opentext™

OpenText™ ApplicationXtender AppWorks

Integration Guide

This guide describes how to configure the integration between AppWorks and ApplicationXtender.

EAXCORE200400-IAG-EN-01

OpenText™ ApplicationXtender AppWorks Integration Guide

EAXCORE200400-IAG-EN-01
Rev.: 2020-Oct-15

This documentation has been created for software version 20.4.
It is also valid for subsequent software releases unless OpenText has made newer documentation available with the product, on an OpenText website, or by any other means.

Open Text Corporation

275 Frank Tompa Drive, Waterloo, Ontario, Canada, N2L 0A1

Tel: +1-519-888-7111
Toll Free Canada/USA: 1-800-499-6544 International: +800-4996-5440
Fax: +1-519-888-0677
Support: https://support.opentext.com
For more information, visit https://www.opentext.com

Table of Contents

Chapter 1

Introduction

OpenText AppWorks provides a single platform for process automation, case management, and low-code application development, to automate complex business processes, enable better decision making, and improve customer experiences.

AppWorks supports integration with ApplicationXtender, both at the document management and metadata levels. Users can manage ApplicationXtender documents in AppWorks and synchronize entity data as application documents in ApplicationXtender Web Access. This guide describes how to configure the integration between AppWorks and ApplicationXtender.

1.1 Prerequisites

Before you configure the integration between ApplicationXtender and AppWorks, ensure that the following requirements are met:

- ApplicationXtender 20.3 or later is installed and configured with OpenText Directory Services (OTDS)
- AppWorks 20.3 or later is installed and configured with the same OTDS as ApplicationXtender

1.2 Integration features

The AppWorks integration with ApplicationXtender includes the following features:

- ApplicationXtender is integrated with the Document Store connector in AppWorks.
- ApplicationXtender supports single sign-on (SSO) access using OTDS-based oAuth authentication.
- AppWorks administrators can map an entity with an application in ApplicationXtender and the entity property with an ApplicationXtender application field. A single entity in AppWorks can be mapped with an application in ApplicationXtender. Administrators can determine whether a document in ApplicationXtender should be deleted when the corresponding entity item is deleted in AppWorks. Any change to entity data is automatically synchronized to the corresponding ApplicationXtender document.
- Developers can build solutions in AppWorks using entity modelling and deploy these solutions in AppWorks. An entity item with a file represents an application document in ApplicationXtender.
- Application users can create an entity item and upload a document, which creates a corresponding document in ApplicationXtender. Application users can

manage the document and perform common document management operations in AppWorks. The operations that are supported include uploading a new document, downloading a document of a single MIME type or multiple MIME types, adding a new document version and managing document versions, viewing metadata and audit history, previewing a document, and deleting a document.

- When entity item data changes in AppWorks, the application record in ApplicationXtender is updated automatically.
- AppWorks web services for ApplicationXtender integration include a web service to fetch the mapping between the AppWorks entity and ApplicationXtender application, as well as a web service to fetch the User Defined List (UDL) values from an ApplicationXtender field. The latter can be used in a dynamic enumerated list in AppWorks to fill values dynamically from ApplicationXtender.

The ApplicationXtender integration with AppWorks includes the following features:

- AppWorks supports single sign-on (SSO) access using OTDS-based oAuth and SAML 2.0 authentication.
- Any index change in ApplicationXtender is automatically synchronized to its corresponding AppWorks entity.
- When a document is created in ApplicationXtender, an AppWorks entity is created and linked to the ApplicationXtender document. If lifecycle functionality is configured, a lifecycle is triggered in AppWorks.
- When a document is deleted from ApplicationXtender, the document that is linked to the corresponding entity record in AppWorks is unlinked.
- ApplicationXtender Web Access users can manually submit an existing ApplicationXtender document to the AppWorks workflow.
- ApplicationXtender Web Access users can view the AppWorks My Inbox.

Chapter 2

Configuration

This chapter describes the steps for configuring and integrating AppWorks with ApplicationXtender.

Note: This chapter provides high-level steps. For detailed information about developing an application in AppWorks, refer to the latest version of the *AppWorks Platform Low-Code Design Guide*.

2.1 Developing an application using AppWorks entity modelling

AppWorks synchronizes entity data with ApplicationXtender. This means that an entity must be created in AppWorks for every application in ApplicationXtender. To develop an application, ensure that the user who is developing the application is assigned to the Developer role.

1. Log in to AppWorks using the OTDS user name and password.
2. Start developing an application using entity modelling. Refer to the *AppWorks Platform Low-Code Design Guide* for more information about developing an application using the AppWorks Platform.
3. Create one or more entities and add the required properties to the entity. Make sure that the application has entities and properties corresponding to the application in ApplicationXtender. Make sure that the data types of properties match the data types of fields in the application. For more information about data types and formats, see "Mapping data types" on page 8. For more information about defining ApplicationXtender flags, see "Mapping flags" on page 9.

   List, Layout, and Create form are added by default.
4. Add a file building block and keep the default configurations. It is mandatory to have a file building block for all entities that are synchronized with applications in ApplicationXtender.
5. Add a form. In the **Name** drop-down menu, select **Default**. In the **Type** drop-down menu, select **Cancellable**.
6. Add a web service building block and generate standard web services.
7. Generate a Find web service with parameters to uniquely identify an item. ApplicationXtender uses entity web services for integration between ApplicationXtender and AppWorks. To add a Find web service, in Advanced configuration, add a new web service called FindDocument. In FindDocument operations, add parameter `DocumentURL`, and set the filter condition to

```
File.StorageTicket contains Parameter DocumentURL
```

.

8. Add other required building blocks. If a workflow is required, add the life cycle building block and model the workflow states and actions. To define unique constraints or any business rule, add the rule building block.

9. When you are finished developing the application, publish it to an organization for configuration and validation. In your workspace, right-click the project and click **Publish to Organization**.

2.1.1 Mapping data types

AppWorks integration with ApplicationXtender allows users to store entity instances as ApplicationXtender application records along with documents. The following table describes how application data types in ApplicationXtender map to entity data types in AppWorks.

ApplicationXtender application data type	AppWorks entity data type
Text	Text
Time (hh:mm:ss)	Text
SSN	Text
Telephone	Text
ZipCode	Text
Custom Type	Text
Integer	Integer
Number/Decimal	Decimal
Date	Date
TimeStamp	Date and Time [1]
Currency	Currency [2]
Boolean Choice	Boolean
User Defined List	Enumerated Text **Note:** Select **Static** to define static values, or **Dynamic** to use the user-defined list from ApplicationXtender.

[1] The Date and Time value is automatically generated in ApplicationXtender.

[2] In AppWorks, the user must choose the currency. The user should also create the Currency format in ApplicationXtender using the same currency symbol that is used in AppWorks. For more information, see “Known limitations“ on page 17.

Note: For more information about configuring data types, refer to the latest version of the *AppWorks Platform Low-Code Design Guide*.

2.1.2 Mapping flags

ApplicationXtender application fields use various flags. The following table describes how the flags in ApplicationXtender map to the ApplicationXtender flags in AppWorks.

ApplicationXtender Flags	Mapping the flags to AppWorks entity
Required	When designing a form in AppWorks, select the corresponding entity property and, in the right pane, mark it as Required. During runtime, an asterisk (*) is displayed for the property.
Read-Only (the user can enter data only once)	Select the corresponding entity property and, in the right pane, navigate to **Advanced configuration** > **Allow participants to change the value**, and select the **Only when first created** check box.
Part of Unique Key	In AppWorks, define rule of type uniqueness.
Validation Mask	Validation Mask is not supported in AppWorks.
Dual Data Entry	See "Configuring a dual property" on page 9.

2.1.2.1 Configuring a dual property

The steps below show an example of how you can configure a dual data entry for a property.

1. Create an entity. For more information about creating an entity, see the *AppWorks Platform Low-Code Design Guide*.
2. Create a property. For example, INSURANCE_AMOUNT.
3. Create another property with the same characteristics as the above property. For example, INSURANCE_AMOUNT_DUPLICATE.
4. Add a rule building block. The Add rule dialog box opens.
5. In the **Type** drop-down menu, select **Event** and provide a name for the rule. For example, `DualEntryCheck`.
6. In the **Event** drop-down menu, select **A property changes**.
7. In the **Condition** field, click **Switch to advanced editor** and type the condition as follows:

```
(item.Properties.INSURANCE_AMOUNT != null OR
item.Properties.INSURANCE_AMOUNT_DUPLICATE != null) AND
item.Properties.INSURANCE_AMOUNT !
= item.Properties.INSURANCE_AMOUNT_DUPLICATE
```

8. In the **Action** drop-down menu, select **Show error**.

   In the text area, enter the message which must be displayed to the user if the above condition fails.

9. Save and publish the project.

   If the user provides different values to INSURANCE_AMOUNT and INSURANCE_AMOUNT_DUPLICATE, the message which you configured in Step 8 will be displayed.

2.2 Configuring document store service with ApplicationXtender

1. In the AppWorks Platform, in the created organization, open the **System Resource Manager**.
2. Right-click the ApplicationXtender document store connector and click **Properties**.
3. Open the **Document Store** tab.
4. In the **Store type** list, select **ApplicationXtender**.
5. In the **Brava server URL** field, enter the URL in the format http://<machine name>:<port>/.
6. In the **Brava capability set** field, enter the capability set. The default value is PS_BV, which is the Brava capability set that is bundled with AppWorks.
7. In the **OTDS details** section, in the **Client ID** field, enter `AXClient`. In the **Client secret** field, enter your client secret key.
8. In the **Rest endpoint URL** field, enter the ApplicationXtender REST URL. For example, http://<your.ApplicationXtender.server> or http://<machine name>:<port>/.
9. In the **Datasource name** field, enter your data source name.
10. Click **Save**.

2.3 Creating an Application Server Connector

To work with the entity web services, create an Application Server Connector for the organization. Ensure that entity web services are attached to the Application Server Connector.

For more information about creating service containers, refer to the latest version of the *AppWorks Platform Administration Guide*.

2.4 Mapping an AppWorks entity to ApplicationXtender application

You can use AppWorks to configure entity properties and ApplicationXtender application fields. The mapping is displayed only when the entity has file building blocks and the document store connector is configured with ApplicationXtender.

The administrator can select the application and map the entity properties with the ApplicationXtender application field. Hovering the mouse over the property or field displays the data types of those properties or fields. The administrator needs to consider the following while mapping:

- Mandatory field values must be mapped.
- The Time Stamp field is automatically generated in ApplicationXtender and does not need to be mapped.

For more information about configuring entity properties in AppWorks, refer to the latest version of the *AppWorks Platform Low-Code Design Guide*.

2.5 Adding an OAuth filter in AppWorks

To add an OAuth filter in AppWorks, contact your account manager.

2.6 Enabling Event Dispatch Broker (EDB) in ApplicationXtender Administrator

1. In ApplicationXtender Administrator, navigate to **Server Management** > **Event Dispatch Broker**.
2. In the **Properties** tab, set **Enabled** to **true**.
3. In the **Event Dispatch Broker URL** field, enter the Event Dispatch Broker URL. For example, http://<machineName>/Broker (Or http://<IP address>/Broker).
4. Click **Save**.

2.7 Enabling Workflow Integration Module (WIM) in ApplicationXtender Administrator

1. In ApplicationXtender Administrator, navigate to **Server Management** > **Workflow Integration Module**.
2. In the **Properties** tab, set **Enabled** to **true**.
3. In the **WIM Host** field, enter the host name. For example, <machineName> (IP address or localhost).
4. Click **Save**.

2.8 Configuring AppWorks in ApplicationXtender Administrator

1. In ApplicationXtender Administrator, navigate to **Server Management** > **AppWorks Server**.
2. In the **AppWorks Platform Server URL** field, enter the server URL. For example, http://<machine name>:<port>/home/system.
3. Enter the AppWorks OTDS resource ID.
4. Create a user in OTDS and assign administrator and sysadmin permissions to that user in AppWorks. For example, `awpuser`.
5. In the **User Name** field, enter the user name that you created in Step 4.
6. In the **Password** and **Confirm password** fields, enter the user password.
7. Click **Save**.

2.9 Viewing the AppWorks My Inbox in ApplicationXtender

ApplicationXtender Web Access users can view the AppWorks My Inbox.

In ApplicationXtender Web Access, click your login name on the title bar and then click **User Settings**. On the **Data Source** tab, in the **Workflow** section, enable **Display AppWorks Task Inbox**.

The Display Workflow Inbox Tasks button displays on the ApplicationXtender Web Access site header. To redirect inbox tasks to the AppWorks site, click **Display Workflow Inbox Tasks** > **Go To AppWorks**.

2.10 Downloading PDF documents

AppWorks uses the export job API with the *usePDFFormat* parameter set to *false* (*usePDFFormat=false*). Documents with one ApplicationXtender page are downloaded as is, while documents with multiple PDF pages can be downloaded as a single PDF file.

To enable downloading multiple files as a single PDF file, do the following:

1. Select the **Use Multipage Files** check box.
2. Clear the **Export in Archived Format** check box.
3. Restart the ApplicationXtender server.

If these options are not set, ApplicationXtender PDF documents with multiple pages will be downloaded as ZIP files and each page will be a PDF document inside that ZIP file.

Note: All other ApplicationXtender documents with more than one page are downloaded as a ZIP file that contains each page as a file inside it.

2.11 Deploying the application

Once the application is designed and working as expected, it can be packaged and deployed to another system like QA or a production server.

For more information about packaging and deploying an application, refer to the latest version of the *AppWorks Platform Administration Guide*.

Chapter 3

Creating entities and documents

3.1 Creating entities in AppWorks

To create an entity, log in to the application in AppWorks and click  (New). On the **Create form** page, provide the values and click **Upload**.

An entity instance is created in AppWorks and a corresponding document is created in ApplicationXtender.

Notes

- While creating an entity instance, the user must fill all mandatory fields before uploading a document. If any of the mandatory fields are not filled, uploading the document into ApplicationXtender will fail.
- In the AppWorks Platform, the page that allows you to configure the mapping between entity properties and ApplicationXtender properties is shown only if the file building block is added during the entity creation stage.

3.2 Creating documents in ApplicationXtender

1. In ApplicationXtender Web Access, in the Event Dispatch Broker enabled application, click **New document**.
2. Add a page and fill in the index values.
3. Click **Save**.

   The document is saved in ApplicationXtender and synchronized to AppWorks.

When you delete a document in ApplicationXtender Web Access, the corresponding entity record in AppWorks is unlinked.

ApplicationXtender Web Access allows you to manually submit an existing ApplicationXtender document to the AppWorks workflow. In the query results, select a document. In the toolbar, click **Workflow** > **AppWorks Workflow** to submit the document.

Chapter 4

Known limitations

This chapter describes the known limitations that users might encounter with the ApplicationXtender AppWorks integration.

Issue/limitation	Description
Entity instance deletion is not successful	When a document is checked out in the ApplicationXtender repository, it is expected that the corresponding entity could not be deleted in AppWorks. However, if the document is checked out by another user, the error is not shown to the user in AppWorks, and entity deletion is successful. The entity is deleted in AppWorks, but the document is not deleted in ApplicationXtender.
Updating is not allowed for read-only properties of ApplicationXtender documents	Read-only properties of ApplicationXtender document are allowed to be filled while creating an entity instance. After the entity instance is created, read-only properties cannot be updated again.
Do not map the timestamp ApplicationXtender properties with entity properties	These properties are automatically generated in ApplicationXtender.
Property changes in ApplicationXtender are not reflected in AppWorks	If any property attributes are changed in ApplicationXtender after mapping is done in the AppWorks Platform, the new attributes that are added are not reflected in AppWorks. You must remap the properties in AppWorks.
AppWorks Platform shows error message "The service 'getApplications' is not supported by the XDS implementation of the Document Store"	**Workaround:** Refresh the browser and clear document store rest cache. Access the following two URLs to clear the cache: • http://<server-name>:<port-number>/home/<organization-name>/app/documentservices/rest/cache/reset • http://<server-name>:<port-number>/home/<organization-name>/app/documentservices/rest/cache/flushOrgEntry

Issue/limitation	Description
Failure to check in a document	After the user uploads a document in AppWorks and performs the check out action in ApplicationXtender, the check in action cannot be performed in AppWorks. An error message appears, indicating that the document is checked out in ApplicationXtender by the same user. Any document that is checked out in ApplicationXtender must be checked in to ApplicationXtender as well.
Failure to display user name for a checked out document	The Properties screen in AppWorks will not display the user name in CheckedOutBy if the document is checked out by another user in ApplicationXtender.
Incorrect document path is displayed after checking out and checking in the document in ApplicationXtender	After uploading a document in AppWorks and performing check out and check in actions in ApplicationXtender, the document path or location still points to the initial uploaded document.
Limitation displaying images in the Preview panel (with and without Opentext Brava! viewer)	AppWorks uses render API to render the contents of a single page in JPEG format. The pages that are not converted to images by ApplicationXtender (for example, Microsoft office files) cannot be viewed.
Issues with viewing PDF documents in Brava! viewer	AppWorks uses the export job API with the *usePDFFormat* parameter set to *true* (*usePDFFormat=true*). With this API, ApplicationXtender converts the document into a PDF and AppWorks displays it in Brava! viewer. Documents which are not properly converted to PDF cannot be viewed using the Open in viewer option.
Currency Types	Users must select the same currency format in ApplicationXtender and AppWorks.
Failure to synchronize index values	When the user opens an existing document in ApplicationXtender and updates the index values, the index values are not synchronized to AppWorks. The user must click Close Document or return to the query results panel for the updated index values to be synchronized to AppWorks.