

# Использование триггеров базы данных вместе с Blended Agent для специализированной обработки

## Содержание

[Введение](#)

[Предварительные условия](#)

[Требования](#)

[Используемые компоненты](#)

[Условные обозначения](#)

[Обзор](#)

[Список дозвона и Список персональных обратных вызовов](#)

[Типовые триггеры](#)

[Вставьте строку во внешнюю базу данных](#)

[Контактная информация обновления во внешней базе данных](#)

[Удалите строку из внешней базы данных](#)

[Выполните внешний исполняемый файл от триггера](#)

[Вопросы производительности](#)

[Рекомендуемые методы](#)

[Дополнительные сведения](#)

## **Введение**

Это Описание технологических решений описывает, как использовать триггеры базы данных Сервера SQL для выполнения специализированной обработки, когда записи контакта в частной базе данных Blended Agent (BA) закрыты или обновлены Менеджером кампании BA.

## **Предварительные условия**

### **Требования**

Компания Cisco рекомендует предварительно ознакомиться со следующими предметами:

- Cisco Intelligent Contact Management (ICM)
- Blended Agent программного обеспечения Cisco ICM
- База данных Microsoft SQL

### **Используемые компоненты**

Информация в этом Описании технологических решений основывается на этих версиях программного и аппаратного обеспечения:

- Cisco ICM Version 4.6.x и позже
- Microsoft SQL 7

Информация в этом Описании технологических решений была создана от устройств в специальной лабораторной среде. Все устройства, используемые в этом Описании технологических решений, запустили с чистой (заданной по умолчанию) конфигурацией. Если используемая сеть является действующей, убедитесь в понимании возможного влияния любой из применяемых команд.

## Условные обозначения

[Дополнительные сведения об условных обозначениях см. в документе Условные обозначения технических терминов Cisco.](#)

## Обзор

BA хранит записи контакта, которые будут набраны в базе данных Microsoft SQL Server. Использование приложения стандартной базы данных, такого как Сервер SQL, упрощает для третьих сторон и пользователей интегрировать их системы и приложения с BA.

Это Описание технологических решений описывает, как использовать триггеры базы данных Сервера SQL для выполнения специализированной обработки, когда Менеджер кампании BA закрывает или обновляет статус любой записи подключения в базе данных.

## Список дозвона и Список персональных обратных вызовов

BA хранит записи контакта в таблице **Dialing\_List**. Это хранит записи персонального обратного вызова в таблице **Personal\_Callback\_List**. Эти таблицы находятся в частной базе данных BA (*customer\_inst\_baA*). В этой таблице CallStatus и CallResult определяют состояние записи контакта.

Имя	Вводите	Размер	Описание
Call Status	CHAR	1 байт	CallStatus указывает на состояние записи: P — запись контакта находится на рассмотрении, чтобы быть вызванной. C — запись была закрыта. См. эти документы для полного описания всех значений CallStatus и их значений: <ul style="list-style-type: none"><li>• <a href="#">Руководство по установке и конфигурации Blended Agent Программного обеспечения Cisco ICM</a> (Cisco ICM Version 4.6. x</li><li>• <a href="#">Руководство пользователя Blended Agent Программного</a></li></ul>

			<a href="#">обеспечения Cisco ICM</a> (Cisco ICM Version 5)
Call Result	SMALLINT	2 байта	<p>CallResult указывает на результат вызова, как возвращено Менеджеру компании ВА номеронабирателем ВА. Таким образом это указывает, отображает ли номеронабиратель VOICE, ЗАНЯТЫЙ, или НИКАКОЙ ОТВЕТ, когда набран контактный номер. См. эти документы для полных описаний всех значений CallResult и их значений:</p> <ul style="list-style-type: none"> <li>• <a href="#">Руководство по установке и конфигурации Blended Agent Программного обеспечения Cisco ICM</a> (Cisco ICM Version 4.6. x</li> <li>• <a href="#">Руководство пользователя Blended Agent Программного обеспечения Cisco ICM</a> (Cisco ICM Version 5)</li> </ul>

Каждый раз, когда Менеджер компании ВА обновляет запись в **Dialing\_List** или в **Personal\_Callback\_List**, поля CallStatus и CallResult обновлены. Поэтому просто создать триггеры базы данных Сервера SQL для мониторинга этих полей.

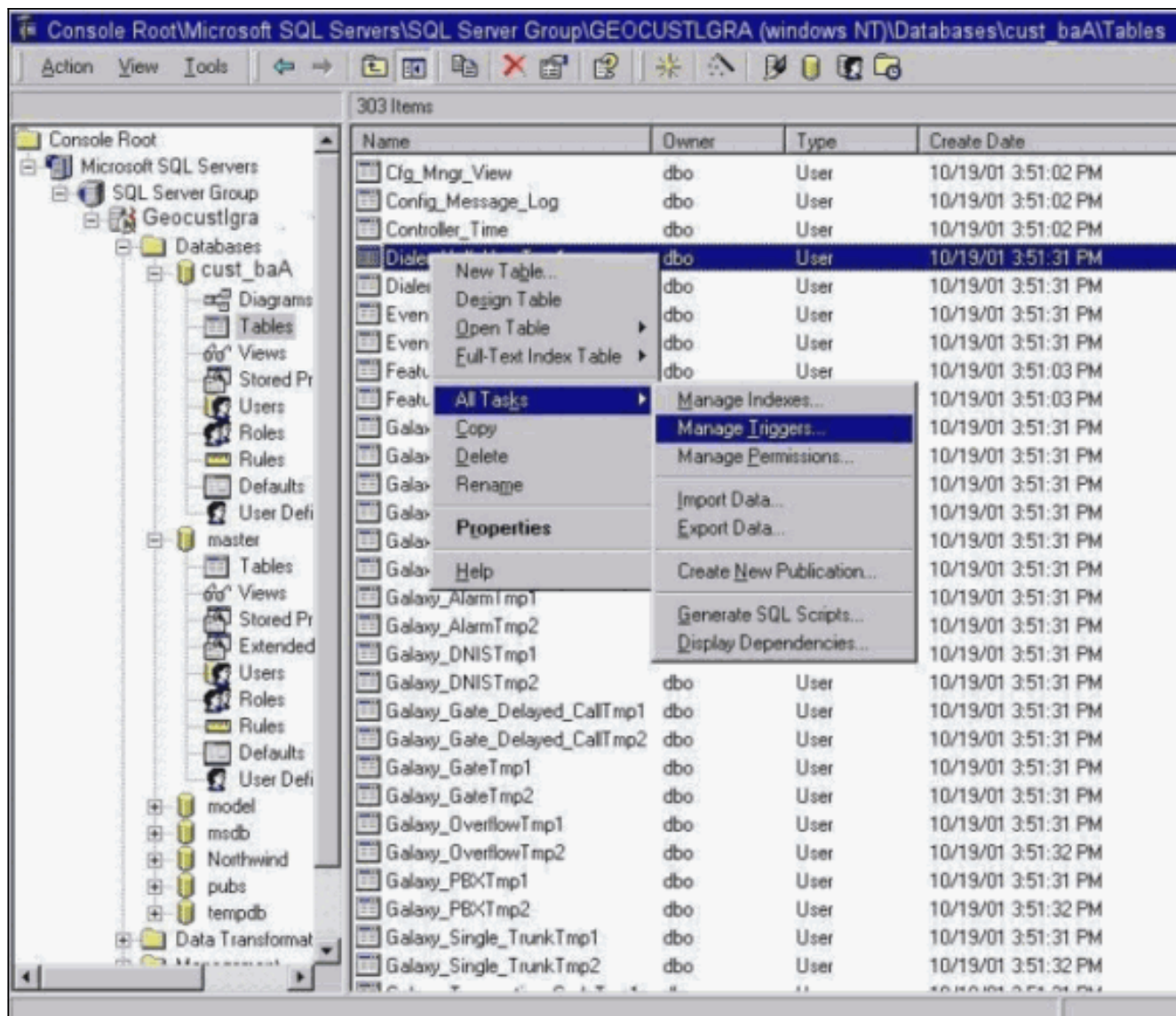
Использование включает таблицы базы данных ВА, позволяет вам выполнять неограниченное количество специализированных целей. Это Описание технологических решений демонстрирует некоторые примеры сценария, где триггеры могут быть полезными.

## [Типовые триггеры](#)

**Примечание:** Все примеры триггера в этом Описании технологических решений обращаются к фиктивной таблице, **CRM\_Table** в базе данных **CRM\_DB**.

Выполните эти шаги для создания триггеров от SQL Server Enterprise Manager:

1. Перейдите к базе данных ВА.
2. Под базой данных нажмите **Значок таблицы**. Это отображает таблицы, которые привязаны к базе данных.
3. Выберите таблицу, на которой вы хотите создать триггер.
4. Щелкните правой кнопкой мыши выбранную таблицу и выберите **All Tasks> Manage Triggers**.



## [Вставьте строку во внешнюю базу данных](#)

Пример триггера в этом разделе вставляет строку в таблицу, **CRM\_Table** (в базе данных **CRM\_DB**) каждый раз, когда Менеджер компании ВА обновляет запись контакта в **Dialing\_List**.

### **CREATE TRIGGER insert\_crm\_data**

```
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
```

**Примечание:** Кроме записей контакта обновления, когда контакты вызывают, Менеджер компании также выполняет некоторые массовые обновления. Они должны быть отфильтрованы триггером. Посмотрите раздел [Рекомендованных правил эксплуатации](#) этого Описания технологических решений для получения дополнительной информации.

## [Контактная информация обновления во внешней базе данных](#)

Пример триггера в этом разделе обновляет запись в **CRM\_Table** таблицы (в базе данных **CRM\_DB**) через соответствие номера учетной записи контакта. Например, если Менеджер кампании ВА обновляет запись в **Dialing\_List** с Номером учетной записи 1A2B3C4, то запись в **CRM\_Table** с тем же номером учетной записи обновлена одновременно.

```
CREATE TRIGGER updt_customer_by_acctnum
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
```

### [Удалите строку из внешней базы данных](#)

Если та запись контакта была закрыта в **Dialing\_List**, пример триггера в этом разделе удаляет запись из **CRM\_Table** таблицы (в базе данных **CRM\_DB**) через соответствие номера учетной записи контакта. Например, если Менеджер кампании ВА закрывает запись в **Dialing\_List** с Номером учетной записи 1A2B3C4, то запись в **CRM\_Table** с тем же номером учетной записи удалена.

```
CREATE TRIGGER del_customer_by_acctnum
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
```

### [Выполните внешний исполняемый файл от триггера](#)

Пример триггера в этом разделе выполняет программу **C:\MYAPP.EXE**. Триггер передает **AccountNumber** и **CallStatus** обновленной записи как параметры командной строки к

**MYAPP.EXE.** Это происходит каждый раз, когда Менеджер кампании ВА обновляет статус точно одной записи.

### CREATE TRIGGER 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
```

**Примечание:** В предыдущей таблице, вместо того, чтобы вызвать `xp_cmdshell` сохраненную процедуру для начала исполняемого файла триггер может также выполнить любую другую сохраненную процедуру.

## Вопросы производительности

- Триггеры Сервера SQL выполняются синхронно. Это означает, существует ли обновление, включают таблицу, команда обновления на той таблице не возвращается, пока триггерное выполнение не закончено. Если триггеры оптимально не разработаны, то обновления базы данных ВА замедляют. Кроме того, Менеджер кампании ВА мог бы встретиться с таймаутами подключения открытых баз данных (ODBC), в то время как это выполняет инструкции SQL для обновления таблиц. Как правило, триггер должен сделать минимальную возможную обработку и должен быть оптимизирован для скорости выполнения.
- Результат триггерного выполнения мог бы повлиять на результат исходной операции SQL. Например, предположите, что триггер обновления определен на **Dialing\_List**. Если триггер обновления отказывает, то операция обновления на **Dialing\_List** могла бы также отказать, и рассматриваемую запись можно оставить в **Dialing\_List** с неверным состоянием вызова (CallStatus).
- Если триггер вложен, то операция SQL не завершает, пока все гнездовые триггеры не выполнены. Поэтому сбой любого триггера мог бы отказать исходную операцию SQL.
- Если триггер выполняет внешнее приложение или другую сохраненную процедуру, то триггерное выполнение не завершено до внешнего приложения, или сохраненная процедура закончена.
- Использование курсоров в триггерах не рекомендуется из-за потенциально негативного воздействия на производительности.
- Триггеры не должны выполнять ЦП или с большой загрузкой памяти задачи.
- Все триггеры должны быть хорошо протестированы и подчеркнуты перед развертываниями.

## Рекомендуемые методы

- Всегда проверяйте, что только одна запись обновлена перед переходом. Это

препятствует тому, чтобы триггер действовал на массовые обновления, сделанные Менеджером кампании. Это примеры массовых обновлений, которые обычно отфильтровываются триггером: Изменение в CallStatus записи от А (АКТИВНОГО) к U (UNKNOWN), когда номеронабиратель с АКТИВНЫМИ записями разъединяет от Менеджера кампании. Менеджер кампании периодически обновляет записи в CallStatus U K P (В СОСТОЯНИИ ОЖИДАНИЯ). **Примечание:** Когда запись контакта обновлена в результате вызова, записи всегда обновляются по одному. Поэтому проверка, чтобы видеть, была ли точно одна запись обновлена гарантии против нежелательных триггерных действий большую часть времени. Когда объемная операция обновляет только одну строку, исключение из этого случая происходит.

- Проверьте для значения CallStatus перед переходом. Например, когда Менеджер кампании передает запись на номеронабирателя BA, Менеджер кампании устанавливает CallStatus в (АКТИВНЫЙ) A. Триггер может проверить для CallStatus. Если Обновленным CallStatus является A, он не продолжается дальше.
- Для оптимизации производительности во время обновлений базы данных BA время и операции загруженности ресурсов не должны быть выполнены в триггере. Можно создать таблицу для приложения в другой базе данных для служения в качестве очереди. Это ускоряет триггерное выполнение. Можно использовать триггер, чтобы вставить, обновить, или удалить данные в той таблице для приложения. Затем другое приложение может опросить ту таблицу для изменений и выполнить дальнейшие задачи по мере необходимости.

## [Дополнительные сведения](#)

- [Руководство по установке и конфигурации Blended Agent программного обеспечения Cisco ICM](#)
- [Руководство пользователя Blended Agent программного обеспечения Cisco ICM](#)
- [Cisco Systems – техническая поддержка и документация](#)