

## OpenText™ AppEnhancer AppWorks

### **Integration Guide**

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

EAXCORE240200-IAG-EN-1

---

## **OpenText™ AppEnhancer AppWorks Integration Guide**

EAXCORE240200-IAG-EN-1

Rev.: 2024-May-28

**This documentation has been created for OpenText™ AppEnhancer AppWorks CE 24.2.**

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>

### **© 2024 Open Text**

Patents may cover this product, see <https://www.opentext.com/patents>.

### **Disclaimer**

No Warranties and Limitation of Liability

Every effort has been made to ensure the accuracy of the features and techniques presented in this publication. However, Open Text Corporation and its affiliates accept no responsibility and offer no warranty whether expressed or implied, for the accuracy of this publication.

---

# Table of Contents

<b>1</b>	<b>Introduction .....</b>	<b>5</b>
1.1	Prerequisites .....	5
1.2	Integration features .....	5
<b>2</b>	<b>Configuration .....</b>	<b>9</b>
2.1	Developing an application using AppWorks entity modelling .....	9
2.1.1	Mapping data types .....	10
2.1.2	Mapping flags .....	11
2.1.2.1	Configuring a dual property .....	11
2.2	Configuring document store service with AppEnhancer .....	12
2.3	Creating an Application Server Connector .....	13
2.4	Mapping an AppWorks entity to AppEnhancer application .....	13
2.5	Adding an OAuth filter in AppWorks .....	14
2.6	Enabling Event Dispatch Broker (EDB) in AppEnhancer Administrator .....	14
2.7	Enabling Workflow Integration Module (WIM) in AppEnhancer Administrator .....	14
2.8	Configuring AppWorks in AppEnhancer Administrator .....	14
2.9	Viewing the AppWorks My Inbox in AppEnhancer .....	15
2.10	Downloading PDF documents .....	15
2.11	Deploying the application .....	15
<b>3</b>	<b>Creating entities and documents .....</b>	<b>17</b>
3.1	Creating entities in AppWorks .....	17
3.2	Creating documents in AppEnhancer .....	17
<b>4</b>	<b>Known limitations .....</b>	<b>19</b>



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

### 1.1 Prerequisites

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

- AppEnhancer 24.2 or later is installed and configured with OpenText Directory Services (OTDS).
- AppWorks 24.2 or later is installed and configured with the same OTDS as AppEnhancer.

### 1.2 Integration features

The AppWorks integration with AppEnhancer includes the following features:

- AppEnhancer is integrated with the Document Store connector in AppWorks.
- AppEnhancer supports single sign-on (SSO) access using OTDS-based oAuth authentication.
- AppWorks administrators can map an entity with an application in AppEnhancer and the entity property with an AppEnhancer application field. Single or Multiple entities in AppWorks can be mapped to an application in AppEnhancer. Administrators can determine whether a document in AppEnhancer should be deleted when the corresponding entity item is deleted in AppWorks. Any change to entity data is automatically synchronized to the corresponding AppEnhancer 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 AppEnhancer.
- Application users can create an entity item and upload a document, which creates a corresponding document in AppEnhancer. 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 AppEnhancer is updated automatically.
- AppWorks web services for AppEnhancer integration include a web service to fetch the mapping between the AppWorks entity and AppEnhancer application, as well as a web service to fetch the User Defined List (UDL) values from an AppEnhancer field. The latter can be used in a dynamic enumerated list in AppWorks to fill values dynamically from AppEnhancer.
- AppEnhancer viewer is the native viewer for documents stored in AppEnhancer and it becomes the default viewer in AppWorks when configured with AppEnhancer. AppWorks users can use the features in AppEnhancer viewer based on the user permissions configured in AppEnhancer.
- AppEnhancer viewer can be configured for read-only mode in AppWorks administration.
- AppWorks administrators can configure parameters that can be static parameters or entity properties. These configured parameters will be sent to the AppEnhancer viewer when the user views the document using the **Preview** panel or in the full viewer by clicking **Open in viewer**.
- AppWorks entity item data is synchronized with the corresponding AppEnhancer application record based on the defined mapping. By default, the synchronization is automatic and any changes to the entity data is automatically synchronized with the AppEnhancer application record. If preferred, administrators can configure synchronization to be manual in AppWorks administration. Subsequently, application users can use the **Synchronize properties** option to synchronize entity data with the application record.

The AppEnhancer integration with AppWorks includes the following features:

- AppWorks supports single sign-on (SSO) access using OTDS.
- Any index change in AppEnhancer is automatically synchronized to its corresponding AppWorks entity.
- When a document is created in AppEnhancer, the AppWorks entity instance is created and linked to the AppEnhancer document automatically if the entity has the keyword default in its name, otherwise the document needs to be manually submitted to the workflow. If lifecycle functionality is configured, a lifecycle is triggered in AppWorks.
- When a document is deleted from AppEnhancer, the document that is linked to the corresponding entity record in AppWorks is unlinked.
- AppEnhancer Web Access users can manually submit an existing AppEnhancer document to the AppWorks workflow.

- AppEnhancer Web Access users can view the AppWorks My Inbox.



## Chapter 2

# Configuration

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



**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 AppEnhancer. This means that an entity must be created in AppWorks for every application in AppEnhancer. 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 AppEnhancer. 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 10](#). For more information about defining AppEnhancer flags, see [“Mapping flags” on page 11](#).  
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 AppEnhancer.
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. AppEnhancer uses entity web services for integration between AppEnhancer 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 AppEnhancer allows users to store entity instances as AppEnhancer application records along with documents. The following table describes how application data types in AppEnhancer map to entity data types in AppWorks.

AppEnhancer 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   <b>Note:</b> Select <b>Static</b> to define static values, or <b>Dynamic</b> to use the user-defined list from AppEnhancer.

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

[2] In AppWorks, the user must choose the currency. The user should also create the Currency format in AppEnhancer using the same currency symbol that is used in AppWorks. For more information, see [“Known limitations” on page 19](#).



**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

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

AppEnhancer 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 <b>Advanced configuration &gt; Allow participants to change the value</b> , and select the <b>Only when first created</b> 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 <a href="#">“Configuring a dual property” on page 11</a> .

### 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 AppEnhancer

1. In the AppWorks Platform, in the created organization, open the **System Resource Manager**.
2. Right-click the AppEnhancer document store connector and click **Properties**.
3. Open the **Document Store** tab.
4. In the **Store type** list, select **OpenText AppEnhancer**.  
The viewer type automatically updates to **AppEnhancer viewer**.
5. In the **AppEnhancer URL** box, enter `http://<appenhancerServer>/AppEnhancer` where `<appenhancerServer>` is the address of the AppEnhancer server.
6. In the **OTDS details** area, in the **Client ID** box enter the client ID.
7. In the **Client secret** box, enter the client secret key.
8. In the **Rest endpoint URL** box, , enter `http://<AppEnhancerServer>/AppEnhancerRest` where `<appenhancerServer>` is the address of the AppEnhancer server.
9. In the **Datasource name** box, 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 AppEnhancer application

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

The administrator can select the application and map the entity properties with the AppEnhancer 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 AppEnhancer and does not need to be mapped.

AppWorks administrators can configure parameters that can be static parameters or entity properties. These configured parameters are sent to AppEnhancer viewer when a user views a document using the preview panel or in the full viewer when **Open in viewer** is selected.

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

For more information about mapping data types, see [“Mapping data types” on page 10](#).

For more information about mapping flags, see [“Mapping flags” on page 11](#).

## 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 AppEnhancer Administrator

1. In AppEnhancer 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 AppEnhancer Administrator

1. In AppEnhancer 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 AppEnhancer Administrator

1. In AppEnhancer 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 AppEnhancer

AppEnhancer Web Access users can view the AppWorks My Inbox.

In AppEnhancer 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 AppEnhancer 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 AppEnhancer 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 AppEnhancer server.

If these options are not set, AppEnhancer 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 AppEnhancer 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 AppEnhancer.

#### **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 AppEnhancer will fail.
- In the AppWorks Platform, the page that allows you to configure the mapping between entity properties and AppEnhancer properties is shown only if the file building block is added during the entity creation stage.

### 3.2 Creating documents in AppEnhancer

To create a new document:

1. In AppEnhancer 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 AppEnhancer.

Documents are automatically synchronized to AppWorks if the entity name contains the keyword `default`. If the entity name does not bear the keyword `default`, then the document must be manually submitted.

To submit the document to AppWorks, in the query results, select a document and then go to **Workflow** > `<entityName>`.



## Chapter 4

# Known limitations

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

Issue/limitation	Description
Entity instance deletion is not successful	When a document is checked out in the AppEnhancer 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 AppEnhancer.
Updating is not allowed for read-only properties of AppEnhancer documents	Read-only properties of AppEnhancer 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 AppEnhancer properties with entity properties	These properties are automatically generated in AppEnhancer.
Property changes in AppEnhancer are not reflected in AppWorks	If any property attributes are changed in AppEnhancer 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"	<b>Workaround:</b> Refresh the browser and clear document store rest cache. Access the following two URLs to clear the cache: <ul style="list-style-type: none"><li>• <code>http://&lt;server-name&gt;:&lt;port-number&gt;/home/&lt;organization-name&gt;/app/documentservices/rest/cache/reset</code></li><li>• <code>http://&lt;server-name&gt;:&lt;port-number&gt;/home/&lt;organization-name&gt;/app/documentservices/rest/cache/flushOrgEntry</code></li></ul>

<b>Issue/limitation</b>	<b>Description</b>
Failure to check in a document	After the user uploads a document in AppWorks and performs the check out action in AppEnhancer, the check in action cannot be performed in AppWorks. An error message appears, indicating that the document is checked out in AppEnhancer by the same user. Any document that is checked out in AppEnhancer must be checked in to AppEnhancer 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 AppEnhancer.
Incorrect document path is displayed after checking out and checking in the document in AppEnhancer	After uploading a document in AppWorks and performing check out and check in actions in AppEnhancer, the document path or location still points to the initial uploaded document.
Currency Types	Users must select the same currency format in AppEnhancer and AppWorks.
Failure to synchronize index values	When the user opens an existing document in AppEnhancer 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.