OnBase®

E-Forms

Reference Guide

Includes:

Installation Guide

Administration Guide

User Guide

Foundation 25.1

Documentation Notice

Document Name
E-Forms

Department/Group
Documentation

Revision Number
Foundation 25.1

Overview

Introduction

The E-Forms module gives users the ability to complete and submit Hyper Text Markup Language (HTML)-based E-Forms into the OnBase document repository. When the form is submitted, OnBase automatically indexes the document by pulling Keyword Values from the E-Form's Keyword fields. Because documents are created in OnBase, E-Forms help reduce the paper usage associated with scanning and importing paper-based forms.

If used in conjunction with OnBase Workflow, submission of an E-Form can trigger a Workflow process. If an E-Form is initiated through a third party web site using the OnBase API, a similar process takes place.

E-Form templates are created outside of OnBase and imported into the Client module. This HTML template is assigned to the E-Form's Document Type and displays all the individual E-Form documents. If the system administrator has configured and imported the E-Form templates, new E-Form documents can be created through the Client module or the Workflow module. If the E-Form is incorporated into OnBase for the first time, an HTML E-Form template must be created.

Two file formats are used for E-Forms: Virtual Electronic Form and Electronic Form. Either the Virtual Electronic Form or Electronic Form file format must be assigned when creating the E-Form's Document Type.

An E-Form configured with file format Virtual Electronic Form stores only associated Keyword Values. The Keyword Values are stored in the database; nothing is stored in any Disk Group. All fields on an E-Form using file format Virtual Electronic Form must be a keyword; otherwise, the values entered into the field will be lost.

An E-Form using file format Electronic Form also saves Keyword Values to the database. However, a file containing all values of the E-Form is also saved to the Document Type's default Disk Group. When this E-Form is opened, the information in the file is used to fill any non-Keyword fields on the E-Form.

Applications

Many customers use OnBase E-Forms in conjunction with OnBase Workflow to implement paperless business processes, whereby standardized internal documents such as expense reports and vacation requests are created, stored, and routed entirely within OnBase. The addition of OnBase Web Server to this configuration enables the submission of online forms, such as order forms and membership applications, making OnBase a vehicle for facilitating e-commerce.

E-Forms can be used for request processes that may require several levels of verification. For example: Before approval, a Human Resources department's vacation request process may require that one or more managers sign the request. The OnBase Digital Signature module can be used in conjunction with Workflow and E-Forms to provide a means of electronically signing documents.

Example Applications:

- Requesting purchase orders.

- Ordering office supplies.
- Reporting software bugs and requesting software enhancements.
- Submitting online questionnaires.
- Creating shipping requests.

Licensing

Beginning in OnBase Foundation EP5, new customers must use simplified licensing to access E-Forms functionality. Existing customers upgrading from a version of OnBase prior to OnBase Foundation EP5 can continue to use legacy licensing to access this functionality.

If you are a new customer as of OnBase Foundation EP5 or greater, see Simplified Licensing on page 2.

If you are upgrading from a version of OnBase prior to OnBase Foundation EP5, see Legacy Licensing on page 2.

Simplified Licensing

The Standard User or Premier User license is required.

Legacy Licensing

E-Forms requires the E-Forms license.

Check your current licensing status by selecting **Utils | Product Licenses** from the Configuration module.

OnBase®

E-Forms

Installation Guide

Foundation 25.1

Installation

Requirements

The following sections outline requirement information specific to E-Forms in OnBase Foundation 25.1.

General Requirements

For general requirement information that applies to E-Forms and other modules, see the sections on the following topics in the **Installation Requirements** manual:

- Databases Supported
- Database/File Servers
- Database Client / Server Compatibility
- Supported Desktop Operating Systems
- Microsoft .NET Framework Requirements
- Microsoft Visual C++ Requirements
- Web Client Browser Requirements
- Unity Client Browser Requirements
- Client Retrieval Workstation Hardware Requirements
- Web Client Hardware Requirements
- Unity Client Platform Hardware Requirements
- Third-Party Software Compatibility
- About Virtual Environments
- 64-bit Support Statement
- Windows User Account Control Statement

Hyland Software - Microsoft Windows Updates

The developers of OnBase are dedicated to ensuring the regular cumulative updates released by Microsoft® are compatible with OnBase. The R&D Department of Hyland Software regularly evaluates the cumulative fixes released and labeled as Critical or Important by Microsoft. The details of the update provided by Microsoft are reviewed for interaction with OnBase, and the update is installed when appropriate for testing its compatibility with OnBase. If you have questions regarding a specific Microsoft cumulative update and its compatibility with OnBase, please contact your support provider.

Windows 10 Updates

For Windows 10 updates, Microsoft has introduced a new release cadence called the Semi Annual Channel (SAC). The SAC reduces the security patch and support cycle for versions of Windows 10 to 30 months. Hyland Software does not expect to encounter incompatibilities with Windows 10 updates, and it does not plan to change its process for the continued release and support of new versions of OnBase because of the new Microsoft SAC cadence. In the unlikely event that a future Windows 10 update introduces an incompatibility that prevents OnBase from operating as designed, Hyland will make commercially reasonable attempts to address the incompatibility in the latest release and the prior release. If an issue is determined to be related to an incompatible version of Windows 10, you may be required to upgrade to the current OnBase release to resolve the issue and maintain compatibility with Windows 10.

Third-Party Software Requirements

The ability to create and edit HTML documents is required for the E-Forms module. In many cases, an HTML editor, such as Microsoft Expression Web, is usually used to create HTML forms that are used in OnBase. Notepad, or any text editor, can be used if the user has HTML coding knowledge.

FormPop and PDFPop Browser Requirements

The following web browsers are supported for use with FormPop and PDFPop:

Web Browser	Supported Versions
Internet Explorer	Internet Explorer 11
Edge	EdgeHTML 14 and higher
Firefox	Firefox 28 and higher (including non-ESR versions)
Safari	Safari 9.1 and higher for OS X and macOS Safari for iOS
Google Chrome	Chrome 29 and higher

Licensing

See Licensing on page 2 for licensing requirements.

Pre-Installation

Before Installing the E-Forms module there must be:

- A functioning OnBase database.
- HTML Document creation software. (e.g. a text editor or HTML editor such as Microsoft Expression Web).

Registration

Workstation registration is not required for E-Forms.

Installation

There is no installation process for E-Forms.

Command Line Switches

No command line switches apply to E-Forms.

Backup/Recovery

Backup

Configuration

The configuration of E-Forms is stored in the database. A proper backup of the database will contain all E-Form configuration information.

E-Form templates are commonly stored in the **System** document disk group (which is configured as the default disk group for the **SYS HTML Forms** Document Type). The data stored on the E-Form is stored in the E-Form's Document Type disk group.

If using E-Forms, it is important that the **System** disk group is backed up along with other disk groups.

E-Form templates are stored in the default disk group configured for the SYS HTML Document Type. E-Form files for individual documents are stored in the default disk group configured for the particular E-Form Document Type. The individual document E-Form file contains all the information posted by submitting the E-Form (edit fields, radio buttons, comments, etc. – including keywords.) Keywords from the E-Form are also stored in the database.

Registry Settings

No Registry Settings apply to E-Forms

External Files

It is necessary to backup the OnBase.ini file.

Recovery

Configuration

Restoring all disk groups and restoring the database from Backup will recover E-Forms and the E-Form data.

External Files

Restore the OnBase.ini file.

Module related .INI Options

The .INI file can be restored from the backup if the recovery machine is intended to be used for exactly the same purpose as the original machine. If this machine will be used for other modules, only the listed INI settings from the table above may need to be recovered.

Backwards Compatibility Options

The E-Forms module has the backwards compatibility option to store blank dates as midnight. When this option is checked in the Configuration module, all E-Forms, regardless of when the option was selected, will now store blank dates as midnight. To access this option:

1. Log in to the Configuration module.
2. Select **Utils | Backwards Compatibility**.
3. Select the **Store blank dates on** E-Forms as midnight check box.

Troubleshooting

The majority of E-Form issues have to do with improper configuration of the E-Form or improper import of the E-Form into OnBase.

- Make sure form fields do not contain extra spaces.
- Make sure all form field mappings are spelled correctly and use the proper case.
- When using radio buttons or drop-down lists, set initial state to **not selected** in the form field properties.
- Verify that any scripts in the E-Form run without error.

HTML Documents vs. E-Forms

HTML documents and E-Forms can look very similar when displayed in OnBase. However, unlike E-Forms, information entered in an HTML document does not persist. This means that, even if an HTML document has a Save or Submit button, you cannot save information entered into an HTML document.

For more information on why this is, see Limitations When Mapping Form Fields to Keyword Types on page 13.

E-Forms That Work Outside of Unity Client Do Not Work Inside Unity Client

As of OnBase 15, the Unity Client renders HTML Forms using the IE 9 Standards Mode. This is not a configurable option, and it can impact how E-forms function within the Unity Client.

If E-forms that work outside of the Unity Client are not functioning correctly within the Unity Client, you may try one of the following methods to have the E-forms function correctly:

- Update the Windows registry.
- Include the **X-UA-Compatible** setting in the E-form itself.

CAUTION: E-forms will support setting the **X-UA-Compatible** value in meta tags. However, this setting should be thoroughly tested in a testing environment before being implemented system-wide.

Some XML documents are not being displayed correctly.

Beginning in Microsoft .NET Framework 2.0, the **XslTransform** class was deprecated and replaced with a new XSLT implementation, **XslCompiledTransform**. This change to the Microsoft .NET Framework enforced stricter standards for how style sheets must be created, and may cause display issues when viewing XML documents in OnBase with style sheets that do not conform to these stricter standards. For example, XML documents may be displayed as shifted down and to the right, or other unexpected behavior may occur when viewing documents with the affected style sheets.

Beginning with OnBase 9.0, this issue has been corrected for all custom and standard style sheets created by Hyland Software, including style sheets for the EDI processors. However, this issue may still affect style sheets created by Hyland Software prior to the release of OnBase 9.0, as well as any style sheets that users have developed on their own, depending on how they were created.

If you are upgrading from OnBase 8.2 or earlier, or if you believe that style sheets created for your OnBase solution are, or may be, affected, please contact Hyland Software's Technical Support for additional information and assistance in resolving this issue.

Common Questions

How do I stop headers and footers from printing on my E-Form?

To stop printing headers and footers on Internet Explorer documents and OnBase E-Forms:

1. In Internet Explorer, select **File | Page Setup**.
2. Remove information in the **Header** and **Footer** fields and click **OK**.

Upgrade Considerations

The following upgrade considerations have been compiled by OnBase subject matter experts. These upgrade considerations are general and applicable to most OnBase solutions and network environments and should be considered each time an upgrade is performed.

Carefully consider the impact of making any changes, including those listed below, prior to implementing them in a production environment.

For additional general information about upgrading OnBase, refer to the **Upgrade Guidelines** documentation and visit the Hyland Community.

E-Forms Upgrade Considerations

There are no additional upgrade considerations for this module.

Contacting Support

When contacting your solution provider, please provide the following information:

- The OnBase module where the issue was encountered.
- The OnBase version and build.
- The type and version of the connected database, such as Microsoft SQL Server 2022 or Oracle 19c, and any Service Pack that has been installed.

- The operating system that the workstation is running on, such as Windows 11 or Windows Server 2022, and any Service Pack that has been installed. Check the supported operating systems for this module to ensure that the operating system is supported.
- The name and version of any application related to the issue.
- The version of Internet Explorer and any Service Pack that has been installed, if applicable.
- A complete description of the problem, including actions leading up to the issue.
- Screenshots of any error messages.

Supplied with the above information, your solution provider can better assist you in correcting the issue.

OnBase®

E-Forms

Administration Guide

Foundation 25.1

Configuration Overview

Configuring an E-Form is a 4-step process. To configure an E-Form:

1. Create an HTML form with an external editor.
2. From the Client module, import the HTML Form into the **SYS HTML Forms** Document Type.
3. Create a Document Type or use an existing Document Type that has a Default File Format of **Electronic Form**. Forms can also be configured as a **Virtual Electronic Form**.
4. Configure the form as an **E-Form** for the Document Type.

The following limitations apply when configuring E-Forms:

- To ensure proper encoding, map all Keyword Types associated with an E-Form's Document Type on the E-Form template. These mappings can be hidden keywords if you do not want all of the Keyword Types displayed on the E-Form.
- If the HTML Form is a Unicode HTML Form, ensure that the form is imported with an HTML Unicode file type.
- If using a Cascading Style Sheet for E-Forms in the Web Client, the form creator is responsible for ensuring font accuracy.
- The Specific Currency Data Type should not be used on documents of type E-Form or Virtual E-Form. This data type is not supported on either of these document types.

Creating an HTML Form with an External Editor

CAUTION: E-forms will support setting the **X-UA-Compatible** value in meta tags. However, this setting should be thoroughly tested in a testing environment before being implemented system-wide.

E-Form templates are written in HTML. HTML code can be written directly in a program such as Notepad®, or in an HTML editor, such as Microsoft Expression Web.

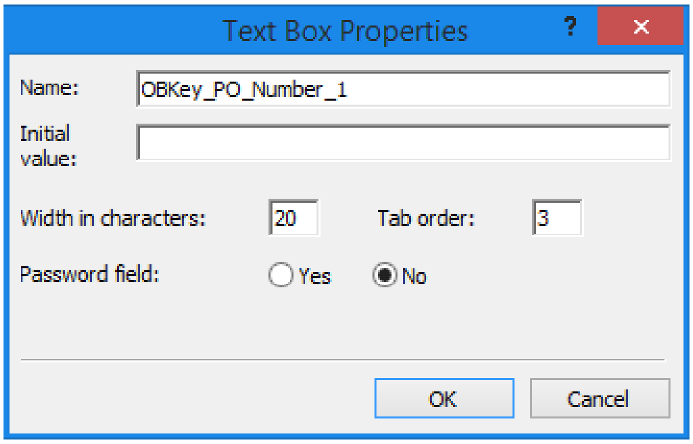
E-Form Templates contain input fields that are mapped to document Keyword Types and other system values. E-Forms can also contain information that is not mapped to keywords or stored in the database. This information is stored with the form and is available for viewing, though the system does not search documents based on this text.

The following limitations apply when configuring E-Form templates:

- When mapping fields in a form that contain either keyword data or non-keyword data, the "name" attribute must be used in order to store data.
- If you have more than one field with the same name property (e.g. <input name="Text1" /> and <input name="Text1" />), the first instance overwrites the value in any subsequent instances.
- An E-form document will continue to use the HTML template revision that was used when the E-Form was created.

Information entered into properly configured form fields is saved as Keyword Values or system values on the E-Form. The most important aspect of forms creation is the proper mapping of form fields to document Keyword Types or system values. For more information, see Mapping Form Fields to Keyword Types on page 13.

The method of identifying Keyword Types and system values in the HTML form depends on the HTML editor. In the following example, the map code for the Keyword Type **PO Number** is mapped in Microsoft Expression Web:



Name: OBKey_PO_Number_1
Initial value:
Width in characters: 20
Tab order: 3
Password field: Yes [✓ No]

Limitations in HTML Editors

The following limitations apply when working with HTML editors:

- Some HTML editors will not store text entries including the character "**&**" correctly in the database. For example, the entry **Shipping & Receiving** may be stored as **Shipping &amp; Receiving** in HTML code. In this case, a search for the Keyword Value **Shipping & Receiving** would not return expected results.
- Some HTML editors store extra spaces as **&nbsp;** which can cause the maximum length of the Keyword Value to be exceeded or produce unexpected search results. It is best practice to not append extra spaces to drop-down Keyword Value select lists. Check the HTML code to confirm that spaces are stored in a way that is compatible with the system database.
- When retrieving an E-Form with **Date & Time**, **Currency**, or **Floating Point** data types, please note that although check boxes and radio buttons may not appear to have values, the associated keyword values are saved within the document and available by viewing the document's keywords.
- Multiple instances of the same keyword type cannot appear on an E-Form if that keyword value is defined by a radio group.
- Fields disabled or enabled using JavaScript are not supported and any attempt to use them yield unexpected results.
- Any codes used to map Keywords, including the Keyword name itself, are not case sensitive

Using Form Tags with E-Forms

Before beginning any form field mapping, the form must be created first. Regardless of whether you use a text or HTML editor, all information on the HTML E-Form must be contained within form tags.

The form tag must include `<form method="POST">` for the E-Form to submit correctly into OnBase. If this tag is not included, the E-Form's **Submit** button acts as a reset button instead of submitting the form into the database.

The following is an example of the correct use of form tags:

```
<html>
    <head>
        <title>New Page 1</title>
    </head>
    <body>
        <form method="POST">
        </form>
    </body>
</html>
```

Mapping Form Fields to Keyword Types

Form fields such as text boxes, check boxes, radio buttons, and multiple select boxes can be mapped to Keyword Types. There are two ways to map form fields to Keyword Types:

1. Map the form field to a Keyword Type Number.
2. Map the form field to a Keyword Type Name.

In most cases, mapping to a Keyword Type Number is recommended. When Keyword Type Names are used, the renaming of keywords (without a subsequent adjustment to E-Forms) can cause E-Forms to malfunction. Keyword Number mapping is effective when Keyword Type names contain characters that are not accepted by HTML.

Mapping to a Keyword Type Name is recommended only if forms must be imported into multiple databases. Keyword Numbers cannot be changed in the system. If a form that uses Keyword Numbers is imported into multiple databases, all form field properties must be changed to map to the Keyword Numbers in the new database. Additionally, if the Keyword Type name contains a character that cannot be used in an HTML attribute without being escaped, the Keyword Type number should be used instead.

Limitations When Mapping Form Fields to Keyword Types

The following limitations apply when mapping form fields to Keyword Types:

- The E-Forms module does not support mapping Keyword Types that belong to Multi-Instance Keyword Type Groups to radio buttons. Keyword Types belonging to Multi-Instance Keyword Type Groups can be mapped to text boxes, scrolling text boxes, or multiple select boxes. There is limited support for mapping Keyword Types that belong to a Multiple Instance Keyword Type Group to check boxes. Keyword Types that belong to a Multiple Instance Keyword Type Group that are mapped to multiple check boxes on an E-Form must be configured with the same value within the HTML code. The value assigned will determine whether the check box is selected or deselected. For example, if the value used is “on,” the check box will only be selected when the Keyword Type value is “on”. All other Keyword Type values will leave the check box deselected.
- When using standard Keyword Type Groups, using check boxes to allow multiple values to be selected for a single Keyword Type within the Keyword Type Group is not supported. If you need to provide the possible options that a user can select, radio buttons or single select lists can be used.

- Configuring Keyword Types on a form that are not assigned to the Document Type the E-Form belongs to is not supported.
- To stay within the maximum number of characters allowed for a Keyword value and avoid truncated values, you should either implement a **maxlength** attribute on the input field corresponding to the maximum Keyword length or use JavaScript to check if the field value length is greater than the maximum length of the associated Keyword and display a message if the value is going to be truncated.
- In the Core-based interfaces, Keywords with underscore characters (_) are treated as spaces. This has implications for filtering. If a system contains two Keywords configured as Keyword A and Keyword_A, both of them would be mapped as Keyword_A. If both Keywords are configured on a filter, the system will choose one randomly to filter with.

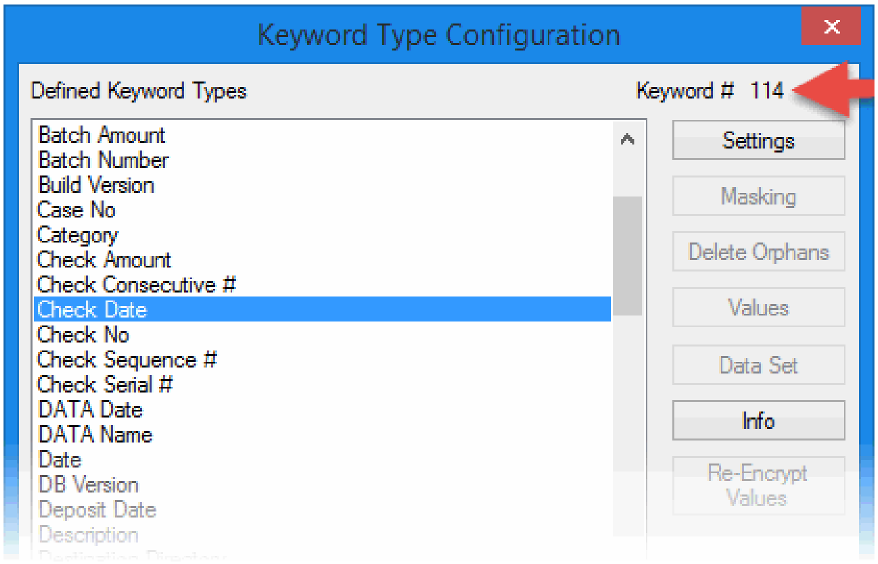
  In the OnBase Client, the underscore character (_) is reserved for HTML. If a Keyword Type name contains an underscore, it must be mapped as a Keyword Type Number.
- When importing HTML documents into the system to be stored as a Document Type (not using them as E-Forms), be aware of fields mapped to OBKey formats. Form objects containing OBKey formats are controlled by Keyword values. This means that HTML documents with fields controlled by OBKey will reflect the currently assigned keyword value. However, changes to these fields will not persist to the underlying keyword value in the Unity or Web Client like they would when saving or submitting an E-Form.

  For example, if an HTML document has an **Account Number** field and a **Save** button, entering an account number and clicking **Save** will not record the entered number.

  Note that the OnBase Client is an exception. Keyword values will persist when saving or submitting an HTML document.
- Each field within an E-Form must have one unique Keyword Type assigned to each field if using a Keyword Type Group. Multiple instances of the same Keyword Type are not supported in E-Forms for Keyword Type Groups.
- An E-Form must have the same number of fields on it as they are Keyword fields in the Keyword Panel. The fields can be added and then hidden.

Keyword Type Number

The Keyword Type Number is saved in the format **OBKey__KeywordNumber_#**. In the OnBase Client, it is displayed by default in the upper right-hand corner of the **Keyword Type Configuration** dialog box.



Batch Amount
Batch Number
Build Version
Case No
Category
Check Amount
Check Consecutive #
Check Date
Check No
Check Sequence #
Check Serial #
DATA Date
DATA Name
Date
DB Version
Deposit Date
Description

The following table defines each part of the Keyword Type Number format:

OBKey__	Every Keyword Value that is mapped must start with the Document Management System code **OBKey__**. **Note: OBKey** is followed by two underscore characters (__) when using a Keyword Number.
KeywordNumber	The Keyword Number to which the form field is mapped. This number is assigned to the Keyword Type when the Keyword Type is created. If the **Customer Name** Keyword Type has a Keyword Number of **110**, a customer name field could be mapped with **OBKey__110_1**.
#	The number of occurrences of the Keyword on the form. For example, the first use of the Customer Name Keyword would be mapped as **110_1**, the second occurrence would be mapped as **110_2**, and so on.

The following limitations apply when configuring the Keyword Type Number format:

- Any codes used to map Keywords, including the Keyword name itself, are not case sensitive.
- When using a currency Keyword Type, the E-Form formats currency according to the user locale (upon submission of the E-Form), unless a format for the currency has been applied to the Keyword Type during configuration.

Keyword Type Name

The Keyword Type Name is saved in the format **OBKey_Keyword_Type_Name_#** .

The following table defines each part of the Keyword Type Name format:

OBKey_	Every Keyword Value that is mapped to a Keyword Type name must start with the Document Management System code **OBKey_**. **Note: OBKey** is followed by one underscore character (_) when using a Keyword Type name. **CAUTION:** The **OBKey** tag should only be used to identify Keyword Type fields. It should never be used to identify non-Keyword fields.
Keyword_Type_Name	The name of the Keyword Type to which the form field is mapped. The Keyword Type name must match the Keyword Type name configured in OnBase. The Keyword Type name must appear in the map with underscores rather than spaces. For example, the Keyword Type **PO Number** can be mapped as **PO_Number** or **po_Number**.
#	The number of occurrences of the Keyword on the form. For example, the first use of the Keyword Type **PO Number** would be mapped as **PO_Number_1**, the second occurrence would be mapped as **PO_Number_2**, and so on.

The following limitations apply when configuring the Keyword Type Number format:

- Any codes used to map Keywords, including the Keyword name itself, are not case sensitive.
- When using a currency Keyword Type, the E-Form formats currency according to the user locale (upon submission of the E-Form), unless a format for the currency has been applied to the Keyword Type during configuration.
- Due to the way Internet Explorer renders multiple, continuous spaces, Keyword Values entered on an E-Form are stored with a single space in the place of any multiple, continuous spaces that are entered. For example, if a value of Sarah_ _Adams (where "_" represents a space) is entered on an E-Form, the value Sarah_Adams is stored. Do not put Keyword Types (including those that use Data Sets) that must store multiple, continuous spaces on an E-Form.

Using Multi-Instance Keyword Type Groups

To use a Multi-Instance Keyword Type Group (allowing multiple Keyword Type values for the same Keyword Type, creating multiple Keyword Type Group instances), an instance of the Keyword Type when configuring the form field must be specified. Each Keyword Type Group instance should be associated with a specific occurrence number. For example, a Multi-Instance Keyword Type group containing **Name**, **Address**, and **Phone Number** Keyword Types must have two sets of data to be entered in the form, the fields must be mapped with the following text:

```
<p><input type="text" name="OBKey_Name_1" size="20"><input
 type="text" name="OBKey_Address_1" size="20"><input type="text"
 name="OBKey_Phone_Number_1" size="20"></p>
<p><input type="text" name="OBKey_Name_2" size="20"><input
 type="text" name="OBKey_Address_2" size="20"><input type="text"
 name="OBKey_Phone_Number_2" size="20"></p>
```

In this example, information for two different people could successfully be entered and saved in an E-Form.

The following limitations apply when using Multi-Instance Keyword Type Groups in E-Forms:

- There cannot be multiple values for the same Keyword Type within the same Keyword Type Group instance. Configuring more than one field for the same instance creates a new instance with only the "orphaned" value in the Keyword Type Group. Additionally, configuring more than one field using the same occurrence number clears the second field from the form. The value will be viewable in the **Add/Modify Keywords** dialog box.
- When configuring a form to use a Multi-Instance Keyword Type Group, all Keyword Types that are in the group must be displayed on the form. Creating a form that does not contain all values of the keyword group can lead to unintentional deletion of Keyword values by an end-user.

Using AutoFill Keyword Set Buttons with Multi-Instance Keyword Type Groups

If you want to use either the **OBBtn_KS###** or the **OBBtn_ExpandKS###** button to automatically fill Multi-Instance Keyword Type Groups on an E-Form, you must specify a button for each instance of the Multi-Instance Keyword Type Group that you want to fill. This is accomplished by specifying the instance you want to fill in the E-Form.

For example, `OBBtn_ExpandKS###_1` would specify that you want the first instance of the Multi-Instance Keyword Type Group (denoted by the `_1`) to be filled with the AutoFill Keyword Set specified (`###`) when this button is clicked. Likewise, if you want to fill a second instance of a Multi-Instance Keyword Type Group, a button configured as `OBBtn_ExpandKS###_2` would need to be on the E-Form.

Note: If you have multiple AutoFill Keyword Set instances on a form, clicking an expand button will expand the first open instance it finds within a Core interface. For example, if you have 4 instances, you have not expanded any AutoFill Keyword Sets, and you click the button for the fourth instance expansion, the first instance is expanded and filled because this was the first empty instance that was found.

Using Auto-Incrementing Keywords

Auto-incrementing Keywords automatically increment upon opening a new E-Form. Each time a new form is opened, regardless of whether the form is submitted, this Keyword number automatically increases by one.

The following table defines each part of the Auto-Incrementing Keyword format:

OBKey_	Every Keyword Value that is mapped to a KeyTypeName must start with the Document Management System code **OBKey_**. **Note: OBKey** is followed by one underscore character (_) when using a KeyTypeName and two underscore characters (__) when using a KeyTypeID.

KeyTypeName_	The name of the Keyword Type to which the form field is mapped. The Keyword Type name must match the Keyword Type name configured in OnBase. The Keyword Type name must appear in the map with underscores rather than spaces. For example, the Keyword Type **PO Number** can be mapped as **PO_Number** or **po_Number**.
KeyTypeID__	The ID number of the Keyword Type to which the form field is mapped. The Keyword Type ID must match the Keyword ID configured in OnBase.
#	The number of occurrences of the keyword on the form. For the purposes of auto-incrementing, this is always initially mapped as **1**.

The following limitations apply when using auto-incrementing Keywords:

- Any codes used to map Keywords, including the Keyword name itself, are not case sensitive.
- Because the keyword automatically increments regardless of whether the E-Form is submitted, a retrieval list of the submitted forms may have gaps in the numbering. For example, the entire list may consist of E-Forms 1, 2, 5, 7, 17, and so on.

Using Keyword Data Set Values

A Keyword Data Set associated with the Keyword Type can automatically populate drop-down fields with Data Set values within E-Forms, HTML Custom Queries, and Workflow HTML user forms. Data Set values are appended to any existing values associated with the drop-down field. External Keyword Data Sets are also supported. The Keyword Type for the Data Set must be assigned to the Document Type. For more information on Keyword Type Data Set configuration, see the **System Administration** module reference guide.

Note: Ensure fields are not mapped to empty Data Sets. If a Data Set is empty, values are not saved regardless of whether values are hard coded in the form for selection.

The field using a data set must be named using the following format:
OBDataset_Keyword_Type_Name_#

The following table defines each part of the Keyword Data Set format:

OBDataset_	Each Keyword Type Data Set mapping must begin with **OBDataset_** or **OBDataset__.** **Note: OBDataset** is followed by one underscore character (_) when using a Keyword Type Name and two underscores (__) when using a Keyword Type Number. The following limitations apply with this piece of the Keyword Data Set format: • If creating scripts to manipulate E-Form data, please note that the **OBDataset_** mapping is converted to an **OBKey_** mapping in the OnBase Client. • When using data sets within E-Forms, data set values should have only one space between each word. Regardless of if more than one space is placed between words within a data set value, only one space is displayed and stored when using the data set value in an E-Form.
Keyword_Type_Name or Keyword Type #	Keyword Type Data Sets are mapped similarly to Keywords. The name of the Keyword Type to which the form field is mapped can be identified by the Keyword Type Name or the Keyword Type Number. The Keyword Type Name must match the Keyword Type Name configured in the system. Map codes are not case sensitive. The Keyword Type Name must appear in the map with underscores rather than spaces. For example, the Keyword Type **Request Type** is mapped as **Request_Type_#** or **request _type_#**.
#	The number of occurrences of the Keyword on the form. For example, the first use of the Keyword Type **Request Type** would be mapped as **Request_Type_1**. The second occurrence would be mapped as **Request_Type_2**, and so on. **Note:** If the Keyword Type is also mapped on the same form using the **OBKey_** or **OBKey__** mappings described above, then subsequent uses of **OBDataset_** mappings must have the number of occurrences properly incremented. For example, if **OBKey__1_1** is included on the form, then a second Data Set mapping must be **OBDataset__1_2**.

The following are examples of data set fields:

Single Selection List:

```
<select size="1" name="OBDataset_KeywordName_1"></select>
```

Multiple Selection List:

```
<select size="1" name="OBDataset_KeywordName_1" multiple="multiple></select>
```

Note: Any codes used to map Keywords, including the Keyword name itself, are not case sensitive.

Using Default Values for Keyword Types

If you want to use a default value for a Keyword Type for an E-Form, it must be coded within the form template. The following is an example of a Keyword Type configured with a default value:

```
<input type="text" VALUE="Default Text" name="OBKey_Keyword _1" size="35">
```

In this example, the **VALUE** attribute is set equal to "Default Text" and by default this text displays in the E-Form for the Keyword Type.

Using Operators for Custom Queries

Some instances call to mapping fields in such a way that Keyword Values are evaluated and results are returned.

The following table defines operators used for custom queries:

CAUTION: These tags are supported only for E-Forms and HTML Form custom queries. Using these tags in standard HTML documents stored in OnBase is not supported, and will yield unexpected results.

<SELECT name=OBOperKey__keyword_ keywordoccurrence>	This tag specifies that a Keyword Value will be evaluated using an operator for the custom query. The Keyword Type Number or the Keyword Type Name must be placed in the tag where **keyword** is specified, along with the occurrence of the Keyword Type within the form. If the Keyword Type Name is used, the Keyword Type Name must match the Keyword Type Name configured in the system. Examples: **<Select name=OBOperKey__318_1>** **<Select name=OBOperKey_Patient_Name_1>** **Note: OBOperKey** is followed by one underscore character (_) when using a Keyword Type name and two underscores (__) when using a Keyword Number.
<OPTION>	This tag specifies what operators are available for each **OBOperKey** instance. The following operators and their corresponding codes are available for use: • **=**(equals) • **&lt;**(less than) • **&lt;=**(less than or equal to) • **&gt;**(greater than) • **&gt;=**(greater than or equal to) • **&lt;&gt;**(Not equal to) There is no space between the codes. • **&quot; &quot;**(Literal) There is a space between each **&quot;** • **IsNull**(null values) Examples: <OPTION>&lt;</OPTION> <OPTION>&lt;=</OPTION> <OPTION>&gt;</OPTION> **Note:** There must be an <OPTION> tag for each operator to be made available.

<INPUT name=OBKey__keyword_keywordoccurrence>	This tag specifies the Keyword Type to which the input field is linked. The Keyword Type Number or the Keyword Type Name must be placed in the tag where **keyword** is specified, along with the occurrence of the Keyword Type within the form. If the Keyword Type name is used, the Keyword Type name must match the Keyword Type name configured in the system. Example: **<INPUT name=OBOperKey__318_1>**
<SELECT name=OBOperator__keyword_keywordoccurrence> <OPTION selected>AND</OPTION> <OPTION>OR</OPTION></SELECT>	The **OBOperator** tag allows the inclusion of the **AND** and **OR** operators after the input field, in order to create more complex search options. The Keyword Type number or the Keyword Type name must be placed in the tag where **keyword** is specified, along with the occurrence of the Keyword Type within the form. If the Keyword Type Name is used, it must match the Keyword Type Name configured in the system. The following limitations apply when working with this operator: • The **Use OR for duplicates** option applies to duplicate Keyword Type fields on an HTML form. If this option is selected, each value in duplicate Keyword Type fields will be searched for separately, as if the values were joined by the **OR** logical operator. • The **Use OR for duplicates** option overrides the **OBOperator** tag. Even if a user selects **AND** to join the two Keyword values, the **OR** operator is used. If this option is not selected, the **OBOperator** tag is respected, and users can refine their searches using the **AND** or **OR** operator.

The following limitations apply to operators used for custom queries:

- Any codes used to map Keywords, including the Keyword name itself, are not case sensitive.
- The selected value can be specified by default for an operator. To do so, the following tag must be used for the option selected by default: **<OPTION selected></OPTION>**.

Example code:

```
<TBODY>
    <TR>
        <TD>Num #1:</TD>
        <TD><SELECT name=OBOperKey__315_1> <OPTION selected>=</OPTION>
 <OPTION>&lt;</OPTION> <OPTION>&lt;=</OPTION> <OPTION>&gt;</OPTION>
 <OPTION>&gt;=</OPTION> <OPTION>&lt;&gt;</OPTION> <OPTION>""</OPTION></
SELECT></TD>
        <TD><INPUT name=OBKey__315_1></TD>
        <TD><SELECT name=OBOperator__315_1> <OPTION selected>AND</
OPTION><OPTION>OR</OPTION></SELECT></TD>
    </TR>
    <TR>
        <TD>Num #2:</TD>
        <TD><SELECT name=OBOperKey__315_2> <OPTION selected>=</
OPTION><OPTION>&lt;</OPTION> <OPTION>&lt;=</OPTION> <OPTION>&gt;</
OPTION><OPTION>&gt;=</OPTION> <OPTION>&lt;&gt;</OPTION> <OPTION>""</OPTION></
SELECT></TD>
        <TD><INPUT name=OBKey__315_2></TD>
    </TR>
    <TR>
        <TD align=middle colSpan=4><INPUT type=submit value=Query
 name=OBBtn_Yes>&nbsp; <INPUT type=submit value=Cancel
 name=OBBtn_Cancel><BR></TD>
    </TR>
</TBODY>
```

Using Dynamic Keywords in E-Forms

If an E-Form contains custom scripting to create additional keyword value fields on the E-Form, you must configure an **OBKeyProperty_** field within the E-Form's HTML. An **OBKeyProperty_** field must be configured for each Keyword Type for which fields are dynamically created in order for the values of those Keyword Types to be stored. This field should be hidden on the form and the value must equal "DYNAMIC".

Note: **OBKeyProperty_** fields are not supported in Core interfaces.

The following field syntax must be used, where `KeywordName` equals the Keyword Type name that is dynamically created:

```
<INPUT id="OBKeyProperty_KeywordName" type="hidden" value="DYNAMIC"
 name="OBKeyProperty_KeywordName">
```

or the following, where `KeywordNumber` equals the Keyword Type number that is dynamically created:

```
<INPUT id="OBKeyProperty_KeywordNumber" type="hidden" value="DYNAMIC"
 name="OBKeyProperty_KeywordNumber">
```

Mapping Form Fields to System Properties and Document Properties

Input form fields on the E-Form can be used to update system information stored in the database. Mapping form fields to system properties is similar to mapping form fields to Keyword Values. System properties can be mapped to Keyword Types via radio buttons, edit fields, combo boxes, check boxes and select lists. Form fields use special tags (described below) to identify system information.

The following specialized tags are available when creating an HTML document for use as an E-Form. The E-Form user can change these values:

System Properties

The following table defines system properties used in E-Forms:

CAUTION: These tags are supported only for HTML Form custom queries. Using these tags in standard HTML documents stored in OnBase is not supported, and will yield unexpected results. If these tags are configured on a standard E-Form, these date fields are ignored on a custom query initiated on an E-Form.

System Property	Description
OBDocumentDate	Used by HTML Form Custom Queries to restrict the search using the document date.
OBFromDate OBToDate	Used by HTML Form Custom Queries to restrict the date range of a search. These fields are not used when custom queries are initiated through a configured button on an E-Form.

The following limitations apply when working with system properties on E-Forms:

- **OBDocumentDate** and **OBFromDate** / **OBToDate** should not exist on the same Custom Query.
- These properties are not case sensitive.

Document Properties

The following document properties can be displayed automatically on a form. The E-Form user cannot change these values.

Document Property	Description
OBProperty_CurrentSessionID	Calls the current session ID. **Note:** This property only applies to Document Types with a File Format of **Electronic Forms** or **Virtual Electronic Forms**. **Note:** This property is supported in the Web Client and Unity Client. When the OnBase Client is configured to use an Application Server, a valid session ID is returned.
OBProperty_CurrentUserLocale	Displays the user locale of the currently logged in user's session on the form. It will be displayed in the format "language-region" for example the United States English locale would be displayed EN-US.
OBProperty_DocumentDate	The document date. The document date is assigned to a document at the time of import.
OBProperty_DateStored	The date when the document was imported into the system. If an invoice from December 28, 1996 was brought into the system on March 11, 1997, December 28, 1996 is the document date and March 11, 1997 is the date stored. Documents cannot be searched based on the date stored.
OBProperty_TimeStored	The time when the document was imported into the system.
OBProperty_UserName	The user who brought the document into the system.
OBProperty_ItemNum	The document handle, which is the unique number that identifies a document in the database.

Document Property	Description
OBRevisionComment	Fields tagged with this allow users to enter revision comments. If this field is populated, the document containing the field is assumed to be a revision and the user will not be prompted to select the revision option if the Document Type is configured for the prompt. **Note:** This property is not supported in the Unity Client. **Note:** An EDM Services license is required for revising.

The following limitations apply when using document properties in E-Forms:

- These tags are supported only for E-Forms. Using these tags in standard HTML documents stored in OnBase is not supported, and will yield unexpected results.
- Document properties are not stored on an E-Form until the E-Form has been submitted, and properties will not be available for viewing until the E-Form is retrieved for the first time.
- If both **OBDocumentDate** and **OBProperty_DocumentDate** are used on the same form, if the value stored for **OBDocumentDate** is changed, the value stored for **OBProperty_DocumentDate** is also changed to the new value.
- Formats for **OBDocumentDate, OBProperty_DateStored**, and **OBProperty_TimeStored** follow the system locale of the workstation. When using the Web Client, the workstation's locale is also used.
- These property names are not case sensitive.

User Properties

The following properties display information about the currently logged in user. A user can only access these properties when they are defined on the E-Form. If these properties are not defined on the E-Form, they are not added to the E-Form upon creation. All of the following properties display the currently logged in user on both new and archived E-Forms when the E-Form is loaded.

CAUTION: These tags are supported only for E-Forms. Using these tags in standard HTML documents stored in OnBase is not supported, and will yield unexpected results.

User Property	Description
OBProperty_CurrentUserEmailAddress	Displays the currently logged in user's email address, which is specified in the Configuration module. If no email address has been specified, this property is left blank.
OBProperty_CurrentUserGroupIDs	Displays the currently logged in user's User Group ID numbers. The property contains a comma (,) delimited list of the User Group ID numbers.
OBProperty_CurrentUserGroupNames	Displays the currently logged in user's User Group names. The property contains a comma (,) delimited list of the User Group names.

User Property	Description
OBProperty_CurrentUserID	Displays the User ID of the currently logged in user.
OBProperty_CurrentUserName	Displays the OnBase user name of the currently logged in user.
OBProperty_CurrentUserRealName	Displays the real name of the currently logged in user, which is specified in the Configuration module. If no real name has been specified, this property is left blank.
OBProperty_CurrentUserDisplayName	Displays the display name of the currently logged in user based on the global client setting in the Configuration module. If the global client setting is set to show the user's real name, and no real name has been configured, this property displays the OnBase user name.

Note: These property names are not case-sensitive.

Buttons

The following specialized buttons can be used on an E-Form.

The following limitations apply when working with buttons on E-Forms:

- These tags are supported only for E-Forms and HTML Form custom queries. Using these tags in standard HTML documents stored in OnBase is not supported and yields unexpected results.
- Any codes used to map Keywords, including the Keyword name itself, are not case sensitive.

Button	Description
OBBtn_CrossReference	This button type causes a properly configured cross-reference to execute. A keyword cross-reference must be set up between the E-Form and the related document. This button must be **type=submit**. Example HTML text is shown below. Quotes ("") are optional. **Button Text** represents the text that is displayed on the E-Form button. Replace **Button Text** with the actual text to display. **<input type="submit" value="Button Text" name="OBBtn_CrossReference">** The following limitations apply when working with this button type: • This button type only functions after a form has been submitted. • Only one cross-reference button can be configured from an E-Form. • The **Cross-Reference** button will be active on an E-Form even if the user viewing the form has no Modify rights for the form. • If the cross-reference is based on multiple keywords, all keywords configured for the cross-reference must be specified on the form in order for the cross-reference to execute properly.

Button	Description
OBBtn_KS### (Where ### represents an AutoFill Keyword set number)	This button type causes the specified Keyword set on the E-Form to be expanded upon E-Form creation when an AutoFill Keyword Set is properly configured and populated. This button must be **type=submit.** Once an E-Form is submitted, this button does not function and the **OBBtn_ExpandKS###** button must be used to expand a new AutoFill Keyword Set. When a primary Keyword is entered on the form and the **OBBtn_KS###** button is clicked, the corresponding Keyword values for the specified Keyword set are filled in on the form. For example, clicking a button mapped as **OBBtn_KS101** populates the Keyword fields on the form that are members of the AutoFill Keyword Set **101**. A number in the database identifies each AutoFill keyword set. By default, this number is displayed in the upper right corner of the **AutoFill Keywords Configuration** dialog box. To view the number associated with an AutoFill Keyword set, select the **Keywords** drop-down menu in the Configuration module, select **AutoFill Keyword Sets,** and click to identify the AutoFill Keyword set. The number is displayed in the upper right hand corner of the dialog box. Example HTML text is shown below. Quotes ("") are optional. **Button Text** represents the text that is displayed on the E-Form button. Replace **Button Text** with the actual text to display. **<input type="submit" value="Button Text" name=" OBBtn_KS###">** **Note:** Null Keyword Values can exist in an AutoFill Keyword set, resulting in null Keyword Values on the E-Form. Existing Keyword Values on the E-form are emptied when the **OBBtn_KS###** button is pressed and values are replaced by those in the AutoFill keyword set. **Note:** The primary Keyword Type of the AutoFill Keyword Set should not be of the date format if this button resides in an E-Form to be used in the Web Client.

Button	Description
OBBtn_KS### (Where ### represents an AutoFill Keyword set number) [cont.]	**Note:** When using this button in conjunction with Multi-Instance Keyword Type Groups, pressing this button opens the **Select Keyset** dialog box and allows the user to select one instance of the MIKG to expand for the AutoFill Keyword Set in which the primary AutoFill Keyword Set value has been entered. **Note:** Despite the AutoFill Keyword Set configuration, only a single instance of an AutoFill Keyword Set may be selected for an E-Form unless Multiple Instance Keyword Type Groups are used on the E-Form. **Note:** This button will not function when used in conjunction with the following Workflow actions: Display HTML Form and Display E-Form for Input. **CAUTION:** Expanding AutoFill Keyword Sets will fill Keyword Types configured as read-only even if the user does not have the **Access Restricted Keywords** privilege. **CAUTION:** Users without the **Access Restricted Keywords** privilege cannot submit E-Forms that include read-only keywords. **CAUTION:** In the Web Client, **OBBtn_KS###** and **OBBtn_ExpandKS###** buttons will not populate Keyword Types that are part of an AutoFill Keyword Set but not displayed on an E-Form. Use these buttons with caution when implementing a solution that involves hidden keywords or Workflow status keywords. **Note:** When this button is used on an E-Form that is read-only, it will be disabled. E-Forms become read-only when a digital signature is applied to them. **CAUTION:** This button is not supported in the Workflow module.

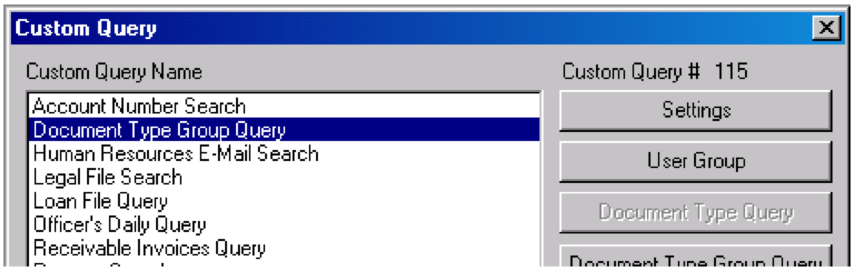
Button	Description
OBBtn_ExpandKS### (Where ### represents an AutoFill Keyword set number)	This button type populates an AutoFill based on a primary value on a new or existing E-Form. If an AutoFill was already expanded, on an E-Form, this button will replace the current secondary values with new values associated with the new AutoFill primary value. **This button must be type=submit.** When a primary keyword is entered on the form and the **OBBtn_ExpandKS###** button is pushed, the corresponding Keyword Values for the specified keyword set are filled in on the form. For example, pressing a button mapped as **OBBtn_ExpandKS101** populates the keyword fields on the form that are members of the AutoFill Keyword set 101. A number in the database identifies each AutoFill Keyword set. By default, this number is displayed in the upper right corner of the **AutoFill Keywords Sets Configuration** dialog box. To view the number associated with an AutoFill Keyword set, select the **Keywords** drop-down menu in the Configuration module, select **AutoFill Keyword Sets**, and click on the AutoFill Keyword Set. The number is displayed in the upper right hand corner of the dialog box. Example HTML text is shown below. Quotes ("") are optional. **Button Text** represents the text that is displayed on the E-Form button. Replace **Button Text** with the actual text to display. <input type="submit" value="Button Text" name="OBBtn_ExpandKS###"> **Note:** Null Keyword Values can exist in an AutoFill Keyword set, resulting in null Keyword Values on the E-Form. Existing Keyword Values on the E-Form are emptied when the **OBBtn_ExpandKS###** button is pressed and values are replaced by those in the AutoFill Keyword Set. **Note:** The primary Keyword Type of the AutoFill Keyword Set should not be of the date format if this button resides in an E-Form to be used in the Web Client. **Note:** Despite the AutoFill Keyword Set configuration, only a single instance of an AutoFill Keyword Set may be selected for an E-Form unless Multiple Instance Keyword Type Groups are used on the E-Form. **Note:** This button will not function when used in conjunction with the following Workflow actions: Display HTML Form and Display E-Form for Input.

Button	Description
OBBtn_ExpandKS### (Where ### represents an AutoFill Keyword set number) [cont.]	When using **OBBtn_ExpandKS###**, the keywords that are updated on the E-Form will not be updated in the database until the form is submitted. If expanded keywords on the form are modified using the **Add/Remove Keywords** dialog box, and the form is not saved, the form will only reflect the changes made in the dialog box. When the E-Form is saved, it will reflect the expansion as well as the changes made using the dialog box. **CAUTION:** In the Web Client, OBBtn_KS### and OBBtn_ExpandKS### buttons will not populate Keyword Types that are part of an AutoFill Keyword Set but not displayed on an E-Form. Use these buttons with caution when implementing a solution that involves hidden keywords or Workflow status keywords. **CAUTION:** This button is not supported for the **Display HTML Form** Workflow action. **CAUTION:** Expanding AutoFill Keyword Sets will fill Keyword Types configured as read-only even if the user does not have the **Access Restricted Keywords** privilege. **Note:** When this button is used on an E-Form that is read-only, it will be disabled. E-Forms become read-only when a digital signature is applied to them.
OBBtn_Yes or OBBtn_Save	Saves information to the database. A new document is created and Keyword Values on the form are saved to the database. This button must be **type=submit**. If the document is configured with file format **Virtual Electronic Form,** nothing is saved to the Disk Group. If the document is configured with file format **Electronic Form,** a file for the document is saved to the Document Type's default Disk Group. This file contains values for all fields on the form, including a duplicate set of the Keyword Values stored to the database. Example HTML text is shown below. Quotes ("") are optional. **Button Text** represents the text that is displayed on the E-Form button. Replace **Button Text** with the actual text to display. **<input type= "submit" value="Button Text" name="OBBtn_Yes">** **<input type="submit" value="Button Text" name="OBBtn_Save">** If you are using the Unity client and you want users to have access to the **Save** and **Save and Close** buttons in the **Electronic Form** ribbon, a **OBBtn_Yes** or a **OBBtn_Save** button must be configured on the form. **Note:** When using **OBBtn_Save** in the Web Client, a value for the **value** attribute must be specified. A single space can be used as the value.

Button	Description
OBBtn_SaveAndClose	Saves information to the database. A new document is created and Keyword Values on the form are saved to the database. Once saved, the E-Form will close. This button must be **type=submit**. If the document is configured with file format **Virtual Electronic Form** nothing is saved to the Disk Group. If the document is configured with file format **Electronic Form** a file for the document is saved to the Document Type's default Disk Group. This file contains values for all fields on the form, including a duplicate set of the Keyword Values stored to the database. Example HTML text is shown below. Quotes ("") are optional. **Button Text** represents the text that is displayed on the E-Form button. Replace **Button Text** with the actual text to display. **<input type= "submit" value="Button Text" name="OBBtn_SaveAndClose">** When this button is used in Workflow, it will function exactly as the **OBBtn_Yes** and **OBBtn_Save** buttons function.
OBBtn_SaveNoClose	Saves information that has been entered into an E-Form without closing the form. The E-Form is automatically saved as a revision if the Document Type is revisable. Additional changes can then be made to the E-Form. This button is useful to implement on long forms, when entering data might be a time-consuming task. Example HTML text is shown below. Quotes ("") are optional. **Button Text** represents the text that is displayed on the E-Form button. Replace **Button Text** with the actual text to display. **<input type= "submit" value="Button Text" name="OBBtn_SaveNoClose">** **Note:** For the E-Form to be saved as a revision, an EDM Services license is required. **Note:** This button has the same functionality as **OBBtn_Save** when used in a Workflow ad hoc task. **Note:** When used in HTML Custom Queries, this button will function similarly to **OBBtn_Yes** and **OBBtn_Save**. The Custom Query window will close and the results of the query will be displayed.

Button	Description
OBBtn_No	**Note:** This button is for use with Workflow. The functionality of this button only occurs in the Workflow window. Outside of Workflow the button has no special functionality. In Workflow, when this button is clicked, the E-Form is removed from the user interaction window and the task list continues. This button does not cancel the rest of a task list. Clicking this button sets the **Check Last Execution Result** rule to **False**. The button refreshes the currently displayed E-Form without saving it. During the refresh, the keyword values on the form are not saved to the database. This button must be **type=submit**. Example HTML text is shown below. Quotes ("") are optional. **Button Text** represents the text that is displayed on the E-Form button. Replace **Button Text** with the actual text to display. **<input type= "submit" value="Button Text" name="OBBtn_No">** **Note:** See the Workflow documentation for more information about this button's functionality in Workflow.
OBBtn_Cancel	When clicked, this button will close the E-Form and any data values entered on the form will not be saved. When an E-Form is used in Workflow and a button of type=submit has the value OBBtn_Cancel, when clicked, the form is not submitted, the Last Execution Result is set to False, and the entire task is aborted. When an E-Form is used in Workflow and a button of type=submit has the value OBBtn_No, when clicked, the form is not submitted, the Last Execution Result is set to False, but the Task is not aborted. Example HTML text is shown below. Quotes ("") are optional. Button Text represents the text that is displayed on the E-Form button. Replace Button Text with the actual text to display. Any Value represents any entered text. **<input type="submit" value="Button Text" name="OBBtn_Cancel">** **Note:** See the Workflow documentation for more information about this button's functionality in Workflow. If you are using the Unity client and you want users to have access to the **Cancel** button in the **Electronic Form** ribbon, a **OBBtn_Cancel** button must be configured on the form. **Note:** When this button is used on an E-Form that is read-only, it will be disabled. E-Forms become read-only when a digital signature is applied to them.

Button	Description
OBBtn_xRefItemnum	Retrieve a document based on the document handle value associated with the document. The keyword in the database must be named **xRefItemnum**. The text box used for entering the document handle value should be named "OBKey_xRefItemnum_1". The button used for retrieving the documents based on handle value must be named "OBBtn_xRefItemnum". Example HTML text is shown below. Quotes ("") are optional. Button Text represents the text that is displayed on the E-Form button. Replace Button Text with the actual text to display. **<INPUT type=submit value="Button Text" name="OBBtn_xRefItemnum">** The form must be submitted after the document handle value is entered in the text box. OBBtn_xRefItemnum can be used to retrieve the document only after the form has been submitted. **CAUTION:** When using the Unity Client or the Integration for Microsoft Outlook, in order to use this feature, user groups must have the **Retrieve by Document Handle/File Name** product right granted in the Configuration module.
OBBtn_AutoSave	This button functions in two ways: • It saves the E-Form • It creates a new AutoFill Keyword Set with the values entered into the E-Form by the user. **Note:** New AutoFill Keyword Sets are not created when used in Core-based Clients. At least one AutoFill Keyword Set must be defined on the form. If an AutoFill keyword set is assigned to the Document Type, the assigned keyword set is the one to which values are added. If no keyword set is assigned to the Document Type, the system searches all document keywords for keyword set primary keys. If a keyword set primary key is found and the rest of the associated keyword set Keyword Types are assigned to the Document Type, a new keyword set will be created. Example HTML text is shown below. **<INPUT type=submit value=AutoSave name=OBBtn_AutoSave>** **Note:** When this button is used on an E-Form that is read-only, it will be disabled. E-Forms become read-only when a digital signature is applied to them.

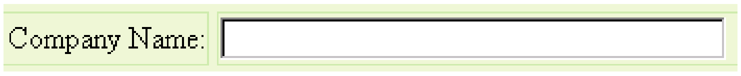
Button	Description
OBBtn_CQ##	Multiple Custom Queries can be run from an E-Form. Buttons must be of type=submit and named **OBBtn_CQ##**, where ## represents the Custom Query Number (for example, **OBBtn_CQ101** executes the 101 Custom Query). The Custom Query Number is displayed by default in the upper right-hand corner of the Custom Query Configuration dialog, as shown below.  Custom Query Custom Query Name Custom Query # 115 Account Number Search Document Type Group Query Human Resources E-Mail Search Legal File Search Loan File Query Officer's Daily Query Receivable Invoices Query Settings User Group Document Type Query Document Type Group Query If the Custom Query is a Keyword Type query, the Custom Query will use the Keyword Values populated on the E-Form. Example HTML text is shown below. Quotes ("") are optional. **Button Text** represents the text that displays on the E-Form button. Replace **Button Text** with the actual text to display. **<input type="submit" value="Button Text" name=" OBBtn_CQ##">** **Note:** This button type only functions after a form has been submitted in all supported interfaces except the Unity Client. The Unity Client allows you to use this button before E-Form submission. **Note:** If this button is used on a disabled E-Form field (i.e., a field on a digitally-signed E-Form), no documents will be returned. This button will return documents when used on a read-only E-Form field. An example of a read-only E-Form field is on an E-Form that a user does not have Modify rights to, or whose previous revision was read-only.
QueryStopUnload	This is a hidden field that will force a user to click a submit button to exit the E-Form when they attempt to close out of a modified E-Form using the X button. The field must use the following syntax: **<input type="hidden" name="QueryStopUnload">** **CAUTION:** If users close the Client before submitting the form with this field configured for the form, the form will not be saved successfully. **CAUTION:** This field is not supported in the Web Client.

Reset Buttons	Description
Type = reset	Any button of the type= **Reset** will clear all changes made to the form fields and restores the form fields to their previously saved values. The E-Form remains open. This button can be named anything. **Note:** Once an AutoFill Keyword Set is initiated, the reset button will not clear the values that populated the fields based on the AutoFill Keyword Set values. Example HTML text is shown below. Quotes (" ") are optional. **Button Text** represents the text that is displayed on the E-Form button. Replace **Button Text** with the actual text to display. Any Value represents any entered text. **<input type="reset" value="Button Text" name="Any Value">** If you are using a reset button in conjunction with the Unity client and you want users to have access to the **Reset** button in the **Electronic Form** ribbon, the **name** of the button must equal **OBBtn_Reset**.
Type = Submit	Any button of type **=submit** button that does not have a **name=** value of any of the **OBBtn** options described above should be named name **=OBBtn_Cancel**.

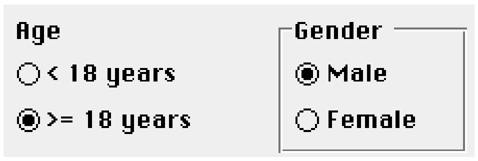
Form Fields

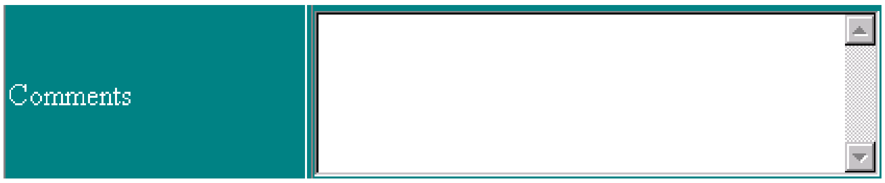
Form fields can be added to a form. Form fields are individual fields on a form that are used to gather information. Some examples are explained below.

CAUTION: In a non-E-Form HTML document, unmapped information contained in form fields cannot be modified or deleted. Unlike form fields mapped to a keyword type using the OBKey convention, if the unmapped form field is changed, the information contained in the field when it was imported into OnBase will populate the field when it is reloaded.

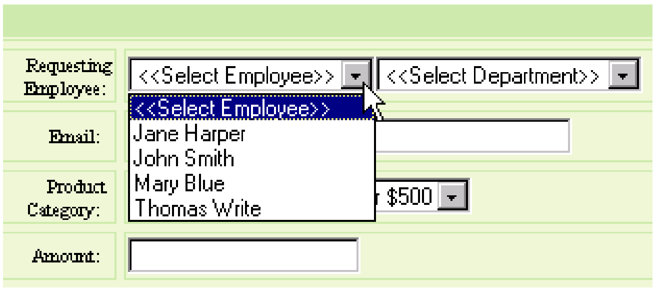
Form Field	Description
One-line text box	Use one-line text boxes to collect a small amount of text, such as a name or number. 

Form Field	Description

Radio Button	Use radio buttons when only one option should be selected from a group.
	 Age
	< 18 years
	>= 18 years
	Gender
	Male
	Female
	Note: A Keyword Type that uses a radio group can only appear once per E-Form.
	When using radio buttons with date values, the "value" tags needs to be the normalized format of the date ("YYYY-MM-DD" for date and "YYYY-MM-DD HH:MM:SS" for date and time). The following is an example:
	`<input type="radio" name="OBKey_EFormDropDownDate_1" value="2010-01-10" />01/10/2010`
	`<input type="radio" name="OBKey_EFormDropDownDate_1" value="2010-02-22" />02/22/2010`
	`<input type="radio" name="OBKey_EFormDropDownDate_1" value="2010-03-30" />03/30/2010`
	`<input type="radio" name="OBKey_EFormDropDownDate_1" value="2010-04-14" />04/14/2010`

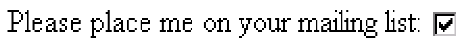
Scrolling Text Box	Use scrolling text boxes to collect one or more lines of text, such as a comment. This field scrolls to accommodate varying amounts of text. **Note:** This form field type should be used in non-keyword fields used for commenting on an E-Form. If this field is mapped to a Keyword Type, line breaks entered in the fields will be converted to 2 space characters before being saved to the database. 

Form Field	Description

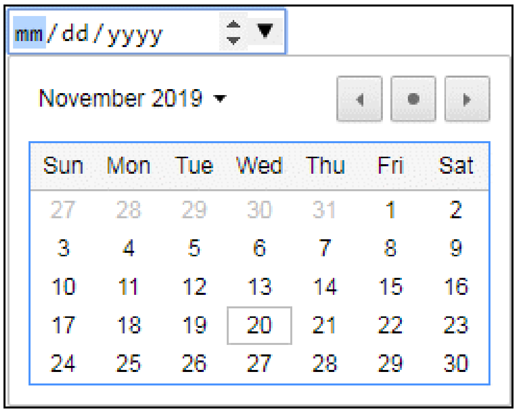
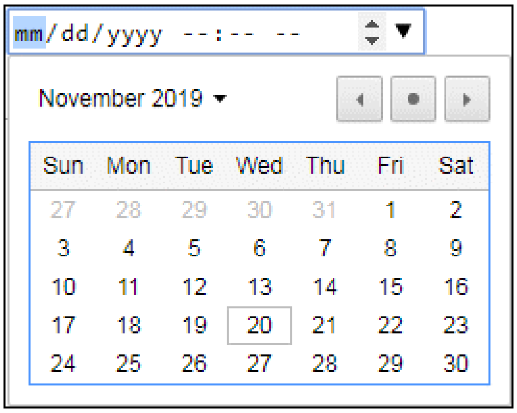
Drop-down Menu	Use a drop-down menu to present a list of choices. One or multiple selections can be configured for drop down menus.
	When using a drop-down list with date values, the "value" tags needs to be the normalized format of the date ("YYYY-MM-DD" for date and "YYYY-MM-DD HH:MM:SS" for date and time). The following is an example:
	`<select size="1" name="OBKey_EFormDropDownDate_1">`
	`    <option value="2010-01-10">01/10/2010</option>`
	`    <option value="2010-02-22">02/22/2010</option>`
	`    <option value="2010-03-30">03/30/2010</option>`
	`    <option value="2010-04-14">04/14/2010</option>`
	`</select>`
	 Requesting Employee:
	<<Select Employee>>
	<<Select Department>>
	<<Select Employee&gt;&gt;
	Jane Harper
	John Smith
	Mary Blue
	Thomas Write
	Email:
	Product Category:
	$500
	Amount:

Multiple Select List	Use a multiple select list to present a list of values. You can select multiple values within the list. **Note:** The Unity API does not support multiple select list fields on E-Forms. E-Forms containing a multiple select list keyword processed by Unity will not lose data; however, Unity will only evaluate the first value selected in the multiple select list field.

Form Field	Description

Check Box	Use check boxes for optional items. The E-Form user can select and clear multiple check boxes. The following is an example of a check box field:
	`<input type="checkbox" name="OBKey_checkbox_1" value="check box 1"/>Check Box 1`
	The check box can be defaulted to be checked by appending "checked" to the end of the tag. For example:
	`<input type="checkbox" name="OBKey_checkbox_1" value="check box 1" checked/>`
	Note: The value="" tag is a required entry for check boxes to function properly.
	When a check box is selected on a system form, the value associated with that check box in the HTML code is stored as a system property or Keyword Value. Documents can be searched for and retrieved based on the value stored.
	Please place me on your mailing list: ☑
	CAUTION: When mapping check boxes to Keyword Types, you must always use unique occurrence numbers for each check box field. For example, one check box would be mapped to **OBKey__110_1**. The next check box would need to be mapped to **OBKey__110_2**, and so on.
	CAUTION: When mapping check boxes to a Keyword Type within a Multi-Instance Keyword Type Group, only one check box can be mapped to a single Keyword Type for each instance of a Multi-Instance Keyword Type Group because there cannot be multiple values for the same Keyword Type within the same Keyword Type Group instance.
	When using check boxes with date values, the "value" tags needs to be the normalized format of the date ("YYYY-MM-DD" for date and "YYYY-MM-DD HH:MM:SS" for date and time). The following is an example:
	`<input type="checkbox" name="OBKey_checkboxdate_1" value="2010-01-10" />01/10/2010`
	`<input type="checkbox" name="OBKey_checkboxdate_2" value="2010-02-22" />02/22/2010`
	`<input type="checkbox" name="OBKey_checkboxdate_3" value="2010-03-30" />03/30/2010`
	`<input type="checkbox" name="OBKey_checkboxdate_4" value="2010-04-14" />04/14/2010`

Form Field	Description

Date Picker	Use date pickers to allow the E-Form user to enter or select a date or date and time value.
	Note: Date picker form fields are only supported when the rendering browser is set to **chromium**. For more information, see the section on configuring the Unity Client configuration file in the **Unity Client** module reference guide.
	The following input types are available:
	• **date**- A date field is presented for entry, including increase and decrease buttons and a drop-down button. When the drop-down button is clicked, a pop-up calendar is displayed for date selection:
	 mm/dd/yyyy
	November 2019
	Sun Mon Tue Wed Thu Fri Sat
	27 28 29 30 31 1 2
	3 4 5 6 7 8 9
	10 11 12 13 14 15 16
	17 18 19 20 21 22 23
	24 25 26 27 28 29 30
	For example:
	`<input type="date" name="OBKey__8_1" class="field" />`
	Note: The **date** input type is not supported for Chrome or Safari.
	• **datetime-local**- A date and time field is presented for entry, including increase and decrease buttons and a drop-down button. When the drop-down button is clicked, a pop-up calendar is displayed for date selection:
	 mm/dd/yyyy --:-- --
	November 2019
	Sun Mon Tue Wed Thu Fri Sat
	27 28 29 30 31 1 2
	3 4 5 6 7 8 9
	10 11 12 13 14 15 16
	17 18 19 20 21 22 23
	24 25 26 27 28 29 30
	For example:
	`<input type="datetime-local" name="OBKey__9_1" class="field" />`

Form Field	Description
Push Button	Use push buttons to: • Submit forms after they are filled out • Clear fields by resetting the form • Perform cross-references, use AutoFill keyword set, etc. **Note:** When using a button element, a type must be specified. For example: `<button name="OBBtn_Save" value= " " type="submit">` **Note:** When using a button element in the Web Client, a value for the **value** attribute must be specified. A single space can be used as the value.

CAUTION: When configuring fields on an E-Form that are not mapped to Keyword Types, **name** attributes for fields cannot contain spaces.

Configuring Default Values on the E-Form

In some instances, you may want a default value to display in a form upon creation. You can configure a default value within the form. To accomplish this, you must set a value attribute defining the default value for input fields or labels. The following is an example:

```
<P>Company Name:<INPUT name=OBKey__233_1 value="ABC Company"></P>
```

In this example, the field would display ABC Company by default.

If you want to provide a default value in a text area box, you must specify the text within the field. The following is an example:

```
<P><TEXTAREA name=S1 rows=9 cols=140> Please enter text. </TEXTAREA></P>
```

In this example, **Please enter text.** is displayed in the field by default.

Working with Embedded Images

This section contains information pertinent to embedding images in HTML documents.

Displaying Images on HTML Documents

You can place images on E-Forms, HTML documents, and HTML Custom Queries to customize their appearance. When adding images to HTML files that will be accessed from OnBase, be sure they comply with the following requirements:

- The image path must point to an absolute path that is accessible to all users who need to view the file. An absolute path is the full path to a file, beginning with the root directory (e.g., http://hostname/share/image.jpg).
- For files accessed from an Apple Safari ® or Mozilla Firefox ® Web browser, images must not be referenced through a UNC path or a file URL. For images to be displayed in Safari or Firefox, you must reference them from a hosted Web site using a URL address (e.g., http://hostname/share/image.jpg).

Displaying Images Stored in the Database on HTML Documents

If an image is already stored in the OnBase database, you can include it on HTML documents, Custom Queries, and E-Forms in the OnBase Client and Web Client.

Note: Displaying an image stored in OnBase on E-Forms is not supported in the Unity Client.

This feature lets you maintain an image on a document or form by updating the image in the OnBase database. Users who have rights to the Document Type the image is stored in can view the image within the document or form.

For example, if you revise a referenced image using EDM Services, the image on the HTML document will reflect the revision. You can also update stored images on HTML documents by modifying the Keyword Values on the referenced images.

To reference an image in OnBase, use the following format:

<img alt="mz:DocTypNum;KeyTypeNum1:KeyValue1;KeyTypeNum2:KeyValue2;">

Where:

- **DocTypeNum** is the image's Document Type number.
- **KeyTypeNum1** is the Keyword Type number of a Keyword Type on the image.
- **KeyValue1** is the value associated with KeyTypeNum1 on the image.

For example:

<img alt="mz:151;138:ONE;">

Notice that **KeyTypeNum** and **KeyValue** are separated by a colon, whereas other parameters are separated by semicolons. You can enter as many **KeyTypeID:KeyValue** pairs as needed to identify the image document.

Be sure to use Keyword Values that uniquely identify the image. If multiple images match the parameters provided, different images may be returned depending on the application. OnBase images must be referenced by a unique Keyword Value to link the image to the document.

Tip: Use the HTML document's file name as the Keyword Value parameter, and then index the image you want to display with the file name. This practice helps ensure that only one image is associated with the HTML document.

Additional Parameters

You can use an image's Document Date and page number to further refine a reference to an image.

When you reference an image by Document Date or page number, be sure to include enough Keyword Value parameters to uniquely identify the image you want to display.

Configuring HTML Forms for System Portability

Note: The following applies to E-Forms, Virtual E-Forms, HTML Documents, HTML Custom Queries, and Workflow User Forms.

When using HTML-based forms and documents, you can configure them to be portable between OnBase systems.

You can use the Application Server delimiter to create forms that will dynamically update the Application Server path to the Application Server defined in the Configuration module (**Utils | Application Server**).

To use this feature, place the following within the beginning and ending <HTML> tags of an HTML form or document:

~[APPLICATION_SERVER_URL]~

When this delimiter is placed in the HTML, the form will pull the Application Server URL as specified in OnBase Configuration and replace all instances of ~[APPLICATION_SERVER_URL]~ with the URL. The "service.asmx" portion of the Application Server URL specified in the configuration is stripped from the value.

For example: If the Application Server URL is specified in Configuration as "http://server/AppServer/service.asmx," the delimiter will replace all instances of ~[APPLICATION_SERVER_URL]~ with "http://server/AppServer."

Note: In some instances it may be helpful to have only the base server URL in the HTML document instead of the entire server path. In these instances, the delimiter ~[APPLICATION_SERVER_BASE_URL]~ can be used to return "http://server" instead of "http://server/AppServer."

Configuring an E-Form for Signature Pad Interface (TWAIN)

Note: E-Forms cannot be signed in the Unity Client or Web Client. Once an E-Form is signed, it cannot be viewed in the Unity Client.

E-Forms must be properly configured for capturing signatures using the Signature Pad Interface (TWAIN) module. An E-Form can hold up to 25 signatures. The user must sign in an image placeholder, in a pre-designated area. When creating the E-Form, ensure you use the following syntax to designate the signing area.

1. If you want to create an E-Form with one signature placeholder, insert this syntax in the form where you want the signature placeholder to reside.

   ```
   <P><IMG height="100" alt="signature" src="\\temp\signature.bmp" width="400"
   border="0"/></P>
   ```

2. If you want to create an E-Form with two signature placeholders, insert this syntax in the form where you want the signature placeholder to reside.

   ```
   <P><IMG height="100" alt="signature" src="\\temp\signature.bmp" width="400"
    border="0"/></P>
   ```

```
<P><IMG height="100" alt="2signature" src="\\temp\signature2.bmp"
 width="400" border="0"/></P>
```

Note: In these examples, the source location for the signature is \\temp. Ensure that this location is a secure network location.

Note: The E-Form should be saved in a text editor such as Notepad. HTML editors are not recommended because they can add extraneous information. For additional E-Forms best practices, see the E-Forms module reference guide's best practices appendix.

You can create a user-friendly label for the signature placeholder by placing the value you want to be displayed in the placeholder's **alt** value. This value should be formatted in one of the following ways:

```
alt="signature:Signature"
alt="2signature:Your Signature"
```

In these examples, **Signature** and **Your Signature** are the signatures that will be listed in the **Signature Capture** dialog box used during signature capture, providing an easy way to identify the exact signature the user wants to capture.

Configuring an HTML Custom Query for Use with Full-Text Search

In order to use Full-Text Search in conjunction with an HTML custom query, a specific field must be added to the HTML custom query form. In order to create a custom query that contains a field for full-text searching, a text field must be placed on the form and named **OBFullTextSearch**. Users can enter text in this field to perform full-text searches.

Importing The HTML E-Form Template into OnBase

To use an HTML E-Form template in OnBase, it must be imported through one of the Client modules (i.e. the OnBase Client, the Unity Client, or the Web Client). The HTML form is stored into the system **SYS HTML Forms** Document Type. The form will not change once it has been imported.

For more information on importing a document into OnBase, see the Client module reference guides.

The following considerations are required when importing an HTML form into OnBase:

- The HTML form must be imported into the **System Documents** Document Type Group as a **SYS HTML Forms** Document Type.
- In the **Keywords** section, type a description of the HTML form into the **Description** text box. It is considered best practice to include the name of the Document Type that will use the form.
- Once an HTML form has been imported into the system, it can be applied to a Document Type in the OnBase Configuration module.

Creating a Document Type

Create a Document Type or use an existing Document Type that has a Default File Format of Electronic Form.

Create an appropriate Document Type in the Configuration module.

Note: E-Forms must have a File Format of **Electronic Form (or Virtual Electronic Form)**.

Associate an HTML form with the Document Type by clicking **E-Form** from the **Document Type** configuration dialog box. Choose an HTML form from the drop-down menu, which displays all documents of the **SYS HTML Forms** Document Type.

Upon completion and submission of an E-Form, the system's default behavior is to ask users if they want to create an additional form of the same type. If you do not wish for this message to be displayed for an E-Form, in the Document Type associated with the form, check the **Don't prompt for new form after submit** check box. This is particularly useful for organizations that have no need to submit the same form consecutively.

CAUTION: The **Don't prompt for new form after submit** setting applies to all revisions of the E-Form.

Note: See Document Type XML Configuration on page 46 for more information on configuring a Document Type for use with an XML Cascading Style Sheet.

Forms Containing Scripts

If an E-Form has an embedded script that may need to be edited outside of OnBase, the E-Form template itself should be edited. This template should be in the Document Type SYS HTML Forms as outlined in the Importing The HTML E-Form Template into OnBase on page 42 section. If an E-Form itself (with Default File Format of Electronic Form or Virtual Electronic Form) is saved outside of OnBase using **File | Save As**, the embedded script will not be saved.

Configuring Document Type Revisions

The Document Type revision feature allows you to change Document Type configuration. Changes are day forward, affecting documents brought into OnBase after the Document Type revision was created. Previously configured settings remain in effect for existing documents.

Specifically, Document Type revisions pertain to the following settings associated with a Document Type:

- Overlay
- View/Print
- E-Form
- Data Mining
- XML Style Sheet

To create a Document Type revision:

1. In the Configuration module, select **Document | Document Types**. The **Document Types** dialog box displays.
2. Select a **Document Type** and click **Add Revision**.
3. A warning dialog box displays. Click **OK** to create a revision of that Document Type. Click **Cancel** to cancel.

   If you clicked **OK**, a dialog box displays indicating that the requested revision has been created.

4. Click the **Overlay** or **View/Print** button to modify the Overlay settings or View/Print settings of the Document Type revision you created.

   If your database is licensed for the appropriate modules, click the **E-Form**, **Data Mining**, or **XML Style Sheet** button to modify the E-Form settings, Data Mining settings, or XML Style Sheet settings of the Document Type revision you created.

   Note: In order to access the **XML Style Sheet** button, your system must be licensed for one of the following modules: E-Forms; XML Tag Import Processor; any one of the EDI processors; Integration for Open Text Fax Server, RightFax Edition; Integration for Biscom FAXCOM; or Integration for Esker Fax.

5. A new numeric identifier displays in the **Revision** list to indicate the new revision.
6. Ensure that the numeric indicator selected in the **Revision** list corresponds to the Document Type revision whose settings are to be defined.
7. Configure the settings as desired.
8. Click **Apply**, then **Close**.
9. From this point forward, all new documents will use the newly-configured settings. All old documents will use the previously-configured settings.

   Note: Previous revisions cannot be selected for current use. They only remain defined for use with previous documents.

XML Style Sheet Configuration

To ensure the highest level of security, more restrictive XSLT parsing has been put into place and it will generate an exception whenever a <xsl:include/> tag is present in a style sheet file. Here is an example of this tag, where FileName.xsl is the external .xsl file being referenced:

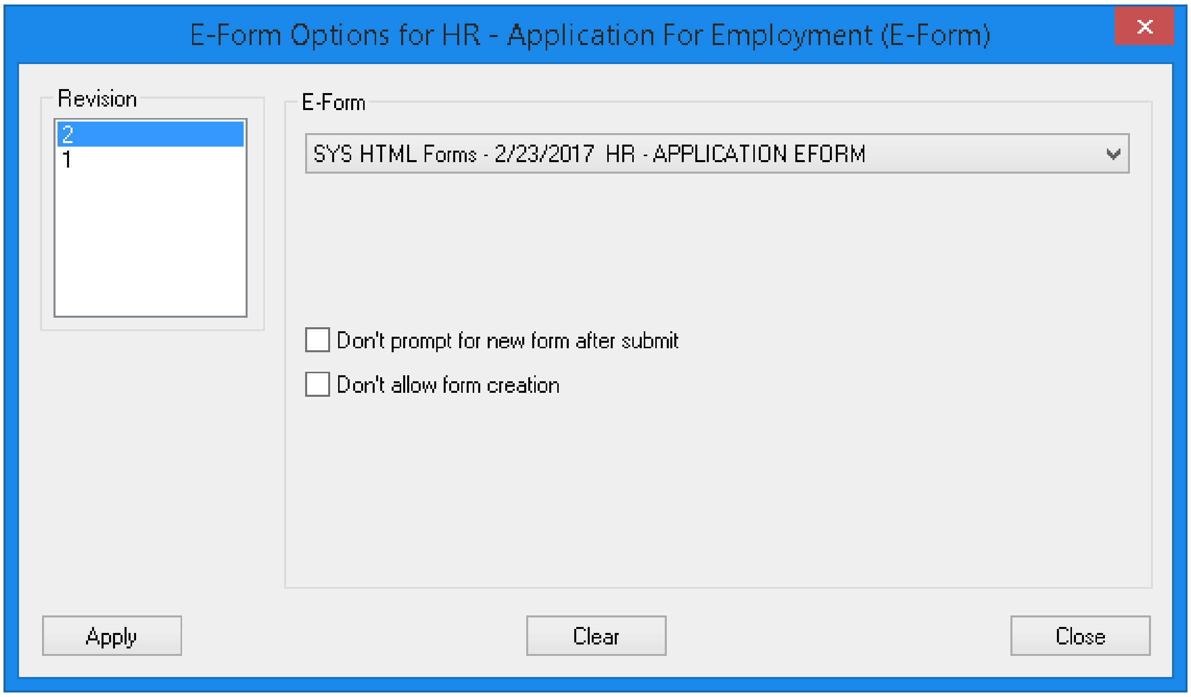
<xsl:include href=”FileName.xsl”/>

This tag informs the XSLT parser to also parse in any XSL content from the specified external file. In order to avoid generating an exception, all content from any referenced.xsl file needs to be copied into the main style sheet .xsl file used for viewing and the <xsl:include> tag needs to be removed. Care must be taken to transfer any <xsl:includes> references embedded within other referenced .xsl files to the main file as well.

It is recommended to not use <xsl:include> in .xsl files for improved security. While it is not recommended to reference external .xsl files, if your solution requires it, you can override this security feature by using the **AllowInsecureExternalXsl** key and set it to **true**. This key must be placed in the Application Server's web.config configuration file under the **appSettings** node.

<add key="AllowInsecureExternalXsl" value="true" />

Once you have created a Document Type revision, clicking the **E-Form** button will display multiple revisions, like the following example:


E-Form Options for HR - Application For Employment (E-Form)
Revision
2
1
E-Form
SYS HTML Forms - 2/23/2017 HR - APPLICATION EFORM
Don't prompt for new form after submit
Don't allow form creation
Apply
Clear
Close

You can select a revision from the **Revision** box and select an **E-Form** from the drop-down list to correspond to the revision. Once an e-form is selected, click **Apply**.

E-Forms and Revision Considerations

When using document revisions with E-Forms, keyword values can be changed in various ways throughout the software. When keyword values are changed other than by viewing and editing the form, these changes may not be reflected in revisions of the forms.

Specifically, when forms are created or modified from the Unity Briefcase, or created or revised through the API, the keyword values specified as part of those revisions will not be available after subsequent revisions have been created.

Additionally, the dynamic nature of forms may also affect what is shown on a form at the time a revision is created compared to what may be shown when a form revision is later retrieved. For example, if a form is using an external data source to populate a field and the external data has changed between when a revision is created and when it is viewed, the revision will display this updated external data instead of presenting the value shown when the revision was created.

Disabling Specific E-Forms

In some cases you may need to disable an E-Form so users cannot create new E-Forms of a specific type. One of these situations is when you convert an existing E-Form to a Unity Form. Once a Unity

Form is configured to replace an E-Form, you would want to disable the E-Form equivalent. To disable an E-Form:

1. In the Configuration module, select **Document | Document Types**. The **Document Types** dialog box displays.
2. Select the Document Type that is associated with the E-Form you want to disable.
3. Click **E-Form**.
4. Select the **Don't allow form creation** option. Selecting this option disables all revisions for an E-Form; you do not have to apply this option for each revision.
5. Click **Apply**.
6. Click **Close**.

Document Type XML Configuration

Style sheets are used with XML documents to define the appearance of the data in the document. The purpose of the style sheet is to allow a single XML document to be re-targeted for different usages and audiences.

Within OnBase, you can assign different style sheets to a Document Type to change the appearance of the document depending on how its being used or viewed. These options are located in the **XML Style Sheet Settings** dialog box.

If the **Keyword-Based XML** check box is enabled in the **Document Type Settings** dialog box, OnBase automatically selects a style sheet for the document based on a specified Keyword Value.

When creating an XML style sheet there are few things that are important to note:

- The style sheet must be accessible to the user and the workstation that the document is being viewed from via a UNC or URL path.
- If the style sheet includes images, the style sheet must include the full path (via a UNC or URL) to the image file.

  If the document is intended to be distributed via e-mail, images in the style sheet must be available via a URL.
- If a style sheet is referenced in the XML file and it does not match the style sheet configured in OnBase, the style sheet specified in OnBase overrides the style sheet specified in the XML file.
- If viewing the XML document in the OnBase Web Client, the style sheet must use UTF-8 encoding.

  Note: It is recommended that the following line be placed below the XML declaration in the style sheet to ensure proper viewing: `<!DOCTYPE xsl:stylesheet [ <!ENTITY nbsp "&#160;"> ]>` An example of an XML declaration is: `<?xml version="1.0" encoding="UTF-8" ?>`
- Stylesheets that use an **<xsl:include>** tag are not supported by default, in order to ensure the highest level of security. While it is not recommended to reference external .xsl files, if your solution requires it, you can override this security feature by using the **AllowInsecureExternalXsl** key and set it to **true**. This key must be placed in the Application Server's web.config configuration file under the **appSettings** node.

  ```
  <add key="AllowInsecureExternalXsl" value="true" />
  ```

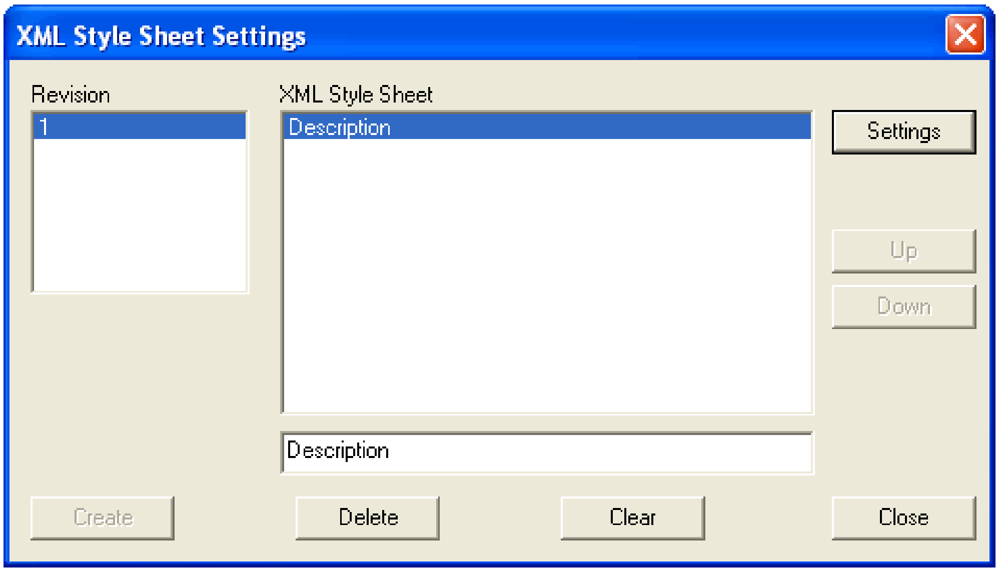
Associating XML Style Sheets at the Document Type Level

You can associate multiple style sheets with a single Document Type, providing the you with the ability to select the style sheet they want to use when viewing, printing or e-mailing the document.

To associate a style sheet with a Document Type:

1. From OnBase Configuration, select **Document | Document Types**.
   The **Document Types** dialog box is displayed.
2. Select the Document Type you would like to configure the style sheet for from the Document Type list and click **XML Style Sheet**.

   The **XML Style Sheet Settings** dialog box is displayed.



Note: The **XML Style Sheet Settings** dialog box will include fields for Keyword configuration if you selected **Keyword-Based XML** in **Document Type Settings**. If this option was selected, the style sheet used when viewing, printing and e-mailing documents is based on the document's Keyword Values. See Associating XML Style Sheets by Keyword Value on page 48 for more information on how to configure these style sheets.

A Document Type that is associated with XML documents should have at least one style sheet configured for it. Each style sheet has the ability to associate two **.XSL** files with it:

- One used when viewing documents.
- One used when printing documents.

Note: Depending on your configuration, you may have the option to select an alternate style sheet from an open document.

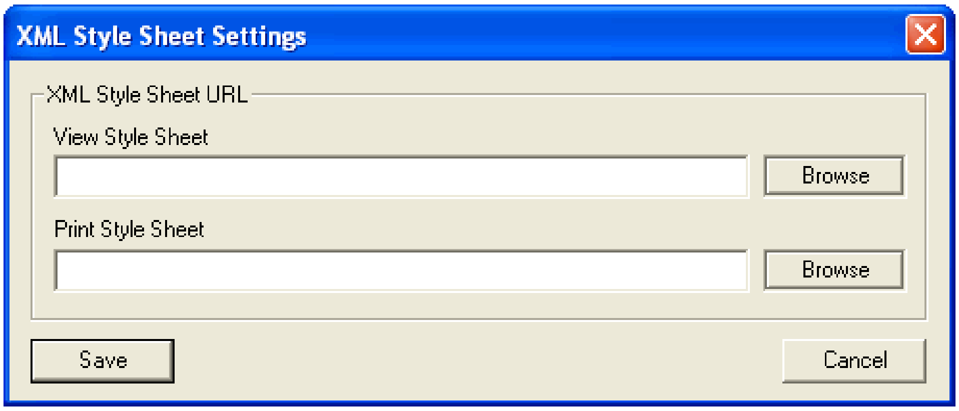
3. In the **Revision** list, select the Document Type's revision number that is to be associated with this style sheet. If revisions are not configured for the Document Type, select **1**.

Tip: OnBase allows you to assign different style sheets to different revisions of the document to account for changes in the appearance of the document, such as if a style sheet is modified or updated.

If a user attempts to view a revision of the Document Type that has not been assigned a style sheet, OnBase uses the style sheet assigned to latest previous revision that was assigned a style sheet.

For example, if you are viewing the 5th revision of a document, but this revision has not been assigned a style sheet, OnBase will check the 4th revision for a style sheet to use. If the 4th revision is not assigned a style sheet, OnBase will check the 3rd revision, etc. until it finds a revision with an assigned style sheet. If no style sheet is associated with any revision of the Document Type, the raw XML data is displayed.

4. Enter the name of the new style sheet in the **New Style Sheet** data entry field.
5. Click **Create**.
   The **XML Style Sheet Settings** dialog box with **XML Style Sheet URL** settings is displayed.



6. In the **View Style Sheet** field, browse to or enter the full path to the .XSL file that is to be associated with the document when it is viewed. This path must be a UNC or URL to which the user has access from the workstation the document is being viewed from.
7. In the **Print Style Sheet** field, browse to or enter the full path of the .XSL file that is to be associated with the document when it is printed. This path must be a UNC or URL to which the user has access from the workstation the document is being viewed from.
8. Click **Save**. The **XML Style Sheet Settings** dialog box is closed.

   The style sheet is added to the XML Style Sheet list in the **XML Style Sheet Settings** dialog box.

   To disassociate a style sheet with a Document Type, select it in the XML Style Sheet list and click **Delete**.
9. Repeat steps 3 through 8 for each style sheet you want to create.

   Note: The first style sheet in the XML Style Sheet list is the default style sheet.
10. When you are finished configuring style sheets, click **Close**.

Associating XML Style Sheets by Keyword Value

OnBase can assign a style sheet to your document based on a Keyword Value assigned to the document.

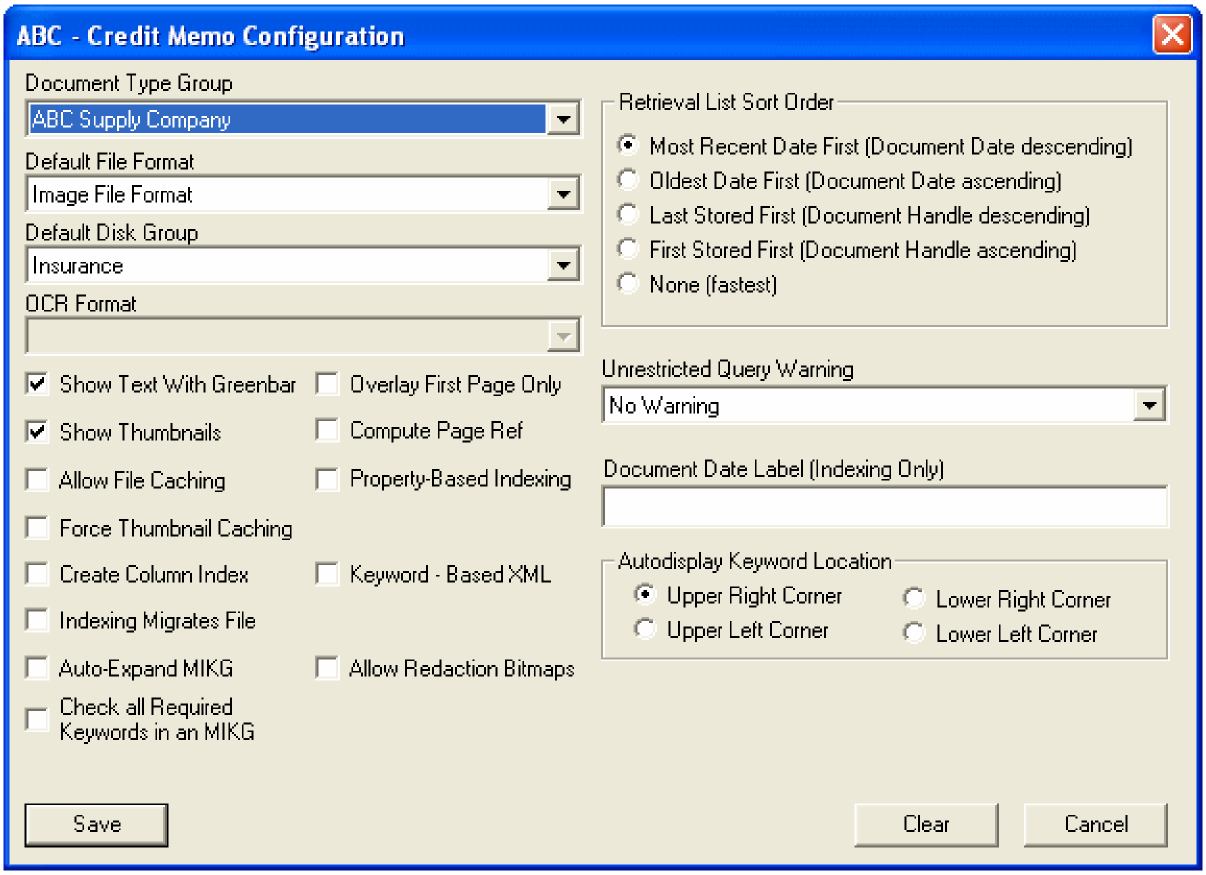
For example, an Accounts Receivable department prints and mails color-coded invoices depending on how past due an invoice is. To enable this, they added a **Status** Keyword Type with four different possible Keyword Values (**Current**, **30 Days Past Due**, **60 Past Due**, **90 Days Past Due** in this example) and configured the Document Type to use a different style sheet depending on the **Status** Keyword Value. The style sheet assigned to the document controls the color it is printed in.

To associate a style sheet with a document based on a Keyword Value:

1. From OnBase Configuration, click **Document | Document Types**.
   The **Document Types** dialog box is displayed.
2. Select the Document Type to be configured from the Document Type list and click **Settings**.

   Tip: Use the Document Type Group drop-down to filter the Document Types list to only contain the Document Types associated with the selected Document Type Group.

   The **[Document Type Name] Configuration** dialog box is displayed, where **[Document Type Name]** is the name of the selected Document Type.



ABC - Credit Memo Configuration

Document Type Group: ABC Supply Company
Default File Format: Image File Format
Default Disk Group: Insurance
OCR Format:

[✓] Show Text With Greenbar
[✓] Show Thumbnails
[ ] Allow File Caching
[ ] Force Thumbnail Caching
[ ] Create Column Index
[ ] Indexing Migrates File
[ ] Auto-Expand MIKG
[ ] Check all Required Keywords in an MIKG
[ ] Overlay First Page Only
[ ] Compute Page Ref
[ ] Property-Based Indexing
[ ] Keyword - Based XML
[ ] Allow Redaction Bitmaps

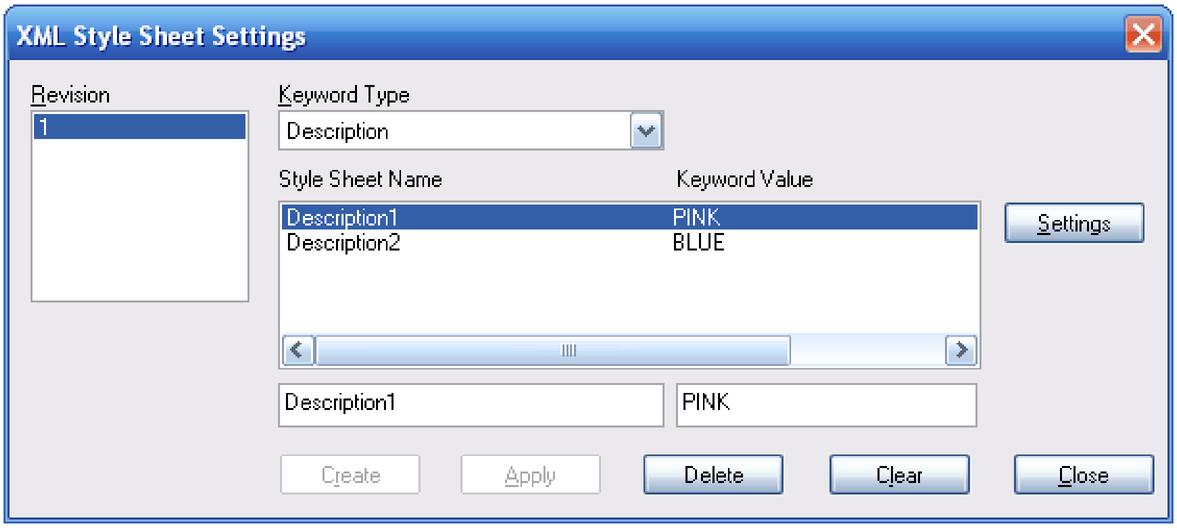
Retrieval List Sort Order
(●) Most Recent Date First (Document Date descending)
( ) Oldest Date First (Document Date ascending)
( ) Last Stored First (Document Handle descending)
( ) First Stored First (Document Handle ascending)
( ) None (fastest)

Unrestricted Query Warning: No Warning
Document Date Label (Indexing Only):

Autodisplay Keyword Location
(●) Upper Right Corner
( ) Upper Left Corner
( ) Lower Right Corner
( ) Lower Left Corner

3. Select the **Keyword-Based XML** check box and click **Save**.
   The **[Document Type Name] Configuration** dialog box is closed and you are returned to the **Document Types** dialog box.
4. With the Document Type to be configured still selected in the Document Type list, click **XML Style Sheet**.

The **XML Style Sheet Settings** dialog box is displayed.



Style Sheet Name	Keyword Value
Description1	PINK
Description2	BLUE

Description1 PINK

Note: If the **Keyword-Based XML** check box was not selected for the Document Type, the **XML Style Sheet Settings** dialog box does not contain fields for Keyword configuration and the style sheet applied to the document is determined by the Document Type. See Associating XML Style Sheets at the Document Type Level on page 47 for more information on how to configure these style sheets.

A Document Type that is associated with XML documents should have at least one style sheet configured for it. Each style sheet has the ability to associate two **.XSL** files with it:

- One used when viewing documents.
- One used when printing documents.

Note: Depending on your configuration, you may have the option to select an alternate style sheet from an open document.

5. In the Revision list, select the Document Type's revision number that is to be associated with this style sheet. If revisions are not configured for the Document Type, select **1**.

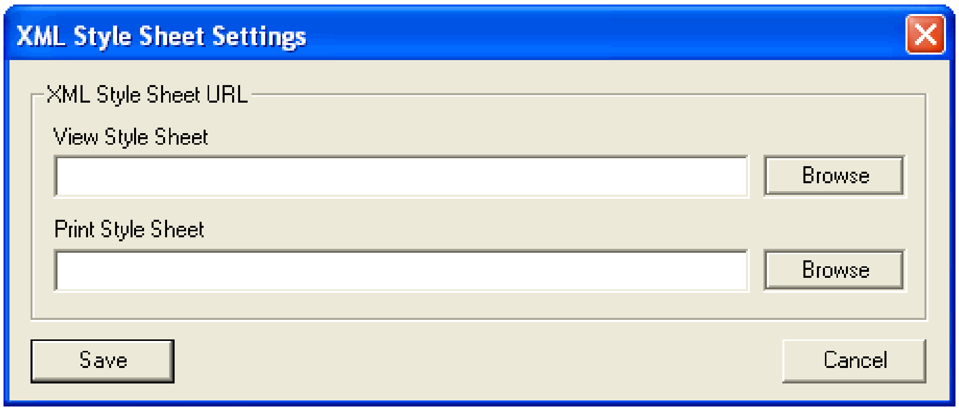
Tip: OnBase allows you to assign different style sheets to different revisions of the document to account for changes in the appearance of the document, such as if a style sheet is modified or updated.

If a user attempts to view a revision of the Document Type that has not been assigned a style sheet, OnBase uses the style sheet assigned to latest previous revision that was assigned a style sheet.

For example, if you are viewing the 5th revision of a document, but this revision has not been assigned a style sheet, OnBase will check the 4th revision for a style sheet to use. If the 4th revision is not assigned a style sheet, OnBase will check the 3rd revision, etc. until it finds a revision with an assigned style sheet. If no style sheet is associated with any revision of the Document Type, the raw XML data is displayed.

6. Using the **Keyword Type** drop-down, select the Keyword Type that is to control the style sheet that is to be applied to the document.

7. In the **Keyword Value** field, enter the Keyword Value that is to be associated with a style sheet.
8. In the **Style Sheet Name** field, enter a name for the new style sheet.
9. Click **Create**.
   The **XML Style Sheet Settings** dialog box with **XML Style Sheet URL** settings is displayed.



10. In the **View Style Sheet** field, browse to or enter or the full path to the .XSL file that is to be associated with the document when it is viewed. This path must be a UNC or URL to which the user has access from the workstation the document is being viewed from.
11. In the **Print Style Sheet** field, browse to or enter the full path of the .XSL file that is to be associated with the document when it is printed. This path must be a UNC or URL to which the user has access from the workstation the document is being viewed from.
12. Click **Save**. The **XML Style Sheet Settings** dialog box is closed.

    The style sheet is added to the XML Style Sheet list in the **XML Style Sheet Settings** dialog box.

    To disassociate a style sheet with a Document Type, select it in the XML Style Sheet list and click **Delete**.
13. Repeat steps 7 through 12 for each style sheet you want to configure.

    You can add as many Keyword Value/style sheet associations as you wish, but they must all be Keyword Values of the same Keyword Type. When configuring style sheets, you can only associate one Keyword Type for each Document Type.
14. When you are finished configuring style sheets, click **Close**.

Editing an XML Style Sheet Associated by Keyword Type

If you need to edit the Keyword Type or Keyword Value selected for the style sheet configuration.

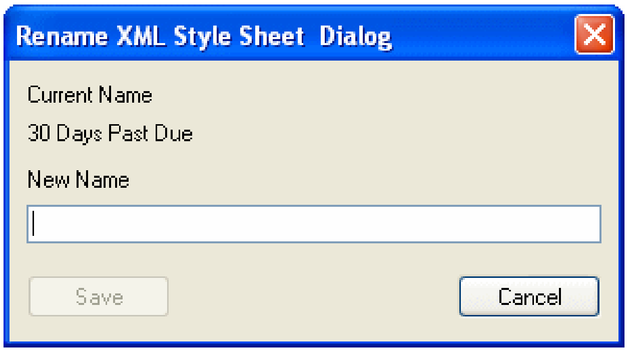
1. Select the Style Sheet/Keyword Value association from the Style Sheet Name/Keyword Value list.
2. Use the **Keyword Type** drop-down to change the assigned Keyword Type or enter a new Keyword Value in the **Keyword Value** field.
3. Click **Apply**.
4. Click **Close**.

Renaming an XML Style Sheet

If you would like to give an existing XML style sheet a different name, you may do so from the **XML Style Sheet Settings**.

To rename an XML style sheet:

1. In OnBase Configuration, select **Document | Document Types**.
   The **Document Types** dialog box is displayed.
2. Select the Document Type associated with the style sheet you would like to rename from the Document Type list and click **XML Style Sheet**.
   The **XML Style Sheet Settings** dialog box is displayed.
3. Double-click the style sheet to be renamed.
   The **Rename XML Style Sheet Dialog** dialog box is displayed.



4. Enter the new name for the style sheet.
5. Click **Save**.

Using Hidden Fields on Forms

You can hide fields on a form. To enter hidden fields on a form, use the following syntax:

```
<INPUT type="hidden">
```

Example:

```
<input type="hidden" name="OBKey_First_Name_3" size="40" value="Name">
```

This syntax will only hide the field on the form. If the field is associated with a Keyword Type, values will still be visible in the Keyword Panel unless the Keyword Type is configured with the HID option selected. For more information on how to configure this setting, see the System Administration documentation in the Configuring Document Type Keywords topic.

Virtual E-Forms

Virtual E-Forms are similar to E-Forms in setup and use; however Virtual E-Forms store only information that is saved in the database, such as Keyword Values and system values. For example, a Virtual E-Form that includes a scrolling text box not mapped to a Keyword Value or system property will not save information entered into the text box.

Configuring Virtual E-Forms is the same as configuring E-Forms. The only difference in configuration is that the Document Type used for a virtual form should have a file format of **Virtual Electronic Form**.

Tip: To strip out non-essential data from forms before storage, re-configure the original E-Form Document Type to use a default file format of Virtual Electronic Form.

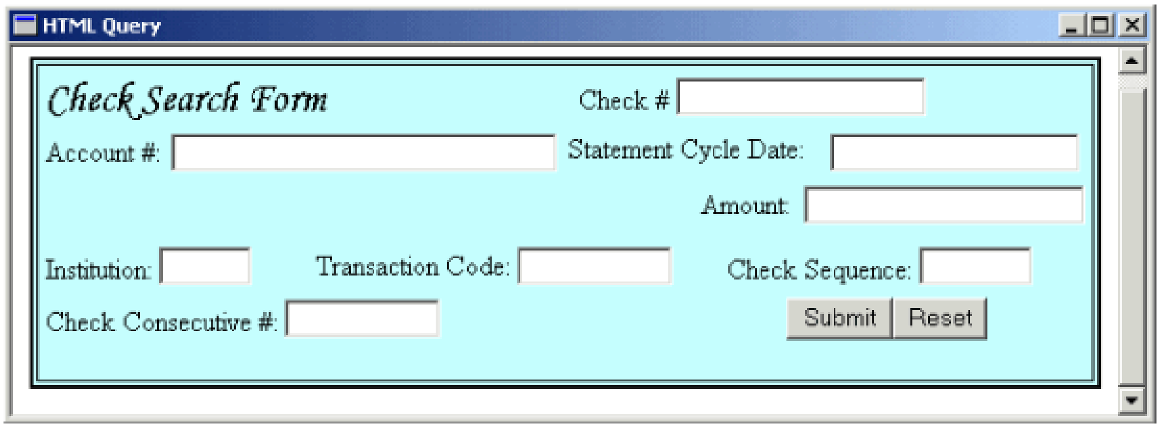
CAUTION: Re-configuring a form will cause all non-database information stored on the form to be lost.

Note: Virtual E-Forms are not revisable.

Configuring HTML Forms for Custom Queries

HTML forms can be used in conjunction with Custom Queries to provide a simple interface for document and folder retrieval.

For example, you may wish to design an HTML form Custom Query that resembles a check, as shown in the example below.



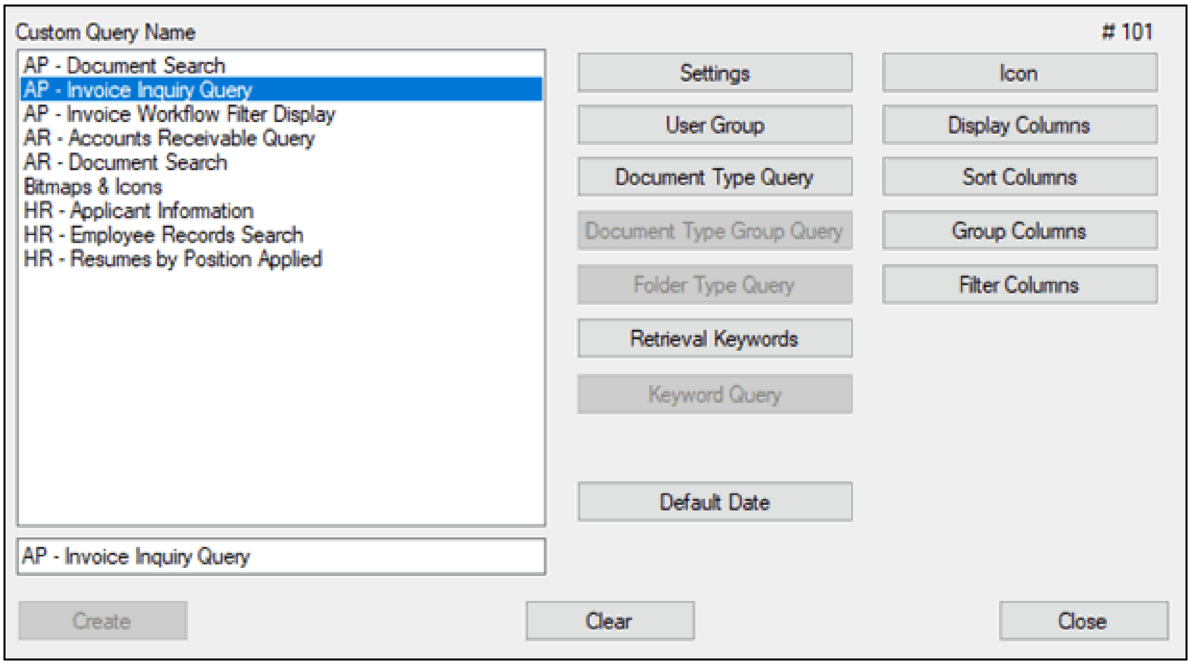
Before you begin configuring an HTML Custom Query, please note the following requirements and limitations

- Create and configure HTML forms before attempting to configure Custom Queries for use with them. For more information on creating and configuring forms, see the **E-Forms** documentation.
- Any images that are included in HTML forms used as Custom Queries must be referenced with an absolute path that is accessible to all users who need to use the form. An absolute path is the full path to a file, beginning with the root directory.
- If required, data validation should be built into the HTML form.
- If you are using Distributed Disk Services, see the Distributed Disk Services documentation for more information about constructing appropriate paths.
- Remember to include Submit and Reset buttons in order to submit your query or reset the form to enter different information.

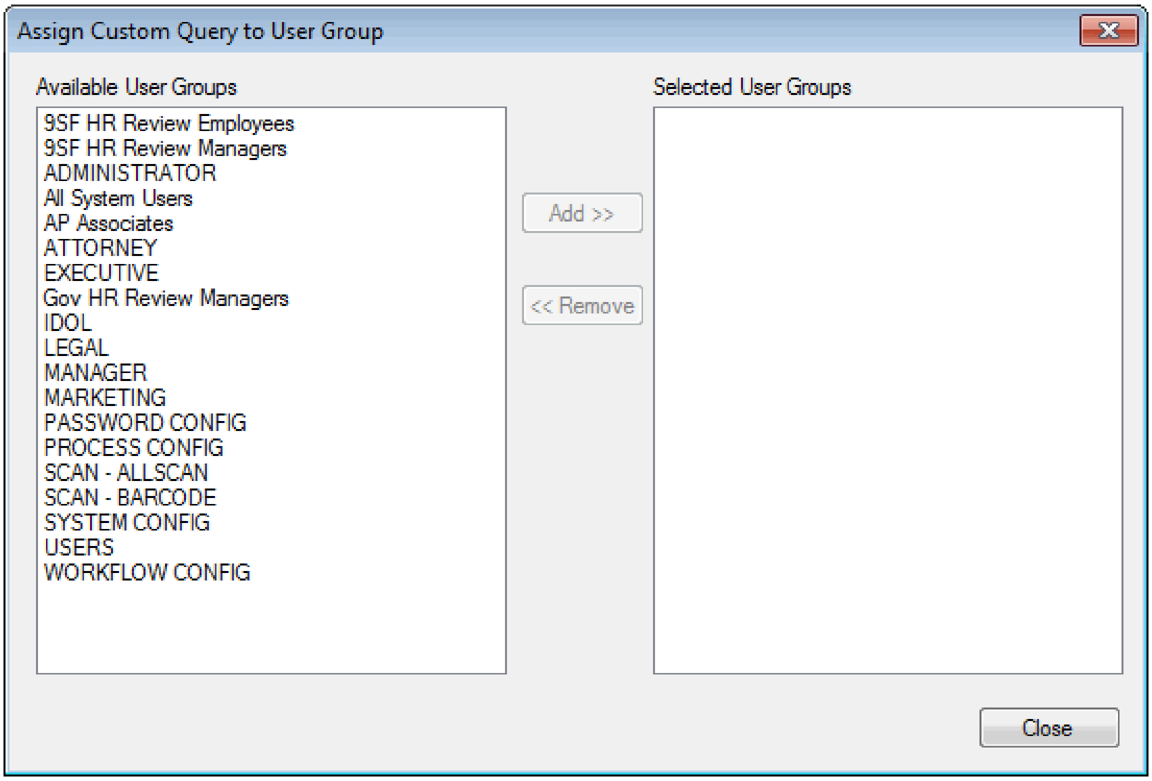
To configure an HTML Custom Query:

1. In OnBase Configuration, select **Queries | Custom Queries**.

The **Custom Query** dialog box is displayed.



2. Enter a name for the Custom Query in the field below the **Custom Query Name** list.
3. Click **Create**.
   The **Assign Custom Query to User Group** dialog box is displayed.

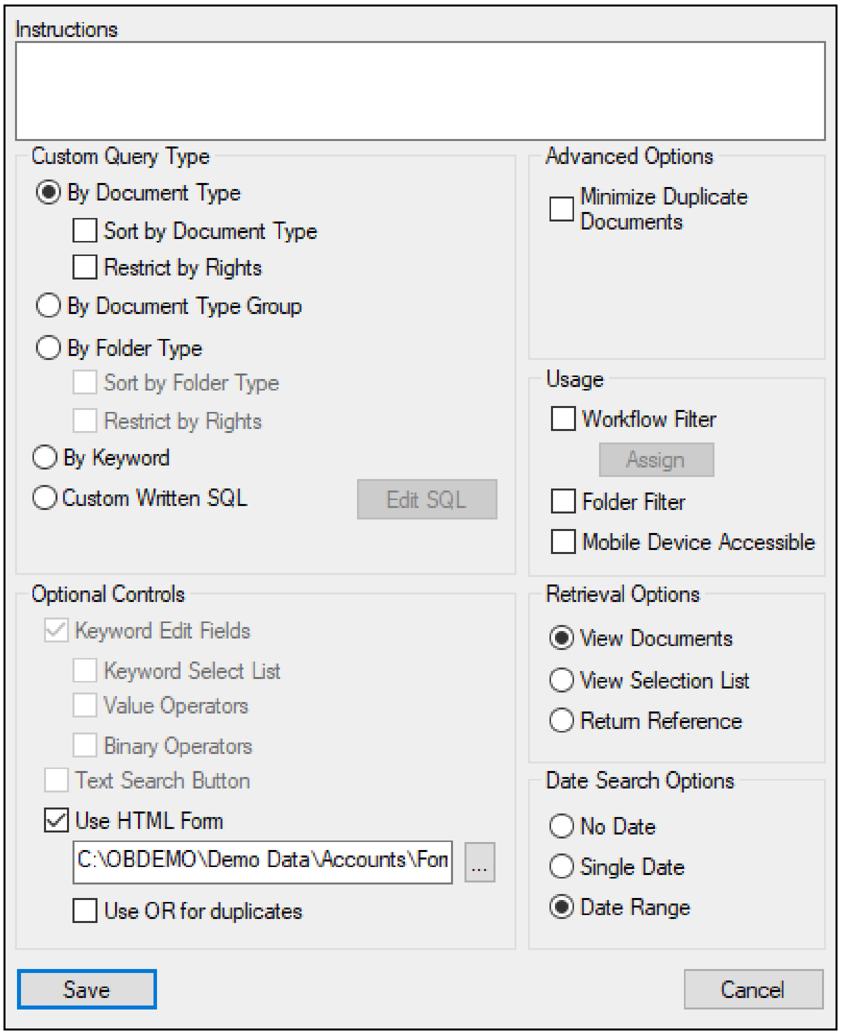


4. Assign the Custom Query to User Group(s).

   To assign the Custom Query to one or more User Groups, select the appropriate User Group(s) from the **Available User Groups** list and click **Add**.

   To remove the Custom Query from one or more User Groups, select the appropriate User Group(s) from the **Selected User Groups** list and click **Remove**.

5. Click **Close**.
   The assigned User Groups are saved, and the **Custom Query Options** dialog box is displayed.



Instructions

Custom Query Type
- [● By Document Type]
  - ☐ Sort by Document Type
  - ☐ Restrict by Rights
- ○ By Document Type Group
- ○ By Folder Type
  - ☐ Sort by Folder Type
  - ☐ Restrict by Rights
- ○ By Keyword
- ○ Custom Written SQL

Advanced Options
- ☐ Minimize Duplicate Documents

Usage
- ☐ Workflow Filter
- ☐ Folder Filter
- ☐ Mobile Device Accessible

Optional Controls
- [✓ Keyword Edit Fields]
  - ☐ Keyword Select List
  - ☐ Value Operators
  - ☐ Binary Operators
- ☐ Text Search Button
- [✓ Use HTML Form]
  - C:\OBDEMO\Demo Data\Accounts\Fon
  - ☐ Use OR for duplicates

Retrieval Options
- [● View Documents]
- ○ View Selection List
- ○ Return Reference

Date Search Options
- ○ No Date
- ○ Single Date
- [● Date Range]

6. Select the Custom Query Type.

   For example, if your form is set up to use Keyword Types to retrieve documents of any Document Type, select **By Keyword**.

7. Select **Use HTML Form**.
8. Enter the file location of the HTML Form file by doing one of the following:
   - Enter the full path to the HTML Form file in the field under the **Use HTML Form** setting.
   - Browse to the HTML Form file by clicking the **Browse** button (...).

9. Select the file and click **Open**.

   Note: The form file must be in a network directory available to all users who will be accessing the form, with the exception of users accessing the form through the Web Client. If users are accessing the form through the Web Client, the Web Services Manager must have access to all network Disk Groups that store files to be called from the Web Client.

10. Select **Use OR for duplicates** if there are duplicate Keyword Type fields on the HTML form and you want each Keyword Type value to be searched for separately, as if the values were joined by the **OR** logical operator. For example, if an HTML form were configured with two Amount fields, a user could enter two different amounts and retrieve documents indexed with either amount.

    If this option is not selected, documents are retrieved only if they are indexed with each Keyword Value entered by the user. Values entered in duplicate Keyword Type fields will be treated as if they were joined by the **AND** logical operator.

11. Click **Save**.
12. Continue configuration by clicking **Document Type Query**, **Document Type Group Query**, **Retrieval Keywords**, or **Keyword Query**, depending on the Custom Query type you selected in step 6.

    Note: You cannot configure an HTML Custom Query to automatically populate a configured default date.

Configuring Auto Number Keyword Types

Auto Number Keyword Types are numeric Keyword Types configured to help track the creation of E-Forms and Unity Forms in the system. They are unique identifiers that cannot be edited or modified during or after form creation.

Note: Auto Number Keyword Types are only supported with E-Forms and Unity Forms in the Unity Client, Web Client, and in Workflow. All other file types are not supported and may result in unexpected behavior or errors.

Note: The Auto Number Keyword Type will not increment when creating an E-Form if the Auto Number Keyword Type is not configured with the **OBKey_** tag.

When a new E-Form or Unity Form using an Auto Number Keyword Type is created in OnBase, the value of the Auto Number Keyword Type is incremented by one. If during the process of form creation, the user decides to cancel or close the form without submitting it, the Auto Number Keyword Type value is still incremented. Even though the last form was not submitted, the next form created increments the Auto Number Keyword Type value. For example, if you submit an E-Form, cancel the submission of a second E-form, and submit a third E-Form, the third E-form's Auto Number Keyword Type value is incremented 2 from the first submitted E-Form.

Since users may cancel documents during creation, the Auto Number Keyword Type value should not be used to determine an accurate count of documents within a specific Document Type. For a more accurate count of submitted form documents into the system, a Workflow solution could be created to capture forms of the specific type on submit using a numeric Keyword Type (given an initial value and used in the Workflow module as a counter) and the Workflow action **Key – Increment Keyword on This Document**. The value of the counter can be used to trigger other Workflow actions.

Note: Auto Number Keywords can only be of a Numeric 9 or Numeric 20 value.

Configuring an Auto Number Keyword Type

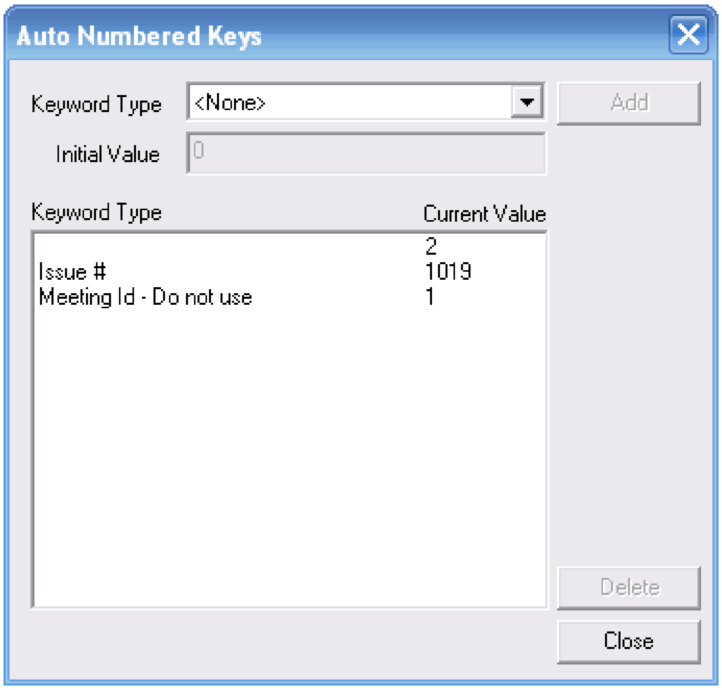
Auto Number Keyword Types are created from numeric Keyword Types already existing in the system.

Prerequisite: Before configuring an Auto Number Keyword Type, create or identify the Keyword Type you would like to use. This Keyword Type must then be part of a Document Type using E-Forms or Unity Forms. All other file types are not supported and may result in unexpected behavior or errors.

Note: In order to view a hidden Auto Number Keyword Type as part of a Document Type in the Web Client, it needs to be marked as read-only.

To configure an Auto Number Keyword Type:

1. In OnBase Configuration, select **Keyword | Auto Number Keywords Types**.
   The **Auto Numbered Keys** dialog box is displayed.



Keyword Type	Current Value
	2
Issue #	1019
Meeting Id - Do not use	1

2. Select the counter Keyword Type from the **Keyword Type** drop-down menu.
3. Enter a beginning counter number into the **Initial Value** field.

   Note: The **Initial Value** field allows up to nine digits.
4. Click **Add** to move the Keyword Type and value to the **Keyword Type/Current Value** list.

   Note: To delete an Auto Number Keyword Type, select the Keyword Type in the list and click the **Delete** button. This will add the Keyword Type back to the **Keyword Type** drop-down list.

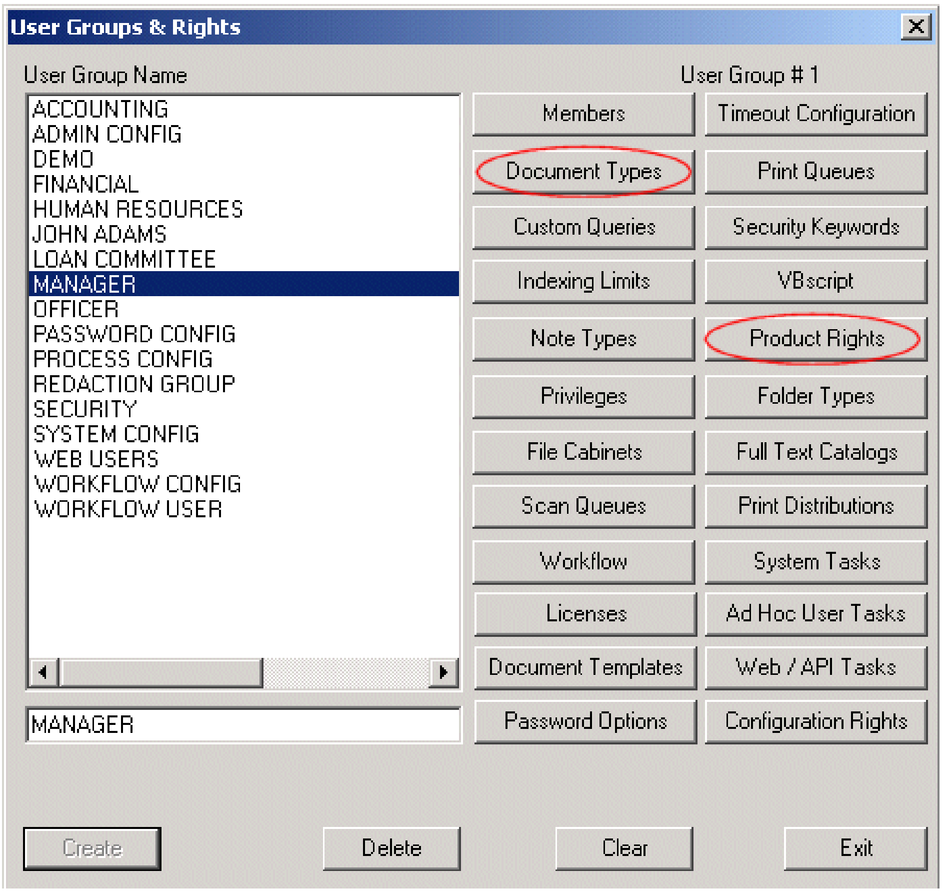
Each time you attempt to create a new E-Form or Unity Form using an Auto Number Keyword Type, the Auto Number Keyword Type value increments by one.

User Groups & Rights

In order to create and fill out forms, users must have product rights to **HTML Forms**. To assign product rights, in OnBase Configuration, select **Users | User Groups/Rights**, select the **User Group Name**, click **Product Rights**, and select the **HTML Forms** check box.

Assign the appropriate rights to the Document Type for the E-Form. In OnBase Configuration, select **Users | User Groups/Rights**. Select the appropriate **User Group** name, and click **Document Types**.

The **User Groups & Rights** dialog box is displayed in OnBase Configuration by selecting **User Groups/Rights** from the **Users** menu.



ACCOUNTING
ADMIN CONFIG
DEMO
FINANCIAL
HUMAN RESOURCES
JOHN ADAMS
LOAN COMMITTEE
MANAGER
OFFICER
PASSWORD CONFIG
PROCESS CONFIG
REDACTION GROUP
SECURITY
SYSTEM CONFIG
WEB USERS
WORKFLOW CONFIG
WORKFLOW USER

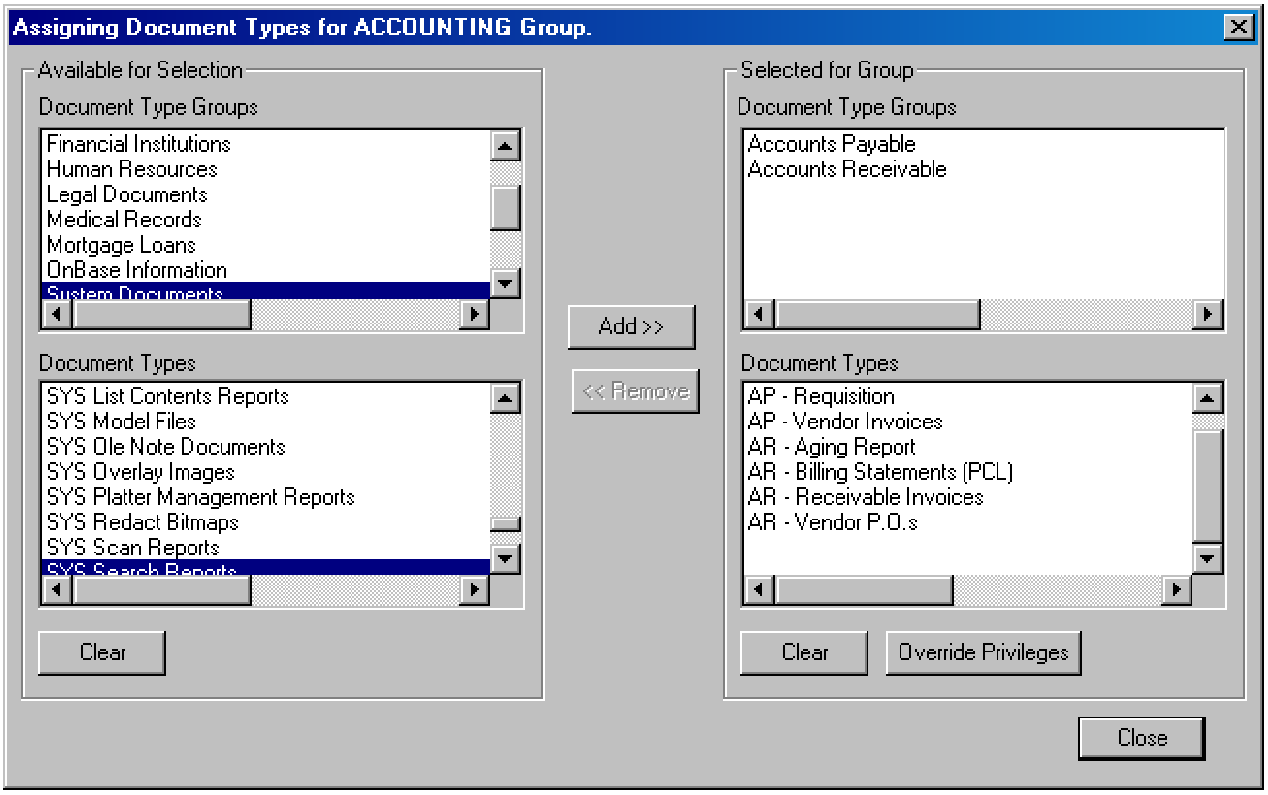
MANAGER

Privileges

In order to create documents, including E-Forms, users must have the document Create privilege. In the Configuration module, select **Users | User Group /Rights**. Select the appropriate user group and click **Privileges**. Select the **Create** privilege listed under **Documents.**

Document Types

In order to process documents as specific Document Types, as well as view the documents that have been processed into the OnBase system, the user group must have rights to the Document Types. This is done under the **Document Types** button. Highlight the Document Types or groups on the left and click **Add** to select.


Assigning Document Types for ACCOUNTING Group.

Available for Selection

Document Type Groups
Financial Institutions
Human Resources
Legal Documents
Medical Records
Mortgage Loans
OnBase Information
System Documents

Document Types
SYS List Contents Reports
SYS Model Files
SYS Ole Note Documents
SYS Overlay Images
SYS Platter Management Reports
SYS Redact Bitmaps
SYS Scan Reports
SYS Search Reports

Selected for Group

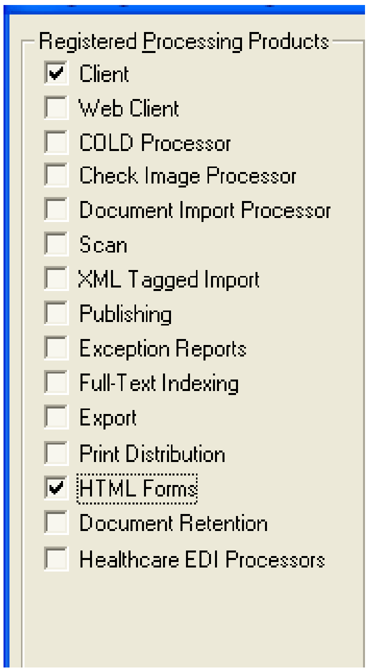
Document Type Groups
Accounts Payable
Accounts Receivable

Document Types
AP - Requisition
AP - Vendor Invoices
AR - Aging Report
AR - Billing Statements (PCL)
AR - Receivable Invoices
AR - Vendor P.O.s

Product Rights

To configure and submit E-Forms, users must belong to a User Group that has Product Rights to the Client module and HTML forms.

Highlight the User Group, click the **Product Rights** button, and select the **Client** and **HTML Forms** options under **Registered Processing Products**.



Registered Processing Products

- [✓] Client
- [ ] Web Client
- [ ] COLD Processor
- [ ] Check Image Processor
- [ ] Document Import Processor
- [ ] Scan
- [ ] XML Tagged Import
- [ ] Publishing
- [ ] Exception Reports
- [ ] Full-Text Indexing
- [ ] Export
- [ ] Print Distribution
- [✓] HTML Forms
- [ ] Document Retention
- [ ] Healthcare EDI Processors

Encoding

Text Encoding for HTML Files

OnBase supports ANSI, UTF-8, and UTF-16 text encodings for HTML files. When HTML files are imported into OnBase, the type of encoding being used in the files is determined by one of the following factors:

- Byte Order Mark (BOM) specified in the HTML file
- Value assigned to the **charset** tag in the HTML file
- Default encoding for the HTML file as determined by the assigned file type:
  - If the file type is **HTML**, the default encoding is set to the local ANSI code page.
  - If the file type is **HTML Unicode**, the default encoding is set to UTF-8.

OnBase checks for these factors in the above order of precedence. If neither a BOM nor a value for the **charset** tag is specified in the HTML file, a default encoding is assigned.

Tip: When creating an HTML file with a Unicode encoding, it is considered a best practice to specify a BOM. This will help to ensure that OnBase correctly detects the encoding on import.

HTML File Types

If file types have not already been assigned to HTML files when they are imported into OnBase, the file types are automatically assigned according to the type of text encoding and the type of database (ANSI or Unicode) being used:

- If an HTML file with UTF-8 or UTF-16 encoding is being imported into OnBase, the file type is automatically set to **HTML Unicode**.
- If an HTML file with ANSI encoding is being imported into OnBase, the file type is automatically set to HTML.

HTML5 Support

The following is supported for HTML5 in the Web Client and Unity Client: `<meta charset="character_set">`

The following <input> elements are supported:

- Type=”telephone”
- Type=”email”
- Type=”URL”
- Type=”password”

Note: This is not supported in the OnBase Client.

System Interaction

Digital Signatures

Signing E-Forms

You can sign E-Forms, Virtual E-Forms, and HTML documents in the same way you would sign an image or text document in the OnBase Client.

Signing an E-Forms, Virtual E-Forms, or HTML document makes it read-only. Changes can no longer be made once it is signed. This includes the disabling of any button or field that could be used to update the form. Text box fields are made read-only instead of being disabled. All Submit buttons will still be active after signing.

Document Knowledge Transfer

E-Forms can be added to a Reading Group in Document Knowledge Transfer.

Image Document Composition

When used in conjunction with Image Document Composition, E-Forms should not be configured to use embedded scripting that utilizes XML libraries. Instead, gather the needed data from the E-Form, then use a script field within Image Document Composition to submit the request.

Office Business Application and Integration for Microsoft Outlook

There is a **Forms** button in the **Create** ribbon group that allows you to create a new form.

Note: The **Save and Close** button is not available on E-Forms when using the Integration for Microsoft Outlook.

CAUTION: Closing the Microsoft Outlook application will not display a prompt to save your work, and any unsaved work may be lost.

WorkView

Forms can be created in WorkView. Forms can be create directly from WorkView objects and populate applicable data in the form from the WorkView object. A form must be configured as any other form must be created. Once the form is created and available in the OnBase system, it must be mapped to a WorkView class within the WorkView Configuration tool. See the WorkView documentation for information about Associating Forms with Classes.

E-Forms Best Practices Overview

The following best practice recommendations were assembled by a team of OnBase subject matter experts. They represent the accumulation of years of experience installing and configuring OnBase solutions.

The following recommendations are general in nature, and are applicable to most OnBase solutions and network environments. Depending on your solution design and your organization's needs, not all of the best practice recommendations listed below may apply to, or be recommended for, your OnBase solution.

Carefully consider the impact of making any changes, including those listed below, to your OnBase solution prior to implementing them in a production environment.

This section describes some best practices for the design and implementation of E-Forms. Best practices have been divided into the following categories:

- General
- Keywords
- Scripting
- Workflow Interaction

General

Perform data validation in E-Forms.

Keep the forms as simple as possible. Adding too much extra functionality into an E-Form can potentially cause performance issues.

Re-use code and configurations that have been successfully implemented in the past.

Add comments to your E-Form code to make changes easier for the system administrator if changes need to be made in the future. In the case where this form becomes part of a support issue, comments in the scripts will help in troubleshooting the issue to see whether the issue is OnBase related or if it is caused by the scripts.

Document what the form is supposed to do and how it should be maintained. This is in addition to adding comments to the pertinent sections of the E-Form code.

You can use a text editor, like Notepad®, to edit and save the revised file as plain text if you are knowledgeable in HTML. HTML editors, like Microsoft Expression Web, are also useful to catch common mistakes and ensure the creation of standards compliant HTML.

Validate any HTML created by using a validation service. The Markup Validation Service from W3C is a useful tool. The service can be found at https://validator.w3.org/

Some HTML editors store extra spaces as **&nbsp;** which can cause the maximum length of the Keyword Value to be exceeded or produce unexpected search results. It is best practice to not append extra spaces to drop-down Keyword Value select lists. Check the HTML code to confirm that spaces are stored in a way that is compatible with the system database.

It is a best practice to thoroughly test all E-Forms in the main browsers the user base anticipates using. Each implementation will have unique circumstances, so choices for scripting languages and functions used should be based on those circumstances.

When importing the HTML E-Form template into the system, it is considered a best practice to include the name of the Document Type that will use the form in the Description keyword value.

Limit the number of fields on an E-Form. E-Forms are a very powerful tool which makes it easy to expand the amount of information stored on the form. The more fields on the form, the longer it takes to load. Note that hidden fields still take time to load despite the fact that they do not display on the form.

If you do not need to store non-keyword and scripting data on an E-Form, it is recommended that you use Virtual E-Forms. Virtual E-Forms do not take up as much a space on a disk group since they store only information that is saved in the database, such as keyword values and system values.

It is recommended to use Document Type revisions when making form template changes.

Do not build business logic into E-Forms if possible.

When developing for the Web Client, load referenced files (scripts, css, xml, etc.) from an external share. Create a separate virtual directory on the Web Server that is separate and distinct from the OnBase installation. This means that the files are untouched by the installer when OnBase is upgraded.

Never edit the E-Form template directly in its disk group location. If you need to edit your SYS – HTML Form, edit your template (not from the disk group) and re-import the form via the OnBase Client. Then reassign the form to the Document Type in the Configuration module. When you edit the form from the disk group, the Item Size for the file will not be updated in the database and when you attempt to retrieve that template, it is possible OnBase will think that the file has been corrupted and it will not display.

Keywords

In most cases, mapping to a Keyword Type number is recommended. When Keyword Type names are used, the renaming of keywords (without a subsequent adjustment to E-Forms) can cause E-Forms to not function. Keyword Type number mapping is effective when Keyword Type names contain characters that are not accepted by HTML. Mapping to a Keyword Type name is recommended only if forms must be imported into multiple databases. Keyword Type numbers cannot be changed in the system. If a form that uses Keyword Type numbers is imported into multiple databases, all form field properties must be changed to map to the Keyword Type numbers in the new database.

When using dates, it is recommended to use drop-down lists or date select controls rather than check boxes or radio button. When using check boxes and radio buttons, leading zero dates do not always display as desired.

When using data sets within fields, limit the total number of data set values to 5000. It is always a good idea to limit field sizes.

Be wary of implementing an OBDataset for which a keyword has an External Dataset configured. The issue is that the OnBase Administrator is not in control of the data in the data set. If a value is ever removed from the LOB system, the forms that were indexed using that value will open and not display a value in the drop-down because no value is in the data set.

Create all hidden keywords in one location in the form. This makes it easier to keep track of them.

Implement hidden keywords as <input type=”text”… in a hidden DIV, and implement a keyboard sequence (Ctrl + Shift + E for example) that makes the DIV visible. This allows an administrator to see all hidden values when troubleshooting a problem with the form.

Scripting

Do not script something for which OnBase already has built in functionality.

Completely separate the HTML and JavaScript. If you need to create an event handler, do not use the ‘onclick' event in the HTML, but attach the event handler in the code.

Store style sheets and any necessary scripting in separate files that are referenced from within the E-Form itself. This cuts down on the size of the form and also allows the code to be re-used somewhere else in a solution.

When a form is coded to use an action ("action="), the action is removed from the E-Form when it is displayed in OnBase by design. Custom JavaScript or VBScript will need to be used to accomplish what the actions would have.

Workflow Interaction

Limit the Number of HTML Fields on the driving document in a Workflow life cycle if it is an E-Form. When the driving document is an E-Form with many HTML fields, loading the driving document if it is an E-Form with many fields can directly impact performance as OnBase must parse the HTML fields and extract the correct field values from post data stored in the disk groups.

If you have a E-Form in a system that you want to import into another system, you can create a Workflow life cycle containing that E-Form. You can export the Workflow life cycle containing the E-Form and upon importing the Workflow life cycle, the E-Form, the associated Document Type, and all associated Keyword Types are imported.

Perform data validation on the form rather than with Workflow logic. Performing data validation should be handled on the E-Form with simple scripting rather than performing the validation within OnBase Workflow. The scripting runs on the client and is lightweight compared to performing the similar logic behind Workflow rules and actions (which may update the database and add overhead to perform the logic).

Always appropriately configure buttons. Default submit button functionality is processed as a cancel button when an E-Form is used in Workflow and a button function is not specified using oBBtn_ syntax.

Offline E-Form Design Considerations

Offline E-Forms are used in conjunction with the Unity Briefcase and Integration for Netsmart Homecare modules. When configuring E-Forms for offline use, there are some special design considerations. The following are special considerations and limitations of offline E-Forms.

- Virtual E-Forms are supported, but are not recommended.
- Signatures applied using the Signature Pad Interface or Digital Signatures module are not supported.
- Electronic Forms that reference external resources most likely will not have access to those resources when used offline. To ensure the integrity of offline E-Forms that use cascading style sheets (CSS) and JavaScript, embed these in the E-Form. To embed an image within a E-Form, unless the image file is stored locally on the client machine, you must use Base64 encoding to make the image available offline.

  Note: Rendering Base64 encoded images is dependent upon your browser.
- When an Electronic Form is used offline, the assignment of the auto-increment keyword is deferred on newly created forms until those forms are synchronized. When the form is synchronized and the form is uploaded and stored into OnBase, the auto-increment keyword is assigned at that point. Auto-increment keywords may be used in offline solutions, but they should not be considered useful on forms created in the field. The keyword should only be considered useful to facilitate the online processes once the form is uploaded.
- Keyword Data Sets embedded within the Electronic Form can be used. Offline forms will not have access to data sets that are external to the form.
- The OBBtn_AutoSave button is not supported in offline forms. External AutoFill Keyword Sets also are not supported.
- The OBBtn_CrossReference button is not supported.
- The OBBtn_xRefItemnum button is not supported.
- The OBBtn_CQ## button is not supported.
- Embedding images stored in OnBase is not supported.
- Using revisable Document Types for offline E-forms may cause unexpected results. It is considered a best practice to use Unity Forms if revisable Document Type functionality is required for forms within a Unity Briefcase solution.

Configuring the ASP.NET Version of LoginFormProc

LoginFormProc presents users with a custom HTML form that they can complete and submit to OnBase as an E-Form.

When configured correctly, LoginFormProc allows users to submit forms using the following Web browsers:

- Apple Safari®
- Google Chrome™
- Microsoft Internet Explorer®
- Mozilla Firefox®

LoginFormProc does not allow you to retrieve documents from OnBase. Another Web Server component, DocPop, is the preferred method for viewing OnBase documents externally. For more information about DocPop and how to use it to retrieve documents from OnBase, see the DocPop reference guide.

LoginFormProc does not allow you to update existing E-Forms or Unity Forms. To update existing E-Forms and Unity Forms, FormPop should be used. For information about configuring FormPop, see FormPop.

Configuration

The login settings for LoginFormProc are set in the Web Server's Web.config. All forms submitted through LoginFormProc are created in OnBase by the user account you specify in Web.config, as described in the following section, LoginFormProc Settings on page 67.

LoginFormProc Settings

Several settings in the Web Server's Web.config file need to be configured for LoginFormProc. These settings are located within the **Hyland.Web.LoginFormProc** element.

username- Specify the user name to be used for creating forms. This account must not be a service account.

password- Specify the password for the supplied user for creating forms.

datasource- Specify the name of the data source to use for creating forms. You must provide a data source for LoginFormProc to work.

prompt- Set to **enable** if you want LoginFormProc to prompt users to create another form. Set to **disable** if LoginFormProc should not prompt users. When prompting is enabled, the HTML form must contain the **OBWeb_Redirect** field, which specifies the path to the form.

Encrypting LoginFormProc Web.config Settings

You can use the Web Application Management Console to encrypt all settings in the **Hyland.Web.LoginFormProc** element. When encrypted, information such as the LoginFormProc **username** and **password** values cannot be viewed within the Web Server's Web.config file.

Encryption is enabled by the **Encrypt Login Form Proc** setting in the Web Application Management Console. When this setting is applied and saved, information within the **Hyland.Web.LoginFormProc** element is replaced with an **EncryptedData** element, which contains the encrypted settings.

For information about using the console to modify and encrypt LoginFormProc settings, refer to the Web Application Management Console module reference guide.

Features

LoginFormProc allows a user to perform several tasks, which are described in the following sections:

- Submitting a new form to the database on page 68
- Opening a read-only copy of the submitted form on page 69
- Redirecting the user to the custom form on page 69
- Setting the Language Parameter on page 70

Note: Values containing HTML or Script code cannot be submitted through LoginFormProc. For example, if a user enters a value of **<text>** and clicks **Submit**, an error is displayed and the form is not submitted. This behavior is by design to prevent a malicious attack on the server. A possible workaround is to use Workflow and scripting. Create a life cycle to capture the incoming E-Form, and create a new form converting text using scripts.

Submitting a new form to the database

Required fields on the custom HTML form:

- **OBDocumentType**
  This field contains the Document Type number, not the Document Type name.
- E-Form fields
- **OBBtn_Save** || **OBBtn_Yes**

The action parameter on the form must include the path to the Web server that will be uploading the form. For example:

```
<form method="post" action="http://[server name]/appnet/loginformproc.aspx?>
```

New forms are submitted to the database using an **OBBtn_Save**(or **OBBtn_Yes**) button that saves information (E-Form fields) to the database. A new E-Form is created in the given Document Type (whose number is specified in the **OBDocumentType** input field), and any keyword values on the form are saved to the database. This button must be **type=submit**.

Either **OBBtn_Save** or **OBBtn_Yes** can be used to submit the form; both function exactly the same.

For example, if the E-Form has **Custom Alpha 250**, **Custom Numeric 9** and **Custom Currency** as keywords and you want users to provide values for them in the new E-Form, then you must include an

HTML Input field for each keyword. These fields allow users to enter keyword values, which will be saved in the new E-Form once it is submitted.

```
<INPUT type="hidden" name="OBDocumentType" value="###" />
<p> Custom Alpha 250 <input type="text" name="OBKey__155_1" size="20"></p>
<p> Custom Numeric 9<input type="text" name="OBKey__125_1" size="20"></p>
<p> Custom Currency <input type="text" name="OBKey__128_1" size="20"></p>
<INPUT type="submit" value="Submit" name="OBBtn_Save" />
```

Opening a read-only copy of the submitted form

After a user submits a new E-Form, LoginFormProc can display read-only copy of the E-Form to show the user the form was properly submitted.

This feature requires the following field on the custom HTML form:

- **OBWeb_ReturnReadOnlyCopy**

When this input is set to **true**, a read-only copy of the submitted form is displayed after the user submits the form. When this input is set to **false**, the read-only copy is not displayed.

For example:

```
<INPUT type="hidden" name="OBWeb_ReturnReadOnlyCopy" value="true" />
```

Note: When **OBWeb_ReturnReadOnlyCopy** is set to **true**, the redirection settings configured for OBWeb_Redirect are overridden. After submitting a form, the user will not be prompted to create a new form.

Redirecting the user to the custom form

Required fields on the custom HTML form:

- **OBWeb_Redirect**

After submitting a new E-Form, the user is prompted to create another new E-Form. Upon choosing **OK**, the user is redirected to the same custom HTML form. For this redirection to work, the value property on the **OBWeb_Redirect** hidden input field must be set to the URL of the custom HTML form.

For example:

```
<INPUT type="hidden" name="OBWeb_Redirect" value="http://servername/appnet/
file.htm">
```

The path to **OBWeb_Redirect** must be provided in the following format: **value="http://[server name]/[virtual directory]/[file.htm]"**.

If the user chooses **Cancel** when prompted to create a new form, then a different target URL must be provided. The **OBWeb_FinalTargetPage** hidden input field is available for this purpose. This field defines the page to which the user is directed, and it takes the same form as the **OBWeb_Redirect** input field. The user is directed to this location under the following conditions:

- The Hyland.Web.LoginFormProc **prompt** setting is set to **disable** in the Web Server's Web.config.

- The Hyland.Web.LoginFormProc **prompt** setting is set to **enable** in the Web Server's Web.config, and the user selects **Cancel** when prompted to create a new form.

Note: When **OBWeb_ReturnReadOnlyCopy** is set to **true**, the redirection settings are overridden. After submitting a form, the user will not be prompted to create a new form.

Setting the Language Parameter

If a form is submitted indirectly by another application, then the form must contain a hidden input field to specify the language parameter. The field's name is **LanguageParam**, and its value specifies the ISO 2-letter code for language and region.

For example, the **LanguageParam** field for English-United States would be the following:

```
<INPUT type="hidden" name="LanguageParam" value="en-us" />
```

The **LanguageParam** parameter is required in certain situations. Forms submitted through Chrome (or an automated process, such as cURL) require the parameter to pass in the locale, but forms submitted directly to LoginFormProc from Internet Explorer, Safari, or Firefox do not require the parameter. If a form submitted through Internet Explorer, Safari, or Firefox contains this parameter, then the parameter takes precedence over the browser's locale.

In certain instances the HTML form is not submitted through LoginFormProc directly. For example, a third-party application may be used to process the data and then submit the post data to LoginFormProc. In this situation, the application is required to pass the **LanguageParam** in as part of the post data.

Licensing

LoginFormProc requires an E-Forms license.

With the Archival API server license, the concurrent license is released immediately without having to wait the 5 minute session timeout required with a standard concurrent license.

If the **LoginFormProc** user is currently logged on to another OnBase application when the form is submitted, the user's session is maintained. This is true regardless of whether the system is licensed for Archival API.

FormPop Overview

FormPop allows users to view and edit E-Forms and Unity Forms using a simplified Web Client viewer interface, without any extra OnBase functionality. This allows users outside of OnBase to follow web links to view and edit forms in OnBase.

FormPop does not allow users to create new forms by following a FormPop link.

For a list of browsers that are supported by FormPop, see FormPop and PDFPop Browser Requirements on page 5.

FormPop and PDFPop Browser Requirements

The following web browsers are supported for use with FormPop and PDFPop:

Web Browser	Supported Versions
Internet Explorer	Internet Explorer 11
Edge	EdgeHTML 14 and higher
Firefox	Firefox 28 and higher (including non-ESR versions)
Safari	Safari 9.1 and higher for OS X and macOS Safari for iOS
Google Chrome	Chrome 29 and higher

Usage

FormPop functionality is described in the following sections:

- Retrieving Forms Using FormPop on page 71
- Editing Existing Forms Using FormPop on page 72

For additional information on the functionality available with forms, see the **E-Forms** or **Unity Forms** help files or module reference guides.

Retrieving Forms Using FormPop

You can use FormPop to retrieve and edit forms in OnBase by clicking a link to the form from a Web site or email message. You can also use the DocPop URL Creator to generate a link and then modify the generated URL, as shown in the example below:

http://WebServer/AppNet/docpop/ **docpop.aspx** ?doctypeid=139

becomes:

http://WebServer/AppNet/docpop/ **FormPop.aspx** ?doctypeid=139

For information on using the DocPop URL Creator, see the **DocPop** module reference guide.

After accessing a FormPop link, a document select list or form is displayed.

Editing Existing Forms Using FormPop

After accessing a FormPop link, edit the necessary fields and save or submit the form.

Configuration

The configuration settings for using FormPop are set in the Web Server's Web.config file. For information on FormPop configuration settings, see FormPop Vars on page 72.

For information specific to configuring E-Forms or Unity Forms, see the **E-Forms** or **Unity Forms** module reference guides.

For information about the variables available in a FormPop query string, see the **DocPop** module reference guide, which contains a comprehensive list of query string variables.

FormPop Vars

FormPop-specific settings are located in the **Hyland.Web.FormPop** element of the Web Server's Web.config file. The only required setting is a data source. You can either configure one in the Web Server's Web.config or pass it along the query string.

FormPop results are displayed using the Web Client.

The following settings are located in the **Hyland.Web.FormPop** element of the Web Server's Web.config file.

username- Enter the user name to use with default login with FormPop, if you want to use a single user account for access. When **enableDefaultLogin** is set to **true**, users can automatically log on to FormPop using the credentials provided in the **username** and **password** settings.

password - Enter the password to use with default login with FormPop, if you want to use a single user account for access. When **enableDefaultLogin** is set to **true**, users can automatically log on to FormPop using the credentials provided in the **username** and **password** settings.

datasource- Enter the name of the data source to use with FormPop. This is a required value.

domain- Enter the domain to log on to if you are using Active Directory authentication.

embedded - Set this to **true** when you are embedding FormPop results in a custom Web page and you want the FormPop results to be displayed in a frame or iframe within the same browser window. When set to **false**, FormPop results are opened in a new window.

- If **embedded** is set to **true** and results are not embedded in another Web page, then the address bar and browser toolbars will be displayed when a user accesses the FormPop URL.
- If FormPop results will not be embedded in Web pages, set **embedded** to **false**. The address bar and toolbars will be hidden when FormPop results are displayed.

enableDefaultLogin- Set this to **true** to have FormPop use the **username** and **password** credentials specified in the **Hyland.Web.DocPop** element. Set this to **false** to have FormPop either attempt other authentication methods (if they are configured) or prompt the user for credentials.

enableHTTPLogin- Set this to **true** to pass login credentials to the server on the query string or to post them through an HTML form. Set this to **false** if FormPop should either attempt other authentication methods (if they are configured) or prompt the user for credentials.

enableAutoLogin- Set this to **true** to use domain credentials to log on to FormPop automatically. When this is set to **false**, FormPop either attempts other authentication methods (if they are configured) or prompts the user for credentials. If you enable this setting, ensure that the Web Server is configured for Active Directory authentication. See the **Legacy Authentication Methods** module reference guide for more information about Active Directory authentication.

Set **enableAutoLogin** to **true** if you are using Integration for Single Sign-On. If OnBase is configured for Active Directory or LDAP authentication, but you want to use Single Sign-On with FormPop, set both **forceSSOAutoLoginOverDomain** and FormPop's **enableAutoLogin** setting to **true**. For more information about Integration for Single Sign-On, see the **Legacy Authentication Methods** module reference guide.

enableHTTPDataSource- Set this to **true** to pass the data source on the query string. Set to **false** to use the FormPop data source in the Web Server's Web.config.

enableChecksum- If set to **true**, a checksum value will be added to the URL query string. To enable checksums, you are also required to enter a checksum key value in the FormPop **checksum** setting, which is used to create the checksum value in the URL. When a user attempts to retrieve a document using the URL, FormPop compares the checksum in the query string to the expected checksum. If the values match, the document is displayed. If the values do not match, the user is presented with an error. This is to prevent users from modifying query strings and accessing documents they should not access. If set to **false**, no checksum is created.

checksum- Enter the unique string value used as a key for external, dynamic checksum creation. This string value should not be well known. The **checksum** setting applies only when **enableChecksum** is set to **true** and an external automated process is being used to dynamically generate FormPop links.

Note: Configuration of this setting is required for checksum creation and validation.

- The Web Server web.config file also has an **enableChecksum** setting within the **<Hyland.Web.DocPop>** node that must be set to **true**. You must also set the **checksum** setting to the appropriate value within that node.
- The Application Server web.config file also contains a Pop integration checksum setting: **ChecksumKey**. This setting is used for checksum generation when the docID is used from outside of the Web Client solution (for example, in Workflow notifications). If you use this feature, the **ChecksumKey** value in the Application Server web.config file must match the **checksum** value in the **Hyland.Web.FormPop** element of the Web Server web.config file. For more information about checksum generation, please refer to the Hyland SDK.
- If you are using the Workflow action **Med - Send HL7 Message**, the Hyland.Web.FormPop **checksum** value should be empty. If an external process will generate the FormPop URLs and you want to use checksums, then a separate virtual directory for FormPop should be configured.

enableCoreQueryAPILicense- This setting requires OnBase to be licensed for Core Query API (Retrievals Per Hour). Set this setting to **true** if you want users to consume Core Query API licenses when using FormPop. Core Query API licenses help prevent the unnecessary consumption of Concurrent Client licenses. When this setting is set to **true**, a Core Query API license is consumed as soon as a user logs on to FormPop and is released immediately after the user logs off. When the **enableCoreQueryAPILicense** setting is set to **false**, a Concurrent Client license is used.

Note: Core Query API licenses are only available for external users.

AutoDisplayOneDocument- Set this setting to **true** to always display only the viewer for FormPop queries that return a single result. When this setting is set to **false**, FormPop displays both the hit list and the viewer for queries that return a single result. This behavior can be overridden by the **viewerOnlyForSingle** variable in the FormPop query string. The **viewerOnlyForSingle** variable has no effect when **AutoDisplayOneDocument** is set to **true**.

Embedding FormPop Results in a Web Page

By default, if you are embedding content from the OnBase Web Server into a web page, the page and the embedded content must be on the same domain. If your solution requires embedding Web Server content into a different domain, you can configure the Web Server to allow this. For more information, see the section on X-Frame-Options in the **Web Server** module reference guide.

Enterprise Integration Server E-Form Design Overview

E-Forms and Virtual E-Forms can be created directly from a line-of-business application with the Enterprise Integration Server. When configuring E-Forms for use the Enterprise Integration Server, there are some special design considerations and limitations.

Non-Keyword Fields

E-Forms can be used to capture non-keyword data. An example of non-keyword data is a college applicant's test scores. While non-keyword data is important to the overall E-Form, it is not important enough to the business process to be captured as Keyword Values.

When capturing non-keyword data with the Enterprise Integration Server, the following format must be followed for E-Form fields:

OBNONKey__REPEATED__<group name>__<field name>_<sequence #>

Format	Description
OBNONKey__REPEATED__	Every non-keyword field that is mapped must start with the Document Management System code **OBNONKey__REPEATED__**. **Note: OBNONKey** and **REPEATED** are each followed by two underscore characters (__).
<group name>	The name of the group of non-keyword data followed by two underscores. For example, **Tests__**. **Note:** If a group name is not included, the field will be treated as ungrouped.
<field name>	The name of the E-Form field, followed by an underscore. For example, **TestName_**. **Tip:** It is considered a best practice to give descriptive and unique names to the non-keyword fields on the E-Form. This assists with mapping of the E-Form fields.

Format	Description

<sequence #>	The number of occurrences of the field on the form. For example, the first use of the **TestName** field would be mapped as **TestName_1**, the second occurrence would be mapped as **TestName_2**, etc.
	E-Form fields must appear in continuous ascending sequence, beginning with 1.
	Correct Use:
	The following is correct use of this format:
	`<INPUT name="OBNONKey__REPEATED__TestName_1">`
	`<INPUT name="OBNONKey__REPEATED__TestName_2">`
	Incorrect Use:
	The following is an incorrect use of this format because the sequence appears in the wrong order:
	`<INPUT name="OBNONKey__REPEATED__TestName_2">`
	`<INPUT name="OBNONKey__REPEATED__TestName_1">`
	The following is an incorrect use of this format because the sequence is not continuous:
	`<INPUT name="OBNONKey__REPEATED__TestName_1">`
	`<INPUT name="OBNONKey__REPEATED__TestName_4">`
	The following is an incorrect use of this format because the sequence did not begin with 1:
	`<INPUT name="OBNONKey__REPEATED__TestName_2">`
	`<INPUT name="OBNONKey__REPEATED__TestName_3">`

See Example on page 76 for an example of the type of non-keyword data that can be captured in an E-Form using the Enterprise Integration Server.

Example

You work in a college admissions office. You want to capture data from the following table in your line-of-business application in an E-Form, but do not want to capture this data using Keyword Values:

High School Test Name	High School Test Score
SAT	1220
ACT	26

Configure the table in the E-Form in the following way:

```
<table border="1">
<tr>
    <th>High School Test Name</th>
    <th>High School Test Score</th>
</tr>
<tr>
```

```
    <td><input type"=text" name="OBNONKey__REPEATED__Tests__TestName_1"></td>
    <td><input type"=text" name="OBNONKey__REPEATED__Tests__TestScore_1"></td>
</tr>
<tr>
    <td><input type"=text" name="OBNONKey__REPEATED__Tests__TestName_2"></td>
    <td><input type"=text" name="OBNONKey__REPEATED__Tests__TestScore_2"></td>
</tr>
</table>
```

When performing Enterprise Integration Server mapping, map the “Test Name” and “Test Score” values from the line-of-business application's schema to the corresponding E-Form values in the E-Form schema's “Tests” group.

When the Enterprise Integration Server generates the E-Form from the line-of-business application, the first “Test Name” value will be used to fill the **OBNONKey__REPEATED__Tests__TestName_1** element. The second “Test Name” value will be used to fill the **OBNONKey__REPEATED__Tests__TestName_2** element.

Special Characters

The Enterprise Integration Server converts special characters in E-Form **name** elements during mapping. The **#** character is converted to **Num**. For example, **PO#** would be converted to **PONum**. Other special characters are converted to an underscore.

Ensure that you do not include **name** elements on your E-Form that may conflict during mapping. For example, do not include elements named **PO#** and **PONum**. Upon conversion, both elements would be converted to **PONum**. The Enterprise Integration Server does not support this E-Form configuration.

Keyword Fields

When creating E-Forms from a line-of-business application with the Enterprise Integration Server and capturing data as Keyword Values, all data received from the line-of-business application must be properly formatted in order to successfully create the E-Form. For example, a twenty character Alphanumeric Keyword Type is configured in OnBase. The line-of-business application sends a thirty character Alphanumeric value to the Enterprise Integration Server. In this example, the E-Form would not be created because the value sent from the line-of-business application is longer than the Keyword Type configured in OnBase.

AutoFill Keyword Sets

When creating E-Forms with the Enterprise Integration Server and configuring the **AllowAutoFillExpansion** binding property to automatically expand AutoFill Keyword Sets when the primary Keyword Type is received, ensure that:

- When an AutoFill Keyword Set is applied at the Document Type or Keyword Type level, all of the Keyword Types in the AutoFill Keyword Set are part of a single group. For example:
    - If the primary keyword is part of a Keyword Type Group, all secondary keywords must be part of the same Keyword Type Group.

    - If the primary keyword is not part of a Keyword Type Group or Multi-Instance Keyword Type Group, all secondary keywords cannot be part of a Keyword Type Group or Multi-Instance Keyword Type Group.
    - If the primary keyword is part of a Multi-Instance Keyword Type Group, secondary keywords can be part of the same Multi-Instance Keyword Type Group, part of a Keyword Type Group, or can be Standalone Keyword Types.
- When an AutoFill Keyword Set is applied at the Keyword Type level, it cannot contain overlapping values with another AutoFill Keyword Set. For example, a Keyword Value is a secondary keyword in one AutoFill Keyword Set and a primary keyword in another AutoFill Keyword Set.

Note: AutoFill Keyword Sets used on E-Forms cannot be automatically expanded if the primary Keyword Type is part of multiple AutoFill Keyword Sets or Keyword Type Groups. If a primary Keyword Type is part of multiple AutoFill Keyword Sets or Keyword Type Groups, the E-Form is created, but any AutoFill Keyword Sets are not expanded. Additionally, secondary Keyword Types in the AutoFill Keyword Set cannot be the primary Keyword Type in another AutoFill Keyword Set.

Unsupported E-Form Features

The following E-Form features are not supported for use with the Enterprise Integration Server:

- AutoFill Keyword Sets containing more than one primary Keyword Value
- Keyword Data Sets
- Tables that grow dynamically (e.g., clicking on a button to expand a table using JavaScript)
- Data validation of non-keyword fields

OnBase®

E-Forms

User Guide

Foundation 25.1

Usage

E-Forms are forms that you complete and submit to the system electronically. E-Forms can be set up to support any business or organizational activity that requires the entry, editing, routing, and approval of information. Because the system maintains complete audit trails for E-Forms, they can be tracked from inception to disposition, and authenticated at every step of the process.

E-Forms are completely configurable, so your system can include whatever forms are desired to support your activities. Your system administrator determines who can use which forms, so the forms available to you may differ from those available to others.

Log Into the Client

E-Forms are created and managed in the Client module. Log into the Client as a member of a user group that has rights to use HTML forms. Contact your system administrator if you cannot use HTML forms and believe you should have the correct rights.

Register the Workstation

No workstation registration is required for the E-Forms module. The E-Forms database license is all that is required to use E-Forms.

Creating E-Forms in the Client

Once a form has been designed and imported into OnBase by a system administrator, you can use the following procedure to create a form in the Client:

1. Open the **New Form Document** dialog box by performing one of the following actions:
   a. Click the **Create New Form** Toolbar Button.
   b. Select **File | New | Forms**. The new Form Document is displayed.
   c. Press **Ctrl + e** on the keyboard.
2. Select the desired form and click **Create** or double-click on the form. If you used the **Create New Form** button, it may have been configured to automatically select the correct E-Form. A blank E-Form is displayed.

   The following limitations apply when working with Keywords in E-Forms:

   - If the Document Type for the E-Form has Default Keyword Values configured for any Keyword Types, those values are filled automatically upon form creation.
   - If your E-Form is configured to use an AutoFill Keyword Set to automatically populate Keyword Values, only one instance of the AutoFill Keyword Set can be associated with the E-Form.
   - If the E-form has been configured using encrypted keywords, depending on your user rights, you may not be able to save values for the encrypted keywords. For more information, contact your System Administrator

3. Fill in the field entries on the form and click on the button that indicates the form is to be saved into the system (for example, **Submit** or **Save**).
4. The prompt **Would you like to create another new document?** is displayed. Select **Yes** to create a new form or **No** to return to the Client module.
5. To access the new form, select the E-Form's Document Type Group and Document Type and click **Find**. Double-click on the form from the **Document Search Results** list. The system administrator determines the appropriate viewing rights to the Document Types.

   Note: Date formats follow the system locale of the workstation.

Printing E-Forms

Depending on the OnBase configuration, E-Forms print using either the OnBase print dialog box or the print dialog box associated with Microsoft® Internet Explorer.

Note: Users cannot print new, unarchived forms in the OnBase Client when the **Use internal printing logic for HTML documents** option is selected. When the option is not selected, users can only print new, unarchived forms using the Internet Explorer print dialog box.

Regardless of the configuration, E-Forms always use the Internet Explorer print dialog box when **Print** is selected from an open document's right-click menu and always use the OnBase print dialog box when **Print** is selected from a Document Search Result list's right-click menu.

You can view a print preview of an E-Form document that is open in the Document Viewer by right-clicking the open E-Form and selecting **Print Preview**.

The print preview of the E-Form is displayed in the **Print Preview** window associated with Microsoft® Internet Explorer. You can use the view settings and page setup controls in the **Print Preview** window to adjust how the E-Form is displayed and printed.

For more information on printing an E-Form, see the **Printing Documents in the Client** section of the **Client** module reference guide.

Creating and Deleting Notes on E-Forms

There are two ways to create a Note on E-Forms:

- Select **Document | Create Note** on the Client Toolbar.
- Right-click on the **Notes** status indicator at the bottom of the open E-Form and select **Add Note**.

Select the desired Note Type and click **Add Note**.

The following limitations apply when working with notes in E-Forms:

- The initial position of a note on an E-Form is saved. If the position of an existing note is changed, the position will only be saved if you right-click on the note and select **Save Note Position**. Otherwise, the new note position will not be saved.
- In order to print an E-Form displaying any notes or annotations, the E-Form must be printed from the OnBase print dialog box, not the Web browser's print dialog box. For more information, see Printing E-Forms on page 81.

To delete a Note on an E-form, perform one of the following actions:

- Right-click on the Note Type and select **Delete Note**.

- Double-click on the Notes status indicator, right-click on the Note Type and select **Delete Note**.

Retrieving and Re-Indexing E-Forms

E-Forms can be retrieved and re-indexed like other documents. To re-index an E-Form after it has been stored, retrieve the document and select **File |Re-index**. Changing the Keyword Value of a keyword that is mapped to a form field displays the new value.

The following limitations apply when retrieving and re-indexing E-Forms:

- If a document is opened and is already in use by another user, any editable fields will be read-only. Submission buttons are disabled when an E-Form is read-only.
- E-Forms should only be re-indexed to Document Types that are configured as E-Form Document Types.

Shortcut Keys in E-Forms

You can use several different keyboard shortcuts to navigate an E-Form. Some shortcuts are assigned specific functionality depending on the type of form field selected .

To navigate an E-Form with no form fields selected , use the following shortcuts :

Shortcut	Action
Home, Ctrl + Home	The cursor moves to the beginning of the E-Form.
End, Ctrl + End	The cursor moves to the end of the E-Form.
Page Up	The E-Form scrolls up to the previous page.
Page Down	The E-Form scrolls down to the next page.

To navigate an E-Form with a text box selected, use the following shortcuts:

Shortcut	Action
End, Ctrl + End	The cursor moves to the end of the text.
Home, Ctrl + Home	The cursor moves to the beginning of the text.

To navigate an E-Form with a text area (a multiple lines of text) selected , use the following shortcuts :

Shortcut	Action
End	The cursor moves to the end of the current line.
Ctrl + End	The cursor moves to the end of the text.
Home	The cursor moves to the beginning of the current line.
Ctrl + Home	The cursor moves to the beginning of the text.

Configuring Forms for Faxing

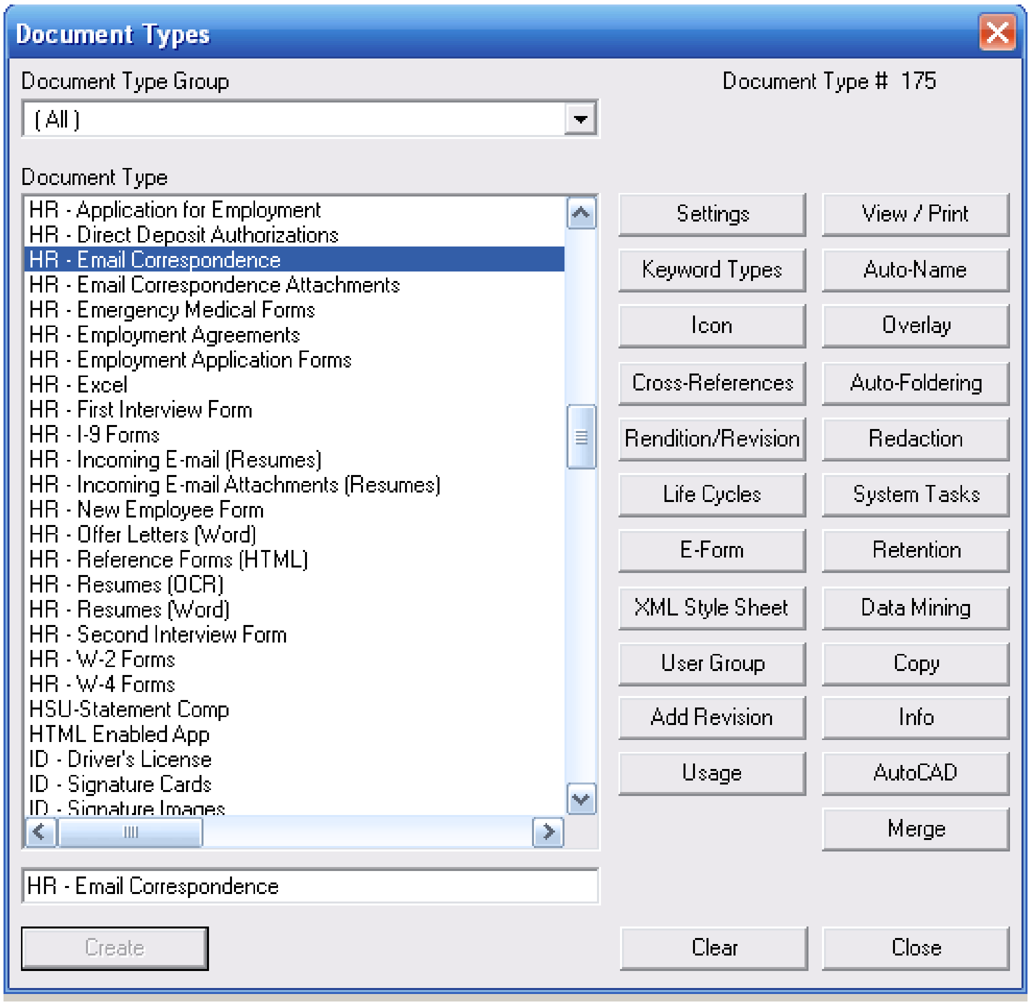
Note: This option is retained for legacy purposes. As of OnBase version 3.9.0, faxing E-Forms is inherent to the module.

E-Forms can be faxed from the Client module. To fax, click **File | Print**, and then select **Fax Compatible**.

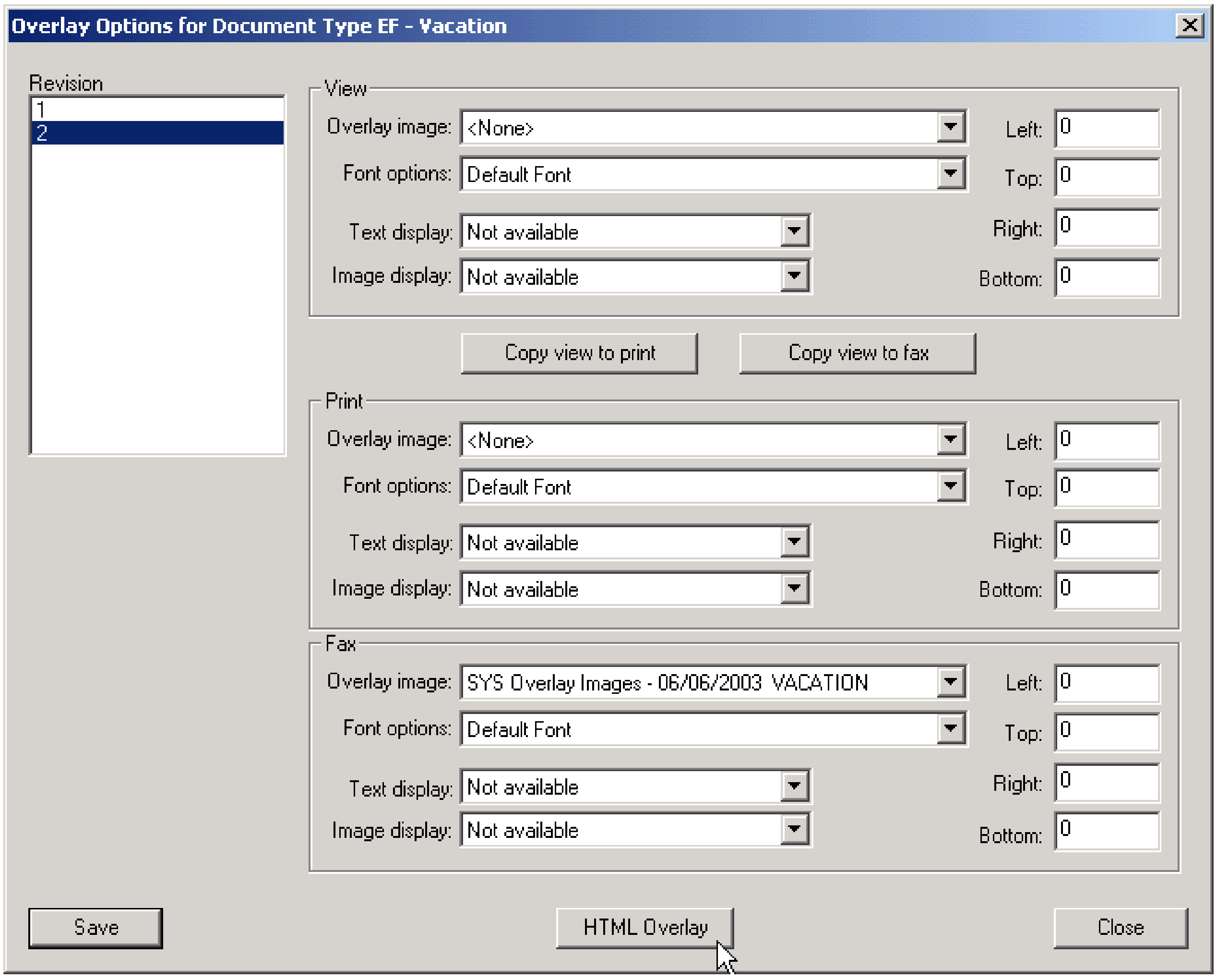
The form must be properly configured in order to use this option. A fax is sent at low resolution to reduce the amount of data that must be transferred. The default image provided to the fax machine does not translate well into low resolution.

To configure a low-resolution image for faxing a form:

1. Print the E-Form to be configured for faxing with all the value fields blank.
2. Scan the printed E-Form to create a low-resolution image (100 dpi). This file will be used as a template when the E-Form is faxed.
3. In OnBase Client, import the file into the **SYS Overlay Images** Document Type.
4. In OnBase Configuration, select **Document | Document Types**.
   The **Document Types** dialog box is displayed.



5. Highlight the Document Type of the E-Form to be configured for faxing and click **Overlay**. The **Overlay Options** dialog box is displayed.



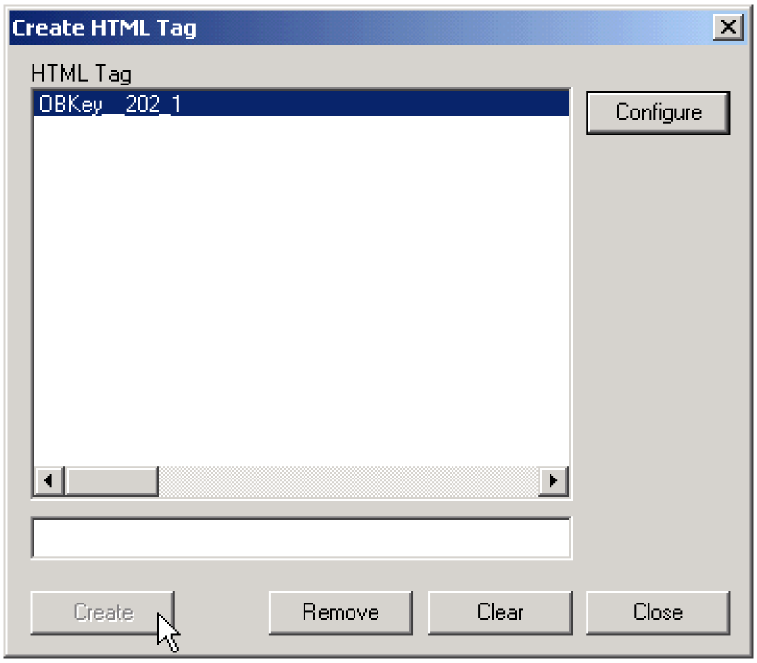
Overlay Options for Document Type EF - Vacation

Revision: 1, 2 (2 selected)

View			
Overlay image:	<None>	Left:	0
Font options:	Default Font	Top:	0
Text display:	Not available	Right:	0
Image display:	Not available	Bottom:	0
Print			
Overlay image:	<None>	Left:	0
Font options:	Default Font	Top:	0
Text display:	Not available	Right:	0
Image display:	Not available	Bottom:	0
Fax			
Overlay image:	SYS Overlay Images - 06/06/2003 VACATION	Left:	0
Font options:	Default Font	Top:	0
Text display:	Not available	Right:	0
Image display:	Not available	Bottom:	0

6. Select the current revision.
7. In the **Fax** section, select the overlay image of the E-Form.
8. Click **Save**.
9. Click **HTML Overlay**.

The **Create HTML Tag** dialog box is displayed.

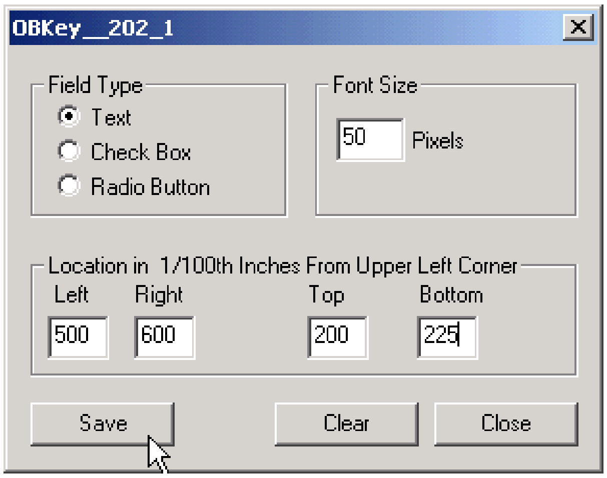


OBKey__202_1

10. In the edit field, enter the name of a field to print.

    The name must exactly match the corresponding tag name in the E-Form HTML template.

11. Click **Create**.
    The tag is saved, and the field configuration dialog box is displayed.



Field Type
(● Text) Check Box / Radio Button

Font Size: 50 Pixels

Location in 1/100th Inches From Upper Left Corner

Left	Right	Top	Bottom
500	600	200	225

12. The information entered in this dialog box is used to insert values into the correct position on the faxed E-Form. Depending on the type of field configured, select **Text**, **Check Box**, or **Radio Button** in the **Field Type** section.
13. In the **Font Size** section, enter the size, in pixels, of the font to print with (50 pixels is a good starting size).

    In the previous example, the value contained in an edit field will print with a height of 50 pixels

14. Enter the position of the field in hundredths of an inch. The position is measured from the upper left corner of the E-Form.

    In the previous example, the edit field is located 5 inches from the left edge (shown as "500" in the **Left** field) of the E-Form and is 1 inch wide (shown by subtracting the **Left** field from the "600" in the **Right** field). It is located 2 inches from the top edge of the E-Form (shown as "200" in the **Top** field) and is ¼ inch high (shown by subtracting the **Top** field from the "225" in the **Bottom** field).
15. Configure each field to print on the form.
16. Test the configuration by printing the E-Form from OnBase Client. Select **Fax Compatible**, but print to a regular printer rather than a fax machine.
17. Adjust font size and positioning as needed in OnBase Configuration.

Using E-Forms in an Institutional Database

E-Forms can be configured for use in an Institutional setting. Each E-Form used in an individual Institutional Database is configured with an **Institution #** K eyword T ype. For more information on configuring E-Forms for Institutional use, s ee the **Institutional Databases** module reference guide .

If a user in an Institutional Database submits an E-Form without including the value for **Institution #**, that value will still be saved to the database. If the E-Form has been configured without a field to input the **Institution #** value, the value will also still be saved to the database.

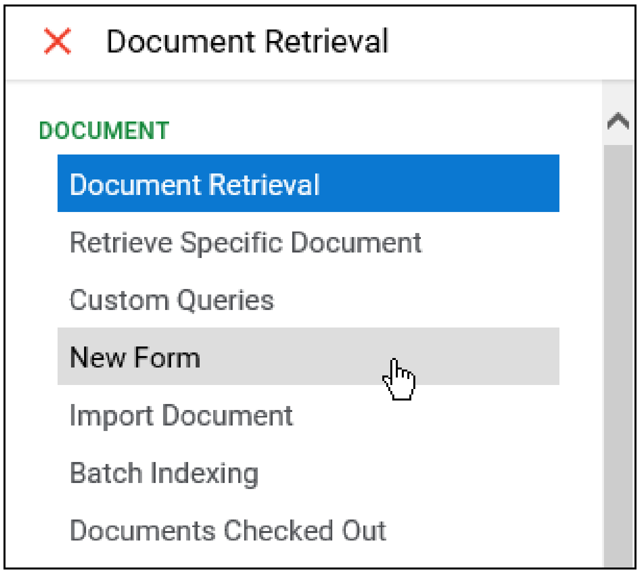
Usage

E-Forms are forms that you complete and submit to the system electronically. E-Forms can be set up to support any business or organizational activity that requires the entry, editing, routing, and approval of information. Because the system maintains complete audit trails for E-Forms, they can be tracked from inception to disposition, and authenticated at every step of the process .

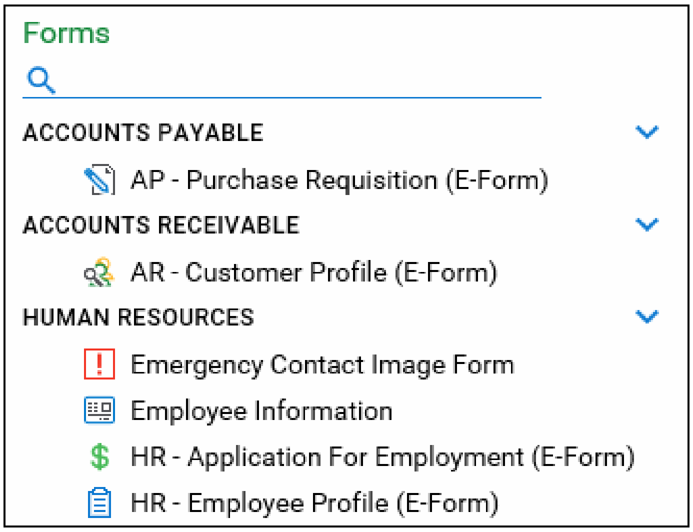
E-Forms are completely configurable, so your system can include whatever forms are desired to support your activities. Your system administrator determines who can use which forms, so the forms available to you may differ from those available to others.

To create a new form:

1. Select the **Main Menu** button in the top-left corner of the Client.
2. Select **New Form** from the **Document** section of the menu.



3. The **New Form** panel is displayed, which lists theforms that are available to you.



Forms

ACCOUNTS PAYABLE
- AP - Purchase Requisition (E-Form)

ACCOUNTS RECEIVABLE
- AR - Customer Profile (E-Form)

HUMAN RESOURCES
- Emergency Contact Image Form
- Employee Information
- HR - Application For Employment (E-Form)
- HR - Employee Profile (E-Form)

You can enter search terms in the field above the list to filter the available forms.

Note: You can navigate down the list of available forms by pressing **Tab** on your keyboard until one of the forms is highlighted. Then use the arrow keys to navigate up and down the list. Press **Enter** on your keyboard to select a form.

4. Select the E-Form that you want to complete and submit. The E-Form is displayed. You can enter the name of the form in the **Find** field and press **Enter** on your keyboard to limit the forms displayed for selection.
5. Complete the form.

   The following limitations apply when working with Keywords on an E-Form:

   - If your E-Form is configured to use an AutoFill Keyword Set to automatically populate Keyword Values, only one instance of the AutoFill Keyword Set can be associated with the E-Form.
   - If the Document Type for the E-Form has Default Keyword Values configured for any Keyword Types, those values are filled automatically upon form creation.
   - If the E-form has been configured using encrypted keywords, depending on your user rights, you may not be able to save values for the encrypted keywords. For more information, contact your System Administrator.
   - When using a date/time formatted Keyword Type, the time will default to 12:00:00am.
6. When you have finished filling out the form, submit the form.

Note: Date formats follow the workstation's system locale.

At this point, depending on how your system is set up, the form may be checked for completeness and validity before being transmitted. If a problem is encountered, you will be notified and given instructions for resolving it.

When the form is successfully transmitted, a message is displayed stating **E-Form for Document Type '<Document Type>' created successfully.** and, depending on your configuration, you are prompted with **Would you like to complete another form?** Click **OK** to create another form. Click **Cancel** if you do not want to create another form.

Note: When you create a new E-Form in a revisable Document Type, a new E-Form is created regardless of whether or not Keyword values of the new E-Form match the Keyword values of an existing E-Form.

7. To complete another form of the same type, click **Yes**. Click **No** if you are finished completing E-Forms of this type.

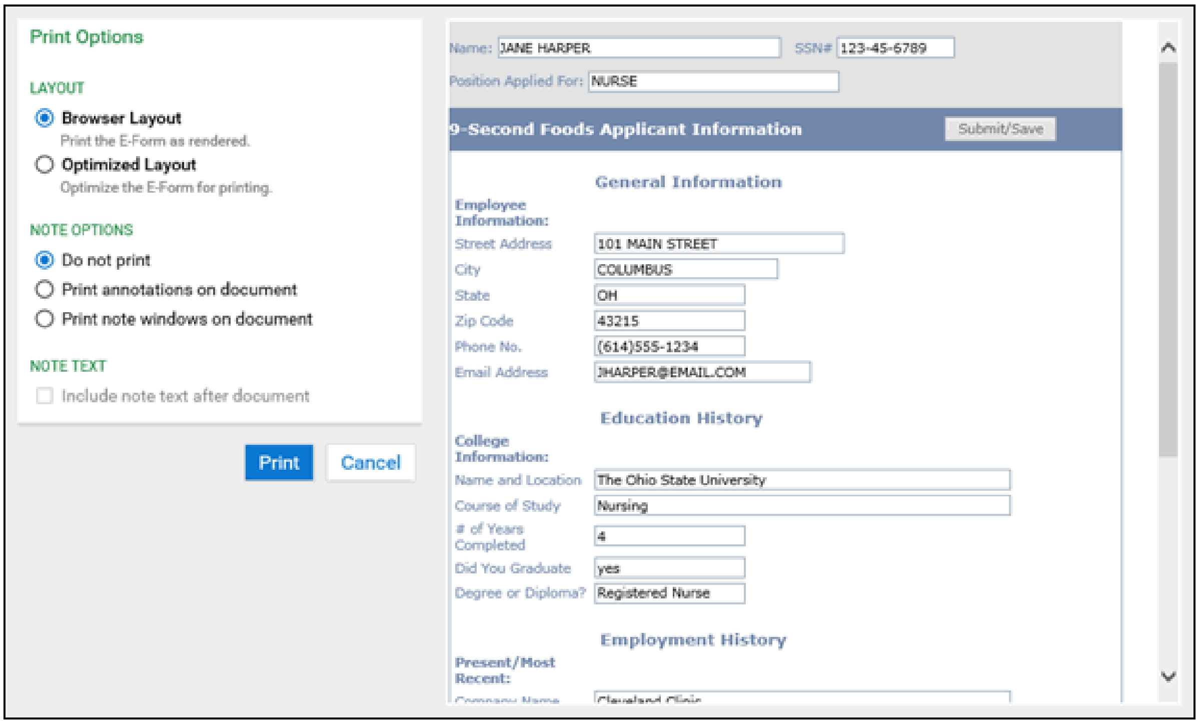
Tip: You can cut (Ctrl + x), copy (Ctrl + c), and paste (Ctrl + v) text to and from form fields.

Printing E-Forms

If you have rights to print E-Form documents, prior to printing an E-Form, you can select an optimized print layout for the form. This process renders the E-Form in a more print-friendly layout.

To optimize an E-Form for printing:

1. Open the context menu by doing one of the following:
   - Right-click on an open E-Form in the Document Viewer.
   - Click the menu button (⋮) on the Document Viewer pane while an E-Form is open.
2. Select the **Print** menu option. The E-Form is displayed in a new window:



Print Options

LAYOUT
- [✓ Browser Layout] Print the E-Form as rendered.
- Optimized Layout Optimize the E-Form for printing.

NOTE OPTIONS
- [✓ Do not print]
- Print annotations on document
- Print note windows on document

NOTE TEXT
- Include note text after document

Print Cancel

Name: JANE HARPER SSN# 123-45-6789
Position Applied For: NURSE

9-Second Foods Applicant Information Submit/Save

General Information

Employee Information:

Street Address	101 MAIN STREET
City	COLUMBUS
State	OH
Zip Code	43215
Phone No.	(614)555-1234
Email Address	JHARPER@EMAIL.COM

Education History

College Information:

Name and Location	The Ohio State University
Course of Study	Nursing
# of Years Completed	4
Did You Graduate	yes
Degree or Diploma?	Registered Nurse

Employment History

Present/Most Recent:

Company Name	Cleveland Clinic

From this window, you can select an optimized print layout for the E-Form. By default, the **Browser Layout** print option is selected. This option prints the E-Form as it is rendered by your Web browser. A preview of how the E-Form will print is displayed on the right side of the screen.

3. To choose an optimized print layout, select **Optimized Layout**. The following options become available:
   - **In-Place**: Removes the form field boxes from the print view, leaving behind the text values from the form fields. The styling of the E-Form is retained.
   - **Row-Based**: Removes the styling of the E-Form, leaving behind only the form field titles and their field values.
   - **Tabular-Based**: Removes the styling of the E-Form, and places the form field titles and their field values into a table layout.

   The E-Form print preview is updated based on the optimized layout setting that you select.
4. Select an option in **Note Options** to specify how to print notes on the E-Form:

   Note: The **Note Options** are not available for row-based or tabular-based optimized layouts.
   - The **Do not print** option does not allow notes to be printed on the E-Form.
   - The **Print annotations on document** option prints the note annotation (graphical representation of a note) on the E-Form.
   - The **Print note windows on document** option prints the title and text of any notes in that note's location on the E-Form, the name of the user that created the note, and the date and time it was created.

   The E-Form print preview is updated based on the **Note Options** that you select.
5. Select the **Include note text after document** option to print the title and text of any notes, the name of the user that created the note, and the date and time it was created below the printed form field titles and values.

   Note: The **Include note text after document** option is only available for row-based or tabular-based optimized layouts.
6. When you are finished selecting a print layout, click **Print**. Your Web browser's print dialog box is displayed.
7. Click **Print** to continue printing the document.

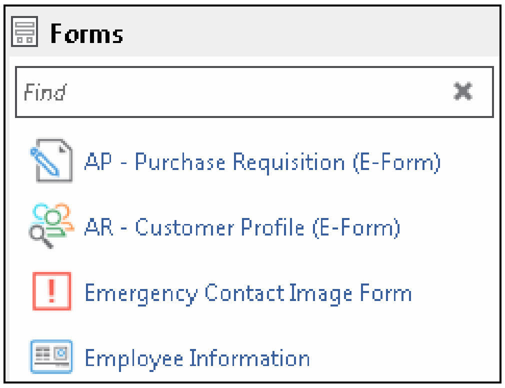
Creating E-Forms

E-Forms are forms that you complete and submit to the system electronically. E-Forms can be set up to support any business or organizational activity that requires the entry, editing, routing, and approval of information. Because the system maintains complete audit trails for E-Forms, they can be tracked from inception to disposition, and authenticated at every step of the process.

E-Forms are completely configurable, so your system can include whatever forms are desired to support your activities. Your system administrator determines who can use which forms, so the forms available to you may differ from those available to others.

To create a new form in the Unity Client:

1. In the **Create** group, click **Forms**.
   The **Forms** panel is displayed.


Forms
Find
AP - Purchase Requisition (E-Form)
AR - Customer Profile (E-Form)
Emergency Contact Image Form
Employee Information

2. If you want to narrow the forms displayed for selection, in the **Find** field, enter text that is contained in the name of the form you want to create.
3. You can sort forms by the Document Type Group they are associated with or list them alphabetically by clicking the **Group by Document Type Group** button to toggle the display.

4. Click on the E-Form you want to create.
   The form is highlighted and displayed in the right pane.
5. Enter the data into the E-Form and submit the E-Form.
   A message asking **Would you like to create a new form?** is displayed.
6. Select an option based on whether you are finished creating forms:
   - Click **Yes** to create another form of the same type.
   - Click **No** if you are finished creating forms.

Limitations When Creating E-Forms

The following limitations apply when entering data into and submitting E-Forms:

- If the Document Type for the E-Form has Default Keyword Values configured for any Keyword Types, those values are filled automatically upon form creation.
- If you select another form while you are entering information into a form, a message is displayed stating **You have not submitted the form. Would you like to return and submit the form before continuing?** Click **Yes** to return to the form and finish completing it. Click **No** to discard the form and open the newly selected form.
- When you create a new E-Form in a revisable Document Type a new E-Form is created, regardless of whether or not Keyword values of the new E-Form match the Keyword values of an existing E-Form.
- When using a date/time formatted Keyword Type, the time will default to 12:00:00am.
- If your E-Form is configured to use an AutoFill Keyword Set to automatically populate Keyword Values, only one instance of the AutoFill Keyword Set can be associated with the E-Form.
- When filling out an E-Form, if the connection is lost to the Application Server, the E-Form cannot be saved. Close the E-Form and reopen the form, and it will be able to be saved.

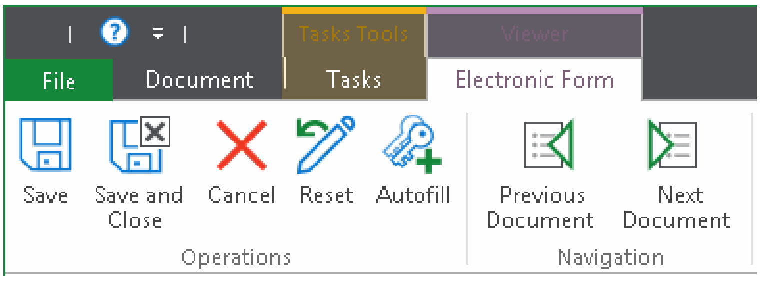
Sending Links to Forms for Creation

If your system is configured to do so, links to forms can be sent via email from the Unity Client that will initiate form creation. To send a link to a form for creation:

1. In the **Home** ribbon, within the **Create** ribbon group, click **Forms**.
2. Do one of the following:
   - Right-click on the form in the **Forms** pane.
   - Click the menu button (⋮) in the **Forms** pane.
3. Select **Mail Recipient (as Link)**.
   An email message will open with a link to the form.
4. Send the email to the proper recipient.

Electronic Form Ribbon

When an E-Form is open for editing or during the E-Form creation process, the **Electronic Form** ribbon is available.



The following buttons are available depending on the configuration of the E-Form you are completing:

Button	Description
Save	Submits the E-Form. This button is available when creating or editing an E-Form.
Save and Close	Submits and closes the E-Form. This button is available when editing an E-Form.
Cancel	Cancels the creation or changes to the E-Form and closes it. Upon clicking **Cancel** for an E-Form during creation or after being edited, a message asking **This form has unsaved data. Are you sure you would like to cancel and lose your changes?** is displayed. Click **Yes** to cancel and discard data. Click **No** to abort the cancellation.
Reset	Clears all changes made to the form fields and restores the form fields to their previously saved values.
Autofill	Expands an AutoFill Keyword Set when a button on an E-Form is configured to expand an AutoFill Keyword Set.
Previous Document	Opens the previous document in the Document Retrieval list.
Next Document	Opens the next document in the Document Retrieval list.