Tungsten AP Agility

Subclasses

Best Practices Guide

Version: 2025.1

Date: 2024-12-02

Table of Contents

Preface

This guide describes the recommended best practices that you should follow while using Tungsten AP Agility and working with subclasses to improve extraction performance.

Related documentation

The full product document set for Tungsten AP Agility is available here:

https://docshield.tungstenautomation.com/Portal/Products/APAgility/2025.1-j1u9a8a63z/APAgility.htm

The product document set for Tungsten TotalAgility is available for the following supported versions:

- 7.9.0
- 7.10.0
- 7.11.0
- 8.0.0
- 8.1.0

Select the documentation link for the Tungsten TotalAgility version that you are currently using.

If you select the incorrect link, it is possible that some features documented are not in your installed version.

In addition to this guide, the AP Agility document set includes the following items:

Tungsten AP Agility Release Notes

This guide contains late-breaking product information that may not be included in other Tungsten AP Agility documentation. Release notes are also available for each of your installed Tungsten Automation applications.

Tungsten AP Agility Installation Guide

This guide provides instructions for installing Tungsten AP Agility and integrating it with Tungsten TotalAgility.

For more information about installing Tungsten TotalAgility refer to *Tungsten TotalAgility Installation Guide*

Tungsten AP Agility On-Premise Multi-Tenant Installation Guide

This guide provides information about installing the on-premise components you may needed to get Tungsten AP Agility to work in a multi-tenant TotalAgility environment or a Tungsten Automation hosted TotalAgility Azure environment.

Tungsten AP Agility Configuration Help

This help provides detailed information on how to configure Tungsten AP Agility for your environment.

Tungsten AP Agility PIX Correction activity Help

This help provides detailed information about how to navigate through the PIX Correction activity. It also includes details about how to use the PIX Correction activity when processing documents with Tungsten AP Agility Invoice Processing.

Tungsten AP Agility Scan activity Help

This help provides information on how to scan documents for Tungsten AP Agility Invoice Processing.

Tungsten AP Agility Line Pairing Correction activity

This help provides use cases and examples for correcting a document that fails line pairing during Invoice Processing.

Tungsten AP Agility Validation activity Help

This help provides information about validating a document in Tungsten AP Agility Invoice Processing.

Additional documentation about validating documents is available in the *Tungsten TotalAgility Validation activity Help*.

Tungsten AP Agility Error Handling Help

This helps provides information on how to handle documents that end up in an error state during Invoice Processing.

Tungsten AP Agility AP Workflow Help

This help provides information about how to use the various activities that are part ofs the AP Agility AP Workflow. This includes general information that is relevant for all activities as well as specific information related to the Approval activity, the Coding activity, the Exception activity, the Line Pairing Exception activity, the Hold activity, and the Comment Request activity.

Analytics for AP Agility Installation Guide

This guide provides information about installing Analytics for AP Agility and integrating it with Tungsten AP Agility.

Analytics for AP Agility Help

This help provides detailed information on how to generate and interpret the various reports that are available in Analytics for AP Agility .

Training

Tungsten Automation offers both on-demand and instructor-led training to help you make the most of your product. To learn more about training courses and schedules, visit the Tungsten Automation Learning Cloud.

Getting help with Tungsten Automation products

The Tungsten Automation Knowledge Portal repository contains articles that are updated on a regular basis to keep you informed about Tungsten Automation products. We encourage you to use the Knowledge Portal to obtain answers to your product questions.

To access the Tungsten Automation Knowledge Portal, go to https://knowledge.tungstenautomation.com/.

The Tungsten Automation Knowledge Portal is optimized for use with Google Chrome, Mozilla Firefox, or Microsoft Edge.

The Tungsten Automation Knowledge Portal provides:

- Powerful search capabilities to help you quickly locate the information you need.

  Type your search terms or phrase into the **Search** box, and then click the search icon.
- Product information, configuration details and documentation, including release news.

  To locate articles, go to the Knowledge Portal home page and select the applicable Solution Family for your product, or click the View All Products button.

From the Knowledge Portal home page, you can:

- Access the Tungsten Automation Community (for all customers).

  On the Resources menu, click the **Community** link.
- Access the Tungsten Automation Customer Portal (for eligible customers).

  Go to the Support Portal Information page and click **Log in to the Customer Portal**.
- Access the Tungsten Automation Partner Portal (for eligible partners).

  Go to the Support Portal Information page and click **Log in to the Partner Portal**.
- Access Tungsten Automation support commitments, lifecycle policies, electronic fulfillment details, and self-service tools.

  Go to the Support Details page and select the appropriate article.

Chapter 1

Overview

Tungsten AP Agility is designed to work out-of-the box for a majority of vendors.

The Invoices class provides generic extraction logic that works well for many documents. However, if the generic extraction provides poor results for a high frequency Vendor it may make sense to create a subclass for that vendor and use custom locators for individual fields.

You can also use subclasses for invoices that tend to require special extraction logic. For example, utility bills may need custom logic to extract any required information that is not typically found on other invoices.

⚠ It is important that you do not delete or rename any of the existing classes, field groups, fields, or locators. This ensures future project upgrades.

For more information on what can and cannot be modified in a project, see the *Tungsten AP Agility Installation Guide*.

Chapter 2

Subclasses

By default, all documents in Tungsten AP Agility are classified to the Invoices class during Invoice Processing. The Invoices class provides generic extraction logic that works well for many documents. The addition of online learning improves extraction results over time, but online learning cannot account for extreme layout or recognition issues.

This means that if the generic extraction consistently provides poor extraction results for a high frequency vendor and online learning does not help with the extraction results, the best practice is to create a subclass for that vendor and use custom extraction logic for any fields that are causing problems

For example, the invoices from one of your vendors has a black box where the Invoice Number in printed in white text. Recognition frequently fails to read the Invoice Number and these invoices regularly end up in the Validation activity for Invoice Number validation. In this case, create a subclass for this vendor and then implement one of the following possible solutions to improve the recognition results:

- Perform special image processing with a Script Locator
- Use a zonal approach and an Advanced Zone Locator

Subclasses and classification

During production, an invoice is classified to the subclass for extraction, and then reclassified back to the Invoices class before it is validated. This is because the Validation activity form that is loaded by Tungsten TotalAgility depends on the class of the document. Since subclasses are added as needed, it is impossible to predict or maintain a separate validation form for each class. This means that a single class (Invoices) and validation form are used.

Create a subclass for a vendor

For each vendor that is causing issues, it is possible to create a subclass under the Invoices class in the InvoiceProcessingAgility project. Once the class is added, you can go on to customize it as required.

More information about working with Transformation Designer project is found in the *TotalAgility Transformation Designer Help*.

You can add a vendor subclass by following these steps:

1. Open the Transformation Designer.

2. Click **Open Project**.
   The **Open Project** window is displayed.
3. From the **Select category** list, select **InvoiceProcessingAgility**.
   The **Select project** list is updated.
4. From the **Select project** list, expand **Shared**, select **InvoiceProcessingAgility** and click **OK**.
   The shared project is displayed.
5. In the **Project Tree**, click on the Show/Hide icon for the **Invoices** class to display its settings.
   The Show/Hide icon is updated and the class settings for the **Invoices** class are displayed.
6. Under **Invoices** right-click on **Classes** and select **Add Class** .
   NewClass1 is added and its name is immediately editable.
7. Enter a name for the new class. For the best results, use the name of the Vendor.
   This new class is a child of the **Invoices** class. As a result, all fields and locators are inherited by your new class. The **Details** for the class are also inherited from the parent class.
   The new class if added and ready for customization.
8. Repeat these steps for any other vendors that require special attention or customization.

What next?

- Train a subclass for classification
- Reclassify the subclass via script
- Add custom extraction logic for a subclass

Train a subclass for classification

In order to ensure that a vendor subclass is correctly classified it is necessary to add several training documents for that vendor and then train the project. These training documents should be typical examples of invoices for that vendor and have a consistent layout.

You can train one or more subclasses by following these steps:

1. Open your **InvoiceProcessingAgility** project in the Transformation Designer.
2. Select your vendor subclass in the **Project Tree**.
3. In the **Documents** pane, open a **Document Set** that contains example invoices for your vendor.
   A list of documents is displayed in the **Documents** pane.
4. Double-click on a document to open that document in the **Document Viewer**.
5. Select a document from the vendor, right-click on the document, and select **Add to Training Set of Selected Class (Classification)**.
   Repeat this so that there are at least four training documents assigned to your class.
   Ensure that you keep a few untrained documents to test the classification training.
6. On the ribbon, select the **Process** tab.
   Several settings related to processing a project are displayed.
7. In the **Train** group, select **Classification**.
   Your project is trained.

8. In the **Documents** pane, select one of the untrained invoices from the vendor, right-click, and select **Classify**.

   If your training documents are adequate, the **Classification Result** matches your vendor class.

   If the **Invoices** class is assigned, repeat the above steps until the **Classification Result** returns the correct classification result.
9. Optionally, repeat these steps and train additional vendor classes.
10. Once all new classes are trained, on the **Project** tab, in the **File** group, select **Save Project**.
11. If you are finished modifying the project, click **Release Project**.

Related topics:

- Reclassify the subclass via script
- Add custom extraction logic for a subclass

Reclassify the subclass via script

In order to ensure that your vendor subclasses are reclassified back to the Invoices class after extraction is performed, but before validation, you can customize the subclass script to reclassify your subclass back to the Invoices class by following these steps:

1. Open your **InvoiceProcessingAgility** project in the Transformation Designer.
2. Select your vendor subclass in the **Project Tree**.
3. Right-click on the vendor subclass and select **Show Script**.

   The **Script Code** window is displayed for the selected vendor subclass.
4. Implement the `DocumentAfterExtract` method to reclassify your subclass back to the **Invoices** class by adding the following content to your script:

   ```
   Private Sub Document_AfterExtract(ByVal pXDoc As CASCADELib.CscXDocument)
      ICAL.ReclassifyToInvoices(pXDoc)
   End Sub
   ```

   If an invoice is classified to a subclass for custom extraction logic, the preserve flag is set on the extraction results so a second round of extraction is not needed when the invoice is reclassified back to the **Invoices** class.
5. Repeat these steps for all other subclasses in your project.

The subclass classification result is recorded in a document text extension called `XValue.OriginalClass`. This value is visible even after the document is reclassified back to the Invoices class.

What next?

- Add custom extraction logic for a subclass

Add custom extraction logic for a subclass

By default, when you create a subclass, everything configured for the parent class is inherited by the subclass. If you view the class contents by clicking on the icon, and then expand the Fields or the Locators, all of the listed items are marked as *<inherited>*. You can however, overwrite these inherited items by creating your own extraction logic that includes additional locator methods mapped to one or more fields.

For more information on configuring extraction using classes, locator methods, and fields, see the *TotalAgility Transformation Designer Help.*

You can add your own custom extraction logic for a subclass by following these steps:

1. Open your **InvoiceProcessingAgility** project in the Transformation Designer.
2. Select your vendor subclass in the **Project Tree**.
3. In the **Project Tree**, click on the Show/Hide icon for the **Invoices** class to display its settings.

   The Show/Hide icon changes to and the class items for the **Invoices** class are displayed.
4. Expand the **Locators** list .

   A list of greyed-out and italic locators are displayed. These are inherited from the **Invoices** class.

   Use these locators as reference when creating your own locator.
5. Right-click on **Locators** and select **Add Locator** .

   The new locator name is available for editing and the **Details** pane is updated for the new locator.
6. Enter a name for your new locator.

   Name the locator based on the name of the vendor subclass, the locator type, and its purpose. For example, if the locator method is a Format Locator that extracts the bank account number for your vendor subclass, use the format: `FL_<vendorName>_AccountNumber`.

   Similarly, a Format Locator that extracts the Invoice Type would be called `FL_<vendorName>_InvoiceType`.

   The name of the locator is updated in the **Details** pane.
7. From the **Locator Methods** list, select a locator method and then click on the properties icon .

   The locator method properties are displayed.
8. Edit the locator method settings as needed.

   Test the extraction results using a set of test documents.

   Once happy with the locator results click **Close**.
9. Repeat steps 5 to 8 to add additional locator methods as needed.
10. Expand the Fields list .

    A list of inherited field groups are displayed.
11. Expand the derived Field Group where the field is located that you want to modify, and select your field .

    Alternatively, create your own custom field. Keep in mind that any custom filed is also needed by the **Invoices** class as every document is classified to the **Invoices** class. This means that any custom fields that are added to subclasses but not the **Invoices** class are lost during production.

For more information on custom fields, refer to the *Tungsten AP Agility Configuration Help*.

The field settings are displayed in the **Details** pane.

12. From the **Field populated by** list, select your newly created locator.
13. Edit any other field settings as needed.
14. Test the extraction for an invoice from this vendor.
15. Repeat steps 10 to 14 to add additional fields as needed.
16. Once you are happy with this vendor, optionally add other subclasses. If you are happy with your project, save and **Release** this project.

    The project is released and used to process invoices.

Related topics:

- Create a subclass for a vendor

Subclasses and the validation form

If you classify a document to a subclass it is possible to hide or show the relevant fields for that subclass on the validation form.

If you want a field displayed on the validation form when a document is classified as a specific subclass, deactivate that field for the Invoices class and then create a field setting entry for the subclass.

Activate the entry for the subclass only. AP Agility generates different document type variants for the Invoices class and the subclass, assigning the appropriate variants before validation.

The document variant cannot be changed on the validation form. This means that if the document is not correctly classified, it is not possible to see the fields for the correct class.