o controle de confirmação para este tipo de aplicativo se for importante iniciá-lo novamente após uma finalização anormal.

O arquivo de entrada é um arquivo de atualização com um código nos registros para indicar que um registro foi processado. Este arquivo e quaisquer arquivos atualizados são colocados sob o controle de confirmação. Quando o código está presente no arquivo de entrada, ele representa uma transação concluída. O programa faz a leitura pelo arquivo de entrada e desvia dos registros com o código concluído. Isto permite que a mesma lógica de programa seja usada para condições normais e de reinício.

Se a aplicação em batch contiver registros de entrada dependentes entre si e contiver chaves ou totais, um objeto de notificação poderá ser utilizado para fornecer informações sobre reinício. Os valores mantidos no objeto de notificação são utilizados para iniciar o processamento novamente a partir da última transação confirmada dentro do arquivo de entrada.

Se registros de entrada forem dependentes entre si, eles podem ser processados como uma transação. Uma tarefa do batch pode bloquear um máximo de 500.000.000 registros. Este limite pode ser reduzido com Query Options File (QAQQINI). Utilize o parâmetro QRYOPTLIB do comando Change Query Attributes (CHGQRYA) para especificar Query Options File para um job utilizar. Utilize o valor COMMITMENT\_CONTROL\_LOCK\_LEVEL em Query Options File como limite de bloqueio para o job.

Qualquer ciclo de confirmação que exceder 2.000 bloqueios provavelmente reduzirá consideravelmente o desempenho do sistema. Do contrário, existirão as mesmas considerações de bloqueio que aquelas dos aplicativos interativos, mas os registros de período de tempo que são bloqueados em uma aplicação em batch talvez sejam menos importantes do que em aplicativos interativos.

### Conceitos relacionados

“Objeto de Notificação de Confirmação” na página 52

Um *objeto de notificação* é uma fila de mensagens, área de dados ou arquivo de banco de dados contendo informações que identificam a última transação concluída com sucesso para uma determinada definição de confirmação, se ela não foi finalizada normalmente.

“Gerenciando o Tamanho da Transação” na página 75

Uma outra maneira de minimizar os bloqueios do registro é gerenciar o tamanho da transação.

### Referências relacionadas

Comando Change Query Attributes (CHGQRYA)

## Controle de Confirmação de Duas Fases

O controle de confirmação de duas fases garante que recursos confirmáveis em vários sistemas permaneçam sincronizados.

O sistema operacional i5/OS suporta confirmação de duas fases, em conformidade com a arquitetura SNA LU 6.2. Para obter informações adicionais detalhadas sobre os protocolos internos utilizados pelo sistema para confirmação de duas fases, consulte o *SNA Transaction Programmer's Reference for LU Type 6.2, GC30-3084-05*. Todos os releases suportados do sistema operacional i5/OS suportam protocolos Presumed Nothing do SNA LU 6.2 e os protocolos Presumed Abort do SNA LU 6.2.

A confirmação de duas fases também é suportada utilizando o TCP/IP como um protocolo DUW (Distributed Unit of Work) DRDA (*Distributed Relational Database Architecture*). Para utilizar as conexões DUW do TCP/IP, todos os sistemas (o solicitador do aplicativo e o servidor do aplicativo) deverão estar com V5R1M0 ou mais recentes. Para obter informações adicionais sobre o DRDA, consulte o Open Group Technical Standard, *DRDA V2 Vol. 1: Distributed Relational Database Architecture* no Web site da Open Group.

Sob a confirmação de duas fases, o sistema executa a operação confirmar em duas ondas:

- Durante a *onda de preparação*, um gerenciador de recurso emite um pedido de confirmação para o seu gerenciador de transações. O gerenciador da transação informa a qualquer outro recurso que gerencia e outros gerenciadores de transação que a transação está preparada para ser confirmada. Todos os gerenciadores de recurso devem responder que estão prontos para confirmar. Isto se chama *votar*.
- Durante a *onda confirmada*, o gerenciador da transação que inicia o pedido de confirmação decide o que fazer, com base no resultado da onda de preparação. Se a onda de preparação for concluída com sucesso e todos os participantes votarem prontos, o gerenciador da transação orientará todos os recursos que gerencia e os outros gerenciadores da transação para confirmar a transação. Se a onda de preparação não for concluída com sucesso, todos os gerenciadores da transação e gerenciadores de recurso serão instruídos para reverter a transação.

## Operações de Confirmação e Rollback com Recursos Remotos

Quando os recursos remotos estão sob o controle de confirmação, o iniciador envia um pedido de confirmação para todos os agentes remotos. O pedido é enviado por toda a rede do programa de transação. Cada agente responde com os resultados da operação confirmar.

Se ocorrerem erros durante a onda de preparação, o iniciador enviará um pedido de rollback a todos os agentes. Se ocorrerem erros durante a onda confirmada, o sistema tentará trazer tantas localizações quanto possíveis para o status confirmado. Estas tentativas poderão resultar em um estado heurístico misto. Consulte Estados da Transação para o Controle de Confirmação de Duas Fases para obter informações adicionais sobre os estados possíveis.

Todos os erros são devolvidos ao iniciador, onde são indicados para o usuário. Se um diário padrão foi especificado no comando Start Commitment Control (STRCMTCTL), entradas C LW serão gravadas. Se ocorrerem erros, elas serão gravadas, mesmo que OMTJRNE(\*LUWID) tenha sido especificado. Você pode utilizar essas entradas, junto com as mensagens de erro e as informações de status para a definição de confirmação, para tentar sincronizar os recursos confirmáveis manualmente.

Quando os recursos remotos estão sob o controle de confirmação, o iniciador envia um pedido de rollback para todos os agentes remotos. O pedido é enviado por toda a rede do programa de transação. Cada agente responde com os resultados da operação de rollback.

### Conceitos relacionados

 [Web Site da The Open Group](#)

### Referências relacionadas

Comando Start Commitment Control (STRCMTCTL)

## Funções no Processamento de Confirmação

Se a confirmação de uma transação envolver mais de um gerenciador de recurso, cada um exercerá uma função na transação. Um gerenciador de recurso é responsável pela confirmação ou rollback das alterações feitas durante a transação.

Os gerenciadores de recurso por tipo de recurso são os seguintes:

### ARQUIVO

Gerenciador de bancos de dados

**DDM** Gerenciador de bancos de dados

**DDL** Gerenciador de bancos de dados

### DRDA

Programa de transação de comunicações

**LU62** Programa de transação de comunicações

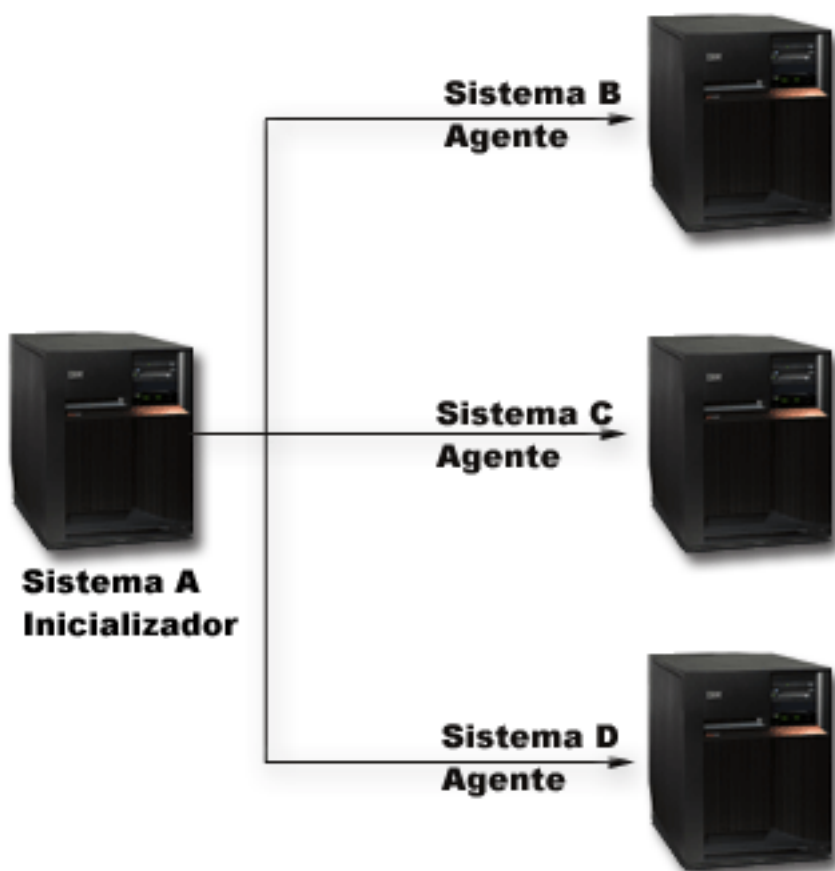
API Programa de saída da API

As figuras a seguir mostram as funções básicas em uma transação. A estrutura mostrada nas figuras chama-se *rede do programa de transação*. A estrutura pode estar em uma árvore de nível único e de níveis múltiplos.

### Funções no Processamento de Confirmação de Duas Fases: Árvore de Único Nível

Quando um aplicativo no Sistema A emite um pedido de confirmação, o gerenciador de recurso neste sistema torna-se o *iniciador*. Para a unidade de trabalho distribuída do DRDA (Distributed Relational Database Architecture) sobre TCP/IP, o iniciador é chamado de *coordenador*.

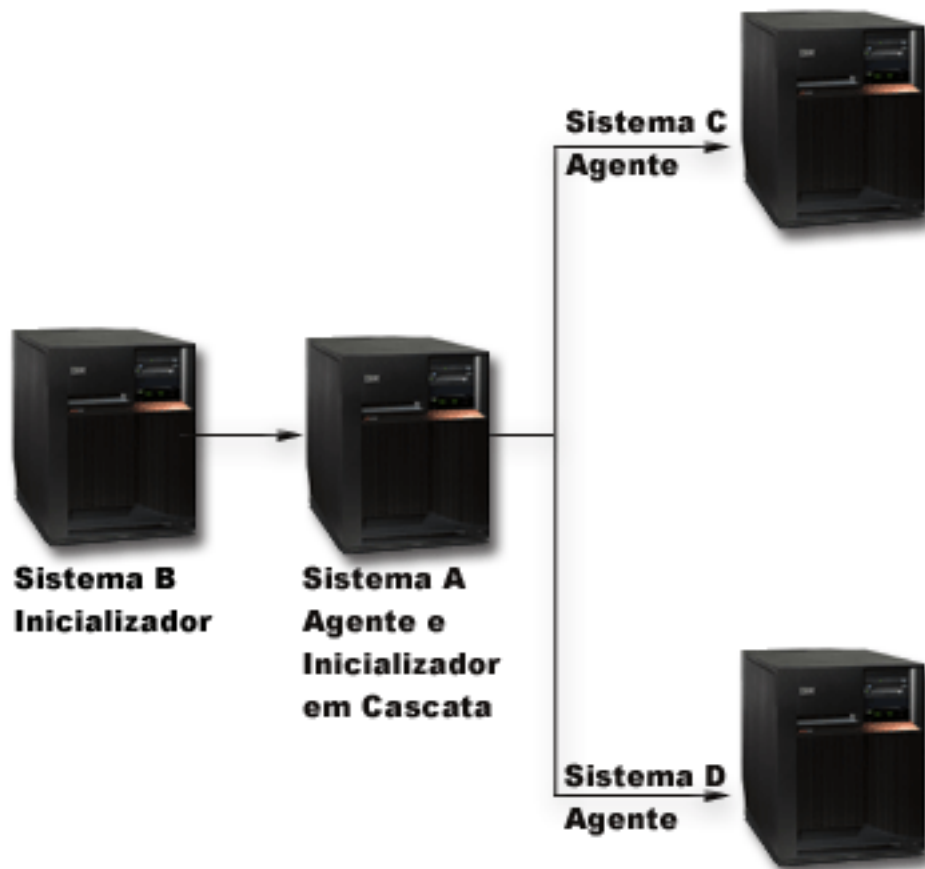
Os gerenciadores de recurso dos outros três sistemas (B, C e D) tornam-se *agentes* dessa transação. Para a unidade de trabalho distribuída DRDA através de TCP/IP, os agentes são às vezes chamados de *participantes*.



### Funções no Processamento de Confirmação de Duas Fases: Árvore de Níveis Múltiplos

Se o aplicativo estiver utilizando as comunicações APPC para executar a confirmação de duas fases, a relação entre os sistemas poderá mudar de uma transação para a seguinte. A figura a seguir mostra os mesmos sistemas quando um aplicativo no Sistema B emite o pedido de confirmação. Essa configuração é uma árvore de níveis múltiplos.

As funções nesta figura não se aplicam à unidade de trabalho distribuída DRDA através de TCP/IP porque as árvores de transações de níveis múltiplos não são suportadas.



A rede do programa de transação tem outro nível porque o Sistema B não está se comunicando diretamente com os Sistemas C e D. O gerenciador de recursos no Sistema A tem agora as funções do agente e do inicializador em cascata.

Para aprimorar o desempenho das transações de duas fases da LU 6.2, o inicializador pode designar a função do último agente a um dos agentes. O último agente não participa da onda de preparação. Na onda confirmada, o último agente confirma primeiro. Se o último agente não se confirma com sucesso, o iniciador orienta os outros agentes para reverterem.

Para a unidade de trabalho distribuída DRDA através de TCP/IP, o coordenador pode designar a função de servidor de ressincronização a um participante. O servidor de ressincronização é responsável por ressincronizar os outros participantes caso haja uma falha na comunicação com o coordenador ou este tenha um defeito do sistema.

#### **Conceitos relacionados**

“Definição de Confirmação para Confirmação de Duas Fases: Permitir Votar Somente Leitura” na página 33

Geralmente, um gerenciador de transação participa de ambas as fases do processamento de confirmação. Para melhorar o desempenho do processamento de confirmação, você pode configurar algumas ou todas as localizações em uma transação para permitir que o gerenciador de transação eleja como leitura.

#### **Estados da Transação para o Controle de Confirmação de Duas Fases**

Uma definição de confirmação é estabelecida em cada localização que faz parte da rede do programa de transação. Para cada uma, o sistema mantém o estado de sua transação atual e da anterior.

O sistema utiliza o estado para decidir entre confirmação ou rollback se uma transação for interrompida por um defeito do sistema ou uma falha na comunicação. Se vários locais participarem de uma transação, os estados das transações em cada local poderão ser comparados para determinar a ação correta (confirmação ou rollback). Este processo de comunicação entre as localizações para determinar a ação correta chama-se *ressincronização*.

A tabela a seguir mostra estes itens:

- Os estados básicos que podem ocorrer durante uma transação.
- Os estados adicionais que podem ocorrer.
- Se um estado irá solicitar ressincronização caso a transação seja interrompida por uma falha de comunicação ou do sistema. Os valores possíveis são os seguintes:

**Não necessário**

Cada localização pode tomar a decisão correta de forma independente.

**Pode ser necessário**

Cada local pode tomar a decisão correta, mas pode ser necessário informar o inicializador da decisão.

**Obrigatório**

O estado de cada localização deve ser determinado antes da decisão correta ser tomada.

- Ação tomada por uma falha de comunicação ou do sistema.

Nome do Estado	Descrição	Ressincronização se a Transação for Interrompida	Ação Tomada por uma Falha de Comunicação ou do Sistema
<b>Estados básicos durante o processamento de confirmação de duas fases:</b>			
Reinicializar (RST)	Desde o tempo limite de confirmação até que um programa emita um pedido para confirmar ou reverter.	Não necessário.	As alterações pendentes são revertidas.
Preparação em Andamento (PIP)	O iniciador iniciou a onda de preparação. Todas as localizações ainda não votaram.	Pode ser necessário.	As alterações pendentes são revertidas.
Preparado (PRP)	Esta localização e todas as outras mais abaixo na rede do programa de transação votaram pela confirmação. Essa localização ainda não recebeu a notificação do iniciador para confirmar.	Solicitado.	Incerto. Depende dos resultados do processo de ressincronização.
Confirmação em Andamento (CIP)	Todas as localizações votaram pela confirmação. O iniciador começou a onda confirmada.	Solicitado.	As alterações pendentes são confirmadas. A ressincronização é executada para garantir que todas as localizações foram confirmadas. Se um rollback heurístico for informado por outra localização, será comunicado um erro.
Confirmado (CMT)	Todos os agentes foram confirmados e retornaram uma resposta para este nó.	Pode ser necessário.	Não existe.
<b>Estados adicionais durante o processamento de confirmação de duas fases:</b>			

Nome do Estado	Descrição	Ressincronização se a Transação for Interrompida	Ação Tomada por uma Falha de Comunicação ou do Sistema
Último Agente Pendente (LAP)	Se um último agente foi selecionado, esse estado ocorrerá no iniciador entre o estado PIP e o estado CIP. O iniciador instruiu o último agente a confirmar e ainda não recebeu uma resposta.	Solicitado.	Incerto. Depende dos resultados do processo de ressincronização.
Voto (VRO)	Este agente respondeu à onda de preparação indicando que não tem alterações pendentes. Se o estado do voto for permitido, este agente não será incluído na onda confirmada.	Pode ser necessário.	Não existe.
Rollback Solicitado (RBR)	<p>Ocorreu um dos seguintes eventos:</p> <ul style="list-style-type: none"> <li>• Um agente emitiu um pedido de rollback antes da operação confirmar.</li> <li>• Ocorreu uma falha na transação.</li> <li>• A API QTNRBRQD foi usada para colocar a transação em um estado solicitado de rollback.</li> </ul> <p>O programa de transação não tem permissão de executar nenhuma alteração adicional no controle de confirmação.</p>	Pode ser necessário.	As alterações pendentes são revertidas.
<b>Condições que ocorrem devido a ações ou erros do operador:</b>			
Rollback forçado	Esta localização e todas as outras mais adiante na rede do programa de transação, exceto o último agente, foram revertidas por intervenção do operador.	Pode ser necessário.	As alterações pendentes já foram revertidas.
Confirmação forçada	Esta localização e todas as outras mais adiante na rede do programa de transação, exceto o último agente, foram confirmadas por intervenção do operador.	Pode ser necessário.	As alterações pendentes já foram confirmadas.

Nome do Estado	Descrição	Ressincronização se a Transação for Interrompida	Ação Tomada por uma Falha de Comunicação ou do Sistema
Heurística Mista (HRM)	Alguns gerenciadores de recurso foram confirmados. Alguns foram revertidos. A intervenção do operador foi utilizada ou ocorreu um erro do sistema. A heurística mista não aparece como um status na exibição da definição de confirmação. As mensagens de notificação são enviadas ao operador.	Pode ser necessário.	O operador deve executar uma operação de restauração em todas as localizações participantes para colocar o banco de dados em um estado consistente.

### Conceitos relacionados

“Definição de Confirmação para Confirmação de Duas Fases: Permitir Votar Somente Leitura”

Geralmente, um gerenciador de transação participa de ambas as fases do processamento de confirmação. Para melhorar o desempenho do processamento de confirmação, você pode configurar algumas ou todas as localizações em uma transação para permitir que o gerenciador de transação eleja como leitura.

“Definição de Confirmação para Confirmação de Duas Fases: Não Aguardar pelo Resultado” na página 36

Quando ocorre um defeito do sistema ou uma falha na comunicação durante uma operação de confirmação de modo que a ressincronização é requerida, o padrão é aguardar o término da ressincronização antes da conclusão da operação de confirmação.

“Iniciando o Controle de Confirmação” na página 51

Para iniciar o controle de confirmação, utilize o comando Start Commitment Control (STRCMTCTL).

“Recuperação do Controle de Confirmação Durante Carregamento Inicial do Programa Após Término Anormal” na página 67

Ao executar um IPL (Carregamento Inicial do Programa) após uma finalização anormal do sistema, o sistema tenta recuperar todas as definições de confirmação que estavam ativas quando o sistema foi finalizado.

“Erros do Controle de Confirmação” na página 106

Ao utilizar o controle de confirmação, é importante compreender quais condições causam erros e quais não causam.

### Definições de Confirmação para Controle de Confirmação de Duas Fases

Para alterar as opções de confirmação para a sua transação depois de iniciar o controle de confirmação, utilize a API Change Commitment Options (QTNCHGCO).

Dependendo de seu ambiente e de seus aplicativos, a alteração das opções de confirmação pode melhorar o desempenho do seu sistema.

**Nota:** Se você estiver utilizando uma unidade de trabalho distribuída DRDA através da conexão TCP/IP, a única opção aplicável será Permitir Voto de Leitura.

#### Referências relacionadas

API Change Commitment Options (QTNCHGCO)

#### Definição de Confirmação para Confirmação de Duas Fases: Permitir Votar Somente Leitura:

Geralmente, um gerenciador de transação participa de ambas as fases do processamento de confirmação. Para melhorar o desempenho do processamento de confirmação, você pode configurar algumas ou todas as localizações em uma transação para permitir que o gerenciador de transação eleja como leitura.

Se a localização não possuir nenhuma alteração confirmável durante uma transação, o gerenciador de transação vota durante a onda de preparação. A localização não participa da onda confirmada. Isto melhora o desempenho geral, pois os fluxos de comunicação que geralmente ocorrem durante a onda confirmada são eliminados durante transações nas quais nenhuma atualização é feita em uma ou mais localizações remotas.

Depois de iniciar o controle de confirmação, você pode utilizar a API Change Commitment Options (QTNCHGCO) para alterar a opção Voto de Leitura Permitido para S. Você talvez queira fazer isto quando as seguintes condições forem verdadeiras:

- Um ou mais sistemas remotos geralmente não possuem quaisquer alterações confirmáveis para uma transação.
- Uma transação não depende de onde o cursor do arquivo (próximo registro) foi definido pela transação anterior. Quando uma localização vota, o aplicativo nunca é notificado quando a transação é revertida. A localização confirmou todas as operações de leitura para os arquivos do banco de dados e, assim, moveu a posição do cursor. Geralmente, a posição do cursor do arquivo é importante somente se você efetua o processamento seqüencial.

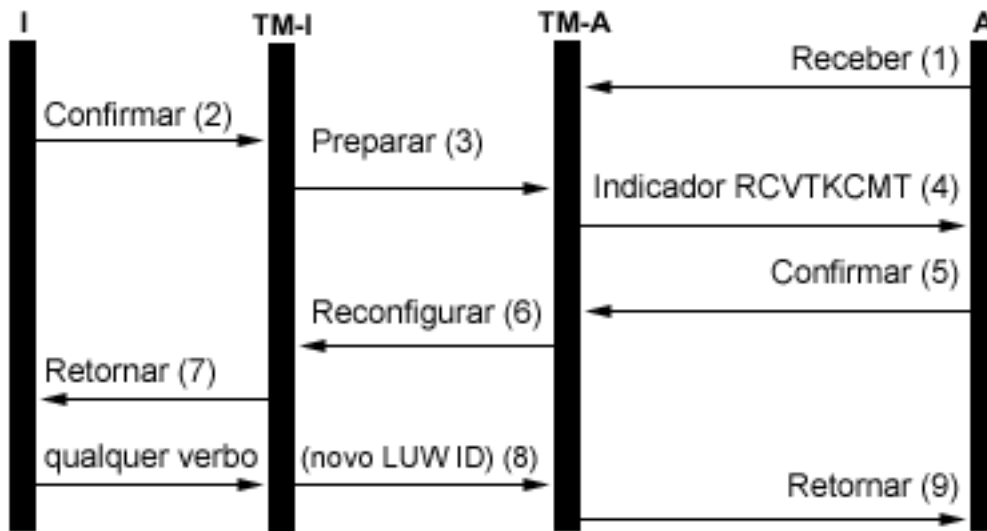
Se sua definição de confirmação está configurada para permitir votar, o aplicativo espera pelo próximo fluxo de mensagem de outra localização.

A opção Permitido Votar é destinada a aplicativos que são de natureza cliente/servidor. Se o objetivo do programa A é apenas satisfazer pedidos do programa I, não fazer qualquer trabalho independente, é apropriado permitir a opção Votar para o programa A.

### **Fluxo de Processamento de Confirmação sem Otimização do Último Agente quando o Agente Vota no Modo de Leitura**

A figura a seguir mostra o fluxo de mensagens entre os programas aplicativos e os gerenciadores de transação, quando um programa aplicativo emite uma instrução de confirmação sem otimização do último agente quando o agente vota somente leitura. O programa aplicativo inicializador e os programas aplicativos agentes não ficam cientes do processamento de confirmação de duas fases. Os números entre parênteses () na figura correspondem aos itens numerados na descrição que segue.





### Legenda

- I** = Iniciador (Aplicativo que inicia a confirmação do pedido)
- TM-I** = Gerenciador de transação para o iniciador
- A** = Agente (Aplicativo que recebe a confirmação do pedido)
- TM-A** = Gerenciador de transação para o agente

A lista a seguir é uma descrição dos eventos para o processamento normal sem a otimização do último agente quando o agente vota no modo de leitura. Isto descreve um fluxo básico. A seqüência de eventos pode tornar-se mais complexa quando a rede do programa de transação tem vários níveis ou quando ocorrem erros.

1. O programa aplicativo A recebe um pedido para indicar que está pronto para receber um pedido do programa I.
2. O aplicativo iniciador (I) emite uma instrução de confirmação.
3. O gerenciador da transação do iniciador (TM-I) assume a função do iniciador nessa transação. Ele inicia a onda de preparação enviando uma mensagem de preparação para todas as outras localizações que estão participando da transação.
4. Os gerenciadores de transação de cada localização diferente assumem a função do agente (TM-A). O programa aplicativo A é notificado pelo TM-A que foi recebido um pedido para confirmação. Para arquivos ICF, a notificação está no formato do indicador ICF Receive Take Commit (RCVTKCMT) em que está sendo definido.
5. O programa aplicativo A responde emitindo uma instrução de confirmação (ou uma instrução de rollback). Este é o voto do programa aplicativo.
6. Se o programa aplicativo A utilizou a API Change Commitment Options (QTNCHGCO) para configurar a opção de confirmação Voto de Leitura Permitido como S e nenhuma alteração foi feita no agente durante a transação, o agente (TM-A) responderá ao inicializador (TM-I) com uma mensagem de reconfiguração. Não há nenhuma onda confirmada para o agente.
7. Um retorno é enviado ao programa aplicativo (A) para indicar que a transação está completa no agente TM-A.
8. A próxima vez que o iniciador (TM-I) emite alguma mensagem para o agente (TM-A), um fluxo de dados ou uma instrução de confirmação, o TM-I faz com que seu atual ID de transação seja enviado com a mensagem. O motivo para isso é que um novo ID de transação pode ter sido gerado no TM-I caso tenha ocorrido um defeito na comunicação entre o TM-I e outro sistema durante a operação de confirmação.

9. Um retorno é enviado ao programa aplicativo (A) para indicar que a transação está completa no agente TM-A. O retorno é atrasado até depois da próxima mensagem ser recebida pois um novo ID de transação deve ser recebido do TM-I antes que a próxima transação possa ser iniciada pelo aplicativo A.

#### Conceitos relacionados

“Funções no Processamento de Confirmação” na página 28

Se a confirmação de uma transação envolver mais de um gerenciador de recurso, cada um exercerá uma função na transação. Um gerenciador de recurso é responsável pela confirmação ou rollback das alterações feitas durante a transação.

“Estados da Transação para o Controle de Confirmação de Duas Fases” na página 30

Uma definição de confirmação é estabelecida em cada localização que faz parte da rede do programa de transação. Para cada uma, o sistema mantém o estado de sua transação atual e da anterior.

“Otimizando o Desempenho para Controle de Confirmação” na página 70

O uso de controle de confirmação requer recursos que podem afetar o desempenho do sistema. Vários fatores afetam o desempenho do sistema relativo ao controle de confirmação.

#### Referências relacionadas

API Change Commitment Options (QTNCHGCO)

#### Definição de Confirmação para Confirmação de Duas Fases: Não Aguardar pelo Resultado:

Quando ocorre um defeito do sistema ou uma falha na comunicação durante uma operação de confirmação de modo que a ressincronização é requerida, o padrão é aguardar o término da ressincronização antes da conclusão da operação de confirmação.

**Nota:** A opção Não aguardar pelo resultado não se aplicará se você estiver utilizando uma unidade de trabalho distribuída DRDA (Distributed Relational Database Architecture) sobre a conexão TCP/IP. A unidade de trabalho distribuída DRDA através das conexões TCP/IP não aguardam pelo resultado.

Considere alterar este comportamento se as condições a seguir forem verdadeiras:

- Os aplicativos que participam são independentes uns dos outros.
- A lógica do programa não precisa dos resultados das transações anteriores para assegurar que os arquivos de banco de dados fiquem sincronizados.

Depois de iniciar o controle de confirmação, você pode utilizar a API Change Commitment Options (QTNCHGCO) para especificar que a definição de confirmação não aguardará pelo resultado da ressincronização. Se você especificar N (Não) para a opção Aguardar pelo resultado, o sistema utilizará um job do servidor do bancos de dados (QDBSRVnn) para tratar a ressincronização de forma assíncrona.

**Nota:** Esses jobs do servidor do bancos de dados são iniciados durante o processo de IPL (Initial Program Load). Se você alterar as opções para o controle de confirmação, isto não terá efeito no número de jobs que o sistema inicia.

Este tópico refere-se somente a dois valores da opção Aguardar pelo resultado resolvida, S (Sim) e N (Não). Na verdade, existem mais dois valores que você pode especificar, L (Sim ou Herdar do Iniciador) e U (Não ou Herdar do Iniciador). Quando você usa esses valores, o valor real usado durante cada operação confirmar é processado para Sim ou Não pelo sistema. O tópico da API QTNCHGCO (Change Commitment Options) apresenta mais detalhes sobre esses valores.

**Nota:** O valor do iniciador somente poderá ser herdado apenas por um agente se o iniciador e o agente suportarem a interrupção presumida.

A opção WFO (Aguardar pelo Resultado) não afeta o processamento normal, sem erros de confirmação. Se ocorrer um erro, a opção WFO determinará se o aplicativo aguardará ou não pela ressincronização, com as seguintes condições:

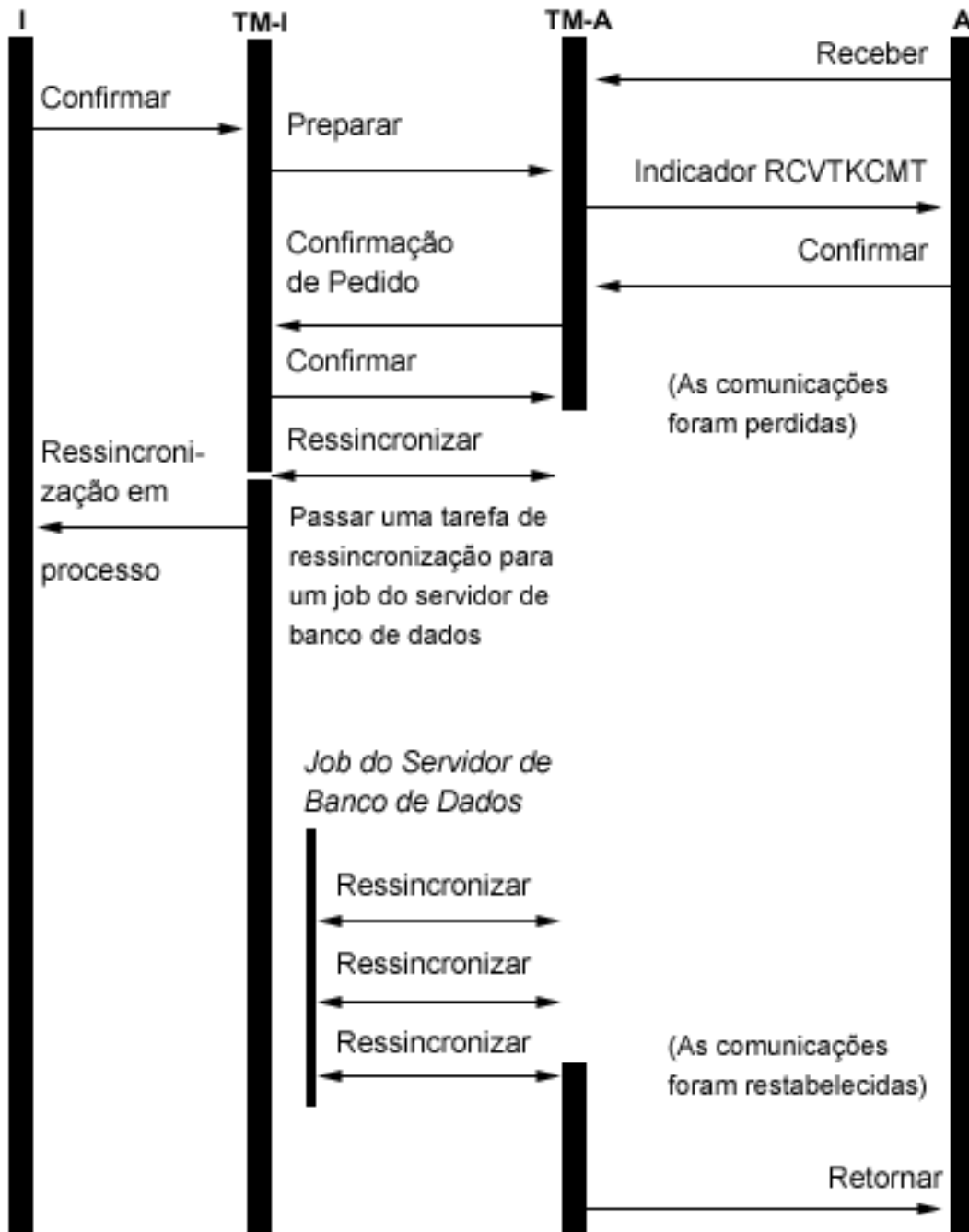
- Se a opção WFO processada for S (Sim), o aplicativo aguardará pelo resultado da ressincronização.
- Se a opção WFO processada for N (Não) e ocorrer uma falha de comunicação durante a onda de preparação ou rollback de uma localização que suporte protocolos de interrupções presumidas, nenhuma ressincronização será executada e a definição de confirmação será revertida.
- Se a definição de confirmação estiver duvidosa (o estado da transação estiver preparado ou o Último Agente Pendente), o aplicativo aguardará pelo resultado da ressincronização independente do valor WFO processado.
- Se a opção WFO processada for N e nenhuma das condições dois ou três for verdadeira, o sistema tentará ressincronizar uma vez. Se não for bem-sucedido, o sistema enviará a mensagem de STATUS CPF83E6 ao aplicativo para indicar que a ressincronização está em andamento.

Como CPF83E6 é uma mensagem de STATUS, ela terá efeito somente se o aplicativo a estiver monitorando. Normalmente, seu aplicativo pode tratar essa mensagem como sendo informativa. Os sistemas que estão participando da transação tentam ressincronizá-la até que a falha seja corrigida. Essas tentativas de ressincronização seguintes são executadas nos jobs do servidor do bancos de dados. Se uma tentativa seguinte falhar, a mensagem CPI83D0 será enviada para QSYSOPR.

#### **Valor de Aguardar pelo Resultado: Sim**

Na figura seguinte, a definição de confirmação do iniciador (I) usa o valor padrão de S (Sim) para a opção Aguardar pelo resultado. Quando as comunicações entre TM-I e TM-A se perdem, o aplicativo A e o aplicativo I aguardam até que a transação seja ressincronizada.





Quando uma conexão é feita com um banco de dados relacional remoto e as conversações não protegidas já foram iniciadas, o sistema muda implicitamente o valor de Aguardar pelo resultado para N. O motivo é que o desempenho das operações confirmar melhora quando o valor Aguardar pelo resultado é N e o sistema remoto suporta a interrupção presumida. Esta alteração implícita do valor Aguardar pelo Resultado é desempenhada para aplicativos DRDA e DDM. Os aplicativos APPC utilizarão o valor padrão Aguardar pelo Resultado de S a menos que chamem a API QTNCHGCO para alterá-lo.

#### Conceitos relacionados

“Estados da Transação para o Controle de Confirmação de Duas Fases” na página 30

Uma definição de confirmação é estabelecida em cada localização que faz parte da rede do programa de transação. Para cada uma, o sistema mantém o estado de sua transação atual e da anterior.

“Erros do Controle de Confirmação” na página 106

Ao utilizar o controle de confirmação, é importante compreender quais condições causam erros e quais não causam.

#### **Referências relacionadas**

API Change Commitment Options (QTNCHGCO)

#### **Definição de Confirmação para Confirmação de Duas Fases: Indicar OK para Omitir:**

Normalmente, o gerenciador da transação em cada localização na rede do programa de transação participa de toda operação de confirmação ou rollback. Para melhorar o desempenho, você pode configurar algumas localizações ou todas elas em uma transação para permitir que o gerenciador da transação indique OK para Omitir.

**Nota:** A opção Indicar OK para Omitir não se aplicará se você estiver utilizando uma unidade de trabalho distribuída DRDA através da conexão TCP/IP.

Se nenhum fluxo de comunicação for enviado à localização durante uma transação, ela omitirá quando for executada uma operação de confirmação ou rollback. Isto melhora o desempenho geral porque o fluxo de comunicação que normalmente ocorre durante a confirmação ou o rollback é eliminado durante as transações, nas quais nenhum dado é enviado a uma ou mais localizações remotas.

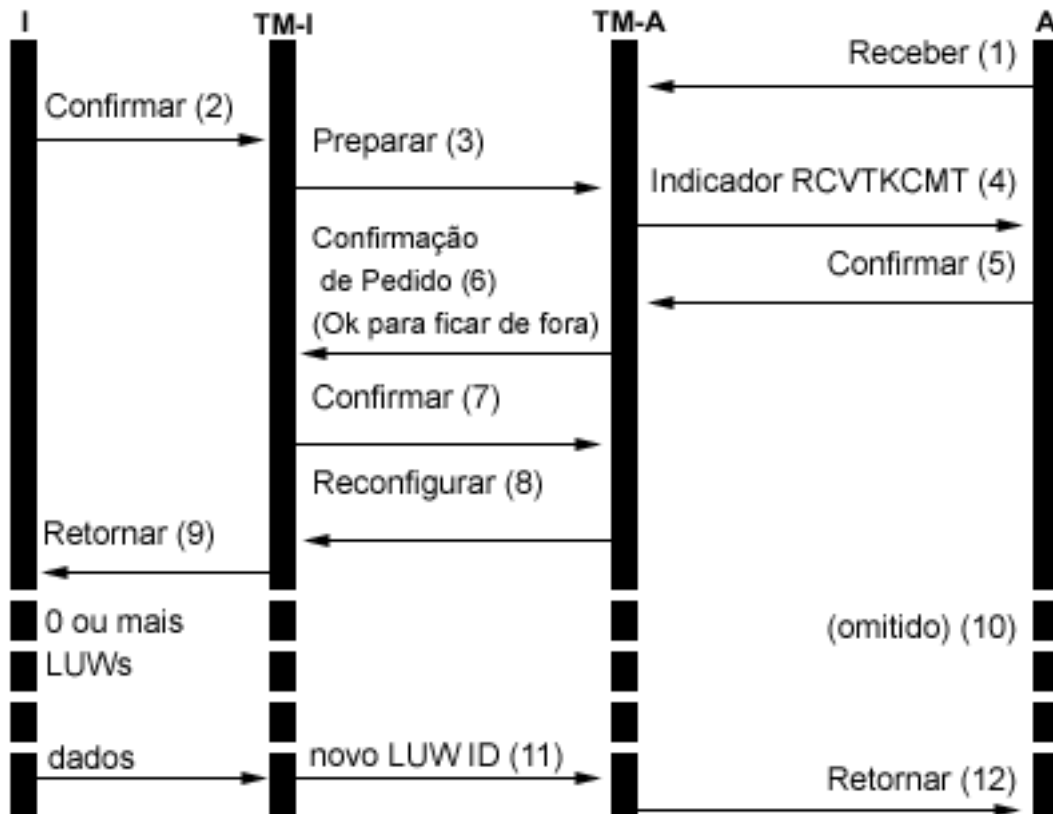
Depois de iniciar o controle de confirmação, você pode utilizar a API Change Commitment Options (QTNCHGCO) para alterar a opção OK para Omitir para S (Sim). Isso poderá ser feito se um ou mais sistemas remotos geralmente não estiverem envolvidos em uma transação.

Se a definição de confirmação estiver configurada para indicar que OK omite, o aplicativo aguarda pelo fluxo da mensagem seguinte de outra localização.

A opção OK para Omitir destina-se a aplicativos que sejam de natureza cliente/servidor. Se o único propósito do programa A for atender a pedidos do programa I e não executar nenhum trabalho independente, então será apropriado permitir a opção OK para Omitir para o programa A.

#### **Fluxo de Processamento de Confirmação sem Otimização do Último Agente quando o Agente Vota em OK para Omitir**

A figura a seguir mostra o fluxo de mensagens entre programas aplicativos e gerenciadores de transação quando um programa aplicativo emite uma instrução de confirmação sem a otimização do último agente e o agente indica OK para Omitir. O programa aplicativo iniciador e os programas aplicativos do agente não sabem da existência do processamento de confirmação de duas fases. Os números entre parênteses () na figura correspondem aos itens numerados na descrição que segue.



**Legenda**

- I = Iniciador (Aplicativo que inicia a confirmação do pedido)
- TM-I = Gerenciador de transação para o iniciador
- A = Agente (Aplicativo que recebe a confirmação do pedido)
- TM-A = Gerenciador de transação para o agente

Aqui está uma descrição dos eventos para processamento normal sem a otimização do último agente quando o agente vota em OK para omitir. Isto descreve um fluxo básico. A seqüência de eventos pode tornar-se mais complexa quando a rede do programa de transação tem vários níveis ou quando ocorrem erros.

1. O programa aplicativo A recebe um pedido para indicar que está pronto para receber um pedido do programa I.
2. O aplicativo iniciador (I) emite uma instrução de confirmação.
3. O gerenciador da transação do iniciador (TM-I) assume a função do iniciador nessa transação. Ele inicia a onda de preparação enviando uma mensagem de preparação para todas as outras localizações que estão participando da transação.
4. Os gerenciadores de transação de cada localização diferente assumem a função do agente (TM-A). O programa aplicativo A é notificado pelo TM-A que foi recebido um pedido para confirmação. Para arquivos ICF, a notificação está no formato do indicador ICF Receive Take Commit (RCVTKCMT) em que está sendo definido.
5. O programa aplicativo A responde emitindo uma instrução de confirmação (ou uma instrução de rollback). Este é o voto do programa aplicativo.
6. Se o programa aplicativo A utiliza a API Change Commitment Options (QTNCHGCO) para definir o OK para Omitir a opção de confirmação para S, será enviado um indicador quando o agente (TM-A) responder ao iniciador (TM-I) com uma mensagem de registro do pedido.

**Nota:** Qualquer alteração feita na opção OK para Omitir a confirmação não será efetivada até a próxima operação de confirmação bem-sucedida.

7. Quando o iniciador (TM-I) receber todos os votos, o TM-I enviará uma mensagem de confirmação. Isto iniciará a onda confirmada.
8. Cada agente (TM-A) confirma e responde com uma mensagem de reinicialização.
9. Um retorno é enviado ao programa aplicativo (I) para indicar que a transação está concluída no iniciador.
10. Qualquer número de transações pode ocorrer no TM-I, nenhuma delas requer alterações no TM-A ou dados do TM-A. TM-A não está incluído nessas transações.
11. Na próxima vez que o iniciador (TM-I) emitir uma mensagem ao agente (A), um novo ID de transação será enviado com a mensagem. Se o iniciador executar qualquer operação de confirmação ou rollback antes de enviar uma mensagem ao agente, nenhuma mensagem será enviada ao agente durante essas operações (o agente é 'omitido' dessas confirmações ou rollbacks). Como podem ter ocorrido a confirmação ou o rollback de uma ou mais transações no inicializador enquanto o agente foi omitido, o inicializador deve comunicar seu ID de transação atual quando a mensagem seguinte é enviada ao agente.
12. Um retorno é enviado ao programa aplicativo (A) para indicar que a confirmação atual está concluída e está participando na transação atual.

#### **Conceitos relacionados**

“Otimizando o Desempenho para Controle de Confirmação” na página 70

O uso de controle de confirmação requer recursos que podem afetar o desempenho do sistema. Vários fatores afetam o desempenho do sistema relativo ao controle de confirmação.

#### **Definição de Confirmação para Confirmação de Duas Fases: Não Selecionar um Último Agente:**

Por padrão, o gerenciador da transação para o iniciador é livre para selecionar qualquer agente como o último durante uma operação confirmar.

**Nota:** A opção Não selecionar último agente não se aplicará se você estiver utilizando uma unidade de trabalho distribuída DRDA através da conexão TCP/IP.

No caso de uma árvore de vários níveis, qualquer agente selecionado como último por seu iniciador também terá liberdade de selecionar um último agente próprio. O desempenho melhora quando um último agente é selecionado durante a operação confirmar, porque dois fluxos de comunicações são eliminados entre um iniciador e seu último agente (a fase de preparação é eliminada nesses sistemas).

Entretanto, quando o iniciador envia o registro do pedido para seu último agente, ele deve aguardar até que receba o voto do último agente antes de poder continuar. Isso é independente do valor Aguardar por resultado da definição de confirmação. Durante o processamento de confirmação normal sem erros, isso não é um problema. Mas se um erro ocorre durante esta janela, o iniciador não consegue continuar até que a ressinchronização seja concluída. Se o aplicativo do iniciador estiver manipulando pedidos de um usuário em um terminal, isso poderá ser considerado uma capacidade de uso.

Leve em conta se o desempenho melhorado durante as operações normais de confirmação é mais importante que o impacto sobre a capacidade de uso quando tal erro ocorre. Observe que se o erro ocorrer antes de o registro do pedido ser enviado ao último agente, o LUW reverterá imediatamente e o iniciador não esperará. Por essa razão, a janela de quando um erro provoca a espera do iniciador é muito pequena; por isso, tal erro é raro.

Se você decidir que o impacto de funcionalidade não supera o desempenho melhorado, você poderá alterar as definições de confirmação para não selecionar um último agente. Depois de iniciar o controle de confirmação, você poderá utilizar a API Change Commitment Options (QTNCHGCO) para alterar a opção Último agente permitido para N.

#### **Referências relacionadas**



### **Efeito do Voto Confiável no Fluxo de Processamento de Confirmação:**

Voto confiável é uma otimização que melhora o desempenho retornando mais cedo para o aplicativo iniciador após uma operação confirmar e eliminando uma mensagem durante uma operação confirmar.

Não existe nenhuma otimização de voto confiável explícito para a unidade de trabalho distribuída DRDA (Distributed Relational Database Architecture) sobre TCP/IP. No entanto, o sistema operacional i5/OS nunca solicita uma configuração de reconfiguração (esquecimento) para conexões TCP/IP. Portanto, uma reconfiguração (esquecimento) é sempre implícita para conexões TCP/IP.

Depois de iniciar o controle de confirmação, você pode utilizar a API Change Commitment Options (QTNCHGCO) para alterar a opção Aceitar Voto Confiável para Y.

O voto confiável pode ser tido como uma promessa de um agente para seu iniciador que não serão tomadas decisões heurísticas no agente se ocorrerem falhas em comunicações enquanto o agente estiver incerto. Um agente que está utilizando a otimização do voto confiável envia um indicador para o iniciador durante a onda de preparação de confirmação. Se o iniciador também estiver utilizando a otimização do voto confiável, ele enviará um indicador para o agente informando que não é necessária a reinicialização em resposta à mensagem de confirmação. Isto elimina a mensagem de reinicialização e permite que o gerenciador da transação retorne ao aplicativo no iniciador tão logo a mensagem de confirmação seja enviada.

Considere o uso da otimização do voto confiável se as seguintes condições forem verdadeiras:

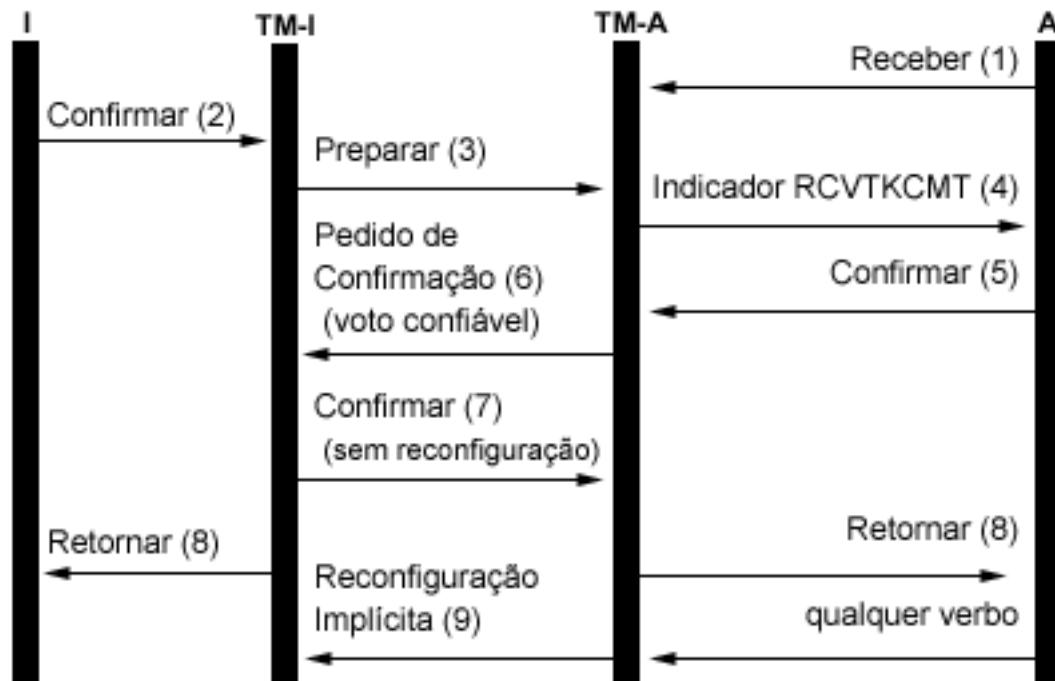
- É improvável que uma decisão de heurística seja feita em um agente incerto no caso de uma falha de sistema ou comunicação, a menos que a falha não possa ser corrigida.
- A lógica do programa não precisa dos resultados das transações anteriores para assegurar que os arquivos de banco de dados fiquem sincronizados.

A otimização de voto confiável será utilizada pelo sistema operacional i5/OS apenas se todas as seguintes condições forem verdadeiras:

- As localizações do iniciador e do agente suportam o nível de interrupção presumido do controle de confirmação.
- A localização do iniciador aceita a indicação de voto confiável do agente. Em inicializadores do i5/OS, isso depende do valor das duas opções de confirmação:
  - O valor da opção de confirmação Aguardar pelo resultado deve ser Não (Sim é o padrão).
  - O valor da opção de confirmação Aceitar o voto deve ser Sim (Sim é o padrão).
- A localização do agente vota confiável durante a onda de preparação. Os agentes do i5/OS sempre votam confiáveis. Isto porque as decisões heurísticas podem ser tomadas somente por um procedimento manual que avise sobre os possíveis efeitos colaterais negativos de se tomar uma decisão dessas.

### **Fluxo de Processamento de Confirmação com Otimização de Voto Confiável**

A figura a seguir mostra o fluxo de mensagens entre programas aplicativos e gerenciadores de transação quando a otimização do voto confiável é utilizada. O programa aplicativo iniciador e os programas aplicativos do agente não sabem da existência do processamento de confirmação de duas fases. Os números entre parênteses () na figura correspondem aos itens numerados na descrição que segue.



#### Legenda

- I = Iniciador (Aplicativo que inicia a confirmação do pedido)
- TM-I = Gerenciador de transação para o iniciador
- A = Agente (Aplicativo que recebe a confirmação do pedido)
- TM-A = Gerenciador de transação para o agente

A lista a seguir descreve os eventos para processamento normal sem a otimização do último agente quando o agente vota confiável. Isto descreve um fluxo básico. A seqüência de eventos pode tornar-se mais complexa quando a rede do programa de transação tem vários níveis ou quando ocorrem erros.

1. O programa aplicativo A recebe um pedido para indicar que está pronto para receber um pedido do programa I.
2. O aplicativo iniciador (I) emite uma instrução de confirmação.
3. O gerenciador da transação do iniciador (TM-I) assume a função do iniciador nessa transação. Ele inicia a onda de preparação enviando uma mensagem de preparação para todas as outras localizações que estão participando da transação.
4. Os gerenciadores de transação de cada localização diferente assumem a função do agente (TM-A). O programa aplicativo A é notificado pelo TM-A que foi recebido um pedido para confirmação. Para arquivos ICF, a notificação está no formato do indicador ICF Receive Take Commit (RCVTKCMT) em que está sendo definido.
5. O programa aplicativo A responde emitindo uma instrução de confirmação (ou uma instrução de rollback). Este é o voto do programa aplicativo.
6. O agente (TM-A) responde ao iniciador (TM-I) com uma mensagem de registro do pedido. Os sistemas i5/OS enviam um indicador de voto confiável com a confirmação do pedido.
7. Quando o iniciador (TM-I) receber todos os votos, o TM-I enviará uma mensagem de confirmação. Se a opção de confirmação Aguardar pelo resultado for N (Não) e a opção Aceitar voto confiável for S (Sim), será enviado um indicador sem reinicialização com a mensagem de confirmação. Isto informa ao agente que nenhuma mensagem de reinicialização será necessária para responder à confirmação.
8. A transação está concluída. Um retorno é enviado aos programas aplicativos (I e A). Este retorno indica que a operação confirmar foi bem-sucedida. Se um dano heurístico ocorrer no sistema A devido

a uma decisão heurística tomada antes da mensagem confirmada ser recebida, o aplicativo I não será informado. Em vez disso, uma mensagem será enviada à fila de mensagens QSYSOPR. Porém, um aplicativo A receberá a indicação de dano heurístico.

9. Na próxima vez que o agente (TM-A) enviar qualquer mensagem ao iniciador (TM-I), seja um fluxo de dados ou uma instrução de confirmação, um indicador de reinicialização implícita será enviada com a mensagem para informar ao TM-I que TM-A concluiu a confirmação com sucesso. O motivo disso é que o TM-I deve reter informações sobre a transação concluída até que seja confirmado que o TM-A tenha recebido com sucesso a mensagem de confirmação na etapa 7 na página 44.

#### **Referências relacionadas**

API Change Commitment Options (QTNCHGCO)

## **Suporte de Transação XA para o Controle de Confirmação**

O DB2 para i5/OS pode participar nas transações globais do X/Open.

A Open Group definiu um modelo padrão de mercado para o trabalho transacional que permite que alterações efetuadas em recursos não relacionados façam parte de uma única transação global. Um exemplo disso são as alterações nos bancos de dados que são fornecidos por dois fornecedores separados. Esse modelo é chamado de *X/Open Distributed Transaction Processing*.

As publicações a seguir descrevem o modelo X/Open Distributed Transaction Processing detalhadamente:

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

Familiarize-se com as informações nestes manuais, especialmente o XA Specification, antes de tentar utilizar o suporte de transação XA fornecido pelo DB2 para i5/OS. Esses manuais podem ser localizados no Web site da Open Group.

Existem cinco componentes para o modelo DTP (Distributed Transaction Processing):

#### **AP (Programa Aplicativo)**

Implementa a função requerida do usuário, especificando uma seqüência de operações que envolvem recursos, como bancos de dados. Define o início e o fim de transações globais, acessa recursos dentro de tempo limite das transações e normalmente toma a decisão de fazer confirmação ou rollback de cada transação.

#### **TM (Gerenciador de Transações)**

Gerencia transações globais e coordena a decisão de iniciá-las, confirmá-las ou efetuar rollback delas, a fim de assegurar a conclusão da transação atômica. O TM também coordena atividades de recuperação com RMs depois da falha de um componente.

#### **RM (Gerenciador de Recursos)**

Gerencia uma parte definida dos recursos compartilhados do computador, como um sistema de gerenciamento de bancos de dados. O AP utiliza interfaces definidas por cada RM para executar trabalho transacional. O TM utiliza interfaces definidas pelo RM para efetuar a conclusão da transação.

#### **CRM (Gerenciador de Recursos de Comunicação)**

Permite que uma instância do modelo acesse outra instância, seja dentro ou fora do domínio do TM atual. Os CRMs estão fora do escopo do DB2 para i5/OS e não são abordados aqui.

## Protocolo de comunicação

Os protocolos são utilizados pelos CRMs para se comunicarem entre si. Isso está fora do escopo do DB2 para i5/OS e não é discutido aqui.

XA Specification (Especificação XA) faz parte do modelo DTP que descreve um conjunto de interfaces que será utilizado pelos componentes do TM e RM do modelo DTP. O DB2 para i5/OS implementa essas interfaces como um conjunto de APIs de estilo de plataforma UNIX e programas de saída. Consulte APIs de XA para obter a documentação detalhada dessas APIs e para obter informações adicionais sobre como utilizar o DB2 para i5/OS como um RM.

## System i Navigator e Transações XA

O System i Navigator suporta o gerenciamento de transações XA como Transações Globais.

Uma transação global pode conter alterações fora e dentro do DB2 para i5/OS. Uma transação global é coordenada por um gerenciador de transações externo, utilizando a arquitetura Open Group XA ou utilizando outra arquitetura semelhante. Um aplicativo confirma ou reverte uma transação global utilizando as interfaces fornecidas pelo gerenciador de transações. O gerenciador de transações utiliza os protocolos de confirmação definidos pela arquitetura XA, ou por outra arquitetura, para concluir a transação. O DB2 para i5/OS atua como um gerenciador de recursos XA ao participar em uma transação global. Existem dois tipos de transações globais:

- **Bloqueios com Escopo de Transação:** Bloqueios adquiridos em nome da transação são examinados nesta. A transação pode mover de um job ou encadeamento para outro.
- **Bloqueios com Escopo de Job:** Bloqueios adquiridos em nome da transação são examinados no job. A transação não pode ser movida do job que a iniciou.

## Considerações para Transações XA

As APIs de XA para bloqueios de transação delimitada são recomendadas para os novos usuários do suporte a transações XA. As APIs de XA para bloqueios de tarefa delimitada continuarão a ser suportadas, mas não possuem mais vantagens sobre as APIs de XA para bloqueios de transação delimitada. As APIs de XA para bloqueios de transação delimitada possuem menos restrições e melhor desempenho nas seguintes situações:

- Se várias conexões SQL forem utilizadas para operar em uma única ramificação de transação de XA.
- Se uma única conexão SQL for utilizada para operar em várias ramificações de transações simultâneas de XA.

Nestas situações, um job separado deve ser iniciado para executar as ramificações das transações de XA quando você utilizar as APIs de XA para os Bloqueios de Escopo de Job.

Entenda as considerações e restrições a seguir antes de utilizar o DB2 para i5/OS como um RM. O termo *encadeamento* se refere a um job que não tem capacidade de encadeamento ou um único encadeamento dentro de um job com capacidade de encadeamento.

As considerações a seguir aplicam-se a transações com bloqueios de escopo de transação e transações com bloqueios de escopo de jobs, a menos que indicado o contrário.

## Considerações do DB2 para i5/OS

- As transações de XA executadas em um banco de dados local devem ser executadas nos jobs que estão em execução no modo do servidor SQL. Para tais transações, se a API `xa_open()` ou `db2xa_open()` for utilizada em uma tarefa que já não esteja sendo executada no modo do servidor SQL, o modo do servidor SQL será iniciado implicitamente. Você pode se referir a XA APIs para restrições nas interfaces do banco de dados suportado.

- As transações XA em um banco de dados remoto devem utilizar o modo do servidor SQL quando você utiliza as APIs de XA para bloqueios de tarefa delimitada. Entretanto, o modo do servidor é opcional para transações XA em um banco de dados remoto quando você utiliza as APIs de XA para bloqueios de transação delimitada. Além disso, as alterações nos arquivos DDM que utilizam os métodos tradicionais de acesso ao banco de dados do i5/OS são permitidas nas transações XA em um banco de dados remoto quando o modo do servidor SQL não é utilizado.
- Durante as chamadas de API do XA, a especificação XA relata quaisquer erros que forem detectados pelo DB2 para i5/OS através dos códigos de retorno. As mensagens de diagnóstico permanecem no log dos jobs quando o significado do erro não puder ser determinado a partir do código de retorno sozinho.

## Considerações sobre SQL Incorporado

- Para utilizar uma conexão SQL (Structured Query Language) para transações XA, você deve utilizar a API (Interface de Programação de Aplicativo) `xa_open()` ou `db2xa_open()` antes de estabelecer a conexão SQL. O banco de dados relacional que será conectado deve ser transmitido para a API `xa_open()` ou `db2xa_open()` pelo parâmetro `xainfo`. O perfil do usuário e a senha a serem utilizados na tarefa à qual a conexão será roteada podem ser transmitidos para a API `xa_open()` ou `db2xa_open()`. Se não forem transmitidos, o perfil utilizará aquele que foi especificado ou utilizado como o padrão durante a tentativa de conexão.

**Nota:** A consideração a seguir aplica-se somente a transações com bloqueios de tarefa delimitada.

- Se o SQL embutido for utilizado para executar transações de XA, o trabalho executado para cada conexão será roteado para um job diferente, mesmo que as conexões sejam estabelecidos no mesmo encadeamento. Isto é diferente do modo do servidor SQL sem XA, em que o trabalho executado para todas as conexões em um único encadeamento é roteado para o mesmo job. Isso porque a especificação XA requer uma chamada separada de preparação, confirmação ou rollback para cada instância do gerenciador de recursos.

**Nota:** A consideração a seguir aplica-se somente a transações com bloqueios de tarefa delimitada.

- Se o SQL embutido for utilizado para executar transações de XA, será possível estabelecer somente uma conexão por banco de dados relacional por encadeamento. Sempre que um encadeamento não estiver associada ativamente a uma ramificação da transação, o trabalho solicitado em uma das conexões do encadeamento fará com que o RM utilize programa de saída `ax_reg()` do TM para determinar se o trabalho deverá se iniciar, retomar ou associar-se a uma ramificação da transação.

Se o trabalho for iniciado em uma ramificação da transação, será executado na conexão do encadeamento com o banco de dados relacional correspondente.

Se o trabalho associar-se a uma ramificação da transação, será roteado novamente pela conexão com o banco de dados relacional correspondente que foi estabelecida no encadeamento que iniciou a ramificação da transação. Observe que o sistema não exige que o perfil do usuário dessa conexão seja o mesmo que o da conexão do encadeamento que está sendo associada. O TM é responsável por garantir que isso não seja uma preocupação de segurança. Os TMs típicos utilizam o mesmo perfil do usuário para todas as conexões. Este perfil do usuário é autorizado em todos os dados gerenciados pelo TM. A segurança de acesso adicional a esses dados é gerenciada pelo TM ou AP, em vez de utilizar as técnicas de segurança padrão do i5/OS.

**Nota:** A consideração a seguir aplica-se somente a transações com bloqueios de tarefa delimitada.

- Se o trabalho tiver que retomar uma ramificação da transação, a conexão utilizada dependerá de saber se a associação da ramificação da transação suspensa foi estabelecida iniciando-se ou associando-se à ramificação da transação.

O trabalho seguinte será executado na mesma conexão até que a API `db2xa_end()` seja utilizada para suspender ou finalizar a associação do encadeamento com essa ramificação da transação.

## Considerações sobre CLI

- Se a CLI for utilizada para desempenhar transações XA, mais de uma conexão poderá ser estabelecida no mesmo encadeamento depois que a API `db2xa_open()` for utilizada. As conexões podem ser utilizadas em outros encadeamentos para desempenhar transações XA, contanto que esses outros encadeamentos utilizem primeiro a API `db2xa_open()` com o mesmo valor de parâmetro `xainfo`.

**Nota:** A consideração a seguir aplica-se somente a transações com bloqueios de tarefa delimitada.

- Se a CLI for utilizada para executar transações de XA, a conexão utilizada para iniciar uma ramificação de transação deverá ser utilizada para todo o trabalho nessa ramificação. Se outros encadeamentos tiver que se unir à ramificação da transação, o manipulador da conexão utilizado para iniciar a ramificação da transação deverá passar para o encadeamento que está se unindo, de modo que possa executar o trabalho nessa mesma conexão. Da mesma forma, se um encadeamento tiver que retomar a ramificação da transação, a mesma conexão deverá ser utilizada.

Como as manipulações de conexão CLI não podem ser utilizadas em uma tarefa diferente, a função de junção está limitada aos encadeamentos em execução na mesma tarefa que iniciou a seção de transação quando a CLI foi utilizada.

## Considerações sobre Bancos de Dados Relacionais Remotos

**Nota:** Estas considerações sobre um banco de dados relacional remoto aplicam-se somente às transações com bloqueios de tarefa delimitada.

- As conexões de XA com um banco de dados relacional remoto serão suportadas somente se o banco de dados relacional residir em um sistema que suporte as conexões DRDA da DUW (Distributed Unit of Work). Isso inclui produtos System i que executam conversões DRDA (Distributed Relational Database Architecture) sobre SNA LU 6.2 ou que utilizam o V5R1 ou posterior ao executar o DRDA utilizando as conexões TCP/IP. Isso também inclui outras plataformas que suportam DRDA sobre SNA LU 6.2 ou que suportam o protocolo XA utilizando o DRDA sobre TCP/IP.
- Antes de utilizar a função de junção de XA, a API `db2xa_open()` deve ser utilizada no encadeamento de junção. O mesmo nome do banco de dados relacional e o RMID devem ser especificados na API `db2xa_open()` no encadeamento que iniciou a ramificação da transação e no encadeamento de junção. Se a ramificação da transação estiver ativa quando uma junção for tentado, o encadeamento de junção será bloqueada. Ela permanecerá bloqueada até que o encadeamento ativo suspenda ou encerre sua associação com a ramificação da transação.

## Consideração sobre Recuperação

- O suporte heurístico manual de confirmação e de rollback fornecido para todas as definições de confirmação poderá ser usado se for necessário impor que uma ramificação da transação faça confirmação ou rollback enquanto estiver no estado preparado.

## Considerações sobre Seção de Transação

- As informações sobre as ramificações de transações de XA são mostradas como parte das informações de controle de confirmação exibidas pelo System i Navigator e pelos comandos Work with Job (WRKJOB), Display Job (DSPJOB) e Work with Commitment Definition (WRKCMTDFN). O nome do TM, o estado da ramificação de transação, o identificador da transação e o qualificador da ramificação são todos mostrados. As definições de confirmação relacionadas a todas as transações ativas de XA atualmente podem ser exibidas utilizando o comando `WRKCMTDFN JOB(*ALL) STATUS(*XOPEN)` ou exibindo a pasta **Transações Globais** no System i Navigator.

**Nota:** O item a seguir aplica-se somente a transações com bloqueios de tarefa delimitada.

- Se uma associação entre um encadeamento e uma seção de transação existente estiver suspensa ou foi finalizada pela API `db2xa_end()`, o encadeamento poderá iniciar uma nova seção de transação. Se a conexão utilizada para iniciar a nova seção de transação foi utilizada antes para iniciar uma seção de transação diferente, e a associação do encadeamento com essa seção de transação foi finalizada ou suspensa pela API `db2xa_end()`, uma nova tarefa do servidor SQL poderá ser iniciada. Um novo job do

servidor SQL será necessário somente se a primeira ramificação da transação ainda não foi concluída pela API `db2xa_commit()` ou `db2xa_rollback()`. Neste caso, outra mensagem de conclusão SQL7908 será enviada para o log dos jobs que identifica o novo job do servidor SQL, assim como o job do servidor SQL original da conexão foi identificado quando a conexão foi estabelecida. Todos os pedidos do SQL para a nova ramificação da transação são roteados para o novo job do servidor SQL. Quando a ramificação da transação for concluída pela API `db2xa_commit()` ou `db2xa_rollback()`, o novo job do servidor SQL será reciclado e retornará ao conjunto de jobs pré-inicial.

- Uma seção de transação é marcada como Somente Rollback nas situações a seguir somente para as transações XA de bloqueios de tarefa delimitada:
  - Um encadeamento finaliza quando ainda estiver associada à ramificação da transação. Isso inclui uma finalização de encadeamento como resultado do término do processo.
  - Falha do sistema.
- Com transações XA para bloqueios de transação delimitada, uma seção de transação terá o rollback efetuado pelo sistema se algum encadeamento ainda estiver associado a ela quando ocorrer uma das seguintes situações:
  - A conexão relacionada à ramificação da transação finalizou.
  - O job que iniciou a ramificação da transação finalizou.
  - Falha do sistema.

**Nota:** A consideração a seguir aplica-se somente a transações com bloqueios de tarefa delimitada.

- Existe uma situação em que uma ramificação da transação será revertida pelo sistema, independente de ainda existirem encadeamentos associados: Isto ocorre quando o job do servidor SQL, para onde o trabalho da conexão está sendo roteado, finalizou. Isto pode ocorrer somente quando o comando de CL End Job (ENDJOB) for utilizado nesse job.
- Uma seção de transação não será afetada se nenhum encadeamento tiver uma associação ativa com ela quando ocorrer uma das seguintes situações: O TM pode confirmar ou efetuar rollback da seção de transação de qualquer encadeamento que tenha utilizado a API `xa_open()` ou `db2xa_open()` com o mesmo valor do parâmetro `xainfo` especificado no encadeamento que iniciou a seção de transação.
  - A conexão relacionada à ramificação da transação finalizou.
  - Um encadeamento ou tarefa que desempenhou o trabalho para a seção de transação utiliza a API `xa_close()` ou `db2xa_close()`.
  - Falha do sistema. Neste caso, a ramificação da transação não é afetada, somente se estiver no estado preparado. No estado de inatividade, o sistema é revertido.
- Quando os XIDs (Identificadores de Transação) de duas seções de transação XA possuem o mesmo GTRID (Identificador de Transação Global), mas diferentes BQUALs (Qualificadores de Ramificação), eles são chamados de *desconexas*. Por padrão, as seções de transação desconexas não compartilham bloqueios. Entretanto, ao utilizar as APIs de XA para bloqueios de transação delimitada, existe uma opção que permite que as transações desconexas compartilhem bloqueios.

#### Conceitos relacionados

“Considerações para Transações XA” na página 24

No ambiente XA, cada banco de dados é considerado um gerenciador de recursos separado. Quando um gerenciador de transações deseja acessar dois bancos de dados sob a mesma transação, ele deve usar os protocolos XA para executar a confirmação de duas fases com os dois gerenciadores de recursos.



Web Site da The Open Group

“Modo do Servidor SQL e Transações com Escopo de Encadeamento para o Controle de Confirmação” na página 50

As definições de confirmação com bloqueios de escopo de job são normalmente examinadas em um grupo de ativação.

#### Tarefas relacionadas

“Quando Forçar as Operações de Confirmação e de Rollback e Quando Cancelar a Ressincronização” na página 116

A decisão para forçar uma operação de confirmação ou de rollback é chamada de *decisão de heurística*. Esta ação permite que um operador confirme ou faça rollback manualmente dos recursos para uma transação que esteja em um estado preparado.

## Modo do Servidor SQL e Transações com Escopo de Encadeamento para o Controle de Confirmação

As definições de confirmação com bloqueios de escopo de job são normalmente examinadas em um grupo de ativação.

Se uma tarefa for multiencadeada, todos os encadeamentos na tarefa terão acesso à definição de confirmação e as alterações feitas em uma determinada transação poderão ser distribuídas por vários encadeamentos. Ou seja, todos os encadeamentos cujos programas são executados no mesmo grupo de ativação participam de uma única transação.

Há casos em que é desejável que o trabalho transacional seja examinado no encadeamento, em lugar de um grupo de ativação. Ou seja, cada encadeamento possui sua própria definição de confirmação e o trabalho transacional de cada uma é independente do trabalho desempenhado em outros encadeamentos.

O DB2 para i5/OS fornece esse suporte utilizando a API Change Job (QWTCHGJB) para alterar o job para execução no modo do servidor SQL. Quando uma conexão SQL é solicitada no modo do servidor SQL, ela é roteada para um job separado. Todas as operações seguintes do SQL executadas para essa conexão também são roteadas para esse job. Quando a conexão for feita, a mensagem de conclusão SQL7908 será enviada ao log de job do modo de servidor SQL que indica a qual job os pedidos SQL serão roteados. A definição de confirmação é de propriedade do job indicado nessa mensagem. Se ocorrerem erros, pode ser necessário consultar os logs de job para que ambos os jobs entendam a origem do problema porque nenhum trabalho real é feito no job que executa as instruções SQL.

Quando executado no modo do servidor SQL, somente as interfaces SQL poderão ser utilizadas para desempenhar trabalho sob o controle de confirmação. É possível utilizar o SQL incorporado ou a CLI (Call Level Interface). Todas as conexões estabelecidas por meio do SQL embutido em um único encadeamento serão roteadas para o mesmo job de backend. Isto permite que o trabalho de todas as conexões seja confirmado por um único pedido de confirmação, exatamente como seria em uma tarefa não executada no modo do servidor SQL. Cada conexão estabelecida por meio da CLI é roteada para um job separado. A CLI requer que o trabalho seja executado para cada conexão que seja confirmada ou revertida de forma independente.

Não é possível executar as operações seguintes sob o controle de confirmação quando executadas no modo do servidor SQL:

- Alterações de registros feitas com interfaces que não sejam SQL;
- Alterações em arquivos DDM;
- Alterações em recursos de confirmação da API.

Não é possível iniciar um controle de confirmação diretamente em um job executado no modo do servidor SQL.

### Conceitos relacionados

“Suporte de Transação XA para o Controle de Confirmação” na página 45

O DB2 para i5/OS pode participar nas transações globais do X/Open.

Executando a CLI do DB2 no Modo do Servidor

Iniciando a CLI do DB2 no Modo do Servidor SQL

Restrições para Executar a CLI do DB2 no Modo do Servidor

### Referências relacionadas



## Iniciando o Controle de Confirmação

Para iniciar o controle de confirmação, utilize o comando Start Commitment Control (STRCMTCTL).

**Nota:** O controle de confirmação não precisa ser iniciado pelos aplicativos SQL. O SQL implicitamente inicia o controle de confirmação em um tempo de conexão quando o nível de isolamento do SQL não é \*NONE.

Ao utilizar o comando STRCMTCTL, você pode especificar estes parâmetros.

### Nível de Bloqueio de Confirmação

Especifique o nível de bloqueio com o parâmetro LCKLVL no comando STRCMTCTL. O nível especificado torna-se o nível padrão do bloqueio de registro dos arquivos de banco de dados abertos e colocados sob o controle de confirmação da definição de confirmação.

### Objeto de Notificação de Confirmação

Utilize o parâmetro NTFY para especificar o objeto de notificação. Um objeto de notificação é uma fila de mensagens, área de dados ou arquivo de banco de dados contendo informações que identificam a última transação concluída com sucesso para uma determinada definição de confirmação, se ela não foi finalizada normalmente.

### Parâmetro Escopo de Confirmação

Utilize o parâmetro CMTSCOPE para especificar o escopo de confirmação. Quando o controle de confirmação é iniciado, o sistema cria uma definição de confirmação. O parâmetro escopo de confirmação identifica o escopo da definição de confirmação. O padrão é alcançar a definição de confirmação do grupo de ativação do programa que efetua o pedido inicial de controle de confirmação. O escopo alternativo é o do job.

### Parâmetro Diário Padrão

Você pode especificar um diário padrão quando inicia um controle de confirmação. O diário padrão pode ser utilizado nestes casos:

- Você deseja capturar as entradas no diário da transação. Essas entradas podem ajudá-lo a analisar o histórico de quais recursos estão associados a uma transação. Eles não são utilizados para aplicar ou remover as alterações criadas no diário. O parâmetro omitir entradas do diário (OMTJRNE) determina se o sistema gravará entradas da transação.
- Você deseja melhorar o desempenho dos jobs que fecham os arquivos e os abrem novamente dentro de uma etapa de roteamento. Se você fechar todos os arquivos atribuídos a um diário que não seja o padrão, todas as informações no sistema sobre o diário serão removidas da etapa de roteamento. Se um arquivo atribuído a esse diário for aberto mais tarde, todas essas informações deverão ser criadas novamente. O sistema manterá as informações sobre o diário padrão com a definição de confirmação, se quaisquer recursos atribuídos ao diário estiverem ativos.

### Parâmetro Texto de Confirmação

Utilize o parâmetro TEXT para identificar o texto específico a ser associado a uma definição de confirmação ao exibir informações sobre as definições de confirmação iniciadas para um job. Se nenhum texto estiver especificado, o sistema fornecerá uma descrição de texto padrão.

### Parâmetro Omitir Entradas do Diário

Se você especificar um diário padrão para melhorar o desempenho, poderá utilizar o parâmetro OMTJRNE para impedir que o sistema grave as entradas no diário da transação. Quando o sistema grava entradas de transação, aumenta significativamente o tamanho do arquivo de histórico e degrada o desempenho durante as operações de confirmação e rollback.

As entradas da transação podem ser úteis quando você está configurando e testando o ambiente de controle de confirmação ou um novo aplicativo.

As entradas da transação são gravadas no diário padrão, independente do valor do parâmetro OMTJRNE, nestas condições:

- Ocorre um erro do sistema durante uma operação de confirmação ou rollback.
- Uma alteração manual é feita em um recurso que participou de uma transação e a alteração causou uma condição heurística mista. Consulte Estados da transação para controle de confirmação de duas fases para obter uma descrição da condição heurística mista. Esse tipo de alteração manual é chamado de decisão heurística.

Você pode utilizar as informações sobre quais recursos participaram da transação para determinar qual ação executará nestas situações.

Você pode utilizar o localizador de informações de entrada do Diário para mostrar os layouts dos dados específicos da entrada para as entradas do diário da transação (controle de confirmação).

### **Conceitos relacionados**

“Estados da Transação para o Controle de Confirmação de Duas Fases” na página 30

Uma definição de confirmação é estabelecida em cada localização que faz parte da rede do programa de transação. Para cada uma, o sistema mantém o estado de sua transação atual e da anterior.

Localizador de Informações de Entradas do Diário

### **Tarefas relacionadas**

“Quando Forçar as Operações de Confirmação e de Rollback e Quando Cancelar a Ressincronização” na página 116

A decisão para forçar uma operação de confirmação ou de rollback é chamada de *decisão de heurística*. Esta ação permite que um operador confirme ou faça rollback manualmente dos recursos para uma transação que esteja em um estado preparado.

### **Referências relacionadas**

Comando Start Commitment Control (STRCMTCTL)

## **Objeto de Notificação de Confirmação**

Um *objeto de notificação* é uma fila de mensagens, área de dados ou arquivo de banco de dados contendo informações que identificam a última transação concluída com sucesso para uma determinada definição de confirmação, se ela não foi finalizada normalmente.

As informações utilizadas para identificar a última transação bem-sucedida para uma definição de confirmação são fornecidas pela *identificação de confirmação* que associa uma operação confirmar a um conjunto específico de alterações de recursos confirmáveis.

A identificação de confirmação da última transação bem-sucedida para uma definição de confirmação será colocada no objeto de notificação somente se a definição não finalizar normalmente. Essas informações podem ser utilizadas para ajudar a determinar onde o processamento de um aplicativo finalizou, de forma que o aplicativo possa ser iniciado novamente.

Para conjuntos de discos independentes, o objeto de notificação deve residir no mesmo conjunto ou grupo de discos independentes que a definição da confirmação. Se você mover a definição de confirmação para outro conjunto ou grupo de discos independentes, o objeto de notificação também deverá residir neles. O objeto de notificação no outro conjunto de discos independentes ou grupo de conjuntos de discos independentes será atualizado se a definição de confirmação finalizar de forma anormal. Se o objeto de notificação não for encontrado no outro conjunto ou grupo de discos independentes, a atualização falhará a mensagem CPF8358.

Se os recursos registrados em diário participarem da transação atual e uma operação de confirmação for executada com uma identificação de confirmação, essa será colocada na entrada do diário de confirmação (código do diário e tipo de entrada de C CM) que identifica essa determinada transação como sendo confirmada. É enviada uma entrada no diário de confirmação contendo a identificação de confirmação para cada diário associado a recursos que participaram da transação.

A tabela a seguir mostra como você especifica a identificação de confirmação e seu tamanho máximo. Se a identificação de confirmação exceder o tamanho máximo, ela será truncada quando gravada no objeto de notificação.

Idioma	Operação	Número Máximo de Caracteres na Identificação de Confirmação
CL	Comando COMMIT	3000 <sup>1</sup>
RPG ILE (Integrated Language Environment)	Código de operação COMMIT	4000 <sup>1</sup>
PLI	Sub-rotina PLICOMMIT	4000 <sup>1</sup>
ILE C	Função _Rcommit	4000 <sup>1</sup>
ILE COBOL	Verbo COMMIT	Não suportado
SQL	Instrução COMMIT	Não suportado
<b>Nota:</b> <sup>1</sup> Se o objeto de notificação estiver em uma área de dados, o tamanho máximo será 2000 caracteres.		

Quando um objeto de notificação é atualizado com a identificação de confirmação, será atualizado da seguinte forma:

#### Arquivo de Banco de Dados

Se um arquivo de banco de dados for utilizado como objeto de notificação, a identificação de confirmação será incluída no final do arquivo. Todos os registros existentes serão deixados no arquivo. Como vários usuários ou jobs podem estar alterando registros ao mesmo tempo, cada identificação de confirmação no arquivo contém informações exclusivas para associar os dados à definição do job e de confirmação que falhou. O arquivo que servir pode ser registrado em diário.

#### Área de Dados

Se uma área de dados for utilizada como objeto de notificação, todo o seu conteúdo será substituído quando a identificação de confirmação for colocada na área de dados. Se mais de um usuário ou job estiver utilizando o mesmo programa, somente a identificação de confirmação da última definição de confirmação que não finalizou normalmente estará na área de dados. Conseqüentemente, um único objeto de notificação da área de dados poderá não produzir as informações corretas para iniciar os programas aplicativos novamente. Para solucionar este problema, utilize uma área de dados separada para cada definição de confirmação de cada usuário ou job da estação de trabalho.

#### Fila de Mensagens

Se uma fila de mensagens for utilizada como um objeto de notificação, a mensagem CPI8399 será enviada para ela. A identificação de confirmação será colocada no texto do segundo nível da mensagem CPI8399. Como ocorre com o uso de um arquivo de banco de dados para o objeto de notificação, o conteúdo de cada identificação de confirmação identifica exclusivamente uma determinada definição de confirmação para um job, de modo que um programa aplicativo pode ser iniciado novamente.

#### Conceitos relacionados

“Controle de Confirmação para Aplicativos Batch” na página 27

As aplicações em batch podem ou não precisar de controle de confirmação. Em alguns casos, um aplicativo batch pode executar uma única função de leitura de um arquivo de entrada e atualização de um arquivo master. Porém, você pode usar o controle de confirmação para este tipo de aplicativo se for importante iniciá-lo novamente após uma finalização anormal.

“Exemplo: Utilizando um Objeto de Notificação para Iniciar um Aplicativo” na página 93

Quando um programa é iniciado após uma finalização anormal, talvez procure uma entrada no objeto de notificação. Se a entrada existir, o programa poderá iniciar uma transação novamente. Depois de iniciada a transação, o programa limpa o objeto de notificação para impedir que se inicie a mesma transação em outra hora.

## Nível de Bloqueio de Confirmação

O valor especificado para o parâmetro LCKLVL no comando Start Commitment Control (STRCMTCTL) torna-se o nível padrão de bloqueio de registro para os arquivos de banco de dados que são abertos e colocados sob controle de confirmação para a definição de confirmação.

O nível padrão de bloqueio de registro não pode ser substituído na abertura de arquivos de banco de dados local. Porém, os arquivos de banco de dados, acessados pelo SQL, utilizam o atual nível de isolamento do SQL em vigor no momento da primeira instrução SQL emitida junto a ele.

O nível de bloqueio deve ser especificado de acordo com suas necessidades, os períodos de espera permitidos e os procedimentos de liberação utilizados com maior frequência.

As descrições a seguir aplicam-se somente aos arquivos abertos sob controle de confirmação:

### Nível de Bloqueio \*CHG

Utilize este valor se quiser proteger registros alterados contra alterações feitas por outros jobs sendo executados ao mesmo tempo. Para arquivos abertos sob controle de confirmação, o bloqueio é mantido no período de duração da transação. Para arquivos que não são abertos sob controle de confirmação, o bloqueio no registro é mantido somente do momento em que o registro é lido até a conclusão da operação de atualização.

### Nível de Bloqueio \*CS

Utilize este valor para proteger registros alterados e recuperados contra alterações feitas por outros jobs sendo executados ao mesmo tempo. Os registros recuperados que não são alterados são protegidos somente até serem liberados ou um registro diferente seja recuperado.

O nível de bloqueio \*CS assegura que outros jobs não poderão ler um registro para atualização que tenha sido lido por este job. Além disso, o programa não pode ler registros para atualização, que foram bloqueados com um tipo de bloqueio de registro de \*UPDATE, em outro job até que aquele job acesse um registro diferente.

### Nível de Bloqueio \*ALL

Utilize este valor para proteger registros alterados e registros recuperados, que estão sob controle de confirmação, contra alterações feitas por outros jobs sendo executados, sob controle de confirmação, ao mesmo tempo. Os registros que são recuperados ou alterados são protegidos até a próxima operação de confirmação ou rollback.

O nível de bloqueio \*ALL assegura que outros jobs não poderão acessar um registro para atualização que tenha sido lido por este job. Isto é diferente do protocolo de bloqueio normal. Quando o nível de bloqueio está especificado como \*ALL, até mesmo um registro que não é lido para atualização não pode ser acessado se estiver bloqueado com um tipo de bloqueio de registro de \*UPDATE em outro job.

A tabela a seguir mostra a duração de bloqueios de registro para arquivos que estão e não estão sob controle de confirmação.

Pedido	Parâmetro LCKLVL	Duração do Bloqueio	Tipo de Bloqueio
Somente leitura	Sem controle de confirmação	Sem bloqueio	Nenhuma
	*CHG	Sem bloqueio	Nenhuma
	*CS	Da leitura até a próxima leitura, confirmação ou rollback	*READ
	*ALL	Da leitura até a confirmação ou rollback	*READ

Pedido	Parâmetro LCKLVL	Duração do Bloqueio	Tipo de Bloqueio
Leitura para atualização, em seguida, atualização ou exclusão <sup>1</sup>	Sem controle de confirmação	Da leitura até atualização ou exclusão	*UPDATE
	*CHG	Da leitura até atualização ou exclusão	*UPDATE
		Depois, da atualização ou exclusão até a próxima confirmação ou rollback <sup>2</sup>	*UPDATE
	*CS	Da leitura até atualização ou exclusão	*UPDATE
		Depois, da atualização ou exclusão até a próxima confirmação ou rollback <sup>2</sup>	*UPDATE
	*ALL	Da leitura até atualização ou exclusão	*UPDATE
Depois, da atualização ou exclusão até a próxima confirmação ou rollback <sup>2</sup>			
Leitura para atualização, depois, liberação <sup>1</sup>	Sem controle de confirmação	Da leitura até liberação	*UPDATE
	*CHG	Da leitura até liberação	*UPDATE
	*CS	Da leitura até liberação, confirmação ou rollback	*UPDATE
		Depois da liberação até a próxima leitura, confirmação ou rollback	*UPDATE
	*ALL	Da leitura até liberação, confirmação ou rollback	*UPDATE
		Depois da liberação até próxima confirmação ou rollback	
Incluir	Sem controle de confirmação	Sem bloqueio	Nenhuma
	*CHG	Da inclusão até a confirmação ou rollback	*UPDATE
	*CS	Da inclusão até a confirmação ou rollback	*UPDATE
	*ALL	Da inclusão até a confirmação ou rollback	*UPDATE
Gravar direto	Sem controle de confirmação	Para duração de gravar direto	*UPDATE
	*CHG	Da gravação direta para confirmação ou rollback	*UPDATE
	*CS	Da gravação direta para confirmação ou rollback	*UPDATE
	*ALL	Da gravação direta para confirmação ou rollback	*UPDATE

Pedido	Parâmetro LCKLVL	Duração do Bloqueio	Tipo de Bloqueio
<b>Notas:</b>			
<p><sup>1</sup>Se uma operação de confirmação ou rollback for desempenhada após uma operação de leitura-para-atualização, mas antes do registro ser atualizado, excluído ou liberado, o registro será desbloqueado durante a operação de confirmação ou rollback. A proteção no registro é perdida assim que a confirmação ou rollback é concluída.</p> <p><sup>2</sup>Se um registro é excluído, mas a confirmação ou rollback ainda não foi emitida para transação, o registro excluído não permanece bloqueado. Se o mesmo job ou um job diferente tenta ler o registro excluído por chave, o job recebe uma indicação de registro não encontrado. Entretanto, se houver no arquivo um caminho de acesso exclusivo com chave, outro job é impedido de inserir ou atualizar um registro com o mesmo valor de chave exclusivo daquele do registro excluído até que a transação seja confirmada.</p>			

Um tipo de bloqueio de registro de \*READ é obtido em registros que não são lidos para atualização quando o nível do bloqueio é \*CS ou \*ALL. Este tipo de bloqueio impede que outros jobs leiam os registros para atualização, mas não impede que os registros sejam acessados a partir de uma operação .

Um tipo de bloqueio de registro de \*UPDATE é obtido em registros que são atualizados, excluídos, incluídos ou lidos para atualização. Este tipo de bloqueio impede que outros jobs leiam os registros para atualização e impede que jobs, sendo executados sob controle de confirmação com um nível de bloqueio de registro de \*CS ou \*ALL acessem registros até mesmo para uma operação .

Programas que não estão usando controle de confirmação podem ler registros bloqueados por outro job, mas não podem ler registros para atualização, independente do valor especificado para o parâmetro LCKLVL.

O nível de bloqueio, especificado para uma definição de confirmação quando o controle de confirmação é iniciado para um grupo de ativação ou para a tarefa, aplica-se apenas a aberturas associadas àquela determinada definição de confirmação.

**Nota:** Os valores de nível de bloqueio \*CS e \*ALL protegem você contra recuperação de um registro que tenha atualmente uma alteração pendente de um job diferente. Porém, os valores de nível de bloqueio \*CS e \*ALL *não* protegem contra a recuperação de um registro utilizando um programa sendo executado em um grupo de ativação que atualmente possui uma alteração pendente de um programa sendo executado em um grupo de ativação diferente dentro do mesmo job.

Dentro do mesmo job, um programa pode alterar um registro que já foi alterado dentro da transação atual contanto que o registro seja acessado novamente utilizando a mesma definição de confirmação. Quando se está utilizando a definição de confirmação de nível de job, o acesso ao registro alterado pode ser feito a partir de um programa sendo executado dentro de qualquer grupo de ativação que esteja utilizando a definição de confirmação de nível de job.

#### **Conceitos relacionados**

“Considerações e Restrições para o Controle de Confirmação” na página 25

Você precisa estar ciente dessas considerações e restrições para o controle de confirmação.

#### **Referências relacionadas**

Comando Start Commitment Control (STRCMTCTL)

---

## **Encerrando o Controle de Confirmação**

O comando End Commitment Control (ENDCMTCTL) encerra o controle de confirmação para a definição de confirmação de nível do job e de nível do grupo de ativação.

A emissão do comando ENDCMTCTL indica ao sistema que a definição de confirmação sendo usada pelo programa que está fazendo o pedido deve ser finalizada. O comando ENDCMTCTL finaliza apenas uma definição de confirmação para o job e todas as outras definições de confirmação para o job permanecem inalteradas.

Se a definição de confirmação do nível do grupo de ativação for finalizada, os programas sendo executados dentro do grupo de ativação não podem mais fazer alterações sob controle de confirmação, a menos que a definição de confirmação do nível do job já tenha sido iniciada para o job. Se a definição de confirmação do nível do job estiver ativa, ela torna-se imediatamente disponível para ser usada pelos programas sendo executados dentro do grupo de ativação que acabou de finalizar o controle de confirmação.

Se a definição de confirmação do nível do job for finalizada, todos os programas sendo executados dentro do job, que estavam usando a definição de confirmação do nível do job, não podem mais fazer alterações sob controle de confirmação sem primeiro iniciar novamente o controle de confirmação com o comando STRCMTCTL.

Antes de emitir o comando ENDCMTCTL, as seguintes condições devem ser atendidas para que a definição de confirmação seja finalizada:

- Todos os arquivos abertos sob controle de confirmação para a definição de confirmação a ser finalizada devem ser primeiro fechados. A finalização da definição de confirmação do nível de job inclui todos os arquivos abertos, sob controle de confirmação, por quaisquer programas sendo executados em quaisquer grupos de ativação que estejam usando a definição de confirmação do nível do job.
- Todos os recursos de confirmação da API para a definição de confirmação a ser finalizada, devem primeiro ser removidos utilizando a API QTNRMVCR. A finalização da definição de confirmação do nível do job inclui todos os recursos de confirmação da API incluídos por quaisquer programas sendo executados em quaisquer grupos de ativação que estejam usando a definição de confirmação do nível do job.
- Um banco de dados remoto associado à definição de confirmação a ser finalizada deve ser desconectado.
- Todas as conversações protegidas associadas à definição de confirmação devem ser finalizadas normalmente utilizando o nível de sincronização correto.

Se o controle de confirmação estiver sendo finalizado em um job interativo e um ou mais recursos confirmáveis associados à definição de confirmação possuir alterações pendentes, a mensagem de indagação CPA8350 é enviada ao usuário perguntando se deve confirmar as alterações pendentes, reverter as alterações pendentes ou cancelar o pedido ENDCMTCTL.

Se o controle de confirmação estiver sendo finalizado em um job batch e um ou mais arquivos fechados associados à definição de confirmação a ser finalizada ainda possuir alterações pendentes, as alterações são revertidas e uma mensagem é enviada:

- CPF8356 se somente os recursos locais foram registrados
- CPF835C se somente os recursos remotos foram registrados
- CPF83E4 se os recursos locais e remotos foram registrados

Se um objeto de notificação estiver definido para a definição de confirmação sendo finalizada, ele poderá ser atualizado.

Quando um grupo de ativação, que possui uma API registrada como o último agente, está sendo finalizado, o programa de saída para a API é chamado para receber a decisão de confirmação ou rollback. Nesse caso, mesmo que o grupo de ativação esteja sendo finalizado normalmente, um pedido de rollback ainda pode ser retornado do programa de saída da API. Desta maneira, a operação implícita de confirmação talvez não possa ser executada.

Depois que a definição de confirmação é finalizada com sucesso, toda a recuperação necessária, se houver alguma, foi executada. Nenhuma recuperação adicional é executada para os recursos de confirmação associados à definição de confirmação que acabou de ser finalizada.

Depois que a definição de confirmação é finalizada, a definição de confirmação do nível do job ou do nível do grupo de ativação pode ser iniciada novamente para os programas sendo executados dentro do grupo de ativação. A definição de confirmação do nível de tarefa poderá ser iniciada apenas se já não estiver iniciada para a tarefa.

Embora as definições de confirmação possam ser iniciadas e finalizadas várias vezes pelos programas que são executados dentro do grupo de ativação, a quantia de recursos do sistema necessária para as operações repetidas de início e finalização podem levar a uma diminuição no desempenho do job e no desempenho geral do sistema. Portanto, é recomendado que uma definição de confirmação seja deixada ativa se um programa a ser chamado posteriormente for utilizá-la.

#### **Conceitos relacionados**

“Atualizações Feitas no Objeto de Notificação” na página 65

O sistema atualizará o objeto de notificação com a identificação de confirmação da última operação de confirmação bem-sucedida dessa definição de confirmação.

#### **Referências relacionadas**

Comando End Commitment Control (ENDCMTCTL)

---

## **Finalização do Controle de Confirmação Iniciado pelo Sistema**

O sistema pode finalizar o controle de confirmação ou executar uma operação implícita de confirmação ou rollback. Às vezes, a finalização do controle de confirmação iniciado pelo sistema é normal. Outras vezes, terminam com uma finalização anormal de sistema ou job.

## **Controle de Confirmação Durante Finalização do Grupo de Ativação**

O sistema finaliza automaticamente uma definição de confirmação do nível do grupo de ativação quando um grupo de ativação é finalizado.

Se existem alterações pendentes para uma definição de confirmação do nível do grupo de ativação e o grupo de ativação está sendo finalizado normalmente, o sistema executa uma operação implícita de confirmação para a definição de confirmação antes que ela seja finalizada. Caso contrário, uma operação implícita de rollback é executada para a definição de confirmação do nível do grupo de ativação antes de ser finalizada se o grupo de ativação estiver sendo finalizado de modo anormal ou se o sistema encontrou erros ao fechar quaisquer arquivos abertos sob o controle de confirmação com escopo para o grupo de ativação.

**Nota:** Uma operação implícita de confirmação ou rollback nunca é desempenhada durante o processamento de finalização do grupo de ativação para as definições de confirmação \*JOB ou \*DFTACTGRP. Isto porque as definições de confirmação \*JOB e \*DFTACTGRP nunca são finalizadas devido à finalização do grupo de ativação. Ao invés disso, estas definições de confirmação são finalizadas explicitamente com um comando ENDCMTCTL ou finalizadas pelo sistema quando o job é finalizado.

O sistema fecha automaticamente quaisquer arquivos com escopo para o grupo de ativação quando o grupo de ativação é finalizado. Isto inclui todos os arquivos de banco de dados com escopo para o grupo de ativação aberto sob controle de confirmação. O fechamento de tais arquivos ocorre antes de qualquer operação implícita de confirmação que possa ser desempenhada para a definição de confirmação do nível do grupo de ativação. Portanto, todos os registros que residem em um buffer de E/S são primeiro forçados para o banco de dados antes de executar qualquer operação implícita de confirmação.



Como parte da operação implícita de confirmação ou rollback que pode ser desempenhada, é feita uma chamada ao programa de saída de confirmação e rollback da API para cada recurso de confirmação da API associado à definição de confirmação do nível do grupo de ativação. O programa de saída deve concluir seu processamento dentro de 5 minutos. Depois que a confirmação e rollback da API e o programa de saída forem chamados, o sistema removerá automaticamente o recurso de confirmação da API.

Se uma operação implícita de rollback for desempenhada para uma definição de confirmação que estiver sendo finalizada devido à finalização de um grupo de ativação, o objeto de notificação, se houver um definido para a definição de confirmação, poderá ser atualizado.

#### **Conceitos relacionados**

“Atualizações Feitas no Objeto de Notificação” na página 65

O sistema atualizará o objeto de notificação com a identificação de confirmação da última operação de confirmação bem-sucedida dessa definição de confirmação.

## **Operações Implícitas de Confirmação e Rollback**

Em algumas instâncias, uma operação de confirmação ou rollback é iniciada pelo sistema para uma definição de confirmação. Estes tipos de operações de confirmação e rollback são conhecidos como *pedidos implícitos de confirmação e rollback*.

Geralmente, uma operação de confirmação ou rollback é iniciada a partir de um programa aplicativo utilizando uma das linguagens de programação disponíveis que suporta controle de confirmação. Estes tipos de operações de confirmação e rollback são conhecidos como *pedidos explícitos de confirmação e rollback*.

As duas tabelas a seguir mostram o que o sistema faz quando ocorrem determinados eventos relacionados a uma definição de confirmação que possui alterações pendentes. Uma definição de confirmação terá alterações pendentes se alguma das seguintes condições for verdadeira:

- Algum recurso confirmável foi atualizado.
- Um arquivo de banco de dados aberto sob controle de confirmação foi lido, pois a leitura de um arquivo altera a posição do arquivo.
- A definição de confirmação possui um recurso da API. Como as alterações em recursos da API são feitos por um programa do usuário, o sistema deve supor que todos os recursos da API possuem alterações pendentes.

A entrada no diário C CM (operação de confirmação) e a entrada no diário C RB (operação de rollback) indicam se a operação era explícita ou implícita.

A tabela a seguir mostra as ações que o sistema toma quando uma tarefa é finalizada, normal ou anormalmente, com base nas seguintes situações:

- O estado da transação.
- O valor do job ação-se-finalizado para a definição de confirmação.
- Se um recurso da API é ou não o último agente.

Estado	API último agente	Opção Ação se Finalizar Tarefa <sup>1</sup>	Operação de confirmação ou de rollback
RST	N/D	N/D	<p>Se a definição de confirmação não estiver associada à uma transação global X/Open, um rollback implícito é executado.</p> <p>Se a definição de confirmação estiver associada a uma transação global X/Open, os seguintes eventos ocorrerão:</p> <ul style="list-style-type: none"> <li>• Se o estado de ramificação da transação não for Ativo (S1), nenhuma ação é executada e a ramificação da transação permanece no mesmo estado.</li> <li>• Se o estado de ramificação da transação for Ativo (S1), um rollback implícito é executado.</li> </ul>
PIP	N/D	N/D	<p>Se a definição de confirmação não estiver associada à uma transação global X/Open, um rollback implícito é executado.</p> <p>Se a definição de confirmação estiver associada à uma transação global X/Open, a ramificação da transação encontra-se no estado Inativo (S2) e é deixada no estado Inativo (S2).</p>
PRP	N/D	AGUARDAR	<p>Se a definição de confirmação não estiver associada à uma transação global X/Open<sup>2</sup>, ocorre o seguinte:</p> <ul style="list-style-type: none"> <li>• A ressincronização é iniciada para receber a decisão do iniciador da operação de confirmação.</li> <li>• A decisão retornada para confirmar ou reverter é executada. Ela é considerada uma operação explícita.</li> </ul>

Estado	API último agente	Opção Ação se Finalizar Tarefa <sup>1</sup>	Operação de confirmação ou de rollback
PRP	N/D	C	Se a definição de confirmação não estiver associada a uma transação global X/Open <sup>2</sup> , uma operação de confirmação implícita será desempenhada.
		R	Se a definição de confirmação não estiver associada à uma transação global X/Open, uma operação de rollback implícita é executada.  Se a definição de confirmação estiver associada à uma transação global X/Open, ocorre o seguinte: <ul style="list-style-type: none"> <li>• Se o job que iniciou a transação for finalizado, a transação é deixada em um estado de preparada até que o XA TM a confirme ou a reverta. Neste caso, o estado de ramificação da transação XA será deixado como Preparado (S3).</li> <li>• Se o job do servidor SQL para o qual o trabalho da transação está sendo encaminhado for finalizado, rollback implícito forçado é executado implicitamente. Neste caso, o estado de ramificação da transação XA será mudado para Concluído Heuristicamente (S5).</li> </ul>
CIP	N/D	N/D	Uma operação de confirmação explícita é executada.
LAP	NÃO	AGUARDAR	1. A ressincronização até o último agente é utilizada para recuperar a decisão de confirmação ou rollback.
			2. A decisão retornada para confirmar ou reverter é executada. Ela é considerada uma operação explícita.
LAP	SIM	AGUARDAR	1. A API do último agente é chamada para recuperar a decisão de confirmação ou rollback.
			2. A operação de confirmação ou rollback é executada. Ela é considerada uma operação explícita.
LAP	N/D	C	Uma operação de confirmação implícita é executada.
		R	Uma operação de rollback implícita é executada.
CMT	N/D	N/D	Uma operação de confirmação já foi concluída para esta definição de confirmação e quaisquer localizações downstream. A operação de confirmação está concluída.
VRO	N/D	N/D	Os agentes local e remoto votaram para somente leitura. Todos os agentes downstream também devem ter votado para somente leitura. Nenhuma ação é exigida.

Estado	API último agente	Opção Ação se Finalizar Tarefa <sup>1</sup>	Operação de confirmação ou de rollback
RBR	N/D	N/D	Uma operação de rollback é exigida. Uma operação de rollback explícita é executada.
<b>Nota:</b>			
<sup>1</sup> Você pode alterar a opção Ação se Finalizar Tarefa com a API Change Commitment Options (QTNCHGCO).			
<sup>2</sup> Se a definição de confirmação estiver associada a uma transação global X/Open, os seguintes eventos ocorrerão:			
<ul style="list-style-type: none"> <li>• Se o job que iniciou a transação for finalizado, a transação é deixada em um estado de preparada até que o XA TM a confirme ou a reverta. Neste caso, o estado de ramificação da transação XA será deixado como Preparado (S3).</li> <li>• Somente para bloqueios com escopo de transação, se o job do servidor SQL para o qual está sendo enviado o trabalho da transação for finalizado, um rollback forçado é executado implicitamente. Neste caso, o estado de ramificação da transação XA será mudado para Concluído Heuristicamente (S5).</li> </ul>			

A tabela a seguir mostra as ações do sistema quando um grupo de ativação é finalizado e aplica-se somente às transações com bloqueios com escopo de job. As ações do sistema baseiam-se nos seguintes itens:

- O estado da transação. (É sempre reconfigurar (RST) quando um grupo de ativação é finalizado.)
- Como é a finalização do grupo de ativação, normal ou anormal.
- Se um recurso da API é ou não o último agente.

**Nota:** Se um recurso da API for registrado como o último agente, isto dá ao último agente o controle da decisão de confirmação ou rollback. O resultado da decisão é considerado uma operação explícita

Estado	API último agente	Tipo de finalização	Operação de confirmação ou de rollback
RST	Não	Normal	Uma operação de confirmação implícita é executada. Se existirem conversações protegidas, a definição de confirmação torna-se o iniciador raiz da operação de confirmação.
RST	Não	Anormal	Um rollback implícito é executado.
RST	Sim	Normal	O programa de saída da API é chamado. A operação de confirmação ou rollback é determinada pela API.
RST	Sim	Anormal	O programa de saída da API é chamado. A operação de confirmação ou rollback é determinada pela API.

### Conceitos relacionados

“Controle de Confirmação Durante Finalização Anormal do Sistema ou Job” na página 63

O sistema finaliza todas as definições de confirmação para um job quando este finaliza de modo anormal. Estas definições de confirmação são finalizadas durante o processamento de finalização do job. Este tópico aplica-se somente às definições de confirmação com bloqueio de escopo do job.

### Referências relacionadas

API Change Commitment Options (QTNCHGCO)

## Controle de Confirmação Durante Finalização Normal da Etapa de Roteamento

O sistema finaliza todas as definições de confirmação para um job quando uma etapa de roteamento é finalizada normalmente.

**Nota:** As informações a seguir aplicam-se apenas às definições de confirmação com bloqueios de tarefa delimitada.

Uma etapa de rota é finalizada normalmente por uma das seguintes situações:

- Uma finalização normal para um job em batch.
- Uma desconexão normal para um job interativo.
- Os comandos Reroute Job (RRTJOB), Transfer Job (TFRJOB) ou Transfer Batch Job (TFRBCHJOB) finalizam a etapa de roteamento atual e iniciam uma nova.

Qualquer outra finalização de uma etapa de roteamento será considerada anormal pelo código de conclusão diferente de zero na mensagem de conclusão do job CPF1164 no log dos jobs.

Antes de finalizar uma definição de confirmação durante a finalização da etapa de roteamento, o sistema executará uma operação de rollback implícita se a definição de confirmação tiver alterações pendentes. Isso inclui chamar a confirmação e rollback da API e o programa de saída para cada recurso de confirmação da API associado à definição de confirmação. O programa de saída deve concluir seu processamento dentro de 5 minutos. Depois que a confirmação e rollback da API e o programa de saída forem chamados, o sistema removerá automaticamente o recurso de confirmação da API.

Se um objeto de notificação estiver definido para a definição de confirmação, ele poderá ser atualizado.

### Conceitos relacionados

“Atualizações Feitas no Objeto de Notificação” na página 65

O sistema atualizará o objeto de notificação com a identificação de confirmação da última operação de confirmação bem-sucedida dessa definição de confirmação.

## Controle de Confirmação Durante Finalização Anormal do Sistema ou Job

O sistema finaliza todas as definições de confirmação para um job quando este finaliza de modo anormal. Estas definições de confirmação são finalizadas durante o processamento de finalização do job. Este tópico aplica-se somente às definições de confirmação com bloqueio de escopo do job.

Quando o sistema é finalizado de modo anormal, ele finaliza todas as definições de confirmação que foram iniciadas e que estão sendo usadas por todos os jobs ativos no momento da finalização anormal do sistema. Estas definições de confirmação são finalizadas como parte do processamento da recuperação do banco de dados que é executada durante o próximo IPL após a finalização anormal do sistema.

**Atenção:** A recuperação para definições de confirmação refere-se a uma finalização anormal para o sistema ou um job devido a um falha de energia, falha de hardware ou falha no sistema operacional ou código interno da licença. Você não deve usar o comando End Job Abnormal (ENDJOBABN) para forçar a finalização anormal de um job. A finalização anormal pode resultar em alterações pendentes para transações ativas para que o job que está sendo finalizado seja parcialmente confirmado ou revertido. O próximo IPL poderá tentar a recuperação para quaisquer transações parciais para o job finalizado com o comando ENDJOBABN.

O resultado da recuperação do controle de confirmação que o sistema executa durante um IPL para um job finalizado com o comando ENDJOBABN é incerto. Isto porque todos os bloqueios para recursos de confirmação são liberados quando a tarefa é finalizada de modo anormal. Alterações pendentes devido a transações parciais tornam-se disponíveis a outros jobs. Estas alterações pendentes podem levar outros programas aplicativos a fazer alterações incorretas adicionais no banco de dados. Da mesma forma, qualquer recuperação de IPL decorrente que é executada posteriormente pode afetar contrariamente as alterações efetuadas por aplicativos após o job ter sido finalizado de modo anormal. Por exemplo, uma tabela SQL pode ser eliminada durante a recuperação de IPL como a ação de rollback para uma criação de tabela pendente. Porém, outros aplicativos talvez já tenham inserido várias linhas na tabela depois que a tarefa foi finalizada de modo anormal.

O sistema desempenha o seguinte para definições de confirmação sendo finalizadas durante uma finalização anormal de tarefa ou durante o próximo IPL após uma finalização anormal de sistema:

- Antes de finalizar uma definição de confirmação, o sistema executará uma operação de rollback implícito se a definição de confirmação possuir alterações pendentes, a menos que o processamento para a definição de confirmação tenha sido interrompido no meio de uma operação de confirmação. Se finalizada no meio de uma operação de confirmação, a transação pode ter o rollback efetuado, ser ressincronizada ou ser confirmada, dependendo do estado da mesma. O processamento para executar a operação implícita de rollback ou para completar a operação de confirmação inclui chamar o programa de saída de confirmação ou rollback da API para cada recurso de confirmação da API associado à definição de confirmação. Depois que a confirmação e rollback da API e o programa de saída forem chamados, o sistema removerá automaticamente o recurso de confirmação da API.

**Atenção:** A finalização do job quando uma transação está incerta (estado da transação é LAP ou PRP) pode causar inconsistências no banco de dados (alterações podem ser confirmadas em um ou mais sistemas e revertidas em outros sistemas).

- Se a opção de confirmação *Ação se Finalizar Job* for COMMIT, as alterações neste sistema serão confirmadas se o job for finalizado, sem considerar se as alterações nos outros sistemas que estão participando da transação foram confirmadas ou revertidas.
- Se a opção de confirmação *Ação se Finalizar Job* for ROLLBACK, as alterações neste sistema são revertidas se o job for finalizado, sem considerar se as alterações nos outros sistemas que estão participando da transação foram confirmadas ou revertidas.
- Se a opção de confirmação *Ação se Finalizar Job* for WAIT, o job não finaliza até que a ressincronização seja concluída para o sistema que toma a decisão de confirmação ou rollback. Para finalizar o job antes da conclusão da ressincronização, uma decisão heurística deve ser tomada e a ressincronização deve ser cancelada.

Não é recomendado terminar de modo anormal o job ou sistema durante uma recuperação de longa execução. Isto leva a ocorrência de outra recuperação assim que o job é encerrado (ou durante o próximo IPL se o sistema for encerrado). O rollback subsequente irá repetir o trabalho executado pelo rollback original e levará significativamente mais tempo para ser executado.

- Se um objeto de notificação estiver definido para a definição de confirmação, ele pode ser atualizado.

Se um processo for finalizado antes do controle de confirmação ser finalizado e as conversações protegidas ainda estiverem ativas, a definição de confirmação poderá ser requerida para confirmação ou rollback. A ação é baseada na opção Estado (State) e a opção Ação se finalizar job (Action if end job) para a definição de confirmação.

### Conceitos relacionados

“Operações Implícitas de Confirmação e Rollback” na página 59

Em algumas instâncias, uma operação de confirmação ou rollback é iniciada pelo sistema para uma definição de confirmação. Estes tipos de operações de confirmação e rollback são conhecidos como *pedidos implícitos de confirmação e rollback*.

“Atualizações Feitas no Objeto de Notificação”

O sistema atualizará o objeto de notificação com a identificação de confirmação da última operação de confirmação bem-sucedida dessa definição de confirmação.

## Atualizações Feitas no Objeto de Notificação

O sistema atualizará o objeto de notificação com a identificação de confirmação da última operação de confirmação bem-sucedida dessa definição de confirmação.

Para atingir os objetivos do objeto de notificação, são consideradas as seguintes alterações não confirmadas:

- Uma atualização num registro que seja estabelecido sob o controle de confirmação.
- Um registro excluído sob o controle de confirmação.
- Uma alteração de nível de objeto feita em um objeto de DDL local sob o controle de confirmação.
- Uma operação de leitura executada para um arquivo de banco de dados que foi aberto sob o controle de confirmação. Isto porque a posição do arquivo retorna ao último tempo limite de confirmação quando uma operação de rollback é executada. Se você executar uma operação de leitura sob o controle de confirmação, a posição do arquivo será alterada e, portanto, existirá uma alteração não confirmada para a definição de confirmação.
- Uma definição de confirmação com um dos seguintes recursos incluídos é sempre considerada como tendo alterações não confirmadas:
  - Um recurso de confirmação da API
  - Um recurso DRDA\* (Distributed Relational Database Architecture)
  - Um recurso da Arquitetura Distributed Database Management (DDM)
  - Um recurso da LU 6.2

O motivo disso é que o sistema não sabe quando uma alteração real é feita no(s) objeto(s) associado(s) a esses tipos de recursos. Tipos de recursos confirmáveis possuem informações adicionais sobre como incluir e trabalhar com esses tipos de recursos.

O sistema faz atualizações no objeto de notificação baseadas nas seguintes formas em que uma definição de confirmação pode ser finalizada:

- Se uma tarefa for finalizada normalmente e não existirem alterações não confirmadas, o sistema não colocará a identificação de confirmação da última operação de confirmação bem-sucedida no objeto de notificação.
- Se uma operação de confirmação implícita for desempenhada para uma definição de confirmação de nível de grupo de ativação quando o grupo de ativação for finalizado, o sistema não colocará a identificação de confirmação da última operação de confirmação bem-sucedida no objeto de notificação.

**Nota:** As operações de confirmação implícitas nunca são desempenhadas para a definição de confirmação \*DFACTGRP ou \*JOB.

- Se o sistema, a tarefa ou um grupo de ativação for finalizado anormalmente antes da primeira operação de confirmação bem-sucedida de uma definição de confirmação, o sistema não atualizará o objeto de notificação porque não existe a última identificação de confirmação. Para diferenciar entre esta condição e uma conclusão normal de programa, seu programa deve atualizar o objeto de notificação com uma entrada específica antes de concluir a primeira operação de confirmação bem-sucedida da definição de confirmação.

- Se ocorrer uma finalização de forma anormal do job ou do sistema depois de pelo menos uma operação de confirmação bem-sucedida, o sistema coloca a identificação de confirmação dessa operação no objeto de notificação. Se a última operação de confirmação bem-sucedida não especificou uma identificação de confirmação, o objeto de notificação não será atualizado. Para uma finalização anormal do job, este processamento do objeto de notificação é executado para cada definição de confirmação que estava ativa no job. Para uma finalização anormal do sistema, este processamento do objeto de notificação é executado para cada definição de confirmação que estava ativa em todos os jobs do sistema.
- O sistema atualizará o objeto de notificação com a identificação de confirmação da última operação de confirmação bem-sucedida dessa definição de confirmação se ocorrerem todos os eventos a seguir:
  - Um grupo de ativação não-padrão é encerrado.
  - Uma operação de rollback implícita for executada para a definição de confirmação de nível do grupo de ativação.
  - Pelo menos uma operação confirmar bem-sucedida foi executada para essa definição de confirmação.

Se a última operação de confirmação bem-sucedida não especificou uma identificação de confirmação, o objeto de notificação não será atualizado. Uma operação de rollback implícita será desempenhada para uma definição de confirmação de nível de grupo de ativação se o grupo de ativação estiver sendo finalizado anormalmente ou se ocorrerem erros ao fechar os arquivos abertos sob o controle de confirmação e que tiverem o escopo definido para esse grupo de ativação. Para obter informações adicionais sobre como alcançar arquivos de banco de dados para grupos de ativação e como os grupos de ativação podem ser finalizados, consulte o manual de referência da linguagem ILE que você está utilizando.

- Se existirem alterações não confirmadas quando um job finalizar normalmente e foi executada pelo menos uma operação confirmar bem-sucedida, a identificação de confirmação da última operação bem-sucedida será colocada no objeto de notificação e as alterações não confirmadas serão revertidas. Se a última operação de confirmação bem-sucedida não especificou uma identificação de confirmação, o objeto de notificação não será atualizado. O processamento deste objeto de notificação é executado para cada definição de confirmação que foi ativada para o job quando este finalizou.
- Se existirem alterações não confirmadas quando o comando ENDCMTCTL for executado, o objeto de notificação será atualizado somente se a última operação de confirmação bem-sucedida especificou uma identificação de confirmação:
  - Para um job batch, as alterações não confirmadas são revertidas e a identificação de confirmação da última operação de confirmação bem-sucedida é colocada no objeto de notificação.
  - Para um job interativo, se a resposta da mensagem de consulta CPA8350 for reverter as alterações, as alterações não confirmadas são revertidas e a identificação de confirmação da última operação de confirmação bem-sucedida é colocada no objeto de notificação.
  - Para um job interativo, se a resposta da mensagem de consulta CPA8350 for confirmar as alterações, o sistema indicará uma identificação de confirmação para utilizar e as alterações serão confirmadas. A identificação de confirmação fornecida na tela de prompt é colocada no objeto de notificação.
  - Para uma tarefa interativa, se a resposta à mensagem de consulta CPA8350 for cancelar o pedido ENDCMTCTL, as alterações pendentes permanecerão e o objeto de notificação não será atualizado.

### **Conceitos relacionados**

“Encerrando o Controle de Confirmação” na página 56

O comando End Commitment Control (ENDCMTCTL) encerra o controle de confirmação para a definição de confirmação de nível do job e de nível do grupo de ativação.

“Controle de Confirmação Durante Finalização do Grupo de Ativação” na página 58

O sistema finaliza automaticamente uma definição de confirmação do nível do grupo de ativação quando um grupo de ativação é finalizado.

“Controle de Confirmação Durante Finalização Normal da Etapa de Roteamento” na página 63

O sistema finaliza todas as definições de confirmação para um job quando uma etapa de roteamento é finalizada normalmente.



“Controle de Confirmação Durante Finalização Anormal do Sistema ou Job” na página 63  
O sistema finaliza todas as definições de confirmação para um job quando este finaliza de modo anormal. Estas definições de confirmação são finalizadas durante o processamento de finalização do job. Este tópico aplica-se somente às definições de confirmação com bloqueio de escopo do job.

“Tipos de Recursos Confirmáveis” na página 13

Esta tabela lista os diferentes tipos de recursos confirmáveis, incluindo FILE, DDL (Data Definition Language), DDM (Distributed Data Management), LU (Logical Unit) 6.2, DRDA (Distributed Relational Database Architecture), API e TCP.

“Objeto de Notificação de Confirmação” na página 52

Um *objeto de notificação* é uma fila de mensagens, área de dados ou arquivo de banco de dados contendo informações que identificam a última transação concluída com sucesso para uma determinada definição de confirmação, se ela não foi finalizada normalmente.

## Recuperação do Controle de Confirmação Durante Carregamento Inicial do Programa Após Término Anormal

Ao executar um IPL (Carregamento Inicial do Programa) após uma finalização anormal do sistema, o sistema tenta recuperar todas as definições de confirmação que estavam ativas quando o sistema foi finalizado.

Da mesma forma, ao ser ativado um conjunto de discos independente, o sistema tenta recuperar todas as definições de confirmação relacionadas àquele conjunto de discos independentes que estavam ativas quando ele foi desativado ou finalizado de modo anormal.

A recuperação é executada por jobs do servidor do banco de dados que são iniciados pelo sistema durante IPL. Os jobs do servidor do banco de dados são iniciados pelo sistema para manipular o trabalho que não pode ou não deve ser executado por outros jobs.

Os jobs do servidor do banco de dados são chamados de QDBSRVnn, onde nn é um número de dois dígitos. O número de jobs do servidor do banco de dados depende do tamanho do seu sistema. Da mesma forma, o nome do job do servidor do banco de dados para um conjunto de discos independentes ou grupo de conjuntos de discos independentes é QDBSxxxVnn, onde xxx é o número do conjunto de discos independentes e nn é um número de dois dígitos. Por exemplo, QDBS035V02 pode ser o nome do job do servidor do banco de dados para o conjunto de discos independentes 35.

Os estados da transação para o controle de confirmação de duas fases mostram as ações que o sistema toma, dependendo do estado da transação quando ocorreu o defeito. Para dois estados, PRP e LAP, a ação do sistema é incerta.

### Notas:

- As situações a seguir aplicam-se somente a definições de confirmação com bloqueios de escopo de job.
- O gerenciador de transação recupera as definições de confirmação associadas às transações XA (se seus bloqueios são com escopo de job ou com escopo de transação) utilizando APIs XA, não o processo de resincronização descrito neste tópico.

O sistema não pode determinar o que fazer até que execute a resincronização com outras localizações que participaram da transação. Esta resincronização é executada após a conclusão do IPL ou operação de ativação.

O sistema utiliza os jobs do servidor do banco de dados para executar esta resincronização. As definições de confirmação que devem ser recuperadas estão associadas aos jobs do servidor do banco de dados. Durante o IPL, o sistema adquire todos os bloqueios de registro e outros bloqueios de objetos que

eram mantidos pela definição de confirmação antes do sistema ser finalizado. Estes bloqueios são necessários para proteger os recursos de confirmação locais até que a ressincronização seja concluída e os recursos possam ser confirmados ou revertidos.

São enviadas mensagens para os logs dos jobs do servidor do banco de dados para indicar o status da ressincronização com as localizações remotas. Se a transação estiver incerta, a ressincronização deve ser concluída com a localização que possui a decisão para a transação, antes que recursos locais possam ser confirmados ou revertidos.

Quando é tomada a decisão para uma transação, as seguintes mensagens podem ser enviadas ao log de tarefa para a tarefa do servidor de banco de dados.

#### **CPI8351**

&1 alterações pendentes tendo rollback efetuado

#### **CPC8355**

A recuperação pós-IPL da definição de confirmação &8 para a tarefa &19/&18/&17 foi concluída.

#### **CPD835F**

Falha na recuperação IPL da definição de confirmação &8 para a tarefa &19/&18/&17.

Outras mensagens relacionadas à recuperação também podem ser enviadas. Estas mensagens são enviadas ao log histórico (QHST). Caso ocorram erros, as mensagens também são enviadas para a fila de mensagens QSYSOPR.

Você pode determinar o progresso da recuperação utilizando o System i Navigator, exibindo o log do job para o job do servidor do banco de dados ou utilizando o comando Work with Commitment Definitions (WRKCMTDFN). Embora com o System i Navigator e a exibição Work with Commitment Definitions você possa forçar o sistema a confirmar ou fazer roll back, é necessário utilizar isso apenas como último recurso. Se você prevê que todas as localizações que participaram da transação eventualmente serão retornadas à operação, deverá permitir que os sistemas se ressincronizem. Isto garante a integridade de seus bancos de dados.

#### **Conceitos relacionados**

“Estados da Transação para o Controle de Confirmação de Duas Fases” na página 30

Uma definição de confirmação é estabelecida em cada localização que faz parte da rede do programa de transação. Para cada uma, o sistema mantém o estado de sua transação atual e da anterior.

---

## **Gerenciando Transações e Controle de Confirmação**

Ao utilizar essas instruções, é possível exibir as informações do controle de confirmação e otimizar o desempenho para o controle de confirmação.

### **Exibindo Informações de Controle de Confirmação**

O System i Navigator pode exibir informações sobre todas as transações (unidades lógicas de trabalho) no sistema. O System i Navigator também pode exibir as informações sobre o job, se houver, associado a uma transação.

**Nota:** Estas operações de exibição não exibem o nível de isolamento para aplicativos SQL.

Para exibir informações, proceda da seguinte forma:

1. No System i Navigator, expanda o sistema que você deseja utilizar.
2. Expanda **Bancos de Dados**.
3. Expanda o sistema com o qual deseja trabalhar.
4. Expanda **Transações**.

**Nota:** Para exibir uma transação associada à uma transação global do X/Open, expanda **Transações Globais**. Para visualizar uma transação gerenciada do DB2, expanda as **Transações do Banco de Dados**.

5. Expanda **Transações Globais** ou **Transações do Banco de Dados**.

Esta exibição mostra as seguintes informações:

- ID da Unidade Trabalho
- Estado da Unidade de Trabalho
- Job
- Usuário
- Número
- Ressincronização em Andamento
- Definição de confirmação

A ajuda on-line fornece informações sobre todas as exibições de status e dos campos em cada exibição.

**Tarefas relacionadas**

“Detectando Conflitos” na página 114

Uma condição de conflito pode ocorrer quando um job mantém um bloqueio sobre um objeto, objeto A, e fica aguardando para obter um bloqueio sobre outro objeto, objeto B. Ao mesmo tempo, outro job ou transação mantém no momento um bloqueio sobre o objeto B e fica aguardando para obter um bloqueio sobre o objeto A.

“Recuperando Transações Após Falha em Comunicações” na página 115

Essas instruções ajudam a manipular as transações que executam trabalho em um sistema remoto após a falha na comunicação com esse sistema.

“Quando Forçar as Operações de Confirmação e de Rollback e Quando Cancelar a Ressincronização” na página 116

A decisão para forçar uma operação de confirmação ou de rollback é chamada de *decisão de heurística*. Esta ação permite que um operador confirme ou faça rollback manualmente dos recursos para uma transação que esteja em um estado preparado.

## **Exibindo Objetos Bloqueados para uma Transação**

Você pode exibir objetos bloqueados para transações globais somente com bloqueios com escopo de transação.

Para exibir objetos bloqueados para uma transação, siga estas etapas:

1. No System i Navigator, expanda o sistema que você deseja utilizar.
2. Expanda **Bancos de Dados**.
3. Expanda o sistema com o qual deseja trabalhar.
4. Expanda **Transações**.
5. Expanda **Transações Globais**.
6. Clique com o botão direito do mouse na transação com a qual deseja trabalhar e selecione **Objetos Bloqueados**.

**Tarefas relacionadas**

“Detectando Conflitos” na página 114

Uma condição de conflito pode ocorrer quando um job mantém um bloqueio sobre um objeto, objeto A, e fica aguardando para obter um bloqueio sobre outro objeto, objeto B. Ao mesmo tempo, outro job ou transação mantém no momento um bloqueio sobre o objeto B e fica aguardando para obter um bloqueio sobre o objeto A.

## Exibindo Jobs Associados a uma Transação

Para exibir os jobs associados a uma transação, siga estas etapas.

1. Na janela do System i Navigator, expanda o sistema que deseja utilizar.
2. Expanda **Bancos de Dados**.
3. Expanda o sistema com o qual deseja trabalhar.
4. Expanda **Transações**.
5. Expanda **Transações Globais** ou **Transações do Banco de Dados**.
6. Clique com o botão direito do mouse na transação com a qual deseja trabalhar e selecione **Jobs**.

Para transações do banco de dados e transações globais com bloqueios com escopo de job, uma lista dos jobs associados à transação é exibida.

Para transações globais com bloqueios com escopo de transação, é exibida uma lista dos jobs com este objeto de transação anexado ou aguardando para que este objeto de transação seja anexado

### Tarefas relacionadas

“Detectando Conflitos” na página 114

Uma condição de conflito pode ocorrer quando um job mantém um bloqueio sobre um objeto, objeto A, e fica aguardando para obter um bloqueio sobre outro objeto, objeto B. Ao mesmo tempo, outro job ou transação mantém no momento um bloqueio sobre o objeto B e fica aguardando para obter um bloqueio sobre o objeto A.

## Exibindo Satus do Recurso de uma Transação

Para exibir o status de recursos de uma transação, siga estas etapas.

1. Na janela do System i Navigator, expanda o sistema que deseja utilizar.
2. Expanda **Bancos de Dados**.
3. Expanda o sistema com o qual deseja trabalhar.
4. Expanda **Transações**.
5. Expanda **Transações Globais** ou **Transações do Banco de Dados**.
6. Clique com o botão direito do mouse na transação com a qual deseja trabalhar e selecione **Status do Recurso**.

## Exibindo as Propriedades da Transação

Para exibir as propriedades de transação, siga estas etapas.

1. Na janela do System i Navigator, expanda o sistema que deseja utilizar.
2. Expanda **Bancos de Dados**.
3. Expanda o sistema com o qual deseja trabalhar.
4. Expanda **Transações**.
5. Expanda **Transações Globais** ou **Transações do Banco de Dados**.
6. Clique com o botão direito do mouse na transação com a qual deseja trabalhar e selecione **Propriedades**.

## Otimizando o Desempenho para Controle de Confirmação

O uso de controle de confirmação requer recursos que podem afetar o desempenho do sistema. Vários fatores afetam o desempenho do sistema relativo ao controle de confirmação.

## Um Fator que Não Afeta o Desempenho

### Abrir um Arquivo

Se você abrir um arquivo sem especificar a opção de abertura de confirmação, nenhum recurso do sistema adicional será utilizado mesmo que uma definição de confirmação tenha sido iniciada.

Para obter informações adicionais sobre como especificar a opção de abertura de confirmação, consulte o manual de referência da linguagem de alto nível apropriado.

## Fatores que Degradam o Desempenho

**Diário** A criação de diário de um arquivo requer recursos do sistema. Porém, na maioria dos casos, a criação de diário funciona melhor com o controle de confirmação do que sem ele. Se você especificar somente imagens de depois, o controle de confirmação alterará as imagens antes e depois enquanto o controle de confirmação estiver em efeito. Geralmente, essa é uma consideração de espaço, não de desempenho.

### Operação de Confirmação

Se alguma alteração foi feita nos recursos criados em diário durante a transação, cada confirmação de uma transação incluirá duas entradas em cada diário relacionado a esses recursos. O número de entradas incluídas pode aumentar significativamente para uma volume grande de transações pequenas. Você pode colocar os receptores de diário em um conjunto de discos separado dos diários.

### Operação de Rollback

Como o controle de confirmação deve efetuar rollback das alterações pendentes registradas no banco de dados, recursos adicionais do sistema serão requeridos sempre que ocorrer um rollback. Além disso, se as alterações de registro estiverem pendentes, uma operação de rollback fará com que as entradas adicionais sejam incluídas no diário.

### Comandos Start Commitment Control (STRCMTCTL) e End Commitment Control (ENDCMTCTL)

Código extra adicional é incluído pelo sistema sempre que uma definição de confirmação é iniciada utilizando o comando STRCMTCTL e é finalizada utilizando o comando ENDCMTCTL. Evite o uso dos comandos STRCMTCTL e ENDCMTCTL para cada transação. Utilize-os somente quando necessário. Você pode estabelecer uma definição de confirmação no início de um job interativo e utilizá-la na duração do job.

### Utilizar Mais de um Diário para Transações do Controle de Confirmação

Com a confirmação de duas fases, os arquivos abertos sob o controle de confirmação podem ser registrados em mais de um diário. Porém, utilizar mais de um diário exige recursos adicionais do sistema para gerenciar a definição de confirmação. O uso de mais de um diário também pode tornar a recuperação mais complicada.

### Bloqueio de Registro

O bloqueio de registro pode afetar outros aplicativos. O número de registros bloqueados dentro de um determinado job aumenta os recursos integrais do sistema utilizados para o job. Aplicativos que precisem acessar o mesmo registro devem aguardar pela finalização da transação.

### Solicitar SEQONLY(\*YES)

Se você solicitar a opção SEQONLY(\*YES) (pelo comando OVRDBF ou se o programa aplicativo tentar implicitamente utilizar SEQONLY(\*YES)) e o arquivo foi aberto somente para entrada somente sob o controle de confirmação com LCKLVL(\*ALL), a opção será alterada para SEQONLY(\*NO). Essa opção pode afetar o desempenho de arquivos de entrada porque os registros não são bloqueados.

### | Solicite uma alteração de nível de registro para um arquivo de banco de dados quando o processamento Salvar enquanto ativo ou o processamento de quiesce ASP independente estiver ativo

| Um pedido para fazer uma alteração de nível de registro sob o controle de confirmação para um  
| arquivo de banco de dados poderá sofrer um atraso se a definição de confirmação estiver em um  
| limite de confirmação e uma operação para salvar enquanto ativo ou de quiesce ASP  
| independente estiver sendo executada em um job diferente. Para uma operação salvar enquanto  
| ativo, isso pode acontecer quando um arquivo é registrado no mesmo diário que alguns objetos  
| no pedido de gravação.

**Nota:** A coluna Status na exibição do comando Work with Active Jobs (WRKACTJOB) mostra CMTW (commit wait) quando um job estiver sendo suspenso para o processamento de ponto de verificação para salvar enquanto ativo.

### **Confirme ou execute rollback das alterações quando o processamento salvar enquanto ativo ou de quiesce ASP independente estiver ativo**

Uma operação de confirmação ou rollback pode sofrer um atraso em um limite de confirmação quando uma operação para salvar enquanto ativa ou uma operação de quiesce ASP independente estiver em execução em um job diferente. Isto pode ter ocorrido quando um recurso de confirmação da API foi incluso anteriormente na definição de confirmação, a menos as seguintes condições sejam verdadeiras:

- Para a operação salvar enquanto ativo, o recurso de API foi incluído utilizando a API Add Commitment Resource (QTNADDCR) e o campo Permitir Processamento de Salvamento Normal possui um valor Y.
- Para a operação de quiesce ASP independente, o recurso de API foi incluído utilizando a API QTNADDCR e o campo Permitir Quiesce ASP Independente possui um valor Y.

Como o job é suspenso durante o pedido de confirmação ou de rollback e como o pedido de confirmação ou de rollback pode ser executado apenas para uma única definição de confirmação por vez, os jobs com mais de uma definição de confirmação com recursos de confirmação de API podem impedir que uma operação salvar enquanto ativo ou operação de quiesce ASP independente seja concluída.

**Nota:** Se você utilizar o novo salvamento com o recurso de transações parciais, o objeto poderá ser salvo sem finalizar uma definição de confirmação.

### **Solicite uma alteração de nível de objeto quando o processamento salvar enquanto ativo ou o processamento de quiesce ASP independente estiver ativo**

Um pedido para fazer uma alteração de nível de objeto sob o controle de confirmação pode sofrer um atraso se a definição de confirmação estiver em um limite de confirmação e uma operação para salvar enquanto ativo ou uma operação de quiesce ASP independente estiver sendo executada em um job diferente. Isso pode acontecer quando uma alteração de nível do objeto for feita enquanto a operação salvar enquanto ativo ou operação de quiesce ASP independente for executada na biblioteca que contém o objeto. Por exemplo, a operação criar tabela SQL sob o controle de confirmação para a tabela MYTBL na biblioteca MYSQLLIB pode sofrer um atraso quando uma operação salvar enquanto ativo ou uma operação de quiesce ASP independente estiver em execução na biblioteca MYSQLLIB.

**Nota:** Se o tempo de espera exceder 60 segundos, a mensagem de consulta CPA8351 será enviada para perguntar ao usuário se a operação continuará aguardando ou será cancelada.

### **Incluir um Recurso da API Utilizando a API QTNADDCR**

Um pedido para incluir um recurso de confirmação da API utilizando a API QTNADDCR poderá sofrer um atraso se todas as definições de confirmação para o job estiverem no limite da confirmação e uma operação salvar enquanto ativo ou operação de quiesce APS independente estiver em execução em um job diferente.

#### **Notas:**

1. Se o tempo de espera exceder 60 segundos, a mensagem de consulta CPA8351 será enviada para perguntar ao usuário se a operação continuará aguardando ou será cancelada.
2. Para a operação salvar enquanto ativo, isso não se aplica aos recursos de API que foram incluídos utilizando a API QTNADDCR se o campo Permitir Processamento de Salvamento Normal tiver um valor Y. Para a operação de quiesce ASP independente, isso não se aplica aos recursos de API que foram incluídos utilizando a API QTNADDCR se o campo Permitir Quiesce ASP Independente tiver um valor Y.

## I Fatores que Aprimoram o Desempenho

### Utilizar um Diário Padrão

O uso de um diário padrão pode ajudar no desempenho se você fechar e abrir novamente todos os arquivos sob o controle de confirmação enquanto a definição de confirmação estiver ativa. Contudo, um diário padrão com OMTJRNE(\*NONE) piora o desempenho de operações de confirmação e rollback.

### Selecionar um Último Agente

O desempenho melhora quando um último agente é selecionado, pois menos interações entre o sistema e o último agente são necessárias durante uma operação confirmar. Porém, se ocorrer uma falha na comunicação durante uma operação de confirmação, ela não será concluída até que a ressincronização seja concluída, independentemente do valor da opção Aguardar pelo Resultado. Tal falha é rara, mas esta opção permite que o autor do aplicativo considere o impacto negativo de fazer com que o usuário espere indefinidamente pela conclusão da ressincronização quando uma falha ocorre. Compare com a melhoria de desempenho fornecida pela otimização do último agente durante operações de confirmação bem-sucedidas. Esta avaliação é geralmente mais significativa para jobs interativos do que para jobs batch.

O padrão é que um último agente tenha permissão de ser selecionado pelo sistema, mas o usuário possa modificar esse valor usando a API QTNCHGCO.

### Não Utilizar a Opção Aguardar pelo Resultado

Quando recursos remotos estão sob o controle de confirmação, o desempenho é aprimorado quando a opção Aguardar pelo Resultado é configurada como N (Não) e todos os sistemas remotos suportam a interrupção presumida. A opção Aguardar pelo Resultado é configurada como N pelo sistema para os aplicativos DRDA e DDM quando a primeira conexão é estabelecida com um sistema remoto. Os aplicativos APPC devem configurar explicitamente a opção Aguardar pelo Resultado, caso contrário será utilizado o valor padrão S.

### Selecionar a Opção OK para Omitir

O desempenho melhora quando a opção OK para Sair é selecionada.

### Selecionar a Opção Voto de Leitura

O desempenho melhora quando a opção Votar em é selecionada.

#### Conceitos relacionados

Gerenciamento de Diário

“Definição de Confirmação para Confirmação de Duas Fases: Indicar OK para Omitir” na página 40  
Normalmente, o gerenciador da transação em cada localização na rede do programa de transação participa de toda operação de confirmação ou rollback. Para melhorar o desempenho, você pode configurar algumas localizações ou todas elas em uma transação para permitir que o gerenciador da transação indique OK para Omitir.

“Definição de Confirmação para Confirmação de Duas Fases: Permitir Votar Somente Leitura” na página 33

Geralmente, um gerenciador de transação participa de ambas as fases do processamento de confirmação. Para melhorar o desempenho do processamento de confirmação, você pode configurar algumas ou todas as localizações em uma transação para permitir que o gerenciador de transação eleja como leitura.

#### Informações relacionadas

API Add Commitment Resource (QTNADDCR)

## Minimizando os Bloqueios

Uma forma típica de minimizar os bloqueios do registro é liberá-los. (Esta técnica não funciona se LCKLVL(\*ALL) foi especificado.)

Aqui está um exemplo para minimizar os bloqueios de registro liberando o bloqueio de registro. Um único aplicativo de manutenção de arquivo geralmente segue estas etapas:

1. Exibe um prompt para uma identificação de registro a ser alterada.
2. Recupera o registro solicitado.
3. Exibe o registro.
4. Permite que o usuário da estação de trabalho faça a alteração.
5. Atualiza o registro.

Na maioria dos casos, o registro é bloqueado do acesso do registro solicitado por meio da atualização. O tempo de espera do registro pode ser excedido por um outro job que esteja aguardando o registro. Para evitar o bloqueio de um registro enquanto o usuário da estação de trabalho está considerando uma alteração, libere o registro após sua recuperação do banco de dados (antes da exibição do registro aparecer). Em seguida, você precisará acessar o registro novamente antes de atualizar. Se o registro foi alterado entre a hora em que foi liberado e a hora em que foi acessado novamente, informe ao usuário da estação de trabalho. O programa poderá determinar se o registro foi alterado, salvando um ou mais campos do registro original e comparando-os aos campos no mesmo registro após a recuperação, conforme a seguir:

- Utilize um campo de contagem da atualização no registro e adicione 1 ao campo logo antes de uma atualização. O programa salva o valor original e compara-o ao valor no campo quando o registro for recuperado novamente. Se ocorreu uma alteração, o usuário da estação de trabalho será informado e o registro aparecerá novamente. O campo de contagem da atualização será alterado somente se ocorrer uma atualização. O registro será liberado enquanto o usuário da estação de trabalho estiver considerando uma alteração. Se você utilizar essa técnica, deverá usá-la em cada programa que atualizar o arquivo.
- Salve o conteúdo do registro de dados inteiro e compare-o ao registro na próxima vez em que for recuperado.

Nos dois casos anteriores, a seqüência de operações impede o uso simples de dados descritos externamente no RPG, onde os mesmos nomes de campo são utilizados no registro-mestre e no arquivo de exibição. Utilizar os mesmos nomes de campos (no RPG) não funciona porque as alterações do usuário da estação de trabalho são sobrepostas quando o registro é recuperado novamente. É possível resolver esse problema movendo os dados do registro para uma estrutura de dados. Ou, se você utilizar a palavra-chave DDS RTNDDTA, poderá continuar a utilizar os dados descritos externamente. A palavra-chave RTNDDTA permite que seu programa leia dados novamente no vídeo sem que o sistema operacional tenha que mover dados do vídeo para o programa. Isso permite que o programa execute as seguintes tarefas:

1. Solicite a identificação do registro.
2. Recupere o registro solicitado do banco de dados.
3. Libere o registro.
4. Salve o campo ou campos usados para determinar se o registro foi alterado.
5. Exiba o registro e aguarde pela resposta do usuário da estação de trabalho.

Se o usuário da estação de trabalho alterar o registro no vídeo, o programa utilizará a seguinte seqüência:

1. Recupera o registro do banco de dados novamente.
2. Compara os arquivos salvos para determinar se o registro do banco de dados foi alterado. Se alterado, o programa libera o registro e envia uma mensagem quando ele aparece.
3. Recupera o registro do vídeo executando uma operação de leitura com a palavra-chave RTNDDTA e atualiza-o no registro do banco de dados.
4. Continua no prompt lógico seguinte porque não existirão registros adicionais para liberação se o usuário da estação de trabalho cancelar o pedido.

LCKLVL(\*CHG) e LCKLVL(\*CS) funcionam nessa situação. Se LCKLVL(\*ALL) for utilizado, você deverá liberar o bloqueio de registro usando uma operação de confirmação ou rollback.

#### **Tarefas relacionadas**



“Detectando Conflitos” na página 114

Uma condição de conflito pode ocorrer quando um job mantém um bloqueio sobre um objeto, objeto A, e fica aguardando para obter um bloqueio sobre outro objeto, objeto B. Ao mesmo tempo, outro job ou transação mantém no momento um bloqueio sobre o objeto B e fica aguardando para obter um bloqueio sobre o objeto A.

## Gerenciando o Tamanho da Transação

Uma outra maneira de minimizar os bloqueios do registro é gerenciar o tamanho da transação.

Nesta discussão, uma transação é interativa. (O controle de confirmação também pode ser utilizado para aplicações em batch, que geralmente são consideradas como uma série de transações.) Muitas das mesmas considerações aplicam-se às aplicações em batch, abordadas em Controle de Confirmação para Aplicações em Batch.

Você pode bloquear no máximo 500.000.000 registros durante uma transação de cada diário associado a ela. Este limite pode ser reduzido com Query Options File (QAQQINI). Utilize o parâmetro QRYOPLIB do comando Change Query Attributes (CHGQRYA) para especificar Query Options File para um job utilizar. Utilize o valor COMMITMENT\_CONTROL\_LOCK\_LEVEL em Query Options File como limite de bloqueio para o job.

Ao selecionar o nível de bloqueio dos registros, considere o tamanho das transações. Utilize o tamanho para determinar por quanto tempo os registros ficarão bloqueados antes da finalização de uma transação. Você deve decidir se uma operação de confirmação ou rollback do controle de confirmação ficará limitada a um único uso da tecla Enter ou se a transação consistirá em vários usos da tecla Enter.

**Nota:** Quanto menor a transação, mais cedo o job que aguarda o início do processamento do ponto de verificação para salvar enquanto ativo poderá continuar e concluir.

Por exemplo, para um aplicativo de entrada do pedido, um cliente poderá solicitar vários itens em um único pedido que requer um registro de detalhe do pedido e uma atualização do registro-mestre do inventário para cada item no pedido. Se a transação estiver definida como o pedido inteiro e cada uso da tecla Enter solicitar um item, todos os registros envolvidos no pedido serão bloqueados enquanto durar o pedido inteiro. Por essa razão, os registros geralmente utilizados (como registros mestre de inventário) podem ser bloqueados por longos períodos de tempo, impedindo o andamento de outros trabalhos. Se todos os itens forem digitados com uma única tecla Enter utilizando um subarquivo, a duração dos bloqueios do pedido inteiro será minimizada.

Em geral, o número e a duração dos bloqueios devem ser minimizados para que vários usuários da estação de trabalho possam acessar os mesmos dados sem longos períodos de espera. Você pode fazer isto se não suspender os bloqueios enquanto o usuário estiver digitando dados na tela. Alguns aplicativos podem não precisar de mais de um usuário da estação de trabalho acessando os mesmos dados. Por exemplo, em um aplicativo de lançamento de dinheiro com diversos registros de item abertos por cliente, a abordagem típica é bloquear todos os registros e atrasá-los até que um usuário da estação de trabalho conclua o lançamento do dinheiro para um certo recibo.

Se o usuário da estação de trabalho pressionar a tecla Enter várias vezes para uma transação, será possível executá-la em diversos segmentos. Por exemplo:

- O primeiro segmento é uma consulta na qual o usuário da estação de trabalho solicita as informações.
- O segundo segmento é uma confirmação da intenção do usuário da estação de trabalho concluir a transação inteira.
- O terceiro segmento é a recuperação e a atualização dos registros afetados.

Esta abordagem permite que o bloqueio de registro fique restrito a um único uso da tecla Enter.

Esta abordagem da primeira consulta é normalmente utilizada em aplicativos em que uma decisão resulte das informações exibidas. Por exemplo, em um aplicativo de reserva de empresa aérea, um cliente talvez queira saber os horários de vôo, as conexões e os assentos disponíveis antes de tomar uma decisão sobre qual vôo tomar. Depois que o cliente toma uma decisão, a transação é digitada. Se a transação falhar (o vôo está cheio no momento), a função de rollback poderá ser utilizada e um pedido diferente digitado. Se os registros estiverem bloqueados desde a primeira consulta até a tomada da decisão, outro agente de reserva estará aguardando até que a outra transação seja concluída.

### Conceitos relacionados

“Controle de Confirmação para Aplicativos Batch” na página 27

As aplicações em batch podem ou não precisar de controle de confirmação. Em alguns casos, um aplicativo batch pode executar uma única função de leitura de um arquivo de entrada e atualização de um arquivo master. Porém, você pode usar o controle de confirmação para este tipo de aplicativo se for importante iniciá-lo novamente após uma finalização anormal.

## Confirmação Temporária

A *Confirmação Temporária* é uma forma de controle de confirmação que limita o número de vezes que o sistema grava entradas do diário associadas a uma transação em disco.

A confirmação temporária pode aprimorar o desempenho da transação, mas pode fazer com que uma ou mais transações sejam perdidas no caso de um defeito do sistema. O controle de confirmação tradicional no DB2 para i5/OS assegura a durabilidade da transação, o que significa que quando uma transação for confirmada, a transação persistirá no sistema. A confirmação temporária não fornece essa durabilidade, embora ainda assim assegure a atomicidade da transação. Ou seja, o sistema garante um limite de confirmação, mas uma ou mais transações completas podem ser perdidas no caso de um defeito do sistema.

Para utilizar a confirmação temporária, tanto para uma tarefa específica quanto através do sistema, especifique \*NO na variável de ambiente `QIBM_TN_COMMIT_DURABLE`. Essa variável pode ser alterada com o comando Add Environment Variable (ADDENVVAR).

Por exemplo, para solicitar a confirmação temporária de uma tarefa específica, execute o seguinte comando a partir da tarefa:

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

Para solicitar a confirmação temporária através do sistema, execute o seguinte comando:

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

**Nota:** Você deve ter autoridade \*JOBCTL para configurar essa variável de ambiente em todo o sistema.

Se a variável de ambiente `QIBM_TN_COMMIT_DURABLE` não tiver sido incluída, ou se a variável de ambiente tiver sido configurada para qualquer valor diferente de \*NO, o sistema não utilizará a confirmação temporária; em vez disso, utilizará o controle de confirmação tradicional para assegurar a durabilidade das transações.

Você pode verificar a existência dessa nova variável de ambiente, e seu valor e nível, se existirem, utilizando o comando Work with Environment Variables (WRKENVVAR).

Para algumas transações, o sistema operacional opta por ignorar seu pedido de confirmação temporária e, em vez disso, desempenha a confirmação tradicional. Isso ocorre em alguns ambientes complexos, em que várias conexões com o banco de dados são requeridas ou operações DDL estão em andamento. O sistema operacional pode determinar quando é apropriado desempenhar o pedido e quando faz mais sentido desempenhar uma operação de confirmação tradicional. Portanto, a solicitação da confirmação temporária não causa danos nesses ambientes.

---

## Cenários e Exemplos: Controle de Confirmação

Esses cenários e exemplos mostram como uma empresa configura o controle de confirmação. Os exemplos de código para programas que utilizam o controle de confirmação também são incluídos nessa coleta de tópico.

O cenário a seguir mostra como a empresa de brinquedos JKL implementa o controle de confirmação para rastrear transações em seu banco de dados local.

Os exemplos a seguir fornecem código de amostra para o controle de confirmação. O problema prático é um programa RPG que implementa o controle de confirmação. Inclui um fluxo lógico que mostra o que acontece em cada etapa.

### Cenário: Controle de Confirmação

A Fábrica de Brinquedos JKL utiliza o controle de confirmação para proteger os registros do banco de dados para manufatura e inventário. Este cenário mostra como a empresa de brinquedos JKL utiliza o controle de confirmação para transferir uma parte do departamento de inventário para o departamento de fabricação.

O tópico Cenário: gerenciamento de diário inclui uma descrição do ambiente de rede da Empresa de Brinquedos JKL. O cenário a seguir mostra como o controle de confirmação funciona em seu sistema de produção JKLPROD.

Este cenário ilustra as vantagens do uso do controle de confirmação em dois exemplos. O primeiro exemplo mostra como o programa de inventário da empresa, Programa A, pode funcionar sem o controle de confirmação e os possíveis problemas que podem ocorrer. O segundo exemplo mostra como o programa funciona com o controle de confirmação.

A Empresa de Brinquedos JKL utiliza um programa aplicativo de inventário, Programa A, em seu sistema JKLPROD. O Programa A utiliza dois registros. Um registro faz acompanhamento dos itens armazenados no estoque. Outro registro faz acompanhamento dos itens que são removidos do estoque e utilizados na produção.

### Programa A sem Controle de Confirmação

Suponha que o seguinte programa aplicativo não utiliza o controle de confirmação. O sistema bloqueia registros lidos para atualização. As seguintes etapas descrevem como o programa aplicativo acompanha um díodo a medida que é removido do estoque e transferido para conta corrente:

- O Programa A bloqueia e recupera o registro do estoque. (Essa ação poderá exigir uma espera, se o registro estiver bloqueado por outro programa.)
- O Programa A bloqueia e recupera o registro de produção. (Isso pode também exigir uma espera.) Agora, o Programa A tem os dois registros bloqueados e nenhum outro programa pode alterá-los.
- O Programa A atualiza o registro do estoque. Isto faz com o que o registro seja liberado para que fique disponível para ser lido para atualização por outro programa.
- O programa A atualiza o registro de produção. Isto faz com o que o registro seja liberado para que fique disponível para ser lido para atualização por outro programa.

Sem o uso do controle de confirmação, um problema precisa ser solucionado para que este programa funcione adequadamente em todas as circunstâncias. Por exemplo, ocorre um problema se o programa A não atualizar os dois registros devido a uma falha do job ou do sistema. Neste caso, os dois arquivos não estão consistentes -- díodos são removidos do registro do estoque, mas não são incluídos no registro da produção. O uso do controle de confirmação permite assegurar que todas as alterações envolvidas na transação estejam concluídas ou que os arquivos sejam retornados ao estado original caso o processamento da transação seja interrompido.

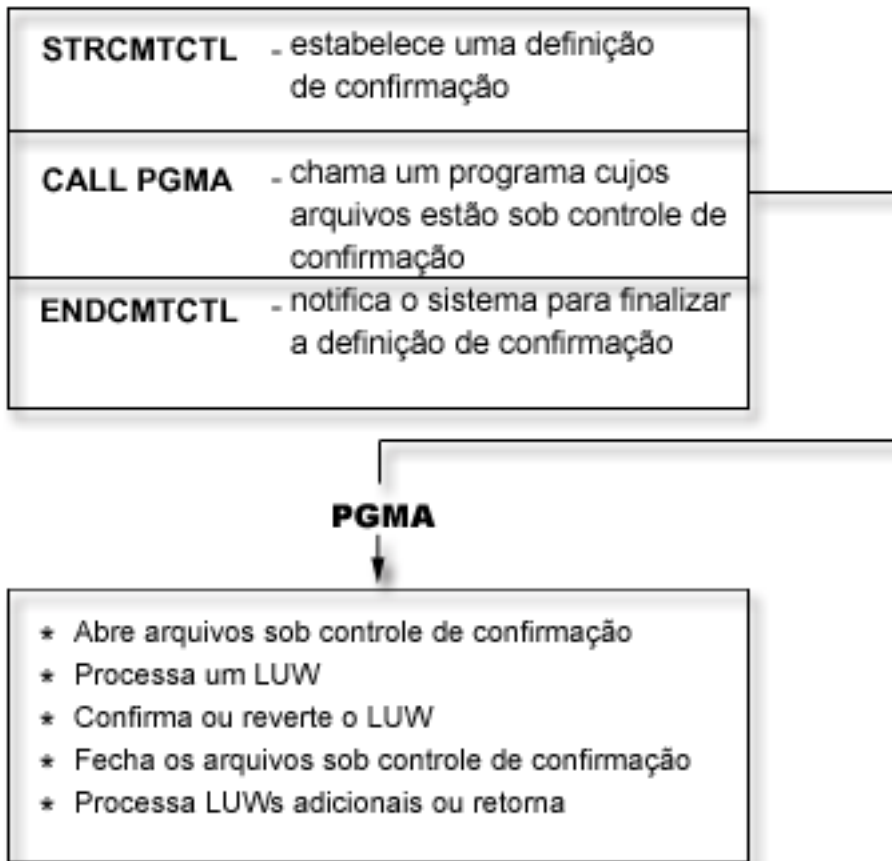
## Programa A com Controle de Confirmação

Se o controle de confirmação for utilizado, o exemplo anterior é alterado da seguinte forma:

1. O controle de confirmação é iniciado.
2. O Programa A bloqueia e recupera o registro do estoque. (Esta ação pode exigir uma espera se o registro estiver bloqueado por outro programa.)
3. O Programa A bloqueia e recupera o registro de produção. (Isto também pode exigir uma espera.) Agora, o Programa A tem os dois registros bloqueados e nenhum outro programa pode alterá-los.
4. O Programa A atualiza o registro do estoque e o controle de confirmação mantém o bloqueio no registro.
5. O Programa A atualiza o registro da produção e o controle de confirmação mantém o bloqueio no registro.
6. O Programa A confirma a transação. As alterações no registro do estoque e no registro da produção tornam-se permanentes nos arquivos. As alterações são registradas no diário, o que supõe que irão aparecer no disco. O controle de confirmação libera os bloqueios em ambos os registros. Agora, os registros estão disponíveis para serem lidos para atualização por qualquer outro programa.

Como os bloqueios em ambos os registros são mantidos pelo controle de confirmação até que a transação seja confirmada, não pode surgir uma situação na qual um registro é atualizado e o outro não. Se houver falha na etapa de roteamento ou do sistema antes da transação ser confirmada, o sistema remove (reverte) as alterações que foram feitas para que os arquivos fiquem atualizados até o ponto em que a última transação foi confirmada.

Para cada etapa de roteamento em que os arquivos devem estar sob controle de confirmação, ocorrem as etapas mostradas na seguinte figura:



As operações que são executadas sob controle de confirmação são registradas no diário. A entrada inicial do diário do controle de confirmação aparece após a primeira entrada de abertura de arquivo sob controle de confirmação. Isto porque a primeira entrada de abertura de arquivo determina qual diário é utilizado para o controle de confirmação. A entrada do diário da primeira operação de abertura é, então, utilizada para verificar operações de abertura subseqüentes a fim de assegurar que todos os arquivos estejam utilizando o mesmo diário.

Quando ocorre uma falha do job ou do sistema, os recursos sob o controle de confirmação são atualizados até um limite de confirmação. Se uma transação é iniciada mas não é concluída antes da finalização de uma etapa de roteamento, esta transação é revertida pelo sistema e não aparece no arquivo após a finalização da etapa de roteamento. Se houver uma finalização anormal do sistema antes da conclusão de uma transação, esta transação é revertida pelo sistema e não aparece no arquivo após um IPL (Carregamento Inicial do Programa) subseqüente, bem-sucedido, do código interno da licença. Sempre que ocorre um rollback, as entradas revertidas são colocadas no diário.

Por exemplo, suponha que a empresa JKL tenha 100 díodos em estoque. A manufatura retira 20 do estoque, para um novo balanço de 80. A atualização do banco de dados ocasiona duas entradas de diário, imagem anterior (100) e imagem posterior (80).

Suponha que houve uma finalização anormal do sistema após as entradas no diário, mas antes de atingir o ponto de confirmação ou de rollback. Após o IPL, o sistema lê a entrada do diário e atualizada o registro correspondente do banco de dados. Esta atualização produz duas entradas de diário que revertem a atualização: a primeira entrada é a imagem anterior (80) e a segunda a imagem posterior (100).

Quando o IPL é concluído com sucesso após a finalização anormal, o sistema remove (ou reverte) quaisquer alterações do banco de dados não confirmadas. No exemplo anterior, o sistema remove as alterações do registro do estoque pois não há no diário uma operação de confirmação para aquela transação. Neste caso, a imagem anterior do registro do estoque é colocada no arquivo. O diário contém as alterações revertidas e uma indicação da ocorrência de uma operação de rollback.

### Conceitos relacionados

Cenário: Gerenciamento de Diário

## Problema Prático para o Controle de Confirmação

Este problema prático pode ajudar a entender o controle de confirmação e seus requisitos. Estas etapas presumem que você esteja familiarizado com o programa licenciado i5/OS, o DFU (Data File Utility) e esta coleta de tópicos.

**Nota:** Utilizando os exemplos de código, você estará concordando com os termos das “Informações sobre o Código de Licença e Renúncia” na página 119.

Antes de começar este problema, siga estas etapas de pré-requisito:

1. Crie uma biblioteca especial para este problema prático. Nas instruções, a biblioteca chama-se CMTLIB. Substitua o nome da biblioteca onde você consulta CMTLIB.
2. Crie arquivos fonte e uma descrição de job.

Para utilizar o controle de confirmação, siga estas etapas:

1. Crie um arquivo físico chamado ITMP (arquivo mestre de item). A DDS (especificação da descrição de dados) para esse arquivo é a seguinte:

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

2. Crie um arquivo físico chamado TRNP (arquivo de transação). Esse arquivo é usado como um arquivo do log de transações. A DDS para esse arquivo é a seguinte:

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

3. Crie um arquivo lógico chamado TRNL (lógico de transação). Este arquivo é utilizado para ajudar a iniciar o aplicativo novamente. O campo **USER** é a seqüência de tipo LIFO. A DDS para esse arquivo é a seguinte:

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

4. Digite o comando STRDFU e crie um aplicativo DFU chamado ITMU para o arquivo ITMP. Aceite os padrões oferecidos pelo DFU durante a definição do aplicativo.
5. Digite o comando CHGDTA ITMU e digite os seguintes registros para o arquivo ITMP:

Item	Disponível
AA	450
BB	375
CC	4000

6. Encerre o programa pressionando F3. Esta entrada fornece alguns dados mediante os quais o programa operará.
7. Crie o Item Process (ITMPCSC) do programa CL da seguinte forma:

```

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

```

Este é o programa de controle que chama o programa ITMPCS. Ele recupera o nome do usuário e passa-o para o programa em processamento. Esse aplicativo supõe que sejam utilizados nomes de usuários exclusivos.

8. Crie um arquivo de tela chamado ITMPCSD no DDS da seguinte forma.

Existem dois formatos, o primeiro da tela de prompt básica e o segundo para permitir que o operador reveja a última operação digitada. Este arquivo de tela é utilizado pelo programa ITMPCS.

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

```

1.00      A          R PROMPT
2.00      A
3.00      A          CA03(93 'Finalização do programa')
4.00      A          CA04(94 'Rever último')
5.00      A          SETOFF(64 'Sem rcd para rvw')
6.00      A          1 2'INVENTORY TRANSACTIONS'
7.00      A          3 2'Quantidade'
8.00      A 61      QTY          5 0I  +1
9.00      A          ERRMSG('Inválido +
10.00     A          quantidade' 61)
11.00     A          +5'ITEM'
12.00     A          ITEM          2  I  +1
13.00     A 62      ERRMSG('Inválido +
14.00     A 63      Número do item' 62)
15.00     A          ERRMSG('Rollback +
16.00     A 64      ocorrida' 63)
17.00     A          24 2'CF4 foi pressionado +
18.00     A          e não existem +
19.00     A          transações para +
20.00     A          este usuário'
21.00     A          DSPATR(HI)
22.00     A          23 2'CF4 Rever última +
23.00     A          transação'
24.00     A          R REVW
25.00     A          1 2'INVENTORY TRANSACTIONS'
26.00     A          +5'REVIEW LAST TRANSACTION'
27.00     A          3 2'Quantidade'
28.00     A          QTY          5 0  +1EDTCDE(Z)
29.00     A          ITEM          2    +1

```

9. Analise o fluxo lógico fornecido em Fluxo Lógico do Programa Prático para o Controle de Confirmação.
10. Digite o comando STRSEU e digite a origem da seguinte forma:

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

```

1.00      FITMP  UF  E          K          DISK
2.00      F*
3.00      FTRNP  0  E          DISK
4.00      F*
5.00      FTRNL  IF  E          K          DISK
6.00      F          TRNR          KRENAMETRNR1
7.00      FITMPCSD CF E          WORKSTN
8.00      C* Digite o parâm. com Nome do usuário para arquivo -TRNP-
9.00      C          *ENTRY  PLIST
10.00     C          PARM          USER  10
11.00     C          LOOP    TAG
12.00     C          EXFMTPROMPT
13.00     C* Verifique CF3 para finalização do programa
14.00     C  93      DO          Finalização do Programa
15.00     C          SETON          LR
16.00     C          RETRN

```

```

17.00 C          END
18.00 C* Verifique CF4 para rever a última transação
19.00 C 94          DO          Rever última
20.00 C* Verifique a existência de um registro para este usuário no arquivo -TRNL-
21.00 C          USUÁRIO  CHAINTRNR1      64  Não encontrado
22.00 C 64          GOTO LOOP
23.00 C          EXFMTREVV
24.00 C          GOTO LOOP
25.00 C          END
26.00 C* Acessar Registro do item
27.00 C          ITEM    CHAINITMR      62  Não encontrado
28.00 C* Manipular Condição -não encontrado-
29.00 C 62          GOTO LOOP
30.00 C* Existe quantidade suficiente?
31.00 C          ONHAND  SUB QTY      TEST  50  61  Minus
32.00 C* Manipular quantidade insuficiente
33.00 C 61          DO
34.00 C* Liberar registro do item bloqueado por CHAIN para atualização
35.00 C          EXCPTRLSITM
36.00 C          GOTO LOOP
37.00 C          END
38.00 C* Alterar ONHAND e atualizar o Registro do item
39.00 C          Z-ADDTEST  ONHAND
40.00 C          UPDATITMR
41.00 C* Teste para Condições de Simulação Especiais
42.00 C          ITEM    IFEQ 'CC'
43.00 C* Programa simulado necessário para rollback
44.00 C          QTY     IFEQ 100
45.00 C          SETON      63  Rev. simultânea
46.00 C*          ROLBK
47.00 C          GOTO LOOP
48.00 C          END
49.00 C* Simular um cancelamento de programa anormal por Div por zero
50.00 C* Operador deve responder -C- para mensagem de consulta
51.00 C          QTD.     IFEQ 101
52.00 C          Z-ADD0      ZERO  30
53.00 C          TESTZ    DIV ZERO  TESTZ  30      Ocorre mensagem
54.00 C          END
55.00 C* Simular um cancelamento do job anormal por DSDPLY.
56.00 C* Operador deve fazer Pedido do Sistema para outro job
57.00 C* e cancelar este com OPTION(*IMMED)
58.00 C          QTD.     IFEQ 102
59.00 C          'CC=102' DSDPLY      Mensagem ocorre
60.00 C          END
61.00 C          END          ITEM=CC
62.00 C* Gravar no arquivo -TRNP-
63.00 C          WRITETRNR
64.00 C* Confirmar a atualiz. para -ITMP- e gravar em -TRNP-
65.00 C*          COMMIT
66.00 C          GOTO LOOP
67.00 OITMR  E          RLSITM

```

11. Digite o comando CRTRPGM para criar o programa ITMPCS a partir da origem digitada na etapa anterior.
12. Digite o comando CALL ITMPCSC, pressione Enter e F4. Uma mensagem declara que não há entradas para este operador.
13. Digite os dados a seguir para ver se o programa opera corretamente:

Quantidade	Item
3	AA
4	BB

14. Pressione F4. A exibição de revisão mostra o item BB inserido pela última vez. Digite os dados a seguir:



Quantidade	Item
5	FF (Ocorre uma mensagem de número de item inválido.)
9000	BB (Ocorre uma mensagem de erro de quantidade insuficiente.)
100	CC (Ocorre uma mensagem de rollback.)
102	CC (Deve ocorrer a operação RPG DSPLY. Pressione a tecla Enter.)
101	CC (O programa deve exibir uma mensagem de consulta informando que uma condição dividir por zero ocorreu ou foi finalizada, dependendo da definição do atributo de job INQMSGRPY. Se a mensagem de consulta aparecer, digite C para cancelar o programa RPG e, em seguida, C para cancelar o programa CL na consulta seguinte. Isto simula uma condição de erro inesperado).

15. Digite o comando Exibir Dados DSPDTA ITMP.

Verifique se os registros AA e BB foram atualizados corretamente. Os valores devem ser AA = 447, BB = 371 e CC = 3697. Observe que as quantidades subtraídas de CC ocorreram, mas os registros da transação não foram gravados.

16. Crie um receptor de diário para o controle de confirmação. Utilize o comando Create Journal Receiver (CRTJRNRCV) para criar um receptor de diário denominado RCVR1 na biblioteca CMRLIB. Especifique um limite de pelo menos 5000 KB. Recomenda-se um limite maior se seu sistema tiver espaço suficiente a fim de maximizar o tempo entre a geração de novos receptores de diário e minimizar os impactos de desempenho de diários com alterações muito freqüentes.

17. Crie um diário para o controle de confirmação. Utilize o comando Create Journal (CRTJRN) para criar um diário denominado JRNTEST na biblioteca CMTLIB. Como esse diário será utilizado somente para controle de confirmação, especifique MNGRCV(\*SYSTEM) DLTRCV(\*YES). Para o parâmetro JRNRCV, especifique o receptor de diário criado na etapa 16.

18. Utilize o comando Start Journal Physical File (STRJRNPF) com os parâmetros FILE(CMTLIB/ITMP CMTLIB/TRNP) JRN(CMTLIB/JRNTEST) para registrar em diário os arquivos para uso no controle de confirmação.

O parâmetro IMAGES utiliza um padrão de \*AFTER, significando que somente alterações after-image dos registros aparecerão no diário. Os arquivos ITMP e TRNP iniciaram a criação de diário.

Normalmente, você salva os arquivos depois de iniciar a criação de diário. Não é possível aplicar alterações criadas em um arquivo restaurado que não tenha o mesmo JID que as entradas do diário. Como este problema prático não requer que você aplique alterações criadas no diário, você poderá deixar de salvar os arquivos registrados em diário.

19. Digite o comando CALL ITMPCSC e as seguintes transações:

Quantidade	Item
5	AA
6	BB

Encerre o programa pressionando F3.

20. Digite o comando Display Journal: DSPJRN CMTLIB/JRNTEST.

Observe as entradas aparecendo no diário. A mesma seqüência de entradas (R UP = atualização de ITMP seguido de R PT = registro incluído no TRNP) ocorre no diário como foi executada pelo programa. Isto porque um arquivo lógico é definido pelo arquivo físico TRNP e o sistema substitui o RPG padrão. Se não houver um arquivo lógico, a suposição de RPG de SEQONLY(\*YES) será utilizada e um bloco de entradas PT aparecerá, porque os registros são mantidos no buffer do RPG até que o bloco esteja cheio.

21. Altere o programa CL ITMPCSC conforme a seguir (as novas instruções são mostradas com um asterisco).

```

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

```

O comando STRCMTCTL configura o ambiente de controle de confirmação. A palavra LCKLVL especifica que os registros lidos para atualização, mas não atualizados, podem ser liberados durante a transação. O comando MONMSG manipula qualquer mensagem de escape do RPG e executa um ROLLBACK no caso de o programa RPG finalizar de forma anormal. O comando ENDCMTCTL finaliza o ambiente de controle de confirmação.

22. Exclua o programa ITMPCSC existente e crie-o novamente.
23. Altere o programa RPG para remover os símbolos de comentário nas instruções 2.00, 4.00, 46.00 e 65.00. A origem agora está pronta para ser utilizada com o controle de confirmação.
24. Exclua o programa ITMPCS existente e crie-o novamente. O programa já está pronto para operar sob o controle de confirmação.
25. Digite o comando CALL ITMPCSC e as seguintes transações:

Quantidade	Item
7	AA
8	BB

26. Utilize o Pedido do Sistema e solicite a opção para exibir o job atual. Quando a tela Exibir Job aparecer, selecione a opção 16 para solicitar a exibição do status do controle de confirmação. Observe os valores na tela. Deve haver duas confirmação porque foram executadas duas instruções de confirmação no programa.
27. Pressione F9 para ver uma lista dos arquivos sob o controle de confirmação e a quantidade de atividade para cada arquivo.
28. Retorne ao programa e encerre-o pressionando F3.
29. Digite DSPJRN CMTLIB/JRNTST e observe as entradas dos arquivos e as entradas especiais do diário para o controle de confirmação:

Entrada	Significado
C BC	Comando STRCMTCTL ocorrido.
C SC	Inicia o ciclo de confirmação. Isto ocorre sempre que a primeira operação do banco de dados na transação causa a inserção, atualização ou exclusão de um registro como parte do controle de confirmação.
C CM	Ocorreu a operação de confirmação.
C EC	Comando ENDCMTCTL ocorrido.

As imagens "antes" e "depois" do controle de confirmação (tipos R UB e R UP) ocorrem automaticamente embora você tenha solicitado inicialmente IMAGES(\*AFTER) para o diário.

30. Digite o comando CALL ITMPCSC e as seguintes transações:

Quantidade	Item
12	AA
100	CC (Esta é a condição para simular a necessidade de utilizar o rollback de um aplicativo. O registro CC no arquivo ITMP, atualizado pela instrução RPG 40.00 é revertido.)

31. Pressione F4 para determinar a última transação digitada.

A última transação confirmada é a entrada do item AA.

32. Utilize o Pedido do Sistema e solicite a opção Exibir Job Atual. Quando a tela Exibir Job aparecer, solicite a exibição do status do controle de confirmação.

Observe os valores na tela e como eles foram alterados pelo rollback.

33. Retorne ao programa.  
34. Retorne à tela de prompt básico e encerre o programa pressionando F3.  
35. Digite o comando DSPJRN CMTLIB/JRNTEST.

Observe as entradas adicionais que aparecem no diário para uso da entrada de rollback (entrada C RB). Quando o registro ITMP é revertido, três entradas são colocadas no diário. Isto porque qualquer alteração no arquivo de banco de dados sob o controle de confirmação produz uma entrada antes (R BR) e depois (R UR).

36. Exiba as entradas com o código de diário R e estes tipos de entrada: UB, UP, BR e UR. Utilize a opção 5 para exibir as entradas completas. Como o campo **Quantidade** está decimal compactado, use F11 para solicitar uma exibição hexadecimal. Observe as seguintes condições:
- O valor disponível do registro ITMP no registro UB
  - Como o valor disponível é reduzido pelo registro UP
  - Como o registro BR é o mesmo que o registro UP
  - Como o registro UR retorna o valor como exibido originalmente para o registro UB

A última entrada é a entrada RB para a finalização de rollback.

37. Digite o comando CALL ITMPCSC; pressione Enter e pressione F4. Observe a última transação digitada.  
38. Digite as seguintes transações:

**Quantidade**

13  
101

**Item**

AA  
CC (Esta é a condição para simular uma condição de erro inesperado, que faz com que um programa finalize. A simulação ocorre dividindo um campo por 0. O programa exibirá uma mensagem de consulta ou finalizará, dependendo da definição do atributo do job INQMSGRPY. Se a mensagem de consulta aparecer, digite C para finalizar o programa. Como o programa CL foi alterado para monitorar erros do programa RPG, a segunda consulta que ocorreu não acontece.)

39. Digite o comando DSPJRN CMTLIB/JRNTEST.  
O mesmo tipo de manipulação de rollback ocorreu, mas desta vez o rollback foi causado pelo parâmetro EXEC do comando MONMSG no programa CL ao invés do programa RPG. Exiba as duas entradas RB para ver qual programa as causou.  
40. Digite o comando WRKJOB e anote o nome do job totalmente qualificado para ser utilizado mais tarde.  
41. Digite o comando CALL ITMPCSC e a seguinte transação:

**Quantidade**

14  
102

**Item**

AA  
CC (A operação RPG DSPLY deve ocorrer na fila de mensagens externa. Utilize a chave do Pedido do Sistema e selecione a opção 1 no menu de pedido de sistema para transferir para um job secundário.)

42. Conecte-se ao segundo job e restabeleça seu ambiente.  
43. Digite o comando ENDJOB e especifique o nome totalmente qualificado do job identificado anteriormente e OPTION(\*IMMED). Isto simula uma finalização anormal do job ou do sistema.

44. Aguarde cerca de 30 segundos, digite o comando CALL ITMPCSC e pressione F4.  
Observe a última transação confirmada. Deve ser o item AA inserido anteriormente.
45. Retorne à tela de prompt básico e encerre o programa pressionando F3.
46. Digite o comando DSPJRN CMTLIB/JRNTEST.  
O mesmo tipo de manipulação de rollback ocorreu, mas desta vez o rollback foi causado pelo sistema ao invés de um dos programas. A entrada RB foi gravada pelo programa QWTPITPP, que é o programa de finalização anormal de gerenciamento de trabalho.

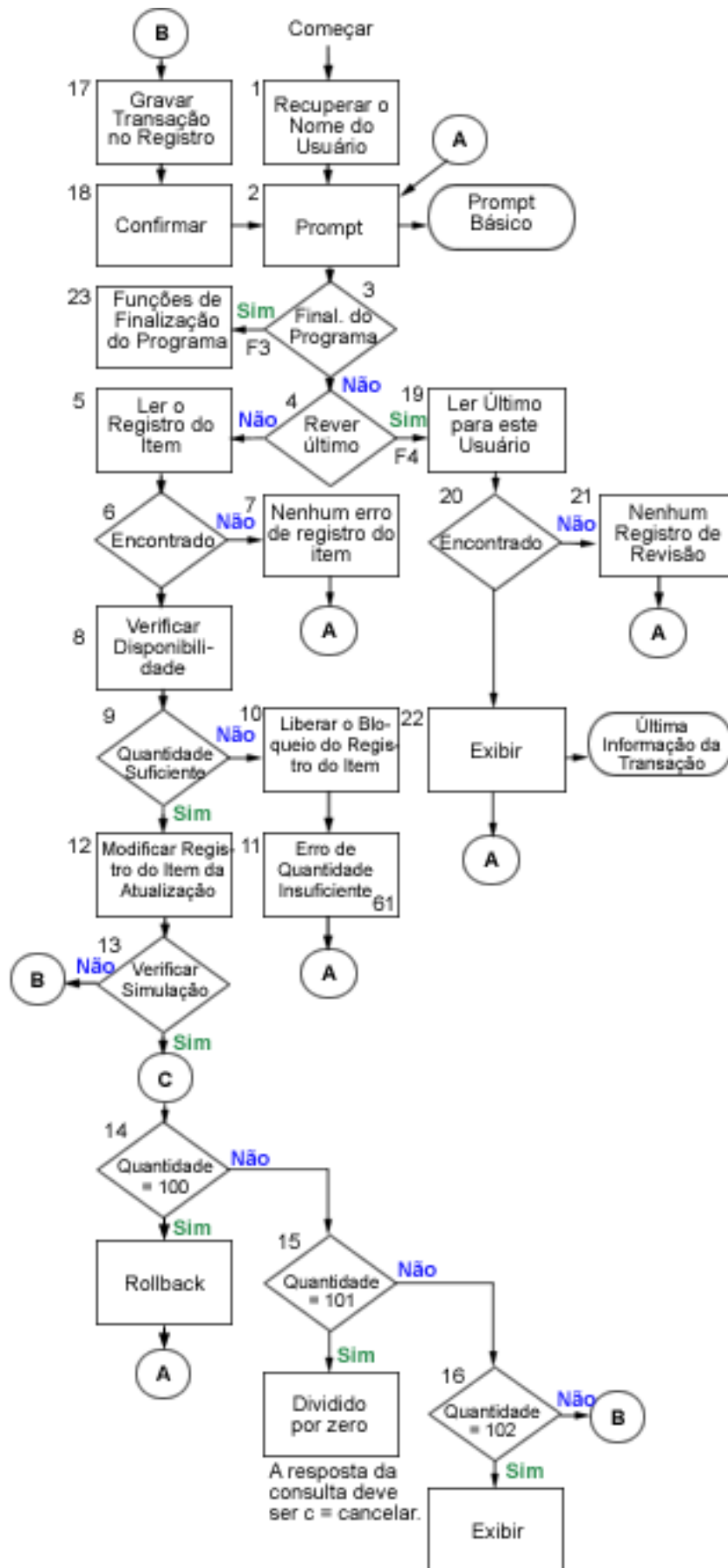
Você utilizou agora as funções básicas do controle de confirmação. Poderá prosseguir com o controle de confirmação nos aplicativos ou tentar alguma outra função, como:

- Uso de um objeto de notificação
- Bloqueio de registros que são lidos somente com LCKLVL(\*ALL)
- Bloqueio de vários registros no mesmo arquivo com LCKLVL(\*ALL)

### **Fluxo Lógico para o Problema Prático**

O fluxo lógico pode ajudá-lo a entender melhor este programa prático para o controle de confirmação. A figura a seguir mostra o fluxo do problema prático para o controle de confirmação.

Consulte “Etapas Associadas ao Fluxo Lógico para o Programa Prático” na página 88 para obter detalhes sobre cada uma das etapas mostradas na figura.



O operador deve fazer um pedido no sistema para receber um job separado e finalizar este.

## **Etapas Associadas ao Fluxo Lógico para o Programa Prático**

Estas etapas estão associadas ao fluxo lógico do problema prático.

1. Recupere o nome do usuário transmitido como um parâmetro. Utilizado para gravar no arquivo TRNP e também para recuperar a última transação digitada por cada operador. Este aplicativo supõe nomes de usuários exclusivos para operadores.
2. Solicite a exibição básica utilizando o nome do formato PROMPT.
3. Se F3 for pressionado, inicie uma função de finalização do programa.
4. Se F4 for pressionado, inicie uma rotina para acessar a última transação digitada pelo operador.
5. Leia o registro do item utilizando o campo ITEM. Como o arquivo é de atualização, este pedido bloqueia o registro.
6. Verifique se há uma condição não localizado no arquivo ITMP.
7. Se nenhum registro ITMP existir, ative o indicador 62 para causar a mensagem de erro e retorne para a etapa 2.
8. Subtraia a quantidade solicitada (QTY) de equilíbrio disponível (ONHAND) em uma área de trabalho.
9. Verifique se existe quantidade suficiente para atender ao pedido.
10. Se existir quantidade insuficiente, libere o bloqueio no registro no arquivo ITMP. Esta etapa será necessária em decorrência da quantidade insuficiente.
11. Ative o indicador 61 para sinalizar uma mensagem de erro de exibição de quantidade insuficiente e retorne para a 2.
12. Altere o campo ONHAND para o novo equilíbrio e atualize o registro ITMR.
13. Verifique a entrada especial no campo ITEM que pode ser utilizada para simular condições em que ROLLBACK é necessário.
14. Verifique QTY=100. Emita uma operação ROLLBACK. Isto simula uma condição em que o programa detecta a necessidade de rollback.
15. Verifique QTY=101. Causa uma exceção no programa que produzirá uma mensagem de consulta. Use dividir por zero para esta função. O operador deve digitar C para cancelar o programa a menos que a opção da descrição de job INQMSGRPH forneça uma resposta automática. Isto simula uma condição em que um erro inesperado ocorreu e o operador cancela o programa.
16. Verifique QTY=102. Emita uma exibição com operação de consulta. Isto pára o programa nesta etapa e permite o uso da tecla System Request para obter um job diferente. Cancele o job que está sendo atualizado. Isto simula uma condição em que uma finalização anormal de job ou sistema tenha ocorrido no meio de um tempo limite de confirmação.
17. Grave o registro de transação no TRNP.
18. Confirme os registros para a transação e retorne para a etapa 2.
19. Leia o primeiro registro no caminho de acesso do arquivo TRNL, utilizando USER como a chave. Como esse arquivo está na seqüência LIFO, este será o último registro de transação digitado por esse usuário.
20. Verifique se existe uma condição de registro não localizado no arquivo TRNL que seria causada se o arquivo não contivesse entradas para esse usuário.
21. Se não existir registro para esse usuário, ative o indicador 64 para causar uma mensagem de erro e retorne para a etapa 2.
22. Exiba a última transação digitada para esse usuário. Estas informações poderão ser utilizadas se o operador esquecer o que foi digitado anteriormente ou quando a transação for reiniciada. Quando o operador responder, retorne para a etapa 2.
23. Execute qualquer uma das funções do programa.

## Exemplo: Utilizando um Arquivo de Log de Transação para Iniciar um Aplicativo

Este exemplo fornece códigos amostra e instruções sobre como usar um arquivo de log de transação para iniciar um aplicativo após uma finalização anormal.

**Nota:** Utilizando os exemplos de código, você estará concordando com os termos das “Informações sobre o Código de Licença e Renúncia” na página 119.

Um *arquivo de log de transação* é utilizado para reiniciar um aplicativo após uma falha do sistema ou job, quando um objeto de notificação não é utilizado. Um arquivo de log de transação é geralmente utilizado em aplicativos interativos para resumir os efeitos de uma transação.

Por exemplo, em um aplicativo de entrada de pedidos, um registro é geralmente gravado em um arquivo de log de transação para cada item pedido. O registro contém o item pedido, a quantidade e o preço. Em um aplicativo de contas a pagar, um registro é gravado em um arquivo de log de transação para cada número de conta que deve receber um lançamento. Este registro normalmente contém informações como o número da conta, a quantia lançada e o fornecedor.

Em muitos dos aplicativos em que o arquivo de log de transação já existe, um usuário da estação de trabalho pode solicitar informações sobre a última transação digitada. Ao incluir o controle de confirmação nos aplicativos em que um arquivo de log de transação já existe, você pode:

- Assegurar-se de que os arquivos do banco de dados serão atualizados até um limite de confirmação.
- Simplificar o reinício da transação.

Você deve ter condições de identificar, exclusivamente, o usuário da estação de trabalho se utilizar o arquivo de log de transação para reiniciar aplicativos sob controle de confirmação. Se nomes de perfil de usuário exclusivos forem utilizados no sistema, estes podem ser colocados em um campo no registro de log de transação. Este campo pode ser usado como a chave para o arquivo.

Os exemplos a seguir supõem que um arquivo de inventário de pedidos está sendo usado para executar transações e que um arquivo de log de transação já existe. O programa executa as seguintes tarefas:

1. Solicita ao usuário da estação de trabalho uma quantidade e o número do item.
2. Atualiza a quantidade no arquivo master de produção (PRDMSTP).
3. Grava um registro no arquivo de log de transações (ISSLOGL).

Se a quantidade disponível do inventário for insuficiente, o programa rejeita a transação. O usuário da estação de trabalho pode perguntar ao programa onde a entrada de dados foi interrompida, pois o número do item, descrição, quantidade, nome do usuário e a data foram gravados no arquivo de log de transação.

### DDS para o Arquivo Físico PRDMSTP

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

1.00      A          R PRDMSTR          TEXT('Registro-mestre')
2.00      A          PRODC T           3          COLHDG('Número do' 'Produto')
3.00      A          DESCRP           20         COLHDG('Descrição')
4.00      A          ONHAND            5 0         COLHDG('Quantia' 'Disponível')
5.00      A                                     EDTCDE(Z)
6.00      A          K PRODC T
```

### DDS para o Arquivo Físico ISSLOGP Utilizado pelo ISSLOGP

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

1.00      A          R ISSLOGR          TEXT('Registro do log do produto')
2.00      A          PRODC T           3          COLHDG('Número do' 'Produto')
```

3.00	A	DESCRP	20	COLHDG('Descrição')
4.00	A	QTY	3 0	COLHDG('Quantidade')
5.00	A			EDTCDE(Z)
6.00	A	USER	10	COLHDG('Nome' 'Usuário')
7.00	A	DATE	6 0	EDTCDE(Y)
8.00	A			COLHDG('Data')

### DDS para o Arquivo 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		

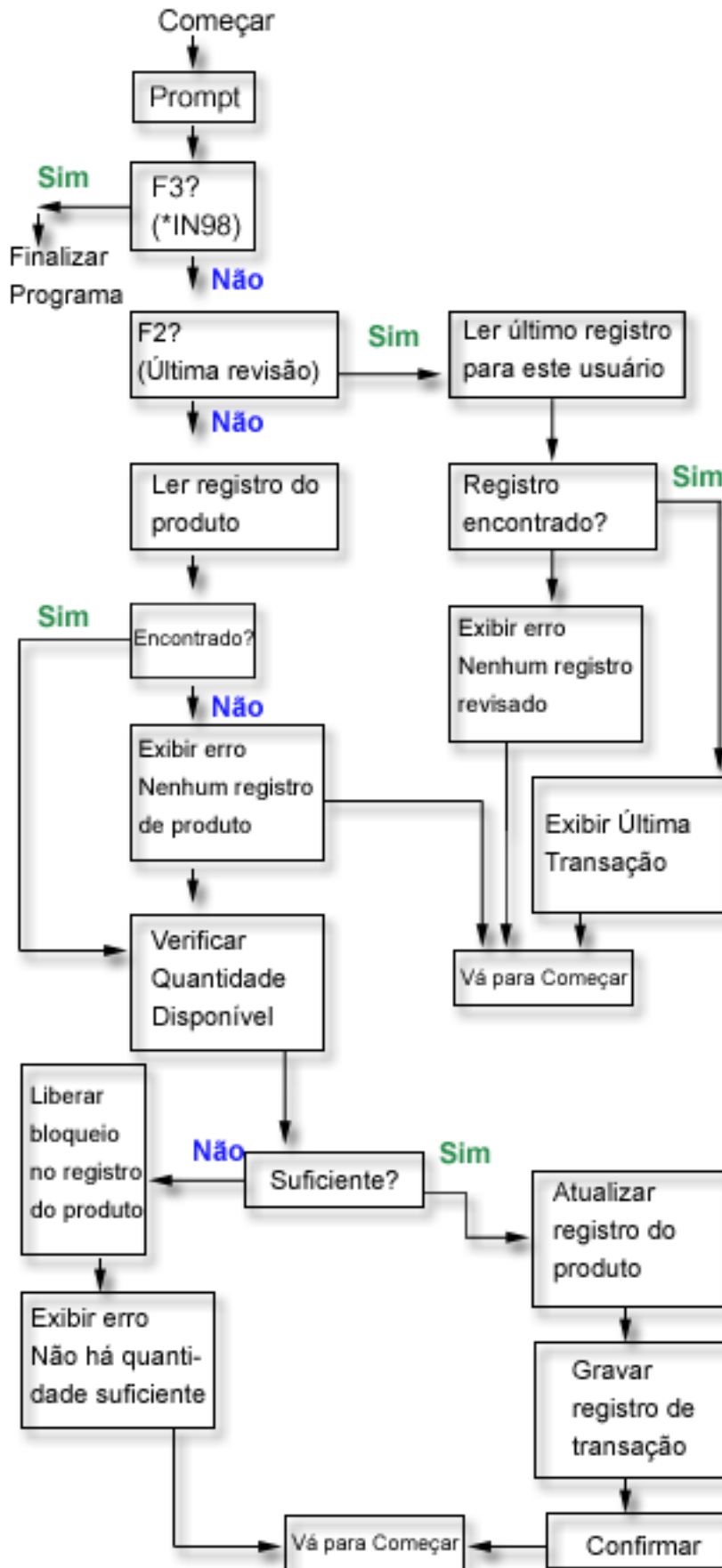
### DDS para o Arquivo de Exibição PRDISSD Utilizado no Programa

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

1.00	A					REF(ISSLOGP)
2.00	A	R PROMPT				
3.00	A					CA03(98 'Finalização do programa')
4.00	A					CA02(97 'Onde Estou')
5.00	A				1 20	'ISSUES PROCESSING'
6.00	A				3 2	'Quantidade'
7.00	A	QTY	R	I	+1	
8.00	A	62				ERRMSG('Insuficiente +
9.00	A					Qtde' 62)
10.00	A					+6'Produto'
11.00	A	PRODCT	R	I	+1	
12.00	A	61				ERRMSG('Nenhum registro +
13.00	A					produto encontrado' 62)
14.00	A	55			15 2	'Não há registro anterior'
15.00	A				24 2	'CF2 Últ. transação'
16.00	A	R RESTART				
17.00	A				1 20	'LAST TRANSACTION +
18.00	A					INFORMATION'
19.00	A				5 2	'Produto'
20.00	A	PRODCT	R		+1	
21.00	A				7 2	'Descrição'
22.00	A	DESCRP	R		+1	
23.00	A				9 2	'Qtde'
24.00	A	QTY	R		+1	REFFLD(QTY)

Este processo é descrito no **Fluxo do Programa**.





O código da operação RPG COMMIT é especificado depois que o arquivo PRDMSTP é atualizado e o registro é gravado no arquivo de log de transação. Como cada prompt para o operador representa um limite para uma nova transação, a transação é considerada uma única transação de Entrada.

O nome do usuário é passado ao programa quando é chamado. O caminho de acesso para o arquivo de log de transação é definido na seqüência LIFO (último a entrar primeiro a sair) para que o programa possa acessar, facilmente, o último registro digitado.

O usuário da estação de trabalho pode reiniciar o programa após uma falha do sistema ou do job utilizando a mesma função que identificou onde foi interrompida a entrada de dados. Nenhum código adicional precisa ser incluído no programa. Se você estiver utilizando atualmente um arquivo de log de transações, mas não estiver utilizando-o para descobrir onde você está, inclua o nome do usuário no arquivo de log de transações (supondo que os nomes de usuários sejam exclusivos) e utilize esta abordagem no programa.

O exemplo a seguir mostra o programa RPG utilizado. As instruções exigidas para controle de confirmação são marcadas com setas (==>).

### 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
6.00    PARM           USER   10
7.00    C*
8.00    C* Inicializar campos usados em Reg Log Trans
9.00    C
10.00   C
11.00   C* Loop de processamento básico
12.00   C
13.00   C          LOOP      TAG
14.00   C
15.00   C 98          GOTO END          Finalização do pgm
16.00   C 97          DO              Onde Estou
17.00   C
18.00   C          EXSR WHERE
19.00   C          GOTO LOOP
20.00   C
21.00   C          PRODC   CHAINPRDMSTR      61      Não encontrado
22.00   C 61          GOTO LOOP
23.00   C          ONHAND  SUB QTY      TEST  50  62  Menos que
24.00   C 62          DO              Insuficiente
25.00   C          EXCPTRLSMST          Liberar bloq
26.00   C          GOTO LOOP
27.00   C          END
28.00   C* Atualizar reg mestre e produzir Reg Log de Transação
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* Finalização do processo do programa
37.00   C*
38.00   C          END      TAG
39.00   C          SETON          LR
40.00   C*
41.00   C* subrotina WHERE para pedidos "Onde Estou"

```

```

42.00      C*
43.00      C          WHERE      BEGSR
44.00      C          USER      CHAINISSLOGL      55      Não encontrado
45.00      C      N55          EXFMTRESTART
46.00      C          ENDSR
47.00      OPRDMSTR E          RLSMST

```

## Programa CL Utilizado para Chamar o 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      RTVJOB      USER(&USER)
5.00      CALL        PRDISS PARM(&USER)
6.00      MONMSG      MSGID(RPG9001) EXEC(ROLLBACK)
7.00      ENDCMTCTL
8.00      ENDPGM

```

Para utilizar o controle de confirmação neste programa, normalmente um nível de bloqueio \*CHG é especificado. O registro é bloqueado pela alteração até que uma operação de confirmação seja executada. Observe que se houver uma quantidade insuficiente do inventário, o registro é liberado explicitamente. (Se o registro não for explicitamente liberado no programa, ele será liberado quando o próximo registro for lido para atualização a partir do arquivo.)

Neste exemplo, não há outra vantagem na utilização do nível de bloqueio \*ALL. Se \*ALL for utilizado, uma operação de rollback ou confirmação deve ser utilizada para liberar o registro quando houver uma quantidade insuficiente.

O código anterior é um programa CL que chama o programa RPG PRDISS. Observe o uso de comandos STRCMTCTL/ENDCMTCTL. O nome do usuário exclusivo é recuperado (comando RTVJOB) e passado ao programa. O uso do comando MONMSG para causar um rollback está descrito no Exemplo: Utilizar um Programa de Processamento Padrão para Iniciar um Aplicativo.

### Conceitos relacionados

“Exemplo: Utilizar um Programa de Processamento Padrão para Iniciar um Aplicativo” na página 100  
Um programa de processamento padrão é uma forma de iniciar seu aplicativo novamente utilizando um arquivo do banco de dados como objeto de notificação para todos os aplicativos. Esta abordagem supõe que os nomes do perfil do usuário são exclusivos para todos os aplicativos que utilizam o programa padrão.

“Exemplo: Objeto de Notificação Exclusivo para Cada Programa” na página 95

O uso de um objeto de notificação simples e exclusivo para cada tarefa permite o uso de uma identificação de confirmação descrita externamente, mesmo que existam vários usuários do mesmo programa.

## Exemplo: Utilizando um Objeto de Notificação para Iniciar um Aplicativo

Quando um programa é iniciado após uma finalização anormal, talvez procure uma entrada no objeto de notificação. Se a entrada existir, o programa poderá iniciar uma transação novamente. Depois de iniciada a transação, o programa limpa o objeto de notificação para impedir que se inicie a mesma transação em outra hora.

É possível utilizar um objeto de notificação das seguintes maneiras:

- Se a identificação de confirmação é colocada em um arquivo de banco de dados, consulte esse arquivo para determinar onde se iniciará novamente cada aplicativo ou job da estação de trabalho.

- Se a identificação de confirmação foi colocada em uma fila de mensagens de uma determinada estação de trabalho, uma mensagem poderá ser enviada para os usuários da estação de trabalho quando eles se conectarem para informá-los da última transação confirmada.
- Se a identificação de confirmação foi colocada em um arquivo de banco de dados que tem uma chave ou nome do usuário, o programa poderá ler este arquivo quando for iniciado. Se existir um registro no arquivo, inicie o programa novamente. O programa pode enviar uma mensagem para o usuário da estação de trabalho identificando a última transação confirmada. Toda recuperação será executada pelo programa. Se existir um registro no arquivo de banco de dados, o programa o excluirá na finalização do programa.
- Para um aplicativo batch, a identificação de confirmação pode ser colocada em uma área de dados que contém número total, definições da chave e outras informações sobre status necessárias para iniciar o aplicativo novamente. Quando o aplicativo é iniciado, ele acessa a área de dados e verifica os valores armazenados lá. Se o aplicativo finalizar normalmente, a área de dados será configurada para a próxima execução.
- Para um aplicativo batch, a identificação de confirmação pode ser enviada para uma fila de mensagens. Um programa executado quando o aplicativo é iniciado, pode recuperar as mensagens da fila e iniciar os programas novamente.

Você pode utilizar várias técnicas para iniciar seus aplicativos novamente, dependendo das necessidades de seu aplicativo. Ao escolher a técnica, considere as seguintes informações:

- Quando existem vários usuários de um programa ao mesmo tempo, uma única área de dados não pode ser utilizada como objeto de notificação porque, após a finalização anormal do sistema, a identificação de confirmação de cada usuário se sobreporá na área de dados.
- Seu design para excluir informações no objeto de notificação deve tratar a situação quando ocorre uma falha imediatamente após o uso das informações:
  - Se as informações forem excluídas imediatamente, elas não existirão se um outro defeito ocorrer antes do processamento da transação interrompida.
  - As informações no objeto de notificação não devem ser excluídas até o processamento bem-sucedido da transação interrompida. Neste caso, haverá mais de uma entrada no objeto de notificação se ele for um arquivo de banco de dados ou uma fila de mensagens.
  - O programa deverá acessar o último registro se houver mais de uma entrada.
- Não será possível utilizar um objeto de notificação para fornecer ao usuário da estação de trabalho a última transação confirmada, pois ele será atualizado somente se ocorrer uma falha no sistema ou no job ou se existirem alterações não confirmadas na finalização normal de um job.
- Se as informações forem exibidas para o usuário da estação de trabalho, deverão ser significativas. Isso pode requerer que o programa converta códigos mantidos no objeto de notificação em informações que ajudarão o usuário a iniciar novamente.
- As informações para iniciar novamente deverão ser exibidas se o usuário da estação de trabalho precisar delas. A lógica adicional no programa é requerida para impedir que as informações sejam exibidas novamente quando não forem mais significativas.
- Um objeto de notificação simples e um programa de processamento padrão poderão fornecer uma função para iniciar novamente se o objeto de notificação for um arquivo de banco de dados. Este programa de processamento padrão é chamado pelos programas que requerem a habilidade de iniciar novamente para minimizar as alterações de cada programa individual.

### **Conceitos relacionados**

“Objeto de Notificação de Confirmação” na página 52

Um *objeto de notificação* é uma fila de mensagens, área de dados ou arquivo de banco de dados contendo informações que identificam a última transação concluída com sucesso para uma determinada definição de confirmação, se ela não foi finalizada normalmente.

## Exemplo: Objeto de Notificação Exclusivo para Cada Programa

O uso de um objeto de notificação simples e exclusivo para cada tarefa permite o uso de uma identificação de confirmação descrita externamente, mesmo que existam vários usuários do mesmo programa.

Nos exemplos a seguir, um arquivo de banco de dados é utilizado como um objeto de notificação e somente nesse programa.

O programa tem dois arquivos de banco de dados (PRDMSTP e PRDLOCP) que devem ser atualizados para recibos do inventário. O arquivo de tela utilizado pelo programa chama-se PRDRCTD. Um arquivo de banco de dados, PRDRCTP, é usado como objeto de notificação. Ele é definido no programa com um arquivo e também é utilizado como definição de uma estrutura de dados para a função de notificação.

**Nota:** Utilizando os exemplos de código, você estará concordando com os termos das “Informações sobre o Código de Licença e Renúncia” na página 119.

### DDS para o Arquivo Físico PRDLOCP

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

1.00      A          R PRDLOCR          TEXT('Registro da localização')
2.00      A          PRODCT           3    COLHDG('Número do' 'Produto')
3.00      A          LOCATN           6    COLHDG('Localização')
4.00      A          LOCAMT           5 0   COLHDG('Quantidade da' 'Localização')
5.00      A          EDTCDE(Z)
6.00      A          K PRODCT
7.00      A          K LOCATN
```

### DDS para o Arquivo de Exibição PRDRCTD

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

1.00      A          REF(PRDMSTP)
2.00      A          R PROMPT
3.00      A          CA03(98 'Finalização do programa')
4.00      A          SETOFF(71 'RESTART')
5.00      A          1 20'PRODUCT RECEIPTS'
6.00      A          3  2'Quantidade'
7.00      A          QTY           3  OI  +1
8.00      A          +6'Produto'
9.00      A          PRODCT   R      I  +1
10.00     A  61          ERRMSG('Nenhum registro +
11.00     A          encontrado no +
12.00     A          arquivo-mestre' 62)
13.00     A          +6'Localização'
14.00     A          LOCATN   R      I  +1REFFLD(LOCATN PRDLOCP)
15.00     A  62          ERRMSG('Nenhum registro +
16.00     A          encontrado no +
17.00     A          arquivo de localização' 62)
18.00     A          9  2'Última Transação'
19.00     A  71          +6'Estas são informações +
20.00     A          de reinicialização'
21.00     A          DSPATR(HI BL)
22.00     A          12 2'Quantidade'
23.00     A          12 12'Produto'
24.00     A          12 23'Localização'
25.00     A          12 35'Descrção'
26.00     A          LSTPRD   R      14 15REFFLD(PRODCT)
27.00     A          LSTLOC   R      14 26REFFLD(LOCATN *SRC)
28.00     A          LSTQTY   R      14  5REFFLD(QTY *SRC)
29.00     A          EDTCDE(Z)
30.00     A          LSTDSC   R      14 35REFFLD(DESCRP)
```

## DDS para o Objeto de Notificação e a Estrutura de Dados 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           PRODCR      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

```

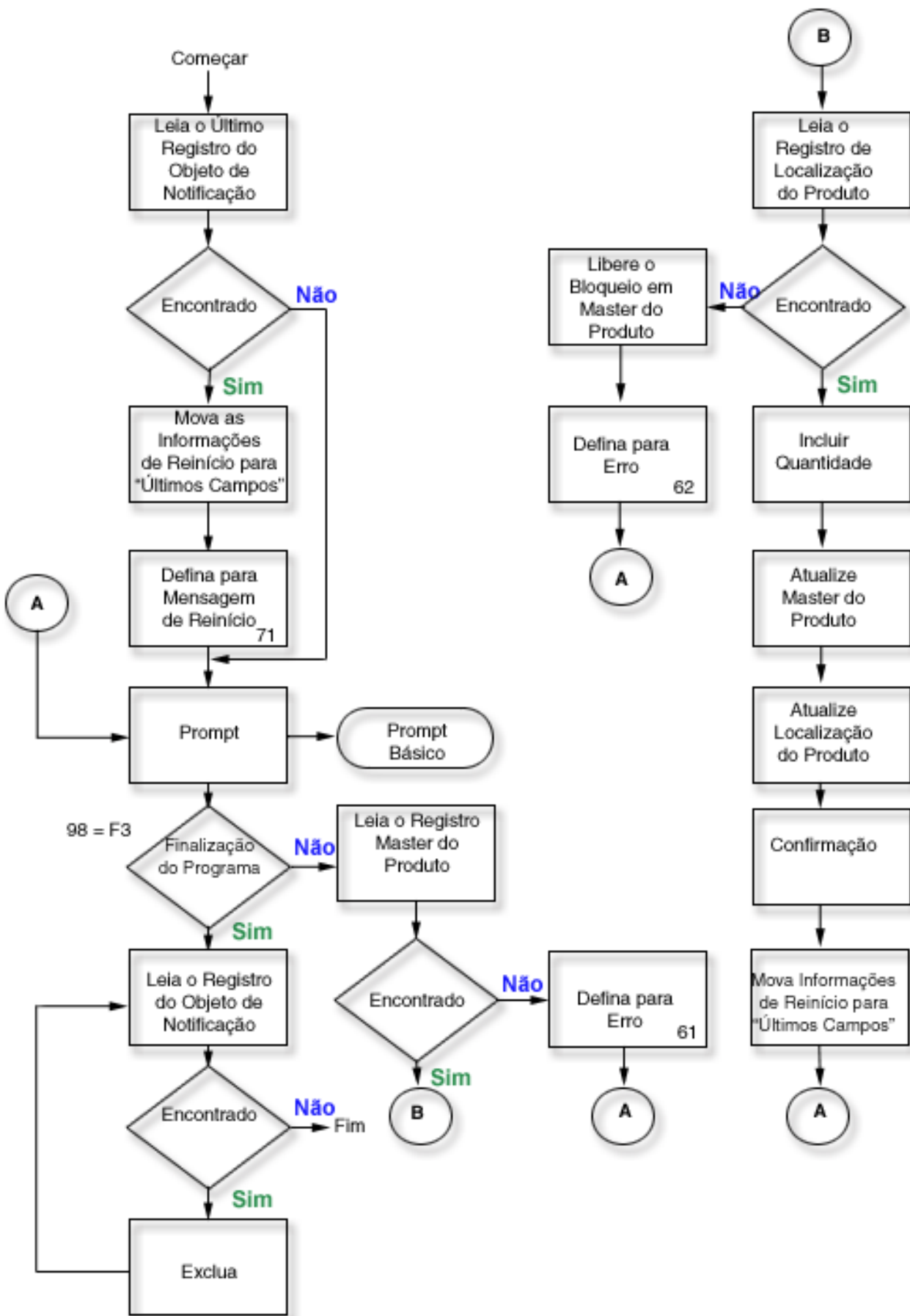
O programa processa o objeto de notificação da seguinte forma:

- No início, o programa processa aleatoriamente o objeto de notificação e exibe um registro, caso exista, para a chave específica:
  - Se existirem vários registros, o último registro dessa chave será utilizado pois o arquivo PRDRCTP está na seqüência LIFO.
  - Se não existir nenhum registro, uma transação não foi interrompida, portanto, não é necessário iniciar novamente.
  - Se o programa falhar antes da primeira operação confirmar bem-sucedida, ele não considera que seja necessário iniciar novamente.
- A rotina para limpar o objeto de notificação ocorre no final do programa:
  - Se ocorreram várias falhas, a rotina poderá manipular a exclusão de diversos registros no objeto de notificação.
  - Embora o sistema coloque a identificação de confirmação em um arquivo de banco de dados, ela deverá ser especificada como uma variável no programa RPG.
  - Como o RPG permite que uma estrutura de dados seja descrita externamente, uma estrutura de dados é uma forma conveniente de especificar a identificação de confirmação. Neste exemplo, a estrutura de dados utiliza a mesma descrição externa que o arquivo de banco de dados usou como objeto de notificação.

O processamento desse programa avisa o usuário sobre um número de produto, uma localização e uma quantidade:

- Dois arquivos devem ser atualizados:
  - Arquivo-mestre do produto (PRDMSTP)
  - Arquivo de localização do produto (PRDLOCP)
- Deve existir um registro em cada arquivo antes de cada um ser atualizado.
- O programa move os campos de entrada para os últimos campos correspondentes depois que cada transação é digitada com sucesso. Esses últimos campos são exibidos para o operador em cada prompt como feedback para o que foi digitado por último.
- Se as informações para iniciar novamente existirem, elas serão movidas para esses últimos campos e uma mensagem especial aparecerá no vídeo.

Este processo está esquematizado na figura a seguir. O nome do usuário é passado ao programa para fornecer um registro exclusivo no objeto de notificação.



O exemplo a seguir refere-se ao código fonte RPG. O objeto de notificação (arquivo PRDRCTP) é utilizado como arquivo normal no início e no final do programa e também é especificado como sendo o objeto de notificação no CL (comando STRCMTCTL) antes de chamar o programa.

## Origem 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* 0 arquivo seguinte é o objeto de notificação específico deste programa.
6.00      F*      É acessado somente em uma situação de reinício e no
7.00      F*      final do programa para excluir registros. Os registros
8.00      F*      são gravados no objeto de notificação pelo Controle de Confirmação.
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* Verificar as informações de reinício - obter último rcd por usuário
17.00     C*      O caminho de acesso ao arquivo PRDRCTP está na seqüência LIFO
18.00     C*
19.00     C          USER          CHAINPRDRCTR          20      Não encontrado
20.00     C N20          DO          Reiniciar
21.00     C          EXSR MOVLST          Mover para último
22.00     C          SETON          71      Reiniciar
23.00     C          END
24.00     C*
25.00     C* Loop de processamento básico
26.00     C*
27.00     C          LOOP          TAG
28.00     C          EXFMTPROMPT
29.00     C 98          GOTO END          Finalização do prog.
30.00     C          PRODC T          CHAINPRDMSTR          61      Não encontrado
31.00     C 61          GOTO LOOP
32.00     C          KEY          KLIST
33.00     C          KFLD          PRODC T
34.00     C          KFLD          LOCATN
35.00     C          KEY          CHAINPRDLOCR          62      Não encontrado
36.00     C 62          DO
37.00     C          EXCPTRLMS T          Bloqueio do release
38.00     C          GOTO LOOP
39.00     C          END
40.00     C          ADD QTY          ONHAND          Incluir
41.00     C          ADD QTY          LOCAM T
42.00     C          UPDATPRDMSTR          Atualizar
43.00     C          UPDATPRDLOCR          Atualizar
44.00     C*
45.00     C* Confirmar e mover para campos anteriores
46.00     C*
47.00     C          CMTID          COMIT
48.00     C          EXSR MOVLST          Mover para último
49.00     C          GOTO LOOP
50.00     C*
51.00     C* Finalização do processamento do programa
52.00     C*
53.00     C          END          TAG
54.00     C          SETON          LR
55.00     C*56.00     C* Excluir quaisquer registros no objeto de notificação
57.00     C*

```



58.00	C	DLTLP	TAG		
59.00	C	USER	CHAINPRDRCTR	20	Não encontrado
60.00	C	N20	DO		
61.00	C		DELETPRDRCTR		Excluir
62.00	C		GOTO DLTLP		
63.00	C		END		
64.00	C*				
65.00	C*	Mover para campos	-Últimos Usados-		para feedback do operador
66.00	C*				
67.00	C	MOVLST	BEGSR		
68.00	C		MOVE PRODC	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	OPRDMSTR	E	RLSMST		

### Conceitos relacionados

“Exemplo: Utilizando um Arquivo de Log de Transação para Iniciar um Aplicativo” na página 89  
Este exemplo fornece códigos amostra e instruções sobre como usar um arquivo de log de transação para iniciar um aplicativo após uma finalização anormal.

“Objeto de Notificação de Confirmação” na página 52

Um *objeto de notificação* é uma fila de mensagens, área de dados ou arquivo de banco de dados contendo informações que identificam a última transação concluída com sucesso para uma determinada definição de confirmação, se ela não foi finalizada normalmente.

“Exemplo: Objeto de Notificação Simples para Todos os Programas”

É vantagem utilizar um único objeto de notificação para todos os programas. É porque todas as informações necessárias para iniciar novamente estão no mesmo objeto e uma abordagem padrão para o objeto de notificação podem ser utilizadas em todos os programas.

### Exemplo: Objeto de Notificação Simples para Todos os Programas

É vantagem utilizar um único objeto de notificação para todos os programas. É porque todas as informações necessárias para iniciar novamente estão no mesmo objeto e uma abordagem padrão para o objeto de notificação podem ser utilizadas em todos os programas.

Nesta situação, utilize uma combinação exclusiva de identificações de usuário e programa para garantir que o programa acesse as informações corretas quando se iniciar novamente.

Como as informações requeridas para iniciar novamente podem variar de programa para programa, não utilize uma estrutura de dados descrita externamente para a identificação de confirmação. Se for utilizado um único objeto de notificação, o programa antecedente poderá descrever a estrutura de dados dentro do programa em vez de externamente. Por exemplo:

1	10	USER
11	20	PGMNAM
21	23	PRODC
24	29	LOCATN
30	49	DESC
50	51	0 QTY
52	220	DUMMY

Em cada programa que utiliza esse objeto de notificação, as informações especificadas para a identificação de confirmação são exclusivas para o programa (os nomes do usuário e do programa não são exclusivos). O objeto de notificação deve ser grande o suficiente para conter o máximo de informações que qualquer programa possa colocar na identificação de confirmação.

### Conceitos relacionados

“Exemplo: Objeto de Notificação Exclusivo para Cada Programa” na página 95

O uso de um objeto de notificação simples e exclusivo para cada tarefa permite o uso de uma identificação de confirmação descrita externamente, mesmo que existam vários usuários do mesmo programa.

“Objeto de Notificação de Confirmação” na página 52

Um *objeto de notificação* é uma fila de mensagens, área de dados ou arquivo de banco de dados contendo informações que identificam a última transação concluída com sucesso para uma determinada definição de confirmação, se ela não foi finalizada normalmente.

## Exemplo: Utilizar um Programa de Processamento Padrão para Iniciar um Aplicativo

Um programa de processamento padrão é uma forma de iniciar seu aplicativo novamente utilizando um arquivo do banco de dados como objeto de notificação para todos os aplicativos. Esta abordagem supõe que os nomes do perfil do usuário são exclusivos para todos os aplicativos que utilizam o programa padrão.

**Nota:** Utilizando o exemplo de código, você está concordando com os termos das “Informações sobre o Código de Licença e Renúncia” na página 119.

Para essa abordagem, o arquivo físico NFYOBJP é usado como objeto de notificação e definido como:

Nome excl. perfil usuário	10 caracteres	
Identificação do programa	10 caracteres	
Informações para iniciar novamente		Campo de Caractere (Deve ser grande para conter a quantidade máxima de informações para iniciar programas novamente que solicitam informações para reiniciar. Este campo é solicitado por programas aplicativos. No exemplo, supõe-se que tenha o comprimento de 200.)

O arquivo é criado com SHARE(\*YES). Os primeiros dois campos no arquivo são a chave do arquivo. (Esse arquivo também pode ser definido como uma estrutura de dados nos programas RPG.)

### Conceitos relacionados

“Exemplo: Utilizando um Arquivo de Log de Transação para Iniciar um Aplicativo” na página 89  
Este exemplo fornece códigos amostra e instruções sobre como usar um arquivo de log de transação para iniciar um aplicativo após uma finalização anormal.

## Exemplo: Código para um Programa de Processamento Padrão

Este exemplo mostra o código para um programa de processamento padrão que pode ser utilizado para iniciar o seu aplicativo novamente utilizando um arquivo de banco de dados como o objeto de notificação para todos os aplicativos.

O aplicativo mostrado no exemplos de códigos a seguir é executado assim:

1. O programa aplicativo recebe o nome do usuário em um parâmetro e utiliza-o com o nome do programa como um identificador exclusivo no objeto de notificação.
2. O programa aplicativo passa um código de pedido R ao programa de processamento de confirmação padrão, que determina se um registro existe no objeto de notificação.
3. Se ele retornar um código 1, o registro foi encontrado e o programa aplicativo apresentará ao usuário as informações necessárias para iniciar novamente.
4. O programa aplicativo prossegue com o processamento normal.
5. Quando uma transação é concluída, os valores são salvos para referência, de modo que o usuário da estação de trabalho possa ver o que foi feito na transação anterior.

As informações salvas não serão fornecidas pelo objeto de notificação porque ele será atualizado somente se ocorrer uma falha do job ou do sistema.

**Nota:** Utilizando os exemplos de código, você estará concordando com os termos das “Informações sobre o Código de Licença e Renúncia” na página 119.

### Exemplo de Programa Aplicativo

```

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 seguir, uma matriz de tempo de compilação que contém as
6.00      F* informações de reinício usadas no próximo exemplo
7.00      F*
8.00      E                      RTXT  50  50  1                      Reiniciar texto
9.00      I*
10.00     I* Estrutura de dados usada para inform. passadas ao objeto de notificação
11.00     I*
12.00     ICMTID          DS
13.00     I                      1  10  USER
14.00     I                      11  20  PGMNAM
15.00     I                      21  23  PRODCY
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 usados para comunicação com prog. padrão
25.00     C*
26.00     C                      MOVE USER10  USER
27.00     C                      MOVE 'PRDR2' PGMNAM
28.00     C                      MOVE 'R'      RQSCOD          Ler Rqs
29.00     C                      CALL 'STDCMT'
30.00     C                      PARM          RQSCOD  1
31.00     C                      PARM          RTNCOD  1
32.00     C                      PARM          CMTID 220          Estrut. dados
33.00     C                      RTNCOD  IFEQ '1'          Reiniciar
34.00     C                      EXSR MOVLST          Mover para último
35.00     C SETON                      71          Reiniciar
36.00     C                      END
37.00     C*
38.00     C* Inicializar campos usados no objeto de notificação
39.00     C*
40.00     C                      MOVEARTXT,1  RSTART          Mover texto
41.00     C*
42.00     C* Loop de processamento básico
43.00     C*
44.00     C          LOOP  TAG
45.00     C                      EXFMT PROMPT
46.00     C  98                      GOTO END
47.00     C          PRODCY CHAINPRDMSTR          61          Não encontrado
48.00     C  61                      GOTO LOOP
49.00     C          KEY    KLIST
50.00     C                      KFLD          PRODCY
51.00     C                      KFLD          LOCATN

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

52.00     C          KEY    CHAINPRDLOCR          62          Não encontrado
53.00     C  62                      DO
54.00     C                      EXCPTRLMSST          Liberar bloqueio
55.00     C                      GOTO LOOP
56.00     C                      END
57.00     C                      ADD  QTY    ONHAND          Incluir
58.00     C                      ADD  QTY    LOCAMT

```

59.00	C		UPDATPRDMSTR		Atualizar
60.00	C		UPDATPRDLOCR		Atualizar
61.00	C*				
62.00	C*		Confirmar e mover para campos anteriores		
63.00	C*				
64.00	C	CMTID	COMIT		
65.00	C		EXSR MOVLST		Mover para último
66.00	C		GOTO LOOP		
67.00	C*		Finalização do processamento do programa		
68.00	C*				
69.00	C	END	TAG		
70.00	C		MOVE 'D'	RQSCOD	Excl. Rqs
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 para campos -Últimos Usados- para feedback do 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	OPRDMSTR E		RLSMST		
86.00	** RTXT		Reiniciar Texto		
87.00			Menu Inventário - Opção de Recibos		

### Fluxo de Processamento:

O programa padrão é chamado a partir dos aplicativos e deve ser iniciado novamente.

Os programas aplicativos passam esta lista de parâmetros para o programa padrão:

- Código de pedido
- Código de retorno
- Nome da estrutura de dados (o conteúdo do objeto de notificação)

Os códigos de pedido executam as seguintes operações:

- R (Ler)
 

Recupera o último registro incluído no objeto de notificação com a mesma chave. O código de retorno é definido como:

  - 0 Nenhum registro está disponível (não é necessário iniciar novamente).
  - 1 Registro retornado no campo de informações para iniciar novamente (é necessário iniciar novamente).
- WA (Gravar)
 

Grava um registro no arquivo. Esse código pode ser utilizado, se você utiliza um objeto de notificação para fins próprios. Por exemplo, se o programa determina que a transação precisa ser iniciada novamente, o programa pode gravar um registro para o objeto de notificação para simular o que o sistema fará se um job ou o sistema falhar.
- DE (Excluir)
 

Exclui todos os registros do objeto de notificação com a mesma chave. O código de retorno é definido como:

  - 0 Não existem registros para exclusão.
  - 1 Um ou mais registros foram excluídos.

- OE (Abrir)  
O código de pedido O é opcional e é utilizado para evitar que se tenha que iniciar o programa de processamento sempre que for chamado.
- CA (Fechar)  
Depois que o código de pedido de abertura é utilizado, o código de pedido de fechamento assegura que o arquivo seja fechado.
- SA (Procurar)  
Retorna o último registro deste usuário. O nome do programa não é utilizado. Este código pode ser usado em um programa inicial para determinar se será necessário iniciar novamente.

### **Exemplo: Código para um Programa de Processamento de Confirmação Padrão**

O Pprograma de Processamento de Confirmação Padrão (STDCMT) executa as funções necessárias para estabelecer comunicação com um objeto de notificação exclusivo utilizado por todos os aplicativos.

Enquanto a função de controle de confirmação grava automaticamente uma entrada no objeto de notificação, um programa padrão gravado pelo usuário deve processá-lo. O programa padrão simplifica e padroniza a abordagem.

O programa é gravado para verificar os parâmetros que foram passados e executar a ação apropriada, da seguinte forma:

#### **O=Abrir**

O programa chamador solicita que o arquivo do objeto de notificação fique aberto no retorno. Como o objeto de notificação é aberto implicitamente pelo programa RPG, o programa não deverá fechá-lo. O indicador 98 é configurado de modo que o programa retorne com LR desativado, para manter as áreas de trabalho do programa, e deixe o objeto de notificação aberto para que possa ser chamado novamente sem código extra em excesso.

#### **C=Fechar**

O programa de chamada determinou que não precisa mais do objeto de notificação e solicita um fechamento. O indicador 98 é desativado para permitir um fechamento total do objeto de notificação.

**R=Ler** O programa chamador solicita que um registro com campos-chave correspondentes sejam lidos e devolvidos. O programa utiliza os campos-chave passados para tentar recuperar um registro de NFOBJP. Se existirem registros duplicados para a mesma chave, o último registro retornará. O código de retorno é definido na mesma proporção e se o registro existia, será devolvido na estrutura de dados CMTID.

#### **W=Gravar**

O programa chamador solicita que um registro seja gravado no objeto de notificação para permitir que ele se inicie novamente na próxima vez que for chamado. O programa grava o conteúdo dos dados passados como um registro no NFOBJP.

#### **D=Excluir**

O programa chamador solicita que os registros desta chave correspondente sejam excluídos. Essa função é executada normalmente na conclusão bem-sucedida do programa de chamada para remover qualquer informação sobre novo início. O programa tenta excluir todos os registros dos campos-chave passados. Se nenhum registro existir, será devolvido um código de retorno diferente.

#### **S=Procurar**

O programa chamador solicita uma pesquisa para um registro para um usuário em particular, independente de qual programa o gravou. Esta função é usada no programa para sign on para indicar que será necessário iniciar novamente. O programa utiliza somente o nome do usuário como chave para ver se existem registros. O código de retorno está definido adequadamente e o conteúdo do último registro desta chave (se existir) é lido e devolvido.

O exemplo a seguir mostra o programa de processamento de confirmação padrão, STDCMT.

### Programa de Processamento de Confirmação Padrão

**Nota:** Utilizando o exemplo de código, você está concordando com os termos das “Informações sobre o Código de Licença e Renúncia” na página 119.

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

```

1.00      FNFYOBJP UF  E      K      DISK      A
2.00      ICMTID      DS
3.00      I                      1  10 UNQUSR
4.00      I                      11  20 UNQPGM
5.00      I                      21  220 BIGFLD
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 Inválido
11.00     C          UNQPGM  CABEQ*BLANKS  BADEND      H2 Inválido
12.00     C*
13.00     C*  '0' para Aberto
14.00     C*
15.00     C          RQSCOD  IFEQ '0'      Aberto
16.00     C          SETON      98      Finalizar LR
17.00     C          GOTO END
18.00     C          END
19.00     C*
20.00     C*  'C' para Fechar
21.00     C*
22.00     C          RQSCOD  IFEQ 'C'      Fechar
23.00     C          SETOF      98
24.00     C          GOTO END
25.00     C          END
26.00     C*
27.00     C*  'R' para Ler - Obter último registro da chave
28.00     C*
29.00     C          RQSCOD  IFEQ 'R'      Ler
30.00     C          KEY      KLIST
31.00     C          KFLD      UNQUSR
32.00     C          KFLD      UNQPGM
33.00     C          KEY      CHAINNFYOBJR  51      Não 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 Gravar
44.00     C*
45.00     C          RQSCOD  IFEQ 'W'      Gravar
46.00     C          WRITENFYOBJR
47.00     C          GOTO END
48.00     C          END
49.00     C*
50.00     C*  'D' para Excluir - Excluir todos os registros da chave
51.00     C*
52.00     C          RQSCOD  IFEQ 'D'      Excluir
53.00     C          KEY      CHAINNFYOBJR  51      Não 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 Pesquisar o último registro deste usuário
65.00 C* (Ignorar a parte -Programa- da chave)
66.00 C*
67.00 C RQSCOD IFEQ 'S' Pesquisar
68.00 C UNQUSR SETLLNFYOBJR 20 Se 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* Processamento de código de pedido inválido
79.00 C*
80.00 C SETON H2 Código RQS inválido
81.00 C GOTO BADEND
82.00 C*
83.00 C* Finalização do processamento do programa
84.00 C*
85.00 C END TAG
86.00 C N98 SETON LR
87.00 C RETRN
88.00 C* A marcação BADEND foi utilizada e depois fracassou no retorno de erro do ciclo RPG
89.00 C BADEND TAG

```

### Conceitos relacionados

“Exemplo: Utilizar um Programa de Processamento Padrão para Decidir Reiniciar ou Não o Aplicativo”

O programa inicial pode chamar o programa de processamento de confirmação padrão para determinar se é necessário iniciar novamente. O usuário da estação de trabalho pode então decidir se deve iniciar novamente.

### Exemplo: Utilizar um Programa de Processamento Padrão para Decidir Reiniciar ou Não o Aplicativo

O programa inicial pode chamar o programa de processamento de confirmação padrão para determinar se é necessário iniciar novamente. O usuário da estação de trabalho pode então decidir se deve iniciar novamente.

O programa inicial passa um código de solicitação de S (procurar) para o programa padrão, que procura qualquer registro para o usuário. Caso exista um registro, as informações para o reinício são passadas para o programa inicial e as informações são exibidas ao usuário da estação de trabalho.

A identificação de confirmação no objeto de notificação contém informações que o programa inicial pode exibir, identificando qual programa precisa ser iniciado novamente. Por exemplo, os últimos 50 caracteres da identificação de confirmação podem ser reservados para conter estas informações. No programa aplicativo, essas informações podem estar em uma matriz de tempo de compilação e ser movidas para a estrutura de dados em uma etapa de inicialização. Exemplo: o código para um programa de processamento de confirmação padrão mostra como incluir isso no programa aplicativo.

O exemplo a seguir mostra um programa inicial, que determina se um registro existe no objeto de notificação.

## Exemplo: Programa Inicial

**Nota:** Utilizando o exemplo de código, você está concordando com os termos das “Informações sobre o Código de Licença e Renúncia” na página 119.

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

1.00      PGM
2.00      DCLF      CMTINLD
3.00      DCL      &RQSCOD *CHAR LEN(1) VALUE(S) /* Search */
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     /* 0 XX é necessário p/ impedir um Nom Pgm em branco */
11.00     CALL     STDCMT PARM(&RQSCOD &RTNCOD &CMTID)
12.00     IF      (&RTNCOD *EQ '1') DO /* RESTART REQD */
13.00     CHGVAR   &INFO %SST(&CMTID 171 50)
14.00     SNDRCVF  RCDFMT (RESTART)
15.00     ENDDO
16.00     /*
17.00     /* Digite instruções normais do programa */
18.00     /* inicial ou -TFRCTL- p/ primeiro programa */
19.00     /* menu */
20.00     ENDPGM
```

### Conceitos relacionados

“Exemplo: Código para um Programa de Processamento de Confirmação Padrão” na página 103  
O Pprograma de Processamento de Confirmação Padrão (STDCMT) executa as funções necessárias para estabelecer comunicação com um objeto de notificação exclusivo utilizado por todos os aplicativos.

“Objeto de Notificação de Confirmação” na página 52

Um *objeto de notificação* é uma fila de mensagens, área de dados ou arquivo de banco de dados contendo informações que identificam a última transação concluída com sucesso para uma determinada definição de confirmação, se ela não foi finalizada normalmente.

---

## Resolvendo Problemas em Transações e Controle de Confirmação

Essas informações podem ajudá-lo a resolver problemas de erro de controle de confirmação, detectar conflitos, recuperar transações após falha nas comunicações, saber quando forçar as operações de confirmação e rollback e quando cancelar a ressincronização e encerrar os rollbacks de longa execução.

### Erros do Controle de Confirmação

Ao utilizar o controle de confirmação, é importante compreender quais condições causam erros e quais não causam.

Em geral, os erros ocorrem quando as funções de controle de confirmação são utilizadas inconsistentemente, como executar um comando End Commitment Control (ENDCMTCTL) quando os arquivos que utilizam a definição de confirmação ainda estão abertos.

### Erros durante o Processamento de Confirmação

Se ocorrer um defeito do sistema ou uma falha na comunicação durante uma operação de confirmação, poderá ser necessário desempenhar a ressincronização para assegurar que os gerenciadores de transações mantenham os dados consistentes em todos os sistemas envolvidos na transação. O comportamento da ressincronização e como ela afeta a operação de confirmação dependem destes fatores :

- A opção de confirmação Aguardar resultado.



- O estado da transação.

Se a falha for catastrófica de modo a nunca poder ser reparada, ou não puder ser reparada a tempo, os operadores de outros sistemas envolvidos na transação deverão tomar uma decisão heurística. A decisão heurística confirma ou reverte as alterações feitas naquele sistema durante a transação. Se a falha for reparada após tal decisão e a ressincronização detectar que a decisão causou problemas de integridade de dados, a mensagem CPD83D9 ou CPD83E9 é enviada para a fila de mensagens QSYSOPR.

#### **Conceitos relacionados**

“Definição de Confirmação para Confirmação de Duas Fases: Não Aguardar pelo Resultado” na página 36

Quando ocorre um defeito do sistema ou uma falha na comunicação durante uma operação de confirmação de modo que a ressincronização é requerida, o padrão é aguardar o término da ressincronização antes da conclusão da operação de confirmação.

“Estados da Transação para o Controle de Confirmação de Duas Fases” na página 30

Uma definição de confirmação é estabelecida em cada localização que faz parte da rede do programa de transação. Para cada uma, o sistema mantém o estado de sua transação atual e da anterior.

### **Condições de Erro**

Se ocorrer um erro, uma mensagem de escape que pode ser monitorada em um programa será enviada.

Aqui estão alguns erros típicos relacionados ao controle de confirmação:

- Comandos STRCMTCTL consecutivos são executados sem um comando ENDCMTCTL interveniente.
- Arquivos são abertos sob controle de confirmação, mas nenhum comando STRCMTCTL foi executado.

Esta não é uma condição de erro para programas que são executados dentro do grupo de ativação que devem usar a definição de confirmação de nível de job. A definição de confirmação de nível de job pode ser iniciada apenas por um único programa, mas quando iniciada por um programa, ela é utilizada por qualquer programa sendo executado em qualquer grupo de ativação que não esteja utilizando uma definição de confirmação de nível de grupo de ativação. Os programas que são executados dentro de um grupo de ativação que devem usar a definição de confirmação de nível de grupo de ativação devem primeiro iniciar esta definição com o comando STRCMTCTL.

- Arquivos abertos para saída sob controle de confirmação não são registrados em diários.
- A primeira operação de abertura de um arquivo compartilhado coloca o arquivo sob controle de confirmação, porém operações de abertura subsequentes do mesmo arquivo compartilhado não colocam.
- A primeira operação de abertura de um arquivo compartilhado não coloca o arquivo sob controle de confirmação, porém operações de abertura subsequentes do mesmo arquivo compartilhado colocam.
- O limite de bloqueio do registro para o job é atingido em uma única transação.
- O programa emite uma operação de leitura, uma operação de confirmação e uma alteração para o mesmo registro. A operação de leitura deve ser emitida novamente após a operação de confirmação, pois esta liberou o bloqueio no registro.
- Para uma localização de uma fase, os recursos colocados sob controle de confirmação não residem na mesma localização que os recursos já sob controle de confirmação para a definição de confirmação.
- Alterações não confirmadas existem quando um comando ENDCMTCTL é emitido.

Esta não é uma condição de erro para o comando ENDCMTCTL se todos os arquivos estiverem fechados, algum banco de dados remoto estiver desconectado e nenhum recurso de confirmação da API ainda estiver associado à definição de confirmação a ser finalizada.

- Um comando de confirmação, rollback ou ENDCMTCTL é executado e um comando STRCMTCTL não foi executado.

Esta não é uma condição de erro para programas que são executados dentro de um grupo de ativação e a definição de confirmação de nível de job está ativa. A definição de confirmação de nível de job pode ser iniciada apenas por um único programa, mas quando iniciada por um programa, ela é utilizada por qualquer programa sendo executado em qualquer grupo de ativação que não esteja

utilizando uma definição de confirmação de nível de grupo de ativação. Os programas que são executados dentro de um grupo de ativação e vão usar a definição de confirmação de nível de grupo de ativação devem primeiro iniciar esta definição com o comando STRCMTCTL.

- Um comando ENDCMTCTL é executado com arquivos ainda abertos sob controle de confirmação para a definição de confirmação.
- Um job que está executando uma operação de gravação possui uma ou mais definições de confirmação que não estão em um limite de confirmação.
- Uma operação salvar-enquanto-ativo foi finalizada, pois outros jobs com recursos confirmáveis não atingiram um limite de confirmação no tempo especificado para o parâmetro SAVACTWAIT.
- Um processo salvar-enquanto-ativo não pôde continuar devido a recursos confirmáveis da API estarem sendo incluídos em mais de uma definição de confirmação para um único job.
- Mais de 1023 definições de confirmação existem para um único job.
- A conversão a uma localização remota foi perdida devido a uma falha do recurso. Isso pode causar o rollback da transação.
- Um recurso de uma fase que está aberto para atualização está presente em um nó que não iniciou a operação de confirmação. Você deve remover o recurso ou o nó que iniciou o pedido de confirmação.
- Uma operação de confirmação é solicitada enquanto a transação encontra-se em estado obrigatório de rollback (RBR). Uma operação de rollback deve ser feita.
- Um programa de saída da API emite um pedido de confirmação ou um pedido de rollback.
- Um programa de disparo emite um pedido de confirmação ou um pedido de rollback para a definição de confirmação sob a qual o programa de disparo foi chamado.

O programa de disparo pode iniciar uma definição de confirmação separada e emitir um pedido de confirmação ou rollback para aquela definição.

## Condições Sem Erro

Aqui estão algumas situações para controle de confirmação em que não ocorrem erros.

- Uma operação de confirmação ou de rollback é executada e nenhum recurso está sob o controle de confirmação. Isto permite incluir as operações de confirmação ou rollback no programa sem considerar se existem recursos sob o controle de confirmação. Permite também especificar uma identificação de confirmação antes de você fazer quaisquer alterações que possam ser confirmadas.
- Uma operação de confirmação ou de rollback é executada e não existem alterações do recurso não confirmadas. Isto permite incluir as operações de confirmação ou rollback no programa sem considerar se existem alterações de recurso não confirmadas.
- Um arquivo sob o controle de confirmação é fechado e existem registros não confirmados. Esta situação permite que outro programa seja chamado para executar a operação de confirmação ou rollback. Isso ocorre independentemente de o arquivo estar compartilhado. Essa função permite que um subprograma faça alterações no banco de dados que façam parte de uma transação que envolva vários programas.
- Um job é finalizado, de forma normal ou anormal, com alterações não confirmadas para uma ou mais definições de confirmação. As alterações de todas as definições de confirmação são revertidas.
- Um grupo de ativação finaliza com alterações pendentes para a definição de confirmação de nível do grupo de ativação. Se o grupo de ativação for finalizado normalmente e não existirem erros encontrados no fechamento de nenhum arquivo aberto sob o controle de confirmação direcionado para o mesmo grupo de ativação que está finalizando, o sistema executará uma confirmação implícita. Caso contrário, será executado um rollback implícito.
- Um programa acessa um registro alterado novamente que não foi confirmado. Isto permite que um programa:
  - Inclua um registro e atualize-o antes de especificar a operação de confirmação.
  - Atualize o mesmo registro duas vezes antes de especificar a operação de confirmação.
  - Inclua um registro e exclua-o antes de especificar a operação de confirmação.

- Acesse um registro não confirmado novamente por um arquivo lógico diferente (sob o controle de confirmação).
- Especifique LCKLVL(\*CHG or \*CS) no comando STRCMTCTL e abra um arquivo com uma operação de confirmação para somente leitura. Nesse caso, não ocorrem bloqueios no pedido. Ele é tratado como se o controle de confirmação não estivesse efetivado, mas o arquivo aparece na opção do menu WRKJOB dos arquivos sob o controle de confirmação.
- Emita o comando STRCMTCTL e não abra nenhum arquivo sob o controle de confirmação. Nesta situação, qualquer alteração de nível de registro feita nos arquivos não será efetuada sob o controle de confirmação.

## **Mensagens de Erro a Serem Monitoradas durante o Controle de Confirmação**

Várias mensagens de erro diferentes podem ser retornadas pelas operações de confirmação ou rollback, ou enviadas ao log do job, dependendo do tipo de mensagem e de quando ocorreu o erro.

As mensagens de erro podem ocorrer durante o seguinte processamento:

- Processamento normal de confirmação ou rollback
- Processamento de confirmação ou rollback durante a finalização do processo do job
- Processamento de confirmação ou rollback durante a finalização do grupo de ativação

Você não pode monitorar nenhuma das mensagens a seguir durante a finalização do grupo de ativação ou finalização do processo do job. Além disso, você só pode monitorar mensagens CPFxxxx. As mensagens CPDxxxx sempre são enviadas como mensagens de diagnóstico, as quais não podem ser monitoradas. Todos os erros encontrados ao finalizar uma definição de confirmação de nível de grupo de ativação durante a finalização do grupo de ativação ou ao finalizar qualquer definição de confirmação durante a finalização da tarefa são deixados no log de tarefa como mensagens de diagnóstico.

As mensagens de erro relacionadas ao controle de confirmação a serem procuradas são como as seguintes:

### **CPD8351**

As alterações talvez não tenham sido confirmadas.

### **CPD8352**

Alterações não confirmadas na localização remota &3.

### **CPD8353**

As alterações no banco de dados relacional &1 talvez não tenham sido confirmadas.

### **CPD8354**

As alterações no arquivo DDM &1 talvez não tenham sido confirmadas.

### **CPD8355**

As alterações no objeto DDL &1 talvez não tenham sido confirmadas.

### **CPD8356**

As alterações com rollback efetuado talvez não tenham sido confirmadas.

### **CPD8358**

Talvez não tenha sido efetuado rollback das alterações no banco de dados relacional &1.

### **CPD8359**

Talvez não tenha sido efetuado rollback das alterações no arquivo DDM &1.

### **CPD835A**

Talvez não tenha sido efetuado rollback das alterações no objeto DDL &3.

### **CPD835C**

O objeto de notificação &1 no &2 não está atualizado.

- CPD835D**  
O recurso DRDA não permite a suspensão do cursor SQL.
- CPF835F**  
Falha da operação de confirmação ou rollback.
- CPD8360**  
Membros ou arquivos ou ambos já estavam desalocados.
- CPD8361**  
O programa de saída da API &1 falhou durante a confirmação.
- CPD8362**  
O programa de saída da API &1 falhou durante o rollback.
- CPD8363**  
O programa de saída da API &1 finalizou após &4 minutos durante a confirmação.
- CPD8364**  
O programa de saída da API &1 finalizou após &4 minutos durante o rollback.
- CPD836F**  
Ocorreu um erro de protocolo durante a operação de controle de confirmação.
- CPD83D1**  
O recurso da API &4 não pode ser o último agente.
- CPD83D2**  
Recurso incompatível com o controle de confirmação.
- CPD83D7**  
Operação de confirmação mudou para rollback.
- CPD83D9**  
Ocorreu uma condição mista heurística.
- CPF83DB**  
Operação de confirmação resultou em rollback.
- CPD83DC**  
Ação Se Problemas Utilizados para determinar operação de confirmação ou rollback; razão &2.
- CPD83DD**  
Conversaõ finalizada; razão &1.
- CPD83DE**  
Informações de retorno inválidas.
- CPD83EC**  
O programa de saída da API &1 votou em rollback.
- CPD83EF**  
Operação de rollback iniciada para próxima unidade lógica de trabalho.
- CPF8350**  
Definição de confirmação não encontrada.
- CPF8355**  
ENDCMTCTL não permitido. Alterações pendentes ativas.
- CPF8356**  
Controle de confirmação finalizou com &1 alterações locais não confirmadas.
- CPF8358**  
O objeto de notificação &1 no &2 não está atualizado.

**CPF8359**

Falha da operação de rollback.

**CPF835A**

A finalização da definição de confirmação &1 foi cancelada.

**CPF835B**

Ocorreram erros durante a finalização do controle de confirmação.

**CPF835C**

O controle de confirmação finalizou com alterações remotas não confirmadas.

**CPF8363**

Falha da operação de confirmação.

**CPF8364**

O parâmetro do controle de confirmação não é válido. Código de razão &3.

**CPF8367**

Não pode efetuar operação de controle de confirmação.

**CPF8369**

Não pode colocar o recurso de confirmação da API sob controle de confirmação; código de razão &1.

**CPF83D0**

Operação de confirmação não permitida.

**CPF83D2**

Confirmação concluída == Ressincronização em andamento foi retornada.

**CPF83D3**

Confirmação concluída == Misto Heurístico foi retornado.

**CPF83D4**

A entrada do diário da unidade lógica de trabalho não foi enviada.

**CPF83E1**

A operação de confirmação falhou devido a violação de limitação.

**CPF83E2**

Operação de rollback exigida.

**CPF83E3**

O nível de aninhamento solicitado não está ativo.

**CPF83E4**

O controle de confirmação finalizou com recursos não confirmados.

**CPF83E6**

A operação de controle de confirmação foi concluída com a ressincronização em andamento.

**CPF83E7**

A confirmação ou rollback da transação global X/Open não é permitida.

## **Monitorando Erros Após um Comando CALL**

Quando um programa que utiliza controle de confirmação for chamado, monitore erros inesperados e execute uma operação de rollback caso ocorra um erro.

Por exemplo, registros não confirmados podem existir quando um programa encontra um erro inesperado, como um erro de divisão por zero RPG.

Dependendo do status do parâmetro de resposta da mensagem de indagação (INQMSGRPY) para um job, o programa envia uma mensagem de indagação ou executa uma ação padrão. Se a resposta do operador ou a ação padrão finalizar o programa, os registros não confirmados ainda existem aguardando por uma operação de confirmação ou rollback.

Se outro programa for chamado e ocasionar uma operação de confirmação, a transação parcialmente concluída do programa anterior é confirmada.

Para impedir que transações parcialmente concluídas sejam confirmadas, monitore mensagens de escape após o comando CALL. Por exemplo, se for um programa RPG, utilize a seguinte codificação:

```
CALL RPGA
MONMSG MSGID(RPG9001)
EXEC(ROLLBACK) /*Rollback se pgm for cancelado*/
```

Se for um programa COBOL:

```
CALL COBOLA
MONMSG MSGID(CBE9001)
EXEC(ROLLBACK) /*Rollback se pgm for cancelado*/
```

### Falha do Processamento Normal de Confirmação ou Rollback

Podem ocorrer erros a qualquer momento durante o processamento de confirmação ou rollback.

A tabela a seguir divide esse processamento em quatro situações. A coluna do meio descreve as ações tomadas pelo sistema quando encontra erros durante cada situação. A terceira coluna sugere o que você ou o aplicativo deverão fazer em resposta às mensagens. Essas sugestões são compatíveis com o modo como o processamento do controle de confirmação é manipulado pelo sistema.

Situação	Processamento de Confirmação ou Rollback	Ação Sugerida
Falha da <b>confirmação</b> de E/S do nível de registro	<ul style="list-style-type: none"> <li>Se o erro ocorrer durante a onda de preparação, a transação será revertida e a mensagem CPF83DB enviada.</li> <li>Se o erro ocorrer durante a onda confirmada, o processamento de confirmação continuará a confirmar tantos recursos restantes quanto possíveis. A mensagem CPF8363 é enviada no final do processamento de confirmação.</li> </ul>	Monitor para mensagens; manipule-o conforme desejar.

Situação	Processamento de Confirmação ou Rollback	Ação Sugerida
Falha do nível do objeto ou do programa de saída de confirmação ou de rollback para o recurso de confirmação da API durante a confirmação	<ul style="list-style-type: none"> <li>• Se o erro ocorrer durante a onda de preparação, a transação será revertida e a mensagem CPF83DB enviada.</li> <li>• Se o erro ocorrer durante a onda confirmada, o processamento continuará a confirmar ou reverter tantos recursos restantes quanto possíveis. Uma das mensagens a seguir retorna, dependendo do tipo de recurso de confirmação: <ul style="list-style-type: none"> <li>– CPD8353</li> <li>– CPD8354</li> <li>– CPD8355</li> <li>– CPD8361</li> </ul> </li> </ul> <p>A mensagem CPF8363 é enviada no final do processamento de confirmação.</p>	Monitor para mensagens; manipule-o conforme desejar.
Falha de <b>rollback</b> de E/S do nível de registro	<ol style="list-style-type: none"> <li>1. Retorna CPD8356</li> <li>2. Tenta continuar o processamento para reverter o nível do objeto ou os recursos de confirmação da API</li> <li>3. Retorna CPF8359 no final do processamento</li> </ol>	Monitor para mensagens; manipule-o conforme desejar.
Falha do nível do objeto ou do programa de saída de confirmação ou de rollback para os recursos de confirmação da API durante o rollback	<ol style="list-style-type: none"> <li>1. Retorna uma das mensagens a seguir, dependendo do tipo de recurso de confirmação: <ul style="list-style-type: none"> <li>• CPD8358</li> <li>• CPD8359</li> <li>• CPD835A</li> <li>• CPD8362</li> </ul> </li> <li>2. Continua o processamento</li> <li>3. Retorna CPF8359 no final do processamento</li> </ol>	Monitor para mensagens; manipule-o conforme desejar.

### Processamento de Confirmação ou Rollback durante Finalização da Tarefa

Todas as situações descritas na tabela anterior também se aplicam quando um job está finalizando, exceto que uma das mensagens a seguir é enviada:

- CPF8356 se somente os recursos locais foram registrados
- CPF835C se somente os recursos remotos foram registrados
- CPF83E4 se os recursos locais e remotos foram registrados

Além disso, uma das duas mensagens poderá parecer específica da conclusão da tarefa se um programa de saída de confirmação ou rollback para um recurso confirmável da API tiver sido chamado. Se o programa de saída de confirmação e rollback não for concluído dentro de 5 minutos, ele será cancelado; uma mensagem de diagnóstico CPD8363 (para confirmação) ou CPD8364 (para rollback) será enviada e o restante do processamento de confirmação ou rollback continuará.

## Processamento de Confirmação ou Rollback durante IPL (Initial Program Load)

Todas as situações descritas na tabela anterior também se aplicam durante a recuperação de IPL para definições de confirmação, exceto se a mensagem CPF835F foi enviada em lugar de CPF8359 ou CPF8363. As mensagens enviadas para uma determinada definição de confirmação podem aparecer no log de tarefa para uma das tarefas QDBSRVxx ou no log QHST. No log QHST, a mensagem CPI8356 indica o início da recuperação de IPL para uma definição de confirmação em particular. A mensagem CPC8351 indica o final da recuperação de IPL para uma determinada definição de confirmação e qualquer outra mensagem relativa à recuperação dessa definição é encontrada entre essas duas mensagens.

Uma das duas mensagens poderá parecer específica de uma definição de confirmação se um programa de saída de confirmação ou rollback para um recurso confirmável da API tiver sido chamado. Se o programa de saída de confirmação e rollback não for concluído dentro de 5 minutos, ele será cancelado; uma mensagem de diagnóstico CPD8363 (para confirmação) ou CPD8364 (para rollback) será enviada e o restante do processamento de confirmação ou rollback continuará.

## Detectando Conflitos

Uma condição de conflito pode ocorrer quando um job mantém um bloqueio sobre um objeto, objeto A, e fica aguardando para obter um bloqueio sobre outro objeto, objeto B. Ao mesmo tempo, outro job ou transação mantém no momento um bloqueio sobre o objeto B e fica aguardando para obter um bloqueio sobre o objeto A.

Realize as seguintes etapas para descobrir se ocorreu uma condição de impasse e, se tiver ocorrido, corrija-a:

1. Localize a tarefa suspensa na lista de tarefas ativas.
2. Veja os objetos que o job está aguardando para bloquear.
3. Para todos os objetos que o job está aguardando para bloquear, veja a lista dos portadores de bloqueio (transações ou jobs) e tente localizar um bloqueio em conflito correspondente ao nível solicitado pelo job suspenso.
4. Se uma transação contiver um bloqueio em conflito, exiba as tarefas associadas a essa transação e verifique se alguma delas está aguardando bloqueio.
5. Determine se este job que está aguardando está tentando bloquear um dos objetos bloqueados pelo job suspenso inicial. Ao localizar o job que está tentando bloquear um dos objetos bloqueados pelo job suspenso inicial, você conseguirá identificar os objetos em questão como os pontos de problema.
6. Investigue a transação para determinar o curso apropriado da ação.
  - a. Veja as propriedades da transação para descobrir o aplicativo que a iniciou e, em seguida, veja o código de aplicativo.
  - b. Ou rastreie as ações da transação até este ponto encontrando o ID do Ciclo de confirmação nas propriedades da transação e depois procurando em um diário as entradas que contém este ID. Para fazer isso, você pode utilizar o comando Retrieve Journal Entry (RTVJRNE) e especificar o parâmetro CMTCYCID.
  - c. Depois de obter informações relevantes, você pode optar por forçar uma operação de rollback ou confirmação.

### Tarefas relacionadas

“Minimizando os Bloqueios” na página 73

Uma forma típica de minimizar os bloqueios do registro é liberá-los. (Esta técnica não funciona se LCKLVL(\*ALL) foi especificado.)

Determinando o Status de um Job

“Exibindo Objetos Bloqueados para uma Transação” na página 69

Você pode exibir objetos bloqueados para transações globais somente com bloqueios com escopo de transação.



“Exibindo Jobs Associados a uma Transação” na página 70

Para exibir os jobs associados a uma transação, siga estas etapas.

“Exibindo Informações de Controle de Confirmação” na página 68

O System i Navigator pode exibir informações sobre todas as transações (unidades lógicas de trabalho) no sistema. O System i Navigator também pode exibir as informações sobre o job, se houver, associado a uma transação.

“Quando Forçar as Operações de Confirmação e de Rollback e Quando Cancelar a Ressincronização” na página 116

A decisão para forçar uma operação de confirmação ou de rollback é chamada de *decisão de heurística*. Esta ação permite que um operador confirme ou faça rollback manualmente dos recursos para uma transação que esteja em um estado preparado.

#### Referências relacionadas

Comando Retrieve Journal Entry (RTVJRNE)

## Recuperando Transações Após Falha em Comunicações

Essas instruções ajudam a manipular as transações que executam trabalho em um sistema remoto após a falha na comunicação com esse sistema.

Em caso de uma falha de comunicação, o sistema geralmente conclui a ressincronização com qualquer sistema remoto automaticamente. No entanto, se a falha for catastrófica de modo que a comunicação jamais será restabelecida com o sistema remoto (se, por exemplo, a linha de comunicação for cortada), você deverá cancelar a ressincronização e restaurar as transações você mesmo. As transações também podem estar suspendendo bloqueios que precisam ser liberados.

1. No System i Navigator, exiba as informações de controle de confirmação para a transação com a qual você está trabalhando.
2. Procure a transação de interesse que está tentando ressincronizar com o sistema remoto. O campo **Ressincronização em Progresso** dessa transação é definido como **sim**.
3. Procure as transações que tiveram conexão com o sistema remoto selecionando o recurso Status de transações individuais.
4. Depois de identificar as transações, você deve forçar uma confirmação ou um rollback, dependendo do estado da transação.
5. Você pode tomar a decisão de confirmar ou efetuar rollback depois de investigar as propriedades da transação.
  - Você pode utilizar o **ID da Unidade de Trabalho** para encontrar outras partes da transação em outros sistemas.
  - Você pode também determinar a confirmação ou rollback do estado da transação. Por exemplo, se uma transação do banco de dados estava desempenhando uma confirmação de duas fases durante a falha na comunicação e seu estado após a falha é "preparado" ou "último agente pendente", você poderá optar por forçar a confirmação na transação.
6. Após impor uma confirmação ou rollback nas transações incertas, pare a ressincronização na conexão com falha para as transações identificadas.

#### Tarefas relacionadas

“Exibindo Informações de Controle de Confirmação” na página 68

O System i Navigator pode exibir informações sobre todas as transações (unidades lógicas de trabalho) no sistema. O System i Navigator também pode exibir as informações sobre o job, se houver, associado a uma transação.

“Quando Forçar as Operações de Confirmação e de Rollback e Quando Cancelar a Ressincronização” na página 116

A decisão para forçar uma operação de confirmação ou de rollback é chamada de *decisão de heurística*. Esta ação permite que um operador confirme ou faça rollback manualmente dos recursos para uma transação que esteja em um estado preparado.

## Quando Forçar as Operações de Confirmação e de Rollback e Quando Cancelar a Ressincronização

A decisão para forçar uma operação de confirmação ou de rollback é chamada de *decisão de heurística*. Esta ação permite que um operador confirme ou faça rollback manualmente dos recursos para uma transação que esteja em um estado preparado.

Ao tomar uma decisão heurística, o estado da transação torna-se misto heurístico se sua decisão estiver inconsistente com os resultados de outras localizações na transação. Você deve ficar responsável por determinar a atitude tomada por todas as outras localizações que participaram da transação e por ressincronizar os registros do banco de dados.

Antes de tomar uma decisão heurística, reúna o máximo de informações que puder sobre a transação. Exiba os jobs associados à definição de confirmação e faça um registro de quais diários e arquivos estão envolvidos. Você pode usar estas informações posteriormente, se precisar exibir entradas do diário e aplicar ou remover, manualmente, alterações registradas no diário.

O melhor lugar para encontrar informações sobre uma transação é olhar na localização que foi o iniciador para aquela transação. No entanto, a decisão de confirmar ou efetuar rollback pode pertencer a um recurso da API ou a um último agente.

Se um recurso da API foi registrado como um recurso do último agente, a decisão final de confirmar ou reverter pertence ao recurso da API. É preciso ver informações sobre o aplicativo e como ele utiliza o recurso da API para determinar se irá confirmar ou reverter.

Se a transação possui um último agente selecionado, este possui a decisão de confirmar ou reverter. Consulte o status do último agente para obter informações sobre a transação.

Quando você precisar tomar decisões heurísticas ou cancelar a ressincronização devido a um defeito do sistema ou uma falha na comunicação que não pode ser reparada, poderá localizar todas as transações incertas utilizando as seguintes etapas:

1. No System i Navigator, expanda o sistema com o qual você deseja trabalhar.
2. Expanda **Bancos de Dados** e o banco de dados local para o sistema.
3. Expanda **Transações**.
4. Expanda **Transações do Banco de Dados** ou **Transações Globais**.

Nesta janela de exibição, você pode ver a definição de confirmação, o status de ressincronização, o atual ID da unidade de trabalho e o atual estado da unidade de trabalho para cada transação. Procure por transações com as seguintes informações:

- Transações com um **Estado da Unidade Lógica de Trabalho** de Preparado ou Último Agente Pendente.
- Transações que mostram o status de **Ressincronização em Andamento** como sim.

Para trabalhar com o job que está participando da transação neste sistema, clique com o botão direito do mouse na transação e selecione **job**.

Ao clicar com o botão direito do mouse na transação, você também pode selecionar **Forçar Confirmação**, **Forçar Rollback** ou **Cancelar Ressincronização**.

Antes de tomar uma decisão heurística ou cancelar a ressincronização, seria bom verificar o status dos jobs em outros sistemas associados à transação. A verificação dos jobs em sistemas remotos pode ajudá-lo a evitar decisões que causam inconsistências de banco de dados entre os sistemas.

1. Clique com o botão direito do mouse na transação com a qual deseja trabalhar.
2. Selecione **Status do Recurso**.

3. No diálogo Status do Recurso, selecione a guia **Conversaço** para conexões SNA; selecione **Conexão** para conexões TCP/IP.

Cada recurso da conversaço representa um sistema remoto que está participando da transaço. Nos sistemas remotos, vocÊ pode utilizar o System i Navigator para ver as transaço

A parte base do ID da unidade de trabalho é igual em todos os sistemas. Ao exibir informaço de controle de confirmaço no sistema remoto, procure pela parte base do ID da unidade de trabalho que é a mesma no sistema local.

Por exemplo, se o ID da unidade de trabalho no sistema local começa com:  
APPN.RCHASL97.X'112233445566, procure o ID da unidade de trabalho no sistema remoto que também começa com APPN.RCHASL97.X'112233445566.

#### **Conceitos relacionados**

“Suporte de Transaço XA para o Controle de Confirmaço” na página 45  
O DB2 para i5/OS pode participar nas transaço globais do X/Open.

“Iniciando o Controle de Confirmaço” na página 51

Para iniciar o controle de confirmaço, utilize o comando Start Commitment Control (STRCMTCTL).

#### **Tarefas relacionadas**

“Detectando Conflitos” na página 114

Uma condiço de conflito pode ocorrer quando um job mantém um bloqueio sobre um objeto, objeto A, e fica aguardando para obter um bloqueio sobre outro objeto, objeto B. Ao mesmo tempo, outro job ou transaço mantém no momento um bloqueio sobre o objeto B e fica aguardando para obter um bloqueio sobre o objeto A.

“Recuperando Transaço Após Falha em Comunicaço” na página 115

Essas instruço ajudam a manipular as transaço que executam trabalho em um sistema remoto após a falha na comunicaço com esse sistema.

“Exibindo Informaço de Controle de Confirmaço” na página 68

O System i Navigator pode exibir informaço sobre todas as transaço (unidades lógicas de trabalho) no sistema. O System i Navigator também pode exibir as informaço sobre o job, se houver, associado a uma transaço.

## **Encerrando um Rollback de Longa Execuço**

VocÊ talvez queira encerrar rollbacks de longa execuço que consomem tempo crítico do processador, recursos de bloqueio ou que ocupam espaço de armazenamento.

Uma operaço de rollback remove todas as alteraço feitas em uma transaço desde a operaço de confirmaço ou rollback anterior. Durante uma operaço de rollback, o sistema libera também os bloqueios relacionados à transaço. Se o sistema contiver milhares de transaço, a conclusão de uma operaço de rollback poderá levar horas. Esses rollbacks de execuço longa podem consumir um tempo crítico do processador, bloquear recursos ou ocupar espaço de armazenamento.

Antes de finalizar um rollback de execuço longa, é necessário saber quais definiço de confirmaço estão sendo revertidas e em que estado estão as definiço de confirmaço. O campo Estado das definiço de confirmaço que estão sendo revertidas é definido como ROLLBACK IN PROGRESS.

Utilize o comando Work with Commitment Definitions (WRKCMTDFN) para verificar o status de um rollback, seguindo estas etapas:

1. Digite WRKCMTDFN JOB(\*ALL) a partir da interface baseada em caracteres.
2. Digite F11 para exibir o campo Estado.

Se vocÊ finalizar um rollback de execuço longa, os arquivos que foram alterados durante a transaço serão deixados com transaço parciais. VocÊ deverá finalizar um rollback se seus arquivos não puderem

ter transações parciais. Para ver quais arquivos foram alterados durante a transação, escolha a opção 5 para exibir o status da lista WRKCMTDFN. Pressione F6 para exibir o status do recurso e selecione Nível de Registro.

Para finalizar um rollback de execução longa, você deve ter a autoridade especial Todos os Objetos (\*ALLOBJ). Siga estas etapas para finalizar um rollback de execução longa:

1. Digite WRKCMTDFN JOB(\*ALL) a partir da interface baseada em caracteres.
2. Digite a opção 20 (Finalizar rollback) na definição de confirmação que deseja finalizar.

Os arquivos com transações parciais têm o campo Existem Transações Parciais, Rollback Finalizado definido como \*YES na saída do comando Display File Description (DSPFD). É necessário remover as transações parciais antes poder utilizar o arquivo. As transações parciais podem ser removidas com a exclusão do arquivo e restauração do arquivo a partir de um salvamento anterior. Se você não tiver um salvamento anterior, o comando Change Journalled Object (CHGJRNOBJ) poderá ser utilizado para reconfigurar o estado Existe Transação Parcial para que seja possível abrir o arquivo. O uso do comando CHGJRNOBJ exige que o arquivo seja editado para um estado consistente. O comando CHGJRNOBJ só deverá ser usado se nenhum salvamento anterior estiver disponível.

## Desativando a Capacidade de Encerrar um Rollback de Longa Execução

Os usuários com a autoridade especial \*ALLOBJ podem finalizar rollbacks por padrão. Se você deseja restringir os usuários que possuem a autoridade especial \*ALLOBJ de finalizar rollbacks, faça isso criando áreas de dados QGPL/QTNNOENDRB.

### Referências relacionadas

Comando Create Data Area (CRTDTAARA)


---

## Informações Relacionadas para o Controle de Confirmação

Manuais do Produto, Publicações do IBM Redbooks, Web Sites e outras coletas de tópico do centro de informações contêm informações relacionadas à coleta de tópico de controle de Confirmação. É possível visualizar ou imprimir quaisquer dos arquivos PDF.




### Manuais

Os seguintes manuais não são incluídos no V6R1 i5/OS Information Center. No entanto, esses manuais podem ser uma referência útil para você. Cada um dos manuais está disponível a partir do IBM

Publications Center ([www.elink.ibm.com/publications/servlet/pbi.wss?](http://www.elink.ibm.com/publications/servlet/pbi.wss?))  como uma cópia impressa que você pode solicitar, em um formato on-line que você pode transferir o download sem encargos, ou ambos.

- Guia do Usuário COBOL/400
- Guia do Usuário RPG/400

### IBM Redbooks

- Advanced Functions and Administration on DB2 para i5/OS  (5529 KB)
- Stored Procedures, Triggers, and User Defined Functions on DB2 para i5/OS  (5836 KB)
- Striving for Optimal Journal Performance on DB2 para i5/OS  (3174 KB)

### Web Sites

- The Open Group ([www.opengroup.org](http://www.opengroup.org)) 

## Outras Informações

- Programação de Banco de Dados
- Programação em SQL
- APIs de XA
- Gerenciamento de diário

### Referências relacionadas

“Arquivo PDF para Controle de Confirmação” na página 1

Você pode visualizar e imprimir um arquivo PDF dessas informações.

---

## Informações sobre o Código de Licença e Renúncia

A IBM concede-lhe uma licença de direitos autorais não exclusivos para usar os exemplos de código de programação, a partir dos quais você pode gerar funções idênticas adaptadas a uma necessidade específica.

SUJEITA ÀS GARANTIAS ESTABELECIDAS POR LEI, QUE NÃO PODEM SER EXCLUÍDAS, A IBM, SEUS DESENVOLVEDORES E FORNECEDORES DO PROGRAMA NÃO OFERECEM GARANTIA OU CONDIÇÕES, SEJAM EXPRESSAS OU IMPLÍCITAS, INCLUINDO, MAS NÃO SE LIMITANDO ÀS GARANTIAS IMPLÍCITAS OU ÀS CONDIÇÕES DE MERCADO, ADEQUAÇÃO A UM DETERMINADO PROPÓSITO E NÃO-INFRAÇÃO EM RELAÇÃO AO PROGRAMA OU SUPORTE TÉCNICO, SE HOUVER.

SOB NENHUMA CIRCUNSTÂNCIA, A IBM, OS DESENVOLVEDORES OU FORNECEDORES DO PROGRAMA SÃO RESPONSÁVEIS PELOS ITENS A SEGUIR, MESMO SE INFORMADOS DE SUA POSSIBILIDADE:

1. PERDA OU DANO DE DADOS;
2. DANOS DIRETOS, ESPECIAIS, ACIDENTAIS OU INDIRETOS, OU QUALQUER ESPÉCIE DE DANO DE CONSEQÜÊNCIA ECONÔMICA; OU
3. PERDA DE LUCROS, NEGÓCIOS, RECEITAS, BENS OU ECONOMIAS.

ALGUMAS JURISDIÇÕES NÃO PERMITEM A EXCLUSÃO OU LIMITAÇÃO DE DANOS DIRETOS, ACIDENTAIS OU CONSEQÜENCIAIS, PORTANTO, ALGUMAS, OU TODAS, LIMITAÇÕES OU EXCLUSÕES ACIMA PODEM NÃO SE APLICAR À REGIÃO DO CLIENTE.



---

## Apêndice. Avisos

Estas informações foram desenvolvidas para produtos e serviços oferecidos nos Estados Unidos.

A IBM pode não oferecer os produtos, serviços ou recursos oferecidos neste documento em outros países. Consulte um representante IBM local para obter informações sobre os produtos e serviços atualmente disponíveis em sua área. Qualquer referência a produtos, programas ou serviços IBM não significa que apenas os produtos, programas ou serviços IBM possam ser utilizados. Qualquer produto, programa ou serviço funcionalmente equivalente, que não infrinja nenhum direito de propriedade intelectual da IBM, poderá ser utilizado em substituição a este produto, programa ou serviço. Entretanto, a avaliação e verificação da operação de qualquer produto, programa ou serviço não-IBM são de responsabilidade do Cliente.

A IBM pode ter patentes ou solicitações de patentes pendentes relativas a assuntos tratados nesta publicação. O fornecimento desta publicação não garante ao Cliente nenhum direito sobre tais patentes. Pedidos de licenças devem ser enviados, por escrito, para:

Gerência de Relações Comerciais e Industriais da IBM Brasil  
Av. Pasteur, 138-146  
Botafogo  
Rio de Janeiro, RJ  
CEP 22290-240

Para pedidos de licença relacionados a informações de DBCS (Conjunto de Caracteres de Byte Duplo), entre em contato com o Departamento de Propriedade Intelectual da IBM em seu país ou envie pedidos de licença, por escrito, para:

IBM World Trade Asia Corporation  
Licensing  
2-31 Roppongi 3-chome, Minato-ku  
Tokyo 106-0032, Japan

**O seguinte parágrafo não se aplica ao Reino Unido e a nenhum país em que tais disposições não estejam de acordo com a legislação local:** A INTERNATIONAL BUSINESS MACHINES CORPORATION FORNECE ESTA PUBLICAÇÃO "NO ESTADO EM QUE SE ENCONTRA" SEM GARANTIA DE NENHUM TIPO, SEJA EXPRESSA OU IMPLÍCITA, INCLUINDO, MAS NÃO SE LIMITANDO ÀS GARANTIAS IMPLÍCITAS DE MERCADO OU DE ADEQUAÇÃO A UM DETERMINADO PROPÓSITO. Alguns países não permitem a exclusão de garantias expressas ou implícitas em certas transações; portanto, essa disposição pode não se aplicar ao Cliente.

Esta publicação pode conter imprecisões técnicas ou erros tipográficos. Periodicamente, são feitas alterações nas informações aqui contidas; tais alterações serão incorporadas em novas edições da publicação. A IBM pode, a qualquer momento, aperfeiçoar e/ou alterar os produtos e/ou programas descritos nesta publicação, sem aviso prévio.

Qualquer referência nestas informações a Web sites não-IBM são fornecidas apenas por conveniência e não representam de forma alguma um endosso a esses Web sites. Os materiais contidos nesses Web sites não fazem parte dos materiais deste produto IBM e a utilização desses Web sites é de inteira responsabilidade do Cliente.

A IBM pode utilizar ou distribuir todas os comentários fornecidos pelo Cliente da maneira que achar conveniente, sem que isso implique em qualquer compromisso ou obrigação para com o Cliente.

Os licenciados deste programa que pretendam obter informações adicionais sobre o mesmo com o objetivo de permitir: (i) a troca de informações entre programas criados independentemente e outros programas (incluindo este) e (ii) a utilização mútua das informações trocadas, devem entrar em contato com a:

Gerência de Relações Comerciais e Industriais da IBM Brasil  
Av. Pasteur, 138-146  
Botafogo  
Rio de Janeiro, RJ  
CEP 22290-240

Tais informações podem estar disponíveis, sujeitas a termos e condições apropriadas, incluindo em alguns casos, o pagamento de uma taxa.

O programa licenciado descrito neste documento e todo o material licenciado disponível para ele são fornecidos pela IBM sob os termos do Contrato com o Cliente IBM, do Contrato Internacional de Licença do Programa IBM, do Contrato de Licença IBM para Código de Máquina ou de qualquer outro contrato equivalente.

Todos os dados sobre desempenho aqui descritos foram determinados em um ambiente controlado. Portanto, os resultados obtidos em outros ambientes operacionais podem variar de maneira significativa. Algumas medidas podem ter sido tomadas em sistemas em fase de desenvolvimento e não há garantia de que tais medidas sejam as mesmas nos sistemas normalmente disponíveis. Além disso, algumas medidas podem ter sido estimadas através de extrapolação. Resultados reais podem variar. Usuários deste documento devem verificar os dados aplicáveis para seu ambiente específico.

As informações referentes a produtos não-IBM foram obtidas juntos aos fornecedores desses produtos, anúncios publicados ou outras fontes de publicidade disponíveis. A IBM não testou tais produtos e não pode confirmar a excelência do desempenho, a compatibilidade ou quaisquer outras reivindicações relacionadas a produtos não-IBM. Dúvidas sobre os recursos dos produtos não-IBM devem ser encaminhadas aos fornecedores desses produtos.

Todas as declarações relacionadas a futuros direcionamentos ou pretensões da IBM estão sujeitas a alterações ou cancelamento sem prévio aviso, e apenas representam objetivos.

Estas informações contêm exemplos de dados e relatórios utilizados em operações de negócios do dia-a-dia. Para ilustrá-los da forma mais completa possível, os exemplos incluem nomes de pessoas, empresas, marcas e produtos. Todos esses nomes são fictícios e qualquer semelhança com nomes e endereços utilizados por uma empresa real é mera coincidência.

#### LICENÇA DE COPYRIGHT:

Estas informações contêm exemplos de programas aplicativos na linguagem fonte, ilustrando as técnicas de programação em diversas plataformas operacionais. O Cliente pode copiar, modificar e distribuir estes programas de exemplo sem a necessidade de pagar à IBM, com objetivos de desenvolvimento, utilização, marketing ou distribuição de programas aplicativos em conformidade com a interface de programação de aplicativo para a plataforma operacional para a qual os programas de exemplo são criados. Esses exemplos não foram completamente testados sob todas as condições. Portanto, a IBM não pode garantir ou implicar a confiabilidade, manutenção ou função desses programas.

Cada cópia ou parte desses programas de amostra ou qualquer trabalho derivado, deve incluir um aviso de copyright com os dizeres:

© (nome da empresa) (ano). Partes deste código são derivadas dos Programas de Amostra da IBM Corp.  
© Copyright IBM Corp. \_digite o ano ou anos\_. Todos os direitos reservados.

Se estiver visualizando estas informações em cópia eletrônica, as fotos e ilustrações podem não aparecer.



---

## Informações da Interface de Programação

Esta publicação de Controle de Confirmação documenta as Interfaces de Programação planejadas que permitem ao cliente gravar programas para obter os serviços do IBM i5/OS.

---

## Marcas Registradas

Os termos a seguir são marcas registradas da International Business Machines Corporation nos Estados Unidos e/ou em outros países:

COBOL/400  
DB2  
Distributed Relational Database Architecture  
DRDA  
i5/OS  
IBM  
IBM (logotipo)  
Integrated Language Environment  
Redbooks  
RPG/400  
System i

Adobe, o logotipo Adobe, PostScript e o logotipo PostScript são marcas ou marcas registradas da Adobe Systems Incorporated nos Estados Unidos e/ou em outros países.

UNIX é uma marca registrada do The Open Group nos Estados Unidos e em outros países.

Outros nomes de empresas, produtos ou serviços podem ser marcas registradas ou marcas de serviço de terceiros.

---

## Termos e Condições

As permissões para o uso dessas publicações estão sujeitas aos seguintes termos e condições.

**Uso Pessoal:** essas publicações podem ser reproduzidas para uso pessoal, não comercial, desde que todos os avisos de propriedade sejam preservados. Não é possível distribuir, exibir ou fazer trabalhos derivados dessas publicações ou de nenhuma parte desse documento, sem consentimento expresso da IBM.

**Uso Comercial:** é permitido reproduzir, distribuir e expor essas publicações exclusivamente dentro de sua empresa, desde que todos os avisos de propriedade sejam preservados. Não é possível fazer trabalhos derivados dessas publicações, ou reproduzir, distribuir ou exibir essas publicações ou qualquer parte deste documento fora da sua empresa, sem o consentimento expresso da IBM.

Exceto conforme concedido expressamente nessa permissão, nenhuma outra permissão, licença ou direito é concedido, seja expressa ou implícita, às publicações ou a qualquer informação, dados, software ou outra propriedade intelectual contida neste documento.

A IBM reserva-se o direito de revogar as permissões aqui concedidas, sempre que, a seu critério, o uso das publicações prejudicar seus interesses ou, conforme determinação da IBM, as instruções anteriormente citadas não estiverem sendo seguidas da forma apropriada.

Não é permitido fazer download, exportar ou reexportar estas informações, exceto em total conformidade com todas as leis e regulamentos aplicáveis, incluindo todas as leis e regulamentos de exportação dos Estados Unidos.

A IBM NÃO FORNECE NENHUMA GARANTIA SOBRE O CONTEÚDO DESSAS PUBLICAÇÕES. AS PUBLICAÇÕES SÃO FORNECIDAS "NO ESTADO EM QUE SE ENCONTRAM" E SEM GARANTIA DE NENHUM TIPO, SEJA EXPRESSA OU IMPLÍCITA, INCLUINDO MAS NÃO SE LIMITANDO ÀS GARANTIAS IMPLÍCITAS DE MERCADO, NÃO-INFRAÇÃO E DE ADEQUAÇÃO A UM DETERMINADO PROPÓSITO.





Impresso em Brazil