

CICS Transaction Server for z/OS



Guia de Tabelas de Dados Compartilhados

Versão 5 Release 4

CICS Transaction Server for z/OS



Guia de Tabelas de Dados Compartilhados

Versão 5 Release 4

Nota

Antes de utilizar estas informações e o produto que elas suportam, leia as informações em “Avisos” na página 77.

Índice

Sobre este PDF	v
-----------------------	----------

Capítulo 1. Introdução às Tabelas de Dados Compartilhados

O conceito de tabelas de dados compartilhados	1
Descrição de tabelas de dados	2
tabela de dados mantida pelo CICS	2
Tabela de dados mantida pelo usuário	3
O ambiente de compartilhamento de tabela de dados	3
O conjunto de dados de origem para tabelas de dados	4
Espaços de dados para tabelas de dados	4
Saídas de usuário globais para tabelas de dados	5
Serviços de tabela de dados compartilhados e acesso de arquivo remoto	6
Como uma tabela de dados é compartilhada	7
LOGON	7
CONEXÃO	7

Capítulo 2. Configurando tabelas de dados compartilhados

Planejando para Usar as Tabelas de Dados	9
Desempenho de uma tabela de dados mantida pelo CICS	9
Desempenho de uma tabela de dados mantida pelo usuário	9
Uso de armazenamento para tabelas de dados compartilhados	9
Requisitos de JCL do MVS ao usar tabelas de dados compartilhados	12
Selecionando arquivos para usar como tabelas de dados	12
Usando estatísticas para selecionar tabelas de dados	14
Verificação de segurança para tabelas de dados	18
Preparando-se para usar suporte de tabela de dados compartilhados	19
Definição de recurso para tabelas de dados	21
Definição de recurso para tabelas de dados mantidas pelo CICS	21
Definição de recurso para tabelas de dados mantidas pelo usuário	22
O comando DEFINE FILE define tabelas de dados	23
Comandos EXEC CICS para tabelas de dados	27
Comandos CEMT para tabelas de dados	28

Capítulo 3. Desenvolvimento para acesso a tabelas de dados

Programação de aplicativos para uma tabela de dados mantida pelo CICS	31
Usando uma tabela de dados mantida pelo CICS durante o carregamento	32
Programação de aplicativos para uma tabela de dados mantida pelo usuário	33

Usando uma tabela de dados mantida pelo usuário durante o carregamento	34
Uso de serviços de memória cruzada para tabelas de dados compartilhados	34
Conexão	35
Diferenças entre remessa de função e serviços de memória cruzada	36
Diferenças entre serviços de tabela de dados compartilhados e VSAM	37

Capítulo 4. Customizando Tabelas de Dados Usando as Saídas de Usuário

Comunicação entre o CICS e programas de saída de tabela de dados compartilhados	39
Saída de usuário XDTRD	43
Saída de usuário XDTAD	44
Saída de usuário XD TLC	45
Ativando saídas de usuário para tabelas de dados	45

Capítulo 5. Administrando tabelas de dados

Abrindo uma tabela de dados	47
Fechando uma tabela de dados	48
Usando suporte de tabelas de dados compartilhados em um sysplex	49
Visão geral do suporte a tabelas de dados compartilhados em um sysplex	49
Como atualizar tabelas de dados mantidas pelo usuário replicadas	50
Programa de exemplo para atualizar uma tabela de dados mantida pelo usuário	52

Capítulo 6. Resolução de problemas de tabelas de dados

Informações de rastreamento para serviços de tabela de dados	65
Pontos de rastreamento de entrada e saída para tabelas de dados compartilhados	65
Pontos de rastreamento de exceção para tabelas de dados compartilhados	69
Analisando erros das tabelas de dados SVC	70
Valores para todos os pontos de rastreamento de tabelas de dados compartilhados	70
Valores para ponto de rastreamento 0B12	71
Valor para ponto de rastreamento 0B19	71
Valores para ponto de rastreamento 0B1A	71
Valores para ponto de rastreamento AP 0B29	72
Valores para ponto de rastreamento 0B2A	72
Analisando erros dos serviços de memória cruzada das tabelas de dados	73
Informações de dump para tabelas de dados	74

Avisos	77
Índice Remissivo	83

Sobre este PDF

Este PDF fornece informações sobre serviços de tabela de dados compartilhados CICS.

Ele destina-se a qualquer pessoa envolvida nas tabelas de dados compartilhados CICS em uma ou mais das seguintes áreas:

- Planeja- mento
- Programação de Aplicativos
- Definição de recursos
- Personalizar
- Operações
- determinação do problema

Para obter detalhes sobre os termos e notação usados, consulte Convenções e terminologia usadas na documentação do CICS no IBM Knowledge Center.

Data deste PDF

Este PDF foi criado em 18 de abril de 2017.

Capítulo 1. Introdução às Tabelas de Dados Compartilhados

O recurso de tabela de dados compartilhados do CICS é uma extensão dos serviços de gerenciamento de arquivo do CICS.

O conceito de tabelas de dados compartilhados

Usando tabelas de dados compartilhados, todos os arquivos que estão definidos como tabelas de dados podem ser compartilhados usando serviços de memória cruzada. Nenhuma mudança é necessária nas definições de arquivo para tabelas de dados existentes.

O conceito de tabelas de dados compartilhados explora o fato de que elas são mais eficientes:

- Para usar serviços de memória cruzada do MVS em vez de remessa de função do CICS para compartilhar um arquivo de dados entre duas ou mais regiões CICS na mesma imagem do MVS.
- Para acessar dados na memória em vez de no DASD.
- Para acessar um arquivo de dados na memória usando serviços integrados ao gerenciamento de arquivo do CICS em vez de usando serviços VSAM e um conjunto de recurso compartilhado local (LSR).

As duas versões de tabelas de dados são:

- Suporte de tabelas de dados básico, fornecido como parte do CICS Transaction Server for z/OS
- Suporte de tabelas de dados compartilhados, fornecido como parte do CICS Transaction Server for z/OS.

O uso de serviços de memória cruzada é um dos principais benefícios das tabelas de dados compartilhados. Ele melhora o desempenho dos aplicativos que atualmente usam remessa de função e torna o compartilhamento de arquivo mais viável para aplicativos que não podem aceitar sobrecarga de desempenho da remessa de função.

O outro principal aprimoramento é que quase todas as solicitações de leitura são suportadas para uso com tabelas de dados. Esse aprimoramento amplia o uso de tabelas de dados para aplicativos que incluem:

- Solicitações de procura
- Solicitações de leitura que usam uma chave imprecisa

Além do uso de serviços de memória cruzada, as tabelas de dados compartilhados oferecem benefícios para desempenho e segurança:

- Grandes reduções de comprimento de caminho podem ser obtidas para acessos remotos porque a remessa de função é evitada para a maioria das solicitações de leitura e procura.
- Quando serviços de memória cruzada forem usados, as solicitações serão processadas pela AOR, liberando a FOR para processar outras solicitações. Isso aumenta a exploração do multiprocessador.

- O aumento da segurança dos dados é fornecido porque informações de registro em tabelas de dados compartilhados são armazenadas fora da região CICS e não são incluídas em dumps do sistema CICS (formatados ou não).
- Para tabelas de dados mantidas pelo CICS, todas as formas de acesso com chave não atualizado (incluindo solicitações de procura e solicitações de leitura de chave imprecisa) são processadas por referência à tabela de dados.
- Para tabelas de dados mantidas pelo usuário, todas as formas de acesso com chave não atualizado (incluindo solicitações de procura e solicitações de leitura de chave imprecisa) são suportadas.
- Qualquer número de arquivos que se referem ao mesmo conjunto de dados de origem que estão abertos ao mesmo tempo pode recuperar dados da tabela de dados mantida pelo CICS.
- Um aprimoramento na saída de usuário XDTRD permite ignorar um intervalo de registros durante o carregamento da tabela de dados.

Descrição de tabelas de dados

Um arquivo CICS é uma representação de um conjunto de dados no DASD. Se você especificar que o arquivo deve usar serviços da tabela de dados, o CICS copiará o conteúdo do conjunto de dados em um espaço para dados MVS quando o arquivo for aberto e usar essa cópia sempre que possível.

Devido à maneira como os serviços da tabela de dados acessam os registros, eles podem ser usados somente com um key-sequenced data set (KSDS) VSAM. O KSDS é chamado *conjunto de dados de origem*. A cópia na memória é chamada *tabela de dados*. O processo de copiar os registros é chamado *carregamento* da tabela de dados.

Quando o arquivo é lido por um aplicativo CICS, normalmente o registro é recuperado da tabela de dados. Quando o arquivo é atualizado pelo aplicativo CICS, o efeito depende do tipo de tabela de dados que você definiu para o arquivo.

Os serviços de tabela de dados do CICS suportam dois tipos de tabela de dados:

- Tabela de dados mantida pelo CICS (CMT)
- Tabela de dados mantida pelo usuário (UMT)

tabela de dados mantida pelo CICS

Uma tabela de dados mantida pelo CICS é uma tabela de dados cujos registros são refletidos automaticamente no conjunto de dados de origem. Quando você atualiza o arquivo, o CICS muda o conjunto de dados de origem e a tabela de dados.

Tratar o conjunto de dados de origem e a tabela de dados como uma única entidade significa que:

- Mudanças no arquivo são feitas no conjunto de dados de origem e na tabela de dados.
- Se outro arquivo for definido para usar o mesmo conjunto de dados de origem, as mudanças que forem feitas por esse arquivo no conjunto de dados de origem também serão feitas na tabela de dados.
- Se outro arquivo for definido para usar o mesmo conjunto de dados de origem, os registros poderão ser recuperados por esse arquivo a partir da tabela de dados.

Uma tabela de dados mantida pelo CICS é fácil de implementar - você precisa conhecer um pouco os serviços da tabela de dados, não precisa mudar seus programas de aplicativo existentes e o suporte de recuperação total do arquivo fica retido.

Um conjunto de dados sendo acessado no modo Record Level Sharing (RLS) não pode ser usado como origem para uma tabela de dados mantida pelo CICS. O conjunto de dados de origem deve ser acessado no modo não RLS.

Tabela de dados mantida pelo usuário

Uma tabela de dados mantida pelo usuário é uma tabela de dados cujos registros não são refletidos automaticamente no conjunto de dados de origem. Quando você atualiza o arquivo, o CICS muda somente a tabela de dados.

Após uma tabela de dados mantida pelo usuário ter sido carregada, ela será independente de seu conjunto de dados de origem; o conjunto de dados de origem não é atualizado quando a tabela de dados é atualizada. Uma tabela de dados mantida pelo usuário permite otimizar os benefícios do uso de uma tabela de dados, permitindo eliminar a atividade no conjunto de dados de origem para solicitações de atualização e de leitura. Portanto, uma tabela de dados mantida pelo usuário é adequada principalmente para aplicativos que fazem atualizações frequentes nos dados de natureza transitória.

Um pequeno número de operações de arquivo não é suportado para tabelas de dados mantidas pelo usuário. Assim, talvez seja necessário fazer pequenas mudanças em programas de aplicativo existentes. Além disso, a recuperação do arquivo é suportada após uma falha de transação, mas não uma falha do sistema.

Um KSDS VSAM base acessado no modo não RLS ou RLS pode ser usado como conjunto de dados de origem para uma tabela de dados mantida pelo usuário. Talvez você queira tornar um conjunto de dados no modo RLS a origem de uma tabela de dados mantida pelo usuário, caso você tenha outras definições de arquivo que acessem o conjunto de dados e o conjunto de dados seja atualizado por outras regiões CICS.

O ambiente de compartilhamento de tabela de dados

O ambiente para compartilhamento de uma tabela de dados é o mesmo que para qualquer arquivo acessado no modo não RLS.

Uma região CICS possui a tabela de dados - essa região é conhecida como *região voltada para arquivos* (FOR). Qualquer outra região que usa a tabela de dados é conhecida como *regiões voltadas para aplicativos* (AOR). Na FOR, o arquivo é conhecido como um *arquivo local*; na AOR, o arquivo é conhecido como *arquivo remoto*.

No contexto de tabelas de dados compartilhados, a FOR também é conhecida como *servidor* e a AOR também é conhecida como *solicitante*.

A mesma região pode ser uma FOR para algumas tabelas de dados e uma AOR para outras.

Para obter informações sobre esses conceitos de intercomunicação, consulte Métodos de Intercomunicação.

O suporte de tabela de dados compartilhados usa compartilhamento de região cruzada sempre que possível para fornecer acesso às tabelas de dados que estão na mesma imagem do MVS que a região CICS solicitante. Isso significa que a maioria dos acessos de leitura dentro da mesma imagem do MVS é satisfeita pelo compartilhamento de região cruzada usando serviços de tabela de dados. Se o compartilhamento de região cruzada não for possível para a solicitação, a remessa de função será usada. Isso significa que solicitações de atualização de regiões CICS dentro da mesma imagem do MVS e todas as solicitações de regiões CICS em diferentes imagens do MVS usam remessa de função. Programação de aplicativo para uma tabela de dados mantida pelo CICS e Programação de aplicativo para uma tabela de dados mantida pelo usuário dizem quando os comandos são satisfeitos pelo compartilhamento de região cruzada ou pela remessa de função.

Nota: Da mesma forma, o XCF/MRO não fornece acesso à tabela de dados compartilhados entre regiões CICS em diferentes imagens do MVS.

Embora o suporte de tabela de dados compartilhados seja destinado principalmente ao compartilhamento de dados dentro de uma imagem do MVS, o suporte pode ser ampliado para um ambiente sysplex para aplicativos que requerem somente acesso de leitura a uma tabela de dados mantida pelo usuário ou que podem operar com dados que podem não estar atualizados. A tabela de dados deve ser replicada em cada região do MVS no sysplex e atualizada periodicamente. Consulte o Usando o suporte de tabelas de dados compartilhados em um sysplex.

O conjunto de dados de origem para tabelas de dados

O conjunto de dados de origem deve ser um KSDS VSAM base, não um índice alternativo. No entanto, atualizações feitas no KSDS via um índice alternativo são refletidas em uma tabela de dados mantida pelo CICS.

A definição de VSAM do KSDS fornece os valores de comprimento da chave e comprimento de registro máximos.

Para uma tabela de dados mantida pelo usuário, as atualizações não são refletidas no conjunto de dados de origem ou seus índices alternativos. Uma tabela de dados mantida pelo usuário é totalmente independente do seu conjunto de dados de origem após o carregamento ser concluído.

Espaços de dados para tabelas de dados

Os registros de tabela de dados são armazenados em um ou mais espaços para dados MVS, independentemente de a tabela de dados ser ou não compartilhada por mais de uma região. Um conjunto separado de espaços para dados é usado para cada região CICS.

O conjunto inicial de espaços para dados, denominado DFHDT001 (para descritores de entrada de tabela), DFHDT002 (para nós de índice) e DFHDT003 (para até 2 GB de dados de registro), é obtido quando o primeiro arquivo definido como uma tabela de dados é aberto na região. Espaços de dados adicionais do DFHDT004 para cima podem ser alocados conforme necessário para registrar dados, até um máximo de 100 espaços para dados por região. Os espaços para dados são usados por todas as tabelas de dados CICS que pertencem a essa região e ficam retidos até a região CICS ser encerrada.

Cada espaço para dados tem o tamanho máximo de 2 GB, portanto, a quantidade máxima de armazenamento de espaço para dados que uma região CICS poderia alocar (supondo que haja recursos suficientes do sistema operacional) é de 200 GB. A saída MVS IEFUSI, que pode ser usada para controlar a quantidade total de espaço para dados que um determinado espaço de endereço pode ter, pode reduzir o tamanho máximo para menos que essa quantidade, até mesmo menos de 2 GB. Dentro desse limite, o CICS aloca armazenamento de espaço para dados em unidades de 16 MB e, em seguida, subaloca esse armazenamento para as tabelas de dados em incrementos de 32 KB para descritores de entrada de tabela ou nós de índice e em incrementos de 128 KB para dados de registro. Se um novo incremento de armazenamento for necessário para uma tabela de dados, mas todo o armazenamento de espaço para dados existente já estiver alocado para as tabelas, o CICS tentará ampliar espaço para dados em 16 MB. Se o espaço para dados for para dados do registro, mas já tiver atingido seu tamanho máximo de 2 GB, e o número máximo de espaços para dados não tiver sido atingido, em vez de ampliar o espaço para dados existente, o CICS criará um novo espaço para dados, que será tratado como uma extensão lógica do conjunto de espaços para dados existente. Se o CICS não puder ampliar o espaço para dados para descritores de entrada de tabela ou nós de índice, como o espaço para dados atingiu o tamanho máximo de 2 GB, ou o CICS não pode alocar mais armazenamento de espaço para dados porque o tamanho total de espaço para dados configurado pela saída IEFUSI da instalação foi atingido, o CICS observará que agora o espaço para dados está cheio.

Se um espaço para dados estiver cheio, quaisquer solicitações de tabela de dados compartilhados que precisarem de armazenamento adicional falharão devido ao armazenamento insuficiente. Para uma tabela de dados mantida pelo CICS, isso significa que quaisquer leituras futuras para os registros afetados (incluindo quaisquer leituras aproximadas perto daquela chave) deverão acessar o conjunto de dados VSAM, usando remessa de função, caso a solicitação não seja emitida a partir da região voltada para arquivos. Para uma tabela de dados mantida pelo usuário, isso significa que o registro não pode ser gravado na tabela. Desenvolvimento para acesso a tabelas de dados tem informações sobre a resposta retornada nessa situação.

O CICS não fornece um recurso para visualizar a atual quantidade total de armazenamento do espaço para dados alocado. No entanto, as estatísticas de controle de arquivos do CICS podem fornecer uma indicação mais exata do armazenamento alocado e usado para cada tabela de dados. Em particular, o campo A17DTALD contém a quantidade de armazenamento de espaço para dados (em KB) atualmente alocada para a tabela.

O armazenamento de espaço para dados que é usado pela tabela de dados é liberado quando o arquivo é fechado na FOR. Esse armazenamento é disponibilizado para reutilização de maneira que a integridade de qualquer AOR usando a tabela de dados fique protegida.

Saídas de usuário globais para tabelas de dados

Três saídas de usuário globais são fornecidas para ampliar o processamento normal realizado pelos serviços de tabela de dados.

- XDTRD seleciona os registros que são copiados na tabela de dados durante o carregamento quando o arquivo está aberto. Para uma tabela de dados mantida pelo usuário, isso também pode ser usado para modificar os registros.
- XDTAD seleciona os registros que são copiados na tabela de dados quando novos registros são incluídos no arquivo.

- XDTLC executa o processamento no final da operação de carregamento.

Serviços de tabela de dados compartilhados e acesso de arquivo remoto

Estes diagramas ilustram a diferença entre usar remessa de função e usar serviços de tabela de dados compartilhados para acessar um arquivo CICS em outra região.

Usando remessa de função

Este diagrama mostra o uso de remessa de função para acessar um conjunto de dados pertencente a outra região CICS.

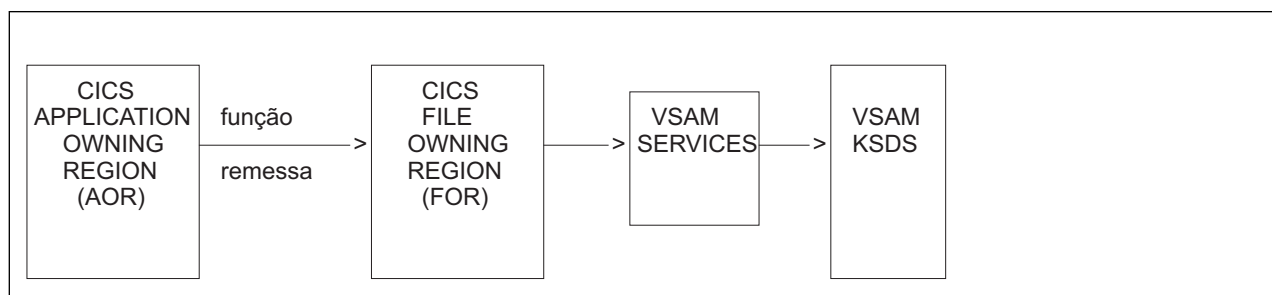


Figura 1. Acesso a dados usando remessa de função

Usando serviços de tabela de dados compartilhados

Este diagrama mostra como um número de AOR pode usar serviços de memória cruzada para executar leituras e procuras, usando serviços de tabela de dados compartilhados em uma FOR para acessar a tabela de dados. (A remessa de função é usada para atualizar solicitações e para qualquer solicitação que precise de acesso ao conjunto de dados de origem, da mesma maneira que a mostrada em Figura 1.)

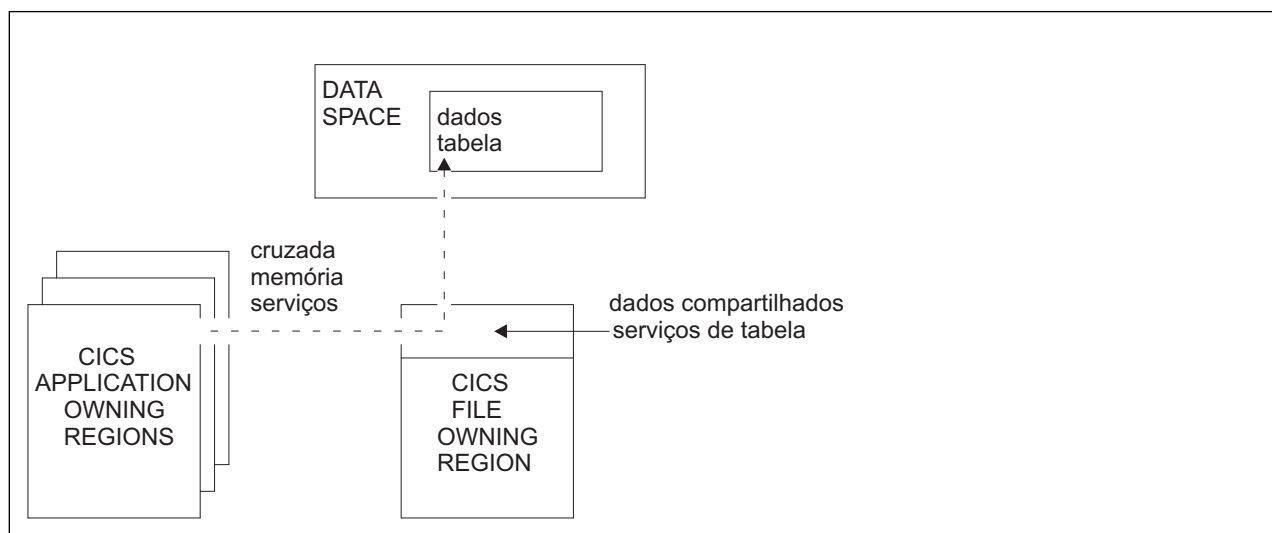


Figura 2. Acesso a dados usando serviços de tabela de dados compartilhados. Este diagrama mostra somente o acesso somente leitura.

Como uma tabela de dados é compartilhada

Duas operações, LOGON e CONNECT, estabelecem uma tabela de dados para compartilhamento.

LOGON

Quando o primeiro arquivo definido como uma tabela de dados é aberto em uma FOR, a FOR tenta se registrar como um servidor de tabela de dados compartilhados. Essa operação é executada automaticamente e é conhecida como **SDT LOGON**. A abertura do arquivo pode ser causada pela FOR ou AOR quando elas acessam o arquivo pela primeira vez.

Independentemente de a operação LOGON ser ou não bem-sucedida, o arquivo será aberto e a tabela de dados será carregada. Se a LOGON for bem-sucedida, todas as outras regiões CICS no sistema operacional MVS serão notificadas de que a tabela de dados está disponível.

Se a LOGON falhar devido a uma condição permanente (como o CICS não ser definido como um subsistema MVS), nenhuma outra tentativa de LOGON será feita durante a execução do CICS.

Se o LOGON falhar devido a uma condição potencialmente temporária, será feita outra tentativa de LOGON na próxima vez que um arquivo definido como tabela de dados for aberto. Esse tipo de condição inclui:

- Falha de uma verificação de segurança
- Falha de obtenção de armazenamento
- Falha de carregamento de programa

Quando solicitações LOGON da região são rejeitadas devido a uma falha na verificação de segurança, mensagens de violação de segurança podem ser emitidas cada vez que um arquivo definido como tabela de dados for aberto.

Após uma FOR efetuar logon com sucesso, ela permanece nesse estado pelo restante da execução do CICS; nenhuma outra solicitação LOGON é emitida.

CONEXÃO

Quando uma AOR com suporte de tabela de dados compartilhados emite uma solicitação de leitura (ou inicia uma sequência de procura) para um arquivo remoto, o CICS tenta estabelecer uma conexão com uma tabela de dados para esse arquivo. Essa operação é executada automaticamente e é conhecida como **SDT CONNECT**.

Se a FOR for registrada como um servidor de tabela de dados compartilhados, o CICS estabelecerá um link de memória cruzada da AOR para a FOR (sujeito a verificações de segurança) e chamará o servidor de tabela de dados compartilhados para perguntar se existe alguma tabela de dados disponível para o arquivo. Se houver, será estabelecida uma conexão entre a AOR e a tabela de dados.

Se a CONEXÃO for bem-sucedida, serviços de memória cruzada serão usados, sempre que possível, para acessar o arquivo enquanto a conexão existir.

Se a CONEXÃO falhar, a solicitação de arquivo terá uma remessa de função exatamente como teria se o suporte de tabela de dados compartilhados não estivesse disponível. A ação tomada para solicitações de arquivo remoto subsequentes depende do tipo de falha, conforme descrito abaixo.

Se a CONEXÃO falhar devido a uma condição permanente (como o CICS não ser definido como um subsistema MVS), nenhuma outra tentativa de CONEXÃO será feita durante a execução do CICS.

Se a CONEXÃO falhar devido a uma condição potencialmente temporária que não esteja sob o controle do proprietário do arquivo, outra tentativa de CONEXÃO será feita para a próxima solicitação adequada após cerca de dez minutos. Esse tipo de condição inclui:

- Falha de uma verificação de segurança
- Falha de obtenção de armazenamento
- Falha de carregamento de programa

Quando solicitações de CONEXÃO da região são rejeitadas devido a uma falha na verificação de segurança, as mensagens de violação de segurança relacionadas podem ser emitidas em intervalos de 10 minutos.

Se a CONEXÃO falhar devido a uma condição potencialmente temporária que esteja sob o controle do proprietário do arquivo, outra tentativa de CONEXÃO será feita para a próxima solicitação adequada após a notificação de que pelo menos um novo arquivo está disponível para acesso compartilhado no sistema MVS. Esse tipo de condição inclui:

- O proprietário do arquivo não efetuou logon como um servidor
- O arquivo não está associado a uma tabela de dados
- O arquivo está desativado, embora esteja associado a uma tabela de dados
- O arquivo está fechado, embora esteja definido como uma tabela de dados

Após uma AOR se conectar a um arquivo remoto com sucesso, ela permanecerá conectada, a menos que ocorra um dos eventos a seguir:

- A AOR exclui sua definição de arquivo remoto
Nesse caso, a conexão é interrompida imediatamente.
- A FOR fecha ou desativa o arquivo
Nesse caso, a desconexão é planejada na próxima solicitação de não atualização e é efetivada após todas as sequências de procura atuais serem finalizadas.
Consulte o Desconexão.

Se esses eventos forem invertidos posteriormente, uma conexão válida será estabelecida da mesma maneira que antes.

Notificação de que um novo arquivo está disponível para acesso compartilhado

Quando uma tabela de dados é aberta por uma FOR, ela se torna disponível para tentativas de CONNECT no início do carregamento para uma tabela de dados mantida pelo CICS ou na conclusão do carregamento de uma tabela de dados mantida pelo usuário. Outras regiões CICS são notificadas de que uma tabela de dados se tornou disponível. A notificação também é feita quando uma tabela de dados (ou um arquivo usando uma tabela de dados mantida pelo CICS) é ativada, tendo sido desativada anteriormente.

Capítulo 2. Configurando tabelas de dados compartilhados

É possível configurar tabelas de dados compartilhados para compartilhar arquivos usando serviços de memória cruzada. O recurso de tabelas de dados compartilhados é uma extensão para os serviços de gerenciamento de arquivo do CICS e pode melhorar o desempenho de aplicativos que usam remessa de função.

Planejando para Usar as Tabelas de Dados

A principal razão para usar as tabelas de dados é aproveitar os benefícios de desempenho.

Essa seção contém as Informações de Diagnóstico, Modificação ou Ajuste.

Desempenho de uma tabela de dados mantida pelo CICS

Se todos os registros de índice e dados de um arquivo estiverem completamente em um conjunto de LSR, a definição de um arquivo como uma tabela de dados mantida pelo CICS não reduz a atividade de E/S do DASD. No entanto, há um potencial de redução considerável no consumo de CPU. Além disso, você pode reduzir o número de buffers no conjunto de LSR.

Se o arquivo não estiver completamente em um conjunto de LSR, o uso de uma tabela de dados mantida pelo CICS pode resultar em reduções na atividade de E/S do DASD e no consumo de CPU.

A economia no consumo de CPU para uma tabela de dados mantida pelo CICS, comparada com um VSAM KSDS residente em um conjunto Recurso Compartilhado Local (LSR), depende do uso do aplicativo.

Desempenho de uma tabela de dados mantida pelo usuário

Após o carregamento de uma tabela de dados mantida pelo usuário, a atividade de E/S do DASD é eliminada de todas as operações da tabela de dados, portanto, a economia do consumo de CPU em comparação com um VSAM KSDS residente em um conjunto de LSR é considerável.

Uso de armazenamento para tabelas de dados compartilhados

Tabelas de dados compartilhados fornecem uso eficiente de dados na memória. Isso significa que são obtidos benefícios de desempenho consideráveis às custas de alguns usos adicionais de armazenamento.

Esta visão geral do uso de armazenamento supõe que você tenha entendido a distinção entre vários tipos de armazenamento, como armazenamento virtual e real, e armazenamento de espaço de endereço e espaço para dados. A maioria do armazenamento usado é o armazenamento de espaço para dados, que é o armazenamento virtual separado do armazenamento virtual de espaço de endereço.

Tabelas de dados compartilhados usam armazenamento virtual da seguinte forma:

- Dados de registro são armazenados em espaços para dados DFHDT003, DFHDT004, DFHDT005 e assim por diante, com novos espaços sendo alocados conforme necessário. O armazenamento de dados do registro total no momento

do carregamento é basicamente o tamanho total de todos os registros (sem chaves, que são armazenados em um armazenamento de entrada de tabela), mais uma pequena quantidade de informações de controle. O armazenamento de espaço para dados é adquirido em unidades de 16 MB e alocado para tabelas individuais em incrementos de 128 KB. O armazenamento é então subalocado em quadros alinhados por página grandes o suficiente para conter o comprimento máximo de registro para a tabela. Quadros de tabela de dados são pouco equivalentes a intervalos de controle VSAM e, normalmente, mantêm um conjunto de registros com chaves semelhantes. Onde for possível, cada novo registro é armazenado no mesmo quadro que o registro existente com a menor chave mais próxima.

Se vários registros aumentarem de comprimento após o carregamento, ou se novos registros forem incluídos aleatoriamente em uma grande parte do arquivo, a quantidade de armazenamento aumentará, provavelmente até duas vezes o tamanho original.

- O armazenamento do descritor de entrada de tabela é alocado do espaço para dados DFHDT001. Ele é alocado em incrementos de 32 KB.
Há um descritor de entrada para cada registro na tabela, mais um descritor de entrada para cada diferença na sequência de chaves (em que um ou mais registros tenham sido omitidos em uma tabela de dados mantida pelo CICS). O tamanho de cada entrada é o comprimento da chave + 9 bytes, arredondado para o próximo múltiplo de 8 bytes.
- O armazenamento do nó de índice é alocado do espaço para dados DFHDT002. Ele é alocado em incrementos de 32 KB.

O tamanho dessa área depende da distribuição e do formato dos valores de chave, bem como do número real de registros, conforme indicado em Tabela 1.

Tabela 1. Formato e distribuição de chave

Distribuição de chave	Formato da chave	Bytes por registro
Denso (todas as chaves são consecutivas)	binária	5.1
	decimal	8.5
	alfabético	19
Esperso (nenhuma chave é consecutiva)	decimal	44
	alfabético	51
Pior caso possível	-	76

- O armazenamento ECSA é usado para alguns pequenos blocos de controle que precisam ser acessados por todas as regiões que compartilham tabelas de dados.

A conversão de um arquivo em uma tabela de dados compartilhados leva a um aumento no uso do armazenamento real, mas o uso do armazenamento real para buffers VSAM LSR pode ser reduzido se forem feitas algumas atualizações. Além disso, um aplicativo que atualmente atinge um alto desempenho replicando tabelas somente leitura em cada região CICS pode estar apto para fazer grandes economias de armazenamento, compartilhando uma única cópia de cada tabela.

Após o armazenamento ser alocado para uma tabela de dados, ele permanece alocado para aquela tabela específica, até que a tabela seja fechada. Por exemplo, se uma tabela de dados aumentar para 1 GB e, depois, todos os registros forem excluídos da tabela, a tabela continuará aumentando 1 GB de armazenamento de espaço para dados. Nenhuma outra tabela de dados poderá usar esse armazenamento até que a tabela de dados proprietária seja fechada.

O espaço livre dentro de uma tabela de dados é controlado e reutilizado quando apropriado. Por exemplo, quando descritores de entrada de tabela ou nós de índice não são mais necessários, eles são incluídos em uma cadeia livre para reutilização dentro da mesma tabela. Da mesma forma, quando todos os registros em um quadro de dados do registro são excluídos, o quadro vazio volta em uma cadeia livre. Quando somente alguns dos registros em um quadro são excluídos, o espaço é reutilizado somente se um novo registro tem uma chave que segue imediatamente outro registro existente no mesmo quadro (ou no quadro anterior, se não houver espaço nesse quadro). Ao contrário de intervalos de controle VSAM, os registros dentro de um quadro não estão necessariamente em uma sequência-chave, pois são localizados indiretamente por meio de descritores; e registros não podem ser movidos para consolidar espaço livre, pois isso não permitiria uma leitura simultânea.

Quando registros são chaves alocadas que aumentam continuamente e que estão sendo excluídas aproximadamente na mesma sequência, normalmente o espaço é reutilizado de forma muito eficiente, pois normalmente os novos registros preenchem um quadro antes de irem para o próximo; e, eventualmente, os quadros antigos ficam completamente vazios, o que permite que eles sejam reutilizados. Esse também é o caso para aumentar chaves dentro de vários intervalos separados, contanto que os intervalos sejam grandes o suficiente para que quadros inteiros sejam liberados. Nessa situação, a quantidade de armazenamento alocado para tabelas de dados se aproxima da quantidade de armazenamento em uso.

Quando novos aplicativos de tabela de dados forem introduzidos, poderá ser útil monitorar o armazenamento alocado e o armazenamento em uso para cada tabela de dados a fim de assegurar que recursos de sistema operacional suficientes estejam disponíveis para suportar o uso atual e futuro. As leituras para o armazenamento alocado mostram o armazenamento pertencente a cada tabela de dados, que não será fornecido até que a tabela de dados seja excluída. As leituras para o armazenamento em uso mostram quanto do armazenamento alocado está em uso. O programa de estatísticas de amostra do CICS, DFH0STAT, fornece essas informações. DFH0STAT está descrito em .

É possível que tabelas de dados compartilhados possam ficar sem espaço para quaisquer descritores, entradas de índice ou dados. A falta de espaço pode ocorrer não só no momento do carregamento, mas também durante a execução normal, quando registros estão sendo incluídos ou até mesmo atualizados. Como agora o CICS usa vários espaços para dados para suportar tabelas de dados compartilhados, os limites para os três tipos de armazenamento aumentam muito e se tornam independentes de outras considerações; por exemplo, as entradas não estão mais dentro do espaço de endereço do CICS. Todavia, o armazenamento disponível ainda é finito. Por exemplo, pode haver números muitos grandes de registros relativamente pequenos, especialmente se consistirem principalmente em dados-chave, em cujo caso, os descritores de entrada ou os nós de índice podem esgotar-se antes do armazenamento para os dados do registro em si, dependendo do comprimento da chave e de outros fatores. Se houver espaço insuficiente para descritores de entrada ou nós de índice, considere dividir as tabelas de dados em diferentes regiões CICS; por exemplo, diferentes FORs. Se uma única tabela de dados ficar sem espaço, seu limite de espaço foi atingido, em cujo caso deve-se considerar se ela deveria ser dividida em duas ou mais tabelas separadas.

Requisitos de JCL do MVS ao usar tabelas de dados compartilhados

Antes de usar tabelas de dados compartilhados, talvez você precise mudar algumas de suas instruções JCL, modificar seus procedimentos operacionais ou aumentar o valor do parâmetro de inicialização MAXUSER do MVS.

Isso porque o MVS não permite que mais de uma etapa de uma tarefa aja como um servidor de tabela de dados compartilhados. Se uma segunda etapa da tarefa tentar agir como um servidor de tabela de dados compartilhados, o CICS emitirá a mensagem DFHFC0405. Além disso, como etapas da tarefa após a etapa do servidor também não poderiam usar serviços de memória cruzada com MRO, é recomendado que nenhuma das etapas da tarefa após a etapa do servidor seja outra execução do CICS.

Se uma tarefa incluindo uma etapa do servidor de tabelas de dados compartilhados terminar antes de todas as etapas da tarefa do solicitante conectadas a esse servidor serem finalizadas, o espaço de endereço do servidor será finalizado pelo MVS. Se o servidor de tabela de dados compartilhados estiver em execução sob o controle de um inicializador em lote, e não como uma tarefa iniciada, um novo inicializador deverá ser iniciado quando ocorrer essa situação.

O MVS finaliza o inicializador em lote com a mensagem *IEF355A INITIATOR TERMINATED, RESTART INITIATOR* pois, por motivos de integridade, o MVS teria que restringir de alguma forma as funções que poderiam ser usadas pela próxima tarefa que é executada sob o inicializador, o que pode fazer a tarefa falhar. O MVS não permite que o ASID do servidor de tabela de dados compartilhados seja reutilizado após todas as etapas da tarefa do solicitante conectadas ao servidor serem finalizadas.

Selecionando arquivos para usar como tabelas de dados

Não é possível determinar nenhuma regra exata sobre se um arquivo se beneficiará da conversão em uma tabela de dados compartilhados. A lista de verificação neste tópico fornece uma orientação geral.

Existem várias considerações, e uma análise dos possíveis usos de suporte de tabelas de dados compartilhados deve ser realizada por alguém que entenda como os arquivos são usados pelos vários aplicativos e a configuração das regiões CICS.

Fontes de informações adicionais que poderiam ajudá-lo a selecionar os arquivos incluem:

- Estatísticas do arquivo. “Usando estatísticas para selecionar tabelas de dados” na página 14 descreve como é possível usar informações de estatísticas como uma das entradas para a tarefa de seleção.
- As estatísticas do conjunto de LSR.
- Entradas de rastreamento.
- Dados de monitoramento.

No entanto, a entrada mais benéfica para o processo de seleção é o total entendimento dos aplicativos e da maneira como eles usam os arquivos.

Se sua instalação estiver usando tabelas de dados pela primeira vez, a lista de verificação a seguir fornecerá alguns princípios gerais para ajudá-lo a selecionar arquivos para definir como tabelas de dados.

- É necessário considerar usar uma tabela de dados mantida pelo CICS primeiro, já que ela é mais fácil de implementar. Se você usar uma tabela de dados mantida pelo CICS, nenhuma mudança será necessária nos aplicativos. Se você usar uma tabela de dados mantida pelo usuário, algumas mudanças podem ser necessárias.
- Use uma tabela de dados mantida pelo CICS se você precisar assegurar a integridade da tabela de dados em uma reinicialização do CICS.
- Use uma tabela de dados mantida pelo CICS se você precisar do registro de atualizações em diário. Se você precisar registro no diário todas as solicitações de acesso, o arquivo não será adequado como uma tabela de dados.
- As saídas de usuário da interface exec XEIIIN e XEIOUT, e as saídas de usuário de controle de arquivos XFCREQ e XFCREQC, não são chamadas na região voltada para arquivos, caso uma solicitação para acessar uma tabela de dados seja satisfeita pelos serviços de memória cruzada. Ao selecionar um arquivo, é necessário assegurar que a operação bem-sucedida do seu arquivo não dependa de nenhuma atividade executada nessas saídas de usuário.
- É necessário estar ciente das implicações de segurança do compartilhamento de uma tabela de dados, conforme descrito em “Verificação de segurança para tabelas de dados” na página 18.
- Se um arquivo for acessado com frequência a partir de outra região, ou se ele for acessado por várias outras regiões, ou se os acessos forem predominantemente solicitações de leitura, os benefícios de torná-lo uma tabela de dados podem ser muitos. Lembre-se de que o ganho de desempenho para um arquivo remoto é maior que para um arquivo local.
- Para uma tabela de dados mantida pelo CICS, selecione arquivos que tenham uma proporção razoavelmente alta de solicitações que acessem somente a tabela de dados (consulte Programação do Aplicativo para Tabelas de Dados). Entre eles, selecione os arquivos com o maior uso dessas solicitações para maximizar os ganhos de desempenho.
Informações sobre o uso de arquivos podem ser localizadas em Estatísticas de controle de arquivos em relatórios DFHSTUP. Nem todas as solicitações de leitura podem aproveitar as vantagens da tabela de dados, portanto, é necessário verificar as informações da tabela de dados no relatório de estatísticas do CICS após verificar se a tabela de dados está sendo usada efetivamente. Consulte Tabelas de dados de monitoramento para obter informações adicionais.
- Para uma tabela de dados mantida pelo usuário, selecione arquivos que tenham uma grande proporção de atividade de atualização, mas que não requerem que as atualizações sejam recuperadas em uma reinicialização do CICS (consulte “Integridade de Dados” na página 23).
- Use medidas de desempenho para estimar as economias aproximadas de CPU, lembrando-se de quaisquer previsões para uso futuro.
- Selecione um ou dois arquivos com as melhores estimativas. Prefira um arquivo pequeno a um arquivo grande quando as economias estimadas forem semelhantes, pois um arquivo pequeno provavelmente usará menos armazenamento real.
- Monitore o consumo de armazenamento real. Se seu sistema já for restrito por armazenamento real, o uso de uma tabela de dados grande poderia aumentar as taxas na página. Isso, por sua vez, poderia afetar adversamente o desempenho do sistema CICS. Use suas ferramentas normais de desempenho, como RMF(Versão 5) para consultar o uso de armazenamento real e as taxas de paginação.

- Considere reduzir o número de buffers no conjunto de LSR, pois o uso de tabelas de dados poderia reduzir o número de vezes que o conjunto de LSR é usado.
- É possível usar a saída de usuário XDTRD para selecionar os registros incluídos na tabela de dados. Além disso, para uma tabela de dados mantida pelo usuário, é possível usar a saída de usuário XDTRD para modificar os registros. É possível otimizar o uso dos armazenamentos real e virtual armazenando na tabela de dados somente os dados necessários.
- Uma tabela de dados muito grande requer mais armazenamento de espaço para dados do que seu limite de região usual configurado pela saída IEFUSI do MVS. Nesse caso, é possível aumentar o limite modificando a saída IEFUSI ou usar um programa de saída de usuário global CICS XDTRD para suprimir alguns registros. A saída IEFUSI é descrita no manual *z/OS MVS Installation Exits* (SA22-7593).

Usando estatísticas para selecionar tabelas de dados

Se seu compartilhamento estiver confinado em uma única imagem do MVS, é necessário considerar quais arquivos têm padrões de acesso que tornam o uso de tabelas de dados compartilhados benéfico.

Se você precisar compartilhar dados entre mais de uma imagem do MVS, é necessário investigar o uso do modo RLS para compartilhar arquivos.

Figura 3 na página 15, Figura 4 na página 15 e Figura 5 na página 16 mostram algumas extrações de um conjunto hipotético de estatísticas de arquivo para arquivos acessados em um modo não RLS que são usados na discussão a seguir para demonstrar como estatísticas do CICS podem ajudar no processo de seleção.

As estatísticas são exibidas como seriam relatadas pelo utilitário de formatação off-line do CICS. As estatísticas de arquivo solicitadas são mostradas, mas estatísticas Intervalo ou Fim do Dia seriam igualmente adequadas. A seção de estatísticas “Informações de Desempenho” do Arquivo, que relata o uso de buffers e sequências VSAM, não é mostrada aqui.

Os números mostrados nas figuras têm o propósito exclusivo de ilustração, e não se deve esperar que as estatísticas em sua instalação se pareçam com eles. Da mesma forma, a configuração de arquivos e regiões CICS foi escolhida para destacar certos pontos; não é sugerido que essa seja a configuração típica ou desejável.

Tabelas de dados de monitoramento discute as estatísticas relatadas para arquivos definidos em tabelas de dados, que podem ser usadas para avaliar os benefícios sendo obtidos.

Requested Statistics Report Collection Date-Time 12/25/99-11:51:51 Last Reset 09:00:00 Applid CICFOR Jobname SDTGSTF1

FILES - Resource Information

File Name	Data Set Name Base Data Set Name (If Applicable)	Data Set Type	RLS File	DT Indicator	Time Opened	Time Closed	Remote Name	Remote Sysid	Lsrpool ID
APPLE	CIC01.CICOWN.APPLES	K	NO		07:44:12	OPEN			1
BANANA	CIC01.CICOWN.BANANAS	K	NO		09:45:08	OPEN			1
ORANGE	CIC01.CICOWN.CITRUS	K	NO		10:51:10	OPEN			2
PEAR	CIC01.CICOWN.PEARS	K	NO		07:30:14	OPEN			3

Requested Statistics Report Collection Date-Time 12/25/99-11:51:51 Last Reset 09:00:00 Applid CICFOR Jobname SDTGSTF1

FILES - Requests Information

File Name	Get Requests	Get Upd Requests	Browse Requests	Update Requests	Add Requests	Delete Requests	Brws Upd Requests	VSAM Data	EXCP Index	Requests	RLS req Timeouts
APPLE	2317265	1020	0	1019	21	1	0	11503	310	0	
BANANA	536452	1674	20344	1674	908	0	0	2651	70	0	
ORANGE	2069454	98560	17831	98327	4543	2563	0	8511	481	0	
PEAR	45871	65493	6512	65493	30109	362	0	3773	231	0	

TOTALS 4969042 166747 44687 166513 35581 2926 0 0

Requested Statistics Report Collection Date-Time 12/25/99-11:51:51 Last Reset 09:00:00 Applid CICFOR Jobname SDTGSTF1

FILES - Data Table Requests Information

File Name	Close Type	Read Requests	Recs ~ in Table	Adds from Reads	Add Requests	Adds rejected - Exit	Adds rejected - Table Full	Rewrite Requests	Delete Requests	Highest Table Size	Storage Alloc(K)
-----------	------------	---------------	-----------------	-----------------	--------------	----------------------	----------------------------	------------------	-----------------	--------------------	------------------

DFHST0223 I There are no data table statistics to report.

Figura 3. Estatísticas do arquivo solicitado CICFOR

Requested Statistics Report Collection Date-Time 12/25/99-11:51:38 Last Reset 09:00:00 Applid CICAOR1 Jobname SDTGSTA1

FILES - Resource Information

File Name	Data Set Name Base Data Set Name (If Applicable)	Data Set Type	RLS File	DT Indicator	Time Opened	Time Closed	Remote Name	Remote Sysid	Lsrpool ID
APPLE	REMOTE				CLOSED	CLOSED	APPLE	CIF1	N
BANANA	REMOTE				CLOSED	CLOSED	BANANA	CIF1	N
ORANGE	REMOTE				CLOSED	CLOSED	ORANGE	CIF1	N
ZUCCHINI	REMOTE				CLOSED	CLOSED	COURGETT	CIA2	N

Requested Statistics Report Collection Date-Time 12/25/99-11:51:38 Last Reset 09:00:00 Applid CICAOR1 Jobname SDTGSTA1

FILES - Requests Information

File Name	Get Requests	Get Upd Requests	Browse Requests	Update Requests	Add Requests	Delete Requests	Brws Upd Requests	VSAM Data	EXCP Index	Requests	RLS req Timeouts
APPLE	1158701	532	0	531	11	1	0	0	0	0	
BANANA	305641	0	19067	0	0	0	0	0	0	0	
ORANGE	58709	32854	4265	32621	1018	1001	0	0	0	0	
ZUCCHINI	78914	0	14765	0	0	0	0	0	0	0	

TOTALS 1601965 33386 38097 33152 1029 1002 0 0

Requested Statistics Report Collection Date-Time 12/25/99-11:51:38 Last Reset 09:00:00 Applid CICAOR1 Jobname SDTGSTA1

FILES - Data Table Requests Information

File Name	Close Type	Read Requests	Recs ~ in Table	Adds from Reads	Add Requests	Adds rejected - Exit	Adds rejected - Table Full	Rewrite Requests	Delete Requests	Highest Table Size	Storage Alloc(K)
-----------	------------	---------------	-----------------	-----------------	--------------	----------------------	----------------------------	------------------	-----------------	--------------------	------------------

DFHST0223 I There are no data table statistics to report.

Figura 4. Estatísticas do arquivo solicitado CICAOR1

FILES - Resource Information

File Name	Data Set Name Base Data Set Name (If Applicable)	Data Set Type	RLS File	DT Indicator	Time Opened	Time Closed	Remote Name	Remote Sysid	Lsrpool ID
COURGETT LEMON	CIC02.CICOWN.COURGETT REMOTE	K	NO NO		08:22:15 CLOSED	OPEN CLOSED	ORANGE	CIF1	1 N

FILES - Requests Information

File Name	Get Requests	Get Upd Requests	Browse Requests	Update Requests	Add Requests	Delete Requests	Brws Upd Requests	VSAM EXCP Data	Requests Index	RLS req Timeouts
COURGETT LEMON	78914 2010745	27469 65706	14765 13566	27469 65706	336472 3525	0 1562	0 0	8212 0	481 0	0 0

TOTALS 2089659 93175 28331 93175 339997 1562 0 0 0

FILES - Data Table Requests Information

File Name	Close Type	Read Requests	Recs in Table	Adds from Reads	Add Requests	Adds rejected - Exit	Adds rejected - Table Full	Rewrite Requests	Delete Requests	Highest Table Size	Storage Alloc(K)
-----------	------------	---------------	---------------	-----------------	--------------	----------------------	----------------------------	------------------	-----------------	--------------------	------------------

DFHST0223 I There are no data table statistics to report.

Figura 5. Estatísticas do arquivo solicitado CICAOR2

Os exemplos usam uma configuração hipotética de três regiões CICS. A maioria dos arquivos usados pelos aplicativos CICS pertence à região voltada para arquivos CICFOR e os aplicativos são executados principalmente em regiões voltadas para aplicativos CICAOR1 e CICAOR2. Essa discussão supõe que cada conjunto de dados mostrado nos relatórios de estatísticas seja um KSDS base VSAM (conforme indicado pelo Tipo de Conjunto de Dados de K), assim, qualquer um deles pode ser definido como tabelas de dados.

Esta seção está focada na identificação de candidatos para definir como tabelas de dados mantidas pelo CICS, pois a decisão de definir uma tabela de dados mantida pelo usuário tem mais chance de vir da consideração de determinados aplicativos do que de um estudo do desempenho geral do arquivo. Devido a esse foco, nenhuma das estatísticas mostradas é para arquivos acessados no modo RLS, pois o conjunto de dados do modo RLS não pode ser a origem de uma tabela de dados mantida pelo CICS.

As estatísticas também mostram quais nomes de arquivo em uma região estão definidos para acessar nomes de arquivos em outra região. O *Sysid Remoto* é o nome dado na conexão entre duas regiões. Nos exemplos, o SYSID de CICFOR é CIF2 e de CICAOR2 é CIA2.

Um arquivo com uma alta razão de leitura-para-atualização

O arquivo APPLE é usado pelos aplicativos que são executados nas regiões voltadas para aplicativos CICAOR1. Ele é definido no CICAOR1 como um arquivo remoto, e a definição de arquivo aponta para o arquivo APPLE pertencente ao CICFOR.

Esse arquivo seria beneficiado se fosse redefinido no CICFOR como uma tabela de dados mantida pelo CICS, pois tem uma alta razão de leituras remotas (1158701 obter Solicitações no período coberto pelos relatórios) para atualizações remotas (11 inclusões, 1 exclusão e 531 atualizações), conforme visto em Figura 4 na página 15.

Consulte Estatísticas de controle de arquivos em relatórios DFHSTUP para obter orientação sobre os significados da seção “ARQUIVOS - Informações sobre Solicitações” de um relatório de estatísticas.

Um arquivo com alta proporção de leituras remotas

O arquivo BANANA é atualizado e lido no CICFOR, mas também é acessado por CICAOR1.

Como todos os acessos remotos são leituras e procuras, sem atualizações, os aplicativos em execução no CICAOR1 provavelmente veriam grandes benefícios se BANANA fosse definido como uma tabela de dados e os aplicativos em CICFOR se beneficiariam com a leitura da tabela de dados locais.

Um arquivo compartilhado por várias regiões

Pode parecer que ORANGE não é um candidato à tabela de dados adequado.

As estatísticas em Figura 4 na página 15 mostram que os números de recuperações remotas de CICAOR1 (58709 Obter Solicitações e 4265 Procurar Solicitações) são relativamente baixos. No entanto, o arquivo remoto LEMON no CICAOR2 também aponta para ORANGE no CICFOR, portanto, a definição de ORANGE no CICFOR como uma tabela de dados mantida pelo CICS compartilhada provavelmente beneficiaria o desempenho dos aplicativos em ambas as AORs.

Um bom candidato a UMT

O arquivo COURGETT pertencente ao CICAOR2 é acessado por meio do nome do arquivo ZUCCHINI no CICAOR1.

CICAOR1 somente lê ou procura o arquivo; qualquer atualização é emitida pela região proprietária. Além disso, sabe-se que essas atualizações são relevantes somente para a execução do CICS do dia e não precisam ser retidas permanentemente (de fato, elas são excluídas no encerramento). Portanto, o arquivo é um excelente candidato para ser definido como uma tabela de dados mantida pelo usuário. Todas as atualizações podem ser feitas na tabela de dados sem qualquer atividade de E/S do VSAM, e todas as recuperações remotas podem ser feitas sem remessa de função.

Um candidato fraco

O arquivo PEAR provavelmente não se beneficiaria muito do suporte de tabelas de dados compartilhados, pois ele não é acessado remotamente e tem muitas solicitações de procura e atualização.

A procura local não oferece muitos benefícios como a leitura local ou qualquer forma de recuperação remota, pois a procura de VSAM (sem considerar o processamento do comando STARTBR) é muito eficiente. Essa análise, é claro, não considera a importância relativa dos vários acessos de arquivo; a leitura pode ser feita por aplicativos críticos, mas o tempo levado para as atualizações pode não ser importante.

Outros possíveis candidatos

Os exemplos anteriores ilustram somente uma pequena amostra das possíveis configurações e usos de arquivos que poderiam se beneficiar do suporte de tabelas de dados compartilhados.

Você também poderia usar o suporte de tabelas de dados compartilhados para evitar a necessidade de arquivos duplicados ou tabelas de dados em cada região. Além de verificar os arquivos existentes, considere mover arquivos de um AOR para um FOR. Mover arquivos de um AOR para um FOR não era possível quando o suporte a tabelas compartilhadas de dados não estava disponível por causa do custo dos acessos a arquivos usando remessa de função.

Verificação de segurança para tabelas de dados

Tabelas de dados compartilhados fazem verificações de segurança no momento do LOGON ou da CONEXÃO para fornecer segurança quando serviços de memória cruzada forem usados. É necessário considerar as implicações das verificações de segurança antes de compartilhar um arquivo que esteja associado a uma tabela de dados.

Tabelas de dados compartilhados devem assegurar que:

- A FOR não possa ser personificada. Isso é evitado verificando no momento do LOGON se a FOR tem permissão para efetuar logon com o *applid* genérico especificado da região CICS.
- Uma AOR não possa acessar dados que não pode ver. Isso é evitado verificando no momento da CONEXÃO se a AOR tem permissão para acessar a FOR e, se a segurança do arquivo estiver em vigor, se a AOR tem permissão para acessar o arquivo solicitado.

Essas verificações de segurança são feitas usando o System Authorization Facility (SAF) para chamar o Resource Access Control Facility (RACF) ou um gerenciador de segurança equivalente.

Nota: Uma região ainda pode usar tabelas de dados localmente, mesmo se não tiver autoridade para agir como um servidor de tabela de dados compartilhados.

O suporte de tabela de dados compartilhados reproduz as principais características da segurança da remessa de função que operam em nível de região, mas as diferenças a seguir devem ser observadas:

- O suporte de tabela de dados compartilhados não fornece nenhum mecanismo para a FOR executar verificações de segurança em nível da transação (o equivalente de ATTACHSEC(IDENTIFY) ou ATTACHSEC(VERIFY)). Portanto, se você considerar que as verificações em nível da transação executadas pela AOR são inadequadas para alguns arquivos, deve-se assegurar que esses arquivos não estejam associados às tabelas de dados na FOR.
- O suporte de tabela de dados compartilhados não suporta segurança pré-configurada.
- O suporte de tabela de dados compartilhados não passa nenhuma informação da lista de parâmetros de instalação (INSTLN) para saídas de usuário de segurança.

Para obter uma descrição das etapas necessárias para implementar segurança de tabela de dados compartilhados, consulte Classes RACF para proteger recursos do sistema.

Verificação de segurança de LOGON

O processamento de LOGON inclui uma verificação de segurança para verificar se a FOR está autorizada a agir como um servidor com o nome do aplicativo especificado.

Esta verificação minimiza o risco de regiões voltadas para aplicativos (AOR) aceitarem registros de dados falsificados de uma região voltada para arquivos (FOR) que é, de fato, um impostor. A verificação nunca é ignorada, mesmo quando SEC=NO é especificado na inicialização do sistema.

Verificações de segurança CONNECT

As verificações de segurança realizadas no momento da CONEXÃO fornecem dois níveis de segurança.

Segurança de ligação

Permite que uma FOR que é executada sem segurança do arquivo CICS possa restringir o acesso compartilhado a AORs selecionadas. (A execução sem segurança do arquivo minimiza sobrecargas de tempo de execução e o número de definições de segurança.)

Segurança do Arquivo

Pode ser ativada na FOR se você precisar de maior granularidade na verificação de segurança. O suporte de tabela de dados compartilhados então implementa essas verificações que se aplicam à AOR como um todo.

O suporte de tabela de dados compartilhados não fornece um meio de implementar essas verificações de segurança feitas por uma FOR em nível de transação quando ATTACHSEC(IDENTIFY) ou ATTACHSEC(VERIFY) é usado com remessa de função.

Preparando-se para usar suporte de tabela de dados compartilhados

Para usar suporte de tabela de dados compartilhados, deve-se executar as tarefas a seguir. Algumas delas já terão sido executadas para uma instalação que usa atualmente remessa de função ou tabelas de dados.

Sobre Esta Tarefa

- Assegure-se de que os módulos a seguir estejam em uma biblioteca do sistema autorizada no LNKST do sistema MVS ou de movê-los para uma biblioteca na concatenação LPALST.
 - DFHDTVC e DFHDTCV, porque todas as regiões que usam tabelas de dados compartilhados devem usar o mesmo nível de código SVC.
 - DFHMVRMS, o stub de saída RESMGR, porque os conjuntos de dados do CICS, JOBLIB/STEPLIB, estão indisponíveis no fim da memória.

Os módulos a seguir são colocados pela instalação do CICS na biblioteca de destino SDFHLINK, que, normalmente, é incluída na concatenação LNKST.

- Se SDFHLINK estiver na concatenação LNKST, será necessário emitir o comando do operador MODIFY LLA, REFRESH e esperar uma mensagem de confirmação, CSV210I LIBRARY LOOKASIDE REFRESHED, para que os módulos sejam disponibilizados.
- Se SDFHLINK não estiver na concatenação LNKST, será necessário copiar os módulos em uma biblioteca adequada que esteja incluída e emitir uma atualização LLA ou copiar os módulos em uma biblioteca na concatenação LPALST e emitir novamente o IPL no sistema MVS especificando CLPA.
- Se algum arquivo em alguma AOR tiver que usar compartilhamento, certifique-se de que o CICS esteja definido como um subsistema MVS.
- Defina a autorização de segurança para que as FORs possam agir como servidores de tabela de dados compartilhados e as AORs possam acessar arquivos pertencentes aos servidores, dependendo do nível de segurança necessário. Em uma única imagem do MVS:
 - Qualquer número de FORs pode agir como servidores de tabela de dados compartilhados
 - Uma única AOR pode usar qualquer número dessas FORs
 - Uma única FOR pode entregar qualquer número de AORs
 - Uma região pode agir como uma AOR para uma tabela de dados e como uma FOR para uma tabela de dados diferente.

- Se duas FORs tiverem que ter o mesmo APPLID, a qualquer momento, somente uma dessas FORs será usada como um servidor de tabela de dados compartilhados. No entanto, não há nada para evitar que uma FOR aja como um servidor de tabela de dados compartilhados e que outra FOR, com o mesmo APPLID, seja usada para solicitações de remessa de função. É necessário verificar se seus procedimentos operacionais não permitem isso, pois existe um risco de as solicitações da tabela de dados usando serviços de tabela de dados compartilhados não serem direcionadas para a mesma região que as solicitações que usam remessa de função.
- Defina esses arquivos na FOR que estão nas tabelas de dados como tabelas de dados mantidas pelo CICS ou tabelas de dados mantidas pelo usuário.
- Crie definições de arquivo remoto adicionais na AOR, se necessário. Não é necessário fazer nenhuma mudança nas definições de arquivo remoto existentes.
- Para qualquer AOR que deverá compartilhar tabelas de dados, especifique ISC=YES como um parâmetro de inicialização do sistema e defina os links MRO ou ISC para as FORs relevantes. Para conexões de interconectividade de Protocolo da Internet (IPIC), especifique o parâmetro de inicialização do sistema equivalente TCP/IP=YES e defina um link IPIC para a FOR relevante.
- Antes de usar tabelas de dados compartilhados, talvez seja necessário mudar algumas de suas instruções JCL, modificar seus procedimentos operacionais ou aumentar o valor do parâmetro de inicialização MAXUSER MVS. Para obter mais informações, consulte "Requisitos de JCL do MVS ao usar tabelas de dados compartilhados" na página 12.

Módulos de Carregamento

Estes módulos de carregamento devem ser instalados em sua região CICS para o uso de tabelas de dados compartilhados.

Tabela 2. Módulos de carregamento usados pelo suporte de tabela de dados compartilhados

Módulo de carregamento	Biblioteca de carregamento	Como foi carregado	Descrição
DFHDTINS	SDFHLOAD	Carregamento do CICS acima da linha de 16 MB	Inicialização
DFHDT SVC	SDFHLINK	MVS LOAD acima da linha de 16 MB da lista de links	Executa todas as funções que precisam de autorização de MVS
DFHDTFOR	SDFHAUTH	MVS LOAD acima da linha de 16 MB	Módulo da FOR da tabela de dados
DFHDTAM	SDFHAUTH	MVS LOAD no armazenamento do subconjunto 252 acima da linha de 16 MB	Gerenciador de acesso de tabela de dados. Inclui código que é executado em um modo de memória cruzada a partir de uma AOR
DFHDTAOR	SDFHAUTH	MVS LOAD acima da linha de 16 MB	Módulo da AOR da tabela de dados
DFHDT CV	SDFHLINK	MVS LOAD em um ECSA da lista de links	Validação de conexão (AOR)
DFHDTXS	SDFHAUTH	MVS LOAD em ECSA	Verificação de segurança de conexão (FOR)

Tabela 2. Módulos de carregamento usados pelo suporte de tabela de dados compartilhados (continuação)

Módulo de carregamento	Biblioteca de carregamento	Como foi carregado	Descrição
DFHMRMS	SDFHLINK	MVS LOAD acima da linha de 16 MB da lista de links	Código de interface EOT/EOM do gerenciador de recursos

Ocupação de armazenamento:

O tamanho total dos módulos que ocupam o armazenamento acima da linha de 16MB é de cerca de 41KB. Para módulos que estão no armazenamento ECSA, cerca de 1,5KB são necessários para cada FOR registrada e cerca de 0,5KB para cada AOR.

Os módulos são todos elegíveis para inclusão na área do pacote de links (LPA), mas somente DFHDTFOR, DFHDTAM, DFHDTAOR e, possivelmente, DFHDTCV são usados com frequência suficiente para serem levados em consideração.

Definição de recurso para tabelas de dados

Você define uma tabela de dados da mesma maneira que um arquivo CICS, exceto que também é necessário especificar o tipo de tabela de dados a ser usado e o número máximo de registros que podem ser mantidos na tabela de dados.

A definição de KSDS VSAM fornece o comprimento máximo do registro e o comprimento da chave.

É possível definir um arquivo como uma tabela de dados usando o comando CEDA DEFINE FILE, conforme descrito em “O comando **DEFINE FILE** define tabelas de dados” na página 23.

Além disso, para mudar ou verificar os atributos de tabela de dados de um arquivo existente, é possível usar:

- Comandos EXEC CICS SET FILE e INQUIRE FILE (consulte “Comandos EXEC CICS para tabelas de dados” na página 27)
- Comandos CEMT SET FILE e INQUIRE FILE (consulte “Comandos CEMT para tabelas de dados” na página 28)

Definição de recurso para tabelas de dados mantidas pelo CICS

O formato de registro de comprimento variável ou de comprimento fixo pode ser especificado para uma tabela de dados mantida pelo CICS.

O comprimento máximo de registro suportado pelo suporte de tabela de dados compartilhados é 32KB. Esse comprimento excede o suportado pelo gerenciamento de arquivo CICS, que impõe o limite real. Consulte o Os comprimentos das áreas transmitidas aos comandos do CICS. O número máximo de registros suportado é 16.777.215.

Apenas o cluster VSAM base pode ter uma tabela de dados mantida pelo CICS baseada nele. Solicitações de leitura por meio de caminhos do índice alternativo

não usam a tabela de dados, mas as mudanças no conjunto de dados de origem por meio de caminhos do índice alternativo são refletidas na tabela de dados.

Observe que o conjunto de dados de origem para uma tabela de dados mantida pelo CICS não pode ser aberto no modo de acesso RLS. Portanto, a definição de arquivo deve especificar RLSACCESS(NO), de forma que quaisquer outros arquivos devem estar associados ao mesmo conjuntos de dados de base.

Após um arquivo que é definido como tabela de dados mantida pelo CICS ser aberto, qualquer outro arquivo não UMT (seja definido como CMT ou não) que nomeia o mesmo conjunto de dados de origem em sua definição usa a mesma tabela de dados automaticamente. Se algum desses outros arquivos for definido como CMTs, mensagens DFHFC0937 serão emitidas para o console quando forem abertas. Essa não é uma situação de erro; os arquivos são abertos e usam a tabela de dados existente sempre que possível.

VSAM SHAREOPTION

Se o conjunto de dados de origem for alocado com DISP=SHR, há o risco de ele ser atualizado por uma região diferente da FOR. Se isso acontecesse, a tabela de dados não corresponderia mais ao conjunto de dados de origem. Para minimizar esse risco, a região cruzada do VSAM, SHAREOPTION, deve ser configurada para 1 ou 2.

- 1 significa que uma região pode ter acesso de atualização ao conjunto de dados ou várias regiões podem ter acesso somente leitura.
- 2 significa que uma região pode ter acesso de atualização ao conjunto de dados e, ao mesmo tempo, muitas regiões podem ter acesso somente leitura.

Independentemente da configuração de DISP, uma mensagem de aviso será emitida se a região cruzada SHAREOPTION for 3 ou 4, ou se for 2, mas a tabela de dados mantida pelo CICS tiver acesso somente leitura (o que significa que outra região pode estar apta para atualizar o conjunto de dados).

Integridade de Dados

Um arquivo que usa uma tabela de dados mantida pelo CICS pode ser definido como um recurso recuperável. O conjunto de dados de origem é recuperado de maneira normal após uma falha de transação ou do sistema.

- Após uma falha do sistema, a tabela de dados é recarregada do conjunto de dados de origem recuperado quando o arquivo é reaberto.
- Após uma falha de transação, as mudanças que forem feitas no conjunto de dados de origem pela restauração de transação dinâmica também serão feitas na tabela de dados.

O registro automático em diário é suportado (da mesma maneira que para qualquer outro arquivo) para operações de arquivo que acessam o conjunto de dados de origem. Operações de arquivo que não acessam o conjunto de dados de origem não são registradas em diário.

Definição de recurso para tabelas de dados mantidas pelo usuário

O formato do registro de comprimento variável deve ser especificado para uma tabela de dados mantida pelo usuário.

O comprimento máximo de registro suportado pelo suporte de tabela de dados compartilhados é 32KB. Esse comprimento excede aquele que é suportado pelo

gerenciamento de arquivo CICS, que impõe o limite real. Consulte o Os comprimentos das áreas transmitidas aos comandos do CICS. O número máximo de registros suportados é 16 777 215.

O conjunto de dados de origem para uma tabela de dados mantida pelo usuário pode ser aberto no modo de acesso RLS. Talvez você queira tornar um conjunto de dados no modo RLS a origem de uma tabela de dados mantida pelo usuário, caso você tenha outras definições de arquivo que acessem o conjunto de dados e o conjunto de dados seja atualizado por outras regiões CICS.

É possível carregar várias tabelas de dados mantidas pelo usuário do mesmo conjunto de dados de origem usando um comando separado ou uma macro para definir cada tabela de dados e fazer todas as definições se referirem ao conjunto de dados.

Embora uma tabela de dados deva ser carregada de um KSDS VSAM, um aplicativo pode então copiar registros em uma tabela de dados mantida pelo usuário de qualquer origem de dados que esteja acessível a partir do espaço de endereço do CICS. Pode ser um arquivo IMS ou DB2. O KSDS que é usado como o conjunto de dados de origem para a tabela de dados pode estar vazio; só é necessário definir o comprimento máximo do registro e a posição e o comprimento da chave.

Integridade de Dados

Uma tabela de dados mantida pelo usuário pode ser definida como um recurso recuperável. Mudanças na tabela de dados não são registradas no log do sistema, mas são mantidas internamente na memória do CICS. Assim, a tabela de dados pode ser recuperada após uma falha de transação (por restauração dinâmica), mas não após uma falha do sistema.

Isso porque o recurso Tabela de Dados Compartilhados do CICS gerencia sua própria recuperação e não usa os serviços do gerenciador de log ou gerenciador de recuperação. A exceção é quando são feitas mudanças em uma tabela de dados recuperável como parte de uma unidade de trabalho distribuída. Nesse caso, assim como com outros recursos recuperáveis, um registro do link é gravado no log do sistema como parte do processo two-phase commit. No entanto, as mudanças em si não são registradas no log do sistema.

Após uma falha do sistema, a tabela de dados é recarregada do conjunto de dados de origem quando o arquivo é reaberto. Lembre-se de que, no momento da falha, o conteúdo do conjunto de dados de origem e da tabela de dados não é o mesmo, a menos que você tenha assegurado que:

- nenhuma mudança foi feita em nenhum ou
- nenhuma mudança foi feita nos dois

O registro automático em diário é suportado somente para solicitações que acessam o conjunto de dados de origem durante o carregamento. Os registros que são acessados pelo processo de carregamento são registrados em diário antes da saída de usuário XDTRD, e os registros que são acessados devido às solicitações do aplicativo são registrados em diário após a saída de usuário XDTRD.

O comando DEFINE FILE define tabelas de dados

Você usa o comando **DEFINE FILE** para definir um arquivo como uma tabela de dados mantida pelo CICS ou tabela de dados mantida pelo usuário.

Detalhes completos de definições FILE são fornecidos em Recursos FILE. Apenas os atributos relacionados a tabelas de dados são descritos neste tópico.

TABLE({NO|CICS|USER|CF})

Especifique **TABLE(CICS)** para definir um arquivo como uma tabela de dados mantida pelo CICS.

Especifique **TABLE(USER)** para definir o arquivo como uma tabela de dados mantida pelo usuário.

Se você não especificar o parâmetro TABLE, ou especificar **TABLE(NO)** ou **TABLE(CF)**, o arquivo não será definido como uma tabela de dados compartilhados CICS.

MAXNUMRECS(NOLIMIT|number)

Especifica o número máximo de registros que podem estar contidos na tabela de dados no intervalo de 1 a 99999999. O padrão é que não existe limite no número máximo de registros.

FILE(name)

Especifica o nome do arquivo.

Para uma tabela de dados mantida pelo CICS, esse nome é usado para se referir à tabela de dados e ao conjunto de dados de origem, que são tratados como uma única entidade pelo CICS.

Para uma tabela de dados mantida pelo usuário, esse nome é usado para se referir somente à tabela de dados.

DSNAME(name)

Especifica o nome de KSDS VSAM a ser usado como o conjunto de dados de origem. Ele deve ser um conjunto de dados base, e não um caminho, ou um conjunto de dados de índice alternativo. Se houver um caminho ou índice alternativo associado ao conjunto de dados de origem, quaisquer atualizações para uma tabela de dados mantida pelo CICS, feitas por meio do arquivo, serão refletidas no conjunto de dados de origem e seus índices alternativos. Para uma tabela de dados mantida pelo usuário, as atualizações não são refletidas no conjunto de dados de origem ou seus índices alternativos. Após o carregamento ser concluído, uma tabela de dados mantida pelo usuário será totalmente independente de seu conjunto de dados de origem.

LSRPOOLID(number|1)

Este atributo está obsoleto, mas é suportado para oferecer compatibilidade com liberações anteriores do CICS.

LSRPOOLNUM(number|1|NONE)

Especifica o número do conjunto de recurso compartilhado local (LSR) VSAM que deve ser usado pela tabela de dados. Deve-se especificar um número de LSRPOOL no intervalo de 1 a 255. O valor padrão é 1, a menos que um valor tenha sido especificado para o atributo NSRGROUP, em cujo caso o valor padrão para LSRPOOLNUM será NONE.

OPENTIME({FIRSTREF|STARTUP})

Especifica quando o arquivo deve ser aberto, na primeira referência ou imediatamente após a inicialização, pela transação CSFU iniciada automaticamente. **OPENTIME(FIRSTREF)** é assumido como padrão.

Lembre-se de que a tabela de dados é carregada quando o arquivo é aberto, portanto, se você estiver usando a saída de usuário XDTRD, certifique-se de que a saída de usuário seja ativada antes de o arquivo ser aberto (consulte Ativando saídas de usuário para tabelas de dados).

RECORDFORMAT({V|F})

Especifica o formato dos registros no arquivo - ou **RECORDFORMAT(V)** para registros de comprimento variável ou **RECORDFORMAT(F)** para registros de comprimento fixo.

RECORDFORMAT(V) é assumido por padrão. Uma tabela de dados mantida pelo usuário deve ter registros de comprimento variável.

ADD(NO|YES), BROWSE(NO|YES), DELETE(NO|YES), READ(YES|NO) e UPDATE(NO|YES)

Especifica as operações de arquivo que podem ser solicitadas para a tabela de dados.

RECOVERY({NONE|BACKOUTONLY|ALL})

Especifica o tipo de suporte de recuperação necessário para a tabela de dados. O padrão é **RECOVERY(NONE)**.

Para uma tabela de dados mantida pelo usuário, somente a restauração de transação dinâmica é suportada pelo CICS, portanto, **RECOVERY(BACKOUTONLY)** e **RECOVERY(ALL)** têm o mesmo significado.

Para uma tabela de dados mantida pelo CICS, o parâmetro **RECOVERY** se aplica ao conjunto de dados de origem; ele deve estar consistente com qualquer outra definição de arquivo para o mesmo conjunto de dados.

Os atributos de recuperação de uma tabela de dados mantida pelo usuário são independentes de quaisquer atributos de recuperação que seu conjunto de dados de origem possa ter.

Ao definir uma tabela de dados mantida pelo usuário, você especifica seus atributos de recuperação na definição de arquivo especificando **RECOVERY(NONE)** se ele for irre recuperável ou **RECOVERY(BACKOUTONLY|ALL)** se ele for recuperável após uma falha de transação.

O conjunto de dados de origem para a tabela de dados mantida pelo usuário pode ser irre recuperável, recuperável somente para restauração (após falhas de transação e sistema) ou de recuperação avançada, independentemente do que você especificou para a tabela de dados mantida pelo usuário.

O conjunto de dados de origem pode adquirir seus atributos de recuperação de uma entre duas formas:

- Tendo os atributos de recuperação para o conjunto de dados definidos no catálogo ICF (isso é possível no CICS Transaction Server para z/OS, Versão 5 Release 4 para arquivos no modo RLS e não RLS).
- Usando outro nome de arquivo para acessar o conjunto de dados como um arquivo CICS ordinário, com os atributos de recuperação especificados na definição de arquivo (isso só é possível no CICS Transaction Server para z/OS, Versão 5 Release 4 para arquivos no modo não RLS).

Exemplo de uma definição de tabela de dados mantida pelo CICS

Este exemplo mostra a definição de uma tabela de dados mantida pelo CICS. Apenas os parâmetros relevantes são mostrados.

```

File      ==> APPLE
Group     ==> FRUIT
Description ==>
VSAM PARAMETERS
DSName    ==> CIC01.CICOWN.APPLES
Password  :      PASSWORD NOT SPECIFIED
RLSACCESS ==> NO      YES|NO
LSRPOOLId ==> 1      1-8 | None
LSRPOOLNum ==> 002    1-255 | None
READINTEG ==> UNCOMMITTED UNCOMMITTED|CONSISTENT|REPEATABLE
DSNSharing ==> Allreqs Allreqs | Modifyreqs
STRings   ==> 005     1 - 255
Nsrgroup  ==>
REMOTE ATTRIBUTES
REMOTESystem ==>
REMOTENAME ==>
REMOTE AND CFDATATABLE PARAMETERS
RECORDSize ==> 00080  1-32767
Keylength  ==> 006    1-255 (1-16 For CF Datatable)
INITIAL STATUS
STatus     ==> Enabled Enabled | Disabled | Unenabled
Opentime   ==> Startup Firstref | Startup
Disposition ==> Share Share | Old
BUFFERS
Databuffers ==> 00002  2 - 32767
Indexbuffers ==> 00001 1 - 32767
DATATABLE PARAMETERS
TABLE      ==> CICS    No | Cics | User | CF
Maxnumrecs ==> 1000000 Nolimit | 1-99999999
CFDATATABLE PARAMETERS
Cfdtpool   ==>
TABLEName  ==>
UPDATEModel ==> Locking Contention | Locking
LOad       ==> No     No | Yes
DATA FORMAT
RECORDFormat ==> F     V | F
OPERAÇÕES
Add        ==> Yes     No | Yes
Browse     ==> No      No | Yes
DElete     ==> Yes     No | Yes
REAd       ==> Yes     Yes | No
Update     ==> Yes     No | Yes
AUTO JOURNALING
JournAl    ==> No      No | 1 - 99
JNLRead    ==> None    None | Updateonly | Readonly | All
JNLSYNCRd  ==> No      No | Yes
JNLUpdate  ==> No      No | Yes
JNLAdd     ==> None    None | Before | After | All
JNLSYNCRw  ==> Yes     Yes | No
RECOVERY PARAMETERS
RECOvery   ==> All     None | Backoutonly | All
Fwdrecovlog ==> 10     No | 1-99
BACKuptype ==> STAtic  STAtic | DYNamic
SECURITY
RESsecnum  : 00       0-24 | Public

```

Exemplo de uma definição de tabela de dados mantida pelo usuário

Este exemplo mostra a definição de uma tabela de dados mantida pelo usuário. Apenas os parâmetros relevantes são mostrados.

```

File      ==> COURGETT
Group     ==> VEGS
Description ==>
VSAM PARAMETERS
DSName    ==> CIC02.CICOWN.COURGETT
Password  :      PASSWORD NOT SPECIFIED
RLSACCESS ==> NO      YES|NO
LSRPOOLId ==> 1      1-8 | None
LSRPOOLNum ==> 002    1-255 | None
READINTEG ==> UNCOMMITTED UNCOMMITTED|CONSISTENT|REPEATABLE
DSNSharing ==> Allreqs Allreqs | Modifyreqs
STRings   ==> 005    1 - 255
Nsrgroup  ==>
REMOTE ATTRIBUTES
REMOTESystem ==>
REMOTENAME ==>
REMOTE AND CFDATATABLE PARAMETERS
RECORDSize ==> 00080 1-32767
Keylength  ==> 006   1-255 (1-16 For CF Datatable)
INITIAL STATUS
STatus     ==> Enabled Enabled | Disabled | Unenabled
Opentime   ==> Firstref Firstref | Startup
Disposition ==> Share Share | Old
BUFFERS
Databuffers ==> 00002 2 - 32767
Indexbuffers ==> 00001 1 - 32767
DATATABLE PARAMETERS
TABLE      ==> User   No | Cics | User | CF
Maxnumrecs ==> 2000000 Nolimit | 1-99999999
CFDATATABLE PARAMETERS
Cfdtpool   ==>
TABLEName  ==>
UPDATEModel ==> Locking Contention | Locking
Load       ==> No     No | Yes
DATA FORMAT
RECORDFormat ==> V     V | F
OPERAÇÕES
Add         ==> Yes    No | Yes
Browse     ==> Yes    No | Yes
DElete     ==> No     No | Yes
REAd       ==> Yes    Yes | No
Update     ==> Yes    No | Yes
AUTO JOURNALING
JJournal   ==> No     No | 1 - 99
JNLRead    ==> None   None | Updateonly | Readonly | All
JNLSYNRead ==> No     No | Yes
JNLUpdate  ==> No     No | Yes
JNLAdd     ==> None   None | Before | After | All
JNLSYNWrite ==> Yes    Yes | No
RECOVERY PARAMETERS
RECOvery   ==> Backoutonly None | Backoutonly | All
Fwdrecovlog ==> No     No | 1-99
BACKuptype ==> STAtic  STAtic | DYNamic
SECURITY
RESsecnum  : 00      0-24 | Public

```

Comandos EXEC CICS para tabelas de dados

É possível usar o comando **EXEC CICS SET FILE** para mudar a definição de um arquivo existente e o comando **EXEC CICS INQUIRE FILE** para verificar a definição de um arquivo existente.

Para obter informações de programação, incluindo detalhes sobre como usar esses comandos e os parâmetros descritos aqui, consulte SET FILE. Os parâmetros que são relevantes para as tabelas de dados são descritos abaixo.

Esta seção contém Interface de Programação de Uso Geral e Informações de Orientação Associadas.

SET FILE

Os parâmetros a seguir são relevantes para tabelas de dados; é possível usá-los somente quando o arquivo estiver fechado e desativado.

É possível especificar um atributo de tabela de dados de um arquivo em uma área de dados de valores de CICS (cvda):

TABLE(cvda)

Especifique um valor de CVDA de **CICSTABLE** para definir o arquivo como uma tabela de dados mantida pelo CICS.

Especifique um valor de CVDA de **USERTABLE** para definir o arquivo como uma tabela de dados mantida pelo usuário.

Especifique um valor de CVDA de **NOTTABLE** para indicar que o arquivo não é uma tabela de dados.

Nota: Também é possível especificar **CFTABLE** para indicar uma tabela de dados de recurso de acoplamento.

MAXNUMRECS(value)

Especifica o número máximo de registros que podem estar contidos na tabela de dados no intervalo de 1 a 99999999. O valor de zero significa sem limite.

INQUIRE FILE

Os parâmetros a seguir são relevantes para tabelas de dados.

É possível solicitar que cada atributo de tabela de dados de um arquivo seja retornado em uma área de dados de valores de CICS (cvda) especificando:

TABLE(cvda)

Se o valor **CICSTABLE** for retornado, o arquivo foi definido como uma tabela de dados mantida pelo CICS.

Se o valor **USERTABLE** for retornado, o arquivo foi definido como uma tabela de dados mantida pelo usuário.

Se o valor **CFTABLE** for retornado, o arquivo foi definido como uma tabela de dados de recurso de acoplamento.

Se o valor **NOTTABLE** for retornado, o arquivo não está definido atualmente como uma tabela de dados.

Se o valor **NOTAPPLIC** for retornado, a opção não será aplicável porque o arquivo é um arquivo remoto.

MAXNUMRECS(cvda)

O valor retornado indica o número máximo de registros que podem estar contidos na tabela de dados. O valor de zero significa sem limite.

Comandos CEMT para tabelas de dados

É possível usar o comando CEMT SET FILE para mudar a definição de um arquivo existente e o comando CEMT INQUIRE FILE para verificar a definição de um arquivo existente.

Os detalhes completos sobre como usar esses comandos, incluindo os parâmetros descritos aqui, são fornecidos em INQUIRE FILE. Os parâmetros que são relevantes para as tabelas de dados são descritos abaixo.

SET FILE

Os parâmetros a seguir são relevantes para tabelas de dados; é possível usá-los somente quando o arquivo estiver fechado e desativado.

{CICSTABLE|USERTABLE|CFTABLE|NOTTABLE}

especificar **CICSTABLE** para definir o arquivo como uma tabela de dados mantida pelo CICS

especificar **USERTABLE** para definir o arquivo como uma tabela de dados mantida pelo usuário

Nota: Também é possível especificar **CFTABLE** para indicar uma tabela de dados de recurso de acoplamento.

especificar **NOTTABLE** para indicar que o arquivo não é uma tabela de dados

MAXNUMRECS(value)

Especifique o número máximo de registros que podem estar contidos na tabela de dados, no intervalo de 1 a 99999999. O valor de zero significa sem limite.

INQUIRE FILE

Os parâmetros a seguir são relevantes para tabelas de dados.

Tabela Dados

Se o valor **CICSTABLE** for retornado, o arquivo foi definido como uma tabela de dados mantida pelo CICS.

Se o valor **USERTABLE** for retornado, o arquivo foi definido como uma tabela de dados mantida pelo usuário.

Se o valor **CFTABLE** for retornado, o arquivo foi definido como uma tabela de dados de recurso de acoplamento.

Se o valor **NOTTABLE** for retornado, o arquivo não está definido atualmente como uma tabela de dados.

MAXNUMRECS(value)

O valor retornado indica o número máximo de registros que podem estar contidos na tabela de dados. O valor de zero significa que não há limite máximo.

Capítulo 3. Desenvolvimento para acesso a tabelas de dados

É possível acessar uma tabela de dados com os mesmos comandos de controle de arquivos EXEC CICS que os usados com qualquer arquivo CICS normal. Esses comandos podem ser completamente usados com uma tabela de dados mantida pelo CICS e, com algumas restrições, com uma tabela de dados mantida pelo usuário.

Para obter informações gerais sobre como usar esses comandos, consulte Usando a função do link de programa distribuído. Para obter informações de programação, consulte Resumo de Comandos do CICS.

Programação de aplicativos para uma tabela de dados mantida pelo CICS

O CICS trata uma tabela de dados mantida pelo CICS e seu conjunto de dados de origem como uma única entidade. Após a tabela de dados ter sido carregada, automaticamente, o CICS mantém o conteúdo da tabela de dados e do conjunto de dados de origem consistente; quaisquer mudanças feitas por um aplicativo no arquivo são refletidas em ambos. Em quase todas as situações, o uso de uma tabela de dados é transparente para o programador de aplicativos.

Todas as opções e comandos de controle de arquivos podem ser usados para uma tabela de dados mantida pelo CICS. Alguns comandos são executados por meio do acesso somente à tabela de dados (usando serviços de memória cruzada para arquivos compartilhados), alguns por meio do acesso somente ao conjunto de dados de origem (usando remessa de função para arquivos compartilhados) e alguns por meio do acesso a ambos.

Geralmente, os comandos a seguir acessam somente tabelas de dados:

- Comandos READ sem as opções UPDATE ou RBA
- Comandos STARTBR, RESETBR, READNEXT e READPREV sem a opção RBA
- Comando ENDBR (a menos que a sequência de procura tenha acessado o conjunto de dados de origem)

Os comandos a seguir acessam somente o conjunto de dados de origem:

- Comandos READ com as opções UPDATE ou RBA
- Comandos STARTBR, RESETBR, READNEXT e READPREV com a opção RBA
- Comando ENDBR para uma sequência de procura que acessou o conjunto de dados de origem

Os comandos a seguir podem acessar a tabela de dados e o conjunto de dados de origem:

- Comandos READ e browse (que geralmente acessariam somente a tabela de dados) que encontram uma diferença na sequência-chave de registros na tabela de dados. Essa diferença pode indicar que um ou mais registros estão ausentes na tabela de dados porque:
 - registros foram suprimidos por uma saída de usuário
 - o número máximo de registros foi atingido
 - há armazenamento virtual insuficiente disponível para a tabela de dados
 - ocorreu algum evento anormal

- Comandos READ, READNEXT e READPREV para registros que estão atualmente sendo processados por um comando WRITE, REWRITE ou DELETE. Esses comandos primeiro precisam acessar a tabela de dados para determinar se essa situação existe.
- Comandos WRITE, REWRITE e DELETE. Esses comandos são sempre executados na FOR, onde primeiro atualizam o conjunto de dados de origem. Se forem bem-sucedidos, uma tentativa de mudança correspondente na tabela de dados será feita usando serviços da tabela de dados compartilhados local na FOR. No caso de um comando WRITE, a adição do registro na tabela de dados pode ser rejeitada pela saída de usuário XDTAD ou pode falhar porque a tabela de dados está cheia ou porque há armazenamento virtual insuficiente disponível.

Leituras genéricas para uma tabela de dados mantida pelo CICS

Para aplicativos que executam leituras genéricas, usando a opção GENERIC no comando READ, existe uma diferença no comportamento de uma tabela de dados mantida pelo CICS em comparação com um arquivo VSAM. Talvez seja necessário modificar esses aplicativos ao converter um arquivo VSAM em uma tabela de dados mantida pelo CICS.

Para uma leitura genérica de um arquivo VSAM, se o CICS retornar uma condição NOTFND porque o registro não foi localizado na tabela, as áreas INTO() e RIDFLD() do comando READ serão mantidas inalteradas. No entanto, para uma leitura genérica de uma tabela de dados mantida pelo CICS, se o CICS retornar uma condição NOTFND, ele limpará as áreas INTO() e RIDFLD() para assegurar que um registro incorreto não seja retornado.

Esse comportamento otimiza o desempenho para tabelas de dados mantidas pelo CICS, mas isso significa que os aplicativos não podem mais depender dos valores originais nas áreas INTO() e RIDFLD() sendo retornadas. Se você tiver algum aplicativo que executa leituras genéricas, modifique-os conforme necessário para tomar a ação apropriada se uma condição NOTFND for retornada e as áreas INTO() e RIDFLD() forem limpas.

Usando uma tabela de dados mantida pelo CICS durante o carregamento

É possível usar uma tabela de dados mantida pelo CICS enquanto ela é carregada. Se o registro necessário já tiver sido carregado, o processamento da solicitação será tratado de forma normal.

Se o registro ainda não tiver sido carregado, o seguinte é feito:

- Para um comando READ, o registro é lido do conjunto de dados de origem e retornado para o programa de aplicativo. Ele é incluído na tabela de dados quando a sequência de carregamento normal o atinge.
- Para um comando WRITE, o registro é incluído no conjunto de dados de origem e na tabela de dados (se ele não for suprimido pela saída de usuário XDTAD).
- Para um comando REWRITE ou DELETE, a mudança é aplicada no conjunto de dados de origem. Essa mudança é então refletida na tabela de dados pelo processo de carregamento normal.

Programação de aplicativos para uma tabela de dados mantida pelo usuário

O CICS trata uma tabela de dados mantida pelo usuário e seu conjunto de dados de origem como entidades separadas. Quando o carregamento é concluído, todos os comandos de controle de arquivos que acessam o nome do arquivo são executados somente na tabela de dados.

Se uma solicitação não puder ser satisfeita a partir de uma tabela de dados mantida pelo usuário, o CICS não acessará o conjunto de dados de origem (como faria para uma tabela de dados mantida pelo CICS). O CICS retorna uma resposta de condição de exceção.

É possível usar as saídas de usuário nos serviços de tabela de dados para colocar somente os registros que você precisa acessar na tabela de dados; não há possibilidade de o conjunto de dados de origem ser acessado para aqueles que você não conseguir carregar. Também é possível usar a saída de usuário XDTRD para modificar cada registro (selecionando, por exemplo, somente um subconjunto de seus campos) quando eles forem carregados.

Registros que foram recuperados no conjunto de dados de origem quando a tabela de dados foi aberta podem estar ausentes da tabela de dados, pois não foram copiados durante o carregamento. Talvez seja devido à supressão da saída de usuário XDTRD ou a um evento anormal, como a tabela de dados cheia.

Algumas solicitações de programação de aplicativos não são suportadas para uma tabela de dados mantida pelo usuário. Por exemplo, solicitações de leitura que usam a opção UPDATE com uma chave imprecisa não são suportadas. Além disso, algumas condições de exceção são exclusivas para tabelas de dados mantidas pelo usuário. Talvez seja necessário mudar aplicativos existentes para obedecer às restrições nas quais comandos e opções podem ser usados ou manipular as condições de exceção que o CICS retornar.

Os comandos a seguir não são suportados; eles retornam a condição INVREQ e um valor de 44 no campo EIBRESP2:

- Comandos com a opção RBA
- Comandos WRITE com a opção MASSINSERT

Os comandos a seguir são suportados (usando serviços de memória cruzada para acesso remoto):

- Comandos READ sem a opção RBA ou opção UPDATE. Se o registro não existir na tabela de dados, a condição NOTFND será retornada.
- Comandos STARTBR, RESETBR, READNEXT e READPREV sem a opção RBA.
- Comandos ENDBR.

Os comandos a seguir são suportados (usando remessa de função para solicitações remotas):

- Comandos WRITE sem as opções RBA ou MASSINSERT. O registro é incluído na tabela de dados (se não for suprimido pela saída de usuário XDTAD).

A condição NOSPAC é retornada nas seguintes situações:

- Não há armazenamento de espaço para dados suficiente para incluir o registro na tabela de dados.

- A tabela de dados já contém o número máximo de registros especificado na definição de arquivo.

O CICS emite a mensagem DFHFC0432 se for feita uma tentativa de gravação, mas houver espaço insuficiente disponível.

A condição SUPPRESSED será retornada se a saída de usuário XDTAD suprimir a adição do registro na tabela de dados.

- Comandos REWRITE sem a opção RBA. O registro é atualizado na tabela de dados. A condição NOSPAC será retornada se houver armazenamento virtual insuficiente para o registro atualizado. O CICS emite a mensagem DFHFC0432 se houver espaço insuficiente disponível.
- Comandos DELETE sem a opção RBA. O registro é excluído da tabela de dados. A condição NOTFND será retornada se o registro não existir na tabela de dados. A condição NOSPAC será retornada se a tabela de dados for recuperável e se houver armazenamento virtual insuficiente para as informações que o CICS gravar sobre o registro excluído.

Usando uma tabela de dados mantida pelo usuário durante o carregamento

Uma tabela de dados mantida pelo usuário pode ser acessada somente pela FOR durante o carregamento. Todas as solicitações remotas têm remessa de função para a FOR, que as processa da mesma maneira que para uma solicitação local.

Enquanto uma tabela de dados mantida pelo usuário está sendo carregada, é possível usar somente solicitações de leitura de não atualização com chaves precisas. Se o registro já tiver sido carregado, o processamento da solicitação será tratado de forma normal. Se o registro ainda não tiver sido carregado, ele será lido no conjunto de dados de origem e enviado para a saída de usuário XDTRD (se ativado):

- Se não for suprimido pela XDTRD, o registro será incluído na tabela de dados e retornado para o programa de aplicativo.
- Se ele for suprimido pela XDTRD, a condição NOTFND será retornada.

A condição LOADING é retornada para outras solicitações que seriam válidas após o carregamento ser concluído.

Uso de serviços de memória cruzada para tabelas de dados compartilhados

Serviços de memória cruzada são usados para satisfazer um comando de programação de aplicativo quando todas as condições listadas aqui tiverem sido atendidas.

- CICS deve recuperar o SYSID do sistema de destino da definição do recurso do arquivo na AOR. Essa condição é atendida quando o comando de programação de aplicativos especifica nenhum SYSID explícito ou especifica um SYSID igual à própria AOR e o SYSID fornecido na definição do recurso do arquivo é igual à FOR.

Em uma sequência de procura única, um aplicativo não deve mudar entre a especificação de um SYSID explícito e a não especificação de um, já que, provavelmente, isso levaria a resultados imprevisíveis.

- O sistema de serviço efetuou logon; ou seja, ele se registrou como um proprietário de tabela de dados compartilhados.

- O sistema solicitante se conectou ao servidor para os arquivos especificados no comando de programação de aplicativo.
- O arquivo suporta a função solicitada.

Nota: A remessa de função de uma solicitação pode resultar em “daisy chain”; ou seja, a solicitação passa por um ou mais nós intermediários do CICS entre a região emitindo a solicitação (uma AOR) e a região que possui o recurso (a FOR). Nesses casos, o uso de serviços de memória cruzada de tabelas de dados compartilhados é limitado ao link final (do último sistema intermediário para a FOR).

Conexão

Comandos não podem usar serviços de memória cruzada até que a conexão seja estabelecida entre a AOR e a tabela de dados remotos.

Além disso, se uma sequência de procura for iniciada antes de a conexão ser estabelecida, todas as solicitações subsequentes na sequência usarão serviços de remessa de função. É provável que isso ocorra se a conexão não puder ser estabelecida no comando STARTBR porque a tabela de dados não está aberta e o comando fará a tabela de dados ser aberta implicitamente. A conexão então é estabelecida na próxima nova solicitação para a tabela de dados, mas a sequência de procura original continua usando serviços de remessa de função.

Desconexão

Quando uma conexão tiver sido estabelecida, ela permanece em vigor até que a AOR exclua sua definição de arquivo remoto ou que a FOR feche ou desative o arquivo.

Os efeitos do fechamento e ou da desativação são os seguintes:

- Se a FOR fechar o arquivo (com ou sem a opção FORCE), a desconexão será planejada na próxima solicitação de não atualização emitida para o arquivo (ou seja, a próxima solicitação de tentativa de uso de serviços de memória cruzada para acessar a tabela de dados).

A desconexão acontece assim que todas as sequências de procura pendentes (se houver) com relação ao arquivo forem finalizadas. Cada sequência de procura é finalizada na próxima solicitação de procura (e a transação é encerrada de forma anormal com o código AFCH, a menos que a solicitação seja um comando ENDBR) ou quando a transação for finalizada.

Após a desconexão ser planejada, todas as solicitações (exceto quaisquer solicitações de procura pendentes, conforme descrito acima) terão uma remessa de função até que uma conexão seja restabelecida.

- Se a FOR desativar o arquivo sem a opção FORCE, a desconexão será planejada no próximo comando READ ou STARTBR de não atualização emitido para o arquivo, a menos que a FOR reative o arquivo antes.

Se planejada, a desconexão acontece assim que todas as sequências de procura pendentes (se houver) com relação ao arquivo forem finalizadas. Tais sequências de procura continuam normalmente; elas não são afetadas pela desativação, a menos que uma procura do conjunto de dados de origem seja iniciada na FOR para satisfazer uma solicitação na sequência de procura.

- Se a FOR desativar o arquivo com a opção FORCE, o efeito será o mesmo de quando um arquivo é fechado, exceto que se a FOR reativar o arquivo antes de a AOR emitir a próxima solicitação de não atualização para o arquivo, a desativação não será observada pela AOR e uma desconexão não será planejada.

Diferenças entre remessa de função e serviços de memória cruzada

Existem inúmeras diferenças entre a maneira como as solicitações são manipuladas, dependendo se você usar remessa de função ou serviços de memória cruzada para acessar a tabela de dados.

Fechando uma tabela de dados

Quando a remessa de função é usada para uma sequência de procuras de um arquivo remoto, o arquivo não pode ser fechado (exceto usando a opção FORCE) até que a sequência de procura termine.

Quando serviços de memória cruzada são usados, é possível que o arquivo seja fechado durante a sequência de procura. Nesse caso, a transação é finalizada com o código de encerramento anormal AFCH na próxima solicitação para esse arquivo. Se seus aplicativos ou procedimentos operacionais dependerem do quiesce da atividade de procura durante o fechamento de um arquivo ou no encerramento normal de uma FOR, será necessário revisá-los antes de usar uma tabela de dados compartilhados para o arquivo.

Desativando uma tabela de dados

Quando a remessa de função é usada para uma sequência de procura de um arquivo remoto, após ser iniciada, a sequência de procura pode continuar normalmente, mesmo se o arquivo for desativado (a menos que a opção FORCE seja usada).

Quando serviços de memória cruzada são usados, o efeito é o mesmo, a menos que, durante a sequência de procura, uma remessa de função seja necessária para um comando STARTBR para a FOR. Isso pode acontecer se, por exemplo, uma diferença em uma tabela de dados mantida pelo CICS tornar necessária a procura de um conjunto de dados de origem VSAM para recuperar registros. O comando STARTBR fornecido por função falhará se o arquivo for desativado por uma solicitação que foi emitida pela FOR após a sequência de procura ser iniciada na AOR. Nesse caso, a sequência de procura não poderá continuar normalmente, portanto, a transação na AOR será encerrada de forma anormal com o código AFCH.

Se a opção FORCE for usada com a solicitação de desativação, todas as solicitações de procura de remessa de função serão sempre finalizadas. Se o arquivo for reativado, é possível que solicitações de procura usando serviços de memória cruzada continuem sem serem afetadas. (Para obter informações sobre FORCE, consulte “Desconexão” na página 35).

Saídas de usuários

Para solicitações com remessa de função, as saídas de usuário da interface exec XEIN e XEIOU, e as saídas de usuário de controle de arquivos XFCREQ e XFCREQC, são chamadas na AOR e na FOR.

Para solicitações de memória cruzada, essas saídas de usuário são chamadas somente na AOR.

Verificação de segurança

Para solicitações com remessa de função, a verificação de segurança na FOR é chamada para a primeira solicitação que se refere a um determinado arquivo em cada unidade de trabalho. Assim, as verificações de segurança em nível de transação podem ser executadas na FOR.

Para solicitações de memória cruzada, a verificação de segurança é chamada somente no momento da CONEXÃO. Assim, essas verificações de segurança em nível de transação não podem ser executadas na FOR.

Falha na solicitação de leitura

Se uma solicitação de leitura usando remessa de função falhar, a área de entrada permanecerá inalterada.

Se uma solicitação de leitura usando serviços de memória cruzada falhar, há uma chance de a área de entrada ser alterada, embora nenhum registro tenha sido recuperado. Portanto, você não deve crer que a área de entrada permanecerá inalterada, embora você possa assegurar que a chave não terá sido alterada.

Bloco de interface EXEC

Você pode observar que solicitações de leitura usando serviços de memória cruzada retornam um valor no campo EIBRESP2. No entanto, as solicitações de remessa de função não; portanto, seus aplicativos não devem depender de o campo ser configurado por solicitações de leitura.

Comprimento de chave

Para solicitações com remessa de função, deve-se especificar o comprimento da chave correto na definição de arquivo remoto na AOR ou explicitamente na solicitação de arquivo (para a correspondência do comprimento da chave na definição de VSAM na FOR). Se não, a condição INVREQ será retornada para qualquer solicitação que acesse o arquivo. Isso se aplica a qualquer arquivo, e não somente aquele definido como tabela de dados.

Para solicitações de memória cruzada, o comprimento da chave na AOR não é usado; as solicitações podem ser concluídas com sucesso, mesmo se o comprimento da chave não for especificado na AOR ou se o comprimento da chave especificado na AOR não corresponder ao da FOR. No entanto, seus aplicativos não devem depender disso, pois algumas solicitações podem ter remessa de função.

Diferenças entre serviços de tabela de dados compartilhados e VSAM

Como os serviços de tabela de dados compartilhados substituem o VSAM para várias solicitações de tabela de dados, há diferenças na maneira como certas solicitações são implementadas.

Leitura durante atualização (transações diferentes)

No caso de um comando READ para um registro de tabela de dados, após um READ UPDATE emitido para esse registro por *outra* transação e antes da solicitação de atualização associada, quando serviços da tabela de dados compartilhados são usados, o comando READ é processado imediatamente.

Quando o VSAM é usado, o comando READ aguarda até que a solicitação de atualização seja concluída.

Leitura durante atualização (mesma transação)

No caso de um comando READ para um registro de tabela de dados, após um READ UPDATE emitido para esse registro pela *mesma* transação e antes da solicitação de atualização associada, quando serviços da tabela de dados compartilhados são usados, o comando READ é processado imediatamente.

Quando o VSAM é usado, a transação incorre uma finalização anormal de tarefa de conflito AFCG.

Exclusão durante procura

Quando serviços de tabela de dados compartilhados são usados para um comando STARTBR ou RESETBR para um registro de tabela de dados, é possível que o registro seja excluído antes de o comando EADNEXT ou READPREV associado ser emitido. Quando o VSAM é usado, o registro não pode ser excluído antes de o comando READNEXT ou READPREV associado ser emitido.

Além disso, quando serviços de tabela de dados compartilhados são usados, se um comando STARTBR ou RESETBR for emitido com uma chave diferente da chave especial 'last record', X'FF...', e o registro selecionado for excluído antes do comando READNEXT, o comando READNEXT lerá o registro sucessor.

Se não houver nenhum registro sucessor, a condição ENDFILE será retornada. Se a opção EQUAL foi usada em STARTBR ou RESETBR, a chave do registro que é lida pode não corresponder à chave especificada.

Se um comando STARTBR ou RESETBR for emitido com a chave especial 'last record', e o registro selecionado for excluído antes do comando READPREV, o comando READPREV lerá o registro precedente ou retornará a condição ENDFILE se não houver nenhuma.

Gravação durante procura

Quando serviços de tabela de dados compartilhados forem usados, se uma procura for lida no final de um arquivo, levantando a condição ENDFILE, e um novo registro for inserido além do final do arquivo, um READNEXT subsequente poderá ler o novo registro.

Quando o VSAM é usado, o READNEXT subsequente pode não conseguir localizar o novo registro, mas sim relatar a condição ENDFILE novamente.

Exclusão durante atualização (mesma transação)

Quando serviços da tabela de dados compartilhados são usados para um comando DELETE que especifica RIDFLD para um registro de tabela de dados após um READ UPDATE ser emitido para esse registro pela mesma transação e antes da solicitação de atualização associada, o comando DELETE é processado com sucesso e a solicitação de atualização associada recebe uma condição NOTFND.

Capítulo 4. Customizando Tabelas de Dados Usando as Saídas de Usuário

Três pontos de saída do usuário global estão incluídos nos serviços da tabela de dados. É possível fornecer um ou mais programas de linguagem assembler para serem executados em cada um desses pontos para ampliar ou modificar a função fornecida pelo CICS.

Nota: *Esta seção contém Interface de Programação Sensível ao Produto e Informações de Orientação Associadas.*

O CICS fornece programas de saída de usuário de amostra na biblioteca SDFHSAMP para as saídas de usuário globais XDTRD, XDTAD e XDTLC. Esses programas, que são reproduzidos nesta documentação, descrevem com amostras de codificação e sequências de definição de dados as convenções usadas em programas de saída de usuário que são usadas com tabelas de dados compartilhados. Essas amostras são somente uma orientação geral e não servem para definir uma interface de programação.

Informações de programação sobre saídas de usuário globais e como usá-las são fornecidas em Programas de Saída de Usuário Global.

Nota: As saídas de usuário da interface EXEC XEIIN e XEIOUT, e as saídas de usuário de controle de arquivos XFCREQ e XFCREQC, não são iniciadas na região voltada para arquivos, caso uma solicitação para acessar uma tabela de dados seja satisfeita por serviços de memória cruzada.

Comunicação entre o CICS e programas de saída de tabela de dados compartilhados

Uma lista de parâmetros é usada para transmitir informações entre o CICS e os programas de saída de tabela de dados.

Na biblioteca CICSTS54.CICS.SDFHMAC, o CICS fornece um copybook denominado DFHXDTDS que contém um DSECT para definir esta lista de parâmetros. Inclua uma instrução COPY DFHXDTDS em cada um de seus programas de saída. O DSECT é mostrado em Figura 6 na página 40.

Os nomes de campo usados neste DSECT são referenciados nas descrições de saída de usuário que seguem a figura.

```

*****
*
*   Lista de Parâmetros da Tabela de Dados para as Saídas de Usuário XDTRD, XDTAD e XDTLC.  *
*
*   Alguns dos parâmetros são usados somente por uma ou duas das saídas.  *
*   Isso é indicado nos comentários para esses parâmetros.  *
*   Os comentários também indicam se o campo é usado para entrada  *
*   (In), saída (Out) ou ambos (In/Out).  *
*
*   Essa definição pode ser usada por programas de saída em execução  *
*   em regiões CICS que estão em um nível para suportar tabelas de  *
*   dados do recurso de acoplamento (CFDT), contanto que a sinalização UEPDTCFT seja usada para testar *
*   se a saída foi chamada de dentro do suporte das tabelas de dados do *
*   recurso de acoplamento e se os parâmetros que são específicos para o  *
*   suporte de CFDT são usados somente quando configurados.  O suporte de CFDT *
*   só estará disponível para programas de saída em execução em regiões CICS no  *
*   nível de CICS Transaction Server versão 1 liberação 3 ou superior.  *
*
*   Essa definição pode ser usada por programas de saída em execução em regiões CICS  *
*   que estão em um nível para suporte tabelas de dados compartilhados (SDT) *
*   ou que têm suporte de SDT instalado, contanto que a sinalização UEPDTSMT  *
*   seja usada para testar se a saída foi chamada de dentro do  *
*   suporte de tabelas de dados compartilhados e se os parâmetros  *
*   específicos do suporte de SDT são usados somente quando configurados.  O suporte de SDT  *
*   só estará disponível para regiões CICS em execução no  *
*   nível CICS/ESA versão 4 liberação 1 ou superior (ou em execução no  *
*   CICS/ESA versão 3 liberação 3, caso o recurso Tabelas de Dados Compartilhados  *
*   esteja instalado).  *
*
*   Essa definição também pode ser usada por programas de saída em execução em  *
*   regiões CICS que não estão em um nível para suportar SDT ou  *
*   CFDT, indicado pelas sinalizações UEPDTCFT e UEPDTSMT  *
*   desativada. Nesse caso, somente os parâmetros relacionados ao  *
*   suporte de tabelas de dados básicos podem ser usados.  O suporte para  *
*   tabelas de dados básicos só estará disponível para regiões CICS em execução em um  *
*   dos níveis a seguir:  *
*   - CICS/MVS versão 2 (mais tabelas de dados SPE em algumas liberações)  *
*   - CICS/ESA versão 3 liberações 1 ou 2  *
*   - CICS/ESA versão 3 liberação 3, se o recurso SDT NÃO estiver instalado  *
*
*   O uso com cautela dessas sinalizações, e dos parâmetros relacionados  *
*   aos vários tipos de suporte de tabelas de dados, deve permitir que  *
*   o mesmo programa de saída de usuário seja usado por mais de um tipo  *
*   de tabela de dados.  *
*
*****
DT_UE_PLIST_DSECT DSECT ,
DT_UE_PLIST      DS 0XL84          Data Table User Exits          X
                                   Lista de Parâmetro
UEPDTCFT         DS CL8           Data table name (In)
UEPDTCFLG        DS 0CL1          Flags (In):

```

Figura 6. Lista de parâmetros de saída de usuário da tabela de dados


```

*-----*
*      As sinalizações UEPDTSMT e UEPDTCFT indicam se a      *
*      saída foi chamada para suporte de tabelas de dados compartilhados ou      *
*      de tabelas de dados do recurso de acoplamento. Se nenhum estiver      *
*      configurado, a saída foi chamada sob o suporte de tabelas de dados      *
*      básicos, que pré-datou tabelas de dados compartilhados.      *
*
*      As sinalizações UEPDTCMT e UEPDTUMT estão disponíveis somente para      *
*      saídas que foram chamadas pelo suporte de tabelas de dados      *
*      compartilhados. Elas distinguem os dois tipos de tabelas de      *
*      dados compartilhados. Observe que em liberações anteriores ao      *
*      CICS Transaction Server 1.3, a sinalização UEPDTUMT NÃO está      *
*      disponível; nessas liberações, uma tabela de dados mantida      *
*      pelo usuário é indicada pela sinalização UEPDTCMT desativada.      *
*
*      Se a saída foi chamada pelo suporte CFDT, a      *
*      tabela de dados só pode ser uma tabela de dados de recurso de acoplamento,      *
*      portanto, não há sinalizações extra para identificar o tipo de      *
*      tabela de dados quando a saída foi chamada pelo suporte das      *
*      tabelas de dados do recurso de acoplamento.      *
*
*      A sinalização UEPDPTOPT está disponível para saídas que      *
*      foram chamadas pelo suporte de tabelas de dados compartilhados ou      *
*      pelo suporte de tabelas de dados de recurso de acoplamento (mas não para saídas      *
*      que foram chamadas pelo suporte de tabelas de dados básicos).      *
*      Portanto, essa sinalização não está disponível em liberações      *
*      anteriores à CICS/ESA 3.3 (mais suporte de SDT).      *
*-----*

```

	DS BL1	
UEPDTSMT	EQU X'80'	Exit invoked by SDT support
UEPDTCMT	EQU X'40'	Table is CICS-maintained
UEPDPTOPT	EQU X'20'	Exit invoked by table loader, X
		so optimization of the load by X
		skipping may be requested X
		(flag is for XDTRD only)
UEPDTCFT	EQU X'10'	Exit invoked by CFDT support X
UEPDUMT	EQU X'08'	Table is user-maintained
*	EQU X'07'	Reserved

Figura 7. Continuação da lista de parâmetros de saída de usuário da tabela de dados

* Os campos a seguir estão disponíveis para saídas que *			
* foram chamadas por todos os tipos de suporte de tabela de dados. *			
* Nem todos os campos estão disponíveis em todos os pontos de saída. *			

UEPDTORC	DS AL1	Data table load return code - X	
		XDTLC only, values below (In)	
	DS BL2	Reserved	
UEPDTRA	DS A	Data record address - XDTRD	X
		and XDTAD only (In)	
UEPDTRBL	DS F	Data buffer length - XDTRD and X	
		XDTAD only (In)	
UEPDTRL	DS F	Data table record length -	X
		XDTRD and XDTAD only, XDTRD	X
		can return new length in here	X
		if it amends record (only	X
		allowed for UMT or CFDT)	X
		(In/Out)	
UEPDTKA	DS A	Key address - XDTRD and XDTAD	X
		only (In)	
UEPDTKL	DS F	Key length - XDTRD and XDTAD	X
		only (In)	

Figura 8. Continuação da lista de parâmetros de saída de usuário da tabela de dados

* Os campos a seguir estão disponíveis para saídas que *			
* foram chamadas pelo suporte de tabelas de dados compartilhados *			
* ou pelo suporte de tabelas de dados do recurso de acoplamento. *			
* Nem todos os campos estão disponíveis em todos os pontos de saída. *			

UEPD DSL	DS F	Length of data set name (In)	
UEPD TSN	DS CL44	Source data set name (In)	
UEPD TSKA	DS A	Address of skip-key area: exit X	
		should return a key of length X	
		UEPD TKL in this area if it has X	
		requested optimisation of load X	
		by skipping - XDTRD only (In)	

* Valores para UEPDTORC (fornecidos somente para saídas XDTLC) *			

UEPD TLCS	EQU 0	load completed successfully	
UEPD TLFL	EQU 128	load failed	

Figura 9. Continuação da lista de parâmetros de saída de usuário da tabela de dados

As saídas de usuário devem configurar um código de retorno no registro 15. Os valores de código de retorno são fornecidos pela macro DFHUEXIT. Os valores válidos para cada saída de usuário são fornecidos nas descrições a seguir.

Se quiser que seus programas de saída continuem funcionando para tabelas de dados básicos, bem como para tabelas de dados compartilhados, é possível verificar UEPDTFLG para descobrir qual versão de suporte de tabelas de dados chamou o programa de saída. Para tabelas de dados compartilhados, esse byte de sinalização também indica qual tipo de tabela de dados está sendo usado e se o programa de saída está sendo chamado durante o carregamento.

O programa de saída deve usar o nome do arquivo (campo UEPDTNAM) ou o nome do conjunto de dados de origem (consulte os campos UEPDTDSN e UEPD DSL) para determinar se alguma ação deve ser tomada para esse arquivo.

É possível ativar vários programas de saída no mesmo ponto de saída, cada um tomando uma ação para um determinado arquivo ou conjunto de dados.

Saída de usuário XDTRD

A saída de usuário XDTRD é chamada antes de o CICS tentar incluir um registro que tenha sido recuperado do conjunto de dados de origem para a tabela de dados. É possível escolher se você deseja carregar o registro na tabela de dados ou não. Para uma tabela de dados mantida pelo usuário, também é possível modificar o registro.

Normalmente, a XDTRD é chamada quando o processo de carregamento recupera um registro durante a cópia sequencial do conjunto de dados de origem. No entanto, ela também pode ser chamada quando um aplicativo recupera um registro que não está na tabela de dados e uma das condições a seguir se aplica:

- Para uma tabela de dados mantida pelo usuário, o carregamento ainda está em andamento.
- Para uma tabela de dados mantida pelo CICS, o carregamento é finalizado antes de o final do conjunto de dados de origem ser atingido (por exemplo, porque a tabela de dados estava cheia).

O registro recuperado do conjunto de dados de origem é passado como um parâmetro para o programa de saída de usuário - consulte os campos UEPDTRA e UEPDTRL. Esse programa pode escolher (dependendo, por exemplo, do valor da chave - consulte os campos UEPDTKA e UEPDTKL) se deve ou não incluir o registro na tabela de dados.

Como alternativa, o programa de saída pode solicitar que todos os registros subsequentes até uma chave especificada sejam ignorados - consulte o campo UEPDTSKA; esses registros não são passados para o programa de saída. Esse recurso está disponível somente durante o carregamento. É possível especificar a chave como uma chave completa ou especificar somente os caracteres principais preenchendo a área skip-key com zeros binários.

A ação necessária é indicada pela configuração do código de retorno. Dependendo do valor de código de retorno, a ação a seguir é tomada pelo CICS:

Tabela 3. Códigos de retorno para a saída de usuário XDTRD. Um valor de UERCPURG deve ser retornado se o programa de saída recebeu uma resposta PURGED para uma chamada que ele emitiu.

Cód. devolução	Ação
UERCDTAC	Inclua o registro na tabela de dados. Esse é o padrão se a saída não estiver ativada.
UERCDTRJ	Não inclua o registro na tabela de dados.
UERCDTOP	Ignore este registro e os registros a seguir até que seja localizada uma chave igual à ou maior que a chave especificada na área skip-key.

Para uma tabela de dados mantida pelo usuário, o programa também pode modificar os dados no registro para reduzir o armazenamento necessário para a tabela de dados. Programas de aplicativo que usam tabela de dados devem estar cientes de quaisquer mudanças feitas no formato do registro pelo programa de saída. Se o comprimento do registro for alterado, o programa de saída deverá configurar o novo comprimento na lista de parâmetros - consulte o campo UEPDTRL. O novo comprimento não deve exceder o comprimento do buffer de dados — consulte o campo UEPDTRBL.

Programa de saída XDTRD de amostra: DFH\$DTRD

DFH\$DTRD é um programa de saída de usuário global XDTRD de amostra. Ele demonstra o uso da saída de usuário XDTRD para tabelas de dados compartilhados. O programa de amostra é fornecido na biblioteca SDFHSAMP.

Saída de usuário XDTAD

XDTAD é chamado para cada registro incluído no conjunto de dados de origem após o carregamento inicial. Você pode escolher se deseja incluir ou não o registro na tabela de dados. A saída de usuário não pode modificar os registros porque, como os registros são gravados pelo aplicativo, supõe-se que eles já estejam no formato usado na tabela de dados.

A saída de usuário XDTAD é chamada quando uma solicitação de gravação é emitida para uma tabela de dados.

- Para uma tabela de dados mantida pelo usuário, a saída de usuário é chamada uma vez - antes de o registro ser incluído na tabela de dados.
- Para uma tabela de dados mantida pelo CICS, a saída de usuário é chamada duas vezes - antes de o registro ser incluído no conjunto de dados de origem e mais uma vez antes de o registro ser incluído na tabela de dados.

O registro gravado pelo aplicativo é passado como um parâmetro para o programa de saída de usuário — consulte os campos UEPDTRA e UEPDTRL. Esse programa pode escolher (dependendo do valor da chave, por exemplo, consulte os campos UEPDTKA e UEPDTKL) se incluirá ou não o registro na tabela de dados. Essa decisão é indicada pela configuração do código de retorno.

Dependendo do valor de código de retorno, a ação a seguir é tomada pelo CICS:

Tabela 4. Códigos de retorno para saída de usuário XDTAD. Um valor de UERCPURG deve ser retornado se o programa de saída recebeu uma resposta PURGED para uma chamada que ele emitiu.

Cód. devolução	Ação
UERCDTAC	Inclua o registro na tabela de dados. Esse é o padrão se a saída não estiver ativada.
UERCDTRJ	Não inclua o registro na tabela de dados.

A saída XDTAD não deve modificar os dados no registro. Se você usou XDTRD para truncar os registros de dados quando a tabela de dados mantida pelo usuário foi carregada, deve-se codificar seu aplicativo para que ele tente somente gravar registros no formato correto para a tabela de dados.

Programa de saída XDTAD de amostra: DFH\$DTAD

DFH\$DTAD é um programa de saída de usuário global XDTAD de amostra. Isso demonstra o uso da saída de usuário XDTAD para tabelas de dados compartilhados. O programa de amostra é fornecido na biblioteca SDFHSAMP.

Saída de usuário XDTLC

A saída de usuário XDTLC é chamada quando o carregamento da tabela de dados é concluído, independentemente de ser ou não bem-sucedido. A saída de usuário não será chamada se a tabela de dados for fechada por algum motivo antes de o carregamento ser concluído.

O programa de saída será informado se o carregamento não foi concluído com sucesso; consulte o campo UEPDTORC. Isso poderia ocorrer, por exemplo, se o número máximo de registros fosse atingido ou se houvesse armazenamento virtual insuficiente. Nesse caso, o programa de saída pode solicitar que o arquivo seja fechado imediatamente configurando o código de retorno.

Dependendo do valor de código de retorno, a ação a seguir é tomada pelo CICS:

Tabela 5. Códigos de retorno para saída de usuário XDTLC. Um valor de UERCPURG deve ser retornado se o programa de saída recebeu uma resposta PURGED para uma chamada que ele emitiu.

Cód. devolução	Ação
UERCDTOK	Nenhuma ação; o arquivo permanece aberto. Esse é o padrão se a saída não estiver ativada.
UERCDTCL	Feche o arquivo.

Programa de saída XDTLC de amostra: DFH\$DTLC

DFH\$DTLC é um programa de saída de usuário global XDTLC de amostra. Ele demonstra o uso da saída de usuário XDTLC para tabelas de dados compartilhados. O programa de amostra é fornecido na biblioteca SDFHSAMP.

Ativando saídas de usuário para tabelas de dados

Para ativar as saídas de usuário de tabela de dados, conclua estas etapas.

Procedimento

1. Decida quais saídas de usuário você deseja usar. Para obter uma descrição de cada saída de usuário, consulte Capítulo 4, “Customizando Tabelas de Dados Usando as Saídas de Usuário”, na página 39.
2. Grave os programas de saída de usuário. Os exemplos estão incluídos em Capítulo 4, “Customizando Tabelas de Dados Usando as Saídas de Usuário”, na página 39.
3. Defina os programas de saída de usuário para o CICS, usando o comando CEDA DEFINE PROGRAM, conforme descrito em Recursos PROGRAM.
4. Ative as saídas de usuário usando o comando **EXEC CICS ENABLE**. Se for necessário, você pode desativar as saídas de usuário posteriormente usando o comando **EXEC CICS DISABLE**.

A menos que você controle a abertura de uma tabela de dados explicitamente, com um comando CEMT ou EXEC CICS, provavelmente será necessário ativar as saídas de usuário durante a inicialização do CICS. Caso contrário, o carregamento da tabela de dados pode começar antes de as saídas de usuário serem ativadas. Para ativar as saídas de usuário durante a inicialização:

5. Grave um ou mais programas program list table post-initialization (PLTPI) que incluam os comandos **EXEC CICS ENABLE** para ativar as saídas de usuário. Para

obter informações de programação sobre programas PLTPI, consulte Gravando Programas de Inicialização e Encerramento.

6. Defina uma tabela de lista de programas (PLT) com uma entrada para cada um desses programas PLTPI, conforme descrito em Tabela de lista de programas (PLT).
7. Especifique o parâmetro **PLTPI=suffix** para inicialização do sistema, conforme descrito em Parâmetro de Inicialização do Sistema PLTPI. Use o sufixo da PLT que foi definido na etapa anterior. Isso faz com que os programas PLTPI sejam executados no segundo estágio de inicialização, antes de quaisquer arquivos serem abertos.

O que Fazer Depois

É possível usar programas PLT shutdown (PLTSD) de uma maneira semelhante para desativar as saídas de usuário durante o encerramento do CICS.

Capítulo 5. Administrando tabelas de dados

Informações sobre os aspectos operacionais das tabelas de dados.

Abrindo uma tabela de dados

Uma tabela de dados deve ser aberta antes de poder ser usada por um aplicativo.

Você abre uma tabela de dados da mesma maneira que abriria qualquer arquivo CICS usando um dos métodos a seguir:

- Automaticamente, pela transação CSFU fornecida pelo CICS, no final da inicialização do CICS, se a tabela de dados for definida com OPENTIME(STARTUP).
- Explicitamente por uma solicitação CEMT ou EXEC do CICS emitida pelo usuário.
- Implicitamente, na primeira referência à tabela de dados, se a tabela de dados estiver definida com OPENTIME(FIRSTREF). O primeiro acesso remoto a uma tabela de dados fechada a abre implicitamente.

Todas as regras e opções para abrir um arquivo CICS também se aplicam a um arquivo definido como uma tabela de dados. Além disso, o carregamento da tabela de dados é iniciado.

Para uma tabela de dados grande, o carregamento poderia levar um tempo significativo. Desenvolvimento para acesso a tabelas de dados discute os comandos de programação de aplicativos que podem ser usados com uma tabela de dados mantida pelo usuário e a maneira como ganhos de desempenho que podem ser obtidos com uma tabela de dados mantida pelo CICS são limitados até que o carregamento seja concluído.

As etapas a seguir são realizadas durante a abertura do arquivo:

1. O bloco de controle de método de acesso (ACB) para o conjunto de dados de origem VSAM é aberto sob um bloco de controle de tarefas (TCB) do MVS separado. Essa etapa é a mesma que para qualquer arquivo CICS.
2. Para a primeira tabela de dados usada por uma região, o CICS:
 - cria conjuntos de armazenamentos MVS para uso pelo suporte de tabela de dados compartilhados
 - cria um espaço para dados MVS para uso pelas tabelas de dados desta região
 - tenta uma operação LOGON como um servidor
3. Uma transação especial do CICS, CFTL, é conectada para carregar a tabela de dados no espaço para dados.
4. A transação que emitiu a solicitação para abrir a tabela de dados agora pode continuar o processamento.
5. O CICS emite uma mensagem DFHFC0940 para indicar que o carregamento foi iniciado. A mensagem é enviada para a fila de dados temporários CSFL.
6. A transação que carrega a tabela de dados lê o conjunto de dados de origem sequencialmente. Sob o controle opcional da saída de usuário XDTRD, a transação copia os registros no armazenamento de espaço para dados.

7. O CICS emite uma mensagem para indicar o resultado do carregamento. O número da mensagem é:
 - se o carregamento for bem-sucedido: DFHFC0941
 - se o carregamento falhar: DFHFC0942, DFHFC0943, DFHFC0945, DFHFC0946, DFHFC0947 ou DFHFC0948

A mensagem é enviada para a fila de dados temporários CSFL. Além disso, se o carregamento falhar, a mensagem será enviada para o console.

8. Quando o carregamento for concluído (seja ele bem-sucedido ou não), a saída de usuário XDTLC será chamada se estiver ativa. Se o carregamento não foi concluído com sucesso, o programa de saída poderá solicitar que a tabela de dados seja fechada.
9. Para uma tabela de dados mantida pelo usuário, o ACB para o conjunto de dados de origem será fechado quando o carregamento for concluído. O conjunto de dados será desalocado se originalmente ele foi alocado dinamicamente e ficará disponível para outras tarefas, contanto que não haja mais nenhum outro ACB aberto para ele.

Nota: Durante uma reinicialização emergencial, qualquer arquivo que requer ação de restauração será reaberto. No entanto, se o arquivo for definido como uma tabela de dados, o carregamento não será iniciado nesse momento; em vez disso, ele será iniciado pela transação CSFU no final da reinicialização emergencial. Isso dá uma oportunidade para quaisquer saídas de usuário que controlam a cópia de registros na tabela de dados durante o carregamento serem ativadas em qualquer estágio do processamento PLTPI.

Fechando uma tabela de dados

Uma tabela de dados é fechada da mesma maneira que para qualquer arquivo CICS.

Use um dos seguintes métodos:

- Explicitamente, por meio de uma solicitação CEMT ou EXEC CICS emitida pelo usuário.
- Implicitamente, quando o CICS for encerrado normalmente.

Todas as regras e opções para fechar um arquivo CICS também se aplicam a uma tabela de dados. Em particular:

- As regras sobre o quiesce dos usuários atuais do arquivo se aplicam (exceto que o arquivo pode ser fechado, mesmo quando uma transação que está em execução em uma AOR esteja no meio de uma sequência de procura).
- Para uma tabela de dados mantida pelo usuário, se a tabela de dados for definida como recuperável, todas as unidades de trabalho que mudaram a tabela de dados deverão ser concluídas para que a tabela de dados possa ser fechada.

O armazenamento de espaço para dados que é usado para registros de tabela de dados é liberado como parte da operação de encerramento. Se um arquivo for reaberto após ter sido fechado, o processamento será o mesmo que se o arquivo não tivesse sido aberto anteriormente.

Usando suporte de tabelas de dados compartilhados em um sysplex

Leia estas informações se você estiver usando atualmente tabelas de dados mantidas pelo usuário em um único ambiente MVS, mas estiver planejando mudar para um ambiente sysplex. Isso também pode ser útil se você já tiver um sysplex, pois ele pode mostrar como explorar suporte de tabelas de dados compartilhados nesse ambiente.

Visão geral do suporte a tabelas de dados compartilhados em um sysplex

Uma tabela de dados compartilhados pode explorar o suporte a tabelas de dados compartilhados somente em uma imagem única do MVS. No entanto, é possível ampliar o uso de tabelas de dados compartilhados para um ambiente sysplex para um aplicativo que requer somente acesso de leitura a uma tabela de dados mantida pelo usuário compartilhada ou para um aplicativo que não requer que mudanças sejam vistas imediatamente.

Observe que uma tabela de dados compartilhados pode ser compartilhada usando remessa de função em imagens do MVS.

É possível replicar uma tabela de dados mantida pelo usuário no sysplex, com uma tabela de dados por MVS. Deve-se ter uma região de servidor de tabela de dados compartilhados em cada imagem do MVS, cada uma com uma tabela de dados mantida pelo usuário que possa ser acessada usando a tabela de dados compartilhados por qualquer uma das outras regiões CICS dentro desse MVS. Essas outras regiões requerem definições de arquivo remoto que se referem à tabela de dados mantida pelo usuário em sua região de servidor. Cada tabela de dados mantida pelo usuário (UMT) deve ter o mesmo conjunto de dados de origem, e esse conjunto de dados de origem deve ser legível por todas as regiões de servidor da tabela de dados compartilhados. Se o acesso for somente leitura, com dados nunca atualizados, efetivamente, isso fornecerá uma tabela de dados mantida pelo usuário compartilhada em um sysplex.

Se, como é provável, os dados subjacentes mudarem de tempos em tempos, mas não precisarem refletir tais mudanças imediatamente nas UMTs, você pode realizar alguns processamentos periodicamente para atualizar o conteúdo das UMTs, para que elas sejam atualizadas para corresponder aos dados subjacentes sem precisar fechar e recarregar as UMTs. As mudanças são aplicadas ao conjunto de dados de origem, e não à tabela de dados mantida pelo usuário, usando aplicativos CICS que se referem ao conjunto de dados por uma definição de arquivo de tabela de não dados ou usando programas em lote. Um exemplo de programa de aplicativo COBOL, que é descrito em “Código de origem para o programa de exemplo atualizar uma tabela de dados mantida pelo usuário replicada” na página 54, ilustra como é possível atualizar as UMTs para refletir o conteúdo atual do conjunto de dados de origem. O programa seria executado em cada imagem do MVS e atualizaria a UMT nessa imagem. Tal programa seria executado em horários regulares durante o dia ou mediante solicitação do usuário. Seria mais eficiente executá-lo nas regiões de servidor da tabela de dados compartilhados para evitar atualizações de remessa de função para a UMT.

Se for crítico que as regiões CICS em todas as imagens do MVS no sysplex sejam sincronizadas em sua visualização dos dados, as transações que leem os dados devem ser interrompidas enquanto os programas de atualização são executados e ser reiniciadas somente após os programas terem sido concluídos em todos os sistemas MVS.

Essa técnica é apropriada somente para tabelas de dados mantidas pelo usuário porque:

- Onde o acesso somente leitura é necessário, uma tabela de dados mantida pelo usuário é a opção usual.
- Não seria possível com uma tabela de dados mantida pelo CICS aplicar atualizações no conjunto de dados de origem enquanto se mantém as tabelas intactas.
- Qualquer atualização feita no conjunto de dados de origem seria refletida somente na tabela no sistema em que a atualização foi feita.

Como atualizar tabelas de dados mantidas pelo usuário replicadas

As etapas a seguir descrevem como configurar um ambiente para atualizar UMTs replicadas. Na prática, você já pode ter algumas em vigor. Por exemplo, você já pode ter arquivos definidos como tabelas de dados. As etapas descritas aqui supõem que você já tenha um ambiente sysplex.

1. Selecione um arquivo que seja apropriado.

Como ilustração, considere um aplicativo que verifica números de cartão de crédito com relação a uma lista de cartões de crédito roubados e que requer rápido acesso a essa lista. A lista é atualizada periodicamente com novos lotes de números de cartões roubados. O aplicativo acessa registros em um conjunto de dados VSAM KSDS denominado PRODN.SOURCEDS usando um nome de arquivo UMTNAME. O aplicativo é executado em um sysplex que consiste em duas imagens MVS. As regiões CICS, CICS1A, CICS1B e CICS1C, são executadas na primeira imagem e CICS2A, CICS2B e CICS2C são executadas na segunda.

2. Configure as definições de arquivo:

- Em cada MVS no sysplex, selecione uma região CICS para ser um servidor de tabela de dados compartilhados para esse arquivo. Nessa região, defina o nome do arquivo por meio do qual seus aplicativos leem os dados como uma tabela de dados mantida pelo usuário, com o nome do conjunto de dados sendo aquele que contém os dados de origem.

Nesta ilustração, CICS1A e CICS2A são configuradas como regiões do servidor, e são definidos para eles arquivos chamados UMTNAME. As definições de arquivo especificam DSNAME como PRODN.SOURCEDS, TABLE como USER e as operações permitidas de YES para READ, BROWSE, ADD, DELETE e UPDATE (pois essa definição de arquivo é usada para ler os dados e para atualizar a UMT quando ela for atualizada).

- Para todas as outras regiões no sysplex, defina o nome do arquivo pelo qual os aplicativos leem os dados como um arquivo remoto com REMOTESYSTEM como a região de servidor da tabela de dados compartilhados no mesmo MVS e REMOTENAME como o nome da UMT nessa região.

Portanto, nesta ilustração, arquivos chamados UMTNAME seriam definidos em CICS1B e CICS1C com o REMOTESYSTEM sendo o sysid para CICS1A e REMOTENAME como UMTNAME, dessa vez com READ e BROWSE sendo as únicas operações permitidas, pois não há necessidade de a UMT ser atualizada por meio dessas definições remotas. Definições de arquivo semelhantes são configuradas em CICS2B e CICS2C, mas para elas, CICS2A é o sistema remoto.

- Em cada região de servidor de tabela de dados compartilhados, configure uma definição de arquivo que possa ser usada para ler o conjunto de dados de origem quando UMTs forem atualizadas.

Nesta ilustração, arquivos denominados SOURCEDDS são definidos para CICS1A e CICS2A, com DSNAMES como PRODN.SOURCEDDS, TABLE como NO e permitindo somente operações READ e BROWSE.

- Em uma região no sysplex (que tem acesso ao conjunto de dados de origem), defina um arquivo que seja usado para aplicar atualizações à origem. A definição de arquivo poderia ser a mesma que a usada pelo programa de atualização para ler o conjunto de dados de origem, mas nesse caso, seria necessário permitir as operações de leitura e atualização. Se preferir, você pode decidir atualizar o conjunto de dados usando um programa em lote, em cujo caso essa definição de arquivo do CICS não seria necessária.

Esta ilustração usa a mesma definição de arquivo que a usada na atualização de UMTs. Nesse caso, uma das regiões precisaria definir SOURCEDDS como permitindo todas as operações de arquivo.

3. Configure o conjunto de dados de origem para que ele possa ser acessado por todos os aplicativos que precisam lê-lo ou atualizá-lo.

Se você tiver DFSMS/MVS versão 1 liberação 3, é possível acessar o conjunto de dados a partir de qualquer região CICS para ler ou atualizar, especificando RLSACCESS(Yes) nas definições de arquivo. Observe que, se você usar o modo de acesso RLS, a menos que o conjunto de dados seja não recuperável, não será possível aplicar nele as atualizações de um programa em lote (pois somente o CICS pode abrir um conjunto de dados recuperável para atualização no modo RLS).

Se você estiver em uma liberação anterior do DFSMS/MVS, é possível configurar o conjunto de dados SHAREOPTIONS para que ele possa ser atualizado pelo programa que aplica atualizações na origem e lido por todos os usuários. Como alternativa, é possível configurar o conjunto de dados para que ele possa ser atualizado somente quando não estiver sendo lido e assegurar que sua abertura seja serializada. Para as opções de compartilhamento operarem em todo o sysplex, deve-se usar GRS (Global Resource Serialization).

Nesta ilustração, se o RLS não estiver disponível, defina PRODN.SOURCEDDS com:

- SHR(2), para que ele possa ser atualizado pela região que executa o programa que aplica mudanças no conjunto de dados e, ao mesmo tempo, lido por todos os programas de atualização,

ou

- SHR(1), e, normalmente, deixe-o aberto para o programa que aplica mudanças; então, quando ele tiver que ser atualizado, feche esse acesso a ele e, em cada região de servidor, sucessivamente, abra-o, execute o programa de atualização, feche-o e desative-o para permitir que a próxima região o abra.

4. Modifique o programa de exemplo para que ele nomeie seus arquivos para a UMT e o conjunto de dados de origem, e para que as definições de dados correspondam ao layout de seus registros. Defina o programa e a transação nas regiões do servidor.

Os nomes de arquivo na ilustração são os mesmos que aqueles no programa (UMTNAME e SOURCEDDS). Defina o programa e uma transação para executá-lo em CICS1A e CICS2A.

5. Agora você deve estar pronto para começar a usar UMTs replicadas.
6. Prepare o conjunto de dados de origem com seu conteúdo inicial.
7. Abra as UMTs nas regiões de servidor de tabela de dados compartilhados para fazer com que o conteúdo do conjunto de dados de origem seja carregado em cada uma.
8. Inicie os aplicativos em todas as regiões no sysplex. Todos eles poderão acessar os dados usando compartilhamento de tabela de dados.
Os aplicativos em execução no MVS 1 acessarão os dados por meio da UMT em CICS1A, e aqueles em execução no MVS 2 os acessarão por meio da UMT em CICS2A.
9. Quando novos dados chegarem, atualize o conjunto de dados de origem.
Nesta ilustração, os dados são atualizados pelo arquivo SOURCEDS.
10. Quando quiser que os aplicativos acessem os novos dados, execute as transações em cada região de servidor que lerá o conjunto de dados de origem e a UMT e atualize a UMT para estar no mesmo nível que o conjunto de dados de origem. Contanto que os aplicativos não sejam invalidados se os dados vistos em um MVS forem um pouco diferentes dos vistos em outro, você não precisa parar a execução dos aplicativos enquanto faz sua atualização.

Programa de exemplo para atualizar uma tabela de dados mantida pelo usuário

Para ajudá-lo a gravar seu próprio programa, aqui está um exemplo de um programa COBOL que demonstra como atualizar uma UMT enquanto ela ainda está aberta para corresponder ao conjunto de dados de origem.

Se atualizações forem aplicadas com frequência ao conjunto de dados de origem e pudessem ser aplicadas enquanto um programa de atualização está em execução, isso significaria que o conjunto de dados de origem nunca seria refletido exatamente pela UMT, pois o registro sendo processado ou os registros já processados poderiam mudar. Isso significa que o programa deve ser tolerante com a possibilidade de mudança de registros. O programa também é gravado para permitir a possibilidade de a própria UMT ser atualizada por outros programas, embora não seja recomendado operar dessa maneira (ou seja, o único programa que atualiza a UMT deve ser o programa de atualização).

Como o programa de exemplo opera

Primeiro, o ambiente é inicializado. É feita uma verificação para saber se o arquivo UMT é local e já está aberto. Se o arquivo UMT for remoto, o programa emitirá uma mensagem e será finalizado. Se o arquivo UMT não estiver aberto, o programa o abrirá e será finalizado (pois a abertura da UMT carregará os dados mais recentes do conjunto de dados de origem sem a necessidade de fazer outros processamentos). Também é feita uma verificação para saber se o arquivo de origem é local; se ele for remoto, o programa emitirá uma mensagem e será finalizado. O arquivo que acessa diretamente o conjunto de dados de origem da UMT é aberto. Operações de procura iniciais são realizadas em ambos os arquivos para permitir que o programa os percorra em sequência.

Se o ambiente for configurado sem erro, a atualização da UMT começará. Isso envolve a recuperação e a comparação de pares de registros, um da UMT e um do conjunto de dados de base.

Os registros recuperados são comparados:

- Se os registros forem iguais, as sinalizações serão configuradas para ler o próximo registro da UMT e do conjunto de dados.
- Se a UMT tiver uma chave maior do que o conjunto de dados, há um registro no conjunto de dados que deverá ser incluído na UMT.
- Se o conjunto de dados tiver uma chave maior que a UMT, há um registro extra na UMT que deverá ser removido.
- Se as chaves forem iguais, mas os registros forem diferentes, a UMT deverá ser atualizada com o registro no conjunto de dados.

Se um registro tiver que ser incluído na UMT, uma operação de gravação será executada.

- Se a operação de gravação for bem-sucedida, o programa continuará processando o próximo par de registros.
- Se a operação de gravação falhar porque um registro foi inserido por outra transação entre as operações de leitura e gravação executadas pelo programa, será feita uma tentativa de excluir o registro e gravá-lo novamente.
- Se a segunda tentativa falhar, o programa processará o próximo par de registros.
- Quando o próximo par de registros for processado, o atual registro da UMT será comparado com o próximo registro no conjunto de dados para verificar outras omissões de registro de UMT.

Se um registro tiver que ser excluído da UMT, uma operação de exclusão será executada.

- Se a operação de exclusão for bem-sucedida, o programa continuará processando o próximo par de registros.
- Se a operação de exclusão falhar porque o registro já foi excluído entre as operações de leitura e exclusão, o programa continuará processando o próximo par de registros.
- Quando o próximo par de registros for processado, o atual registro do conjunto de dados será comparado com o próximo registro na UMT para verificar outros registros que não devem estar na UMT.

Se um registro tiver que ser atualizado na UMT, uma leitura para a operação de atualização será executada para obter um bloqueio no registro.

- Se essa ação for bem-sucedida, o registro atualizado será regravado para a UMT e o programa continuará processando o próximo par de registros.
- Se a operação falhar porque outra transação excluiu o registro, uma operação de gravação será executada para colocá-lo de volta.
- Se a operação de gravação falhar, o programa continuará processando o próximo par de registros.
- Quando o próximo par de registros for processado, novos registros serão lidos da UMT e do conjunto de dados.

Quando o final dos dois arquivos tiver sido atingido, e não houver mais registros para processar, o programa executará procuras finais no conjunto de dados e na UMT e retornará. Observe que o exemplo não fecha o arquivo que acessa diretamente o conjunto de dados. Se o conjunto de dados não puder operar uma atualização em um ambiente compartilhado, o arquivo que o acessa deverá ser configurado para FECHADO DESATIVADO para permitir sua atualização.

O programa captura quaisquer erros inesperados e emite uma mensagem de erro na tela. Apenas a primeira operação na UMT é verificada (operações de exclusão, gravação ou leitura / regravação). Se isso falhar com um código de retorno causado pela mudança de um registro após ele ser lido originalmente, será feita uma tentativa final de corrigir o registro, mas essa tentativa não será verificada. Isso é para evitar que o programa entre em um estado de loop.

Há comentários adicionais no código.

Configurando e executando o programa de exemplo

Edite o programa, de acordo com os comentários no exemplo, para corresponder ao formato dos registros sendo atualizados.

'UMTNAME' e 'SOURCEDS' devem ser renomeados para corresponderem às definições de arquivo.

Converta, compile e vincule o programa usando um compilador COBOL.

Defina o programa para CICS e defina uma transação para o programa. Defina o arquivo (UMTNAME) para apontar para UMT e dê a ele um conjunto de dados de origem do qual carregar ao ser aberto pela primeira vez. Defina o outro arquivo (SOURCEDS) para apontar diretamente para o conjunto de dados de origem do qual a UMT está definida para carregamento.

Cada sysplex deve ter uma região CICS na qual a UMT a ser atualizada reside. Nessas regiões, as definições necessárias para a execução da transação de atualização devem estar instaladas. Em todas as outras regiões no sysplex, a UMT deve ser definida como um arquivo remoto, apontando para a UMT na região pertencente à UMT. Não é necessário executar a transação de atualização nas regiões que têm a UMT definida como remota.

A estratégia de atualização usada dependerá da maneira como o conjunto de dados estiver configurado. Se o conjunto de dados de origem estiver configurado como RLS, todas as UMTs poderão ser atualizados ao mesmo tempo. Quaisquer atualizações no conjunto de dados de origem também poderiam ser aplicadas. Se o conjunto de dados tiver SHAREOPTIONS configurado para que possa ser lido por diversos sistemas de uma vez, assim como com RLS, uma atualização simultânea também pode ocorrer. Caso contrário, quando o conjunto de dados de origem fosse atualizado, o arquivo que é usado para ler o conjunto de dados de origem para atualização precisaria ser fechado e desativado em cada sistema enquanto durar a atualização. Se todos os UMTs forem atualizados em série, o conjunto de dados de origem pode ser aberto e fechado para cada região da UMT sucessivamente quando necessário para atualização.

Código de origem para o programa de exemplo atualizar uma tabela de dados mantida pelo usuário replicada

Este código de origem não é fornecido na biblioteca de amostras do CICS, somente nesta documentação.

Programa de exemplo para atualizar uma UMT replicada: CBL XOPTS(SP)

Nome do programa

UMTUPDT COBOL

Nome Descritivo

Aplicativo CICS para atualizar dinamicamente uma UMT com o atual conteúdo de um conjunto de dados

Visão geral

Este programa demonstra como atualizar uma tabela mantida pelo usuário (UMT) para correspondência dos dados no conjunto de dados de origem do qual foi carregada enquanto permanece em uso por um (ou mais) sistemas CICS. Ele pode ser usado para atualizar uma UMT replicada em diferentes sysplexes, para que todos correspondam ao conjunto de dados de origem. Ele deve ser executado no FOR.

Requisitos

Este programa deve ser convertido, compilado e vinculado como um programa CICS COBOL e definido para CICS. Um nome de transação deve ser definido para esse programa. Um arquivo de UMT, atualmente chamado UMTNAME, é usado para acessar a UMT, e um arquivo de conjunto de dados de origem, atualmente chamado SOURCEDS, é usado para acessar diretamente o conjunto de dados do qual a UMT foi carregada. Essas definições devem ser instaladas somente na região na qual a UMT reside (o FOR). Nenhuma região no mesmo sysplex que usa a UMT remotamente precisa executar qualquer processo de atualização.

Descrição

Primeiro, o programa inicializará os dois arquivos necessários e começará a procurá-los desde o início. A abertura da UMT fará com que ela seja carregada se não for aberta. Se a UMT não for aberta e for carregada, a operação do programa será efetivamente redundante e o código de atualização não será executado. O programa também irá verificar um nome de sistema remoto. Se houver algum presente para qualquer arquivo, o programa não será executado. Isso é para evitar que ocorra uma remessa de função, o que obviamente comprometeria o desempenho.

O programa lerá continuamente um par de registros de dois arquivos e irá compará-los, incluindo, excluindo ou atualizando quaisquer registros na UMT que não correspondam ao conjunto de dados de origem.

As chaves do par de registros são comparadas. Se a chave para a UMT e a chave para o conjunto de dados de origem forem iguais, e se os registros corresponderem, nenhuma atualização será necessária. Se as chaves forem iguais, mas os registros forem diferentes, o registro no conjunto de dados de origem será usado para atualizar a UMT. Se a chave na UMT for maior que a chave no conjunto de dados de origem, os registros no conjunto de dados de origem serão gravados na UMT até que as chaves fiquem iguais ou até que a chave da UMT fique menor que a chave do conjunto de dados de origem. Se a chave da UMT for menor que a chave do conjunto de dados de origem, os registros na UMT serão removidos até que as chaves fiquem iguais ou até que a chave da UMT fique maior que o conjunto de dados de origem. Isso continua até o final de ambos os arquivos ser atingido ou até que ocorra um erro inesperado.

Quaisquer erros que foram inesperados serão relatados na tela e a operação do programa parará. Alguns erros são descobertos e é feita uma nova tentativa de atualizar a UMT. Se essa tentativa falhar, nenhuma ação adicional é tomada para esses registros e o programa continua processando o próximo par.

Modificando o programa

Este programa pode não funcionar do jeito que ele é. A estrutura de

registro que ele usa supõe que uma chave de 4 caracteres seja usada para acessar um registro de 40 caracteres. As mudanças a seguir precisarão ser feitas para permitir que esse programa funcione com diferentes tipos de registro.

A chave acessa a UMT e o conjunto de dados de origem deve ser alterado. As variáveis que armazenam a chave são UMT-KEY e DS-KEY.

O comprimento dos registros é mantido em UMT-LEN e DS-LEN.

As variáveis de registro da UMT e do conjunto de dados de origem devem ser alteradas. As variáveis que armazenam isso são UMT-REC (que contém UMT-REC-KEY e UMT-REC-TEXT) e DS-REC (que contém DS-REC-KEY e DS-REC-TEXT). Obviamente, arquivos adicionais podem ser incluídos conforme a necessidade.

O nome do arquivo da UMT é configurado como UMTNAME. Isso pode ser alterado para corresponder a qualquer UMT já definida. O arquivo do conjunto de dados de origem é configurado como SOURCEDS e também pode ser alterado.

Código de origem

```
IDENTIFICATION DIVISION.  
PROGRAM-ID. UMTUPDT.  
  
ENVIRONMENT DIVISION.  
EJECT.  
  
DATA DIVISION.  
  
SEÇÃO DE ARMAZENAMENTO DE FUNCIONAMENTO.  
  
* Declare as variáveis de registro UMT e DS  
77 UMT-KEY          PIC X(4)  VALUE '0000'.  
77 UMT-LEN          PIC 9(2)  VALUE 40.  
01 UMT-REC.  
    03 UMT-REC-KEY   PIC X(4)  VALUE SPACES.  
    03 UMT-REC-TEXT  PIC X(36) VALUE SPACES.  
  
77 DS-KEY          PIC X(4)  VALUE '0000'.  
77 DS-LEN          PIC 9(2)  VALUE 40.  
01 DS-REC.  
    03 DS-REC-KEY    PIC X(4)  VALUE SPACES.  
    03 DS-REC-TEXT   PIC X(36) VALUE SPACES.  
  
* Declare outras variáveis de trabalho  
* Sequências de saída de tela  
01 MESSAGE-OUTPUT  PIC X(26) VALUE 'UMT SUCCESSFULLY REFRESHED'.  
01 REMOTE-OUTPUT   PIC X(25) VALUE 'FILE RESOURCE NOT LOCAL'.  
01 ERROR-OUTPUT.  
    03 ERROR-OPNAME  PIC X(8)  VALUE SPACES.  
    03 FILLER        PIC X(15) VALUE ' RETURNED RESP '.  
    03 ERROR-RESP    PIC X(8)  VALUE SPACES.  
    03 FILLER        PIC X(7)  VALUE ' RESP2 '.  
    03 ERROR-RESP2   PIC X(8)  VALUE SPACES.  
    03 FILLER        PIC X(10) VALUE ' FOR FILE '.  
    03 ERROR-FILE    PIC X(8)  VALUE SPACES.  
  
* Fim de sinalizações de arquivo  
77 UMT-EOF          PIC 9(1)  VALUE 0.  
77 DS-EOF           PIC 9(1)  VALUE 0.  
  
* Sinalizações de recuperação de registro  
77 GET-NEXT-UMT     PIC 9(1)  VALUE 1.  
77 GET-NEXT-DS      PIC 9(1)  VALUE 1.
```



```

* Variáveis de consulta de arquivo
77 REM-SYS-NAME    PIC X(4)  VALUE SPACES.
77 OPEN-STAT      PIC S9(8)  BINARY.

* Sinalizações de operação de programa
77 PROCESS-FILES   PIC 9(1)  VALUE 1.
77 REM-FILE        PIC 9(1)  VALUE 0.
77 UMT-STARTBR     PIC 9(1)  VALUE 0.
77 DS-STARTBR      PIC 9(1)  VALUE 0.

* Variáveis de resposta EXEC CICS
77 RESPONSE        PIC S9(8)  BINARY.
77 RESPONSE2       PIC S9(8)  BINARY.

COPY DFHAID.
COPY DFHBMSCA.

LINKAGE SECTION.
EJECT.

PROCEDURE DIVISION USING DFHEIBLK.

*****
* Processamento principal começa aqui.
*****
MAIN-PROCESSING SECTION.

* Verifique a UMT e o conjunto de dados para processamento
  PERFORM FILE-CHECK.

* Se a verificação de arquivo foi concluída sem problemas, processe a UMT
  IF (PROCESS-FILES = 1)

* Apronte a UMT e o DS para acesso
  PERFORM INITIALIZE

* Chame a rotina de atualização até que o final de ambos
arquivos seja atingido
  PERFORM UPDATE-UMT UNTIL (DS-EOF = 1 AND UMT-EOF = 1)

  END-IF.

* Saia normalmente do programa
  PERFORM TRAN-FINISH.

MAIN-PROCESSING-EXIT.
GOBACK.
EJECT

*****
* Procedimentos começam aqui.
*****
* Verifique o status aberto dos arquivos e se eles não são remotos
*****
FILE-CHECK SECTION.

* Consulte a UMT para obter informações de status aberto
e remoto
  MOVE SPACES TO REM-SYS-NAME.
  EXEC CICS INQUIRE FILE('UMTNAME')
    OPENSTATUS(OPEN-STAT)
    REMOTESYSTEM(REM-SYS-NAME)
    RESP(RESPONSE)

```

```

        RESP2(RESPONSE2)
    END-EXEC.
* Emita um erro se a consulta na UMT falhou
    IF (RESPONSE NOT = DFHRESP(NORMAL))
        MOVE 'INQUIRE ' TO ERROR-OPNAME
        MOVE 'UMTNAME ' TO ERROR-FILE
        PERFORM PROCESS-ERROR
    END-IF.
* O nome do sistema não ficará em branco se o arquivo for definido
como remoto
* Não desejamos executar nenhum processamento se o arquivo for remoto
    IF (REM-SYS-NAME NOT = SPACES)
        MOVE 0 TO PROCESS-FILES
        MOVE 1 TO REM-FILE
    ELSE
* Se a UMT não for aberta, sua abertura a atualizará
        IF (OPEN-STAT NOT = DFHVALUE(OPEN))
            EXEC CICS SET FILE('UMTNAME')
                OPEN
                RESP(RESPONSE)
                RESP2(RESPONSE2)
            END-EXEC
* Verifique se a abertura da UMT foi bem-sucedida
            IF (RESPONSE NOT = DFHRESP(NORMAL))
                MOVE 'OPEN ' TO ERROR-OPNAME
                MOVE 'UMTNAME ' TO ERROR-FILE
                PERFORM PROCESS-ERROR
            ELSE
* Não desejamos executar nenhum processamento, já que a abertura
atualizará a UMT
                MOVE 0 TO PROCESS-FILES
            END-IF
        END-IF
    END-IF.

* Consulte o conjunto de dados de origem para obter os status aberto
e remoto
    MOVE SPACES TO REM-SYS-NAME.
    EXEC CICS INQUIRE FILE('SOURCEDS')
        REMOTESYSTEM(REM-SYS-NAME)
        OPENSTATUS(OPEN-STAT)
        RESP(RESPONSE)
        RESP2(RESPONSE2)
    END-EXEC.
* Emita um erro se a consulta no conjunto de dados falhou
    IF (RESPONSE NOT = DFHRESP(NORMAL))
        MOVE 'INQUIRE ' TO ERROR-OPNAME
        MOVE 'SOURCEDS' TO ERROR-FILE
        PERFORM PROCESS-ERROR
    END-IF.
* Não faça nenhum processamento se esse for um arquivo remoto
    IF (REM-SYS-NAME NOT = SPACES)
        MOVE 0 TO PROCESS-FILES
        MOVE 1 TO REM-FILE
    ELSE
* Abra o conjunto de dados de origem
        IF (OPEN-STAT = DFHVALUE(CLOSED))
            EXEC CICS SET FILE('SOURCEDS')
                OPEN
                RESP(RESPONSE)
                RESP2(RESPONSE2)
            END-EXEC
* Verifique se a abertura do conjunto de dados foi bem-sucedida
            IF (RESPONSE NOT = DFHRESP(NORMAL))
                MOVE 'OPEN ' TO ERROR-OPNAME
                MOVE 'SOURCEDS' TO ERROR-FILE
                PERFORM PROCESS-ERROR
            END-IF
        END-IF
    END-IF

```

```

        END-IF
    END-IF
END-IF.

FILE-CHECK-EXIT.
EXIT.
EJECT

*****
* Inicialize os arquivos prontos para leitura sequencial *
*****
INITIALIZE SECTION.

* Comece a procurar a UMT no primeiro registro
EXEC CICS STARTBR FILE('UMTNAME')
    RIDFLD(UMT-KEY)
    GTEQ
    RESP(RESPONSE)
    RESP2(RESPONSE2)
END-EXEC.

* Se a UMT estiver vazia (NOTFND), trate como o final da UMT e preencha
IF (RESPONSE = DFHRESP(NOTFND))
    MOVE 1 TO UMT-EOF
ELSE

* Emita um erro se o início da procura pela UMT falhou
    IF (RESPONSE NOT = DFHRESP(NORMAL))
        MOVE 'STARTBR ' TO ERROR-OPNAME
        MOVE 'UMTNAME ' TO ERROR-FILE
        PERFORM PROCESS-ERROR
    END-IF
END-IF.

* Configure a sinalização de início de procura da UMT
MOVE 1 TO UMT-STARTBR.

* Comece a procurar o conjunto de dados no primeiro registro
EXEC CICS STARTBR FILE('SOURCEDS')
    RIDFLD(DS-KEY)
    GTEQ
    RESP(RESPONSE)
    RESP2(RESPONSE2)
END-EXEC.

* Se o conjunto de dados estiver vazio, trate como o final do
conjunto de dados uma UMT vazia
IF (RESPONSE = DFHRESP(NOTFND))
    MOVE 1 TO DS-EOF
ELSE

* Emita um erro se o início da procura pelo conjunto de dados falhou
    IF (RESPONSE NOT = DFHRESP(NORMAL))
        MOVE 'STARTBR ' TO ERROR-OPNAME
        MOVE 'SOURCEDS' TO ERROR-FILE
        PERFORM PROCESS-ERROR
    END-IF
END-IF.

* Configure a sinalização de início de procura do conjunto de dados
MOVE 1 TO DS-STARTBR.

INITIALIZE-EXIT.
EXIT.
EJECT

*****
* Atualize a UMT de acordo com os estados registro / chave *
*****

```

UPDATE-UMT SECTION.

```
* Vá para os próximos registros de UMT e conjunto de dados
  PERFORM READ-FILES.

* Se os registros forem os mesmos, vá para o próximo registro
  IF UMT-REC = DS-REC
    MOVE 1 TO GET-NEXT-UMT
    MOVE 1 TO GET-NEXT-DS
  ELSE

* Se a UMT estiver atrás de um conjunto de dados, haverá
um registro extra na UMT, portanto, exclua-o.
* Exclua também os registros da UMT se EOF DS foi atingido antes
de EOF UMT
    IF (UMT-EOF = 0 AND (UMT-KEY < DS-KEY OR DS-EOF = 1))
      PERFORM UMT-DELETE
    END-IF

* Se a UMT estiver à frente do conjunto de dados, haverá um registro
extra no DS, portanto, inclua na UMT
* Além disso, inclua registros na UMT se o EOF foi atingido
antes do EOF DS
    IF (DS-EOF = 0 AND (UMT-KEY > DS-KEY OR UMT-EOF = 1))
      PERFORM UMT-WRITE
    END-IF

* Se as duas chaves forem iguais, mas com registros diferentes, atualize a UMT
    IF ((DS-EOF = 0 AND UMT-EOF = 0) AND UMT-KEY = DS-KEY)
      PERFORM UMT-UPDATE
    END-IF

  END-IF.
```

UPDATE-UMT-EXIT.

EXIT.
EJECT

```
*****
* Leia o próximo registro dos dois arquivos
*****
READ-FILES SECTION.
```

```
* Se as sinalizações forem configuradas para ler o próximo
registro de UMT, execute isto
  IF (GET-NEXT-UMT = 1 AND UMT-EOF = 0)
    MOVE SPACES TO UMT-REC
    EXEC CICS READNEXT FILE('UMTNAME')
      RIDFLD(UMT-KEY)
      INTO(UMT-REC)
      RESP(RESPONSE)
      RESP2(RESPONSE2)
    END-EXEC

* Configure a sinalização EOF se o final da UMT foi atingido
  IF (RESPONSE = DFHRESP(ENDFILE))
    MOVE 1 TO UMT-EOF
  ELSE

* Emita um erro se o código de retorno de READ for inesperado
  IF (RESPONSE NOT = DFHRESP(DUPKEY) AND
    RESPONSE NOT = DFHRESP(NORMAL))
    MOVE 'READNEXT' TO ERROR-OPNAME
    MOVE 'UMTNAME ' TO ERROR-FILE
    PERFORM PROCESS-ERROR
  END-IF
END-IF.
```

```

* Se as sinalizações forem configuradas para ler o próximo registro de
conjunto de dados, execute isto
    IF (GET-NEXT-DS = 1 AND DS-EOF = 0)
        MOVE SPACES TO DS-REC
        EXEC CICS READNEXT FILE('SOURCEDS')
            RIDFLD(DS-KEY)
            INTO(DS-REC)
            RESP(RESPONSE)
            RESP2(RESPONSE2)
        END-EXEC
* Configure a sinalização EOF se o final do conjunto de dados foi atingido
    IF (RESPONSE = DFHRESP(ENDFILE))
        MOVE 1 TO DS-EOF
    ELSE
* Emita um erro se o código de retorno de READ for inesperado
        IF (RESPONSE NOT = DFHRESP(DUPKEY) AND
            RESPONSE NOT = DFHRESP(NORMAL))
            MOVE 'READNEXT' TO ERROR-OPNAME
            MOVE 'SOURCEDS' TO ERROR-FILE
            PERFORM PROCESS-ERROR
        END-IF
    END-IF.

READ-FILES-EXIT.
EXIT.
EJECT

*****
* Tentativa de excluir um registro da UMT *
*****
UMT-DELETE SECTION.

* Exclua o último registro de leitura na UMT
    EXEC CICS DELETE FILE('UMTNAME')
        RIDFLD(UMT-KEY)
        RESP(RESPONSE)
        RESP2(RESPONSE2)
    END-EXEC.
* Permita os códigos de retorno NORMAL e NOTFND, caso o registro tenha sido
* excluído desde sua primeira leitura, caso contrário, emita um erro
    IF (RESPONSE = DFHRESP(NORMAL) OR
        RESPONSE = DFHRESP(NOTFND))
* Configure as sinalizações para obter o próximo registro de UMT,
mas mantenha o mesmo registro de
conjunto de dados
        MOVE 1 TO GET-NEXT-UMT
        MOVE 0 TO GET-NEXT-DS
    ELSE
        MOVE 'DELETE ' TO ERROR-OPNAME
        MOVE 'UMTNAME ' TO ERROR-FILE
        PERFORM PROCESS-ERROR
    END-IF.

UMT-DELETE-EXIT.
EXIT.
EJECT

*****
* Tentativa de gravar um registro na UMT *
*****
UMT-WRITE SECTION.

* Tentativa de gravar o registro ausente usando a chave do conjunto de dados
    EXEC CICS WRITE FILE('UMTNAME')
        RIDFLD(DS-KEY)
        FROM(DS-REC)
        RESP(RESPONSE)

```

```

        RESP2(RESPONSE2)
    END-EXEC.
* Se a UMT teve um registro gravado nessa posição desde a
* leitura, exclua-o e tente outro pela última vez.
* Se a gravação for mal-sucedida novamente, vá para o próximo par de registros
    IF RESPONSE = DFHRESP(DUPREC)
        EXEC CICS DELETE FILE('UMTNAME')
            RIDFLD(DS-KEY)
            RESP(RESPONSE)
            RESP2(RESPONSE2)
        END-EXEC
        EXEC CICS WRITE FILE('UMTNAME')
            RIDFLD(DS-KEY)
            FROM(DS-REC)
            RESP(RESPONSE)
            RESP2(RESPONSE2)
        END-EXEC
    ELSE
        * Emita um erro se o código de retorno da primeira gravação for inválido
        * (mas permita um código de retorno de supressão por saída de usuário)
        IF (RESPONSE NOT = DFHRESP(NORMAL) AND
            RESPONSE NOT = DFHRESP(SUPPRESSED))
            MOVE 'UMTNAME ' TO ERROR-FILE
            MOVE 'WRITE ' TO ERROR-OPNAME
            PERFORM PROCESS-ERROR
        END-IF
    END-IF.

    * Configure as sinalizações para manter o mesmo
    registro de UMT e obtenha o próximo registro de
    conjunto de dados
    MOVE 0 TO GET-NEXT-UMT.
    MOVE 1 TO GET-NEXT-DS.

    UMT-WRITE-EXIT.
    EXIT.
    EJECT

*****
* Tentativa de atualizar um registro na UMT para corresponder ao DS *
*****
    UMT-UPDATE SECTION.

    * Tentativa de obter um bloqueio no registro usando leitura para atualização
    EXEC CICS READ FILE('UMTNAME')
        RIDFLD(UMT-KEY)
        INTO(UMT-REC)
        UPDATE
        RESP(RESPONSE)
        RESP2(RESPONSE2)
    END-EXEC.
    * Se o registro foi excluído desde a leitura original, grave-o.
    * Se a gravação for mal-sucedida, vá para o próximo par de registros
    IF RESPONSE = DFHRESP(NOTFND)
        EXEC CICS WRITE FILE('UMTNAME')
            RIDFLD(UMT-KEY)
            FROM(DS-REC)
            RESP(RESPONSE)
            RESP2(RESPONSE2)
        END-EXEC
    ELSE
        * Se a leitura for atualizada com sucesso, grava o registro do
        conjunto de dados na UMT
        IF RESPONSE = DFHRESP(NORMAL)
            EXEC CICS REWRITE FILE('UMTNAME')
                FROM(DS-REC)
                RESP(RESPONSE)

```

```

        RESP2(RESPONSE2)
    END-EXEC
* Emita um erro se a regravação falhou
    IF RESPONSE NOT = DFHRESP(NORMAL)
        MOVE 'REWRITE ' TO ERROR-OPNAME
        MOVE 'UMTNAME ' TO ERROR-FILE
        PERFORM PROCESS-ERROR
    END-IF
    ELSE
* Emita um erro se a leitura para atualização falhou
        MOVE 'READUPDT' TO ERROR-OPNAME
        MOVE 'UMTNAME ' TO ERROR-FILE
        PERFORM PROCESS-ERROR
    END-IF
    END-IF.

* Configure sinalizações para obter o próximo registro para a UMT
e o conjunto de dados
    MOVE 1 TO GET-NEXT-UMT.
    MOVE 1 TO GET-NEXT-DS.

    UMT-UPDATE-EXIT.
    EXIT.
    EJECT

*****
* Saia normalmente do programa *
*****
    TRAN-FINISH SECTION.

* Termine a operação de procura para a UMT
    IF (UMT-STARTBR = 1)
        EXEC CICS ENDBR FILE('UMTNAME')
            RESP(RESPONSE)
            RESP2(RESPONSE2)
        END-EXEC
    END-IF.

* Termine a operação de procura para o conjunto de dados
    IF (DS-STARTBR = 1)
        EXEC CICS ENDBR FILE('SOURCEDS')
            RESP(RESPONSE)
            RESP2(RESPONSE2)
        END-EXEC
    END-IF

* Emita uma mensagem na tela se a UMT foi atualizada
    IF (REM-FILE = 0)
        EXEC CICS SEND TEXT
            FROM(MESSAGE-OUTPUT)
            ERASE
            RESP(RESPONSE)
            RESP2(RESPONSE2)
        END-EXEC
    ELSE
* Emita uma mensagem se o arquivo foi definido como remoto
        EXEC CICS SEND TEXT
            FROM(REMOTE-OUTPUT)
            ERASE
            RESP(RESPONSE)
            RESP2(RESPONSE2)
        END-EXEC
    END-IF.

* Termine o programa e retorne ao CICS
    EXEC CICS RETURN

```

```

END-EXEC.

TRAN-FINISH-EXIT.
EXIT.
EJECT

*****
* Exiba uma mensagem de erro na tela e saia do programa *
*****
PROCESS-ERROR SECTION.

* Copie os últimos códigos de retorno na mensagem
  MOVE RESPONSE TO ERROR-RESP.
  MOVE RESPONSE2 TO ERROR-RESP2.

* Emita uma mensagem na tela
  EXEC CICS SEND TEXT
    FROM(ERROR-OUTPUT)
  ERASE
    RESP(RESPONSE)
    RESP2(RESPONSE2)
  END-EXEC.

* Termine o programa e retorne ao CICS
  EXEC CICS RETURN
  END-EXEC.

PROCESS-ERROR-EXIT.
EXIT.

```

Capítulo 6. Resolução de problemas de tabelas de dados

Use as informações de dump e rastreo que são produzidas pelo CICS para ajudá-lo a determinar a causa de um problema com tabelas de dados compartilhados.

Explicações das mensagens de diagnóstico e dos códigos de finalização anormal de tarefa produzidas pelas tabelas de dados compartilhados estão contidas em Mensagens CICS.

Informações de rastreo para serviços de tabela de dados

A tabela de rastreamento produzida pelo CICS ajuda a determinar a causa de um problema. Ela mostra o fluxo de controle por meio de módulos CICS. As entradas descritas aqui são incluídas na tabela de rastreamento pelos serviços de tabela de dados.

Para obter informações sobre o conteúdo da tabela de rastreamento e sobre como obtê-la, consulte Rastreo do CICS.

Existem dois tipos de pontos de rastreo:

- Pontos de rastreo de entrada e saída para cada um dos serviços fornecidos pelo suporte de tabela de dados compartilhados. O rastreo de nível de controle 2 deve ser ativado para obter esses pontos de rastreo.
- Pontos de rastreo de exceção.

Pontos de rastreo de entrada e saída para tabelas de dados compartilhados

Estes pontos de rastreo de entrada e saída são fornecidos pelos serviços de tabela de dados compartilhados.

0B13 Entrada para serviço Leitura Remota

0B14 Saída do serviço Leitura Remota

0B1B Entrada para serviço Inicializar Suporte de Tabela de Dados

0B1C Saída do serviço Inicializar Suporte de Tabela de Dados

0B1D Entrada para serviço Logon

0B1E Saída do serviço Logon

0B1F Entrada para serviço Carregar

0B20 Saída do serviço Carregar

0B21 Entrada para serviços Abrir, Fechar, Configurar Ativação e Estatísticas

0B22 Saída dos serviços Abrir, Fechar, Configurar Ativação e Estatísticas

0B23 Entrada para serviço Leitura Local

0B24 Entrada do serviço Leitura Local

0B25 Entrada para serviços Atualizar (incluir registro, incluir, substituir, excluir)

0B26 Entrada dos serviços Atualizar

0B2D Entrada para serviços Conectar e Desconectar

0B2E Saída dos serviços Conectar e Desconectar

O formato de cada um desses pontos de rastreio é descrito em Rastreio do CICS.

Sinalizações de função e qualificador para tabelas de dados compartilhados

Para tabelas de dados compartilhados, cada ponto de rastreio de entrada e saída contém um campo de função e a maioria deles contém um campo de sinalizações de qualificador. O campo de função é um byte que identifica a função que estava sendo executada; o campo de sinalizações de qualificador é um byte que contém sinalizações que qualificam algumas das funções.

Os valores para esses campos são:

Tabela 6. Valores e sinalizações de qualificador e função

Função	Sinalizações de qualificador
X'00' Inicializar	X'00' como servidor de tabela de dados compartilhados X'80' como solicitante da tabela de dados compartilhados
X'02' Incluir entrada da origem	X'00' inclusão emitida como resultado de um conjunto de dados para a solicitação de leitura da tabela X'40' inclusão emitida pela transação de carregamento
X'03' Gravar entrada na tabela	X'00' gravação concluída X'80' pré-gravação para CMT
X'04' Regravar entrada na tabela	X'00' regravação concluída X'80' pré-regravação para CMT
X'05' Excluir entrada na tabela	X'00' exclusão concluída X'80' pré-exclusão para CMT
X'06' Confirmar atualizações da tabela de dados mantida pelo usuário feitas por esta unidade de trabalho	
X'07' Recuperar atualizações da tabela de dados mantida pelo usuário feitas por esta unidade de trabalho	
X'08' Carregar tabela de dados (somente no rastreio de saída)	X'00' carregamento OK X'80' arquivo de origem está vazio

Tabela 6. Valores e sinalizações de qualificador e função (continuação)

Função	Sinalizações de qualificador
X'09' Apontar para um registro	X'80' correspondência igual X'40' maior que a correspondência X'20' menor que a correspondência (o que está acima pode ter várias combinações) X'10' testar se a tabela de dados está ativada
X'0A' recuperar o registro por chave	X'80' correspondência igual X'40' maior que a correspondência X'20' menor que a correspondência (o que está acima pode ter várias combinações) X'10' testar se a tabela de dados está ativada
X'0B' Recuperar registro por token	X'80' correspondência igual (atalho interno para uma sequência de registros) X'40' maior que a correspondência X'20' menor que a correspondência (o que está acima pode ter várias combinações) X'10' testar se a tabela de dados está ativada
X'0C' Efetuar logon como servidor	
X'0E' Abrir uma tabela de dados	
X'0F' Fechar uma tabela de dados	
X'10' Coletar estatísticas	
X'11' Configurar estado de ativação	X'00' ativar tabela de dados X'80' desativar tabela de dados X'40' forçar desativação (sempre combinado com desativar)
X'15' Conectar a uma tabela de dados compartilhados	
X'16' Interromper conexão com uma tabela de dados compartilhados	
X'17' Processar a conclusão do carregamento	

Códigos de resposta para tabelas de dados compartilhados

Cada ponto de rastreamento de saída para tabelas de dados compartilhados contém um campo de código de razão e código de resposta de dois bytes.

O primeiro byte é o código de resposta, para o qual os possíveis valores são:

X'01' Bem-sucedida

X'02' Exceção

X'03' Desastre

X'04' Inválidos

X'06' Eliminado

Códigos de razão para tabelas de dados compartilhados

Cada ponto de rastreamento de saída para tabelas de dados compartilhados contém um campo de código de razão e código de resposta de dois bytes.

O segundo byte é o código de razão, para o qual os valores possíveis são fornecidos abaixo. Esse código de razão pode estar acompanhado de informações de código de erro. O código de erro é um campo de quatro bytes que também é relatado em uma mensagem de erro ou ponto de rastreamento de exceção. Os valores possíveis estão descritos em Mensagens CICS, “Analisando erros das tabelas de dados SVC” na página 70 e “Analisando erros dos serviços de memória cruzada das tabelas de dados” na página 73.

X'01' Registro não está em uma tabela de dados

X'02' Duplicado (registro já está na tabela de dados)

X'03' Tabela de dados cheia (já contém o número máximo de registros)

X'04' Registro rejeitado pela saída de usuário

X'05' Falha ao obter armazenamento

X'06' Registro não está na tabela de dados (e sabe-se que a tabela está completa)

X'07' Falha no serviço da tabela de dados

X'08' Não autorizado a se conectar ao arquivo

X'09' Recurso não é uma tabela de dados

X'0A' Sistema remoto não efetuou login como um servidor

X'0B' Falha na solicitação de carregamento

X'0C' A tabela de dados está desativada

X'0D' Solicitação de inclusão (do DASD) deliberadamente não processada

X'0E' Registro muito longo

X'0F' Token de tabela de dados inválido

X'10' Registro não está na tabela de dados (mas pode estar no conjunto de dados de origem)

X'11' Tabela de dados não fechada, já que outros arquivos ainda estão usando-a

X'12' Reservado(a)

X'13' Registro está na tabela de dados, mas atualmente não é válido

X'14' Arquivo não pode ser fechado, já que está desativado

X'15' Erro do protocolo

X'16' CICS não está em um subsistema MVS

X'17' Não autorizado a se conectar a este arquivo

X'18' CICS não pode usar serviços de memória cruzada

X'19' Formato do bloco de parâmetros de interface não reconhecido

UMT e outras sinalizações para tabelas de dados compartilhados

Esse byte de sinalização está incluído no ponto de rastreamento de entrada em OPEN.

Os bits significativos no momento da abertura são:

B'1.....'

tabela de dados mantida pelo CICS

B'01.....'

Tabela de dados mantida pelo usuário recuperável

B'00.....'

Tabela de dados mantida pelo usuário não recuperável

Pontos de rastreamento de exceção para tabelas de dados compartilhados

Estes pontos de rastreamento de exceção são fornecidos pelos serviços da tabela de dados compartilhados.

AP 0B0A

Função não reconhecida na chamada para DFHDTRE

AP 0B0B

Função não reconhecida na chamada para DFHDTRR

AP 0B0C

Função não reconhecida na chamada para DFHDTUP

AP 0B0D

Função não reconhecida na chamada para DFHDTST

AP 0B0E

Função não reconhecida na chamada para DFHDTSS

AP 0B0F

Função não reconhecida na chamada para DFHDTRC

AP 0B10

Erro ao inicializar gerenciamento de registros

AP 0B11

Erro no gerenciador de registro OPEN

AP 0B12

Erro no gerenciador de registro CLOSE

AP 0B15

Erro inesperado ao chamar recuperação de PC

AP 0B19

Erro ao chamar tabelas de dados SVC ao inicializar como servidor

AP 0B1A

Erro ao chamar tabelas de dados SVC ao inicializar como solicitante

AP 0B27

CLOSE não pôde localizar bloco de tabela

AP 0B28

CLOSE não pôde localizar bloco de arquivo

AP 0B29

Erro ao chamar tabelas de dados SVC ao efetuar login como servidor

AP 0B2A

Erro ao chamar tabelas de dados SVC ao se conectar ou desconectar

AP 0B2B

A saída XDTRD retornou um comprimento de registro inválido (ou seja, mudou o comprimento para um CMT ou aumentou o comprimento para uma UMT)

AP 0B2C

Índice de conexão excede o tamanho máximo suportado

AP 0B2F

Erro desastroso ao adquirir armazenamento para passar parâmetros para a transação de carregamento

O formato de cada um desses pontos de rastreo é descrito em Rastreo do CICS.

Analizando erros das tabelas de dados SVC

Após um erro de uma chamada para as tabelas de dados SVC, um ponto de rastreo de exceção é sempre feito, incluindo um campo de código de erro para identificar o motivo do erro. Esses pontos de rastreo são AP 0B12, 0B19, 0B1A, 0B29 e 0B2A.

Há três categorias de erro SVC:

1. Condições que devem ocorrer, como o arquivo remoto em uma tentativa de conexão não ser uma tabela de dados ou o sistema remoto não ter efetuado logon como um servidor de tabelas de dados compartilhados. O CICS toma a ação apropriada para tais condições e nenhuma informação de diagnóstico é necessária.
2. Erros que poderiam ser causados por problemas no ambiente que pode ser possível corrigir. Para esses erros, uma mensagem é emitida com o código de razão para o erro. A explicação do código de razão está incluída na explicação da mensagem em Mensagens CICS.
3. Erros que indicam algum tipo de problema lógico, ou mau uso de rotinas, possivelmente em uma tentativa de contornar as verificações de integridade e segurança. Esses erros são tratados pelo controle de arquivos do CICS como erros desastrosos, resultando em um dump do sistema (se você tiver ativado tal dump) e, na maioria dos casos, na finalização anormal da transação com um AFCZ ABEND. Para eles, o valor do campo de resposta e razão normalmente é X'0215'.

Os tópicos a seguir explicam os códigos de erro para a terceira categoria de erros. Esses códigos de erro são vistos somente na entrada de rastreo de exceções. O formato do código de erro é X'ffaaaaaa', em que *ff* identifica o tipo de falha e *aaaaaa* são as informações adicionais fornecidas para algumas das falhas. Os valores possíveis de *ff* para cada ponto de rastreo estão descritos nos seguintes tópicos.

Valores para todos os pontos de rastreo de tabelas de dados compartilhados

Os códigos de erro a seguir podem ocorrer para os pontos de rastreo de exceção 0B12, 0B19, 0B1A, 0B29 e 0B2A.

X'01' Foi especificada uma função que requer que o responsável pela chamada seja autorizado via CICS AFCB (bloco autorizado de controle de função), mas o responsável pela chamada não foi autorizado.

- X'0A'** O responsável pela chamada passou um código de função inválido.
- X'0B'** O responsável pela chamada especificou um formato inválido de chamada SVC.
- X'0C'** Um endereço da lista de parâmetros inválido foi passado para o SVC.
- X'0D'** Foi especificada uma função que requer que o valor passado no registro 1 seja 0, mas ele não era. As informações adicionais contêm três bytes de baixa ordem do valor passado.
- X'12'** Foi especificada uma função que requer que o responsável pela chamada esteja no estado do supervisor Chave 0, mas o responsável pela chamada não estava.

Valores para ponto de rastreo 0B12

O ponto de rastreo de exceção AP 0B12 é emitido se um erro é retornado pelo SVC na inclusão ou exclusão de uma entrada da lista de acesso quando uma tabela de dados compartilhados está sendo fechada.

Além dos erros que podem ocorrer em todos os pontos de rastreo, os seguintes são possíveis:

- X'02'** Serviços de tabela de dados compartilhados ainda não foram inicializados (um bloco âncora para a região não foi criado).
- X'0E'** O espaço para dados especificado STOKEN é inválido ou o responsável pela chamada não está autorizado a usá-lo.
- X'0F'** A região CICS não concluiu a inicialização como um servidor.
- X'13'** Falha na tentativa de excluir uma entrada da lista de acesso porque a entrada especificada não foi criada pelas tabelas de dados SVC.

Todos os outros erros fazem com que seja emitida uma mensagem contendo o código de erro.

Valor para ponto de rastreo 0B19

O ponto de rastreo de exceção AP 0B19 é emitido se um erro é retornado pelo SVC na inicialização como um servidor de tabela de dados compartilhados.

Além dos erros que podem ocorrer em todos os pontos de rastreo, os seguintes são possíveis:

- X'02'** Está sendo feita uma tentativa de incluir uma entrada da lista de acesso antes de a região CICS ter executado a inicialização da tabela de dados compartilhados (um bloco âncora para a região ainda não foi criado).
- X'0E'** O espaço para dados especificado STOKEN é inválido ou o responsável pela chamada não está autorizado a usá-lo.
- X'0F'** Estava sendo feita uma tentativa de incluir uma entrada da lista de acesso antes de a região CICS ter concluído a inicialização do servidor.

Todos os outros erros fazem com que seja emitida uma mensagem contendo o código de erro.

Valores para ponto de rastreo 0B1A

O ponto de rastreo de exceção AP 0B1A será emitido se um erro for retornado pelo SVC na inicialização de um solicitante de tabela de dados compartilhados.

Além dos erros que podem ocorrer em todos os pontos de rastreo, o seguinte é possível:

X'05' A região CICS já foi inicializada como um solicitante de tabela de dados compartilhados, mas agora está em execução sob um bloco de solicitações diferente de quando foi inicializada originalmente.

Todos os outros erros fazem com que seja emitida uma mensagem contendo o código de erro.

Valores para ponto de rastreo AP 0B29

O ponto de rastreo de exceção AP 0B29 é emitido se um erro é retornado pelo SVC no logon como um servidor de tabela de dados compartilhados.

Além dos erros que podem ocorrer em todos os pontos de rastreo, os seguintes são possíveis:

X'02' A região CICS está tentando se registrar (efetuar logon) como um servidor, mas ainda não foi inicializada (um bloco âncora para a região não foi criado).

X'04' Esta região CICS já está registrada (efetuou logon) como um servidor de tabela de dados compartilhados.

X'0F' A região CICS não concluiu a inicialização do servidor.

X'14' O bloco âncora AFCS não existe.

X'15' O bloco de segurança CICS não existe.

X'16' Ou o responsável pela chamada não está em execução em uma chave de proteção do usuário (sua chave PSW é menor que 8) ou o TCB do responsável pela chamada normalmente não é executado em uma chave de proteção do usuário (TCBPKF é menor que 8).

Todos os outros erros fazem com que seja emitida uma mensagem contendo o código de erro.

Valores para ponto de rastreo 0B2A

Se o campo de código de função contiver X'15', o ponto de rastreo de exceção AP 0B2A indica um erro em CONNECT (ou seja, na tentativa de estabelecer uma conexão com um arquivo remoto).

Além dos erros que podem ocorrer em todos os pontos de rastreo, os seguintes são possíveis:

X'02' Serviços de tabela de dados compartilhados ainda não foram inicializados (um bloco âncora para a região não foi criado).

X'03' A região solicitante não concluiu a inicialização como um solicitante de tabelas de dados compartilhados.

X'05' A região CICS está em execução sob um bloco de solicitações (RB) diferente de quando foi inicializada como um solicitante de tabela de dados. A parte de informações adicionais do código de erro contém o endereço RB sob o qual a chamada foi feita.

X'72' O LINK para o módulo DFHACEE substituível pelo usuário para localizar o ID do usuário de segurança do espaço de endereço inicial falhou. A parte

de informações adicionais do código de erro contém dois bytes do código ABEND do LINK. O campo de resposta e razão que acompanha esse erro é X'020B'.

Todos os outros erros fazem com que seja emitida uma mensagem contendo o código de erro.

Se o campo de código de função contiver X'16', o ponto de rastreio de exceção 0B2A indicará um erro em DISCONNECT (ou seja, na tentativa de interromper a conexão com um arquivo remoto). Além dos erros que podem ocorrer em todos os pontos de rastreio, os seguintes são possíveis:

- X'02' Serviços de tabela de dados compartilhados ainda não foram inicializados (um bloco âncora para a região não foi criado).
- X'03' A região solicitante não concluiu a inicialização como um solicitante de tabelas de dados compartilhados.
- X'05' A região CICS está em execução sob um bloco de solicitações (RB) diferente de quando foi inicializada como um solicitante de tabela de dados. A parte de informações adicionais do código de erro contém o endereço RB sob o qual a chamada foi feita.
- X'07' O responsável pela chamada forneceu um índice inválido no vetor de conexões do arquivo. A parte de informações adicionais do código de erro contém os três bytes de baixa ordem do índice do responsável pela chamada.
- X'10' A conexão especificada foi interrompida anteriormente e não existe mais. A parte de informações adicionais do código de erro contém os três bytes de baixa ordem do índice do responsável pela chamada no vetor de conexões do arquivo.

Todos os outros erros fazem com que seja emitida uma mensagem contendo o código de erro.

Analizando erros dos serviços de memória cruzada das tabelas de dados

Após um erro inesperado dos serviços de memória cruzada das tabelas de dados, uma **entrada de rastreio de exceções X'0B15'** é feita. Isso inclui a resposta e os códigos de razão, além de um campo de código de erro identificando a causa do erro. Tais erros são todos causados pela distorção das rotinas ou do sistema ou por um possível mau uso das rotinas.

Para uma resposta e um código de erro de X'0215', o formato do código de erro é X'ffaaaaaa', em que *ff* identifica o tipo de falha e *aaaaaa* são informações adicionais fornecidas para algumas das falhas. Os valores possíveis de *ff* são:

- X'01' Uma falha na tentativa de localizar o CICS AFCB (bloco autorizado de controle de função) feito pela rotina de recuperação de memória cruzada ou rotina de consulta de vetor de conexão.
- X'02' A região CICS solicitante ainda não executou a inicialização das tabelas de dados compartilhados (um bloco âncora para a região ainda não foi criado e configurado).
- X'03' A região solicitante não concluiu a inicialização como um solicitante de tabelas de dados compartilhados.

- X'05' A solicitação de recuperação foi emitida sob um bloco de solicitações diferente daquele que executou a inicialização como um solicitante de tabela de dados compartilhados.
- X'06' A entrada do vetor de conexão para o arquivo remoto não contém o índice de ligação correto.
- X'07' O índice da entrada do vetor de conexão para o arquivo remoto está além do fim do vetor de conexão.
- X'08' A entrada do vetor de conexão para o arquivo remoto não está marcada como em uso.
- X'09' A rotina de recuperação de memória cruzada não foi chamada por meio do mecanismo correto.

Uma resposta e uma razão de X'0400' significam que o código de função passado para o código de gerenciamento de registros em execução na região de servidor era um valor desconhecido.

Informações de dump para tabelas de dados

Informações relevantes para tabelas de dados estão incluídas em um dump do sistema CICS para ajudá-lo a determinar a causa de um problema.

Para obter informações sobre o conteúdo de dumps e como obtê-los, consulte Usando Dumps na Determinação de Problema.

Os principais blocos de controle que são usados pela tabela de dados compartilhados estão incluídos na área FILE CONTROL de um dump formatado da região voltada para arquivos. Esses blocos de controle são denominados:

Área Global da Tabela de Dados

Também conhecida como *bloco de cabeçalho da tabela de dados compartilhados*, portanto, usa o destaque DFHDTHEADER.

Área Base da Tabela de Dados

Também conhecida como *bloco de tabelas de dados compartilhados*, portanto, usa o destaque DFHDTTABLE.

Área do Caminho da Tabela de Dados

Também conhecida como *bloco de arquivo da tabela de dados compartilhados*, portanto, usa o destaque DFHDTFILE.

O conteúdo da tabela de dados não está incluído no dump do sistema CICS, pois o armazenamento de espaço para dados no qual a tabela de dados reside não faz parte do espaço de endereço do CICS. As entradas de tabela residem no espaço para dados DFHDT001, os nós de índice em DFHDT002 e os dados de registros nos espaços para dados DFHDT003, DFHDT004, DFHDT005 e assim por diante, com novos espaços para dados sendo incluídos conforme necessário. Se desejar ver conteúdo da tabela de dados, peça ao operador do sistema para usar o comando **DUMP** do MVS para solicitar um dump do espaço para dados apropriado pertencente à tarefa de inicialização do CICS adequada.

O comando do operador `DISPLAY J,CICS-startup-jobname` mostra informações sobre uma tarefa do CICS, incluindo DSPNAMEs dos espaços para dados que ele possui. Para efetuar dump do conteúdo do espaço para dados DFHDT003, é possível usar o comando **DUMP** do MVS da seguinte forma:

1. Enter

DUMP COMM=(title for your dump)

2. Isso gera uma mensagem do console MVS
* id IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND
3. Responda à mensagem com
REPLY id, DSPNAME='jobname'.DFHDT003

e será feito dump do armazenamento de espaço para dados.

Nota: É possível, usando a seguinte notação de asterisco, efetuar dump do conteúdo de *todos* os espaços para dados pertencentes ao CICS:

REPLY id, DSPNAME='jobname'.DFHDT*

No entanto, isso deve ser usado com cuidado, pois se houver muitos espaços para dados, o conjunto de dados de dump ficará enorme.

4. Use DISPLAY DUMP,TITLE para ver qual conjunto de dados SYS1.DUMPnn foi usado.

Avisos

Estas informações foram desenvolvidas para produtos e serviços oferecidos nos Estados Unidos. Este material pode estar disponível por meio da IBM em outros idiomas. No entanto, talvez seja necessário ter uma cópia do produto ou da versão do produto naquele idioma para poder acessá-lo.

É possível que a IBM® não forneça os produtos, serviços ou recursos discutidos nesta publicação em outros países. Consulte um representante IBM local para obter informações sobre produtos e serviços disponíveis atualmente em sua área. Qualquer referência a produtos, programas ou serviços IBM não significa que apenas 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 usado. No entanto, é responsabilidade do usuário avaliar e verificar a operação de qualquer produto, programa ou serviço não IBM.

A IBM pode ter patentes ou solicitações de patentes pendentes relativas a assuntos tratados neste documento. O fornecimento desse documento não concede ao Cliente nenhuma licença para essas patentes. Pedidos de licença 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:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

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 A ELAS NÃO SE LIMITANDO, AS GARANTIAS IMPLÍCITAS DE NÃO INFRAÇÃO, COMERCIALIZAÇÃO OU ADEQUAÇÃO A UM DETERMINADO PROPÓSITO. Algumas jurisdições não permitem a exclusão de garantias expressas ou implícitas em certas transações; portanto, essa disposição pode não se aplicar ao Cliente.

Estas informações podem conter imprecisões técnicas ou erros tipográficos. São feitas alterações periódicas nas informações aqui contidas; tais alterações serão incorporadas em futuras edições desta publicação. A IBM pode, a qualquer momento, aperfeiçoar e/ou alterar os produtos e/ou programas descritos nesta publicação, sem aviso prévio.

Quaisquer referências nestas informações a websites não IBM são fornecidas apenas para conveniência e não representam de forma alguma um endosso a esses websites. Os materiais contidos nesses websites não fazem parte dos materiais deste produto IBM e o uso desses websites é de inteira responsabilidade do cliente.

A IBM pode utilizar ou distribuir as informações fornecidas da forma que julgar apropriada sem incorrer em qualquer obrigação para com o Cliente.

Licenciados deste programa que desejam obter informações sobre este assunto com 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

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 nesta publicação e todo o material licenciado disponível são fornecidos pela IBM sob os termos do Contrato com o Cliente IBM IBM, do Contrato Internacional de Licença do Programa IBM ou de qualquer outro contrato equivalente.

As informações relativas a produtos não IBM foram obtidas junto aos fornecedores dos respectivos produtos, de seus anúncios publicados ou de outras fontes disponíveis publicamente. A IBM não testou esses produtos e não pode confirmar a precisão de seu desempenho, compatibilidade nem quaisquer outras reivindicações relacionadas a produtos não IBM. Dúvidas sobre os recursos de produtos não IBM devem ser encaminhadas diretamente a seus fornecedores.

Estas informações contêm exemplos de dados e relatórios utilizados nas operações diárias de negócios. Para ilustrá-las da forma mais completa possível, os exemplos incluem os nomes de pessoas, empresas, marcas e produtos. Todos estes nomes são fictícios e qualquer semelhança com nomes e endereços usados por uma empresa real é mera coincidência.

COPYRIGHT LICENSE:

Estas informações contêm programas aplicativos de amostra no idioma de origem, que ilustram as técnicas de programação em diversas plataformas operacionais. O Cliente pode copiar, modificar e distribuir estes programas de amostra 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 amostra são criados. Esses exemplos não foram totalmente testados sob todas as condições. Portanto, a IBM não pode garantir ou implicar a confiabilidade, manutenção ou função destes programas. Os programas de exemplo são fornecidos "NO ESTADO EM QUE SE ENCONTRAM", sem garantia de nenhum tipo. A IBM não é responsável por nenhum dano decorrente do uso dos programas de amostra.

Informações sobre a interface de programação

O CICS fornece algumas documentações que podem ser consideradas como Interfaces de Programação e alguma documentação que não pode ser considerada como uma Interface de Programação.

As Interfaces de Programação que permitem ao cliente gravar programas para obter os serviços do CICS Transaction Server para z/OS, Versão 5 Release 4 estão incluídas nas seguintes seções da documentação do produto on-line:

- Desenvolvendo Aplicativos
- Desenvolvendo programas do sistema
- Visão geral da segurança do RACF
- Desenvolvendo para interfaces externas
- Referência: desenvolvimento de aplicativos
- Referência: programação do sistema
- Referência: conectividade

Informações que NÃO são destinadas ao uso como uma Interface de Programação do CICS Transaction Server para z/OS, Versão 5 Release 4, mas que podem ser interpretadas erroneamente como Interfaces de Programação, estão incluídas nas seções a seguir da documentação do produto on-line:

- Resoluções de Problemas e Suporte
- Referência: diagnósticos

Se você acessar a documentação do CICS em manuais no formato PDF, as Interfaces de Programação que permitem ao cliente gravar programas para obter os serviços do CICS Transaction Server para z/OS, Versão 5 Release 4 estão incluídas nos seguintes manuais:

- Guia de Programação de Aplicativos e Referência de Programação de Aplicativos
- Serviços de Transação de Negócios
- Guia de Customização
- Bibliotecas de Classe C++ OO
- Referência de Interfaces de Ferramentas de Depuração
- Guia de Programação de Transação Distribuída
- Guia de Interfaces Externas
- Guia de Interface de Programação de Front End
- Guia do IMS Database Control
- Guia de Instalação
- Guia de Segurança
- Transações Fornecidas
- CICSplex SM - Gerenciando Cargas de Trabalho
- CICSplex SM - Gerenciando o Uso de Recurso
- Guia de Programação de Aplicativos e Referência de Programação de Aplicativos do CICSplex SM
- Aplicativos Java no CICS

Se você acessar a documentação do CICS em manuais no formato PDF, as informações que NÃO são destinadas ao uso como uma Interface de Programação

do CICS Transaction Server para z/OS, Versão 5 Release 4 , mas que podem ser interpretadas erroneamente como Interfaces de Programação, estão incluídas nos manuais a seguir:

- Áreas de Dados
- Referência de Diagnóstico
- Guia para Determinação de Problemas
- CICSplex SM Problem Determination Guide

Marcas Registradas

IBM, o logotipo IBM e `ibm.com` são marcas comerciais ou marcas registradas da International Business Machines Corp., registradas em muitos países no mundo todo. Outros nomes de produtos e serviços podem ser marcas comerciais da IBM ou de outras empresas. Uma lista atual de marcas comerciais da IBM está disponível na web em Informações de Copyright e Marcas Comerciais, no endereço www.ibm.com/legal/copytrade.shtml.

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.

Intel, o logotipo Intel, Intel Inside, o logotipo Intel Inside, Intel Centrino, o logotipo Intel Centrino, Celeron, Intel Xeon, Intel SpeedStep, Itanium e Pentium são marcas ou marcas registradas da Intel Corporation ou de suas subsidiárias nos Estados Unidos e/ou em outros países.

Java e todas as marcas registradas e logotipos baseados em Java são marcas ou marcas registradas da Oracle e/ou suas afiliadas.

Linux é uma marca registrada de Linus Torvalds nos Estados Unidos e/ou em outros países.

Microsoft, Windows, Windows NT e o logotipo Windows são marcas registradas da Microsoft Corporation nos Estados Unidos e/ou em outros países.

UNIX é uma marca registrada do The Open Group nos Estados Unidos e em outros países.

Termos e condições para a documentação do produto

As permissões para uso destes documentos são concedidas de acordo com os termos e condições a seguir.

Aplicabilidade

Esses termos e condições são adicionais a quaisquer termos de uso para o website da IBM.

utilizar o Personal

O Cliente pode reproduzir essas publicações para seu uso pessoal, não comercial, desde que todos os avisos do proprietário sejam preservados. Você não pode distribuir, exibir ou fazer trabalho derivativo destas publicações ou qualquer parte delas, sem o consentimento expresso da IBM.

Uso comercial

É possível reproduzir, distribuir e exibir estas publicações somente dentro

de sua empresa, desde que todos os avisos do proprietário sejam preservados. Não é possível fazer trabalhos derivativos dessas publicações, nem reproduzir, distribuir ou exibir essas publicações, ou de qualquer parte delas fora de sua empresa, sem o consentimento expresso da IBM.

Direitos

Exceto quando concedido expressamente nesta permissão, nenhuma outra permissão, licença ou direito são concedidos, seja de maneira expressa ou implícita, para as publicações ou quaisquer informações, dados, software ou outra propriedade intelectual contida aqui.

A IBM reserva-se o direito de retirar as permissões concedidas aqui, sempre que, a seu critério, o uso das publicações for prejudicial ao seu interesse ou, conforme determinado pela IBM, as instruções anteriores que não estiverem sendo seguidas adequadamente.

Não é permitido fazer download, exportar ou exportar novamente estas informações, exceto em conformidade total com todos os regulamentos e leis aplicáveis, incluindo todos os regulamentos e leis de exportação dos Estados Unidos.

A IBM NÃO OFERECE NENHUMA GARANTIA QUANTO AO CONTEÚDO DESTAS PUBLICAÇÕES. ESTAS PUBLICAÇÕES SÃO FORNECIDAS "NO ESTADO EM QUE SE ENCONTRAM" E SEM QUAISQUER GARANTIAS DE QUALQUER TIPO, EXPRESSAS OU IMPLÍCITAS, INCLUINDO, MAS NÃO SE LIMITANDO A, GARANTIAS IMPLÍCITAS DE COMERCIALIZAÇÃO, NÃO INFRAÇÃO E ADEQUAÇÃO A UM DETERMINADO PROPÓSITO SÃO EXCLUÍDAS.

Declaração de privacidade on-line da IBM

Os produtos de Software IBM, incluindo soluções de software como serviço ("Ofertas de Software"), podem usar cookies ou outras tecnologias para coletar informações de uso do produto, para ajudar a melhorar a experiência do usuário final, para customizar interações com o usuário final ou para outros propósitos. Em muitos casos, nenhuma informação de identificação pessoal é coletada pelas Ofertas de Software. Algumas das Ofertas de Software podem ajudar a permitir a coleta de informações de identificação pessoal. Se esta Oferta de Software usar cookies para coletar informações pessoalmente identificáveis, informações específicas sobre o uso de cookies desta oferta serão descritas abaixo:

Para a Interface com o Usuário da Web do CICSplex SM (interface principal):

Dependendo das configurações implementadas, esta Oferta de Software pode usar cookies de sessão e persistentes que coletam o nome de usuário de cada usuário e outras informações pessoalmente identificáveis para propósitos de gerenciamento de sessão, autenticação, usabilidade do usuário aprimorada ou outros propósitos de rastreamento de uso ou funcionais. Esses cookies não podem ser desativados.

Para a Interface com o Usuário da Web do CICSplex SM (interface de dados):

Dependendo das configurações implementadas, esta Oferta de Software pode usar cookies de sessão que coletam o nome de usuário de cada usuário e outras informações pessoalmente identificáveis para propósitos de gerenciamento de sessão, autenticação ou outros propósitos de rastreamento de uso ou funcionais. Esses cookies não podem ser desativados.

Para o CICSplex SM Web User Interface (página do "hello world"):

Dependendo das configurações implementadas, esta Oferta de Software

pode usar cookies de sessão que não coletam informações pessoalmente identificáveis. Esses cookies não podem ser desativados.

Para o CICS Explorer:

Dependendo das configurações implementadas, esta Oferta de Software pode usar preferências de sessão e persistentes que coletam o nome do usuário e a senha de cada usuário, para os propósitos de gerenciamento de sessões, autenticação e configuração de conexão única. Essas preferências não podem ser desativadas, embora o armazenamento de uma senha de usuário em disco em formato criptografado somente possa ser ativado pela ação explícita do usuário ao marcar uma caixa de seleção durante a conexão.

Se as configurações implementadas para esta Oferta de Software fornecerem a você, como cliente, a capacidade de coletar informações pessoalmente identificáveis de usuários finais por meio de cookies e outras tecnologias, o Cliente deverá buscar seu próprio conselho jurídico sobre as leis aplicáveis a essa coleta de dados, incluindo quaisquer requisitos de aviso e consentimento.

Para obter mais informações sobre o uso de várias tecnologias, incluindo cookies, para esses propósitos, consulte Política de privacidade IBM e Declaração de privacidade on-line da IBM, a seção intitulada “Cookies, Web Beacons e Outras Tecnologias” e o Declaração de Privacidade de Produtos de Software e Software como Serviço IBM.

Índice Remissivo

A

- abrindo uma tabela de dados 47
- ambiente sysplex
 - atualizando UMTs replicadas 50
 - código do programa de exemplo 54
 - configuração e execução do programa de exemplo 54
 - introdução 4
 - operação do programa de exemplo 52
 - usando tabelas de dados compartilhados em 49
- AOR (regiões voltadas para aplicativos)
 - definição 3
 - operação CONNECT 7
- arquivo local
 - definição 3
- arquivo remoto
 - definição 3
- ativação de saídas de usuário 45
- atualizando UMTs replicadas 50
- atualizar solicitações
 - introdução 2

B

- benefícios
 - de tabelas de dados 9

C

- campo EIBRESP2 33, 37
- carregamento
 - uso de CMT durante 32
 - uso de UMT durante 34
- CEMT
 - comando INQUIRE 28, 29
 - comando SET 28, 29
- código de encerramento anormal AFCH 35, 36
- código de encerramento anormal AFCZ 70
- códigos de encerramento anormal AFCH 35, 36
- AFCZ 70
- códigos de razão
 - em pontos de rastreio 68
- códigos de resposta
 - em pontos de rastreio 67
- comando CEDA DEFINE FILE
 - descrição 24
 - exemplo para CMT 25
 - exemplo para UMT 26
 - parâmetro LOG 24
 - parâmetro MAXNUMRECS 24
 - parâmetro OPENTIME 24
 - parâmetro RECORDFORMAT 24
 - parâmetro TABLE 24
- comando INQUIRE FILE
 - descrição 28, 29

- comando INQUIRE FILE (*continuação*)
 - parâmetro MAXNUMRECS 28, 29
 - parâmetro TABLE 28, 29
- comando SET FILE
 - descrição 28, 29
 - parâmetro MAXNUMRECS 28, 29
 - parâmetro TABLE 28, 29
- compartilhamento
 - em um sysplex 4
 - environment 3
 - operação CONNECT 7
 - operação LOGON 7
 - operações da tabela de dados compartilhados 6, 7
- Compartilhamento no Nível do Registro 2
- comprimento da chave
 - comparação com remessa de função 37
- comunicação
 - entre o CICS e saídas de usuário 39
- conceitos de tabelas de dados 1
- condição INVREQ 33
- condição LOADING 34
- condição NOSPACE 33
- condição NOTFND 33, 34
- condição SUPPRESSED 33
- CONEXÃO
 - por AOR 7
 - verificação de segurança 19
- conjunto de dados de origem
 - com vários arquivos 21
 - deve ser KSDS 4
 - independente de UMT 3
 - para tabelas de dados 2
 - usado com CMT 2
- CONNECT
 - por AOR 35
- considerações de MVS 12, 19
- controle de arquivo
 - saídas do usuário 39
- controle de arquivos
 - comandos
 - suportados pelos serviços de memória cruzada 31
 - visão geral para a CMT 31
 - visão geral para UMT 33
- copybook DFHXDTS 39
- customização por saídas de usuário 39

D

- daisy chain 35
- definição do recurso
 - comando DEFINE FILE 24
 - descrição 21
 - visão geral para uma CMT 21
 - visão geral para uma UMT 22
- desativando uma tabela de dados 36
- desconexão
 - de AOR e tabela de dados 8, 35

- desempenho
 - benefícios de tabelas de dados 9
 - de um CMT 9
 - de uma UMT 9
- determinação de problema para tabelas de dados 65
- DFHDTVCV 19
- DFHDT SVC 19
- DFH MVRMS 19
- disponibilidade de tabelas de dados 1
- DSECT
 - para lista de parâmetros de saída de usuário 39

E

- erros SVC 70
- espaço para dados
 - dump de conteúdos 74
 - uso por tabelas de dados 9
- espaços de dados
 - uso por tabelas de dados 4
- estado inicial da tabela de dados
 - definindo pelo CEDA 24
- estatística
 - para selecionar tabelas de dados 14
- externa
 - para saídas de usuário 39
 - programação sensível ao produto 39

F

- fechando uma tabela de dados 36, 48
- filas de dados temporários
 - usadas para mensagens 47
- file
 - usado como uma tabela de dados 1, 12
- FOR (região voltada para arquivos)
 - definição 3
 - operação LOGON 7
- Função
 - para pontos de rastreio 66

G

- gap 10, 31, 36
- gerenciamento de arquivo
 - usando remessa de função 6
 - usando serviços de memória cruzada 6
- Global Resource Serialization, consulte GRS 51
- GRS (Global Resource Serialization) 51

I

- índices alternativos 4, 21
- informações de dump do sistema 74

informações de dump para tabelas de dados 74

informações de rastreo
códigos de razão 68
códigos de resposta 67
função e sinalizações de qualificador 66
para tabelas de dados 65
pontos de entrada e saída 65
pontos de exceção 69

instalação
considerações de MVS 19
lista de parâmetros 18

integridade
de dados da CMT 22
de dados da UMT 23

integridade de dados
de um CMT 22
de uma UMT 23

interface de programação sensível ao produto 39

interface exec
saídas do usuário 39

K

KSDS (conjunto de dados de chave em sequência)
com uma UMT 22
usado como um conjunto de dados de origem 2, 4

L

lista de parâmetros
para saídas de usuário 39
LOGON
pela FOR 7
verificação de segurança 18

M

mensagens
no final do carregamento 47
no início do carregamento 47
módulos de carregamento
necessários para tabelas de dados 20

N

notificação
para operações CONNECT 8

O

operações para tabelas de dados 47

P

parâmetro INSTLN 18
parâmetro SYSID 34
planejamento para tabelas de dados 9, 19

programação de aplicativos
para um CMT
descrição 31
para uma UMT
descrição 33
visão geral 33
programação do aplicativo
extensões para tabelas de dados compartilhados 1

R

recuperação de tabelas de dados
definindo pelo CEDA 24
durante uma reinicialização
emergencial 48
recurso de autorização do sistema 18
registro no diário 22
registro no diário automático 22, 23
Resource Access Control Facility
usado como gerenciador de segurança 18
restauração de transação dinâmica 23, 24
RLS (Record Level Sharing) 2

S

SAF, System Authorization Facility
usado para verificação de segurança 18
Saída de usuário XDTAD
descrição 44
Saída de usuário XDTLC
descrição 45
Saída de usuário XDTRD
descrição 43
saídas de usuário
amostras de programa de saída 39
descrição 39
para controle de arquivos 36
para interface exec 36
saídas do usuário
ao incluir registros 44
ativando 45
comunicação com CICS 39
definição 45
DSECT para lista de parâmetros 39
durante o carregamento 43
habilitando 45
lista de parâmetros 39
no final do carregamento 45
para controle de arquivos 39
para interface exec 39
saída XDTAD 44
saída XDTLC 45
saída XDTRD 43
visão geral 5
segurança BIND 19
segurança do arquivo 19
selecionando arquivos
para usar como tabelas de dados 12
serviços de memória cruzada
analisando erros 73
comandos suportados 31

serviços de memória cruzada
(*continuação*)
comparação com remessa de função 6, 36
uso por aplicativo 34
vantagens 1

servidor
definição 3
SHAREOPTION, VSAM 22
sinalizações de qualificador
para pontos de rastreo 66
solicitações de atualização
para um CMT 31, 32
para uma UMT 33
solicitações de exclusão
comparação com VSAM 38
solicitações de leitura
comparação com remessa de função 37
comparação com VSAM 37
introdução 2
para um CMT 31, 32
para uma UMT 33
solicitações de procura
comparação com remessa de função 36
comparação com VSAM 38
para um CMT 31
para uma UMT 33
solicitante
definição 3

T

tabela de dados mantida pelo CICS
definição do recurso 21
descrição 2
desempenho 9
integridade de dados 22
registro no diário 22
visão geral 2
tabela de dados mantida pelo usuário
atualizar solicitações 33
definição do recurso 21
descrição 3
desempenho 9
integridade de dados 23
registro no diário 23
replicação em um sysplex 49
solicitações de leitura 33
solicitações de procura 33
uso durante carregamento 34
visão geral 3
Tabela de dados mantidos do CICS
atualizar solicitações 31, 32
solicitações de leitura 31, 32
solicitações de procura 31
uso durante carregamento 32
tamanho da tabela de dados
definindo pelo CEDA 24
definindo pelo comando SET 28, 29
localizando pelo comando INQUIRE 28, 29
tipo de tabela de dados
definindo pelo CEDA 24
definindo pelo comando SET 28, 29

tipo de tabela de dados (*continuação*)
localizando pelo comando
INQUIRE 28, 29
transação CFTL 47
transação CSFU 47, 48

U

uso de armazenamento
descrição 9

V

vários arquivos
com o mesmo conjunto de dados de
origem 21
verificação de segurança
comparação com remessa de
função 18, 36
considerações sobre RACF 18
na conexão da AOR 19
no logon da FOR 18
para tabelas de dados 18
uso de SAF 18
visão geral de tabelas de dados
compartilhados 1
VSAM
bloco de controle de método de
acesso 47, 48
cluster de base 21
comparação com tabelas de dados 37
índices alternativos 4, 21
SHAREOPTION 22
VSAM RLS
adequado para UMT 3, 22
atributos de recuperação 24
inadequado para CMT 2, 21
para compartilhamento de dados 14

