



JET GLOBAL
DATA TECHNOLOGIES



**Using Jet Analytics with
Microsoft Dynamics 365
for Finance and Operations**

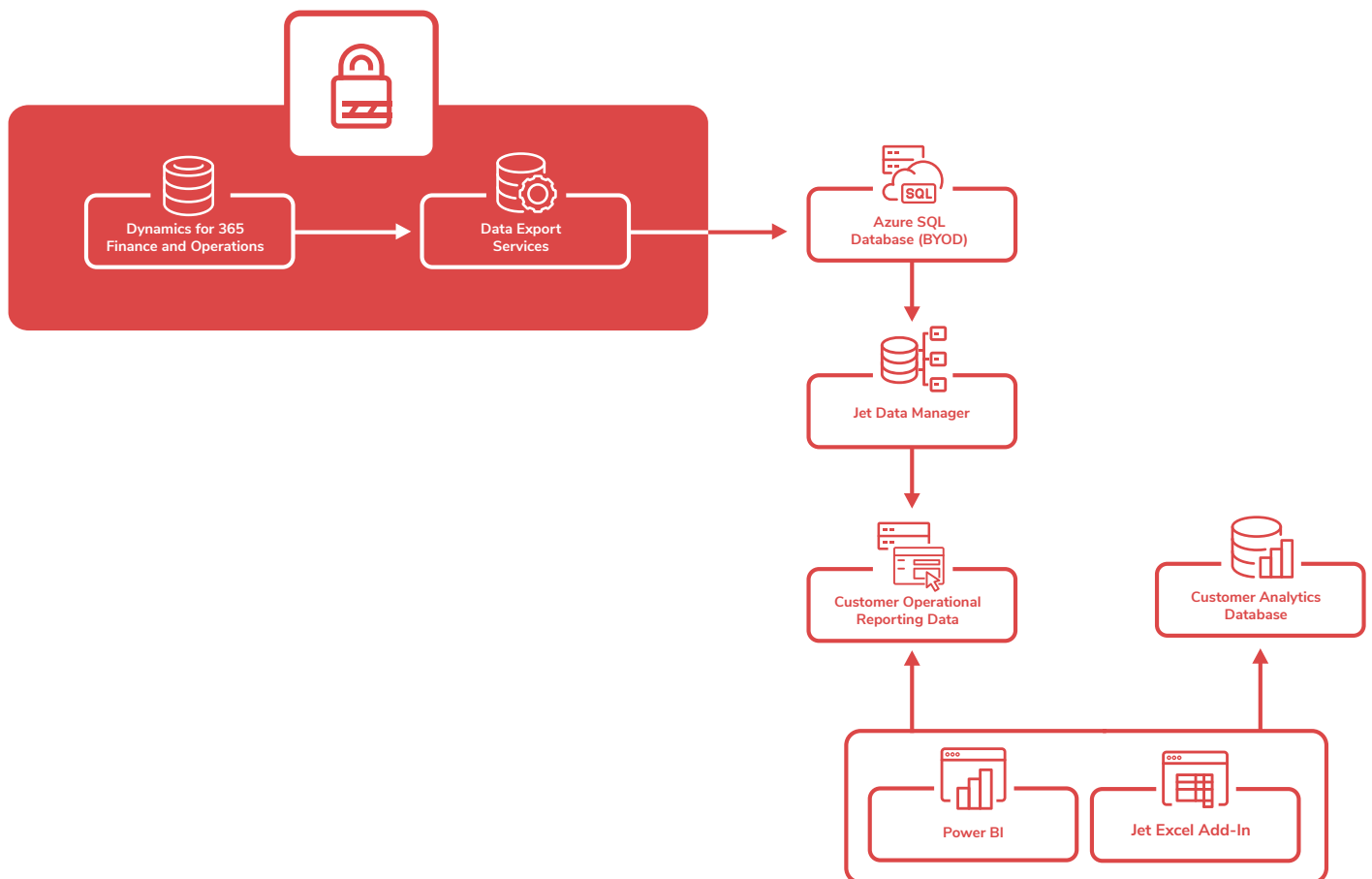
Table of Contents

- Overview.....3
- Installation4
 - Data Entities.....4
 - Import Data Entities4
 - Publish Data Entities.....6
 - Setting up the BYOD database.....6
 - Exporting data into your BYOD database10
 - Setting up data export with templates.....11
 - Removing a Jet Data Entity12
 - Building a New Data Entity12
- Jet Analytics Installation.....14

Overview

With Microsoft Dynamics 365 for Finance and Operations being primarily in the Cloud, some changes are required to the approach on how businesses gain access to the back-end data. Previously, business users had full direct access to all SQL tables and fields in AX 2012 and earlier. For Jet Analytics, the Jet Data Manager (JDM) could connect directly to this SQL database for simple and fast access to the data. However, since direct access to the SQL database is no longer available with Finance and Operations, we have created a project that enables you to deploy the JDM in a *bring your own database* (BYOD) model.

This document will cover how to setup and deploy Jet Analytics for Microsoft Dynamics 365 for Finance and Operations using this model. Below is a diagram of this deployment concept.



Installation

Data Entities

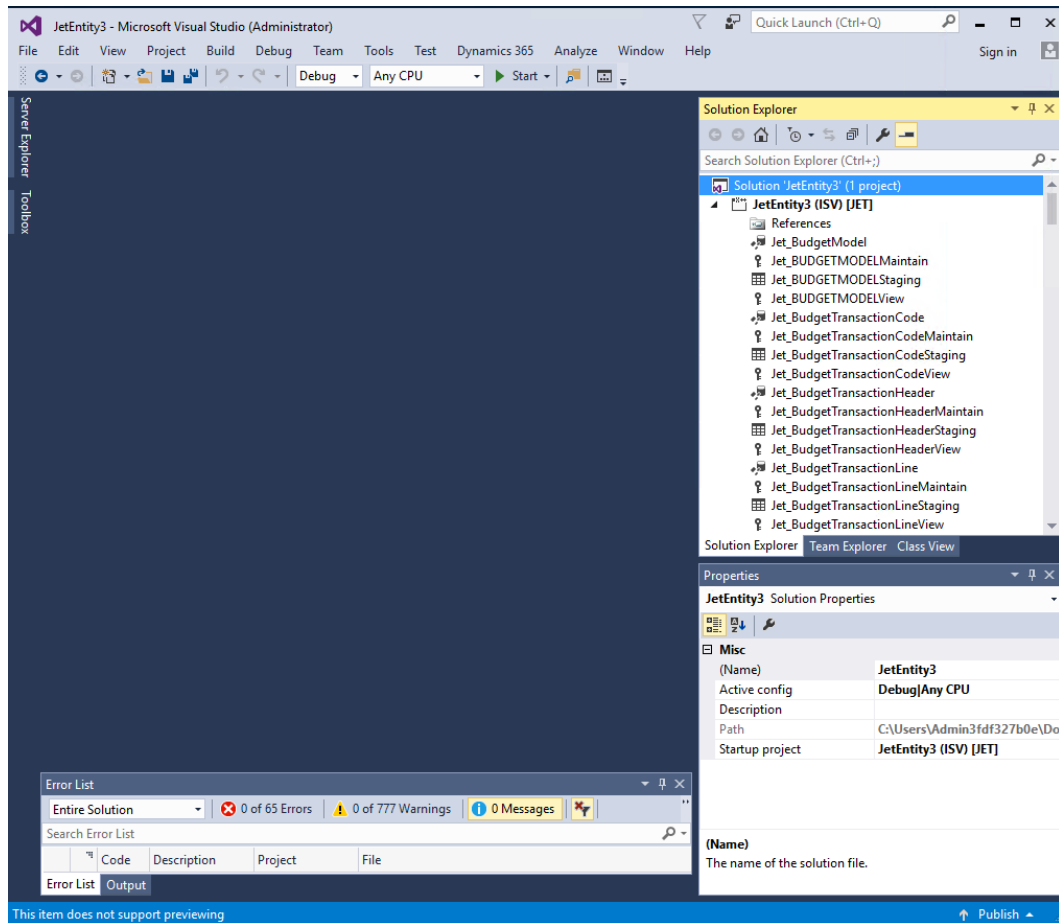
A data entity is an abstraction from the physical implementation of database tables. For example, in normalized tables, most of the data for customers might be stored in a Customer table, and then the rest might be spread across a small set of related tables. In this case, the data entity for the customer concept appears as one de-normalized view, in which each row contains all the data from the customer table and its related tables. A data entity encapsulates a business concept into a format that makes development and integration easier. The abstracted nature of a data entity can simplify application development and customization. Later, the abstraction also insulates application code from the inevitable churn of the physical tables between versions. To summarize: Data entities provide conceptual abstraction and encapsulation (de-normalized view) of underlying table schemas to represent key data concepts and functionalities.

In our deployment model, we have created 1 data entity for each table that we will be using. We also will add all fields from the table to the data entity so that if we need another field we will not have to re-generate the data entities. This also allows us to setup incremental loading on the data export side.

Import Data Entities

First, you will need to import the Jet Data Entities into your Finance and Operations environment. In the installation files you will find a file with the .xpp extension. This will house the definition of the data entities. To import this file, you will need to login to your Finance and Operations Development environment. You can access this by:

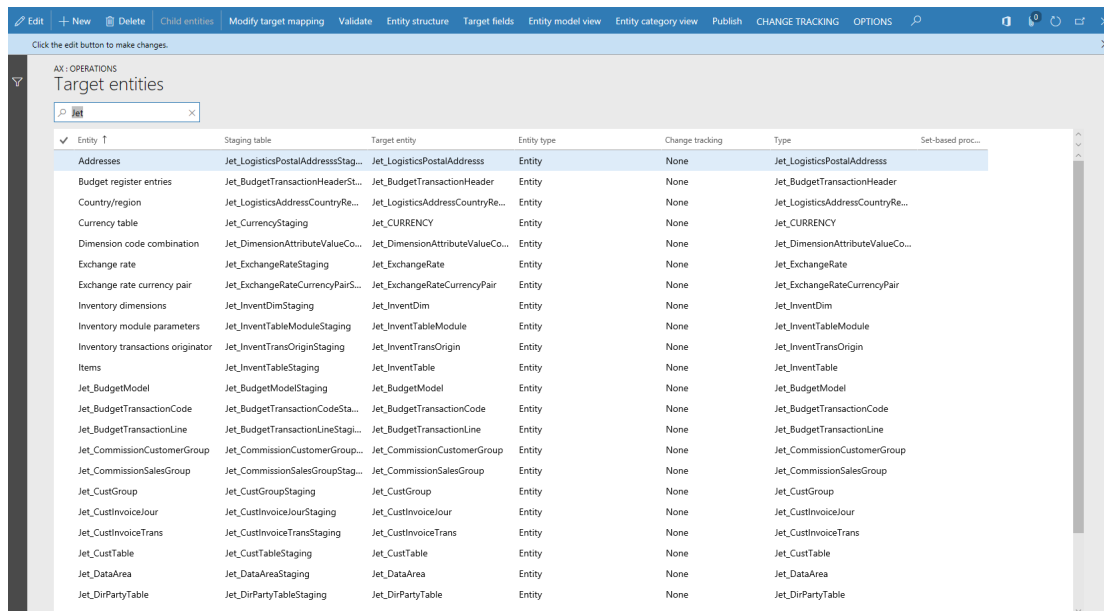
1. Login to Lifecycle Services
2. Click on the Project you are looking to use.
3. Click on the Environment you need to import the data entities into.
4. Click the VM to open the remote desktop and login.
5. Click Start – Find Visual Studio and then run it as an administrator.
6. Click on the Finance and Operations menu option and select Import Project.
7. Find the .xpp file that you downloaded from Jet and click OK
8. Once loaded it should look like this:



- Now right click on the Solution 'JetEntity3' and choose "Build". This will push all the Jet Data Entities into your Finance and Operations Environment. If there are any errors you will have to resolve them. If you need to re-publish the data entities that have already been published you might need to delete them from Finance and Operations and re-publish them. To delete the data entities, go to the section of this document that explains how to remove data entities from Finance and Operations.

Publish Data Entities

Once you have built your project in Visual Studio the data entities should now all show up in Finance and Operations. If you login to your Finance and Operations environment and go to *Data Management – Data Entities* you should have a list off all of your data entities. Set a filter on the Staging table name of Jet and you should see an entire list of all the Jet Data Entities you just built in Visual Studio.



Entity	Staging table	Target entity	Entity type	Change tracking	Type	Set-based proc...
Addresses	Jet_LogisticsPostalAddressStag...	Jet_LogisticsPostalAddress	Entity	None	Jet_LogisticsPostalAddress	
Budget register entries	Jet_BudgetTransactionHeaderSt...	Jet_BudgetTransactionHeader	Entity	None	Jet_BudgetTransactionHeader	
Country/region	Jet_LogisticsAddressCountryRe...	Jet_LogisticsAddressCountryRe...	Entity	None	Jet_LogisticsAddressCountryRe...	
Currency table	Jet_CurrencyStaging	Jet_CURRENCY	Entity	None	Jet_CURRENCY	
Dimension code combination	Jet_DimensionAttributeValueCo...	Jet_DimensionAttributeValueCo...	Entity	None	Jet_DimensionAttributeValueCo...	
Exchange rate	Jet_ExchangeRateStaging	Jet_ExchangeRate	Entity	None	Jet_ExchangeRate	
Exchange rate currency pair	Jet_ExchangeRateCurrencyPairS...	Jet_ExchangeRateCurrencyPair	Entity	None	Jet_ExchangeRateCurrencyPair	
Inventory dimensions	Jet_InventDimStaging	Jet_InventDim	Entity	None	Jet_InventDim	
Inventory module parameters	Jet_InventTableModuleStaging	Jet_InventTableModule	Entity	None	Jet_InventTableModule	
Inventory transactions originator	Jet_InventTransOriginStaging	Jet_InventTransOrigin	Entity	None	Jet_InventTransOrigin	
Items	Jet_InventTableStaging	Jet_InventTable	Entity	None	Jet_InventTable	
Jet_BudgetModel	Jet_BudgetModelStaging	Jet_BudgetModel	Entity	None	Jet_BudgetModel	
Jet_BudgetTransactionCode	Jet_BudgetTransactionCodeSta...	Jet_BudgetTransactionCode	Entity	None	Jet_BudgetTransactionCode	
Jet_BudgetTransactionLine	Jet_BudgetTransactionLineStagi...	Jet_BudgetTransactionLine	Entity	None	Jet_BudgetTransactionLine	
Jet_CommissionCustomerGroup	Jet_CommissionCustomerGroup...	Jet_CommissionCustomerGroup	Entity	None	Jet_CommissionCustomerGroup	
Jet_CommissionSalesGroup	Jet_CommissionSalesGroupStag...	Jet_CommissionSalesGroup	Entity	None	Jet_CommissionSalesGroup	
Jet_CustGroup	Jet_CustGroupStaging	Jet_CustGroup	Entity	None	Jet_CustGroup	
Jet_CustInvoiceJour	Jet_CustInvoiceJourStaging	Jet_CustInvoiceJour	Entity	None	Jet_CustInvoiceJour	
Jet_CustInvoiceTrans	Jet_CustInvoiceTransStaging	Jet_CustInvoiceTrans	Entity	None	Jet_CustInvoiceTrans	
Jet_CustTable	Jet_CustTableStaging	Jet_CustTable	Entity	None	Jet_CustTable	
Jet_DataArea	Jet_DataAreaStaging	Jet_DataArea	Entity	None	Jet_DataArea	
Jet_DirPartyTable	Jet_DirPartyTableStaging	Jet_DirPartyTable	Entity	None	Jet_DirPartyTable	

Once you have verified that your data entities exist you will need to publish them to the BYOD server. Most of the following text was taken from a Microsoft article found here: <https://docs.microsoft.com/en-us/dynamics365/unified-operations/dev-itpro/analytics/export-entities-to-your-own-database>

Setting up the BYOD database

This topic explains how administrators can export data entities from Microsoft Dynamics 365 for Finance and Operations into their own Microsoft Azure SQL database. This feature is also known as bring your own database (BYOD). The BYOD feature was released in Microsoft Dynamics AX with platform update 2 (August 2016). Minor improvements and bug fixes have been included in subsequent platform updates.

The BYOD feature lets administrators configure their own database, and then export one or more data entities that are available in Finance and Operations into it. Specifically, this feature lets you complete these tasks:

- Define one or more SQL databases that you can export entity data from Finance and Operations into.
- Export either all the records (full push) or only the records that have changed (incremental push).
- Use the rich scheduling capabilities of the Finance and Operations batch framework to enable periodic exports.
- Access the entity database by using Transact-SQL (T-SQL), and even extend the database by adding more tables.

Finance and Operations Finance and Operations doesn't allow T-SQL connections to the production database. If you're upgrading from a previous version of Finance and Operations, and you have integration solutions that require direct T-SQL access to the database, BYOD is the recommended upgrade path.

Creating a SQL database

Before you can configure the entity export option and use the BYOD feature, you must create a SQL database by using Azure portal.

For one-box development environments, you can create a database in the local Microsoft SQL Server database. However, this database should be used only for development and testing purposes. For production environments, you must create a SQL database.

You should also create a SQL user account for sign-in to the database. Write down the server name, database name, and the SQL user ID and password. You will use this information when you configure the entity export option in the next section.

Configuring the entity export option

Start the Finance and Operations Finance and Operations client and then, in the Data Management workspace, select the Configure Entity export to database tile.

If you've configured any databases, a list is shown. Otherwise, you must configure a new database. In this case, select New, and then enter a unique name and a description for the new database.

Note that you can export entities into multiple databases.

Enter the connection string in the following format:

Data Source=<logical server name>,1433; Initial Catalog=<your DB name>; Integrated Security=False; User ID=<SQL user ID>; Password=<password>

In this connection string, the logical server name should resemble nnnn.database.windows.net. You should be able to find the logical server name in Azure portal. The following illustration shows an example of a connection string.

Source name	Description	Type	Default extension
ExportDB	Export to BYOD	Azure SQL DB	

Configure

Connection string
Data Source=serverName;Initial Catalog=databaseName;Integrated Security=False;User ID=userid;Password=***

Validate Hide

Create clustered column store indexes
Yes

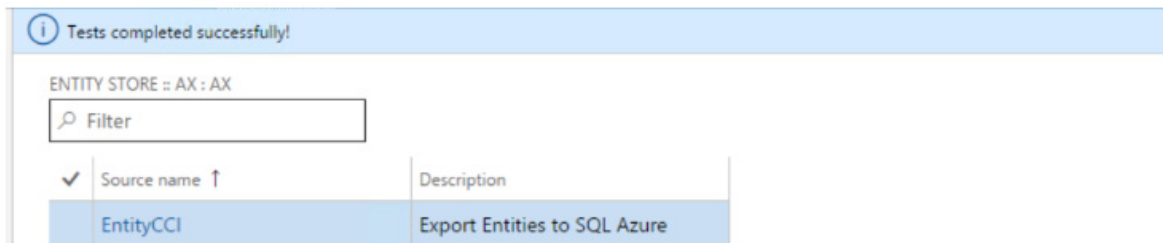
Enable triggers in target database
No

Select Validate, and make sure that the connection is successful.

The "Create clustered column store indexes" option optimizes the destination database for selected queries by defining clustered column store indexes for entities that are copied from Finance and Operations. However, CCIs are currently supported only on SQL premium databases. If you wish to enable this option you must create a SQL premium database.

The “Enable triggers in target database” option sets export jobs to allow SQL triggers in the target database. This option lets you hook downstream processes into the trigger to orchestrate actions that must be started after records have been inserted. One trigger is supported per bulk insert operation. The size of the bulk insert is determined by the Maximum insert commit size parameter in the Data Management framework.

When the validation is passed, the database that you configured for entity export appears in the list of databases as shown in the following illustration.



You can now publish one or more entities to the new database by selecting the Publish option on the menu.

Publishing the entity schema to the database

The Publish page enables several scenarios:

- Publish new entities to the database.
- Delete previously published entities from the database. (For example, you might want to re-create the schema).
- Compare published entities with the entity schema in Finance and Operations. (For example, if new fields are added to Finance and Operations later, you can compare the fields with your database schema).
- Configure change tracking functionality that enables incremental updates of your data.

The following sections discuss each option:

Publish

The Publish option defines the entity database schema on the destination database. When you select one or more entities, and then select the Publish option, a batch job is started. This job creates the entities in the destination database. When the database definition job is completed you receive a message which can be accessed by using the bell symbol in the upper right. The actual data update occurs when you export data. At this point, you’re just creating the schema.

Drop entity

The Drop entity option deletes the data and the entity definition from the destination database.

Compare source names

The Compare source names option lets you compare the entity schema in the destination with the entity schema in Finance and Operations. This option is used for version management. You can also use this option to remove any unwanted columns from the destination table.

Configure change tracking

Change tracking is a feature that is provided in SQL Server and SQL Databases. Change tracking enables the database to track changes that are made on tables. The system uses change tracking to identify changes that are made to tables as transactions are performed in Finance and Operations.

The Change tracking option on the Publish page lets you configure how changes are tracked on the underlying entity.

Change Tracking	Entity Name	Table Name	Source Table	Target Table	Entity Type	Change Tracking	Entity Name
No	Applicant certificate competency	HcmApplicantCertificateStaging	HcmApplicantCertificateEntity	Entity	None	Hcr	
No	Applicant course competency	HcmApplicantCourseStaging	HcmApplicantCourseEntity	Entity	None	Hcr	
No	Applicant education competency	HcmApplicantEducationStaging	HcmApplicantEducationEntity	Entity	None	Hcr	
No	Applicant exam competency	HcmApplicantExamStaging	HcmApplicantExamEntity	Entity	None	Hcr	
No	Applicant position of trust comp...	HcmApplicantTrustedPositionSt...	HcmApplicantTrustedPositionEn...	Entity	None	Hcr	
No	Applicant professional experien...	HcmApplicantProfessionalExperi...	HcmApplicantProfessionalExperi...	Entity	None	Hcr	
No	Applicant project experience co...	HcmApplicantProjectRoleStaging	HcmApplicantProjectRoleEntity	Entity	None	Hcr	
No	Applicant skill competency	HcmApplicantSkillStaging	HcmApplicantSkillEntity	Entity	None	Hcr	
<input checked="" type="checkbox"/>	Applicants	HcmApplicantStaging	HcmApplicantEntity	Entity	None	Hcr	

The following table describes the change tracking options that are available:

Enable primary table

An entity consists of several tables. Select this option to track all changes that are made to the primary table of the entity. When changes are made to the primary table, the corresponding record is inserted into or updated in the destination database. Although data from the whole entity is written to the destination table, the system triggers the insert or update option only when the primary table is modified.

Enable entire entity

Select this option to track all changes to the entity. These changes include changes to all the tables that make up the entity. When changes are made to the entity, corresponding updates are made to the destination.

Enable custom query

This option lets a developer provide a custom query that the system runs to evaluate changes. This option is useful when you have a complex requirement to track changes from only a selected set of fields. You can also select this option when the entities that will be exported were built by using a hierarchy of nested views. For more information, see Enable change tracking for an entity by using a custom query.

For the change tracking functionality to work, you must enable the Change tracking option in the Finance and Operations database. This option is enabled by default.

If you republish an entity that exists in the destination database, the system warns you that existing data will be deleted because of the new operation.

When you confirm the publish operation, the system publishes the schema to the database, and you're notified when the operation is completed.

By selecting the "Show published only" option on the Publish page, you can show only the entities that were published to a given destination database. The Publish function creates the entity schema in the database. You can navigate to the database and see the table schemas that were created, together with corresponding indexes.

Exporting data into your BYOD database

After entities are published to the destination database, you can use the Export function in the Data Management workspace to move data. The Export function lets you define a Data movement job that contains one or more entities.

You can use the Export page to export data from Finance and Operations into many target data formats, such as a comma-separated values (CSV) file. This page also supports SQL databases as another destination.

The screenshot shows the 'Export' configuration interface. On the left, under 'JOB DETAILS', there are several input fields and dropdown menus: 'Name' (OneTimeExport), 'Target data format' (EntityCCI), 'Entity name' (empty), 'Use sample file' (No), 'Skip staging' (Yes), 'Default refresh type' (Incremental push only), and 'Generate data package' (No). On the right, under 'SELECTED FILES AND ENTITIES', there is a search filter, a list of entities, and a selection box for 'Applicants'. Below the entity list are 'View map' and 'Filter' options. An 'Add entity' button is located at the bottom right of the 'JOB DETAILS' section.

When you add an entity for data export, you can select to do an incremental export (which is also known as incremental push) or a full push. For incremental push to work, you must enable the Change tracking option in the Finance and Operations database and specify an appropriate change tracking option, as described earlier in this topic.

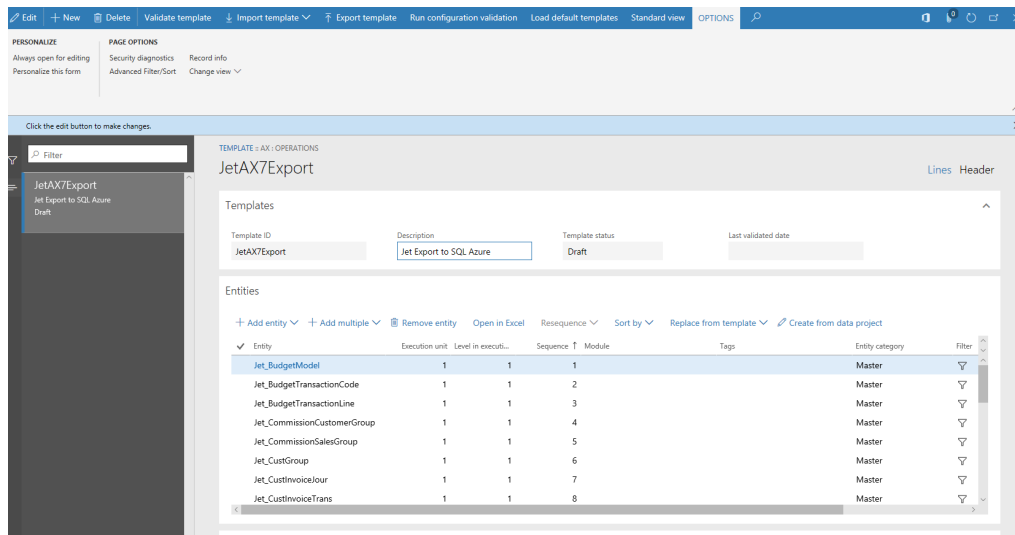
If you select to do an incremental push, whenever a new record is inserted, or a record is added, the corresponding change will be reflected in the destination entity. In Microsoft Dynamics 365 for Finance and Operations with platform update 8, records that are deleted in the source aren't updated in the destination.

Full push truncates the table and inserts all the records from the selected entity. You can create a data project that has multiple entities. You can schedule this data project to run by using the Finance and Operations batch framework. You also schedule the data export job to run on a periodic basis by selecting the Export in batch option.

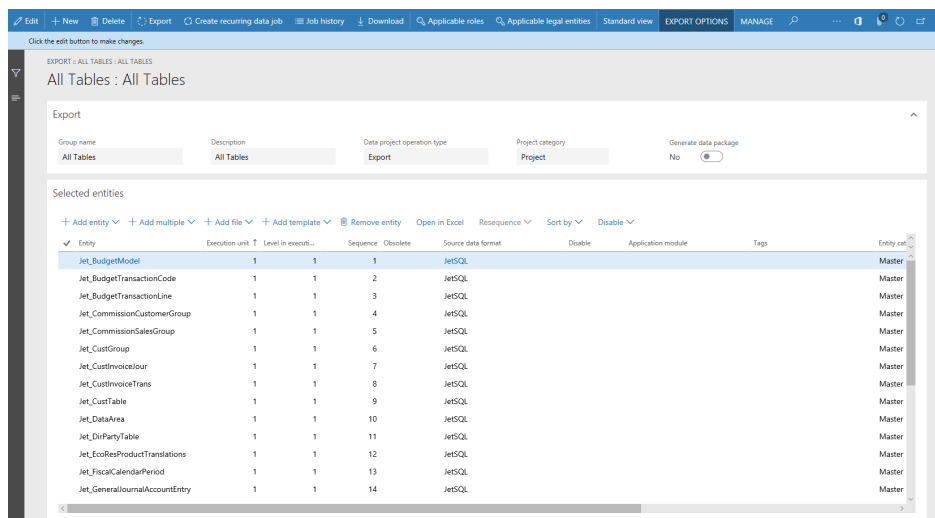
When we created our Jet specific entities we have created them in a way that all company data will be exported. When you publish in Finance and Operations you only have to create the export in one company and it will export all the companies in your Finance and Operations instance. DataReadId has been added to all the company specific tables so you can split out what data pertains to each company.

Setting up data export with templates

In the previous section we went through setting up the export with each entity. With the export you can also create templates that will have all the entities already added to the export so that you don't have to add each entity individually. In the download files you will find a file of "Jet_XXXXXXXX.zip". These files contain the entity export templates. To use the templates you will need to import them first. To import a template go to "Data Management – Templates". Then click "Import Template" on the top ribbon. Select the .zip file to import the template and you now have a template that should look like this:



Now go back to "Data Management – Export" and then click "Add Template". Pick the template you just created and select the correct data export database. It will add all the entities from the template so you don't have to select each individual one.



Now you can just click export and it will export all the data entity data to your BYOD database. You can watch the progress on the Data Management page under job history. There you can see how many records have been exported and if any error out you can see the errors. After you setup the export job you can also schedule it to be run at a certain time so it can run every night or at a time that you choose.

Removing a Jet Data Entity

If you want to modify a data entity you will need to remove it from Finance and Operations, go back into Visual Studio make the changes, and rebuild the project.

Click on “Data Management – Configure Entity to Export Database”. Select the source name where you want to remove the entity from. Select “Publish” and in the list check “Show published only”. This will give you a list of all entities that are published in this source. Check the boxes for the entities that you wish to un-publish. This will remove all data and table schema from the data source. Make sure you check the correct data entities that you are removing.

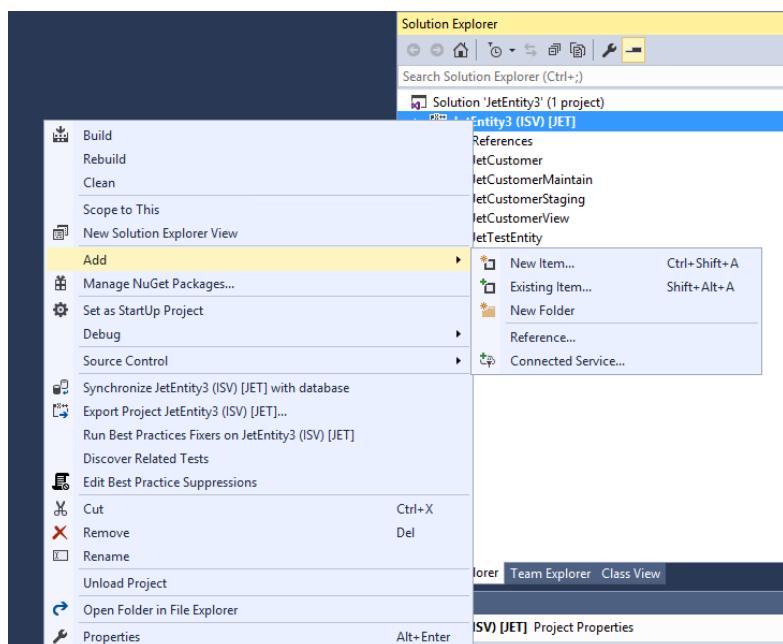
Now that this entity no longer exists in the BYOD database you can remove the data entity definition by going to “Data Management – Entities”. Filter on the staging name of “Jet” and it will list all the Jet data entities. Select the ones you want to remove and click “Delete”. This will remove the entity from the Finance and Operations environment.

You can now go back to the Visual Studio environment make changes to the entities and build the project again. It will then push up the entities to the Finance and Operations environment again. When you build the project and already have some of the data entities published you may receive errors on those entities. That is because those data entities have already been published to your Finance and Operations environment. Just make sure there are no errors on the data entity that you have updated.

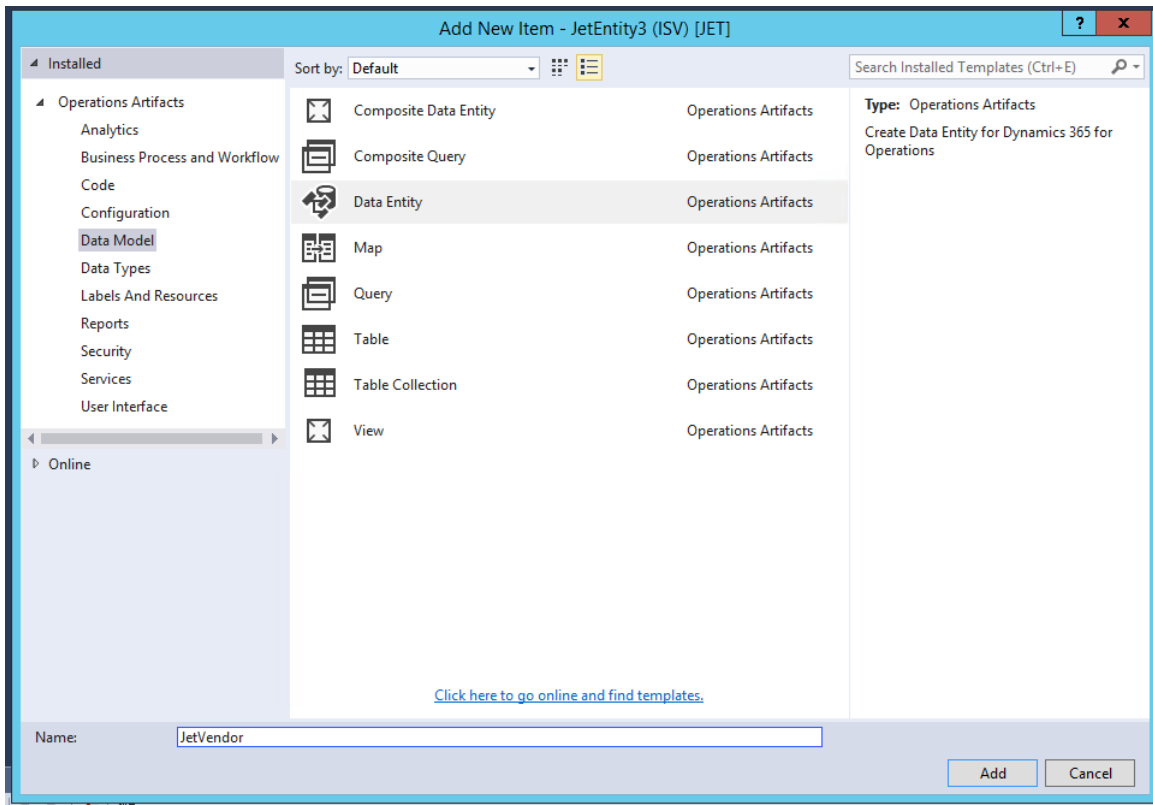
Building a New Data Entity

There may be times when you need to create a new data entity from scratch. Below are the steps to do that. First open Visual Studio in the development environment. Be sure to run it as an administrator.

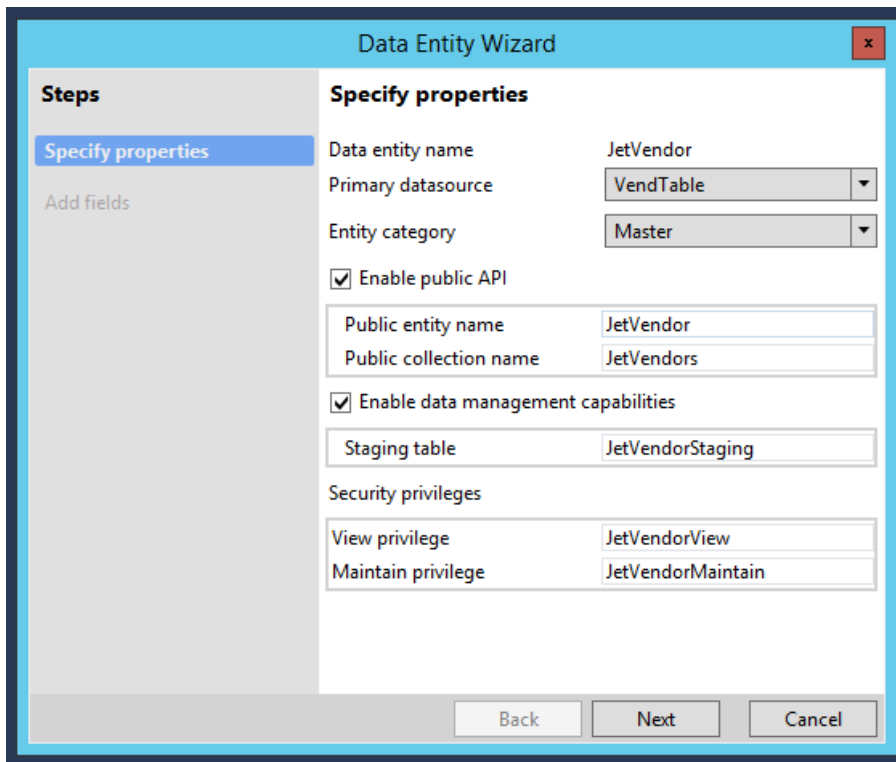
1. In Solution Explorer right click the project JetEntity3(ISV)[JET] and chose Add → New Item.



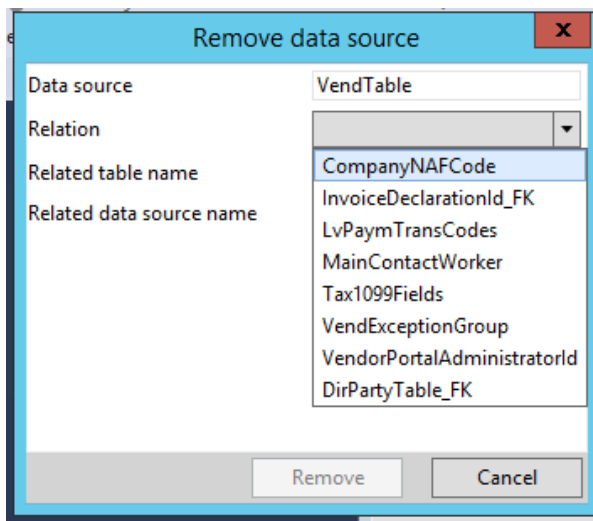
2. Select “Data Model – Data Entity”. Give it a proper name. The example below has entities prefixed with “Jet” and then the table Name (i.e.: JetVendor).



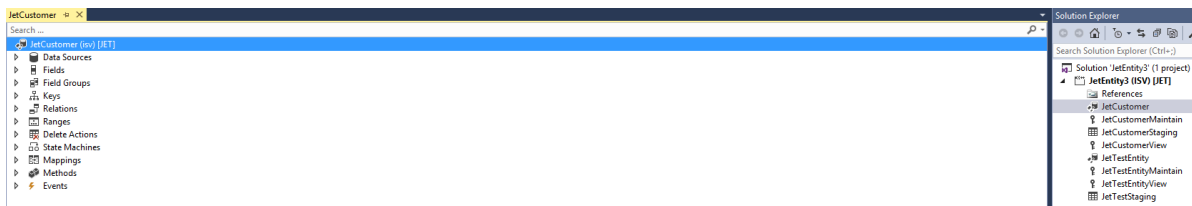
3. Select the Name of the Table from the Dropdown in the Primary datasource. This should have a full list of all the tables in Finance and Operations.



- Click Next and you should see a list of all the fields in the table that was selected from the Primary datasource area. Next click remove data source and select in the relation drop down the first table to remove. Do this for all the tables to remove all related tables. This is so we only get the primary table.



- Once you have removed all of the related tables click finish and your entity is now built.
- If you need to modify the entity double click the Entity object and you will see an expandable list of everything in the entity.



- Once you are done with the entity you need to right click on the project and select Build Solution. This will publish the Entities up to Finance and Operations. If you log into Finance and Operations you will find the entities and you can schedule them to be deployed over to a SQL Azure site.

Notes:

- Make sure the primary key on the entity is set up properly. The wizard sometimes does not create the entity key properly.
- If there is a RecId and/or DataAreald field in the base table it is a best practice to always add them. You will have to add them with the names of RecId1 and DataAreald1 since they are both reserved words in the Finance and Operations development environment.
- If you make changes to the data entity right-click the data entity name in the design window and select regenerate staging table. This will ensure the staging table definition is in line with the data entity changes.
- Since the Jet entities are for reporting only the read-only property is set to "Yes".

Jet Analytics Installation

To install Jet Analytics follow the installation instructions found in our [online help center article](#).