Dynamics AX 2012 - Data Refresh

For Parameter Tables 🡪 AX Data Refresh

For Transactional Tables 🡪 Test Data Transfer Tool (not supported in Production)

Source: <https://blogs.msdn.microsoft.com/axinthefield/ax-data-refresh-with-powershell/>

[1. Installation 2](#_Toc489430106)

[1.1. Create DynamicsAXTools database 2](#_Toc489430107)

[1.2. Running PowerShell Script 2](#_Toc489430108)

[2. Working with Environment Configurations 3](#_Toc489430109)

[2.1. Creating an Environment Backup 4](#_Toc489430110)

[2.1.1. Export Environment Configuration 4](#_Toc489430111)

[2.1.2. Stop AOS Services 4](#_Toc489430112)

[2.1.3. Restore DB Backup (\*.bak) 4](#_Toc489430113)

[2.1.4. Delete Live Data from Environment 5](#_Toc489430114)

[2.1.5. Restore Environment Configuration 5](#_Toc489430115)

[2.1.6. Start AOS Services 5](#_Toc489430116)

[3. How to add new tables 6](#_Toc489430117)

[4. Additional AX Tools 9](#_Toc489430118)

[4.1. Batch Jobs Maintenance 9](#_Toc489430119)

[4.2. Check SystemSequences Table 9](#_Toc489430120)

[4.3. AX Service Tools 9](#_Toc489430121)

[4.4. Delete Environment Store 9](#_Toc489430122)

[4.5. Export CSV Files 9](#_Toc489430123)

[4.6. Import CSV Files 10](#_Toc489430124)

[4.7. Update GUID - Globally Unique Identifier 10](#_Toc489430125)

# Installation

Download the PowerShell Tools for SQL Server (SQL Server Management Objects - SMO)

Windows PowerShell Extensions for Microsoft SQL Server (PowerShellTools.msi)

* SQL 2012 – <https://www.microsoft.com/en-us/download/details.aspx?id=29065>
* SQL 2014 – <https://www.microsoft.com/en-us/download/details.aspx?id=42295>
* SQL 2016 – <https://www.microsoft.com/en-us/download/details.aspx?id=52676>

# Create DynamicsAXTools database

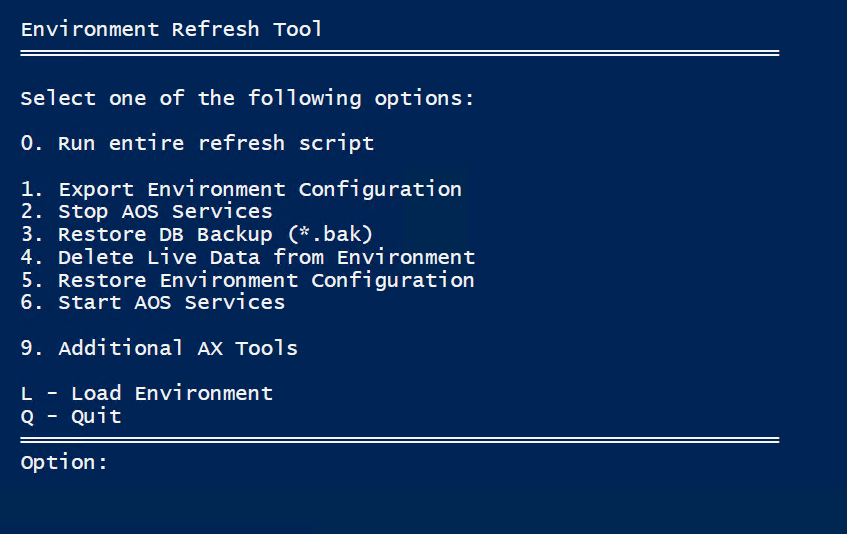
1. Run the script: RFR\_Database.sql
2. Update the script with you SQL Server Name and Database Name.

$ParamDBServer = 'DBServer\_Name' #Change SQL Server Name. Server\Instance, (local)

$ParamDBName = 'DynamicsAxTools' #Change DB Name (if you changed during creation)

# Running PowerShell Script

1. Open Windows PowerShell ISE.
2. F5 to start.



# Working with Environment Configurations

The script allows us to create environment stores to export parameters and re-import them after a database refresh.

There are two variables grouping tables to export and truncate only.

* Tables in **RFRTables** will be exported from the database.
* Tables in **RFRTables** and **RFRDelTables** will be truncated before restoring the original configuration.

$Script:RFRTables = @('USERINFO', 'SYSUSERINFO', 'SECURITYUSERROLE', 'SYSSERVERCONFIG', 'EPWEBSITEPARAMETERS', 'BATCHGROUP', 'BATCHSERVERGROUP', 'BATCHSERVERCONFIG', 'DMFPARAMETERS', 'SRSSERVERS', 'BIANALYSISSERVER', 'BIANALYSISSERVICESDATABASE', 'BICONFIGURATION', 'SYSCLUSTERCONFIG', 'SYSVERSIONCONTROLPARAMETERS', 'AIFWEBSITES', 'AIFSERVICE', 'AIFPORT')

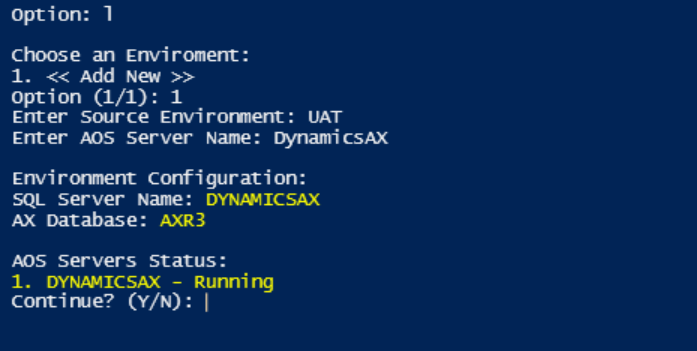
$Script:RFRDelTables = @('SYSSERVERSESSIONS', 'SYSCLIENTSESSIONS')

Creating an environment store is the easier way to keep different configurations on the same location. Each environment store has a unique name representing a specific environment or project phase. The AOS server status comes from SysServerSessions table in AX.

Options L and R to create, load or release a configuration.

* Function L – Load/Create an environment.
* Function R – Release active environment.

1. Enter the desired Environment Name
2. Enter an AOS Server name to collect database settings.

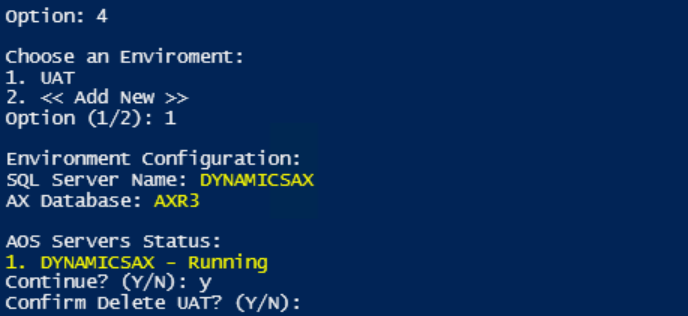


1. Before each step of the process if an environment is not loaded the script will ask for one.





1. To delete an environment. Go to Menu > Option 9 > Option 4



# Creating an Environment Backup

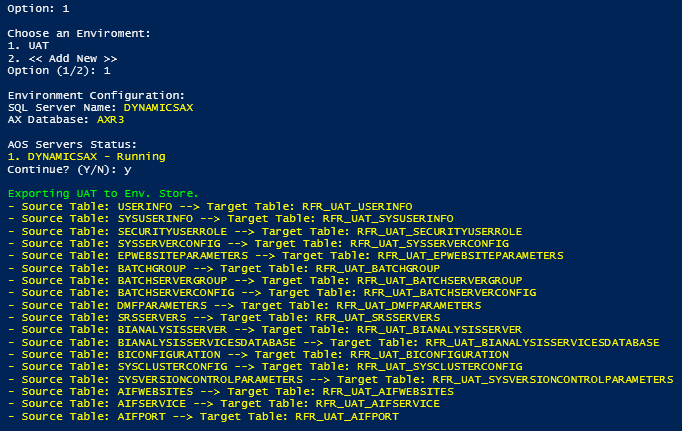
This option will export every AX tables set in $Script:RFRTables to DynamicsAxTools database.

Option 0 runs all tasks at once (from 1 to 6). If not, each individual task can be executed separately in the case of different teams are responsible for different steps of the process.

# Export Environment Configuration

Option 1 exports the designated AX tables to a backup database. It is important to export the data before running any database change. This configuration backup allows us to restore the same settings after a database restore.

1. Menu > Option 1
2. Select or Create an environment
3. Confirm (Yes or No)



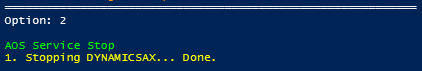
# Stop AOS Services

Option 2 shutdown all AOS servers on the environment.

The script selects AOS servers from:

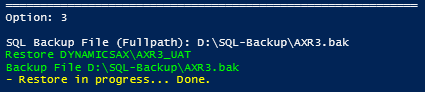
1. SysServerSession
2. or SysServerConfig (from environment backup)
3. or Manually enter server names





# Restore DB Backup (\*.bak)

Option 3 may be performed by the DBA team through SSMS. Although we still have an option to run it within the script.

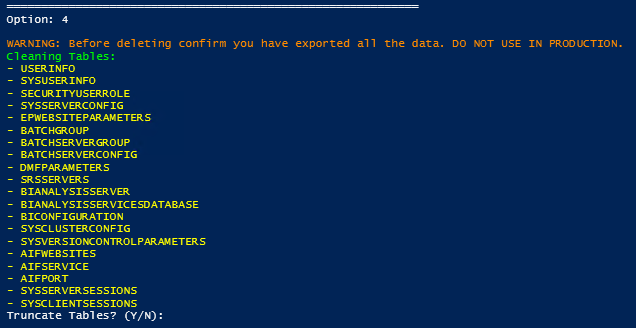


# Delete Live Data from Environment

**\*\* MAKE SURE THE SCRIPT IS NOT POITING TO ANY PRODUCTION SYSTEM \*\***

Option 4 executes a truncate statement for each table in **RFRTables** and **RFRDelTables**.

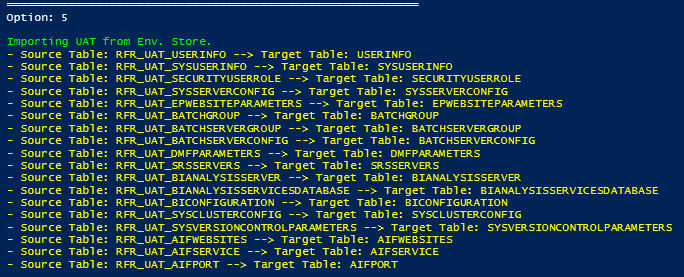
Tables in **RFRDelTables** are truncate only and no backups are taken. Use it for temporary data that might need to be deleted after a database restore; e.g. SysClientSessions and SysServerSessions.



**Before truncating the data, please review carefully the database connection and tables involved.**

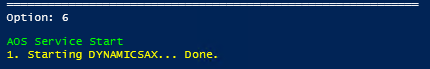
# Restore Environment Configuration

Option 5 restores the environment configuration to AX database.



# Start AOS Services

Option 6 starts all AOS servers on the environment.



# How to add new tables

The table group provided in the script should be enough to store some basic AX configuration, such as Users, Permissions, Batch, SSRS servers. It is possible to add or remove tables from the group to cover specific configuration needs.

**Basic Refresh Table Group:**

$Script:RFRTables = @('USERINFO', 'SYSUSERINFO', 'SECURITYUSERROLE', 'SYSSERVERCONFIG', 'EPWEBSITEPARAMETERS', 'BATCHGROUP', 'BATCHSERVERGROUP', 'BATCHSERVERCONFIG', 'DMFPARAMETERS', 'SRSSERVERS', 'BIANALYSISSERVER', 'BIANALYSISSERVICESDATABASE', 'BICONFIGURATION', 'SYSCLUSTERCONFIG', 'SYSVERSIONCONTROLPARAMETERS', 'AIFWEBSITES', 'AIFSERVICE', 'AIFPORT')

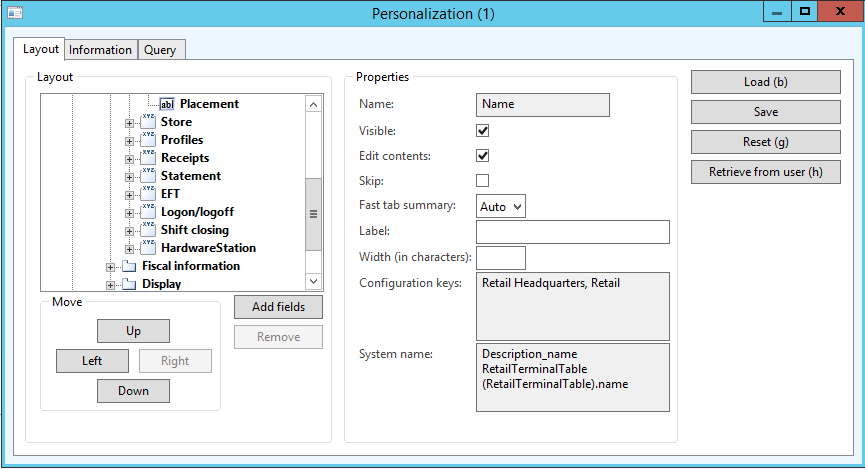
**Demo: Customizing for Retail Settings**

Two methods to find tables related to an AX process:

* AX Client Interface
* Using SQL Profiler

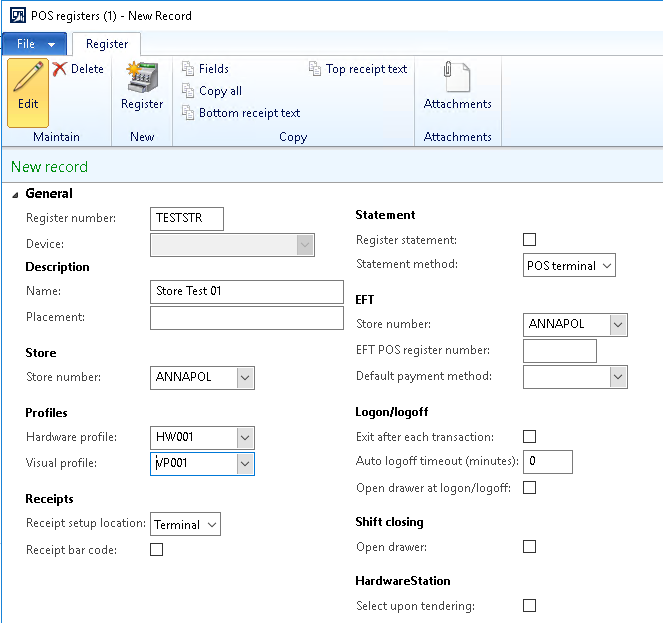
**AX Client Interface**

1. Open Retail/Setup/POS/POS registers
2. Right click on a field
3. Personalize
4. System Name

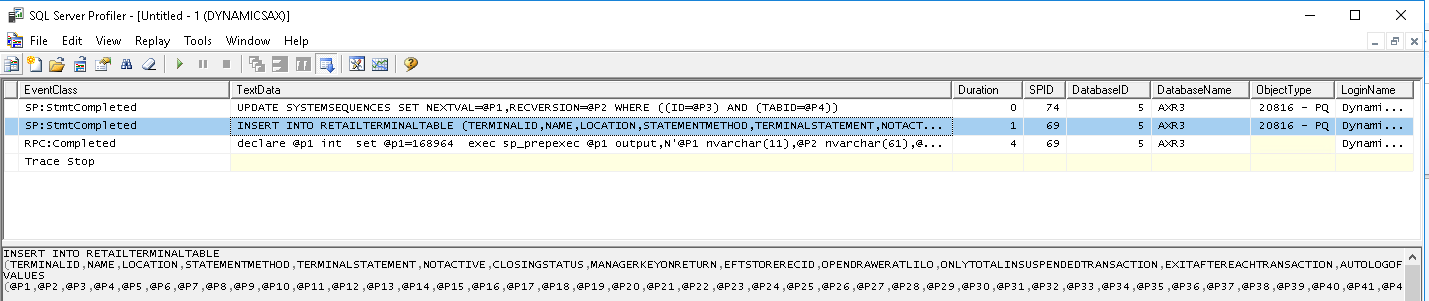


**SQL Profiler** - For a complex process with several tables.

1. Creating a new POS Terminal in AX



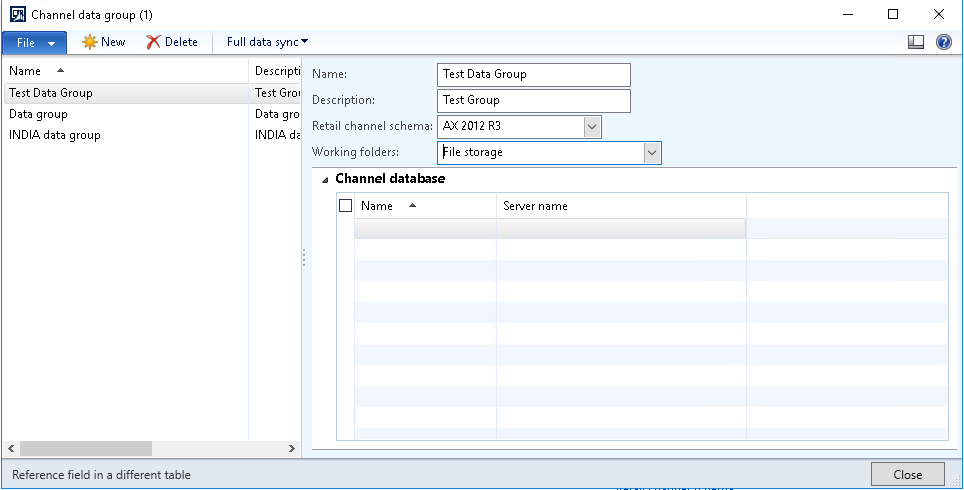
1. Collect a SQL trace



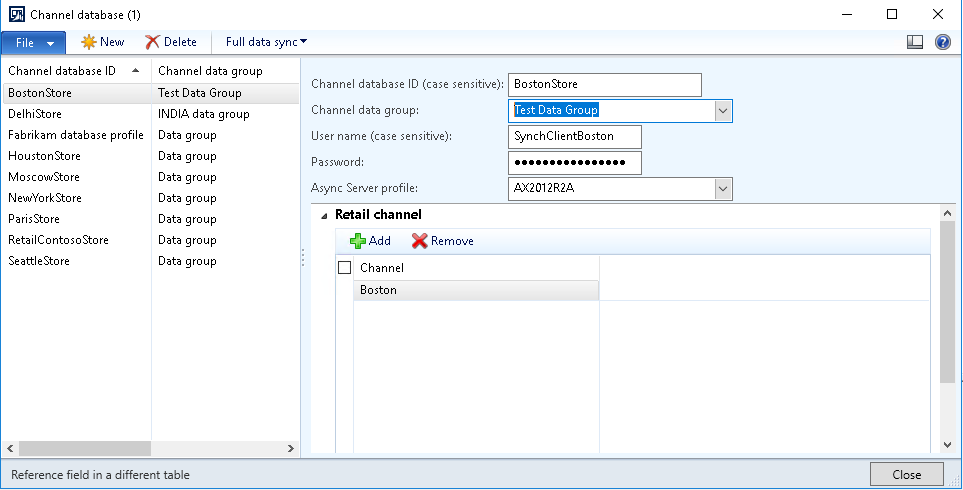
In the trace, a new POS terminal was created on RetailTerminalTable.

INSERT INTO RETAILTERMINALTABLE (TERMINALID,NAME,LOCATION,STATEMENTMETHOD,TERMINALSTATEMENT,NOTACTIVE,CLOSINGSTATUS,MANAGERKEYONRETURN,EFTSTORERECID,OPENDRAWERATLILO,ONLYTOTALINSUSPENDEDTRANSACTION,EXITAFTEREACHTRANSACTION,AUTOLOGOFFTIMEOUT,RETURNINTRANSACTION,…)

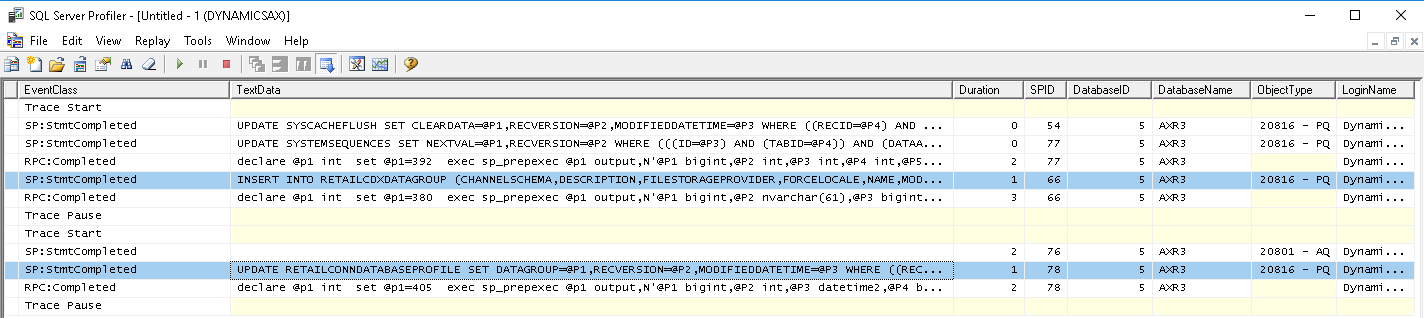
1. Creating a Channel Data Group.



1. Add a Store to the group.



1. Collect a SQL trace



In the trace, it is possible to see AX changing RetailCDXDataGroup and RetailConnDatabaseProfile.

INSERT INTO RETAILCDXDATAGROUP (CHANNELSCHEMA,DESCRIPTION,FILESTORAGEPROVIDER,FORCELOCALE,NAME,MODIFIEDDATETIME,RECVERSION,PARTITION,RECID)

UPDATE RETAILCONNDATABASEPROFILE SET DATAGROUP=@P1,RECVERSION=@P2,MODIFIEDDATETIME=@P3 WHERE ((RECID=@P4) AND (RECVERSION=@P5))

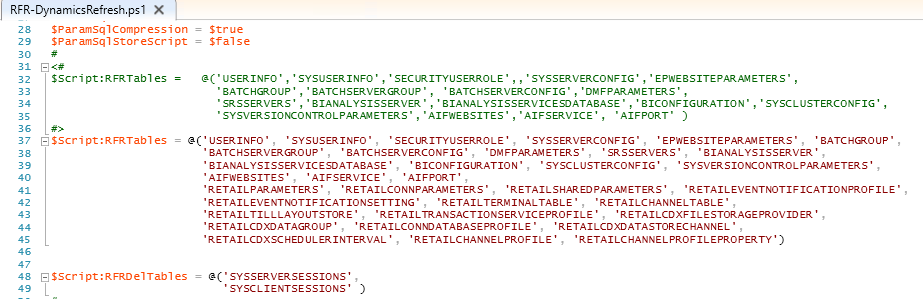
1. Repeat the process for all forms/tables.

On Contoso’s environment, the retail tables are:

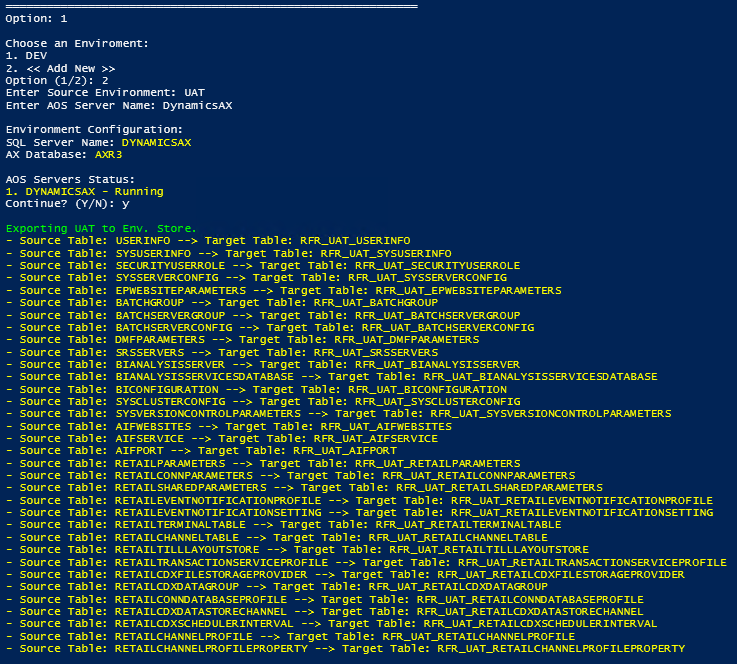
|  |  |
| --- | --- |
| AX Form | Tables |
| Retail Parameters | RetailParameters |
| Retail Scheduler Parameters | RetailConnParameters |
| Retail Shared Parameters | RetailSharedParameters |
| Email Notification Profiles | RetailEventNotificationProfile  RetailEventNotificationSetting |
| POS Registers | RetailTerminalTable |
| Retail Stores | RetailChannelTable  RetailTillLayoutStore |
| Real-Time Service Profile | RetailTransactionServiceProfile |
| Async Working Folders | RetailCDXFileStorageProvider |
| Channel Data Group | RetailCDXDataGroup |
| Channel Database | RetailConnDatabaseProfile |
| Store Channel Database | RetailCDXDataStoreChanne |
| Async Server Profile | RetailCDXSchedulerInterval |
| Retail Channel Profile | RetailChannelProfile  RetailChannelProfileProperty |

1. Change the table group in the refresh script.

$Script:RFRTables = @('USERINFO', 'SYSUSERINFO', 'SECURITYUSERROLE', 'SYSSERVERCONFIG', 'EPWEBSITEPARAMETERS', 'BATCHGROUP', 'BATCHSERVERGROUP', 'BATCHSERVERCONFIG', 'DMFPARAMETERS', 'SRSSERVERS', 'BIANALYSISSERVER', 'BIANALYSISSERVICESDATABASE', 'BICONFIGURATION', 'SYSCLUSTERCONFIG', 'SYSVERSIONCONTROLPARAMETERS', 'AIFWEBSITES', 'AIFSERVICE', 'AIFPORT', 'RETAILPARAMETERS', 'RETAILCONNPARAMETERS', 'RETAILSHAREDPARAMETERS', 'RETAILEVENTNOTIFICATIONPROFILE', 'RETAILEVENTNOTIFICATIONSETTING', 'RETAILTERMINALTABLE', 'RETAILCHANNELTABLE', 'RETAILTILLLAYOUTSTORE', 'RETAILTRANSACTIONSERVICEPROFILE', 'RETAILCDXFILESTORAGEPROVIDER', 'RETAILCDXDATAGROUP', 'RETAILCONNDATABASEPROFILE', 'RETAILCDXDATASTORECHANNEL', 'RETAILCDXSCHEDULERINTERVAL', 'RETAILCHANNELPROFILE', 'RETAILCHANNELPROFILEPROPERTY')



1. Run the script to export the new table group.



# Additional AX Tools

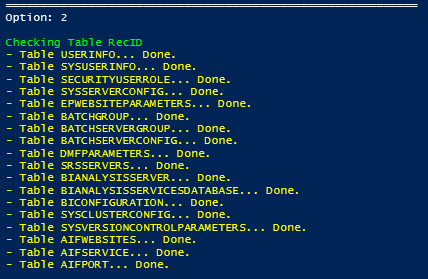
# Batch Jobs Maintenance

The Batch Jobs Menu allow us to run some basic batch maintenance after a database refresh:

1. Disable all batch jobs – Moves all batch groups from all batch servers. It avoids a batch job from a different environment to run in production.
2. Move Batch Groups to a different server (EnableBatch is ON) – Moves all batch groups to a specific server, where the option Enable Batch is true.
3. Clean Batch History – Batch History cleanup.

# Check SystemSequences Table

Check SystemSequences provides us the possibility to check RecIDs from imported tables to avoid duplicates. AOS server restart is required.

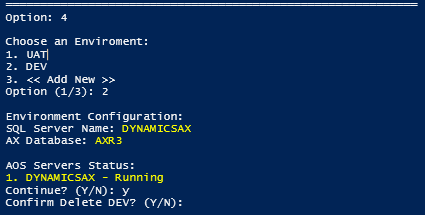


# AX Service Tools

1. Start AOS Services
2. Stop AOS Services
3. Restart AOS Services
4. Check AOS Services Status

# Delete Environment Store

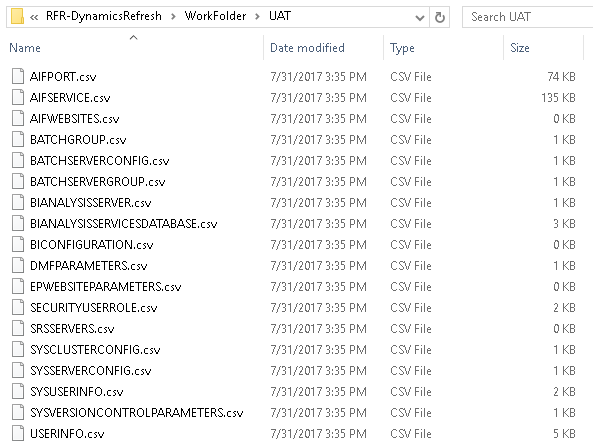
Delete Environment Store removes the active environment from the database.



# Export CSV Files

Export to CSV creates a stored procedure in the database to generate INSERT statements for each table. This process runs slow on large tables (1000+ rows).

All tables in RFRTables are exported to the *Script\_Location\WorkFolder\EnvironmentName*.



SQL Binary or Spatial Data Objects are data types not supported to export to CSV. (Timestamp, Geography, Geometry, Hierarchyid, Image, Binary, Varbinary)

# Import CSV Files

Import CSV reads all files in *Script\_Location\WorkFolder\EnvironmentName* and validates against RFRTables variable. Before importing a csv file, the script truncates the table to avoid PK violations.

# Update GUID - Globally Unique Identifier

Updating GUID is not common but avoids AX cache issue. If a user has access to test and production on the same box both environments most likely will share the same AUC files, which can cause problems if they have different code.