

# Use machine learning-based Microsoft Cognitive Services for product recommendations

This document explains how you can use the new Cortana Intelligence services management module for product recommendations.

White paper

November 2016

Send feedback.

[www.microsoft.com/dynamics/ax](http://www.microsoft.com/dynamics/ax)



# Contents

<b>Overview and terminology</b>	<b>4</b>
Cortana Intelligence Suite	4
Microsoft Cognitive Services	5
Recommendations API	5
<b>Installing and configuring prerequisites</b>	<b>5</b>
<b>Cortana Intelligence services management</b>	<b>6</b>
Recommendations API – Data model	7
Model	7
Catalog	7
Usage file	8
Model build	8
Recommendations API – Integration workflow	9
Step 1. Set up global parameters	9
Step 2. Create Recommendation models	10
Step 3. Upload Catalog and Usage datasets	12
Step 4. Create Recommendation model builds (Train the model)	14
Step 5. Consume recommendations (Score the model)	16
Automated Recommendations API builds and data synchronization	17
Automated ingestion of Catalog and Usage datasets	18
Build automation	20
<b>Frequently asked questions</b>	<b>22</b>
<b>Appendix A: Provisioning an Azure Cognitive Services account</b>	<b>23</b>
<b>Appendix B: Using out-of-box entities to generate Catalog and Usage datasets</b>	<b>24</b>
Step 1. Configure DIXF parameters	25
Step 2. Set up target entities	25
Step 3. Set up the data source format for export	25
Step 4. Generate the source mapping for the entity	26
Step 5. Export data for Recommendation entities	27
<b>Appendix C: Customizing out-of-box Cortana Intelligence–related entities</b>	<b>27</b>

Cold item placement	27
Default data for out-of-box entities	28
<b>Appendix D: Sample applications that have Recommendations API integration</b>	<b>28</b>
<hr/>	
<b>Appendix E: References</b>	<b>29</b>
<hr/>	

# Use machine learning–based Microsoft Cognitive Services for product recommendations

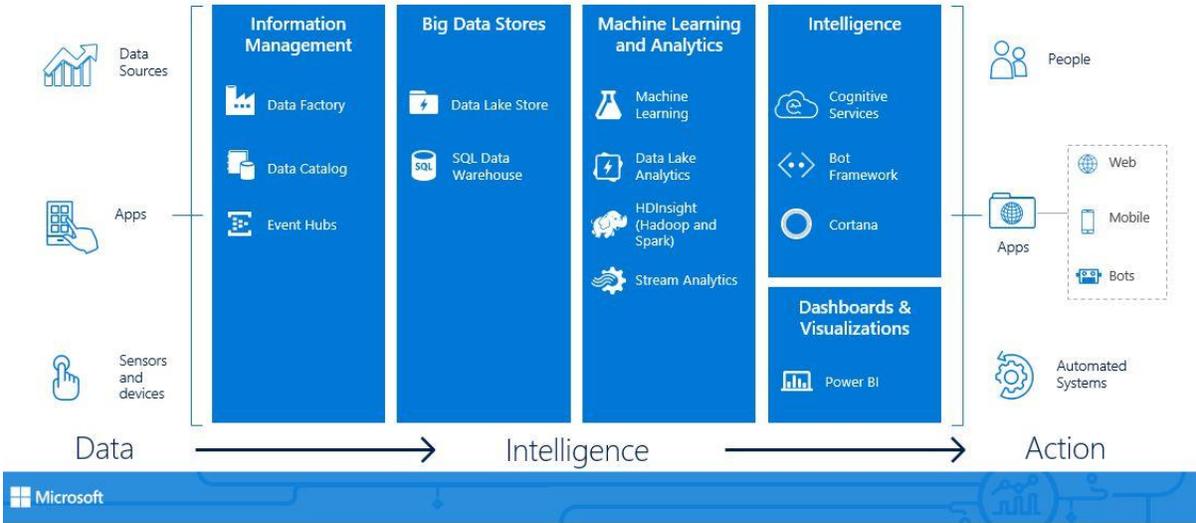
This document will help you make the most of the new **Cortana Intelligence services management** module. This module introduces machine learning capability in Microsoft Dynamics AX 2012 R3 through integration with the Microsoft Cognitive Services Recommendation application programming interface (API) for product recommendations. As with any integration, several technologies, or layers, are involved. The next section provides a brief overview of these technologies and terminology that will be used in this document.

## Overview and terminology

### Cortana Intelligence Suite

Microsoft Cortana Intelligence suite (CIS) is a collection of technologies for information management, big data storage, machine learning and analytics, and data visualizations. Cortana assistant on Windows 10, Chabot that is made by using Bot framework, Microsoft Cognitive Services, and many other applications use one or many technologies that are marketed under the Cortana Intelligence brand.

### Cortana Intelligence Suite (CIS)



# Microsoft Cognitive Services

The Microsoft Cognitive Services APIs are a suite of several general-purpose machine learning APIs that are made available in Microsoft Azure and can be used for any number of applications. These APIs simplify the whole process by abstracting away the complex machine learning models and the operationalization aspects, so that users can focus on real business problems. There are several categories of Azure Machine Learning APIs. The Knowledge category includes the Recommendations API. This document will focus primarily on the Recommendations API, because our feature provides deep integration capability with this API in Microsoft Dynamics AX 2012 R3 Cumulative Update (CU) 12.

## Recommendations API

The Recommendations API exposes a general-purpose recommender system capability that is wrapped in an easy-to-use representational state transfer (REST) API. Our goal is to enable several business processes that involve customer recommendations to use this functionality.

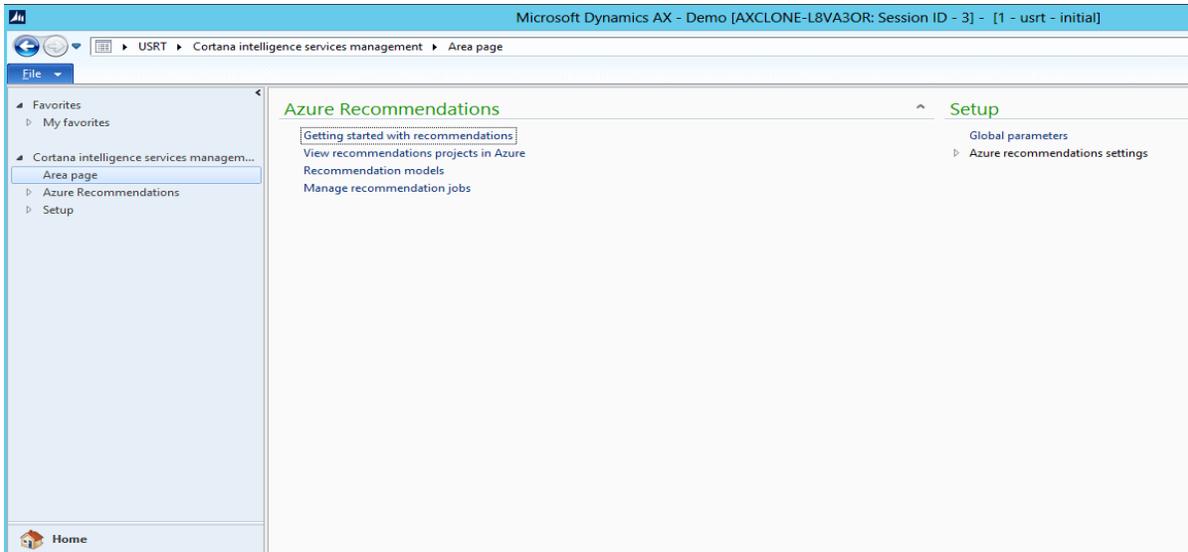
## Installing and configuring prerequisites

- The Cortana Intelligence services management feature is available as part of AX 2012 R3 CU 12. You can also consume the feature by itself by downloading and applying the hotfix for KB 3201875 from Microsoft Dynamics Lifecycle Services (LCS) or the hotfix server. Note that, if you apply the hotfix separately, we recommend that you use one of the following versions of AX 2012 R3:
  - AX 2012 R3 CU 8 or a higher rollup with the Entity Store hotfix (KB 3147499)
  - AX 2012 R3 CU 11
- You must also have an Azure account that can provision a Cognitive Services account.

**Note:** You can also provision a free Cognitive Services account. For more details, see [Appendix A](#).

# Cortana Intelligence services management

The **Cortana Intelligence services management** module is the hub that will host all machine learning–related assets in AX 2012 R3. Currently, the module area page contains the **Azure Recommendations** and **Setup** sections. The **Azure Recommendations** section contains entry points that are related to Recommendations API integration.

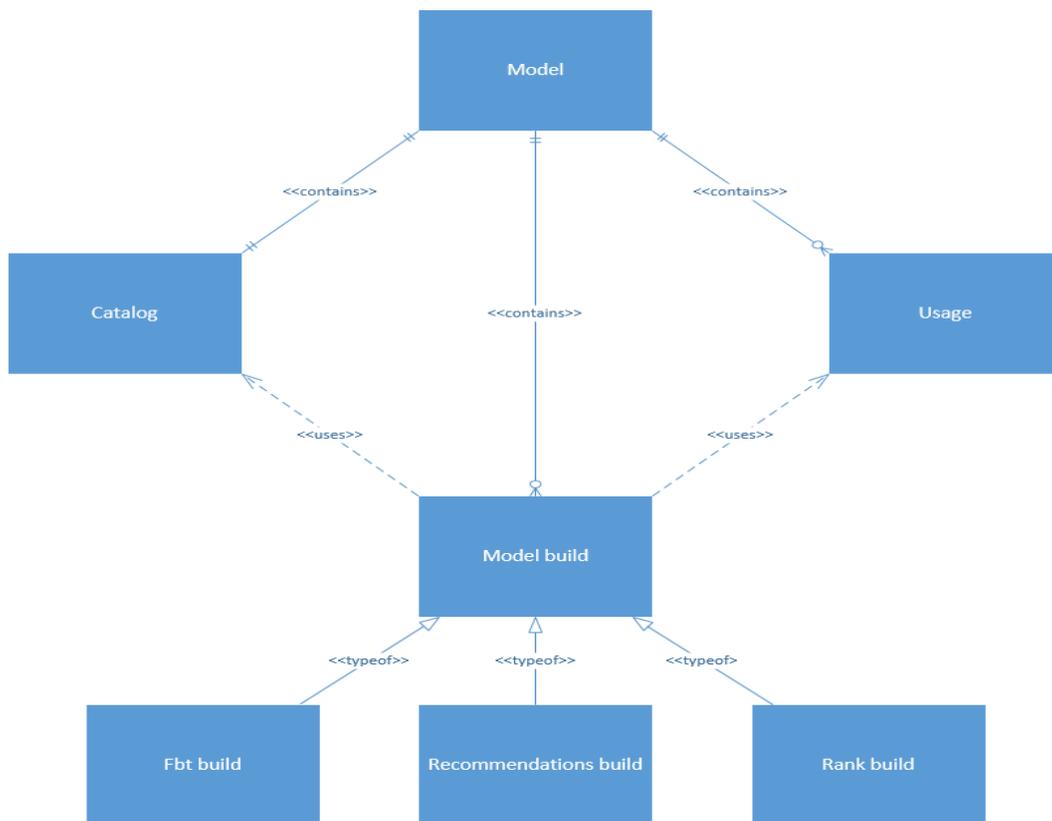


Click **Getting started with recommendations** to open the Recommendations API documentation in a browser window.

Click **View recommendations projects in Azure** to open the **Recommendations UI** in a browser window. You can use this UI to view and edit the Recommendations API projects. To view content that is associated with the Cognitive Services account for the first time, you must provide the Cognitive Services account key (see [Appendix A](#)).

# Recommendations API – Data model

As we mentioned earlier, the Cortana Intelligence services management feature deals primarily with the Recommendations API. At a high level, there are four main entities that you interact with when you use this feature.



## Model

A model or project is the root container or consumption context for the API. The model is also the context of the Azure Machine Learning model that will be provided datasets (specifically, Catalog and Usage datasets) to “train,” and that will then be used to score product recommendations.

## Catalog

The Catalog entity is the dataset that contains the list of all possible items (for example, a product catalog) that the Recommendations API can recommend from. The Catalog entity for the API follows a data contract. Each tuple of the dataset must be in the following format:

<ID>, <Item name>, <Item category>, [<Description>, <Features list>]

The first three parameters refer to the item ID and its descriptors: the name and category. The dataset can contain an optional description and a features list, which is free text in the format *key=value*, where attributes and attribute values that are related to an item can be included, so that the Azure Machine Learning model can generate relevant associations.

Only one Catalog dataset can be associated with a model. The API ingests the catalog in .csv format, but without the header row. This feature includes a Data Import/Export Framework (DIXF)-based entity to export the Catalog dataset from Microsoft Dynamics AX. The DMFRetailProductCategoryEntity entity is based on EcoResProductCategory.

## Usage file

The Usage file or Usage dataset is the set of all transactions for the catalog items (for example, sales transactions or click stream events from an e-commerce portal) and has the following canonical representation:

```
<User ID>, <Item ID>, <Transaction date>, <Event type>
```

You can upload as many Usage files as you want, but there is a restriction on the maximum size for a Usage file (for details, see the Recommendations API). The Recommendations API ingests this dataset in .txt format, but without any header row. The out-of-box entity for exporting this dataset from Microsoft Dynamics AX is DMFRetailProductTransactionEntity, which is based on RetailTransactionSalesTransaction.

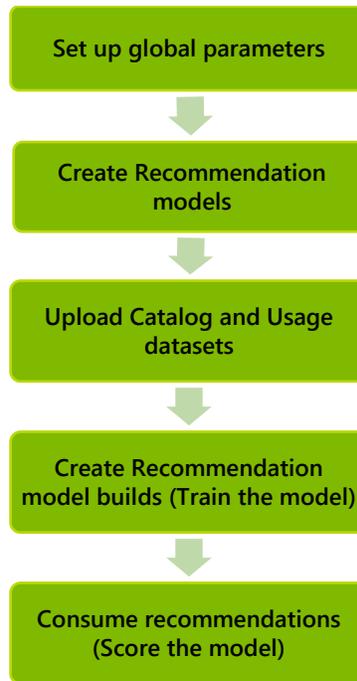
## Model build

The underlying Azure Machine Learning model is first trained with the datasets that are uploaded against it. The training of a model is the process of extracting the data patterns in the Usage dataset, so that predictions can be made on untrained or new data that is presented (this step is called as scoring). In the context of an API, the output or result of training is the build. API consumers can generate recommendations by using any of the specific builds or a default (active) build. The API supports multiple flavors of builds that are useful in different scenarios. The following types are currently supported:

- **Frequently bought together (FBT)** – This type of build helps answer questions about what items are bought together, based on the co-occurrence of items in previous transactions. The scoring context is always a single item.
- **Recommendations build** – This type of build generates recommendations for a basket or group of items for scenarios of the “customers who liked this product also liked these products” type. This build type supports item-to-item (I2I) recommendations or user-to-item (U2I) recommendations. In other words, recommendations can be generated for a specific items or groups of items or for a user based on his transaction history.
- **Rank build** – This type of build helps pick the most important parameters or features to include in recommendations.

# Recommendations API – Integration workflow

The following illustration shows the steps in the process of consuming recommendations by using the capabilities in AX 2012 R3. In the sections that follow, we will explain how you can manually follow this process to generate Recommendations API-based models that can be used in several scenarios to provide meaningful product recommendations.

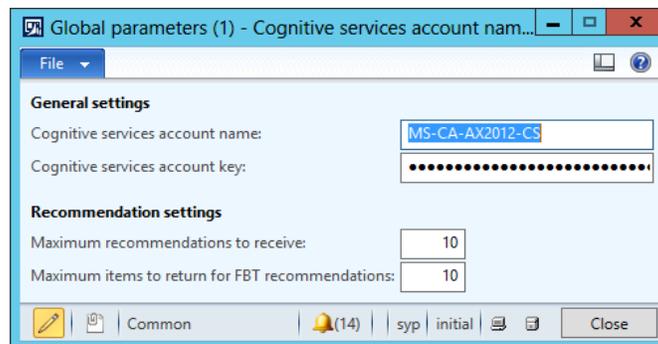


## Step 1. Set up global parameters

The first step if you want to work with Recommendation models is to set up a few default values.

### Global parameters

Start the AX 2012 R3 client, go to the **Cortana Intelligence services management** module, and click **Setup > Global parameters** to open the **Global parameters** form for the feature.



The **General settings** section contains fields where you can enter the Cognitive Services account name and account key. For information about how to obtain this information, see [Appendix A](#).

The **Recommendation settings** section contains options that let you set the maximum number of recommendations to receive for builds of the **FBT** and **Recommendations build** types.

## Default parameters for model builds

Each build type that the API supports has several parameters that you can fine-tune to obtain better-quality predictions. In Microsoft Dynamics AX, you can set default parameters for each type of build by using the **Configure recommendation model build parameters** form at **Cortana Intelligence services management > Setup > Azure recommendations settings > Configure recommendation model build parameters**.

Parameter	Value
Allow placing cold items:	1
Enable feature correlation:	0
Enable modeling insights:	0
Enable user to item recommendations:	1
Features list to enhance recommendations:	
Feature list for use in item recommendation reason:	
Minimum usage points an item must be in for modeling:	2
Maximum usage points an item must be in for modeling:	10000
Number of features to find in the model:	10
Number of model iterations desired:	10
Minimum transactions to consider a user in modeling:	2
Maximum transactions to consider a user in modeling:	10000
Specify if features can be used to enhance recommendations:	0

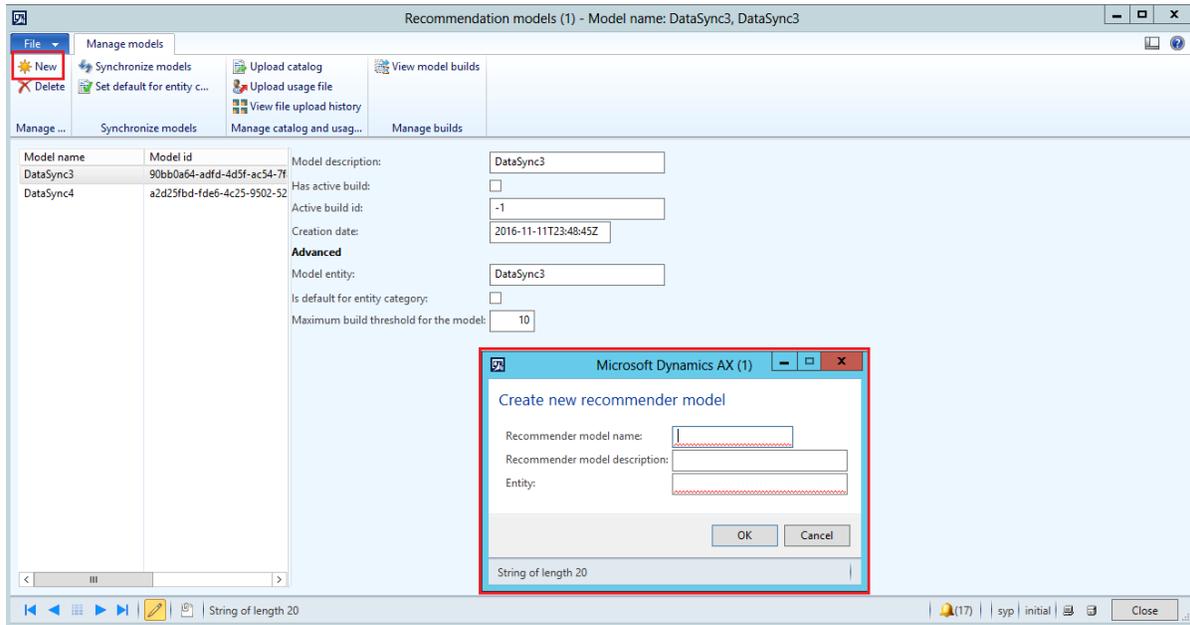
The form lets you modify and update each parameter for every build. Default settings allow for easy creation of model builds when these are automated, as we will see later. Note that users can change the default settings to other values when a new model build is triggered, without affecting the default values themselves. For a detailed description of what each parameter indicates in the context of a build type, see the Recommendations API document ([Appendix E](#) includes a link to this document).

## Step 2. Create Recommendation models

The next step is to create a Recommendations model. A Recommendations model is just a container or a project that uses the general-purpose models behind the scene, and trains them by using the Catalog and Usage files that are provided. To create a model, open the **Recommendation models** form by clicking **Cortana Intelligence services management > Azure Recommendations > Recommendation models**. The Action Pane in the form includes several buttons. The order (from left to right) of the buttons reflects the order that the functions are used in.

## New

Click **New** on the Action Pane to open the **Create new recommender model** dialog box. Enter valid values in the **Recommender model name**, **Recommender model description**, and **Entity** fields. Then click **OK**.



If model creation is successful, a new model appears on the models list page. Any errors that occur are shown in the Infolog.

**Note:** The **Entity** (or **Model entity**) attribute is not required by the Recommendations API. It is a loose association that is maintained in Microsoft Dynamics AX to indicate the scenario or use case that a model is being created for. This behavior gives users the flexibility to switch between multiple models for a scenario or use different models for different scenarios.

## Synchronize models

In many cases, models that are created might have to be used somewhere else. Because a lot of data context is associated with models, it can be hard to re-create them in different environments if you must. Therefore, the Recommendations API and our integration provides model management capability within Microsoft Dynamics AX. To use this functionality, just click **Synchronize models** on the Action Pane. This function synchronizes models and their associated builds, and will pull down all the models that are associated with the specified Cognitive Services account that was entered earlier.

## Notes:

- When you synchronize models, Microsoft Dynamics AX will try to delete any models that are present locally in Microsoft Dynamics AX but not in the Recommendations API repository for that account. This situation can occur if the local Microsoft Dynamics AX instance is stale in comparison to the different Microsoft Dynamics AX instance that was used to create and manage these models. We do not provide model management across instances. If this capability is something that you want, you can ask your implementation partner to develop it for you.
- All local Microsoft Dynamics AX context that is associated with a model, such as the **Model entity, Is default for entity category**, and **Maximum build threshold for the model** properties, are reset when models are synchronized. We recommend that you review these properties after you synchronize the models on any new environment.

## Set default for entity category

This **Set default for entity category** button on the Action Pane sets a specific model as the default model for the entity category. The entity category is a labeled reference to a scenario where you plan to use the model. Therefore, you can create more than one model to use in a specific scenario. However, at any given point, you can automatically use the default model, or you can write logic to switch between multiple models, based on the entity label.

## Step 3. Upload Catalog and Usage datasets

After you create a Recommendation model or project, the next step is to upload Catalog and Usage datasets to the model. For our current integration, you can use either of the following methods:

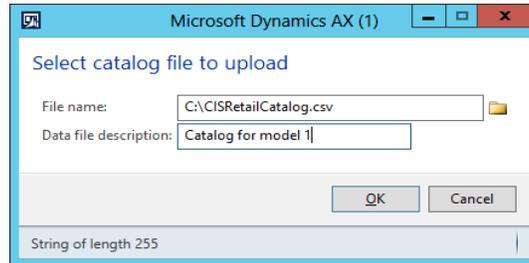
- **Manual upload** – This method allows for explicit upload of Catalog and Usage datasets. To generate the dataset itself in Microsoft Dynamics AX, follow the steps that are outlined in [Appendix B](#). After you have the exported data files, follow the rest of this section to upload the datasets to the model.
- **Automated data synchronization** – This feature hides all the operationalization complexity that is related to data movement, and makes it easier for a Microsoft Dynamics AX consultant or developer to perform data management operations directly in Microsoft Dynamics AX. The **Manage recommendation jobs** form lets you configure automated data synchronization with an existing Recommendation model in both batch and interactive modes. You do not have to explicitly export the data as in the previous steps. For detailed information about this functionality, see the [Automated Recommendations API builds and data synchronization](#) section.

## Upload the Catalog dataset

- 1 In the **Recommendation models** form, on the Action Pane, click **Upload catalog** to open the catalog file upload dialog box.
- 2 Browse to select the catalog data file that you generated earlier by using the out-of-box entity. Enter the required description text for the data file, and then click **OK**.

- 3 Wait for the confirmation that the upload has been completed. You can view upload statistics for the current operation and all previous operations by clicking **View file upload history**. The history form shows the upload status, and also the number of records that were successfully imported and processed for inclusion in the Azure Machine Learning model.

**Note:** You can also view the models and uploaded dataset in the Recommendations UI. To open the UI, click **View recommendations in Azure** or open the Recommendations API portal directly in a browser.

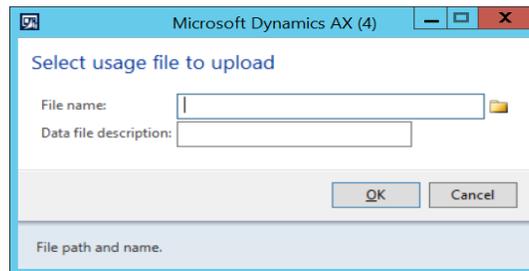


## Upload the Usage dataset

The next step is to upload Usage file data. This data typically consists of the sales transactions or click stream event data (from an e-commerce portal), and is used to train the Recommendation Machine Learning models. Both the quality and the quantity of datasets matter. Good-quality Usage datasets yield better recommendations, and continual use of a model that uses the latest transaction data produces better predictions, because the data patterns will reflect the current trends. You can specify how often reuploading and retraining occur, based on your business requirements.

To upload the Usage dataset manually, follow these steps.

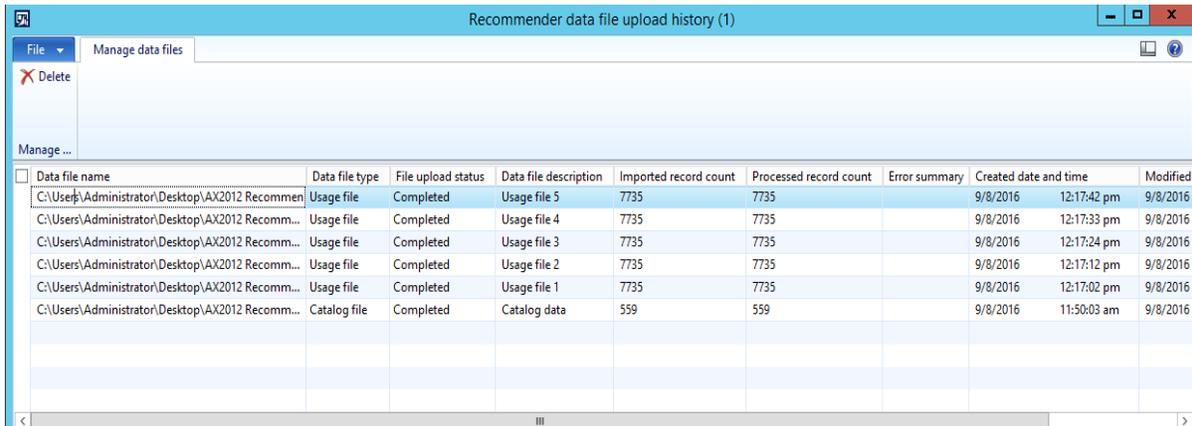
- 1 In the **Recommendation models** form, click **Upload usage file** to open the catalog file upload dialog box.



- 2 Browse to select the Usage file that was manually exported, enter the required description for the data file, and then click **OK**.

## View history

You can view the history of all previous data uploads to the model by using the **Recommender data file upload history** form. Select a model on the models list page, and then click **View file upload history**. For each upload attempt, the form that appears shows the type, the status, import statistics, and the time of the operation. The value in the in the **Processed record count** column includes all records that were successfully processed (and that the model will consume).



Data file name	Data file type	File upload status	Data file description	Imported record count	Processed record count	Error summary	Created date and time	Modified
C:\Users\Administrator\Desktop\AX2012 Recommen...	Usage file	Completed	Usage file 5	7735	7735		9/8/2016 12:17:42 pm	9/8/2016
C:\Users\Administrator\Desktop\AX2012 Recommen...	Usage file	Completed	Usage file 4	7735	7735		9/8/2016 12:17:33 pm	9/8/2016
C:\Users\Administrator\Desktop\AX2012 Recommen...	Usage file	Completed	Usage file 3	7735	7735		9/8/2016 12:17:24 pm	9/8/2016
C:\Users\Administrator\Desktop\AX2012 Recommen...	Usage file	Completed	Usage file 2	7735	7735		9/8/2016 12:17:12 pm	9/8/2016
C:\Users\Administrator\Desktop\AX2012 Recommen...	Usage file	Completed	Usage file 1	7735	7735		9/8/2016 12:17:02 pm	9/8/2016
C:\Users\Administrator\Desktop\AX2012 Recommen...	Catalog file	Completed	Catalog data	559	559		9/8/2016 11:50:03 am	9/8/2016

## Step 4. Create Recommendation model builds (Train the model)

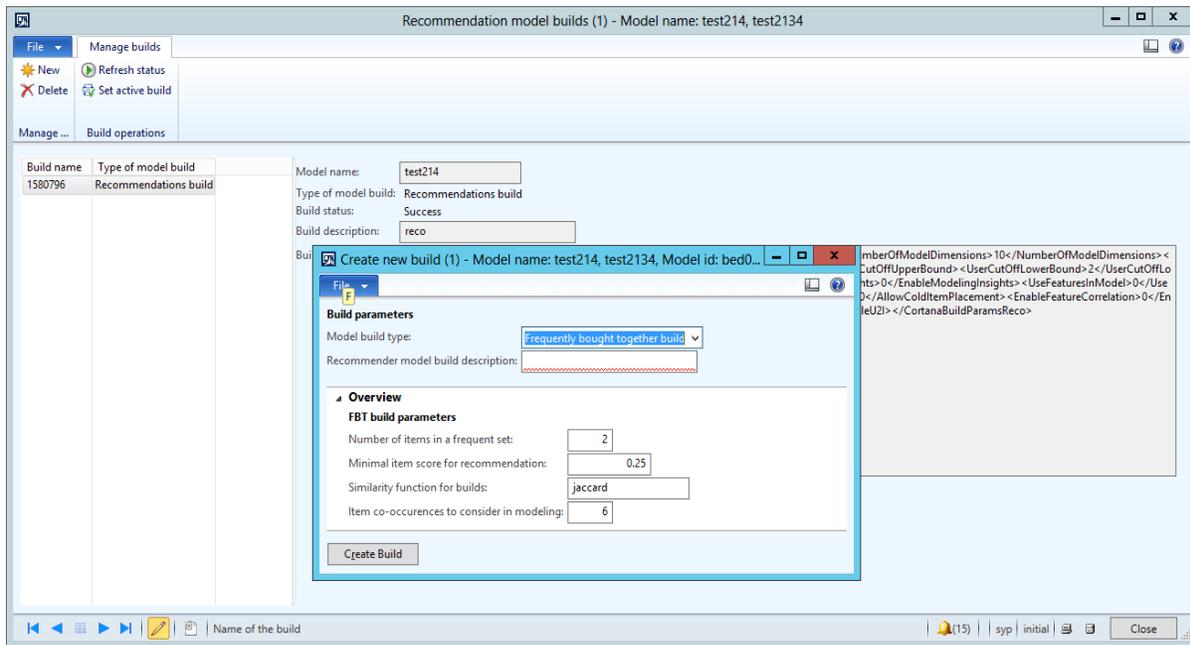
The next step is to train the Azure Machine Learning models. To train the models, you trigger builds on the Azure Machine Learning model. These builds can then be used to generate recommendations. As we described earlier, the Recommendations API supports three types of builds: **FBT (Frequently bought together)**, **Recommendations build**, and **Rank build**. For more information about the meaning of each build type and its parameters, see [Build types and model quality](#).

Follow these steps to trigger a new build.

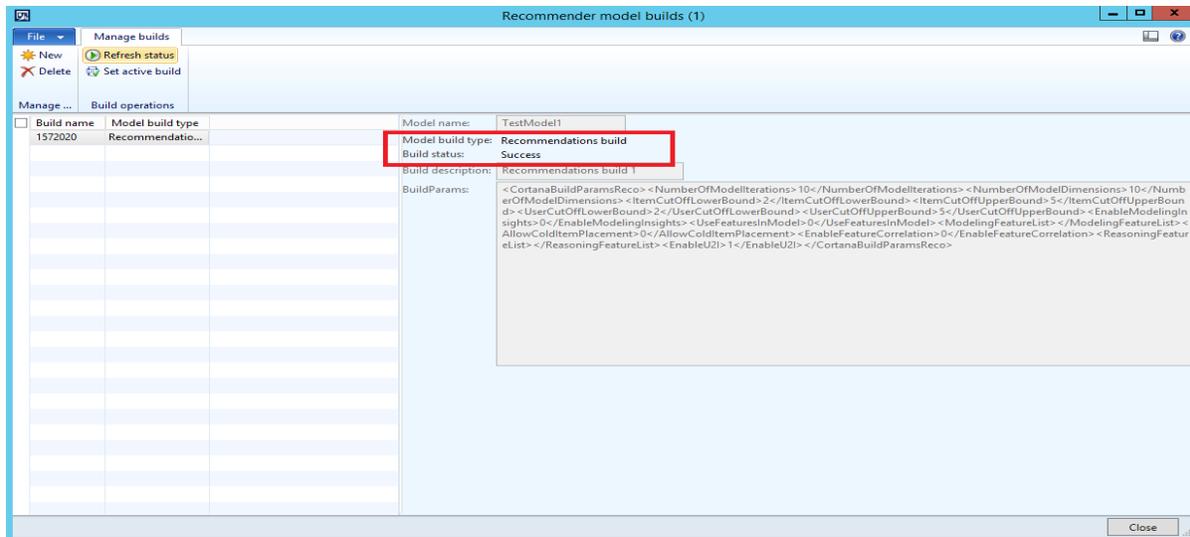
### Create a new build

- 1 In the **Recommendations model** form (**Cortana Intelligence services management > Azure Recommendations > Recommendation models**), select the model to create a build for.
- 2 On the Action Pane, click **View model builds**.

- Click **New** to open the model build dialog box. In the **Model build type** field, select a build option, enter description text for the build, and then click **Create Build**. The **Overview** FastFab shows default build parameters for the build type. You can change the value of any parameter for the build without affecting the global default settings for that build type.

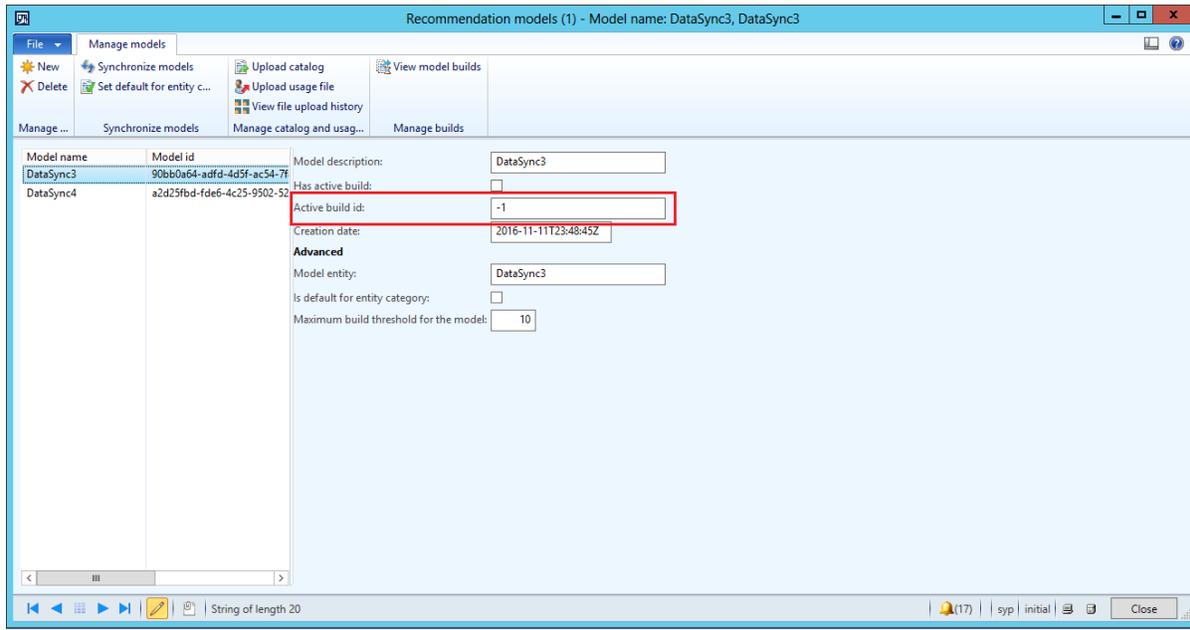


- Press F5 to update the form, and view the new build. You can click **Refresh status** to update the status of a build. Note that only one build of each type can be in progress at any time.



## Set active build

- 1 Select the newly created build, and click **Set active build** to indicate that this build will be used for scoring. Note that a Recommendations API model can have many builds of different types. You must explicitly indicate the default build that should be used for scoring when a given model is used.
- 2 You must confirm the active build on a model by using the synchronize functionality to make sure that the **Active build id** value is set correctly.



## Step 5. Consume recommendations (Score the model)

The Recommendations API integration lets you implement scenarios in AX 2012 R3 that can take advantage of the power of machine learning to easily provide product recommendations. We do not provide consumption scenarios as part of the Cortana Intelligence services management capability.

Here are some of the most common scenarios that can use this integration:

- **Customer service integration** – The Recommendations API integration can be used to provide product recommendations for customers who interact with a call center agent to place catalog orders. At the point of interaction, the call center agent can both upsell and cross-sell items that might be of interest to the user. Note that a mix of recommended items can be included, based on builds of both the **FBT** and **Recommendations build** types, to provide meaningful upsell capabilities.
- **Recommendations in a Microsoft Dynamics AX for Retail online store** –In AX 2012 R3, the Retail online store or the Retail online sample store can be extended to use product recommendations. The online sample store is preconfigured to include recommendations-related controls in the **Products/Item** view. By default, these recommendations currently use the related items for a product and are based on the static configuration when a new product is added to the retail catalog. To use the Recommendations API, you can use custom logic to extend the Generate related products (RetailRelatedProductsJob) batch job that currently computes the various categories of recommendations. Even if you do not use the Retail online store, you can take advantage of a

customization of this type, provided that the Retail channel database is being consumed by your e-commerce front end.

- **Recommendations in an online store that is based on another e-commerce platform** – The **Cortana Intelligence services management** module is not dependent on Microsoft Dynamics AX for Retail. You can use this module even if you have an online e-commerce site (custom or from another e-commerce vendor). Therefore, you can keep Microsoft Dynamics AX as the system that all other integrations point to. In this way, you can help guarantee that business logic always go through your enterprise resource planning (ERP) application.
- **Recommendations in emails** – The Recommendations API integration can also help you with the challenge of recommend products to users. These recommendations, which are often sent by email, aggregate the history of the user’s purchase and also use the items that are purchased by similar customers. Because you can create and use rich HTML-based email templates in AX 2012 R3, and can also configure those templates so that they are used in a batch-based server setup, you can build a rich scenario around this use case. You can even segment this issue based on customers (top x percent or bottom x percent) and items (place cold items).
- **Recommendations in clienteling scenarios** – You can surface recommendations in Retail Modern POS devices if you are using those devices together with Microsoft Dynamics AX for Retail, or if you have another clienteling system connected to Microsoft Dynamics AX.
- **Recommendations on printed invoices** – You can print product recommendations on store receipts if you use Microsoft Dynamics AX for Retail, or if you have another retail system connected to Microsoft Dynamics AX.
- **Recommendations in sales order taking scenarios** – You can surface recommendations in sales orders if you have a scenario that calls for this behavior.
- **Recommendations in mobile apps** – If you have customer-facing mobile apps that your customers use to place orders for your products, and if these apps are connected to Microsoft Dynamics AX, you can use this module to help guarantee that business logic flows through Microsoft Dynamics AX.

If you are interested in learning how to build these consumption scenarios, there is a sample implementation of the call center scenario. For more details, see [Appendix D](#).

## Automated Recommendations API builds and data synchronization

The Recommendations API integration provides capabilities for automating the model build creation and data synchronization processes. In a typical enterprise environment, we do not expect that these processes will be entirely manual. Therefore, some advanced options are available to help users manage the workflow steps that we described in the previous section.

Operation	Automated	Type	Recurrence is supported	Comments
Create Recommendation models.	No	Interactive	No	Each Cognitive Services account can have a maximum of 10 models. Models are not created very often.

Operation	Automated	Type	Recurrence is supported	Comments
Upload Catalog and Usage datasets.	Yes	Both interactive and non-interactive (batch)	Yes	The native AX 2012 R3 change tracking feature is used to control full and delta dataset exports that can be synced with the specified Recommendations model.
Create Recommendation model builds.	Yes	Both interactive and non-interactive (batch)	Yes	A maximum of 20 builds per model is supported. Circular truncation logic supports recurring builds, based on the <b>Maximum build threshold</b> property that is set up on the model.

All the automated build and data synchronization processes use the **Manage recommendation jobs** form at **Cortana Intelligence services management > Azure Recommendations > Manage recommendation jobs**.

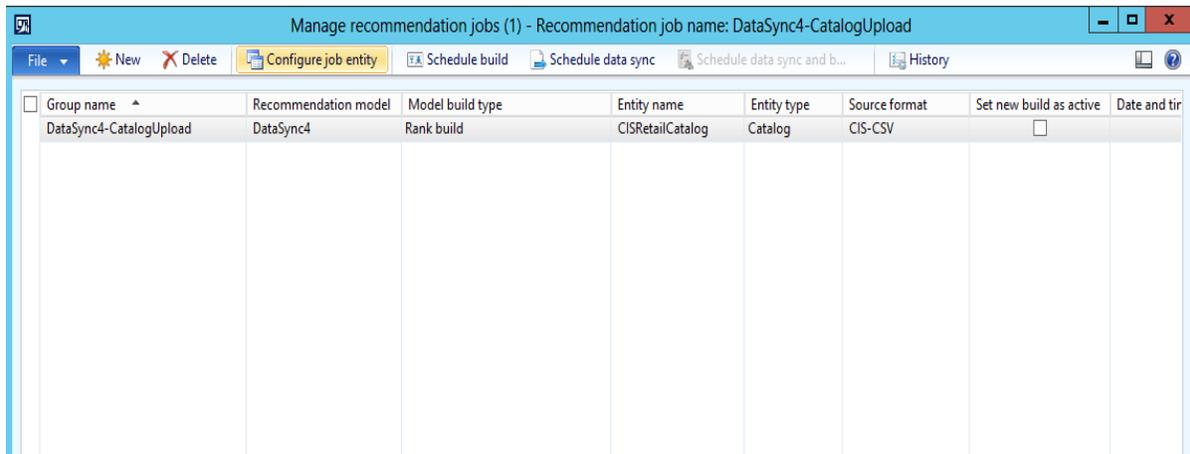
## Automated ingestion of Catalog and Usage datasets

The Recommendations API model validates the datasets when they are uploaded, to make sure that the product references in the Usage file can be correlated to the Catalog dataset. Therefore, you must run Catalog dataset ingestion job first, before you schedule Usage file jobs. The same procedure is used to set up data synchronization jobs for both Catalog and Usage entities, unless differences are noted in specific steps.

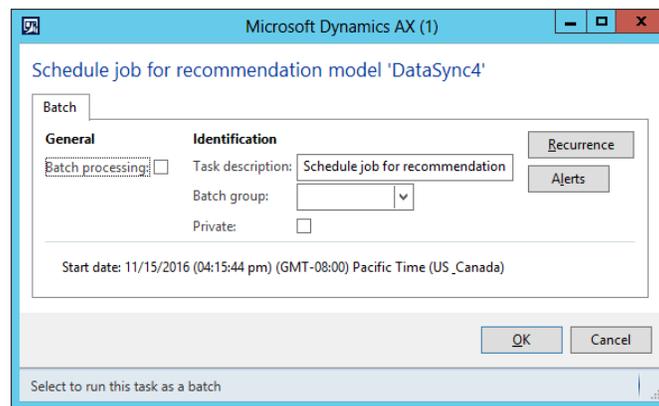
- 1 Switch the Microsoft Dynamics AX client to the company to export the data from.
- 2 In the **Manage recommendation jobs** form, click **New**.
- 3 In the **Create recommendation job** dialog box, enter a valid name, and then click **OK**.
- 4 In the **Recommendation model** column, select the model to upload the dataset to the name of the DIXF entity that you previously created (see [Step 2. Set up target entities](#) in [Appendix B](#)).
- 5 In the **Source format** column, set the value to the delimited file-based source (see [Step 3. Set up the data source format for export](#) in [Appendix B](#)).
- 6 In the **Entity type** column, select the type of entity (**Catalog** or **Usage**).
- 7 Press Ctrl+S to save your changes.
- 8 On the Action Pane, click **Configure job entity**, and wait for the **Processing group** form to appear.
- 9 On the Action Pane, click **Entities**, and verify that the entity that you selected in the **Manage recommendation jobs** form is the only entity that is present.
- 10 Configure a source map that has the desired attribute sequence for the catalog entity (see [Step 4. Generate the source mapping for the entity](#) in [Appendix B](#)).

**Note:** If you have an existing sample for the catalog entity, you can skip most of the steps in that procedure. Just use your existing sample as the sample file, generate the source mapping alone, and validate the mapping.

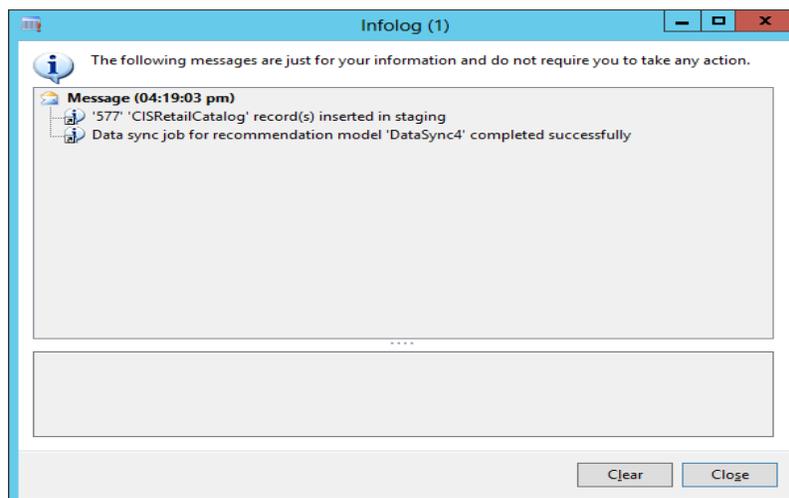
- 11 Close the **Entities** and **Processing group** forms. In the **Manage recommendation jobs** form, the **Schedule build** and **Schedule data sync** buttons on the Action Pane should now be available.



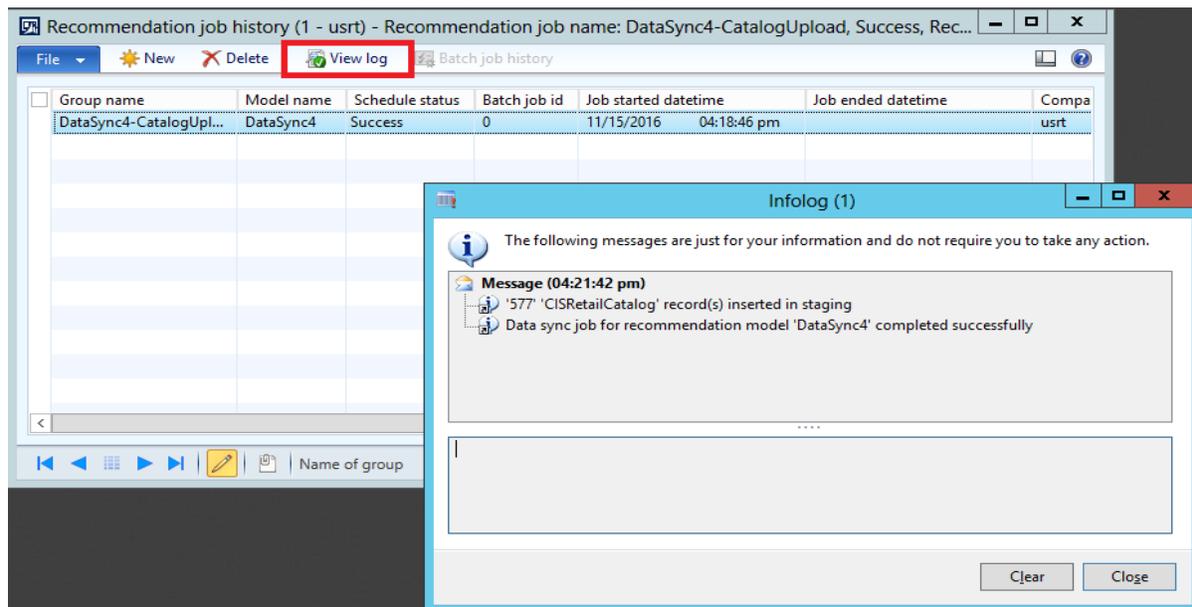
- 12 Click **Schedule data sync** to open the batch job dialog box for the recommendation model job schedule.



- 13 Click **OK** to interactively upload the data for the specific type to the model selected. Alternately, you can check the *Batch processing* flag to run this on the batch server in non-interactive mode. If running interactively, an Infolog with the number of exported records is shown when the operation completes.



- 14 You can view the log of this operation as well any previous ones by selecting the job in the **Manage recommendations job** form and then clicking the **History** action pane menu. Click **View log** to see logs for this operation.



#### Notes:

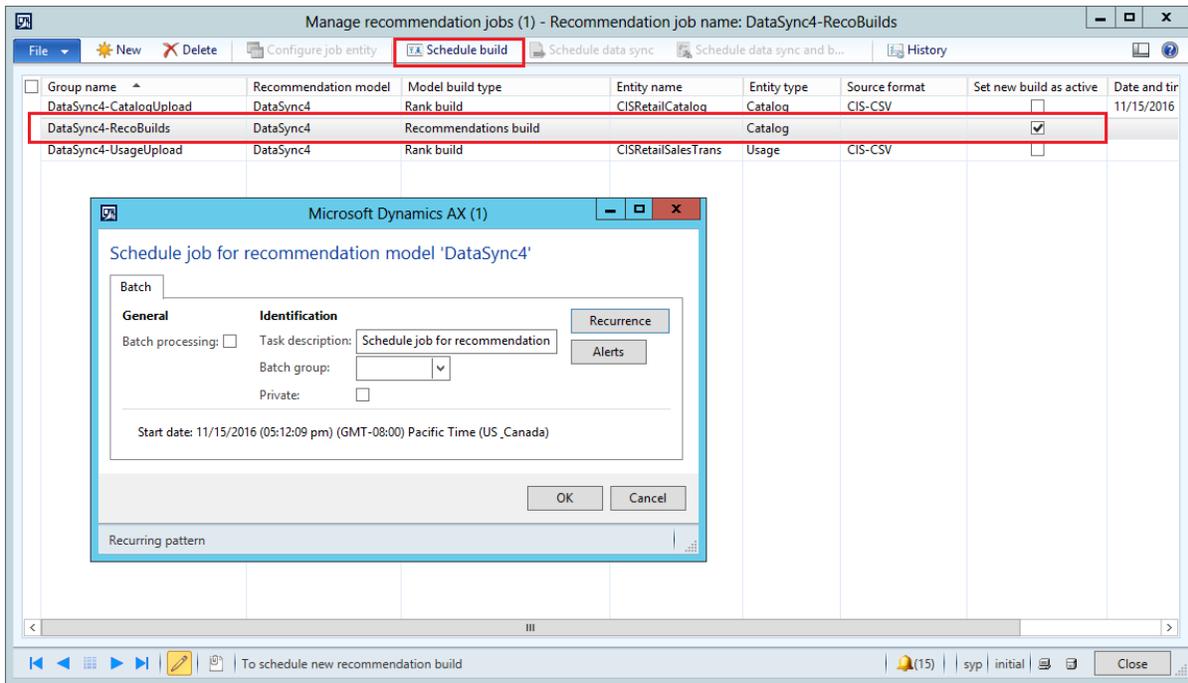
- When recurrence is set and batch processing is enabled, the job scheduler pushes only delta records to the Recommendation model. In other words, by setting the recurrence to every 24 hours at a specific time, you enable a daily data push of transactions to the specified model.
- For Usage data jobs, you can trigger data synchronization and then a build immediately afterward. Click **Schedule data sync and build** on the Action Pane.

## Build automation

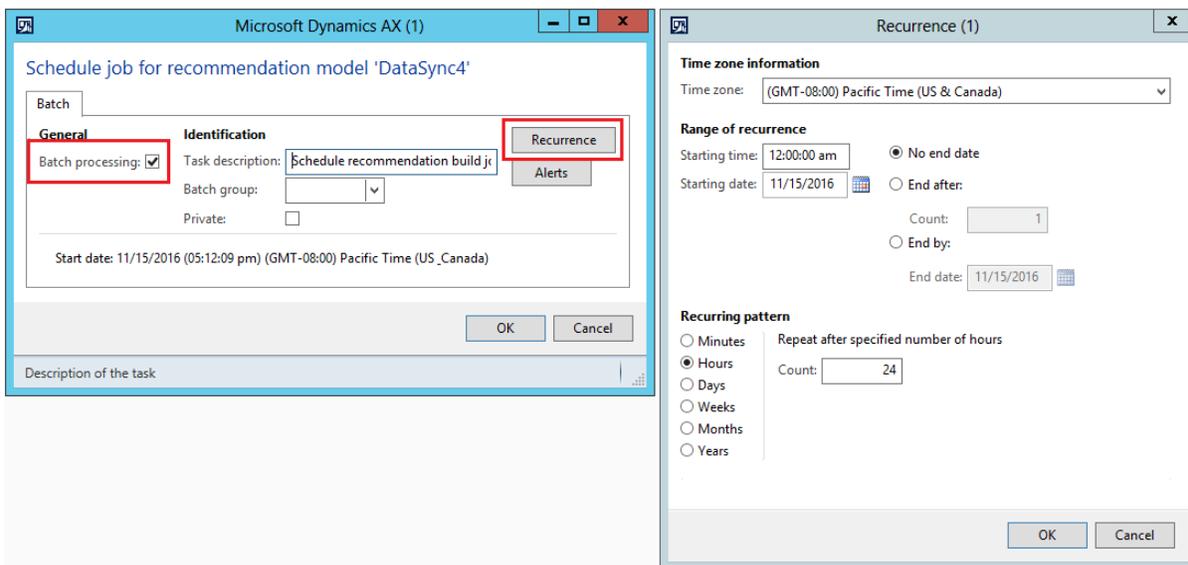
The build step is the training step for the Recommendations API model. You can schedule the build job independently of the data synchronization itself, and you can set the job to any frequency that you want. However, newer builds are likely to produce more accurate recommendations. Also, note that the API currently enforces a limit of 20 builds per model. Therefore, to enable automation of the builds, we set up a rolling build by using the **Maximum build threshold for the model** property on a model. By default, this property is set to **10** for a new model, but you can set it to any value between **1** and **20**. When you set this property, older builds are deleted as new builds are generated.

- 1 Switch the Microsoft Dynamics AX client to the company to set up recurring builds for.
- 2 In the **Manage recommendation jobs** form, click **New**.
- 3 In the **Create recommendation job** dialog box, enter a valid name, and then click **OK**.
- 4 In the **Recommendation model** column, select the model to create builds for.

- In the **Model build type** column, set the value to the type of build to generate. We also recommend that you select the **Set new build as active** check box for this job.



- Click **OK** to trigger a new build of the **Recommendations build** type for the specified model. Alternatively, to set up recurrence that uses batch processing, select the **Batch processing** check box, and then click **Recurrence**. After you set up the recurrence, a message in the Infolog confirms that the job has been added to the batch queue.



- To see the build status, view the model builds in the **Recommendation model builds** form. You can also view the status in the **History** form and then click **View log** in that form to view the logs.

# Frequently asked questions

- Why do I encounter the following “Unauthorized” exception when I create a model or model build, when I upload Catalog or Usage datasets, or when I perform any other operation?

System.AggregateException: One or more errors occurred. ---> Microsoft.Rest.HttpOperationException: Operation returned an invalid status code 'Unauthorized' at Microsoft.CortanaIntelligence.RecommendationsServiceClient.CognitiveServices.Recommendations.<GetallmodelsWithHttpMessagesAsync>d\_\_51.MoveNext()

An exception that resembles the previous exception probably indicates that the Cognitive Services account key is set incorrectly. To resolve this issue, sign in to your subscription in the Microsoft Azure portal, and copy the correct Cognitive Services account key to the similarly named field in the **Global parameters** form in the **Cortana Intelligence services management** module.

- Why do I receive the following exception?

System.AggregateException: One or more errors occurred. ---> System.Net.Http.HttpRequestException: An error occurred while sending the request. ---> System.Net.WebException: Unable to connect to the remote server ---> System.Net.Sockets.SocketException: No connection could be made because the target machine actively refused it

An exception that resembles the preceding exception are probably related to network connectivity. To resolve this issue, verify whether the Microsoft Dynamics AX server and/or the client computer has Internet access. In some cases, firewall restrictions or missing Domain Name Service (DNS) configurations can also cause this issue.

- What should I do if the creation of a Recommendations API model fails?

There are several reasons why model creation typically fails. For example, if the number of allowed models per Cognitive Services account is exceeded (the current limit is 10), you will receive a message that resembles the following message:

System.AggregateException: One or more errors occurred. ---> Microsoft.CortanaIntelligence.RecommendationsServiceClient.RecommendationsClientException: Request failed with code 'ApiException' message '(EXT-0109) An API exception occurred. User '1a2b00f5-2f80-43ff-81e4-7016f6dc9ef9@mp.com' reached his model limitation '10'

To resolve this issue, delete any unused models by using the Delete functionality in the **Recommendation models** form in Microsoft Dynamics AX.

- What should I do if the upload of Catalog/Usage data fails?

In some cases, if you do not provide a valid description for the Catalog or Usage file when you upload the dataset, you might encounter an error and receive the following message:

System.AggregateException: One or more errors occurred. ---> Microsoft.Rest.HttpOperationException: Operation returned an invalid status code 'InternalServerError'

To resolve this issue, provide a valid description for the file that is being uploaded.

- What should I do if the build of a Recommendations API model fails?
  - If the number of allowed builds per model is exceeded (the current limit is 20), you will receive a message that resembles the following message:

System.AggregateException: One or more errors occurred. --->

Microsoft.CortanaIntelligence.RecommendationsServiceClient.RecommendationsClientException: Request failed with code 'ApiException' message '(EXT-0109) An API exception occurred. Model with id: 'a3e6585d-c71e-4d78-bb39-798537129654' for user: '1a2b00f5-2f80-43ff-81e4-7016f6dc9ef9@mp.com' has '20' builds while maximum allowed is '20'. Consider deleting older builds.'

To resolve this issue, delete any unused builds in a model by using the Delete functionality in the **Recommendation model builds** form in Microsoft Dynamics AX.

- The Recommendations API does not allow more than one build of any build type to be triggered in parallel. If you try to trigger multiple builds of the same build type, you will encounter a failure and receive the following message:

System.AggregateException: One or more errors occurred. --->

Microsoft.CortanaIntelligence.RecommendationsServiceClient.RecommendationsClientException: Request failed with code 'BadArgument' message '(EXT-0108) Passed argument is invalid. Cannot start new build of type 'Fbt' when another build '1585939' of same type is running for model 'a3e6585d-c71e-4d78-bb39-798537129654'

To resolve this issue, wait until the previous build is completed (error or success status). You can use the **Refresh build** command in the **Recommendation model builds** form to update and find the latest status of a build.

## Appendix A: Provisioning an Azure Cognitive Services account

The AX 2012 R3 Recommendations API integration requires that an Azure Cognitive Services account be provisioned. To provision a new Cognitive Services account for the Recommendations API, follow the instructions at <https://www.microsoft.com/cognitive-services/en-us/sign-up>.

Alternatively, if you have an Azure account that has a valid subscription, you can provision an account in the subscription by following these steps.

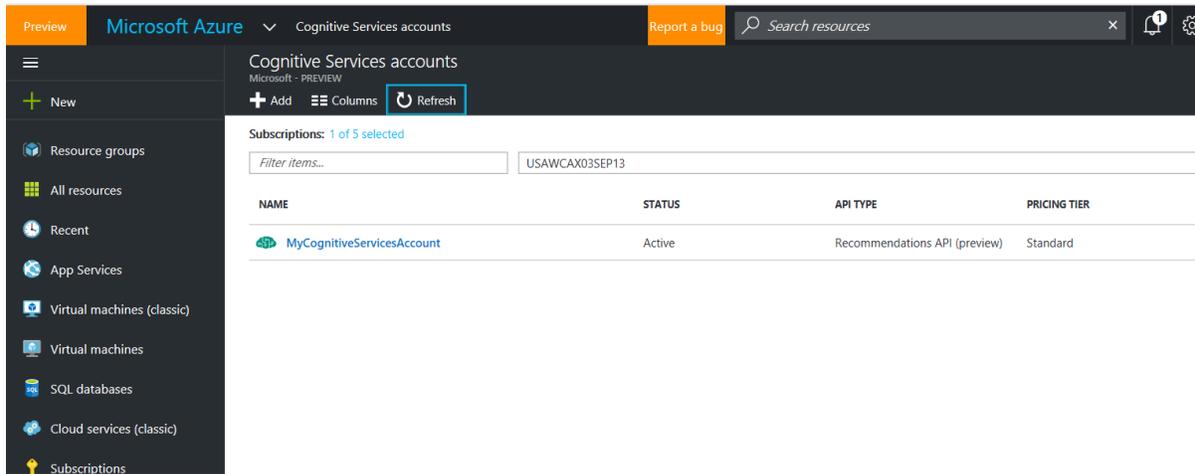
- 1 Sign in to your subscription in the Azure portal (<https://ms.portal.azure.com>).
- 2 In the list of all services, search for Cognitive Services accounts.
- 3 Click **Add**.
- 4 On the create page, specify the following values for each attribute:
  - **Account name:** Enter Any valid value.
  - **Subscription:** Select a subscription.
  - **API type:** Select **Recommendations API (preview)**.

- **Location:** The default value is **West US**.
- **Pricing tier:** Select any relevant pricing tier for your use case.

**Note:** The pricing for every Cognitive Services API is independent. For the pricing details for the Recommendations API, see <https://azure.microsoft.com/en-us/pricing/details/cognitive-services/recommendations>.

- **Resource group:** Select an existing resource group, or add a new resource group. We recommended that you create a new group for this application.

5 Click **Create** to provision the account.



6 After an account is provisioned, click the account, and then open the **Keys** form, and make a note of the **Account Name** property value, and the value of either the **KEY 1** property or the **KEY 2** property.

**Note:** The **Account Name** property will be mapped to the **Cognitive services account name** field at **Cortana Intelligence Services management > Settings > Global parameters**. The **KEY 1** or **KEY 2** property must be mapped to the **Cognitive services account key** field in the Recommendations API feature's **Global parameters** form in AX 2012 R3.

## Appendix B: Using out-of-box entities to generate Catalog and Usage datasets

As we mentioned earlier, the Recommendations API has a flexible but fixed data contract for the Catalog and Usage datasets. To help generate these datasets, we include some DIXF-based export entities. To generate data by using these entities, follow these steps.

## Step 1. Configure DIXF parameters

Follow the instructions at <https://technet.microsoft.com/en-us/library/dn720760.aspx> to set up and configure DIXF in your AX 2012 R3 environment.

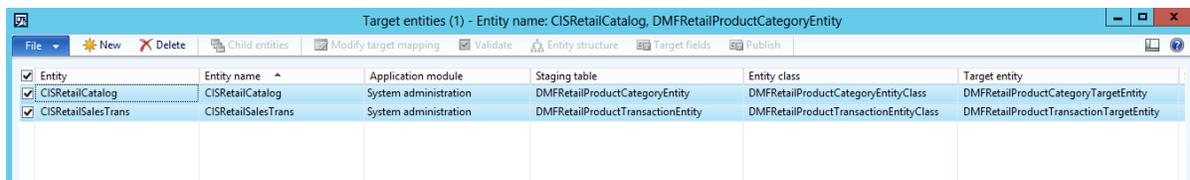
Make sure that the DIXF parameters are validated at the end of this step.

## Step 2. Set up target entities

The following DIXF entities have been included out of the box as part this feature.

Entity type	Staging table	Entity class	Target entity
Catalog	DMFRetailProductCategoryEntity	DMFRetailProductCategoryEntityClass	DMFRetailProductCategoryTargetEntity
Usage	DMFRetailProductTransactionEntity	DMFRetailProductTransactionEntityClass	DMFRetailProductTransactionTargetEntity

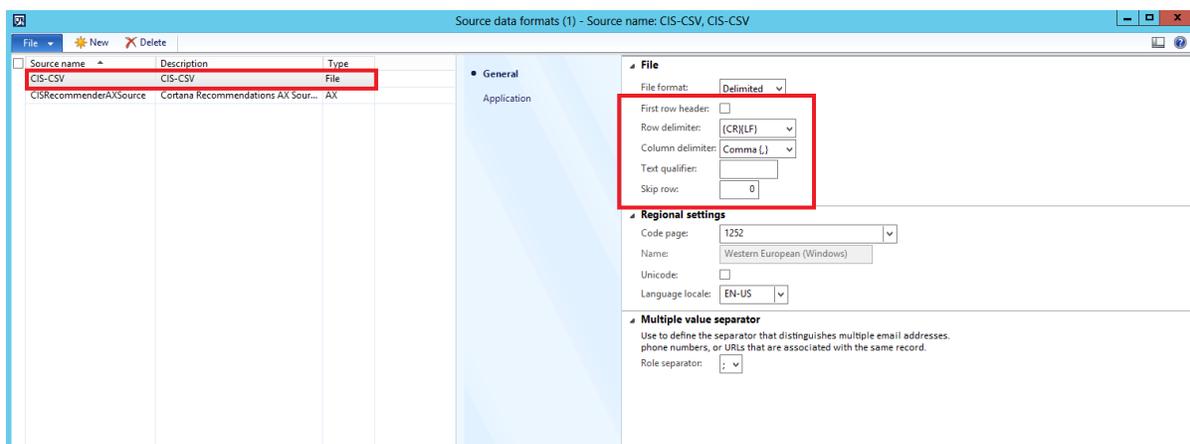
Use the Target entities form (**Data import export framework > Setup > Target entities**) to add these entities to your AX 2012 R3 environment.



**Note:** You can set the **Entity**, **Entity name**, and **Application module** values as you want.

## Step 3. Set up the data source format for export

Set up a DIXF data source for exporting the data files from Microsoft Dynamics AX. Note that the first row header is not required, and the row columns are comma-delimited.



## Step 4. Generate the source mapping for the entity

You must generate the source mapping for these entities and rearrange the entity attributes so that the columns are in the order that the Recommendations API expects. Follow these steps.

- 1 Create a new processing group. Click **Data import export framework > Common > Processing group**, click **New**, and enter a name. Save the new group.
- 2 Click **Entities**.
- 3 In the **Select entities for processing group** form, click **New**, and add each entity that you set up in [Step 2. Set up target entities](#). For each entity, complete all the following steps. Select the file-based source format that you created in [Step 3. Set up the data source format for export](#).
- 4 On the Action Pane, click **Generate source file**.
- 5 In the **Source file generation** wizard, click **Next**, and then select **Mandatory** and **Present in source** for all fields. Click **Finish** to exit the wizard.
- 6 On the Action Pane, click **Entity attributes** to open the **Source attributes** form. Make sure that the entity attributes are in the following order for each entity.

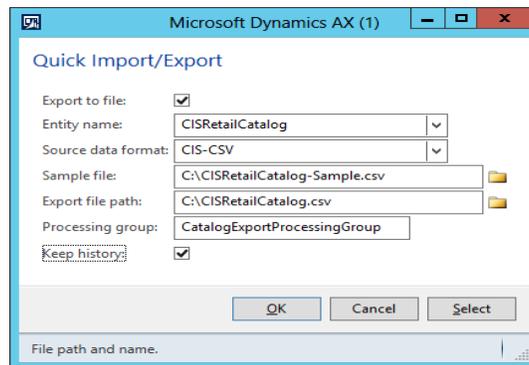
Entity	Order of attributes
Catalog	Product, ProductName, Category, CategoryName, FeaturesList
Usage	CustAccount, Product, TransDate, EventType

- 7 Click **Export file** to export the sample in the preceding order.
- 8 Save the file locally so that it has a .csv file name extension, and then close the form.
- 9 In the **Entities** form, click the file explore button for **Sample file path**, and select the sample that you generated in the previous step.
- 10 Click **Generate source mapping**.
- 11 After you receive a mapping confirmation message, click **Validate**, and wait for confirmation that the operation was successful.
- 12 On the Action Pane, click **Modify source mapping** to open the **Mapper** form. Confirm that all fields have been mapped correctly.
- 13 Close all forms.

## Step 5. Export data for Recommendation entities

You can use the **Quick Import/Export** form to export the data. As we noted earlier, you must complete this only if you are manually importing datasets against a model.

- 1 Open the **Quick Import/Export** form by clicking **Data import export framework > Common > Quick import export**.
- 2 Configure all the parameters in the form by using the previously created values for **Entity name**, **Source data format**, **Sample file**, and setting new values for **Export file path** and **Processing group**.



- 3 Make sure that the **Export to file** check box is selected.
- 4 Click **OK**, and wait for the operation to be completed.
- 5 Browse to and inspect the exported data file. Verify that data was exported in the format that the Recommendations API requires.

**Note:** The complete reference for collecting data that can be used with the Recommendations API is available at <https://docs.microsoft.com/en-us/azure/cognitive-services/cognitive-services-recommendations-collecting-data>.

## Appendix C: Customizing out-of-box Cortana Intelligence-related entities

The out-of-box entities that we reference in this document conform to the data contract that the Recommendations API imposes for these entities. However, you can customize these entities for several reasons.

### Cold item placement

Items that do not have significant usage (that is, items that not many transactions exist for) are called *cold items*. For the Recommendations build of the API, you can use features to include cold items in recommendations. You might notice the following column at the end of each row in the catalog data files:

```
"Category=22565422690,ParentCategory=22565422689"
```

In this case, `Category` and `ParentCategory` are the features that we provide that enable the models to infer the relation between items. The **FeaturesList** column in the out-of-box `Catalog` entity holds free-form text (in the `key=value` format). You can use this column to provide a rich set of features about an item, such as the product attributes and price. The model uses this information to create useful correlations that are relevant to pushing cold items in product recommendations. Note that this technique can be very effective if you want to push product recommendations that are based on specific attributes, such as the price or product attributes. For models to use this information, the Recommendations build must also enable the following build parameters.

Build parameter	Description
<code>useFeaturesInModel</code>	<b>True.</b> This parameter indicates whether features can be used to enhance the recommendation model.
<code>allowColdItemPlacement</code>	<b>True</b> if cold item placement is desired.
<code>modelingFeatureList</code>	A comma-separated list of feature names that should be used in the Recommendations build to enhance the recommendation.  <b>Note:</b> To create a Recommendations build that uses the default features list that we supply with our <code>Catalog</code> entity, you must use " <b>Category,ParentCategory</b> " as the value for this parameter.

## Default data for out-of-box entities

The out-of-box entities for `Catalog` and `Usage` produce datasets that primarily contain the `Reclds` for **Item**, **Customer**, and **Item category**. **No Personally identifiable information (PII) data is exported by using the out-of-box entities.** However, you can customize these fields to send the item name or customer name. Although these customizations are supported, they are not recommended.

## Appendix D: Sample applications that have Recommendations API integration

In an earlier section, we noted a few scenarios that can be implemented by using this feature. To help promote use of this feature, we have a stand-alone GitHub repository, <https://go.microsoft.com/fwlink/?linkid=836353>, that includes a sample implementation that takes advantage of the Recommendations API integration feature. The GitHub repository contains following material.

Item	Description	Comment
License	Massachusetts Institute of Technology (MIT) License	Code that is provided as-is under MIT License
Readme.docx	Readme file	

Item	Description	Comment
App code for Recommendations	Menu items, classes, and tables that you can use in your scenario to display product recommendations	X++ application code
Reference implementation – Customizing Call center application for Recommendations	Reference implementation of the call center scenario that we described in the <a href="#">Step 5. Consume recommendations (Score the model)</a> section	This reference implementation provides a document that explains how you can use the sample Microsoft Dynamics AX code this is provided in the GitHub repository together with code snippets to customize your <b>MCRCustomerService</b> and <b>SalesTable</b> forms.

## Appendix E: References

- **Microsoft Azure** – <https://azure.microsoft.com>
- **Cortana Intelligence Suite** – <https://www.microsoft.com/en-us/cloud-platform/cortana-intelligence-suite>
- **Microsoft Cognitive Services** – <https://www.microsoft.com/cognitive-services>
- **Recommender systems** – [https://en.wikipedia.org/wiki/Recommender\\_system](https://en.wikipedia.org/wiki/Recommender_system)
- **Recommendations API:**
  - **Main page** – <https://www.microsoft.com/cognitive-services/en-us/recommendations-api>
  - **API reference** – <https://westus.dev.cognitive.microsoft.com/docs/services/Recommendations.V4.0/operations/56f30d77eda5650db055a3db>
  - **UI** – <https://recommendations-portal.azurewebsites.net/#/projects>
  - **Collecting data** – <https://docs.microsoft.com/en-us/azure/cognitive-services/cognitive-services-recommendations-collecting-data>
  - **Build types and their usage** – <https://docs.microsoft.com/en-us/azure/cognitive-services/cognitive-services-recommendations-buildtypes>
- **Quick start guide** – <https://docs.microsoft.com/en-us/azure/machine-learning/machine-learning-recommendation-api-quick-start-guide>
- **Dynamics-Cortana Team blog** <https://blogs.msdn.microsoft.com/intel/>

[Send feedback.](#)

Microsoft Dynamics is a line of integrated, adaptable business management solutions that enables you and your people to make business decisions with greater confidence. Microsoft Dynamics works like and with familiar Microsoft software, automating and streamlining financial, customer relationship, and supply chain processes in a way that helps you drive business success.

United States and Canada toll-free: (888) 477-7989

Worldwide: (1) (701) 281-6500

[www.microsoft.com/dynamics](http://www.microsoft.com/dynamics)

© 2016 Microsoft Corporation. All rights reserved.

This document is provided "as-is." Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it.

Some examples are for illustration only and are fictitious. No real association is intended or inferred.

This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.