



## Commission Plan Support for Microsoft Professional Services Tools Library

---

*For Microsoft Dynamics GP versions:*

- 2018
- 2016
- 2015
- 2013
- 2010
- 10

**Copyright**

Manual copyright © 2011 EthoTech, Inc. All rights reserved.

Your right to copy this documentation is limited by copyright law and the terms of the software license agreement. As the software licensee, you may make a reasonable number of copies or printouts for your own use. Making unauthorized copies, adaptations, compilations, or derivative works for commercial distribution is prohibited and constitutes a punishable violation of the law.

Unless otherwise noted, all names of companies, products, street addresses, and persons contained herein are fictitious and are used solely to document the use of this product.

**Trademarks**

All company or product names mentioned are trademarks or registered trademarks of EthoTech, Inc., or of their respective holders.

**Warranty**

EthoTech disclaims any warranty regarding the sample code contained in this documentation, including the warranties of merchantability and fitness for a particular purpose.

**Limitation of liability**

The content of this manual is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by EthoTech. EthoTech assumes no responsibility or liability for any errors or inaccuracies that may appear in this manual. Neither EthoTech nor anyone else who has been involved in the creation, production or delivery of this documentation shall be liable for any indirect, incidental, special, exemplary or consequential damages, including but not limited to any loss of anticipated profit or benefits, resulting from the use of this documentation or sample code.

**Publication date**

December 2017

**Contents**

- Overview ..... 4
- Customer Combiner ..... 5
  - Alter SQL Stored Procedure ..... 5
  - Before Each Customer Combiner Run ..... 10
- Customer Modifier ..... 11
  - Alter SQL Stored Procedure ..... 11
  - Before Each Customer Modifier Run ..... 14
- Item Combiner ..... 15
  - Alter SQL Stored Procedure ..... 15
  - Before Each Item Combiner Run ..... 18
- Item Modifier ..... 19
  - Alter SQL Stored Procedure ..... 19
  - Before Each Item Modifier Run ..... 21

## Overview

Microsoft Dynamics GP Professional Services Tools Library (PSTL) provides functionality to mass update many entities in Dynamics GP. Due to the design of PSTL, some of these mass updates may fail based on data that exists in Commission Plan tables. They may also fail based on SQL triggers created by Commission Plan.

For example, if customer A and customer B are combined into customer B and both customers have a Customer Plan assigned, the combiner will fail with error "Violation of PRIMARY KEY constraint PKASI28050. Cannot insert duplicate key in object dbo. ASI28050." This occurs because the PSTL customer combiner finds every row in every table in the database containing value "A" in the CUSTNMBR column, changes the value to "B", and attempts to insert a new row. In our example, the combiner fails because a row already exists in table ASI28050 and the CUSTNMBR column is the primary key for the table, which requires the CUSTNMBR column to have a unique value for each row.

PSTL includes SQL stored procedures that run before a PSTL tool attempts to update the database. Logic may be added to these stored procedures to update tables so that PSTL doesn't encounter a situation which would cause a failure. In our above example, we can alter the taCustomerCombinerPre stored procedure to remove the Customer Plan assignment from customer A before the PSTL customer combiner attempts its updates.

The PSTL stored procedures exist in the company database, so if you use PSTL in multiple companies you'll need to alter the stored procedures in each database.

You'll only need to alter a stored procedure once, unless you re-install PSTL or upgrade to a new Dynamics GP version.

This document provides detailed information for altering the SQL stored procedures for PSTL tools that may have failures related to Commission Plan.

This document provides information for disabling the SQL triggers created by Commission Plan, using the Dynamics GP user interface.

## Customer Combiner

Table ASI28050 holds the customer plan > customer assignments for Commission Plan. When running the PSTL customer combiner to combine customers A & B into customer B, the combiner fails if records exist in table ASI28050 for both customers.

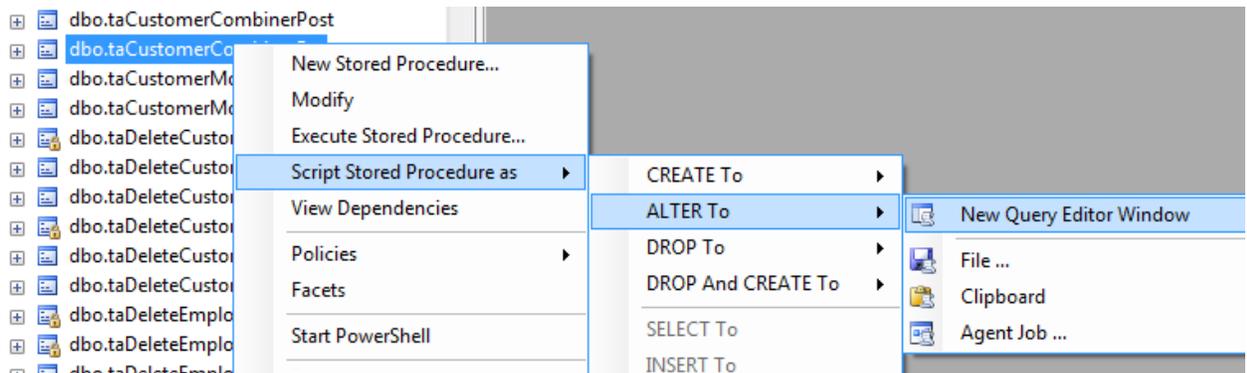
Table ASI28210 holds the salesperson > customer assignments (Customer Commission Plan Info window) for Commission Plan. When running the PSTL customer combiner to combine customers A & B into customer B, the combiner fails if records exist in table ASI28210 for both customers.

The combiner also fails if the Default checkbox is marked for a customer plan in the Customer Commission Plan Setup window.

## Alter SQL Stored Procedure

Use the following steps to automatically remove the ASI28050 record for customer A and merge the salespeople for customers A & B before the customer combiner runs:

1. Open SQL Management Studio, expand the company database, expand Programmability, and expand Stored Procedures.
2. If you are running the Customer Combiner from PSTL:
  - a. Right-click procedure **dbo.taCustomerCombinerPre** and choose *Script Stored Procedure As > ALTER To > New Query Editor Window* (which creates a new script in the query editor window).



- b. Find the following statement in the Alter procedure taCustomerCombinerPre script that you created in step a:

```
/* Create Custom Business Logic */
```

### Screenshot of Alter procedure taCustomerCombinerPre

```

/***** Object: StoredProcedure [dbo].[taCustomerCombinerPre]    Script Date: 04/15/2011 16:46:56 *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO

ALTER procedure [dbo].[taCustomerCombinerPre]
/* *****
(c) 2001 Microsoft Great Plains, Inc. *****
PROCEDURE NAME: taCustomerCombinerPre * * *
DATABASE: *          Company * * *
RETURN VALUE: * *    0 = Successful *    non-0 = Not successful * *
REVISION HISTORY: * * Date          Who    Comments * -----
***** */
@I_charStartCustomer char(30) output,
@I_charEndCustomer char(30) output,
@cStartCustomer char(50) output,
@cEndCustomer char(50) output,
@O_iErrorState int output
as select @O_iErrorState = 0
/* Create Custom Business Logic */

/* End Create Custom Business Logic */
return
GO

```

- c. Just after the */\* Create Custom Business Logic \*/* statement, add the following statements:

-- Commission Plan: if a customer plan exists for both the source and destination customers, remove the customer plan from the source customer.

```

IF (SELECT COUNT(*) FROM ASI28050 WHERE CUSTNMBR = @I_charStartCustomer) > 0 AND
(SELECT COUNT(*) FROM ASI28050 WHERE CUSTNMBR = @I_charEndCustomer) > 0
DELETE ASI28050 WHERE CUSTNMBR = @I_charStartCustomer

```

-- Commission Plan: if salespeople have been assigned to the source customer, merge them with the salespeople assigned to the destination customer.

-- If both customers currently have salespeople assigned, the salespeople assigned to the destination customer will be the currently assigned salespeople for the combined customer.

```

IF (SELECT COUNT(*) FROM ASI28210 WHERE CUSTNMBR = @I_charStartCustomer) > 0
BEGIN

```

```

    DECLARE @NextIdx INT, @CountIdxStartCust INT, @CountIdxEndCust INT, @i INT
    SET @NextIdx = ISNULL((SELECT MAX(ETI_CP_Cust_SP_Index) FROM ASI28210 WHERE
CUSTNMBR = @I_charStartCustomer OR CUSTNMBR = @I_charEndCustomer),0)
    SET @CountIdxStartCust = ISNULL((SELECT MAX(ETI_CP_Cust_SP_Index) FROM ASI28210
WHERE CUSTNMBR = @I_charStartCustomer), 0)
    SET @CountIdxEndCust = ISNULL((SELECT MAX(ETI_CP_Cust_SP_Index) FROM ASI28210
WHERE CUSTNMBR = @I_charEndCustomer), 0)

```

```

SET @i = 1

```

```

WHILE (@i <= @CountIdxStartCust)

```

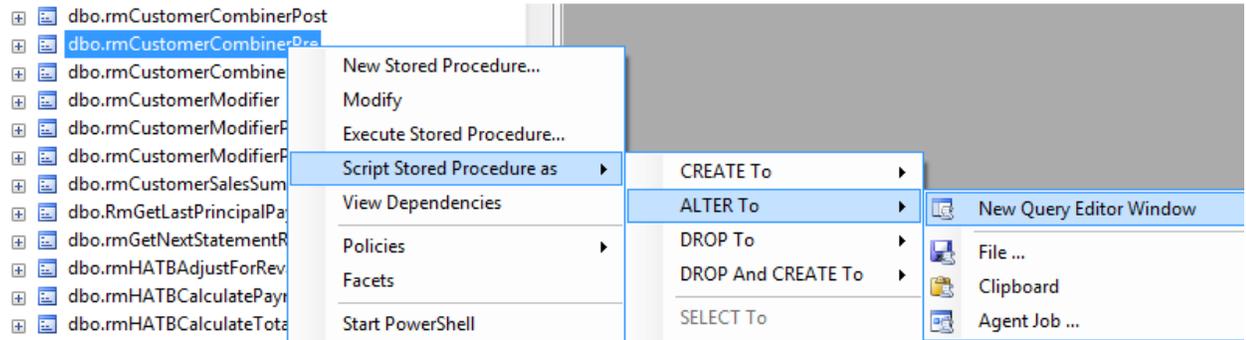
```

BEGIN
  IF (SELECT COUNT(*) FROM ASI28210 WHERE CUSTNMBR = @I_charStartCustomer AND
ETI_CP_Cust_SP_Index = @i) > 0
  BEGIN
    SET @NextIdx = @NextIdx + 1
    UPDATE ASI28210 SET CUSTNMBR = @I_charEndCustomer, ETI_CP_Cust_SP_Index =
@NextIdx WHERE CUSTNMBR = @I_charStartCustomer AND ETI_CP_Cust_SP_Index = @i
    UPDATE ETI28910 SET CUSTNMBR = @I_charEndCustomer, ETI_CP_Cust_SP_Index =
@NextIdx WHERE CUSTNMBR = @I_charStartCustomer AND ETI_CP_Cust_SP_Index = @i
  END
  SET @i = @i + 1
END

SET @i = 1
WHILE (@i <= @CountIdxEndCust)
BEGIN
  IF (SELECT COUNT(*) FROM ASI28210 WHERE CUSTNMBR = @I_charEndCustomer AND
ETI_CP_Cust_SP_Index = @i) > 0
  BEGIN
    SET @NextIdx = @NextIdx + 1
    UPDATE ASI28210 SET ETI_CP_Cust_SP_Index = @NextIdx WHERE CUSTNMBR =
@I_charEndCustomer AND ETI_CP_Cust_SP_Index = @i
    UPDATE ETI28910 SET ETI_CP_Cust_SP_Index = @NextIdx WHERE CUSTNMBR =
@I_charEndCustomer AND ETI_CP_Cust_SP_Index = @i
  END
  SET @i = @i + 1
END
END

```

- d. Run the script, which will recreate the taCustomerCombinerPre stored procedure with our custom logic.
3. If you are running the Customer Combiner from Dynamics GP (version 2013 Service Pack 2 or later):
    - a. Right-click procedure **dbo.rmCustomerCombinerPre** and choose *Script Stored Procedure As > ALTER To > New Query Editor Window* (which creates a new script in the query editor window).



- b. Find the following statement in the Alter procedure `rmCustomerCombinerPre` script that you created in step a:

**`select @O_iErrorState = 0`**

*Screenshot of Alter procedure `rmCustomerCombinerPre`*

```

/***** Object: StoredProcedure [dbo].[rmCustomerCombinerPre]    Script Date: 11/01/2013 11:34:47 *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO

ALTER procedure [dbo].[rmCustomerCombinerPre]
    @I_charStartCustomer char(30) output,
    @I_charEndCustomer char(30) output,
    @cStartCustomer char(50) output,
    @cEndCustomer char(50) output,
    @O_iErrorState int output

as select @O_iErrorState = 0

return
GO

```

- c. Just after the `select @O_iErrorState = 0` statement, add the following statements:

-- Commission Plan: if a customer plan exists for both the source and destination customers, remove the customer plan from the source customer.

```

IF (SELECT COUNT(*) FROM ASI28050 WHERE CUSTNMBR = @I_charStartCustomer) > 0 AND
(SELECT COUNT(*) FROM ASI28050 WHERE CUSTNMBR = @I_charEndCustomer) > 0
DELETE ASI28050 WHERE CUSTNMBR = @I_charStartCustomer

```

-- Commission Plan: if salespeople have been assigned to the source customer, merge them with the salespeople assigned to the destination customer.

-- If both customers currently have salespeople assigned, the salespeople assigned to the destination customer will be the currently assigned salespeople for the combined customer.

```

IF (SELECT COUNT(*) FROM ASI28210 WHERE CUSTNMBR = @I_charStartCustomer) > 0
BEGIN
    DECLARE @NextIdx INT, @CountIdxStartCust INT, @CountIdxEndCust INT, @i INT
    SET @NextIdx = ISNULL((SELECT MAX(ETI_CP_Cust_SP_Index) FROM ASI28210 WHERE
CUSTNMBR = @I_charStartCustomer OR CUSTNMBR = @I_charEndCustomer),0)

```

```

SET @CountIdxStartCust = ISNULL((SELECT MAX(ETI_CP_Cust_SP_Index) FROM ASI28210
WHERE CUSTNMBR = @I_charStartCustomer), 0)
SET @CountIdxEndCust = ISNULL((SELECT MAX(ETI_CP_Cust_SP_Index) FROM ASI28210
WHERE CUSTNMBR = @I_charEndCustomer), 0)

```

```

SET @i = 1
WHILE (@i <= @CountIdxStartCust)
BEGIN
    IF (SELECT COUNT(*) FROM ASI28210 WHERE CUSTNMBR = @I_charStartCustomer AND
ETI_CP_Cust_SP_Index = @i) > 0
    BEGIN
        SET @NextIdx = @NextIdx + 1
        UPDATE ASI28210 SET CUSTNMBR = @I_charEndCustomer, ETI_CP_Cust_SP_Index =
@NextIdx WHERE CUSTNMBR = @I_charStartCustomer AND ETI_CP_Cust_SP_Index = @i
        UPDATE ETI28910 SET CUSTNMBR = @I_charEndCustomer, ETI_CP_Cust_SP_Index =
@NextIdx WHERE CUSTNMBR = @I_charStartCustomer AND ETI_CP_Cust_SP_Index = @i
    END
    SET @i = @i + 1
END

```

```

SET @i = 1
WHILE (@i <= @CountIdxEndCust)
BEGIN
    IF (SELECT COUNT(*) FROM ASI28210 WHERE CUSTNMBR = @I_charEndCustomer AND
ETI_CP_Cust_SP_Index = @i) > 0
    BEGIN
        SET @NextIdx = @NextIdx + 1
        UPDATE ASI28210 SET ETI_CP_Cust_SP_Index = @NextIdx WHERE CUSTNMBR =
@I_charEndCustomer AND ETI_CP_Cust_SP_Index = @i
        UPDATE ETI28910 SET ETI_CP_Cust_SP_Index = @NextIdx WHERE CUSTNMBR =
@I_charEndCustomer AND ETI_CP_Cust_SP_Index = @i
    END
    SET @i = @i + 1
END
END

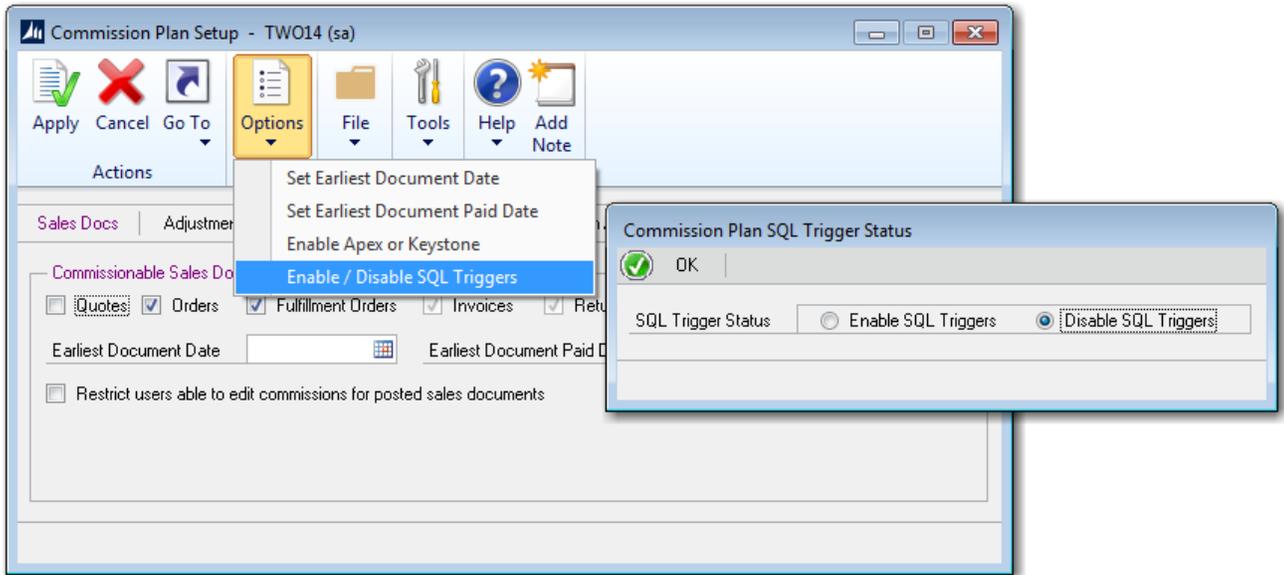
```

- d. Run the script, which will recreate the rmCustomerCombinerPre stored procedure with our custom logic.

## Before Each Customer Combiner Run

If you are running The EthoSeries build 14.00.14 or later:

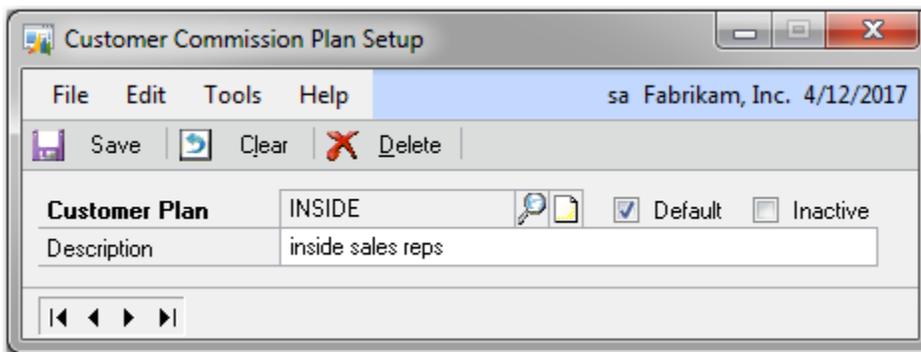
Before running the Customer Combiner, use the Commission Plan Setup window to disable SQL triggers.



After the combiner completes, enable SQL triggers.

If you are running The EthoSeries build 14.00.13 or earlier:

Before running the Customer Combiner, use the Customer Commission Plan Setup window to ensure that no customer plan has the Default checkbox marked. If a customer plan is marked as Default, unmark the checkbox and save the record.



After the combiner completes, mark the Default checkbox for the desired customer plan and save the record.

## Customer Modifier

Table ASI28050 holds the customer plan > customer assignments for Commission Plan. When running the PSTL customer modifier to change customer number A to customer number B, the modifier fails if records exist in table ASI28050 for both customers.

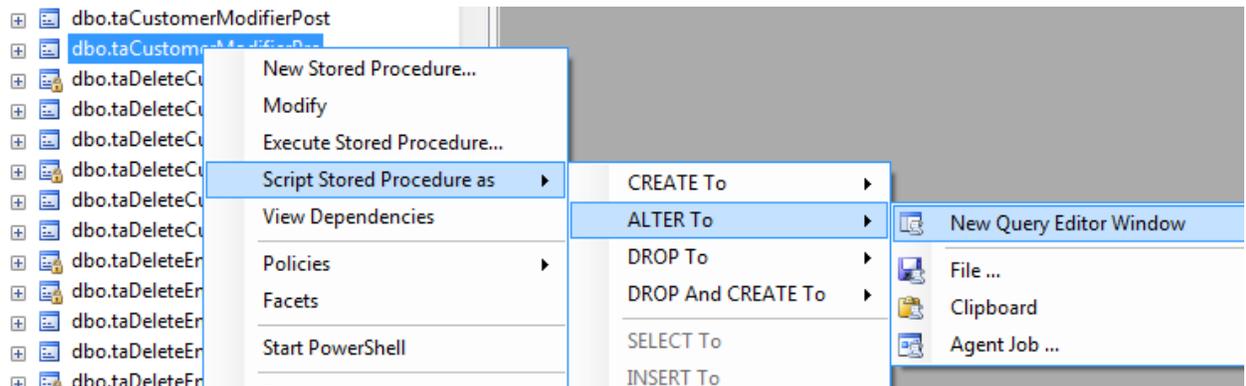
Table ASI28210 holds the salesperson > customer assignments (Customer Commission Plan Info window) for Commission Plan. When running the PSTL customer combiner to combine customers A & B into customer B, the combiner fails if records exist in table ASI28210 for both customers.

The modifier also fails if the Default checkbox is marked for a customer plan in the Customer Commission Plan Setup window.

## Alter SQL Stored Procedure

Use the following steps to automatically remove the ASI28050 and ASI28210 records for customer number B before the customer modifier runs:

1. Open SQL Management Studio, expand the company database, expand Programmability, and expand Stored Procedures.
2. If you are running the Customer Modifier from PSTL:
  - a. Right-click procedure **dbo.taCustomerModifierPre** and choose Script Stored Procedure As > ALTER To > New Query Editor Window (which creates a new script in the query editor window).



- b. Find the following statement in the Alter procedure taCustomerCombinerPre script that you created in step 2:

```
/* Create Custom Business Logic */
```

### Screenshot of Alter procedure taCustomerModifierPre

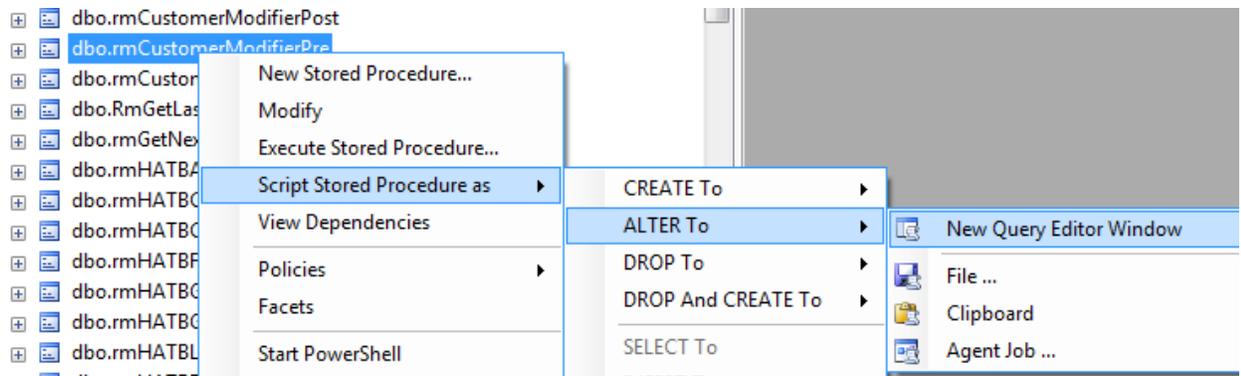
```
/* Object: StoredProcedure [dbo].[taCustomerModifierPre] Script Date: 04/15/2011 17:03:28 */
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO

ALTER procedure [dbo].[taCustomerModifierPre]
/* *****
(c) 2001 Microsoft Great Plains, Inc. *****
PROCEDURE NAME: taCustomerModifierPre * * *
DATABASE: * Company * * *
RETURN VALUE: * * 0 = Successful * non-0 = Not successful * *
REVISION HISTORY: * * Date Who Comments * -----
***** */
@I_charStartCustomer char(30) output,
@I_charEndCustomer char(30) output,
@cStartCustomer char(50) output,
@cEndCustomer char(50) output,
@O_iErrorState int output
as select @O_iErrorState = 0
/* Create Custom Business Logic */

/* End Create Custom Business Logic */
return
GO
```

- c. Just after the `/* Create Custom Business Logic */` statement, add the following statements:
    - Commission Plan: if a customer plan exists for both the source and destination customers, remove the customer plan from the destination customer.  
IF (SELECT COUNT(\*) FROM ASI28050 WHERE CUSTNMBR = @I\_charStartCustomer) > 0 AND (SELECT COUNT(\*) FROM ASI28050 WHERE CUSTNMBR = @I\_charEndCustomer) > 0  
DELETE ASI28050 WHERE CUSTNMBR = @I\_charEndCustomer
    - Commission Plan: if salespeople have been assigned to both the source and destination customers, remove the salesperson assignments from the destination customer.  
IF (SELECT COUNT(\*) FROM ASI28210 WHERE CUSTNMBR = @I\_charStartCustomer) > 0 AND (SELECT COUNT(\*) FROM ASI28210 WHERE CUSTNMBR = @I\_charEndCustomer) > 0  
DELETE ASI28210 WHERE CUSTNMBR = @I\_charEndCustomer
  - d. Run the script, which will recreate the taCustomerModifierPre stored procedure with our custom logic.
3. If you are running the Customer Modifier from Dynamics GP (version 2013 Service Pack 2 or later):
    - a. Right-click procedure **dbo.rmCustomerModifierPre** and choose Script Stored Procedure As > ALTER To > New Query Editor Window (which creates a new script in the query editor window).



- b. Find the following statement in the Alter procedure `rmCustomerModifierPre` script that you created in step 2:

**select @O\_iErrorState = 0**

*Screenshot of Alter procedure `rmCustomerModifierPre`*

```

/***** Object: StoredProcedure [dbo].[rmCustomerModifierPre]    Script Date: 11/01/2013 12:30:34 *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
|
ALTER procedure [dbo].[rmCustomerModifierPre]
    @I_charStartCustomer char(30) output,
    @I_charEndCustomer char(30) output,
    @cStartCustomer char(50) output,
    @cEndCustomer char(50) output,
    @O_iErrorState int output

    as select @O_iErrorState = 0

    return
GO

```

- c. Just after the `select @O_iErrorState = 0` statement, add the following statements:

```

-- Commission Plan: if a customer plan exists for both the source and destination customers,
remove the customer plan from the destination customer.
IF (SELECT COUNT(*) FROM ASI28050 WHERE CUSTNMBR = @I_charStartCustomer) > 0 AND
(SELECT COUNT(*) FROM ASI28050 WHERE CUSTNMBR = @I_charEndCustomer) > 0
    DELETE ASI28050 WHERE CUSTNMBR = @I_charEndCustomer

-- Commission Plan: if salespeople have been assigned to both the source and destination
customers, remove the salesperson assignments from the destination customer.
IF (SELECT COUNT(*) FROM ASI28210 WHERE CUSTNMBR = @I_charStartCustomer) > 0 AND
(SELECT COUNT(*) FROM ASI28210 WHERE CUSTNMBR = @I_charEndCustomer) > 0
    DELETE ASI28210 WHERE CUSTNMBR = @I_charEndCustomer

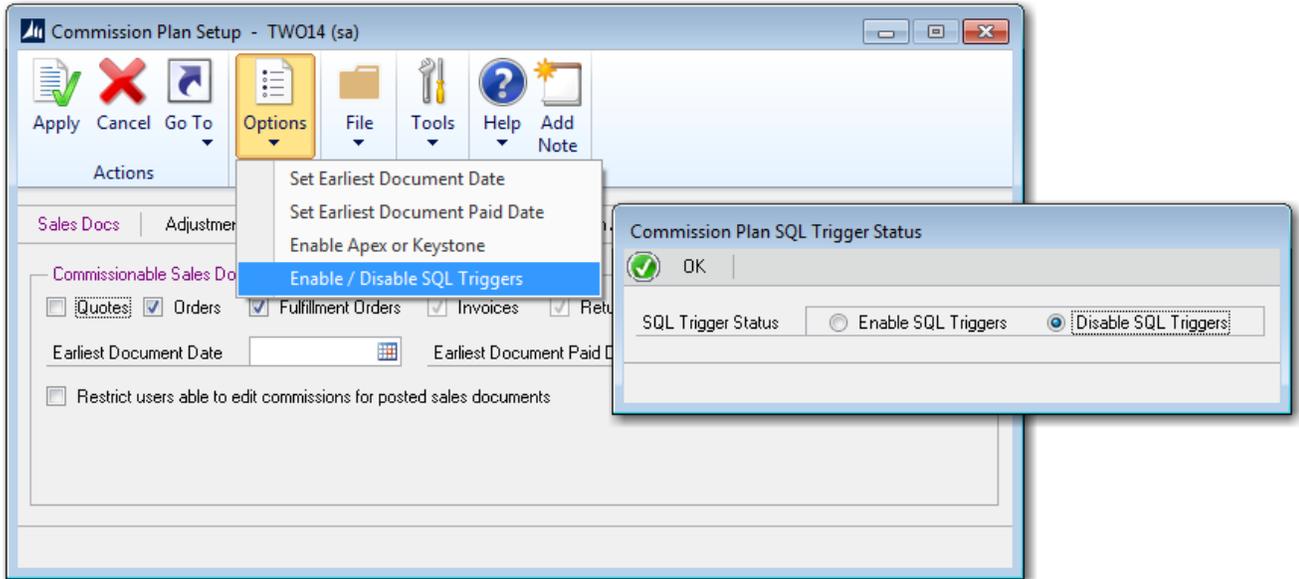
```

- d. Run the script, which will recreate the `rmCustomerModifierPre` stored procedure with our custom logic.

## Before Each Customer Modifier Run

If you are running The EthoSeries build 14.00.14 or later:

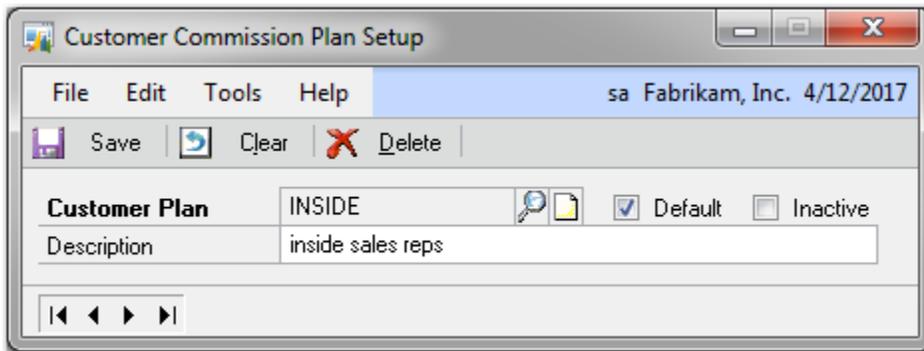
Before running the Customer Combiner, use the Commission Plan Setup window to disable SQL triggers.



After the modifier completes, enable SQL triggers.

If you are running The EthoSeries build 14.00.13 or earlier:

Before running the Customer Modifier, use the Customer Commission Plan Setup window to ensure that no customer plan has the Default checkbox marked. If a customer plan is marked as Default, unmark the checkbox and save the record.



After the modifier completes, mark the Default checkbox for the desired customer plan and save the record.

## Item Combiner

Table ASI28040 holds the item plan > item assignments for Commission Plan. When running the PSTL item combiner to combine items A & B into item B, the combiner fails if records exist in table ASI28040 for both items.

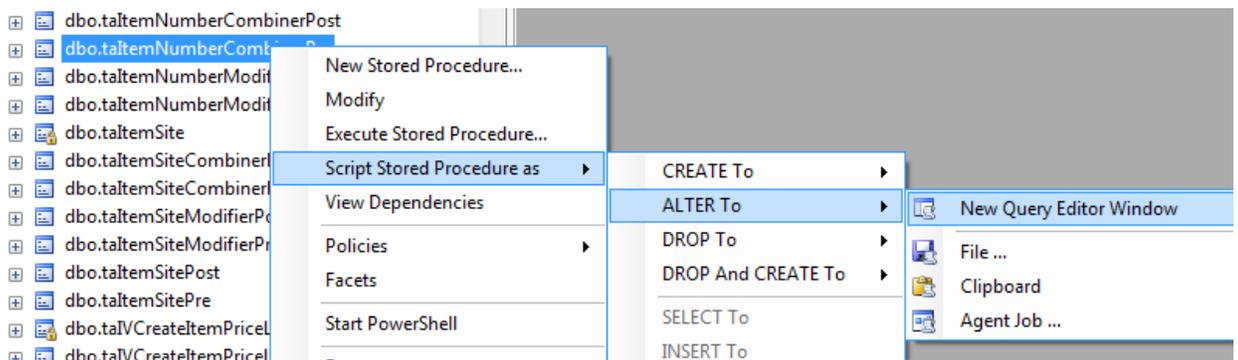
Table ASI28220 holds the salesperson > item assignments (Item Commission Plan Info window) for Commission Plan. When running the PSTL customer combiner to combine items A & B into item B, the combiner fails if records exist in table ASI28220 for both items.

The combiner also fails if the Default checkbox is marked for an item plan in the Item Commission Plan Setup window.

## Alter SQL Stored Procedure

Use the following steps to automatically remove the ASI28040 record for item A and merge the salespeople for items A & B before the item combiner runs:

1. Open SQL Management Studio, expand the company database, expand Programmability, and expand Stored Procedures.
2. Right-click procedure **dbo.taItemNumberCombinerPre** and choose *Script Stored Procedure As > ALTER To > New Query Editor Window* (which creates a new script in the query editor window).



3. Find the following statement in the Alter procedure taItemNumberCombinerPre script that you created in step 2:

```
/* Create Custom Business Logic */
```

### Screenshot of Alter procedure taItemNumberCombinerPre

```

/***** Object: StoredProcedure [dbo].[taItemNumberCombinerPre]    Script Date: 04/15/2011 17:42:42 *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO

ALTER procedure [dbo].[taItemNumberCombinerPre]
/* *****
(c) 2001 Microsoft Great Plains, Inc. *****
PROCEDURE NAME: taItemNumberCombinerPre * * *
DATABASE: *                Company * * *
RETURN VALUE: * * 0 = Successful * non-0 = Not successful * * REVISION HISTORY: * * Date Who
***** */
@I_charStartItem char(31) output,
@I_charEndItem char(31) output,
@cStartItem char(50) output,
@cEndItem char(50) output,
@O_iErrorState int output
as select @O_iErrorState = 0
/* Create Custom Business Logic */

/* End Create Custom Business Logic */
return
GO

```

4. Just after the */\* Create Custom Business Logic \*/* statement, add the following statements:

-- Commission Plan: if an item plan exists for both the source and destination items, remove the item plan from the source item.

```

IF (SELECT COUNT(*) FROM ASI28040 WHERE ITEMNMBR = @I_charStartItem) > 0 AND (SELECT
COUNT(*) FROM ASI28040 WHERE ITEMNMBR = @I_charEndItem) > 0
DELETE ASI28040 WHERE ITEMNMBR = @I_charStartItem

```

-- Commission Plan: if salespeople have been assigned to the source item, merge them with the salespeople assigned to the destination item.

-- If both items currently have salespeople assigned, the salespeople assigned to the destination item will be the currently assigned salespeople for the combined item.

```

IF (SELECT COUNT(*) FROM ASI28220 WHERE ITEMNMBR = @I_charStartItem) > 0
BEGIN
    DECLARE @NextIdx INT, @CountIdxStartItem INT, @CountIdxEndItem INT, @i INT
    SET @NextIdx = ISNULL((SELECT MAX(ETI_CP_Item_SP_Index) FROM ASI28220 WHERE
ITEMNMBR = @I_charStartItem OR ITEMNMBR = @I_charEndItem),0)
    SET @CountIdxStartItem = ISNULL((SELECT MAX(ETI_CP_Item_SP_Index) FROM ASI28220
WHERE ITEMNMBR = @I_charStartItem), 0)
    SET @CountIdxEndItem = ISNULL((SELECT MAX(ETI_CP_Item_SP_Index) FROM ASI28220
WHERE ITEMNMBR = @I_charEndItem), 0)

```

```

SET @i = 1
WHILE (@i <= @CountIdxStartItem)
BEGIN
    IF (SELECT COUNT(*) FROM ASI28220 WHERE ITEMNMBR = @I_charStartItem AND
ETI_CP_Item_SP_Index = @i) > 0
    BEGIN
        SET @NextIdx = @NextIdx + 1
    
```

```

    UPDATE ASI28220 SET ITEMNMBR = @I_charEndItem, ETI_CP_Item_SP_Index = @NextIdx
WHERE ITEMNMBR = @I_charStartItem AND ETI_CP_Item_SP_Index = @i
    UPDATE ETI28911 SET ITEMNMBR = @I_charEndItem, ETI_CP_Item_SP_Index = @NextIdx
WHERE ITEMNMBR = @I_charStartItem AND ETI_CP_Item_SP_Index = @i
    END
    SET @i = @i + 1
    END

SET @i = 1
WHILE (@i <= @CountIdxEndItem)
BEGIN
    IF (SELECT COUNT(*) FROM ASI28220 WHERE ITEMNMBR = @I_charEndItem AND
ETI_CP_Item_SP_Index = @i) > 0
    BEGIN
        SET @NextIdx = @NextIdx + 1
        UPDATE ASI28220 SET ETI_CP_Item_SP_Index = @NextIdx WHERE ITEMNMBR =
@I_charEndItem AND ETI_CP_Item_SP_Index = @i
        UPDATE ETI28911 SET ETI_CP_Item_SP_Index = @NextIdx WHERE ITEMNMBR =
@I_charEndItem AND ETI_CP_Item_SP_Index = @i
    END
    SET @i = @i + 1
    END
END

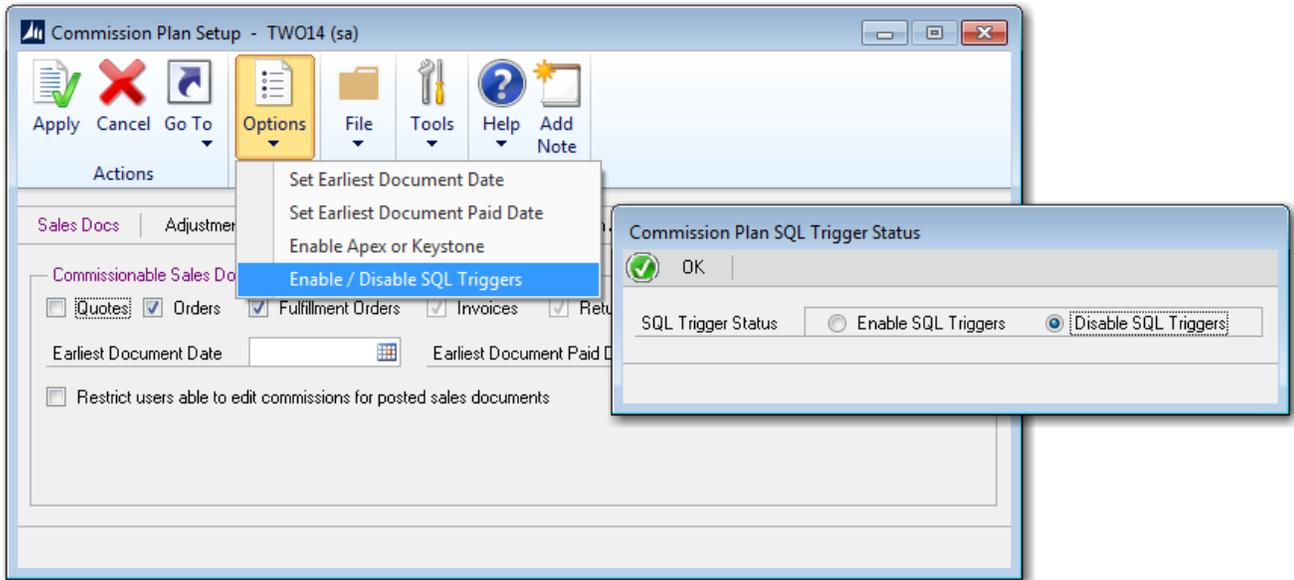
```

5. Run the script, which will recreate the taltemNumberCombinerPre stored procedure with our custom logic.

## Before Each Item Combiner Run

If you are running The EthoSeries build 14.00.14 or later:

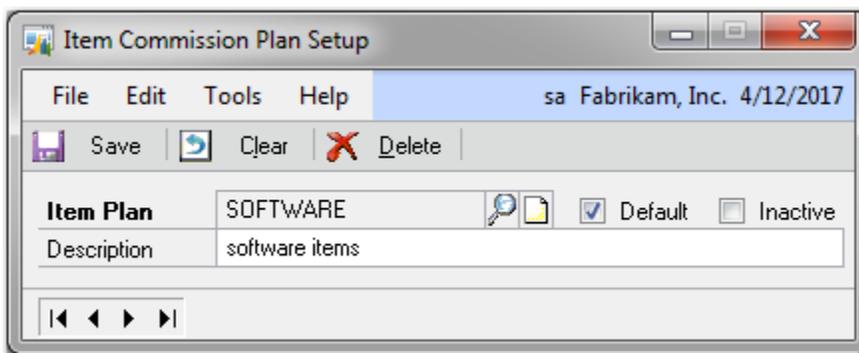
Before running the Customer Combiner, use the Commission Plan Setup window to disable SQL triggers.



After the combiner completes, enable SQL triggers.

If you are running The EthoSeries build 14.00.13 or earlier:

Before running the Item Combiner, check if the Default checkbox is marked for an item plan in the Item Commission Plan Setup window. If an item plan is marked as Default, unmark the checkbox and save the record.



After the combiner completes, re-mark the Default checkbox for the desired item plan.

## Item Modifier

Table ASI28040 holds the item plan > item assignments for Commission Plan. When running the PSTL item modifier to change item number A to item number B, the modifier fails if records exist in table ASI28040 for both item number A and item number B.

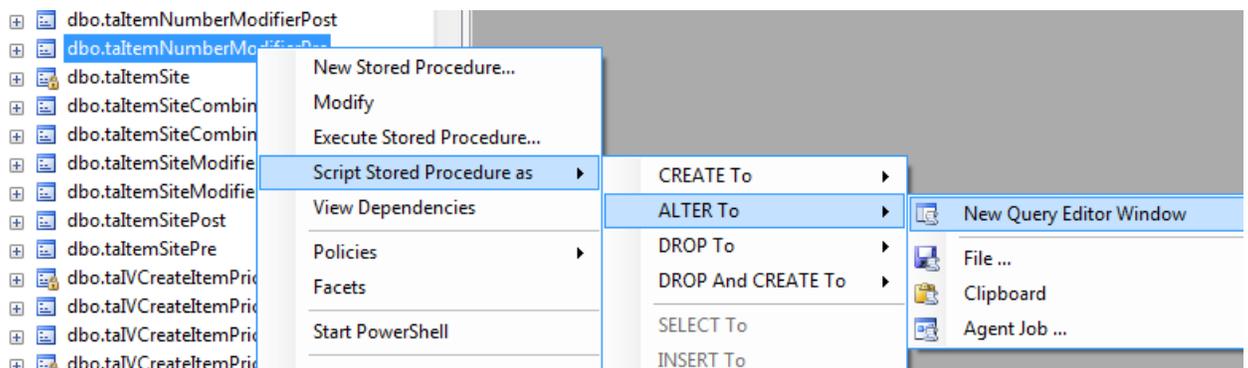
Table ASI28220 holds the salesperson > item assignments (Item Commission Plan Info window) for Commission Plan. When running the PSTL item modifier to change item number A to item number B, the modifier fails if records exist in table ASI28220 for both items.

The modifier also fails if the Default checkbox is marked for an item plan in the Item Commission Plan Setup window.

## Alter SQL Stored Procedure

Use the following steps to automatically remove the ASI28040 & ASI28220 records for item number B before the item modifier runs:

1. Open SQL Management Studio, expand the company database, expand Programmability, and expand Stored Procedures.
2. Right-click procedure **dbo.taItemNumberModifierPre** and choose Script Stored Procedure As > ALTER To > New Query Editor Window (which creates a new script in the query editor window).



3. Find the following statement in the Alter procedure taItemNumberModifierPre script that you created in step 2:

```
/* Create Custom Business Logic */
```

### Screenshot of Alter procedure taltemNumberModifierPre

```
/****** Object: StoredProcedure [dbo].[taltemNumberModifierPre]    Script Date: 04/15/2011 17:13:53 *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO

ALTER procedure [dbo].[taltemNumberModifierPre]
/* *****
(c) 2001 Microsoft Great Plains, Inc. *****
PROCEDURE NAME: taltemNumberModifierPre * * *
DATABASE: *                Company * * *
RETURN VALUE: * * 0 = Successful *    non-0 = Not successful * *
REVISION HISTORY: * * Date      Who      Comments * -----
***** */
@I_charStartItem    char(30) output,
@I_charEndItem      char(30) output,
@cStartItem         char(50) output,
@cEndItem           char(50) output,
@O_iErrorState      int output
as select @O_iErrorState = 0
/* Create Custom Business Logic */

/* End Create Custom Business Logic */
return
GO
```

4. Just after the */\* Create Custom Business Logic \*/* statement, add the following statements:

-- Commission Plan: if an item plan exists for both the source and destination items, remove the item plan from the destination item.

```
IF (SELECT COUNT(*) FROM ASI28040 WHERE ITEMNMBR = @I_charStartItem) > 0 AND (SELECT
COUNT(*) FROM ASI28040 WHERE ITEMNMBR = @I_charEndItem) > 0
DELETE ASI28040 WHERE ITEMNMBR = @I_charEndItem
```

-- Commission Plan: if salespeople have been assigned to both the source and destination items, remove the salesperson assignments from the destination item.

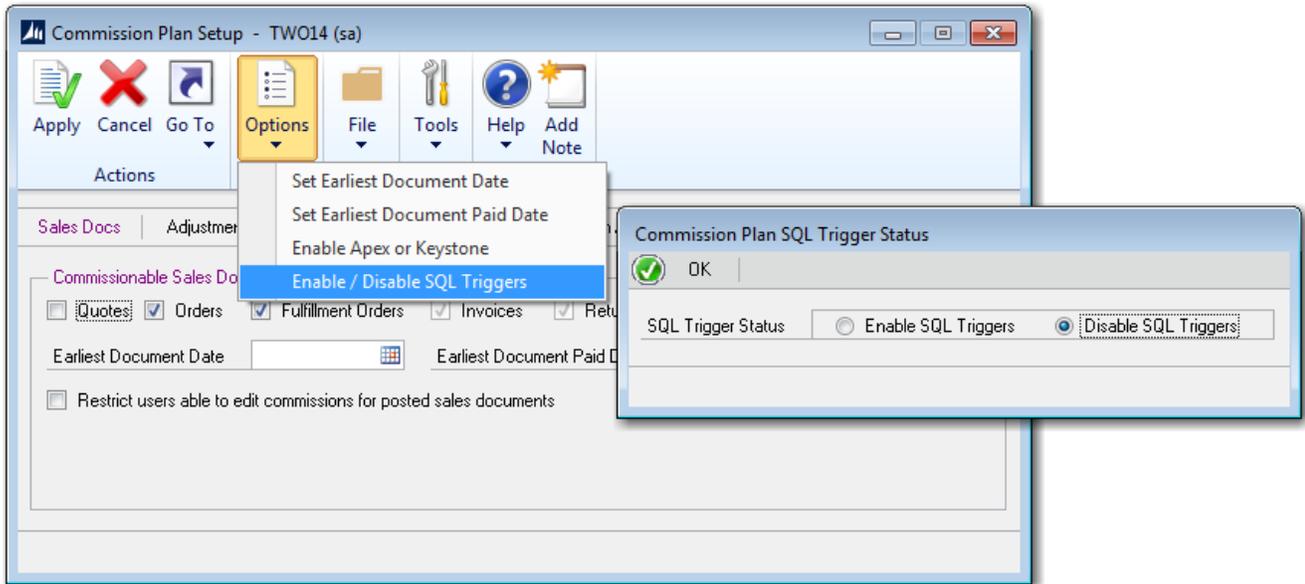
```
IF (SELECT COUNT(*) FROM ASI28220 WHERE ITEMNMBR = @I_charStartItem) > 0 AND (SELECT
COUNT(*) FROM ASI28220 WHERE ITEMNMBR = @I_charEndItem) > 0
DELETE ASI28220 WHERE ITEMNMBR = @I_charEndItem
```

5. Run the script, which will recreate the taltemNumberModifierPre stored procedure with our custom logic.

## Before Each Item Modifier Run

If you are running The EthoSeries build 14.00.14 or later:

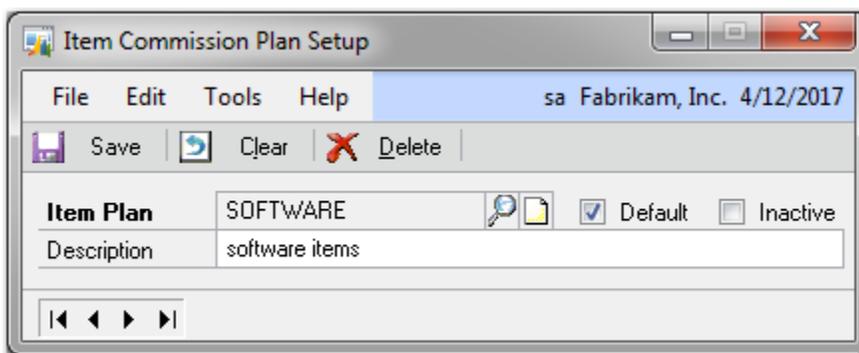
Before running the Customer Combiner, use the Commission Plan Setup window to disable SQL triggers.



After the modifier completes, enable SQL triggers.

If you are running The EthoSeries build 14.00.13 or earlier:

Before running the Item Modifier, check if the Default checkbox is marked for an item plan in the Item Commission Plan Setup window. If an item plan is marked as Default, unmark the checkbox and save the record.



After the modifier completes, re-mark the Default checkbox for the desired item plan.