

Usando disparadores do base de dados com Blended Agent para fazer o processamento característico da aplicação

Índice

[Introdução](#)

[Pré-requisitos](#)

[Requisitos](#)

[Componentes Utilizados](#)

[Convenções](#)

[Visão geral](#)

[Dialing List e Personal Callback List](#)

[Exemplo de disparadores](#)

[Introduza uma fileira em um base de dados externo](#)

[Atualize a informação de contato em um base de dados externo](#)

[Suprima de uma fileira de um base de dados externo](#)

[Execute um executável externo de um disparador](#)

[Considerações de desempenho](#)

[Práticas recomendadas](#)

[Informações Relacionadas](#)

[Introdução](#)

Este White Paper descreve como usar disparadores do base de dados do servidor SQL para executar o processamento característico da aplicação, quando os registros de contato no base de dados privado do Blended Agent (BA) são fechados ou atualizados pelo gerenciador de campanha BA.

[Pré-requisitos](#)

[Requisitos](#)

A Cisco recomenda que você tenha conhecimento destes tópicos:

- Cisco Intelligent Contact Management (ICM)
- Blended Agent do Software Cisco ICM
- Base de dados Microsoft SQL

[Componentes Utilizados](#)

A informação neste White Paper é baseada nestes versão de software e hardware:

- Versão do ICM 4.6.x de Cisco e mais tarde
- SQL 7 de Microsoft

A informação neste White Paper foi criada dos dispositivos em um ambiente de laboratório específico. Todos os dispositivos usados neste White Paper começaram com uma configuração esclarecida (PADRÃO). Se a sua rede estiver ativa, certifique-se de que entende o impacto potencial de qualquer comando.

Convenções

Consulte as [Convenções de Dicas Técnicas da Cisco](#) para obter mais informações sobre convenções de documentos.

Visão geral

O BA armazena os registros de contato a ser discados em um base de dados do Microsoft SQL server. O uso de um aplicativo de base de dados padrão, tal como o servidor SQL, facilita para que terceiras partes e os usuários integrem seus sistemas e aplicativos com o BA.

Este White Paper descreve como usar disparadores do base de dados do servidor SQL para executar o processamento característico da aplicação, quando o gerenciador de campanha BA fecha ou atualiza o estado de alguns conecta o registro no base de dados.

Dialing_List e Personal_Callback_List

O BA armazena os registros de contato na tabela do **Dialing_List**. Armazena os registros pessoais da rechamada na tabela do **Personal_Callback_List**. Estas tabelas estão no base de dados privado BA (*customer_inst_baA*). Nesta tabela, o CallStatus e o CallResult identificam o estado de um registro de contato.

Nome	Tipo	Tamanho	Descrição
Call Status	CARVÃO ANIMAL	1 byte	O CallStatus indica o estado do registro: p — O registro de contato é PENDENTE ser chamado. c — O registro foi fechado. Refira estes documentos para uma descrição completa de todos os valores do CallStatus e de seus significados: <ul style="list-style-type: none">• Guia de definição e configuração do Blended Agent do Software Cisco ICM (versão do ICM 4.6.x de Cisco)• Guia do Usuário do Blended Agent do Software Cisco ICM (versão do ICM 5 de Cisco)
Call	SMA	2	O CallResult indica o resultado do

Result	LLINT	bytes	<p>atendimento como retornado ao gerenciador de campanha BA pelo discador BA. Assim, indica se a VOZ dos indicadores do discador, OCUPADO, ou SEM RESPOSTA quando o número do contato é discado. Refira estes documentos para descrições completas de todos os valores do CallResult e de seus significados:</p> <ul style="list-style-type: none"> • Guia de definição e configuração do Blended Agent do Software Cisco ICM (versão do ICM 4.6.x de Cisco) • Guia do Usuário do Blended Agent do Software Cisco ICM (versão do ICM 5 de Cisco)
--------	-------	-------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Sempre que o gerenciador de campanha BA atualiza um registro no **Dialing_List** ou no **Personal_Callback_List**, os campos do CallStatus e do CallResult são atualizados. Por este motivo, é simples criar disparadores do base de dados do servidor SQL para monitorar estes campos.

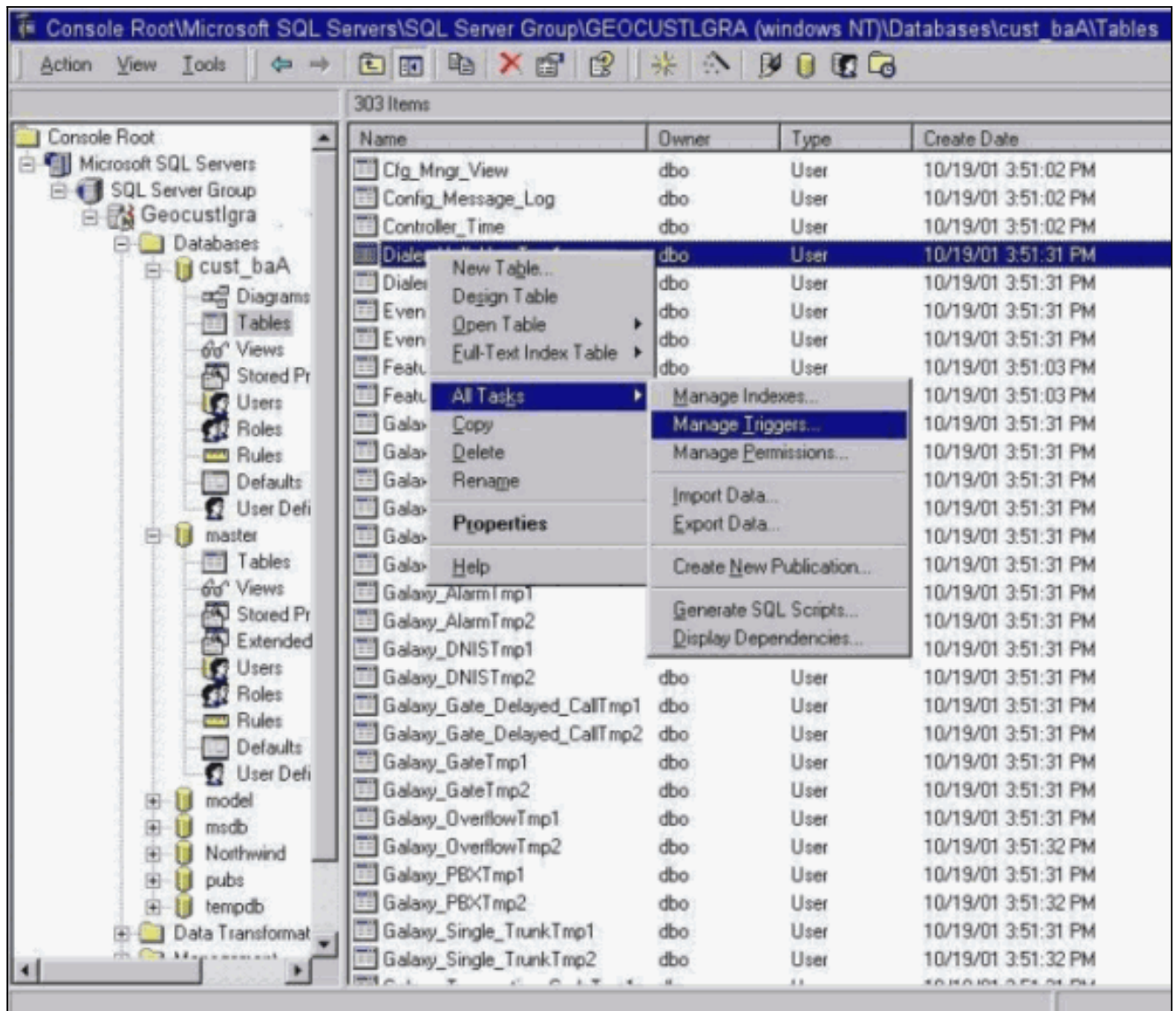
O uso dos disparadores nas tabelas de base de dados BA permite que você realize um número ilimitado de objetivos característicos da aplicação. Este White Paper demonstra alguns exemplos de cenário onde os disparadores podem ser úteis.

[Exemplo de disparadores](#)

Nota: Todos os exemplos de disparador neste White Paper referem uma tabela fictícia, o **CRM_Table** no base de dados **CRM_DB**.

Termine estas etapas para criar disparadores da enterprise manager do servidor SQL:

1. Detalhe ao base de dados BA.
2. Sob o base de dados, clique o ícone da **tabela**. Isto indica as tabelas que são associadas com o base de dados.
3. Selecione a tabela em que você quer criar um disparador.
4. Clicar com o botão direito a tabela selecionada e selecione o **todas as tarefas > gerenciar disparadores**.



[Introduza uma fileira em um base de dados externo](#)

O exemplo de disparador nesta seção introduz uma fileira na tabela, **CRM_Table** (no base de dados **CRM_DB**) sempre que o gerenciador de campanha BA atualiza um registro de contato no **Dialing_List**.

CRIE o insert_crm_data do DISPARADOR

```
CREATE TRIGGER insert_crm_data ON Dialing_List FOR
UPDATE AS
/* Proceed only if EXACTLY ONE Record is updated */
!--- See next Note for an explanation. IF (SELECT COUNT
(*) FROM INSERTED) = 1 BEGIN INSERT CRM_DB..CRM_Table
SELECT inserted.AccountNumber, GETDATE (),
inserted.CallStatus FROM inserted END
```

Nota: A não ser registros de contato da atualização quando os contatos são chamados, o gerenciador de campanha igualmente executa algumas atualizações maiorias. Estes devem ser filtrados para fora pelo disparador. Veja a seção das [práticas recomendadas](#) deste White Paper para mais informação.

[Atualize a informação de contato em um base de dados externo](#)

O exemplo de disparador nesta seção atualiza um registro na tabela **CRM_Table** (no base de dados **CRM_DB**) através de um fósforo do account number do contato. Por exemplo, se o gerenciador de campanha BA atualiza o registro no **Dialing_List** com account number 1A2B3C4, a seguir o registro em **CRM_Table** com o mesmo account number é atualizado simultaneamente.

CRIE o updt_customer_by_acctnum do DISPARADOR

```
CREATE TRIGGER updt_customer_by_acctnum ON Dialing_List
FOR UPDATE AS
/* Proceed only if EXACTLY ONE Record is updated */
IF (SELECT COUNT (*) FROM INSERTED) = 1
BEGIN

    DECLARE @CS CHAR (1)
    DECLARE @ACCTNUM VARCHAR (25)
    SELECT
    @CS = inserted.CallStatus,
    @ACCTNUM = inserted.AccountNumber
    FROM inserted
    UPDATE Northwind..CRM_Table
    SET Status = @CS
    Where Northwind..CRM_Table.AccountNumber = @ACCTNUM

END
```

[Suprima de uma fileira de um base de dados externo](#)

O exemplo de disparador nesta seção suprime de um registro da tabela **CRM_Table** (no base de dados **CRM_DB**) através de um fósforo do account number do contato, se esse registro de contato foi fechado no **Dialing_List**. Por exemplo, se o gerenciador de campanha BA fecha o registro no **Dialing_List** com account number 1A2B3C4, a seguir o registro em **CRM_Table** com o mesmo account number é suprimido.

CRIE o del_customer_by_acctnum do DISPARADOR

```
CREATE TRIGGER del_customer_by_acctnum ON Dialing_List
FOR UPDATE AS
/* Proceed only if EXACTLY ONE Record is updated */
IF (SELECT COUNT (*) FROM INSERTED) = 1
BEGIN

    DECLARE @CS CHAR (1)
    DECLARE @ACCTNUM VARCHAR (25)
    SELECT
    @CS = inserted.CallStatus,
    @ACCTNUM = inserted.AccountNumber
    FROM inserted
    IF @CS = 'C'
    BEGIN
        DELETE Northwind..CRM_Table
        WHERE Northwind..CRM_Table.AccountNumber =
@ACCTNUM
    END

END
```

[Execute um executável externo de um disparador](#)

O exemplo de disparador nesta seção executa o programa **C:\MYAPP.EXE**. O disparador passa o AccountNumber e o CallStatus da linha de comando updated record as argumentos ao

MYAPP.EXE. Isto ocorre sempre que o gerenciador de campanha BA atualiza o estado de exatamente um registro.

CRIE O DISPARADOR `run_executable`

```
CREATE TRIGGER run_executable ON Dialing_List FOR UPDATE
AS
/* Proceed only if EXACTLY ONE Record is updated */
IF (SELECT COUNT (*) FROM INSERTED) = 1
BEGIN
    DECLARE @CMD VARCHAR (100)
    SELECT
        @CMD = 'C:\MYAPP.EXE ' + inserted.AccountNumber + '
' + inserted.CallStatus
    FROM inserted
    EXEC master ..xp_cmdshell @CMD
END
```

Nota: Na tabela precedente, um pouco do que chama o procedimento armazenado do `xp_cmdshell` para começar um executável, o disparador pode igualmente executar todo o outro procedimento armazenado.

Considerações de desempenho

- Os disparadores do servidor SQL são executados *synchronously*. Isto significa se há um disparador da atualização em uma tabela, um comando `update` nessa tabela não retorna até que a execução do disparador esteja terminada. Se os disparadores não são projetados otimamente, a seguir as atualizações da base de dados BA estão retardadas. Também, o gerenciador de campanha BA pôde encontrar intervalos da conectividade de bases de dados aberto (ODBC) quando executar instruções SQL para atualizar as tabelas. Geralmente, um disparador deve fazer o processamento possível mínimo e deve ser aperfeiçoado para a velocidade de execução.
- O resultado da execução do disparador pôde impactar o resultado da operação SQL original. Por exemplo, supõe que um disparador da atualização está definido no **Dialing_List**. Se o disparador da atualização falha, a seguir a operação da atualização no **Dialing_List** pôde igualmente falhar e o registro na pergunta pode ser deixado no **Dialing_List** com um `CallStatus` incorreto.
- Se um disparador é aninhado, a seguir a operação de SQL não termina até que todos os disparadores aninhados estejam executados. Consequentemente, a falha de todo o disparador pôde falhar a operação SQL original.
- Se um disparador executa um aplicativo externo ou um outro procedimento armazenado, a seguir a execução do disparador não está completa até o aplicativo externo ou o procedimento armazenado é terminado.
- O uso dos cursores nos disparadores não é recomendado, devido potencialmente ao impacto negativo no desempenho.
- Os disparadores não devem executar o CPU ou as tarefas da memória intensa.
- Todos os disparadores devem bem ser testados e forçado antes do desenvolvimento.

Práticas recomendadas

- Verifique sempre que somente um registro está atualizado antes que você continue. Isto

impede que o disparador atue nas atualizações maiorias feitas pelo gerenciador de campanha. Estes são exemplos das atualizações maiorias que são filtradas geralmente para fora pelo disparador: Uma mudança no CallStatus de um registro de **A** (ATIVO) a **U** (DESCONHECIDO), quando um discador com disconexões ATIVAS dos registros do gerenciador de campanha. O gerenciador de campanha atualiza periodicamente registros no CallStatus **U a P** (DURANTE). **Nota:** Quando um registro de contato é atualizado em consequência de um atendimento, os registros estão atualizados sempre um de cada vez. Conseqüentemente, a verificação para ver se exatamente um registro era proteções actualizadas contra ações indesejáveis do disparador na maioria das vezes. Uma exceção a este caso ocorre quando uma operação do volume atualiza somente uma fileira.

- Verificação para o valor do CallStatus antes que você continuar. Por exemplo, quando o gerenciador de campanha envia um registro ao discador BA, o gerenciador de campanha ajusta o CallStatus a **A** (ATIVO). Um disparador pode verificar para ver se há o CallStatus. Se o CallStatus actualizado é **A**, não continua promover.
- A fim aperfeiçoar o desempenho durante atualizações da base de dados BA, o tempo e as operações dos recursos intensivos não devem ser executados no disparador. Você pode criar uma tabela específica de aplicativo em um outro base de dados para servir como uma fila. Isto acelera a execução do disparador. Você pode usar o disparador para introduzir, atualizar, ou suprimir de dados nessa tabela específica de aplicativo. Então, um outro aplicativo pode votar que tabela para mudanças e executa umas tarefas mais adicionais como necessárias.

[Informações Relacionadas](#)

- [Guia de definição e configuração do Blended Agent do Software Cisco ICM](#)
- [Guia do Usuário do Blended Agent do Software Cisco ICM](#)
- [Suporte Técnico e Documentação - Cisco Systems](#)