Create and Use Stored Procedure Cycle_Invoice_Date_Change

04/26/2024 2:58 pm EDT

Description of Issue:

There may be times when someone will use the wrong invoice date when generating and posting a Cycle Batch in SedonaOffice.

The stored procedure in this document can be used to update the invoice and posting date on the invoice and the date and accounting period on the GL journal entries for those invoices.

This procedure will not change anything to do with the deferred income. If the user has created and posted a Cycle Batch for the wrong month, additional data work would be required and may be billable.

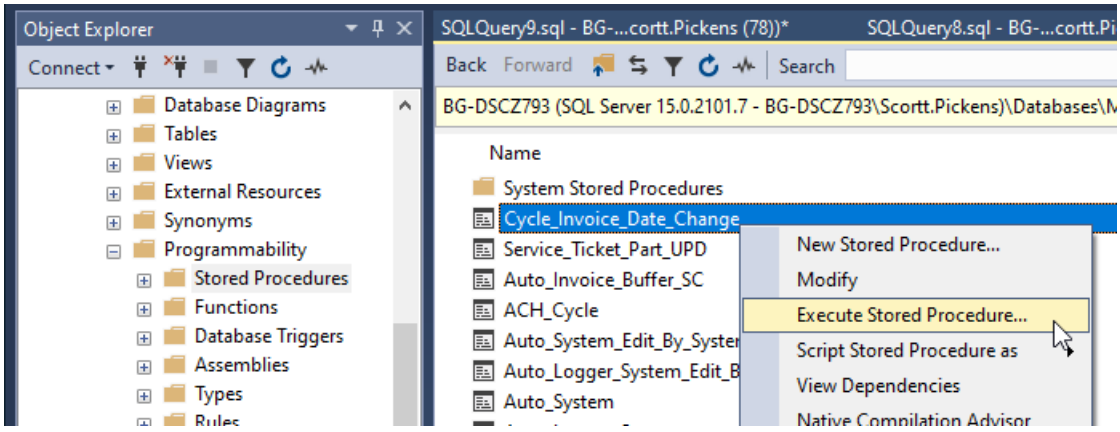
Resolution:

The SQL script in this document can be run against the database to create the stored procedure. Once created it can be used if the issue occurs again in the future.

Once the SP is created, it can be run to update the cycle batch invoices. This will update ALL invoices in the batch.

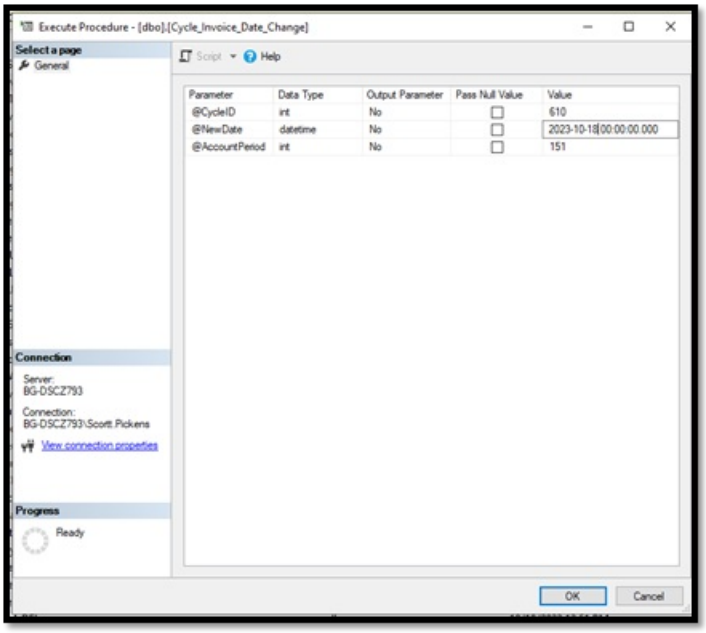
The procedure can be run from a query window or from SQL Management Studio Object Explorer.

You can right-click on the stored procedure and select Execute Stored Procedure.

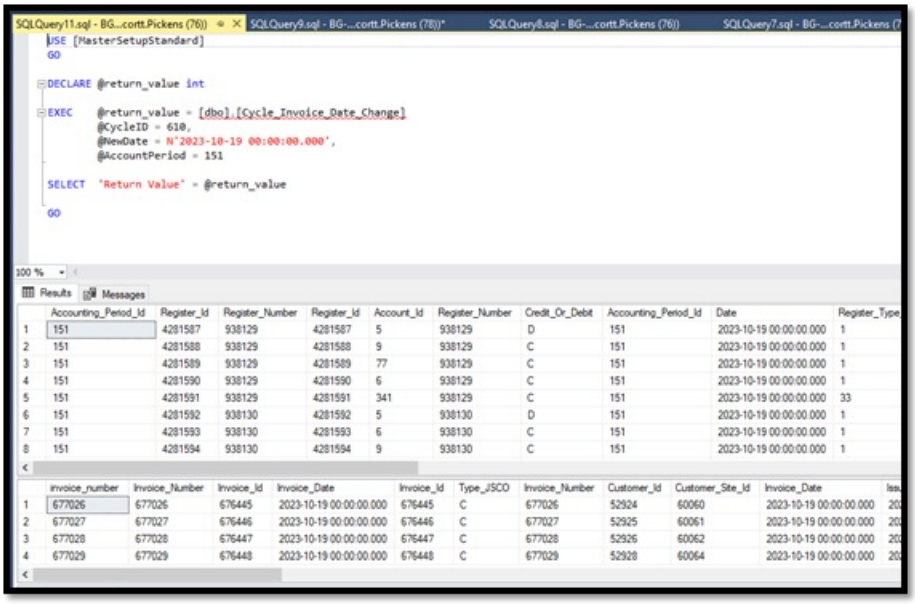


You will need to know the Cycle_ID, GL accounting period for the new date, and the new invoice date.

Enter the variables needed for the procedure and click OK.



You should see the procedure run and return the results of the change for review.

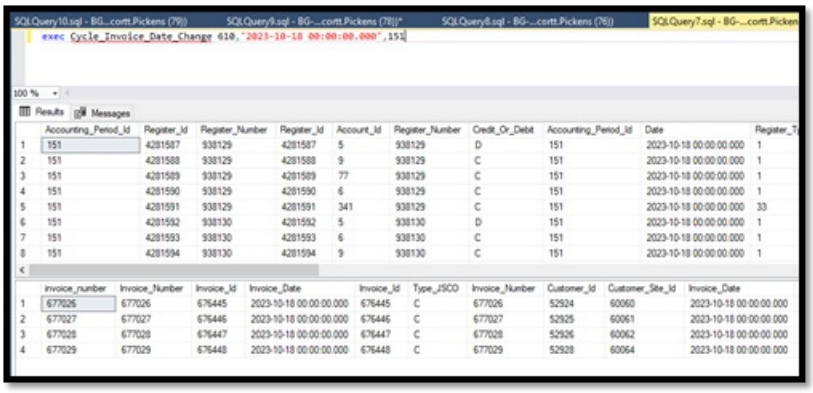


You can also execute the stored procedure from a query window.

You will need to provide the variables in the statement. Here is an example of how to run the execute statement. Be sure the correct variables are sent in the correct order.

```
CycleID = 610
NewDate = '2023-10-18 00:00:00.000'
AccountPeriod = 151

exec Cycle_Invoice_Date_Change 610,'2023-10-18 00:00:00.000',151
```



The stored procedure will create backup tables of the AR_Invoice and the GL_Register if there is a problem these can be used to restore the data back to its original state.

It is recommended to test the changes in the Sandbox before making the changes in the production database.

Execute this script against the databases needed to create the procedure.

```
/****** Object:  StoredProcedure [dbo].[Cycle_Invoice_Date_Change]    Script Date: 10/19/2023 4:56:57 PM ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:      <Scott Pickens>
-- Create date: <10/19/2023>
-- Description: <Cycle_Invoice_Date_Change>
-- =============================================
-- Cycle_ID, New Date and Accounting Period is needed for scripts to work ---
-- Find accounting period use the start date for the period in question ---
-- select * from GL_Accounting_Period where Start_Date='2023-10-01 00:00:00.000'
-- Set these variables with the Cycle Batch, Date, and Accounting Period to use to update
-- the invoices and GL entries ---

CREATE PROCEDURE [dbo].[Cycle_Invoice_Date_Change]

@CycleID int,
@NewDate datetime,
@AccountPeriod int

AS

--- SET @CycleID = 610
--- SET @NewDate = '2023-10-15 00:00:00.000'
--- SET @AccountPeriod = 151

BEGIN

---Drop Backup tables if they exist
IF OBJECT_ID(N'dbo.AR_Invoice_DateFix_Backup', N'U') IS NOT NULL
     DROP TABLE [dbo].[AR_Invoice_DateFix_Backup];

IF OBJECT_ID(N'dbo.GL_Register_DateFix_Backup', N'U') IS NOT NULL
      DROP TABLE [dbo].[GL_Register_DateFix_Backup];
```

```
--- Backup Tables ---
Select * into AR_Invoice_DateFix_Backup from AR_Invoice
Select * into GL_Register_DateFix_Backup from GL_Register

--- Drop temp table if it exists ---
IF OBJECT_ID(N'dbo.gl_Reg_Number123', N'U') IS NOT NULL
      DROP TABLE [dbo].[gl_Reg_Number123];

--- Update Invoice with new date based on Cycle Batch ---
update i
set i.Invoice_Date =@NewDate ,Posting_Date = @NewDate
--select c.invoice_number,i.Invoice_Number, i.Invoice_Id,i.Invoice_Date,i.*
from AR_Cycle_Customer c
inner join AR_Invoice i on i.Invoice_Number = c.Invoice_Number
where c.Cycle_Id = @CycleID

--- Create Temp Table for GL Register Number ---
select
r.register_number,r.Primary_Register_Number,c.invoice_number,i.Invoice_Id,i.Invoice_Date
into gl_Reg_Number123
from AR_Cycle_Customer c
inner join AR_Invoice i on i.Invoice_Number = c.Invoice_Number
inner join GL_Register r on i.Register_Id =r.Register_Id
where c.Cycle_Id = @CycleID

--- Update Date and Accounting Period in GL_Register ---
update r
Set r.Date =@NewDate,r.Accounting_Period_Id = @AccountPeriod
--select r.Register_Id,r.Register_Number,r.*
from gl_Reg_Number123 n
inner join GL_Register r on r.Register_Number = n.Register_Number

--- Display GL records after fix ---
select r.Accounting_Period_Id,r.Register_Id,r.Register_Number,r.*
from gl_Reg_Number123 n
inner join GL_Register r on r.Register_Number = n.Register_Number

---Display Invoice records after fix ---
select c.invoice_number,i.Invoice_Number, i.Invoice_Id,i.Invoice_Date,i.*
from AR_Cycle_Customer c
inner join AR_Invoice i on i.Invoice_Number = c.Invoice_Number
where c.Cycle_Id = @CycleID

END
```