

# Data management overview

Article • 09/10/2024

This article describes how you can use the data management framework to manage data entities and data entity packages in finance and operations.

The data management framework consists of the following concepts:

- **Data entities** - A data entity is a conceptual abstraction and encapsulation of one or more underlying tables. A data entity represents a common data concept or functionality, for example, Customers or Vendors. Data entities are intended to be easily understood by users familiar with business concepts. After data entities are created, you can reuse them through the Excel Add-in, use them to define import/export packages, or use them for integrations.
- **Data project** - A project that contains configured data entities, which include mapping and default processing options.
- **Data job** - A job that contains an execution instance of the data project, uploaded files, schedule (recurrence), and processing options.
- **Job history** - Histories of source to staging and staging to target jobs.
- **Data package** - A single compressed file that contains a data project manifest and data files. This is generated from a data job and used for import or export of multiple files with the manifest.

The data management framework supports using data entities in the following core data management scenarios:

- Data migration
- Set up and copy configurations
- Integration

## Data entities

Data entities provide conceptual abstraction and encapsulation of underlying table schema that represent data concepts and functionalities. In Microsoft Dynamics AX 2012, most tables, like the Customer and Vendor tables, were de-normalized and split into multiple tables. This was beneficial from a database design point of view, but made it difficult for implementers and ISVs to use without a thorough understanding of the physical schema. Data entities were introduced as part of data management to be used as a layer of abstraction to easily understand by using business concepts. In previous versions there were multiple ways to manage data, such as Microsoft Excel Add-ins, AIF, and DIXF. The concept of data entities combines those different concepts into one. After

# Data entities overview

Article • 05/20/2024

This article defines and provides an overview of data entities. It includes information about the capabilities of data entities, the scenarios that they support, the categories that are used for them, and the methods for creating them.

## Overview

A *data entity* is an abstraction from the physical implementation of database tables. For example, in normalized tables, a lot of the data for each customer 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 entity provides conceptual **abstraction** and **encapsulation** (de-normalized view) of underlying table schemas to represent key data concepts and functionalities.

## Capabilities

A data entity has the following capabilities:

- It replaces diverging and fragmented concepts of AXD, Data Import/Export Framework (DIXF) entities, and aggregate queries with single concept.
- It provides a single stack to capture business logic, and to enable scenarios such as import/export, integration, and programmability.
- It becomes the primary mechanism for exporting and importing data packages for Application Lifecycle Management (ALM) and demo data scenarios.
- It can be exposed as OData services, and then used in tabular-style synchronous integration scenarios and Microsoft Office integrations.

# Configuration data projects

Article • 07/01/2022

Configuration data projects are used to manage the movement of company configuration data between instances of your application. They are intended to support the following scenarios:

- **Export of configurations** – Create configurations of entities, and use the data management framework to export the configurations to a package.
- **Import of configurations** – Upload a configuration package, and use the data management framework to import the package.

Configuration data packages are created by using data import and export projects in the **Data management** workspace. The **Data import** and **Data export** pages let you add and remove the entities that you require in order to manage the movement of company and shared data. After you create the list of entities in your configuration, you can export or import the configuration by using the data management framework to create a package. You can export packages locally and then move them to another instance for import.

Configuration data templates are predefined lists of entities for each module area that can be used in a data project. You can create, view, and modify these templates by using the **Template** page in the **Data management** workspace.

## Important

Default configuration templates were delivered in Microsoft Dynamics 365 Finance, Enterprise edition July 2017 update. The Configuration data project feature is available in Microsoft Dynamics 365 for Operations platform update 7. You can create and use your own templates in the current product release.

## Process for working with configuration data projects

We recommend that you follow this process when you start to use configuration data projects.

1. Set up your system by getting the new data configuration user interface (UI) and setting default file extensions.
2. Set up configuration templates for both export and import.

# Copy configuration data between companies or legal entities overview

Article • 02/02/2024

There are two options for copying configuration data in finance and operations:

- To move data between instances, you must first export it from one company, and then import it to another company.
- To move data from one legal entity to another legal entity in the same instance, you can use the **Copy into legal entity** feature.

## Export a configuration

The **Data management** workspace is your hub for managing configuration data projects and exporting data packages. To build a configuration, you must define a data project and export the information that is represented by entities.

To create an export configuration data project, follow these steps.

1. Open the **Data management** workspace. If you're in Standard view, select **Enhanced view**.
2. Select the **Export** tile.
3. Select **New** to create an export configuration data project, and enter an ID and name for the configuration.
4. Set the operation type for the data project to **Export**, and set the project category to **Configuration**.
5. Add the entities that represent the information that you want to export. You can add entities by using several methods:
  - **Add one entity** – Enter the first part of the name of the entity until it appears in the lookup.
  - **Add multiple entities** – Enter any part of the entity name, use the lookup for the module, enter any part of the tag name, or use the lookup for the entity category to show a list of entities. Press Tab to move focus away from the **Lookup** field and activate the filter. In the grid, select the entities to add.
  - **Add a file** – Browse to a file that contains a name that matches the name of an entity and a file name extension that matches the file name extension that

# Configuration data packages

Article • 04/23/2024

Configuration data packages are available as process data packages from Microsoft Dynamics Lifecycle Services (LCS). These data packages can help improve the repeatability of implementations and accelerate the configuration.

Data packages contain configuration entity spreadsheets. These entity spreadsheets contain best practice data that you can use to create an initial golden build. The data entities in the data packages are also sequenced appropriately to help guarantee a successful single-click import of the data.

The entity spreadsheets include three types of data:

- **Business data** – The spreadsheet contains standard business data for a mid-sized trade or retail company. This data combines best practices and business standards that can be used as a starting point for your configuration.
- **Sample data** – The spreadsheet contains data that can be used as an example for business-specific data. This data can be imported and used as an example, but it must be changed for individual business practices.
- **No data** – The spreadsheet doesn't contain any data. Several areas of the product are unique to each business and its business practices. These areas must be configured specifically for the organization. These spreadsheets should be reviewed and updated for the organization as appropriate.

For more information about the type of data that is included in each entity spreadsheet in the data packages see the [Data packages](#) section of this article. You can modify individual spreadsheets before you import the data packages, or you can import the data packages as they have been supplied and then update your data in the system.

## Using configuration data packages

You can access configuration data packages from LCS. You can either apply them to an LCS environment, or download them so that you can manually import them.

1. Open your LCS project, and open the Asset library.
2. In the list of asset types, select **Process data package**.
3. Click **Import**.
4. Select the configuration data package.
5. Click **Pick**.

# Configuration data packages

Article • 04/23/2024

Configuration data packages are available as process data packages from Microsoft Dynamics Lifecycle Services (LCS). These data packages can help improve the repeatability of implementations and accelerate the configuration.

Data packages contain configuration entity spreadsheets. These entity spreadsheets contain best practice data that you can use to create an initial golden build. The data entities in the data packages are also sequenced appropriately to help guarantee a successful single-click import of the data.

The entity spreadsheets include three types of data:

- **Business data** – The spreadsheet contains standard business data for a mid-sized trade or retail company. This data combines best practices and business standards that can be used as a starting point for your configuration.
- **Sample data** – The spreadsheet contains data that can be used as an example for business-specific data. This data can be imported and used as an example, but it must be changed for individual business practices.
- **No data** – The spreadsheet doesn't contain any data. Several areas of the product are unique to each business and its business practices. These areas must be configured specifically for the organization. These spreadsheets should be reviewed and updated for the organization as appropriate.

For more information about the type of data that is included in each entity spreadsheet in the data packages see the [Data packages](#) section of this article. You can modify individual spreadsheets before you import the data packages, or you can import the data packages as they have been supplied and then update your data in the system.

## Using configuration data packages

You can access configuration data packages from LCS. You can either apply them to an LCS environment, or download them so that you can manually import them.

1. Open your LCS project, and open the Asset library.
2. In the list of asset types, select **Process data package**.
3. Click **Import**.
4. Select the configuration data package.
5. Click **Pick**.

# Configuration data templates

Article • 07/01/2022

Configuration data templates are predefined lists of entities for each module area that can be used in a data project. You can create, view, and modify these templates by using the **Template** page in the **Data management** workspace.

## Important

Default configuration templates available out-of-the-box will always have the latest version of an entity. Templates can be created from an existing data project as needed.

## Create a new configuration data template

The **Template** page in the **Data management** workspace provides tools that let you create a template of entities. This page resembles the configurations page, and the two features work in a similar manner. You must use **Enhanced view** to take advantage of the new features.

To create a template, follow these steps.

1. Select **New** to create a template. Enter an ID and name for the template. Notice that the status is set to **Draft**.
2. Add or remove entities as you require.
3. Organize the list by using the **Sort by** button to reorder your entities by entity group, or by unit, level, and sequence.
4. To change the sequence of any of the entities, manually edit the unit, level, or sequence. Alternatively, use the **Resequence** button to update any entities that you've selected. The **Resequence** button appears only if you select more than one entity.
5. To add filters to an entity, use the **Filter** button. Then review the results of the filters by using the **Preview** button. If you add a filter, the **Filter** button is changed to an **Edit** button.
6. If you don't want all fields to be mapped, you can use the **View map** button to exclude fields from the mapping.
7. Select **Validate template** to change the status to **Validated**.

Your template can now be used in a project. However, you might want to use some additional features to control the export process.

# Enable change tracking for entities

Article • 07/01/2022

Change tracking enables incremental export of data from finance and operations apps by using Data management. In an incremental export, only records that have changed are exported. To enable incremental export, you must enable change tracking on entities. If you don't enable change tracking on an entity, you can only enable a full export each time.

Change tracking can be enabled for both bring your own database (BYOD) and non-BYOD scenarios. This includes retrieving record changes through Dataverse virtual entities.

## Note

Change tracking will track record deletion only for bring your own database (BYOD) and Dataverse virtual entity use cases, if the entity supports it. Other non-BYOD scenarios will not include tracking record deletion. Deletion is tracked only for the root data source in the entity.

## Enable change tracking for BYOD

You can enable change tracking when you publish one or more entities to a data store (BYOD).

1. In the **Data management** workspace, select **Configure entity export to database**.

2. Select the database to export data to, and then select **Publish**.

You can publish one or more entities to your database. Select **Show published only** to see a list of entities that have previously been published.

3. Select an entity that is published, and then select **Change tracking**.

4. Select the appropriate option for change tracking for your environment.

An entity can be modeled by using more than one table. The options let you specify the granularity at which changes can be tracked in an entity.

# Row version change tracking for tables and data entities

Article • 02/07/2024

Finance and operations apps have a change tracking functionality option that's known as *row version change tracking*. This option enables Microsoft Dataverse to be used for incremental synchronization of data. Change tracking is a prerequisite for several features, such as Data archival, Synapse integration, Mobile offline, and Relevance search. The eventual goal is to unify all existing finance and operations data synchronization frameworks into one that's based on Dataverse synchronization services.

For row version change tracking functionality, a new column of type, **rowversion**, must be added to all tables in the data entity that requires change tracking. For more information about the **rowversion** column type, see [rowversion](#). For information about how to add a **rowversion** column to a table, see [Enable row version change tracking for tables](#).

The **rowversion** column performs version stamping of table rows. SQL Server maintains a database-level counter that's incremented for each insert or update operation. Changes to a table row can be detected by comparing the current value in the **rowversion** column with the previous value.

## Enable row version change tracking functionality

Dynamics 365 Finance version 10.0.34 or later requires that you enable the **Sql row version change tracking** configuration key on the **License configuration** page (**System administration > Setup > Licensing > License configuration**). Configuration keys are edited only in maintenance mode. For more information about maintenance mode, see [Maintenance mode](#). When you exit maintenance mode after you enable the **Sql row version change tracking** configuration key, the database synchronization adds a **rowversion** column to tables that are enabled for row version change tracking.

### Note

The **rowversion** column is read-only in SQL Server. Therefore, direct SQL Data Manipulation Language (DML) statements that are run by using the X++ [Statement](#)

# Find information about standard data entities

Article • 08/12/2022

## ⓘ Important

Customers using Human Resources, the functionality noted in this article is currently available in both the stand-alone Dynamics 365 Human Resources and the merged Finance infrastructure. Navigation might be different than noted while we make updates. If you need to find a specific page, you can use Search.

The application ships with many default data entities. Data entities are frequently updated, so for documentation, we rely on the data entity templates to indicate which order data entities should be imported in, and on reports for a list of data entities that ship with each release.

## Configuration data packages

Configuration data packages on Microsoft Dynamics Lifecycle Services (LCS) contain configuration entity spreadsheets. Configuration entity spreadsheets contain best practice data that you can use to create an initial golden build of an implementation. The data entities in the data packages are also sequenced appropriately using an XML file to help guarantee a successful single-click import of the data. We recommend that you download and review the configuration data packages to understand how we recommend that you order your data imports. For more information, see [Configuration data templates](#) and [Copy configuration data between companies or legal entities overview](#).

## Reports

Microsoft provides the following reports for data entities, which can be downloaded from [Technical reference reports](#):

- Aggregate data entities: Lists the aggregate data entities, and the fields that each contains.
- Aggregate measures: Lists the aggregate measures.
- Config keys: Lists the configuration keys.
- Config key groups: Lists the configuration key groups.

# Data templates with multiple worksheets

Article • 09/29/2023

## Important

Customers using Human Resources, the functionality noted in this article is currently available in both the stand-alone Dynamics 365 Human Resources and the merged Finance infrastructure. Navigation might be different than noted while we make updates. If you need to find a specific page, you can use Search.

Data management in the application supports Microsoft Excel-based templates for data entities. These templates can contain one or more worksheets. Templates with multiple worksheets are often used when it is convenient to manage data in a single file and import it to multiple data entities. An example would be sites and warehouses.

## Upload a file once and map it to all entities

Let's take an example where there is one Excel file with worksheets called **Sites** and **Warehouses**. To set up the data import project, you would add the first data entity, **Sites** and then upload the file. You will be able to select **Sites** as the worksheet to be used for this entity.

If you add the second entity **Warehouses** without leaving the **Add file** form, the worksheet lookup will let you select the **Warehouses** worksheet without having to upload the file again. The only reason to upload a new file would be if the **Warehouses** data was in a different file.

# Enable change tracking for entities

Article • 07/01/2022

Change tracking enables incremental export of data from finance and operations apps by using Data management. In an incremental export, only records that have changed are exported. To enable incremental export, you must enable change tracking on entities. If you don't enable change tracking on an entity, you can only enable a full export each time.

Change tracking can be enabled for both bring your own database (BYOD) and non-BYOD scenarios. This includes retrieving record changes through Dataverse virtual entities.

## Note

Change tracking will track record deletion only for bring your own database (BYOD) and Dataverse virtual entity use cases, if the entity supports it. Other non-BYOD scenarios will not include tracking record deletion. Deletion is tracked only for the root data source in the entity.

## Enable change tracking for BYOD

You can enable change tracking when you publish one or more entities to a data store (BYOD).

1. In the **Data management** workspace, select **Configure entity export to database**.

2. Select the database to export data to, and then select **Publish**.

You can publish one or more entities to your database. Select **Show published only** to see a list of entities that have previously been published.

3. Select an entity that is published, and then select **Change tracking**.

4. Select the appropriate option for change tracking for your environment.

An entity can be modeled by using more than one table. The options let you specify the granularity at which changes can be tracked in an entity.

# Configuration keys and data entities

Article • 04/23/2024

Before you use data entities to import or export data, we recommended that you first determine the impact of configuration keys on the data entities that you are planning to use.

To learn more about configuration keys, see the [License codes and configuration keys report](#).

## Configuration key assignments

Configuration keys can be assigned to one or all of the following artifacts.

- Data entities
- Tables used as data sources
- Table fields
- Data entity fields

The following table summarizes how configuration key values on the different artifacts that underlie an object change the expected behavior of the object.

[+] Expand table

Configuration key setting on data entity	Configuration key setting on table	Configuration key setting on table field	Configuration key on data entity field	Expected behavior
Disabled	Not evaluated	Not evaluated	Not evaluated	If the configuration key for the data entity is disabled, the data entity will not be functional. It does not matter whether the configuration keys in the underlying tables and fields are enabled or disabled.
Enabled	Disabled	Not evaluated	Not evaluated	If the configuration key for a data entity is enabled, the data management framework checks the configuration key on each of the underlying tables. If the

# Configuration data packages

Article • 04/23/2024

Configuration data packages are available as process data packages from Microsoft Dynamics Lifecycle Services (LCS). These data packages can help improve the repeatability of implementations and accelerate the configuration.

Data packages contain configuration entity spreadsheets. These entity spreadsheets contain best practice data that you can use to create an initial golden build. The data entities in the data packages are also sequenced appropriately to help guarantee a successful single-click import of the data.

The entity spreadsheets include three types of data:

- **Business data** – The spreadsheet contains standard business data for a mid-sized trade or retail company. This data combines best practices and business standards that can be used as a starting point for your configuration.
- **Sample data** – The spreadsheet contains data that can be used as an example for business-specific data. This data can be imported and used as an example, but it must be changed for individual business practices.
- **No data** – The spreadsheet doesn't contain any data. Several areas of the product are unique to each business and its business practices. These areas must be configured specifically for the organization. These spreadsheets should be reviewed and updated for the organization as appropriate.

For more information about the type of data that is included in each entity spreadsheet in the data packages see the [Data packages](#) section of this article. You can modify individual spreadsheets before you import the data packages, or you can import the data packages as they have been supplied and then update your data in the system.

## Using configuration data packages

You can access configuration data packages from LCS. You can either apply them to an LCS environment, or download them so that you can manually import them.

1. Open your LCS project, and open the Asset library.
2. In the list of asset types, select **Process data package**.
3. Click **Import**.
4. Select the configuration data package.
5. Click **Pick**.

# Import data for customers in Dynamics 365 projects

Article • 08/07/2024

**Applies to:** *Dynamics 365 Commerce, Dynamics 365 Finance, Dynamics 365 Intelligent Order Management, Dynamics 365 Project Operations, Dynamics 365 Supply Chain Management*

You can use data entities to import data for customers in Dynamics 365 projects, such as for data migration or integration scenarios. This article outlines the contexts and use cases with links to the relevant entities.

## Context and problem

Dynamics 365 contains multiple entities for importing customers in a Dynamics 365 project. Choosing the best one requires understanding the similarities and differences between the available entities. The **Customer definitions** entity is recommended for most scenarios. The **Customer definitions** entity, when combined with the **Customers details V2** entity, supports most of the same functionality as the **Customers V3** entity.

The highest performance can usually be achieved using a combination of the **Customer definitions** and **Customer details V2** entities. In most cases, using one of those entities, or even using both of them, one after the other, will outperform the **Customers V3** entity.

Use the **Customers V3** entity when any of the approximately 80 fields that aren't supported by the **Customer definitions** and **Customer details V2** entities must be included in the import.

## Entities

- [Customer definitions](#)
- [Customers V3](#)
- [Customer details V2](#)

## Next steps

- Learn how you can use entities to import sales orders at [Import sales orders](#)

# Customer definitions entity

Article • 04/28/2023

The **Customers definitions** entity supports creating and updating customers; and includes the most common fields for a typical customer.

## When to use this entity

Use the **Customer definitions** entity as the default entity for importing customers as long as it contains the necessary fields, or the missing fields can be added from the existing data sources, and when performance is a high priority.

## Summary

[ ] [Expand table](#)

Type	Name
Data management entity name	<b>Customer definitions</b>
OData public entity	CustomerBase
OData public collection	CustomersBase
Related menu items	Accounts receivable / Customers / All customers
Related entities	Customers V3, Customer details V2
Performance pattern	Multiple threads supported – Recommendation is to set <b>Import threshold record count</b> to 1000 and set <b>Import task count</b> to 8. For more information, see <a href="#">Parallel imports</a> .
Application Object Tree (AOT) name	CustCustomerBaseEntity

## Fields

[ ] [Expand table](#)

# Customers V3 entity

Article • 04/28/2023

The **Customers V3** entity supports creating and updating customers; and includes all the fields for a typical customer. It also includes about 80 fields that the other customer entities support.

## When to use this entity

Use the **Customers V3** entity when the scenario includes approximately 80 fields that aren't supported by the **Customer definitions** and **Customer details V2** entities. Do not use the **Customers V3** entity in scenarios where high performance is required, including where the volume or frequency is high.

## Summary

[ ] [Expand table](#)

Type	Name
Data management entity name	Customers V3
OData public entity	CustomerV3
OData public collection	CustomersV3
Related menu items	Accounts receivable / Customers / All customers
Related entities	Customer definitions, Customer details V2
Performance pattern	Single thread only
Application Object Tree (AOT) name	CustCustomerV3Entity

## Fields

[ ] [Expand table](#)

Field	Description
CUSTOMERACCOUNT	Specifies the primary key.

# Customer details V2 entity

Article • 04/28/2023

The **Customer details V2 entity** supports updating the specialized fields for a customer. It doesn't support creating customers.

## When to use this entity

Use the **Customer details V2 entity** to supplement the **Customer definitions entity**. The **Customer details V2 entity** is most often used to update the INVOICEACCOUNT and DEFAULTDIMENSIONDISPLAYVALUE fields. You can update any of the fields it supports independently of the fields being updated with the **Customer definitions entity**, especially if the fields from either entity are updated with different patterns.

## Summary

[+] Expand table

Type	Name
Data management entity name	Customer details V2
OData public entity	N/A
OData public collection	N/A
Related menu items	Accounts receivable / Customers / All customers
Related entities	Customers V3, Customer definitions
Performance pattern	Multiple threads supported – Recommendation is to set <b>Import threshold record count</b> to 1000 and set <b>Import task count</b> to 8. For more information, see <a href="#">Parallel imports</a> .
Application Object Tree (AOT) name	CustCustomerDetailV2Entity

## Fields

[+] Expand table

# Import product data in Dynamics 365 projects

Article • 08/07/2024

**Applies to:** *Dynamics 365 Commerce, Dynamics 365 Intelligent Order Management, Dynamics 365 Project Operations, Dynamics 365 Supply Chain Management*

You can use data entities to import data for goods in Dynamics 365 projects, such as for data migration or integration scenarios. This article outlines the contexts and use cases with links to the relevant entities.

## Context and problem

Dynamics 365 contains multiple entities for importing products in a Dynamics 365 project. Choosing the best one depends on the type of products that you want to import.

- The **Products V2** entity specifically refers to shared distinct products and product masters. These products are shared between all legal entities and need to be released before using them in transactions.
- The **Released products V2** entity requires that the shared products be already created. When using this entity, the product will be released on the current legal entity.

To create both the shared product and releasing the product to a specific legal entity in one step, use the **Released product creation V2** entity. However, this entity doesn't allow any updates, and the fields available are a limited set.

## Entities

- [Products V2](#)
- [Released products V2](#)

## Next steps

- [Released product creation V2](#)
- [Product variants](#)

# Products V2 entity

Article • 04/28/2023

The **Products V2** entity imports and updates shared distinct products and product masters.

## When to use this entity

Use the **Products V2** entity when creating or updating shared distinct products and product masters without them being released to a specific legal entity.

When importing shared product variants, use the **Product Variants** entity instead. The product master should already exist.

## Summary

[ ] [Expand table](#)

Type	Name
Data management entity name	Products V2
OData public entity	ProductV2
OData public collection	ProductsV2
Related menu items	Product information management/ Products/ All products and product masters Product information management/ Products/ Products Product information management/ Products/ Product masters
Related entities	Release products V2, Released product creation V2
Performance pattern	Multiple threads supported – Recommendation is to set <b>Import threshold record count</b> to 1000 and set <b>Import task count</b> to 8. For more information, see <a href="#">Parallel imports</a> .
Application Object Tree (AOT) name	EcoResProductV2Entity

## Fields

# Released products V2 entity

Article • 04/28/2023

The **Released products V2 entity** imports and updates released distinct products and product masters. It requires that the shared product has already been created.

## When to use this entity

Use the **Released products V2 entity** to create and update released distinct products and product masters in a specific legal entity. The shared products must already exist.

When importing released product variants, use the **Released product variants entity** instead. This entity requires that shared product variants already exist.

## Summary

[+] Expand table

Type	Name
Data management entity name	Released products V2
OData public entity	ReleasedProductV2
OData public collection	ReleasedProductsV2
Related menu items	Product information management/ Products/ Released products
Related entities	Release products V2, Released product creation V2
Performance pattern	Multiple threads supported – Recommendation is to set <b>Import threshold record count</b> to 1000 and set <b>Import task count</b> to 8. For more information, see <a href="#">Parallel imports</a> .
Application Object Tree (AOT) name	EcoResReleasedProductV2Entity

## Fields

[+] Expand table

# Import sales order data in Dynamics 365 projects

Article • 08/07/2024

*Applies to: Dynamics 365 Commerce, Dynamics 365 Finance, Dynamics 365 Intelligent Order Management, Dynamics 365 Project Operations, Dynamics 365 Supply Chain Management*

You can use data entities to import data for sales orders in Dynamics 365 projects, such as for data migration or integration scenarios. This article outlines the contexts and use cases with links to the relevant entities.

## Context and problem

Dynamics 365 contains two entities for importing and updating sales orders:

- **Sales order headers V2**

Creates the general specifications of the order.

- **Sales order lines V2**

Creates the lines, including the specific concepts such as items, quantities, and sales price.

You must first create the headers of sales orders before you import the lines. Sales lines are always related to a specific header.

## Entities

- [Sales order headers V2](#)
- [Sales order lines V2](#)

## Next steps

- Import customer payments

## Related information

# Sales order headers V2

Article • 04/28/2023

The **Sales order headers V2** entity supports importing and updating sales order headers.

## When to use this entity

Use the **Sales order headers V2** entity first when you create sales orders. Then create sales line using the **Sales order lines V2** entity.

## Summary

[ ] Expand table

Type	Name
Data management entity name	Sales order headers V2
OData public entity	SalesOrderHeaderV2
OData public collection	SalesOrderHeadersV2
Related menu items	Accounts receivable / Orders / All sales orders Sales and marketing/ Sales orders/ All sales orders
Related entities	Sales order lines V2
Performance pattern	Single thread recommended
Application Object Tree (AOT) name	SalesOrderHeaderV2Entity

## Fields

[ ] Expand table

Field	Description
SALESORDERNUMBER	Specifies the primary key. This field is required.
CURRENCYCODE	Currency code. This field requires that currencies are configured in <b>General ledger/ Currencies/ Currencies</b> .

# Sales order lines V2

Article • 04/28/2023

The **Sales order lines V2** entity supports importing and updating sales order lines.

## When to use this entity

Use the **Sales order lines V2** entity after the sales order header has already been created.

## Summary

[ ] Expand table

Type	Name
Data management entity name	Sales order lines V2
OData public entity	SalesOrderLine
OData public collection	SalesOrderLines
Related menu items	Accounts receivable/ Orders/ All sales orders Sales and marketing/ Sales orders/ All sales orders
Related entities	Sales order headers V2
Performance pattern	Single thread recommended
Application Object Tree (AOT) name	SalesOrderLineV2Entity

## Fields

[ ] Expand table

Field	Description
INVENTORYLOTID	Specifies a unique identifier assigned to a specific lot of inventory items. The INVENTORYLOTID field requires that the inventory number sequence is previously configured. Otherwise, you can change the entity mapping and select Auto-generated. This field is the primary key.

# Import data for vendor invoices in Dynamics 365 projects

Article • 08/07/2024

*Applies to: Dynamics 365 Commerce, Dynamics 365 Finance, Dynamics 365 Intelligent Order Management, Dynamics 365 Project Operations, Dynamics 365 Supply Chain Management*

You can use data entities to import data for vendor invoices in Dynamics 365 projects, such as for data migration or integration scenarios. This article outlines the contexts and use cases with links to the relevant entities.

## Context and problem

Use the **Vendor invoice header** and **Vendor invoice line** entities to import vendor invoices with the most typical fields.

Vendor invoices can also have charges and attachments that must be imported. The **Vendor invoice charges** and **Vendor invoice document attachment V2** entities have a dependency on the vendor invoice header. The **Vendor invoice line charges** entity has a dependency on the vendor invoice line.

## Entities

- [Vendor invoice header](#)
- [Vendor invoice line](#)
- [Vendor invoice document attachment V2](#)
- [Vendor invoice charges](#)
- [Vendor invoice line charges](#)

## Next steps

- Import vendor payments

## Related information

# Vendor invoice header entity

Article • 06/20/2023

The **Vendor invoice header** entity supports creating and updating vendor invoice headers; and includes all the necessary fields.

## When to use this entity

Use the **Vendor invoice header** entity to create a vendor invoice header before importing anything else related to a vendor invoice.

## Summary

[+] Expand table

Type	Name
Data management entity name	Vendor invoice header
OData public entity	VendorInvoiceHeader
OData public collection	VendorInvoiceHeaders
Related menu items	Accounts payable / Invoices / Pending vendor invoices Accounts payable / Invoices / Open vendor invoices Accounts payable / Vendors / Invoice
Related entities	Vendor invoice charges Vendor invoice document attachment V2 Vendor invoice line
Performance pattern	Multiple threads supported
Application Object Tree (AOT) name	VendorInvoiceHeaderEntity

## Fields

[+] Expand table

Field	Description
CURRENCY	The currency for the vendor invoice.

# Vendor invoice line entity

Article • 04/28/2023

The **Vendor invoice line** entity supports creating and updating vendor invoices lines. It includes all the fields related to vendor invoice lines.

## When to use this entity

Use the **Vendor invoice line** entity to create vendor invoices before importing the related vendor invoice lines.

## Summary

[+] Expand table

Type	Name
Data management entity name	Vendor invoice line
OData public entity	VendorInvoiceLine
OData public collection	VendorInvoiceLines
Related menu items	Accounts payable / Invoices / Pending vendor invoices / Edit / Lines Accounts payable / Invoices / Open vendor invoices / New / Vendor tax invoice / Lines Accounts payable / Invoices / Open vendor invoices / New / One-time vendor and invoice / Lines Accounts payable / Vendors / Invoice / New / Vendor invoice / Lines
Related entities	Vendor invoice line charges
Performance pattern	Multiple threads supported
Application Object Tree (AOT) name	VendorInvoiceLineEntity

## Fields

[+] Expand table

# Vendor invoice document attachment V2 entity

Article • 04/28/2023

The **Vendor invoice document attachment V2 entity** supports creating and updating documents attached to a vendor invoice header.

## When to use this entity

Use this entity to create or update vendor invoice header document attachments.

## Summary

[+] Expand table

Type	Name
Data management entity name	Vendor invoice document attachment V2
OData public entity	VendorInvoiceDocumentAttachment
OData public collection	VendorInvoiceDocumentAttachments
Related menu items	Accounts payable / Invoices / Pending vendor invoices Accounts payable / Invoices / Open vendor invoices Accounts payable / Vendors / Invoice
Related entities	Vendor invoice header
Performance pattern	Multiple threads supported – Recommendation is to set <b>Import threshold record count</b> to 1000 and set <b>Import task count</b> to 8. For more information, see <a href="#">Parallel imports</a> .
Application Object Tree (AOT) name	VendorInvoiceDocumentAttachmentV2Entity

## Fields

[+] Expand table

# Vendor invoice charges entity

Article • 04/28/2023

The **Vendor invoice charges** entity supports creating and updating vendor invoice charges applied to the invoice header. It includes all the fields related to the vendor invoices charges.

## When to use this entity

The **Vendor invoice charges** entity is used to add charges to a vendor invoice header.

## Summary

[ ] [Expand table](#)

Type	Name
Data management entity name	Vendor invoice charges
OData public entity	VendorInvoiceHeaderCharge
OData public collection	VendorInvoiceHeaderCharges
Related menu items	Accounts payable / Invoices / Pending vendor invoices Accounts payable / Invoices / Open vendor invoices Accounts payable / Vendors / Invoice
Related entities	Vendor invoice header
Performance pattern	Multiple threads supported
Application Object Tree (AOT) name	VendorInvoiceHeaderChargeEntity

## Fields

[ ] [Expand table](#)

Field	Description
Headerreference	Specifies the first segment of the primary key. Specifies a foreign key to a <b>Vendor invoice header</b> .

# Vendor invoice line charges entity

Article • 04/28/2023

The **Vendor invoice line charges** entity supports creating and updating vendor invoice line level charges applied at invoice line document; and includes all the fields related to the vendor invoices line charges.

## When to use this entity

The **Vendor invoice line charges** entity can be used to apply charges to a vendor invoice line.

## Summary

[\[\] Expand table](#)

Type	Name
Data management entity name	Vendor invoice line charges
OData public entity	VendorInvoiceLineCharge
OData public collection	VendorInvoiceLineCharges
Related menu item	Accounts payable / Invoices / Pending vendor invoices Accounts payable / Invoices / Open vendor invoices Accounts payable / Vendors / Invoice
Related entities	Vendor invoice line
Performance pattern	Multiple threads supported
Application Object Tree (AOT) name	VendorInvoiceLineChargeEntity

## Fields

[\[\] Expand table](#)

Field	Description
Headerreference	Specifies the first segment of the primary key. Specifies a foreign key to a <b>Vendor invoice header</b> .

# Import general journals in Dynamics 365 projects

Article • 02/20/2024

*Applies to Dynamics 365 Commerce, Dynamics 365 Finance, Dynamics 365 Intelligent Order Management, Dynamics 365 Project Operations, Dynamics 365 Supply Chain Management*

You can use data entities to import data from general journals in Dynamics 365 projects, such as for data migration or integration scenarios. This article outlines the contexts and use cases with links to the relevant entities.

## Context and problem

There are separate entities for importing general journals with the data management framework (DMF) and OData. The **General journal** entity combines the header and lines into a single entity and supports DMF. We recommend that you use this entity in most scenarios. The **Ledger journal header** entity and **Ledger journal line** entity separate the header and line and support OData, including the Excel add-in.

## Entities

- [General journal](#)
- [Ledger journal header](#)
- [Ledger journal line](#)

## Next steps

- Learn how to use entities to import sales orders at [Import sales orders](#).
- Learn how to use entities to import customers at [Import data for customers](#).

## Related content

- [General journal processing](#)
- [Importing vouchers by using the General journal entity](#)
- [Financial tags](#)
- [Financial journal posting performance](#)
- [Journal posting failure because of imbalance](#)

# General journal entity

Article • 02/20/2024

**Applies to:** Dynamics 365 Commerce, Dynamics 365 Finance, Dynamics 365 Intelligent Order Management, Dynamics 365 Project Operations, Dynamics 365 Supply Chain Management

The **General journal** entity supports creating and updating general journal headers and lines for the Bank, Customer, Ledger, and Vendor account types.

## When to use this entity

The **General journal** entity should be used for all high volume import scenarios. Using the data management framework (DMF) is recommended for high volume scenarios, and the **General journal** entity supports set-based processing that performs well with high volume. This documentation assumes the **General journal** entity is using set-based processing.

## Summary

[ ] [Expand table](#)

Data management entity name	<b>General journal</b>
OData public entity	N/A
OData public collection	N/A
Related menu item	General ledger / Journal entries / General journals General ledger / Journal entries / Journal post
Related entities	LedgerJournalHeader, LedgerJournalLine
Performance pattern	Set-based: Disabling set-based processing or using multiple threads isn't recommended.
Application Object Tree (AOT) name	LedgerJournalEntity

## Fields

# LedgerJournalHeader entity

Article • 02/20/2024

The **LedgerJournalHeader** entity supports creating and updating general journal headers.

## When to use this entity

Use the **LedgerJournalHeader** entity for synchronous scenarios with OData. For high volume scenarios, we recommend you use the **General journal** entity that uses the data management framework.

## Summary

[ ] [Expand table](#)

Data management entity name	<b>LedgerJournalHeader</b>
OData public entity	LedgerJournalHeader
OData public collection	LedgerJournalHeaders
Related menu item	General ledger / Journal entries / General journals
Related entities	LedgerJournalLine, General journal
Performance pattern	N/A
Application Object Tree (AOT) name	LedgerJournalHeaderEntity

## Fields

[ ] [Expand table](#)

Field name	Comments
JOURNALBATCHNUMBER	Specifies the primary key. This field is required.
LINENUMBER	Specifies the second segment of the primary key. This field isn't required for an insert.
JOURNALNAME	Specifies the journal name template. This field is required.

# LedgerJournalLine entity

Article • 02/20/2024

The **LedgerJournalLine** entity supports creating and updating general journal lines.

## When to use this entity

The **LedgerJournalLine** entity should be used for synchronous scenarios with OData. The **General journal** entity uses Data management and is recommended for high volume scenarios.

## Summary

[+] Expand table

Data management entity name	N/A
OData public entity	LedgerJournalLine
OData public collection	LedgerJournalLines
Related menu items	General ledger / Journal entries / General journals
Related entities	LedgerJournalHeader, General journal
Performance pattern	N/A
Application Object Tree (AOT) name	LedgerJournalLineEntity

## Fields

[+] Expand table

Field name	Comments
JOURNALBATCHNUMBER	Specifies the first segment of the primary key. This field is required. It must reference an existing, unposted header to be able to create or update lines.
LINENUMBER	Specifies the second segment of the primary key. This field isn't required for an insert.

# Clean up data management job history

Article • 01/27/2025

## Clean up data

1. Go to Data management > Job history cleanup.
2. In the Job history pane, set the Number of days to retain history, Number of hours to execute the job, and Batch job recurrence fields.
3. To schedule the job to run regularly in the background, select the Batch processing field.
4. To define a recurrence, select Recurrence, and then, in Recurrence definition, select No end date.

### Note

To clean large staging tables, we recommend that you set an execution time of six hours and a batch recurrence of at least once per day.

## Execution history cleanup batch error

If a Job history cleanup batch job has already been scheduled, it must be deleted before a new recurrence can be rescheduled. If you try to schedule a batch recurrence when one has already been scheduled, an error occurs. To address the error, follow these steps.

1. Go to System administration > Inquiries > Batch jobs.
2. Search for the description Job history cleanup.
3. Delete batch jobs that are in a Waiting state.
4. If a batch job is in an Executing state, you can either cancel it or select Remove recurrence. The Remove recurrence function removes the batch job schedule after the current cleanup execution is completed.
5. After the job is either deleted or in an Ended state, create a new batch job recurrence.

## Cleanup job history in cloud environments

Past job history cleanup activity is tracked in the DMFStagingHistoryCleanupTable table. It can be viewed by clicking **Cleanup job history** on the **Job history cleanup** page.

# Clean up staging tables

Article • 01/17/2024

In Microsoft Dynamics 365 Finance version 10.0.38, the **Truncate staging table** feature cleans up individual Data Management staging tables. The truncate action permanently removes all records from the staging table for the selected entity and should be used with caution. Currently, the execution of data import/export jobs that involve the selected staging table (imports or exports through staging) is affected if that staging table is truncated.

## Important

Before you use this feature, confirm that none of the currently running jobs involve the selected staging table.

This feature is available to users who have the Data Management Administrator role. It can be accessed via the data entities in the **Data management** workspace. Select a staging table, and then select **Truncate staging table**.

---

## Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

# Data import and export jobs overview

Article • 03/11/2025

To create and manage data import and export jobs, you use the **Data management** workspace. By default, the data import and export process creates a staging table for each entity in the target database. Staging tables let you verify, cleanup, or convert data before you move it.

## ⓘ Note

This article assumes that you are familiar with [data entities](#).

## Data import/export process

Here are the steps to import or export data.

1. Create an import or export job where you complete the following tasks:
  - Define the project category.
  - Identify the entities to import or export.
  - Set the data format for the job.
  - Sequence the entities, so that they're processed in logical groups and in an order that makes sense.
  - Determine whether to use staging tables.
2. Validate that the source data and target data are mapped correctly.
3. Verify the security for your import or export job.
4. Run the import or export job.
5. Validate that the job ran as expected by reviewing the job history.
6. Clean up the staging tables.

The remaining sections of this article provide more information about each step of the process.

## ⓘ Note

In order to refresh the Data import/export form to see the latest progress, use the form refresh icon. Browser level refresh isn't recommended because it interrupts

# Data import/export framework parameters

Article • 08/10/2023

This article describes the Data import/export framework parameters and options for data import and export.

To access the parameters, in finance and operations apps, go to **Data Management > Framework parameters**.

On the **Bring your own database** tab, the following parameter is available:

- **Enable all company export:**
  - Select **No** to export only selected companies.
  - Select **Yes** to export all companies.

On the **Compatibility options** tab > **Flat file compatibility options**, the following parameters are available:

- **Enable header for fixed width file:**
  - Select **No** if the exported file isn't required to contain headers.
  - Select **Yes** if the exported file is required to contain headers.
- **Validate file format with file extension:**
  - Select **No** if files should always be added to the project when there's a mismatch between the file types.
  - Select **Yes** if files should not be added to the project when there's a mismatch between the file types.

On the **Compatibility options** tab > **XML file compatibility options**, the following parameters are available:

- **Full export:**
  - Select **No** to export empty fields with null values. Here's an example:  
`<AmountCur/>`
  - Select **Yes** to export empty fields with default values. Here's an example:  
`<AmountCur>0.0000</AmountCur>`
- **Entity names in non uppercase:**
  - Select **No** to show entity names in uppercase letters.
  - Select **Yes** to show entity names in non-uppercase letters.

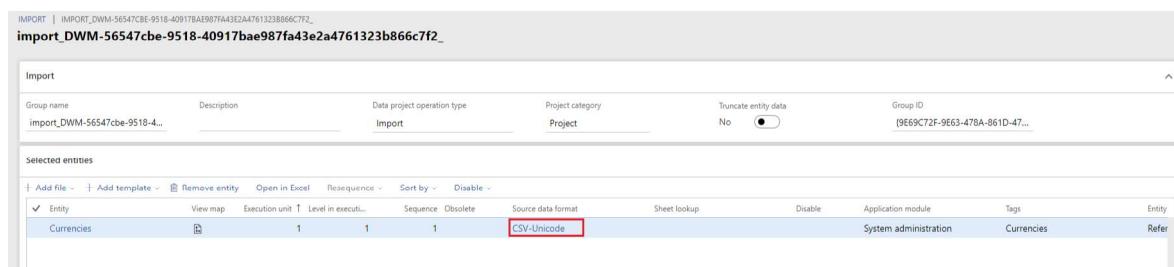
# Synchronize date and time in import jobs

Article • 09/29/2023

It's important to set the time zone for your import job to Coordinated Universal Time (UTC). You might see unexpected dates and times in your imported data if you use a different setting. Without the correct setting, the import process converts the UTC date to the local format, and then system settings converts it again.

This dual conversion causes dates to change between applications. For example, the dual conversion could cause an employee's start date to be different between Dynamics 365 Human Resources and Dynamics 365 Finance due to differences in local time zones. Setting the import job to UTC resolves this problem.

1. In Dynamics 365 finance and operations, select **Data management**.
2. Select **Import projects**, and then select the project.
3. Under **Source date format**, select **CSV-Unicode**.



4. Change **Timezone** to **Coordinated Universal Timezone**, and change **Language** to **En-US**.

# Importing vouchers by using the General journal entity

Article • 10/29/2024

This article provides tips for importing data into the General journal by using the General journal entity. Find technical information about the entity in [General journal entity](#).

You can use the General journal entity to import vouchers that have an account or offset account type of **Ledger**, **Customer**, **Vendor**, or **Bank**. The voucher can be entered as one line, using both the **Account** field and the **Offset account** field, or as a multi-line voucher, where only the **Account** field is used (and the **Offset account** is left blank on each line). The General journal entity doesn't support every account type. Instead, other entities exist for scenarios where different combinations of account types are required. For example, to import a project transaction, use the Project expense journal entity. Each entity is designed to support specific scenarios. This means other fields might be available in entities for those scenarios. However, other fields might not be available in entities for different scenarios.

## Data Management Framework and OData entities for general journal imports

The choice of frameworks and entities can significantly affect the efficiency and integrity of data handling processes. Two notable entities are the Data Management Framework (DMF) entity and the OData entity (specifically, **LedgerJournalLineEntity**). Each of these entities has its unique advantages and limitations. By understanding those advantages and limitations, organizations can make informed decisions about how to optimize their data import processes.

### DMF entity

The DMF entity is designed primarily for scenarios where high-volume imports must be handled. Because the DMF entity is optimized for tasks of this type, it's an invaluable tool for organizations that work with large datasets. Nevertheless, this entity also has limitations.

### Advantages of the DMF entity

# Optimize data migration for finance and operations apps

Article • 07/01/2022

Data migration is a key success factor in almost every implementation. A primary concern of some customers is the speed that data can be migrated at, especially if there are vast amounts of data and a small cutover window. The [Data migration framework](#) is also used to move data as part of business requirements and operations.

The information in this article represents a set of steps and actions that you can use to optimize the performance of data migration.

## Note

Testing results in a Tier-1 environment should not be compared or extrapolated to performance in a Tier-2 or higher sandbox environment.

Not all standard entities have been optimized for data migration. Some entities have been optimized for integration with Open Data Protocol (OData), and if a required entity can't be optimized to meet the performance requirements, we recommend that you [create a new optimized entity](#). A developer can accelerate this process by duplicating an existing entity.

Begin the optimization phase by using a subset of the data. For example, if you must import one million records, consider starting with 1,000 records, then increase the number to 10,000 records, and then increase it to 100,000 records.

After you've identified the entities that to use, you should go through the following sections to explore opportunities for optimization.

## Disable change tracking

You can [enable and disable change tracking](#) from the list of entities.

1. In the **Data management** workspace, select the **Data entities** tile.
2. On the **Target entities** page, select the entity in the grid, and then, on the Action Pane, on the **Change tracking** tab, select **Disable Change Tracking**.

## Enable set-based processing

# Data entity export performance tips

Article • 06/04/2024

This article provides tips that can help improve performance during export.

## Avoid computed columns that have JOINS

Computed columns shouldn't have joins. Instead, add all the tables that are required for the computation as data sources.

Computed columns are computed values that SQL Server must return as part of the process of running the data entity view. They have many possible uses, such as returning default values, casing logic, and formatting values. Computed column formula must be written so that SQL Server can quickly compute values and doesn't need large amounts of storage space.

When you write a computed column formula, the main thing to avoid is a join to some table or view. Such a join forces SQL Server to compute the values row by row and keep them all in temporary storage. Instead, add the tables that are needed for the formula as data sources in the entity view. SQL Server can then join all the table data to all the rows in one operation instead of repeating the process for each row.

To test whether a computation is causing slowness, follow these steps.

1. Connect to your sandbox database, and run the `sp_helptext` command to get the view definition.
2. Copy the output to another SQL Server Management Studio window, and make the following changes:
  - Remove the line breaks.
  - Remove `CREATE VIEW [SOME NAME] AS` from the start of the view definition command, so that only the `select` command is present.
3. Run the command, and record the time that's required to return all data.
4. Comment out the computed column formula, run the command again, and record the time that's required to return all data. If the required time is more than one to two seconds faster without the computation, the computation is causing slowness.

## Avoid implementing postLoad when possible

# Design principles and best practices for data entities

Article • 05/15/2024

This article describes design principles for data entities. It also includes guidelines for the names of data entities, fields, relation roles, roles, and OData EntityTypes and EntitySets.

## Entity design principles

A data entity should provide a holistic object that encapsulates the relevant business logic in a single consumable contract. The contract is then exposed through application interfaces (APIs), such as OData, import and export, integration, and the programming model. Each data entity should be designed to meet the following goals.

### Encapsulation

- Each entity should provide an abstraction between the physical data model and the consumer of the entity. The entity should encapsulate the underlying tables that, together, can define an object that has a specific purpose in the business. Consumers of the entity might be form clients, services, and integration.
- Each entity should encapsulate multiple related tables to represent the domain object. In some situations, single table entities are acceptable.

### A single public contract

- The public contract for an entity should be the same across all integration end points. For example, the customer entity must expose the same fields or API to both OData and import/export. This principle guarantees that the published schema for an entity is consistent, regardless of the mechanism for consumer interaction.
- When an entity is consumed, the business logic that is executed within the entity during CRUD operations must not vary based on the type of consumer.

### Simplicity

- The consumer of an entity should be able to interact with the entity based on the accepted industry or domain definitions of the entity. The behavior details of the

# Build and consume data entities

Article • 07/01/2022

This tutorial shows how to build an entity and how to consume some out-of-band (OOB) entities in an integration scenario. You will also preview how these data entities will be consumed in various integrations scenarios, such as data import and export, integration, and OData services.

When you are ready to build your first entity for production, you will need to:

- Create your own package and model. For more information, see [Models and packages](#).
- Create a new project and set the model property to the one that you just created.

## Prerequisites

This tutorial requires that you access an environment by using Remote Desktop, and that you are provisioned as an administrator on the instance.

Throughout this tutorial, `baseUrl` refers to the base URL of the instance.

- In the cloud environment, the base URL is obtained from Microsoft Dynamics Lifecycle Services (LCS).
- On a [local virtual machine \(VM\)](#), the base URL is  
`https://usnconeboxax1aos.cloud.onebox.dynamics.com`.
- Download FMLab sample code to C:. For details, see [FMLab sample code](#).

## Key concepts

- Developing a data entity in Microsoft Visual Studio
- Enabling a data entity for data management and OData services
- Consuming a data entity in integration scenarios

## Business problem

Fleet Management stores customer data in the FMCustomer table and address data in the FMAddressTable table. To access or update customer information, users must access multiple tables. Instead, you can create a business object that functionally represents customer information, and that you can use to build integration solutions.

# Behavioral properties on data entities

Article • 04/23/2024

Every data entity has properties that let you override the same property values on the tables or views that are the data sources of that entity. Your choices affect the behavior of the entity. In the following table, the first column lists the properties that are discussed in this article. The top row lists the levels where the property is found in the entity designer. The levels are listed in order of increasing granularity: the data source level is more granular than the entity level but less granular than the field level.

[+] Expand table

	Entity level	Data source level	Field level
ReadOnly	Applies	Applies	.
AllowEdit	.	.	Applies
AllowEditOnCreate	.	.	Applies
Mandatory	.	.	Applies

## Entity level

In the designer for your data entity, when you click the name at the root node, the **Properties** pane includes the **Is Read Only** property. The following table describes the behavioral differences between the **Yes** and **No** values of this property.

[+] Expand table

Group	Property name	Display name	Values	Default	Description
Behavior	IsReadOnly	Is Read Only	No, Yes	No	<ul style="list-style-type: none"><li><b>No:</b> Data modification operations (CUD) <i>are</i> allowed, <i>unless</i> an individual data source node in the entity's designer is set to <b>IsReadOnly = Yes</b>.</li><li><b>Yes:</b> Only read operations are allowed, regardless of the <b>IsReadOnly</b> settings on the individual data source nodes in the entity's designer.</li></ul>

# Computed columns and virtual fields in data entities

Article • 06/19/2024

This article provides information about computed and virtual fields, which are the two types of unmapped fields that a data entity can have. The article includes information about the properties of unmapped fields, and examples that show how to create, use, and test them.

The sample code is targeted towards creating or modifying an entity that is a part of solution that you own. Extending an existing entity requires slight modifications.

## Overview

A data entity can have additional *unmapped* fields beyond those that are directly mapped to fields of the data sources. There are mechanisms for generating values for unmapped fields:

- Custom X++ code
- SQL executed by Microsoft SQL Server

The two types of unmapped fields are computed and virtual. Unmapped fields always support read actions, but the feature specification might not require any development effort to support write actions.

## Computed field

- Value is generated by an SQL view computed column.
- During read, data is computed by SQL and is fetched directly from the view.
- For writes, custom X++ code must parse the input value and then write the parsed values to the regular fields of the data entity. The values are stored in the regular fields of the data sources of the entity.
- Computed fields are used mostly for reads.
- If possible, it's a good idea to use computed columns instead of virtual fields, because they are computed at the SQL Server level, whereas, virtual fields are computed row by row in X++.

## Virtual field

# Cross-company behavior of data entities

Article • 06/03/2022

This article provides information about how data entities interact with the cross-company concept. To understand this aspect of data entities, you must understand how tables and views apply the cross-company concept. Therefore, this article begins with a brief review of tables and views, and then explains how data entities are related.

## Review of tables and views for cross-company

Each table has a **SaveDataPerCompany** property, and each view has a **AllowCrossCompany** property. The following table describes these two properties.

[ ] [Expand table](#)

	Table	View
Property name	SaveDataPerCompany	AllowCrossCompany
Relevant CRUD mode	CUD	R
Timing of effect	Run time, Design time	Run time, mostly. At design time, this setting causes the view to have <b>dataAreaId</b> in its list of selected fields. However, the filter for a specific <b>dataAreaId</b> value is added later, at run time.
Meaning of value = Yes	At design time, the system automatically adds a <b>dataAreaId</b> field to the table, even though the field isn't displayed in the Application Object Tree (AOT). Every record in the table is tagged with the company (or legal entity) that it belongs to. The system automatically adds a filter to the SQL <b>Where</b> clause to limit the returned set of rows to one <b>dataAreaId</b> value.	At run time, the system does <i>not</i> automatically add a filter for <b>dataAreaId</b> on the <b>Where</b> clause of the SQL <b>Select</b> statement that it sends to the underlying Microsoft SQL Server system. Therefore, SQL <b>Select</b> statements from the view can return a set of records that contains records for <i>multiple</i> companies.
Meaning of value = No	The system does <i>not</i> add a <b>dataAreaId</b> field to the table. The table is said to be a shared table, because none of its records contain any formal company-specific data.	The system automatically adds a filter to the SQL <b>Where</b> clause to limit the returned set of rows to one <b>dataAreaId</b> value. However, the <b>AllowCrossCompany</b> property is

# Country/region codes and configuration keys

Article • 04/23/2024

This article provides scenarios that are applicable from an implementation perspective for both configuration keys and country/region.

## Customer table schema

[\[+\] Expand table](#)

Field name	Field label	Country/region context
CustNum	Customer number	
CustName	Customer name	
EinvoiceEANNum	EAN	DK
FiscalCode	Fiscal code	IT

## Sample data

[\[+\] Expand table](#)

CustNum	CustName	EinvoiceEANNum{DK}	FiscalCode{IT}	DataAreaId
1	Contoso Denmark	AA	{Empty}	DK
2	Contoso Italy	{Empty}	DD	IT

## Sample entity

[\[+\] Expand table](#)

Field name	Country/region context
CustomerNumber	
CustomerName	
EAN	DK

# Super types and sub types

Article • 04/26/2024

## ⓘ Important

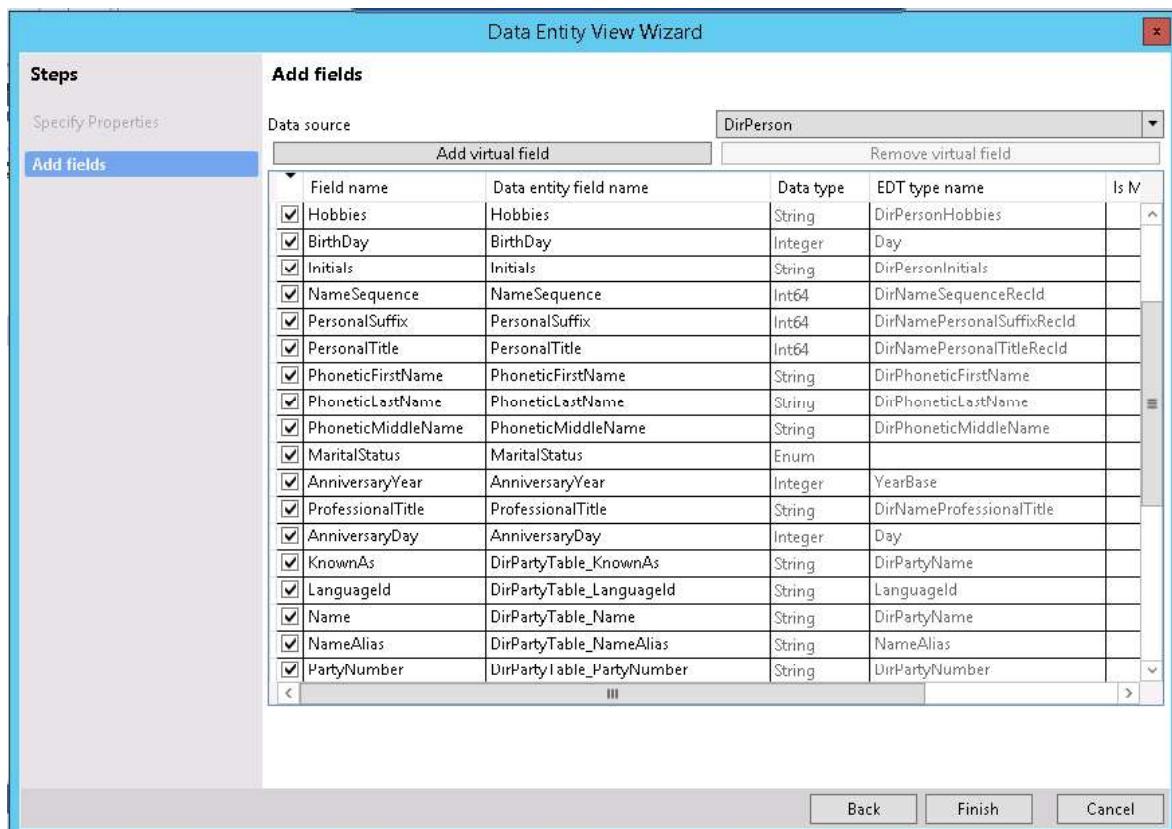
Customers using Human Resources, the functionality noted in this article is currently available in both the stand-alone Dynamics 365 Human Resources and the merged Finance infrastructure. Navigation might be different than noted while we make updates. If you need to find a specific page, you can use Search.

Describes support for inheritance patterns in data entities.

## Patterns

There are several ways to create entities for tables that involve inheritance:

- **Leaf/concrete type as data source:** If a concrete type is used as a data source, fields are displayed for both the base type and the current type. For example, in the following screen shots, if DirPerson is the data source, data source fields from both DirPerson and DirPartytable appear.



# Data entity wizard rules

Article • 04/23/2024

This article provides information about the natural key expansion of surrogate foreign key fields and the expansion of child/parent relations.

## Natural key expansion of surrogate foreign keys

A surrogate foreign key field's extended data type must be **RefRecId** or a derivative. The natural key expansion of a surrogate foreign key field uses the rules in the following list. These rules are listed in the order of evaluation.

1. **Replacement key** – The replacement key fields
2. **Primary key** – The primary index key fields
3. **Alternate key** – The first unique alternate key
4. **Auto-identification key** – The auto-identification fields

Surrogate foreign key fields that are nested in the natural key are recursively expanded. Recurring nested surrogates are limited to the first occurrence. If you select the **is mapped** property of a surrogate foreign key (that is, if you set the property to **true**), the related data source is automatically added to the entity, and the **is mapped** property of each field in the related data source's natural key is selected. In addition, any nested surrogate foreign key data sources are recursively added to the entity. If you clear the **is mapped** property of a surrogate foreign key (that is, if you set the property to **false**), the related data source are automatically removed and unmapped from the entity and any nested surrogate foreign key data sources. The effect of selecting and clearing the **is mapped** property of a surrogate foreign key field differs from the effect of using the **Add data source** and **Remove data source** buttons. If you add a surrogate foreign key data source, the **is mapped** property of the parent data source surrogate foreign key field isn't automatically set to **true**. If you are removing a surrogate foreign key data source, the **is mapped** property of the parent data source surrogate foreign key field isn't automatically set to **false**. By default, the **is mapped** property of the root data source's surrogate foreign key field is set to **true**. Therefore, by default, surrogate foreign key relations are expanded to one level.

## Expansion of parent/child relations

# Behavioral properties on data entities

Article • 04/23/2024

Every data entity has properties that let you override the same property values on the tables or views that are the data sources of that entity. Your choices affect the behavior of the entity. In the following table, the first column lists the properties that are discussed in this article. The top row lists the levels where the property is found in the entity designer. The levels are listed in order of increasing granularity: the data source level is more granular than the entity level but less granular than the field level.

[+] Expand table

	Entity level	Data source level	Field level
ReadOnly	Applies	Applies	.
AllowEdit	.	.	Applies
AllowEditOnCreate	.	.	Applies
Mandatory	.	.	Applies

## Entity level

In the designer for your data entity, when you click the name at the root node, the **Properties** pane includes the **Is Read Only** property. The following table describes the behavioral differences between the **Yes** and **No** values of this property.

[+] Expand table

Group	Property name	Display name	Values	Default	Description
Behavior	IsReadOnly	Is Read Only	No, Yes	No	<ul style="list-style-type: none"><li><b>No:</b> Data modification operations (CUD) <i>are</i> allowed, <i>unless</i> an individual data source node in the entity's designer is set to <b>IsReadOnly = Yes</b>.</li><li><b>Yes:</b> Only read operations are allowed, regardless of the <b>IsReadOnly</b> settings on the individual data source nodes in the entity's designer.</li></ul>

# Validations, default values, and unmapped fields

Article • 08/25/2023

This article describes how data entity values are validated, how default values can be provided, and how to use fields that are not mapped to data source values, but instead contain virtual or computed data (unmapped fields).

## Validations

Validations can be defined on the tables that back up entities, at both the field level and the record level. Validations can also be defined at the data entity level.

### Table (data source) vs. entity validation

Entities are backed by tables (data sources), and validations are defined for these tables at both the field level (`Table.validateField()`) and the record level (`Table.validateWrite()`). The validations are respected by data entities that are built by using those tables. Although these validations are intrinsic to the tables that back a data entity, validations can also be defined at the data entity level. Like table-based validations, entity-based validations can be written at the field level (`DataEntity.validateField()`) or the record level (`DataEntity.validateWrite()`).

### Table-based validation behavior

Table validations are fired automatically as a part of the CUD operations.

**Table.ValidateField, AllowEdit, AllowEditOnCreate** Field-level validations are fired automatically when you perform inserts or updates on the data entity. This is true for all paths (X++, OData, and so on). These validations occur during the mapping process, when fields are mapped from an entity to individual data sources.

# Security and data entities

Article • 08/25/2023

## ⓘ Note

Data entities do not support the Extensible Data Security (XDS) concepts.

## Entry points

Data entities support entry point security. This support resembles the support that menu items and pages have. To give you flexibility when you define a security model, data entities allow for a separate security configuration for each integration mode. Currently, two entry points/integration modes are identified for a data entity.

[+] Expand table

Entry point	Description
Data services	The ability to use OData services (API) for the entity.
Data management	The ability to use asynchronous integration options for the entity, such as import/export and connector integration.

## Target scenarios

Data entities can support multiple categories of scenarios. Each category might have to be secured separately.

- **Data management (file-based import/export, and so on)** – Typically, a data manager performs these scenarios. These scenarios might provide access to data that isn't usually accessible through the UI for the client. Therefore, you will often want to secure data management scenarios independently of access to the related page, so that a data manager can perform only import/export operations.
- **General integration via OData** – Many integration scenarios require that data entities be exposed as services, so that data can be accessed via OData (for example, from an online storefront or a Process Lifetime Management [PLM] system). Often, you will want to control access to data entities that are built for this purpose independently of page access. In other words, you will want to grant access to the service interface without granting access through the client UI.

# Create read-only entities that expose financial dimensions

Article • 06/19/2024

In this article, we describe how to build an entity for registered transactions that are registered.

## Note

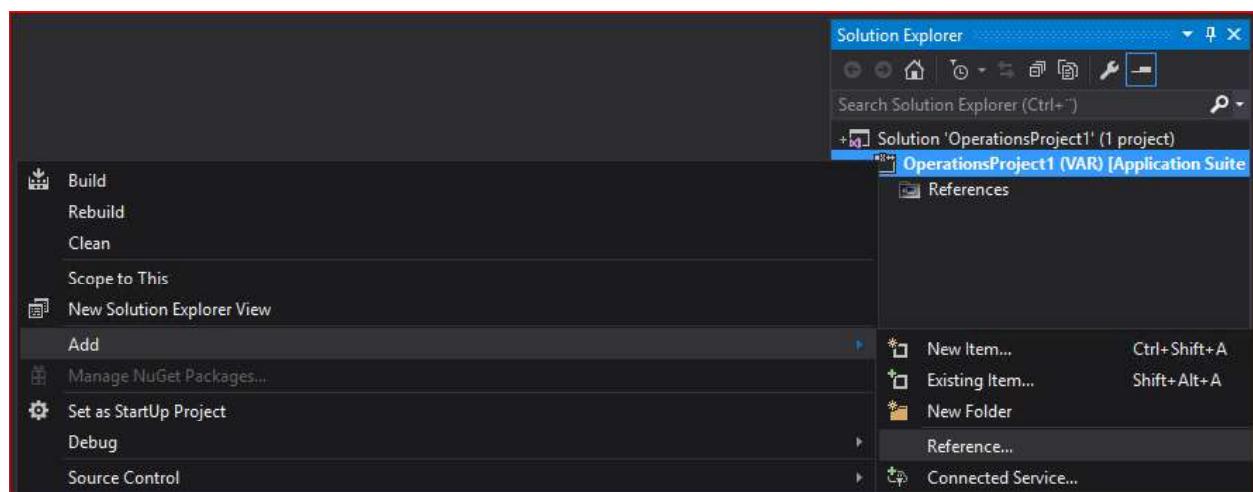
This article comes from Per Baarsoe Jorgensen of the Solutions Architecture team. It describes a real-world scenario that we have encountered as we work with customers.

Imagine a scenario where we must expose all vendor invoice line transactions together with the financial dimensions that were applied through the distributions. Because easy consumption by a third-party tool is essential, we will create an entity for this scenario. As a result, the entity should not have to be joined with other related entities but should be able to provide value on its own.

We will walk through the process of creating a sample entity to meet these requirements. (We will leave out instructions for integrating with Microsoft Azure DevOps, because those steps are already well documented.)

## Create a basic entity

The first step is to create a new element in a project by selecting **New Item**.



In the form that opens, under Data Model, we select the **Data Entity** element type.

# Bring your own database (BYOD)

Article • 04/03/2025

This article explains how administrators can export data entities from the application into their own Microsoft Azure SQL database. This feature is also known as *bring your own database* (BYOD).

The BYOD feature lets administrators configure their own database, and then export one or more data entities that are available in the application into the database. (Currently, more than 1,700 data entities are available.) Specifically, this feature lets you complete these tasks:

- Define one or more SQL databases that you can export entity data into.
- Export either all the records (*full push*) or only the records that have changed or been deleted (*incremental push*).
- Use the rich scheduling capabilities of the 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.

## Entity store or BYOD?

If you followed the series of [blog posts about Microsoft Power BI integration](#), you'll be familiar with Entity store. Entity store is the operational data warehouse. Entity store provides built-in integration of operational reports with Power BI. Ready-made reports and analytical workspaces use Entity store. If you write Power BI reports by using data in your application environment, you should use Entity store.

However, the BYOD feature is recommended for the following scenarios:

- You must export data into your own data warehouse.
- You use analytical tools other than Power BI, and those tools require T-SQL access to data.
- You must perform batch integration with other systems.

### Note

The application 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

# Automated Entity store refresh

Article • 04/23/2024

## Overview

Entity store refresh is automated and managed by the system. Administrators do not need to schedule or monitor the Entity store refresh with the system batch schedules. The refresh operation is based on anticipated latency. This functionality is enabled in Platform update 23. As an administrator you do need to opt-in to use this feature.

## Enable automated refresh

Complete the following steps to enable automated Entity store refresh.

1. Go to **System administration > Set up > Entity store**. On the **Entity store** page, a message indicates that you can switch to the **Automated Entity store refresh** option. This option is managed by the system. An admin does not have to schedule or monitor the Entity store refresh.
2. Select **Switch now**.

### **Important**

This action isn't reversible. After you switch to the **Automated Entity store refresh** option, you can't revert to the old user interface (UI) experience.

3. Select **Yes** to continue.

You will now see the new experience.

# Data task automation

Article • 06/03/2022

Data task automation lets you easily repeat many types of data tasks and validate the outcome of each task. Data task automation is very useful for projects that are in the implementation phase. For example, you can automate the creation and configuration of data projects. You can also configure and trigger the execution of import/export operations, such as the setup of demo data and golden configuration data, and other tasks that are related to data migration. You can also create automated testing of data entities by using task outcome validation.

## Important

Data task automation isn't currently supported for on-premises environments. The user who executes data task automation must be in the same tenant as the application environment and LCS project.

We recommend the following approach for data task automation.

### 1. Identify the data-related tasks that will benefit from automation.

We recommend that implementation teams review their configuration management plan and data migration plan to identify potential data tasks for automation, and also to identify data entity test cases.

### 2. Define tasks.

Tasks are defined in an XML manifest. You can keep your manifest under source control as part of configuration management in your application lifecycle management (ALM) strategy.

### 3. Put the data packages that are related to data task automation in the Shared asset library of Microsoft Dynamics Lifecycle Services (LCS). You can also use a specific LCS project as you require.

Data task automation manager can consume packages from any sandbox and/or production environment that is related to the LCS project.

## Important

- The user account that runs Data task automation manager must have access to LCS and to the LCS project that is referenced in the manifest for data packages.
- Although data task automation can be run in any environment in the cloud, we strongly recommend that you not run any import/export tasks that use integration application programming interfaces (APIs) in a production

# Data validation checklist workspace

Article • 06/05/2024

This article provides an overview of the **Data validation checklist workspace** and the associated configuration.

The **Data validation checklist** workspace lets you track data validation processes across companies, areas, and people. The checklist can be used during a new implementation, after an upgrade, or after a migration. Depending on your view of the **Data validation checklist** workspace, you'll see either all tasks and statuses for a data validation project, or just the tasks that are assigned to you.

You must first select a data validation project at the top of the workspace. All data that is shown in the workspace is then filtered by the selected data validation project.

## Summary tiles

The **Summary tiles** provide an overview of the process, and indicators help you keep the data validation process on track. You can see all remaining tasks, completed tasks, in progress tasks, and not started tasks for the process. This information is for all companies that are included in the selected data validation project.

## Tasks and status section

In the **Tasks and status** section, the status of the overall data validation project is displayed in various ways: status by legal entity, by area, and by task list. You can select the filter to view the status for a specific company. Each status tab provides a breakdown by both the percentage that has been completed and the number of tasks that remain.

The last tab is for the detailed task list. This list shows the full task list. You can filter the task list in several ways. Click **Edit task** to change the status of a task or assign a task. Click **Attachments** to view attachments for a task.

The task name is a hyperlink to the page where the user must go to complete the work. You can set this hyperlink by using the **Menu item name** field when you edit or create a task from the **Configure data validation project** form.

You can attach files, notes, images, and URLs to a task by using the **Attachments** action. For example, you can attach a report file that was printed for a task. An icon appears in the **Attachment** column for the task if an attachment is present.

# Data management error descriptions and known limitations

Article • 01/17/2025

This article documents the scenarios where specific errors are seen. It doesn't provide a complete list of errors and scenarios. However, keep checking back, because the list is continuously updated.

## 1. Import to target failed due to an update conflict as more than one process is trying to update the same record at the same time

When you use recurring imports (enqueue API), the files are sent to the endpoint at a high frequency. If sequential processing of messages isn't enabled, data management tries to process the files in parallel. When files are processed in parallel, if multiple files have the same record, multiple threads try to update the same record at the same time. You must update the data so that the same records aren't repeated across multiple files. The entity is expected to handle cases where each record occurs in only one file. To mitigate the issue, sequentially process the files. If this issue isn't a data issue, and the entity isn't expected to process in parallel, the entity isn't subject to parallel processing. Enable sequential processing of messages in the recurring job.

## 2. There are fields, which aren't mapped to Entity <EntityName>

It's a common practice to use the export functionality to generate the entity template file that can be used later for imports. If you export the template in fixed-width format where the **First row header** option is set to **No** in the source data format setup, the exported template doesn't have the column names. Then, when the file is imported, this error occurs.

## 3. Data package download - Error downloading data package for job ". Record for ID - {GUID} not found

In one scenario, the dev environment points to the database in another environment (for example, the UAT environment), and the export job is run from the source environment (dev). The exported file is uploaded to the blob storage that's associated with the source environment (dev). This job shows up in the target environment (UAT)

# Archive data in Dynamics 365 finance and operations apps with Dataverse

Article • 10/31/2024

This article describes how to archive data in Microsoft Dynamics 365 finance and operations apps. Finance and operations apps support custom retention policies for securely archiving and retaining unlimited data for the long term in a cost-efficient way. Finance and operations apps set no limit on active data and therefore support your business growth. Nevertheless, you might want to consider moving historical, inactive data that's required for compliance and regulatory reasons to Dataverse long term retention.

## ⓘ Note

This feature doesn't limit the total number of records that can be archived. However, the current limitation in the initial synchronization of data replication to Dataverse is limited to a maximum of 500 million records in the largest table. Prior to testing this feature, it's recommended to trim any tables that have more than 500 million records.

## Business application data lifecycle

The business application data lifecycle has three stages:

1. Active data
2. Transitions to historical, inactive data that's required for compliance and regulatory reasons
3. Transitions to deleted data

Finance and operations apps enable organizations to achieve the following benefits through archiving:

- Secure historical, inactive application data for the long term to meet audit, legal, and regulatory requirements.
- Reduce the size of the application database and the capacity that consumed, to potentially improve application performance that's associated with very large tables.

# Set up and manage archive data in finance and operations apps

Article • 06/14/2024

This article describes how to set up and manage archive data in Microsoft Dynamics 365 Finance and operations apps.

## Required privileges

The following privileges are required:

- In Power Platform:
  - The **System Administrator** and **System Customizer** roles in [Power Platform admin center](#).

To confirm that you have these privileges, follow these steps.

1. In the Power Platform Admin Center, go to **Environments**.
2. Select the relevant environment.
3. In the **Access** section, select users.
4. Select **Installing user > Roles**. Confirm the necessary privileges.

- In Microsoft Dynamics 365 Finance and operations apps:

- The **System administrator** role in Microsoft Dynamics 365 Finance and operations apps.

To confirm that you have these privileges, follow these steps.

1. In Dynamics 365 Finance and operations apps, go to **System administration**.
2. Select **Users > Users**.
3. Select **Installing user > Roles**. Confirm the necessary permissions.

- In Microsoft Dynamics Lifecycle Services:

- The **Organization Admin** role to create environments. Additionally, the **Project owner** or **Environment manager** role must be assigned to the user in the **Project security** role field in Lifecycle Services.

## Prepare the environment

# Archive customization (preview)

Article • 03/25/2025

This article describes how the archive feature in Microsoft Dynamics 365 finance and operations apps supports customization. The archival framework supports extensions to include custom table fields and custom tables in supported functional scenarios.

## Add custom fields in history tables and business intelligence entities

Custom fields that are added to a standard table must be added to the corresponding history table and the business intelligence (BI) entity. The customized BI entity must be refreshed in Dataverse to archive data with Dataverse long term retention.

### History tables

Transactions records are moved to the history tables. The schema of a history table must match its corresponding live table. All columns in the live table must be present in its mirrored history table.

**Column exclusion rule:** `SysRowVersion` and `SysDataState` columns are added by the platform and managed by using table metadata properties. These columns don't have to be added to the history tables.

### Business entity

Dataverse interacts with finance and Operations. These virtual entities are used to retrieve data from the finance and operations database and save it to the corresponding tables in the Dataverse long term retention.

 **Important**

Don't add relationships between the entities.

### Step 1: Add fields to the history table via extensions

The archival framework requires that all live table columns are mirrored in the corresponding history tables. Use table extensions to add the custom fields to history tables. For more information about how to add fields to history tables through extension in finance and operations apps, see [Add fields to tables through extension](#).

### Step 2: Add fields to BI entities via extensions

Additional fields that are added to live tables must be added to the corresponding BI entities.

### Step 3: Refresh the virtual entity in Dataverse

The customized business entity must be refreshed in Dataverse to archive data in the Dataverse long-term retention store.

## Add new tables to the archive scenario

Additional tables can be included in the archive scenario if they have a direct or indirect relationship with the main live table.

To create a history table that corresponds to the live table in the archive scope, follow these steps.

1. Create a new history table that mirrors all fields from the corresponding live table, including all metadata properties on the live table.  
See the [column exclusion rule](#) earlier in this article.
2. Don't mirror indexes from the live table in the history table. For most history tables, a clustered index on the `RecId` column is sufficient. Create an additional index to improve query performance as required and to maintain foreign key relationships.
3. Extend the `ArchiveAutomationJobRequestCreator` class for a scenario to add the new table to archive table chart.

## Code example

# View archived data in Dataverse long-term retention

Article • 07/22/2024

This article describes how to view archived data in Microsoft Dataverse long-term retention.

## View data by using Fabric

You can use Fabric to view both the live (active) and archived (inactive long term retained) Dynamics 365 Finance and Operations application data. To use this capability, you must link your Dataverse environment to Fabric. If the setup for the link to Fabric isn't working, follow the steps in [Link to Microsoft Fabric](#).

Confirm that Fabric is enabled. If it isn't enabled, go to

<https://app.fabric.microsoft.com> using at least a Conditional access administrator user account. In Microsoft Fabric, select the settings gear and go to **Admin portal > Tenant settings**. Enable the **Users can create Fabric items** setting.

### ⓘ Note

If you don't have a Fabric subscription, you can use a Fabric trial to access data in a Dataverse-managed Azure data lake. After the trial period expires, you must have the Fabric capacity or Premium per capacity Power BI SKU for users.

## View archived data in the Dataverse-managed data lake

To view the archived data in Dataverse, follow these steps.

1. After the archival job is completed, go to the Power Apps maker portal.
2. Select **Azure synapse > Microsoft one Lake**.
3. Select **View in Microsoft Fabric**.

The data in the Dataverse-managed data lake is available in Dataverse tables that have the "mserp\_" prefix. In Dataverse tables that have this prefix, you can use the `msft\_datastate` column to filter the data through a SQL `WHERE` clause:

- To filter for inactive (archived) application data: `WHERE msft_datastate=1`

# Archive Dynamics 365 Finance General ledger data

Article • 04/10/2024

This article explains how to archive Microsoft Dynamics 365 Finance General ledger data.

## ⓘ Important

- General ledger archiving can occur only for fiscal years that the year-end close process has been run for.
- The archive jobs for different years must be run in chronological order. For example, 2020 General ledger data must be archived before 2021 General ledger data.

## Prerequisites

The following prerequisites must be met before you archive General ledger transactions:

- No periods for a fiscal year and the company can be open.
- Year-end close must be run for the fiscal year for the company.
- General ledger transactions for the previous fiscal year for the company must be archived.

The screenshot shows the 'Archive with Dataverse long term retention' page. The top navigation bar includes 'Finance and Operations', a search bar, and various system icons. The main area is titled 'Archive with Dataverse long term retention'. On the left, there's a sidebar with 'Functional scenarios' and three options: 'General ledger' (selected), 'Inventory transactions', and 'Sales orders'. The main content area has a 'Fiscal calendar' dropdown set to 'Fiscal'. Below it is a table with columns: 'Group by', 'Fiscal year', 'Company', 'Ready to archive', 'Job status', and 'Results'. The table data is as follows:

Group by	Fiscal year	Company	Ready to archive	Job status	Results
2015 (7) [Fiscal year]	2015	demf	Ready		
	2015	gimf	Ready		
	2015	rumf	Ready		
	2015	usmf	Ready		
	2015	uspi	Not ready		
	2015	usr	Ready		
2016 (12) [Fiscal year]	2016	ussi	Ready	Completed	11,242 records moved
	2016	bmf	Not ready		
	2016	demf	Not ready		
2017 (12) [Fiscal year]	2017	frit	Not ready		

## Set up an archival job

To set up an archival job, follow these steps.

# Archive Dynamics 365 Finance customer invoice data

Article • 12/10/2024

This article explains how to archive Microsoft Dynamics 365 Finance customer invoice data.

## Important

Customer invoice data can be archived only for fiscal periods that are on hold or closed.

## Prerequisites

Before you can archive customer invoice transactions, the following prerequisites must be met:

- In Feature management, enable the **(Preview) Archive Customer invoice with long term retention** and **Archive with Dataverse long term retention** features. Learn more about how to set up the archive feature in [Set up and manage archive data in finance and operations apps](#).
- The fiscal period must have a status of **On hold** or **Permanently closed** for the date range that is used in the archive job.

## Set up an archival job

To set up an archival job, follow these steps.

1. Go to **System administration > Workspaces > Archive with Dataverse long term retention**.
2. In the **Archive with Dataverse long term retention** workspace, select **Customer invoice**.
3. Select **New long term retention** to open a wizard that you can use to schedule a new **Customer invoice long term retention** job.
4. Enter a name for the job, and then select **Next**. Each new long term retention job can be scheduled for sales order invoices, free text invoices, or both.

# Archive Dynamics 365 Finance Tax transactions data

Article • 06/08/2024

This article explains how to archive Dynamics 365 Finance Tax transactions data.

When you archive Tax transactions, data from the following tables is moved to history tables:

- TaxTrans
- TaxTrans\_BR
- TaxTrans\_IN
- TaxTrans\_IT
- TaxTrans\_RU
- TaxTrans\_W
- TaxTransExtensionTH
- TaxTransGeneralJournalAccountEntry
- TaxTransSubledgerJournalAccountEntry
- TaxTrans\_Reporting

## Prerequisites

Confirm that your environment is configured to use the archive feature. For more information, see [Set up and manage archive data](#).

In addition, the following prerequisites must be met before you archive Tax transactions:

- All periods in the fiscal year must be either permanently closed or in an on-hold state.
- Tax transactions for the previous fiscal year for the company must be archived.
- The archive jobs for different years must be run in chronological order. For example, 2020 Tax transactions data must be archived before 2021 Tax transactions data.

## Set up an archival job

To set up an archival job, follow these steps.

1. Go to **System administration > Archive with Dataverse long term retention** to open the **Archive with Dataverse long term retention** workspace.

# Archive Dynamics 365 Supply Chain Management Sales orders data

Article • 04/09/2024

This article explains how to archive Dynamics 365 Supply Chain Management Sales orders data.

## Prerequisites

The following prerequisites must be met before sales orders can be archived:

- The sales orders are fully invoiced.
- The sales orders shouldn't be part of an intercompany order chain.

## Install the solution in Microsoft Power Platform

For information about how to install the solution in Microsoft Power Platform, see [Set up and management](#).

## Turn on the feature in Dynamics 365 Supply Chain Management

The Archive with Dataverse long term retention feature should be enabled.

## Set up an archival job

To schedule the long-term retention job, follow these steps.

1. In Dynamics 365 Supply Chain Management, go to **System administration > Archive with Dataverse long term retention workspace**.
2. Select **Sales orders scenario**.
3. Select **New long term retention** to open a wizard that you can use to schedule a new sales order archival job with Dataverse long-term retention.
4. Enter a name for the job, and then select **Next**.

# Archive Dynamics 365 Supply Chain Management Inventory transactions data

Article • 11/15/2024

This article explains how to archive Dynamics 365 Supply Chain Management Inventory transactions.

The *Archive with Dataverse long term retention* feature optimizes storage and system performance by moving `InventTransArchive` records to a Microsoft Azure data lake and replicating corresponding records to the `InventTransArchiveHistory` table. Records in the `InventTransArchive` table represent inventory transactions that have already been consolidated (see also [Consolidate inventory transactions](#)).

## Prerequisites

Before you can use this feature, you must enable it for your system and consolidate the transactions that you want to archive, as described in the following subsections.

## Turn on the features in Supply Chain Management

If your system doesn't already include the features described in this article, go to the [Feature management](#) workspace and turn on the following feature:

- *Archive with Dataverse long term retention* – This feature moves archived inventory transactions from the `InventTransArchive` table to Dataverse long term retention and replicates the data to the `InventTransArchiveHistory` table.

## Considerations before you purge inventory transactions

- The **Reverse** function of the *Inventory transaction consolidation* feature isn't available for purged inventory transaction records.
- The **unpurge** activity isn't available from the Dataverse long term retention to Supply Chain Management.

## Consolidate inventory transactions before you purge

# Archive Dynamics 365 Supply Chain Management inventory-related documents

Article • 09/07/2024

This article explains how to archive Microsoft Dynamics 365 Supply Chain Management inventory-related documents.

For the removed inventory transactions, a scheduled job moves the corresponding inventory transactions originator from the `InventTransOrigin` table to the `InventTransOriginHistory` table.

## Prerequisites

Before you can use this feature, you must enable it for your system.

## Turn on the feature in Dynamics 365 Supply Chain Management

To enable the archive feature in Feature management, follow these steps.

1. In Dynamics 365 finance and operations apps, go to **Feature management**.
2. Select the **Archive with Dataverse long term retention** feature for overall archival service integration. This feature enables all supported finance and operations functional scenarios for archiving through Dataverse long term retention.

The **Archive with Dataverse long term retention** workspace should now be available in the **Workspaces** list in finance and operations apps.

## Set up an archival job

To move inventory journal records to Dataverse long term retention, follow these steps.

1. Go to **Workspaces > Archive with Dataverse long term retention**.
2. Select **Inventory transactions originator**.
3. Select **New long term retention job** to open a wizard that you can use to schedule a new inventory transactions originator archival job with Dataverse long term

# Archive Dynamics 365 Supply Chain Management Inventory journal data

Article • 04/10/2024

This article explains how to archive Dynamics 365 Supply Chain Management Inventory journal data.

## Prerequisites

Before you archive inventory journal data, the following prerequisites must be met:

- The ledger period of the inventory journal that has to be archived must be closed.
- The period must be at least one year before the "from" period date of the archive.

## Turn on the feature in Dynamics 365 Supply Chain Management

1. In finance and operations apps, go to **Feature management**.
2. Select the **Archive with Dataverse long term retention** feature for overall archival service integration. This feature enables all supported finance and operations functional scenarios for archiving through Dataverse long term retention.

The **Archive with Dataverse long term retention** workspace should now be available in the **Workspaces** list in finance and operations apps.

## Set up an archival job

To move inventory journal records to Dataverse long term retention, follow these steps.

1. Go to **Workspaces > Archive with Dataverse long term retention**.
2. Select **Inventory journal**.
3. Select **New long term retention job** to open a wizard where you can schedule a new Inventory Journal archival job with Dataverse long term retention.
4. Enter a name for the job, and then select **Next**.
5. Select the period of the inventory journal that must be purged, and then select **Next**.
6. Enter the start date and time of the job, and then select **Next**.
7. Select **Finish** to schedule the archive job.

# Archive Commerce transactions

Article • 08/02/2024

This article describes how to archive Microsoft Dynamics 365 Commerce transactions.

## Prerequisites

To archive Microsoft Dynamics 365 Commerce transactions, you must first enable the archival framework in Dynamics 365 Commerce headquarters. Learn how to enable the archival framework in [Set up and manage archive data](#).

## Enable the feature using the Feature management workspace

The **Archive with Dataverse long term retention** feature should be enabled.

To enable the feature in Commerce headquarters, go to the **Feature management** workspace (**Systems administration > Workspaces > Feature management**), and turn on the **Archive with Dataverse long term retention** feature flag.

## Set up an archival job

To set up an archival job, follow these steps.

1. In headquarters, go to **System administration > Archive with Dataverse long term retention workspace**.
2. Select **Retail transactions**.
3. Select **New long term retention job** to open a wizard where you can schedule a new retail transaction archival job with Dataverse long-term retention.
4. Enter a name for the job, and then select **Next**.
5. Specify the date of the oldest retail transactions to archive (the **From** date).
6. Specify the date of the newest retail transactions to archive (the **To** date).
7. Select the legal entity (company) to archive the sales orders for.
8. Select **Next**.

# Finance and operations apps archive with Dataverse long term retention FAQ

Article • 02/12/2025

## Does testing archiving with Dataverse long-term retention require a sandbox environment?

Yes, it requires use of a Tier-2 or greater Sandbox instance. It won't work on Tier CHE.

## How long will my archival job take to complete?

Data archival jobs are assigned a lower priority by the application. The duration of an archival job is dependent upon the data volumes. A job can take around 7-14 days based on data volume.

## After data is archived, can I access the live data and the archived data from Dataverse long term retention?

Yes, you can use Microsoft Fabric (with the Power BI option) to query live and archived data. Dataverse makes it easy to set up Fabric.

## Can I view archived data from my finance and operations application?

Yes, an application-specific inquiry page is available so that you can view archived data. After data is permanently purged from history tables, it's no longer available for viewing. You can use Fabric (with the Power BI option) to query data from Dataverse long term retention.

## Can I restore data from Dataverse long term retention back to live tables?