E-Forms

Version 10.0.1

COPYRIGHT

Attribute	Detail
Document Name	E-Forms
Department/Group	Documentation
Revision Number	10.0.1
Part Number	E-FormsRM-10.0.1- -OB

i

Exposure

Usage - Client Module

Usage - Web Client

Usage - Java Web Client

Usage - Desktop

Usage - Unity Client

Configuration

Installation

E-Forms Best Practices

Offline E-Form Design

EXPOSURE

1

OVERVIEW

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

E-Forms requires the E-Forms license.

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

USAGE - CLIENT MODULE

2

USAGE

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 Forms in the Client

Once a form has been designed and imported into OnBase by a system administrator, follow these instructions 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 displays.
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 displays.
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 (e.g. **Submit** or **Save**).

Note: 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.

Caution: 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.

4. The prompt **Would you like to create another new document?** displays. 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 an E-Form

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

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

For more information on printing an E-Form, see "Printing Documents in the Client" in the Client Help.

Creating and Deleting Notes on an E-Form

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

- 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**.

Note: 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.

Note: 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 an E-Form on page 4.

To delete a Note on an E-form:

- 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.

Note: 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.

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:

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:

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:

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.

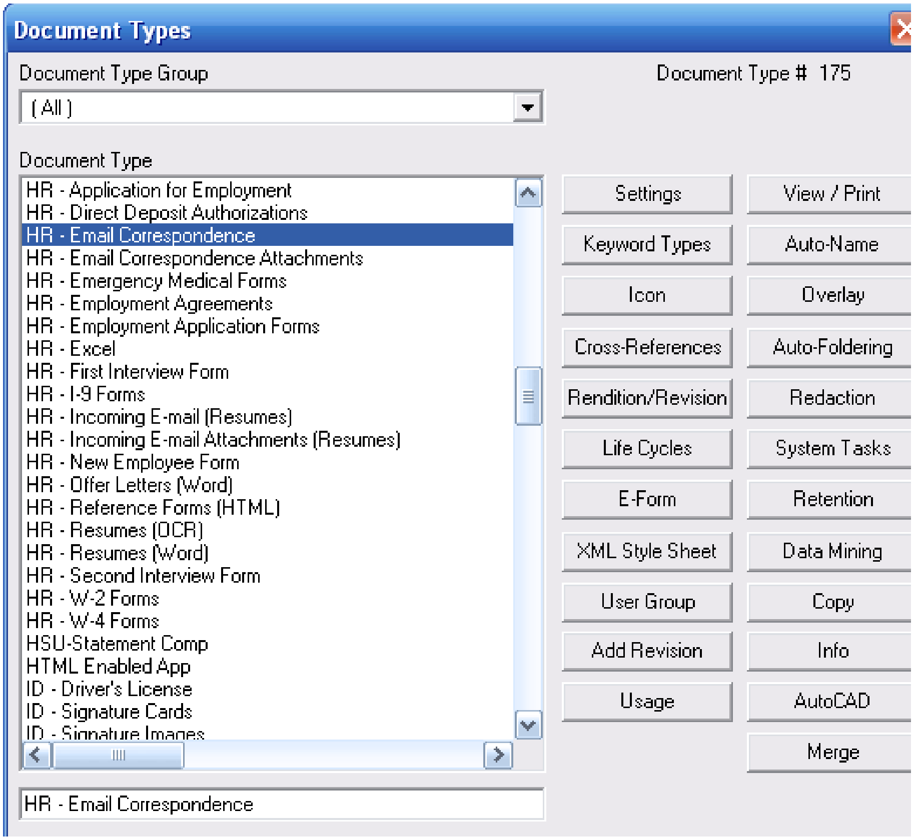
Faxing E-Forms

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.** 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. The following describes how to configure a low-resolution image that looks good when faxed.

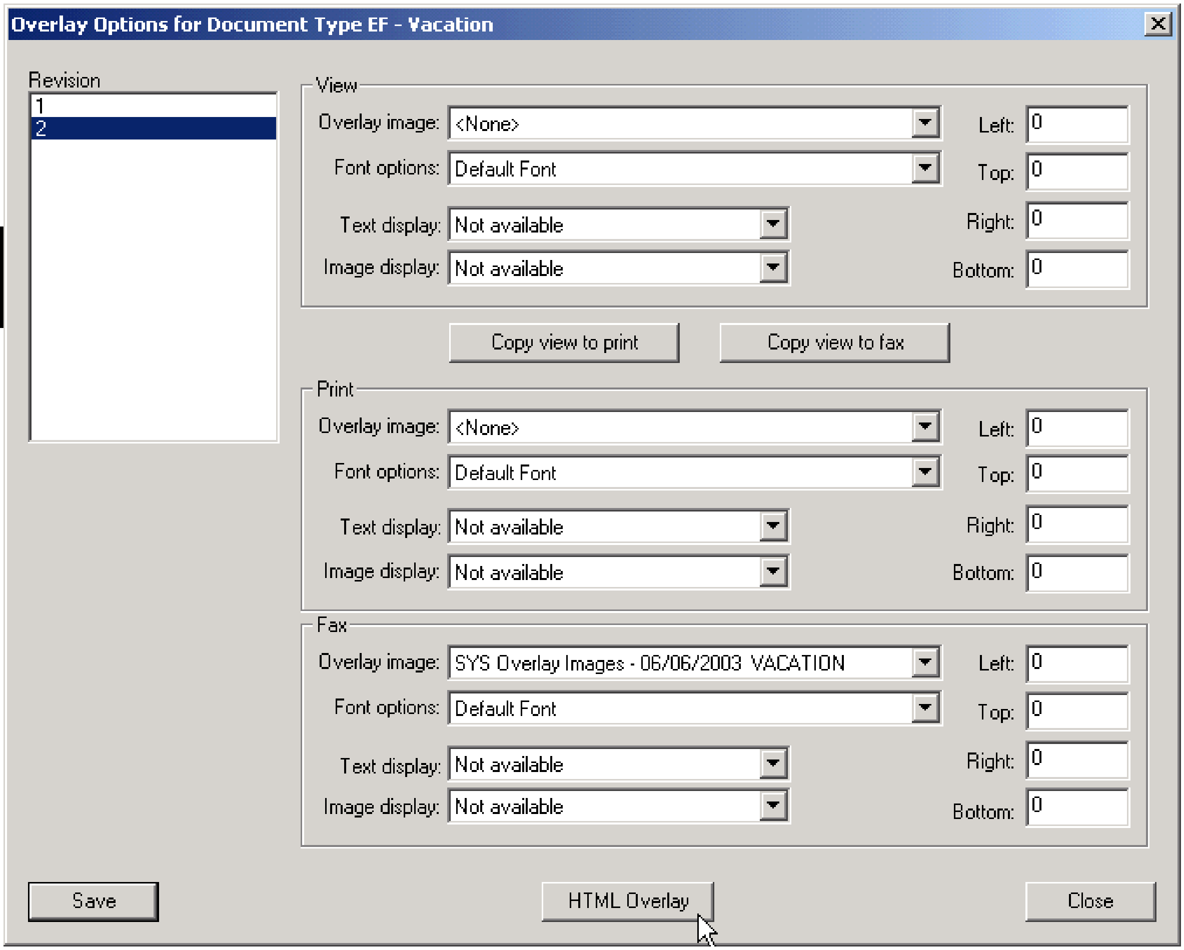
First, print the E-Form to be configured for faxing with all the value fields blank. Scan the printed E-Form to create a low-resolution image (100 dpi). This file will be used as a template when the E-Form is faxed. In the Client module import the file into the **SYS Overlay Images** Document Type.

In the Configuration module, select **Document | Document Types**. The **Document Types** dialog box displays. Highlight the Document Type of the E-Form to be configured for faxing. Select the **Overlay** button to open the **Overlay Options** dialog box.

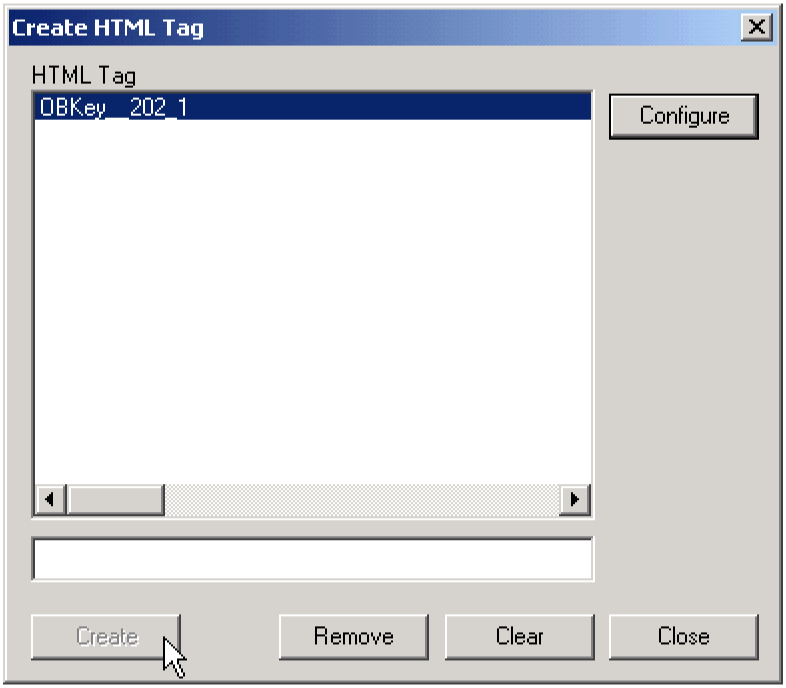


2

In the **Overlay Options** dialog box select the current revision. In the **Fax** section select the overlay image of the E-Form. Select the **Save** button to save the selection. Click the **HTML Overlay** button to open the **Create HTML Tag** dialog box.

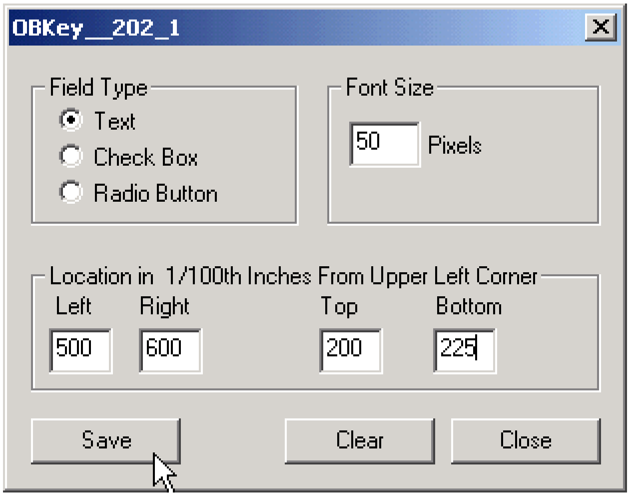


In the edit field of the **Create HTML Tag** dialog box, enter the name of a field to print. The name must exactly match the corresponding tag name in the E-Form HTML template. Select the **Create** button to save the tag and open the field configuration dialog box.



2

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. In the **Font Size** section enter the size, in pixels, of the font to print with (50 pixels is a good starting size). Finally, 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 example, the value contained in an edit field will print with a height of 50 pixels. The edit field is located 5 inches from the left edge of the E-Form and is 1 inch wide. It is located 2 inches from the top edge of the E-Form and is ¼ inch high.



Configure each field to print on the form. Test the configuration by printing the E-Form from the client module. Select Fax Compatible, but print to a regular printer rather than a Fax. Adjust font size and positioning as needed in 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 #** keyword type. See the documentation for Institutional Databases for more information about configuring E-Forms for Institutional use.

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.

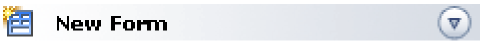
USAGE - WEB CLIENT

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 way.

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.

1. Select **New Form** from the **Document Retrieval** drop-down.
2. This displays the **New Form** panel, which lists the E-Forms that are available to you.



3. Select the E-Form that you want to complete and submit. The E-Form displays.
4. Complete the form.

Caution: 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.

5. When you have finished filling out the form, submit the form.

   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, the system asks you if you wish to complete another one.
6. To complete another form of the same type, click **Yes**. Click **No** if you are finished completing E-Forms of this type.

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

USAGE - JAVA WEB CLIENT

Usage

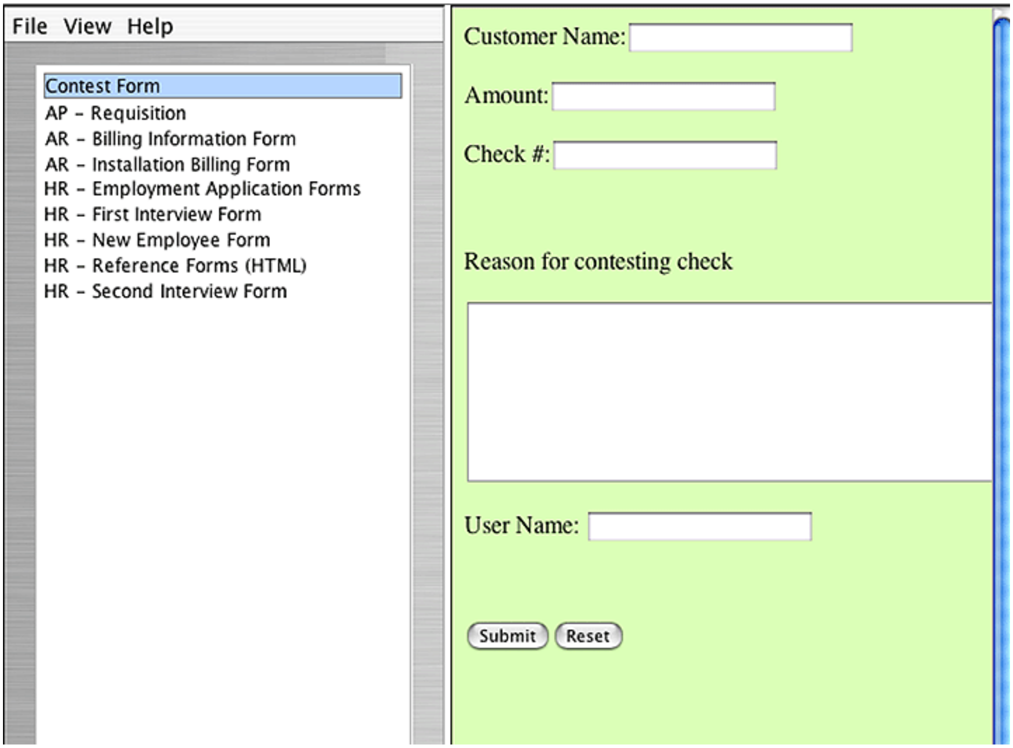
E-Forms are forms that you complete and submit to the system electronically. E-Forms can be set up to support any business or organizational activity that requires the entry, editing, routing, and approval of information. Because the system maintains complete audit trails for E-Forms, they can be tracked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

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 Java Web Client:

4

1. Select **View | New Form**.



2. From the left pane, select the form you want to complete.
3. Enter the appropriate information.

Caution: 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.

4. Submit the form. The **Confirm New Form** dialog box is displayed asking **Would you like to complete another form?** Click **Yes** to create a new form. Click **No** if you are done creating forms.

In addition, you can open existing E-Forms and edit them. Submit the E-Form when charges have been made appropriately. If you do not have rights to modify the E-Form, the fields will be disabled, as well as the submit button.

USAGE

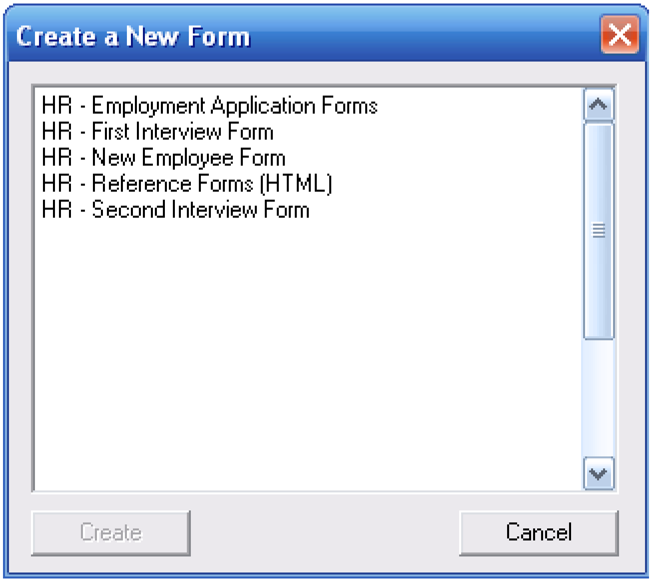
E-Forms are forms that you complete and submit to the system electronically. E-Forms can be set up to support any business or organizational activity that requires the entry, editing, routing, and approval of information. Because the system maintains complete audit trails for E-Forms, they can be tracked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

Creating New E-Forms

The Desktop connections allows for the creation and storage of E-Forms into OnBase. To create a new E-Form:

1. From the system tray icon right-click menu, select **New Electronic Form**.

   The **Select a Form to Create** dialog box is displayed.



2. Select a form from the list of available E-Forms.

3. Click **Create**. The selected E-Form is displayed.
4. Enter the appropriate information and submit the E-Form. The form is stored in the OnBase database. If the form is associated with a Workflow life cycle, it will enter that life cycle.

Note: To submit E-Forms, your system must be properly licensed and configured for the E-Forms module. See the OnBase Client and E-Forms Help for more information.

Caution: 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.

Note: For date formatting, when using the Desktop in Remote Mode, the server's system locale is used. When using the Desktop in Local Mode, the system locale of the workstation is used.

Note: If your system is also licensed for the Signature Pad Interface module, you can apply a signature on an E-Form. See your System Administrator to obtain licensing and signature privileges.

Note: When using Remote Mode Desktop, the **Add Note** button will not add notes to the E-Form before the E-Form is submitted.

HTML Documents

HTML documents are displayed in the Desktop Viewer with the same right-click options as in Internet Explorer.

Updating E-Forms

To update an E-Form:

1. Retrieve the E-Form you want to update.
2. Select the text in the field you want to update and type the new text.
3. Repeat step 2 for any other fields you want to update.
4. Click the submission button. Your E-Form is saved with updates.

Note: Currently, the Desktop does not support forms containing the pound sign character (#), check box form fields, or radio button form fields.

Note: If your system is also licensed for the Signature Pad Interface module, you can apply a signature on HTML forms. See your System Administrator to obtain signature and licensing privileges.

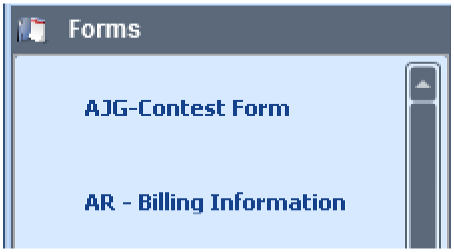
CREATING E-FORMS

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 way.

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:

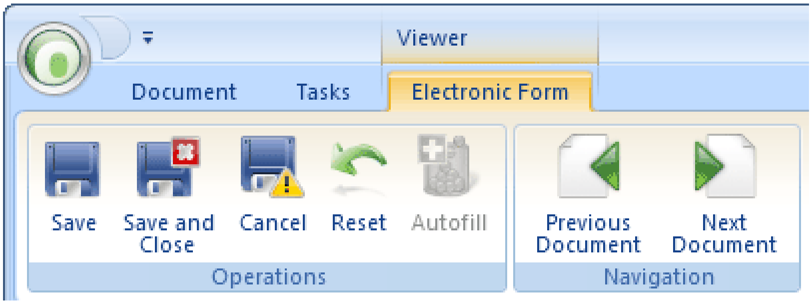
1. In the **Create** menu, click **E-Forms**. The **Forms** panel is displayed.



2. Click on the E-Form you want to create. The form will display in the right pane.
3. 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. Click **Yes** to create another form of the same type or click **No** if you are finished creating forms.

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 using:

Button	Description
Save	The **Save** button submits the E-Form. This button is available when creating or editing an E-Form.
Save and Close	The **Save and Close** button submits and closes the E-Form. This button is available when editing an E-Form.
Cancel	The **Cancel** button 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	The **Reset** button will clear all changes made to the form fields and restores the form fields to their previously saved values.
Autofill	This is reserved for future functionality.
Previous Document	Opens the previous document in the Document Retrieval list.
Next Document	Opens the next document in the Document Retrieval list.

Configuration

Configuring an E-Form is a 4-step process:

1. Create an HTML form with an external editor. (Use the E-Form Validator to verify the HTML.)

Note: To ensure proper encoding, map all Keyword Types associated with a 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.

2. From the Client module, import the HTML Form into the **SYS HTML Forms** Document Type.

Note: If this is a unicode HTML Form, ensure that the form in imported with an HTML Unicode file 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.

Caution: UTF-16 E-Forms are not supported in the OnBase Client.

Note: If using a Cascading Style Sheet for E-Forms in the Web Client, the form creator is responsible for ensuring font accuracy.

Caution: Encrypted keywords are not supported in E-Forms when using the Unity Client.

Caution: 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

Note: In order for an E-Form to function correctly in a Japanese environment, the E-Form must be encoded for UTF-8.

E-Form templates are written in HTML. HTML code can be written directly in an program such as Notepad®.

Note: HTML editors, like FrontPage®, are not recommended because they can add extraneous information.

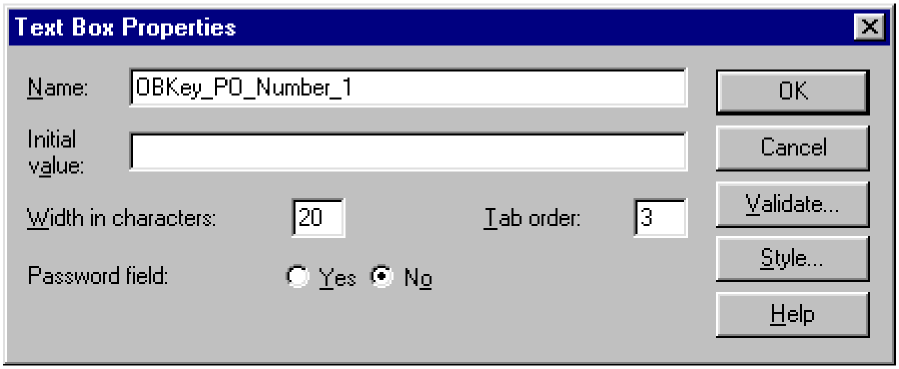
E-Form Templates contain input fields that are mapped to document Keyword Types and other system values. E-Forms can also contain information that is not mapped to keywords or stored in the database. This information is stored with the form and is available for viewing, though the system does not search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

Note: 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 part of forms creation is the proper mapping of form fields to document Keyword Types or system values. See "Mapping Form Fields to Keyword Types" on page 24 for details.

The way to identify 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 FrontPage™. When the form is complete, save it as an HTML file.

Example:



Note: 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 & Receiving** in HTML code. In this case, a search for the Keyword Value **Shipping & Receiving** would not return expected results.

Note: Some HTML editors store extra spaces as ** ** 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.

Note: 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.

Caution: Fields disabled using JavaScript are not supported and any attempt to used them will yield unexpected results.

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

Using Form Tags on 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 that the form method = post for the E-Form to submit correctly into OnBase. If this is not included, the E-Form's **Submit** button will act as a reset button, instead of actually submitting the form into the database.

An example of 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

Caution: 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 checked or unchecked. 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 unchecked.

Caution: 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.

Note: To stay within the maximum number of characters allowed for a keyword value and avoid truncated values, the administrator creating the E-Form 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.

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(s) that cannot be used in an HTML attribute without being escaped, the Keyword Type number should be used instead.

Note: 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.

7

Note: 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, as form objects containing OBKey formats are controlled by Keyword Values. When items need to remain static, and are not dependent upon Keyword Values, do not map fields as Keywords or to OBKey formats.

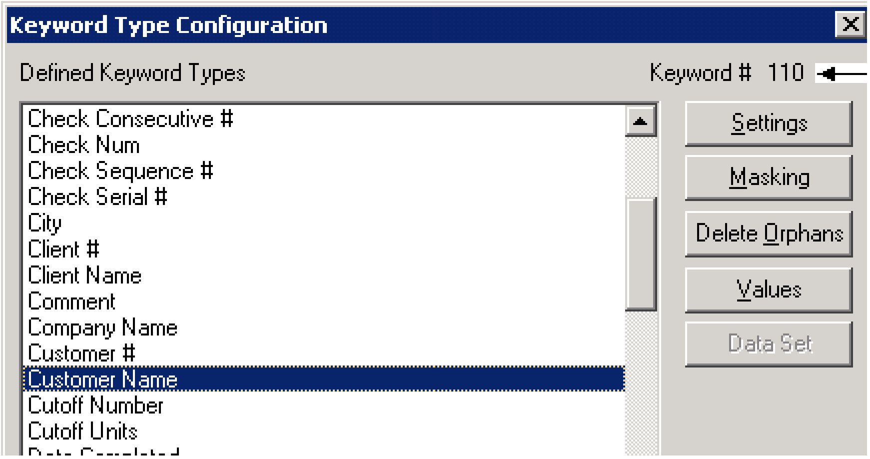
For example, if an HTML document contains check boxes that are checked, importing the document into the system may clear the check boxes if they are mapped using OBKey formats and the Keyword Values do not validate the check.

Caution: 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. Date & Time formatted Keyword Types are not supported for check box fields.

KEYWORD TYPE NUMBER

Format: **OBKey__KeywordNumber_#**

The Keyword Type Number is displayed by default on the upper, right-hand corner of the **Keyword Type Configuration** screen, as shown below.



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 keyword Customer Name would be mapped as **110_1**, the second occurrence would be mapped as **110_2**, etc.

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

Note: When using a currency Keyword Type, the E-Form will format 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

Format: **OBKey_Keyword_Type_Name_#**

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**, etc.

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

Note: When using a currency Keyword Type, the E-Form will format 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.

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:

```
<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.

Caution: 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.

Caution: 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 autofill 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, denoted by the _1, of the Multi-Instance Keyword Type Group 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.

7

Using Auto-Incrementing Keywords

Auto-incrementing keywords will 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 will automatically increase by one.

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 ID type 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 will initially always be mapped as **1**.

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

Note: Because the keyword will automatically increment regardless of whether the E-Form was 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, etc.

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 using the following code in the table below. Data Set values are appended to any existing values associated with the drop-down field. External Keyword Data Sets are also supported. See the System Administration manual or help files for more information regarding Keyword Type Data Set Configuration.

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

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.

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

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. **Caution:**If creating scripts to manipulate E-Form data, please note that the **OBDataset_** mapping is converted to an **OBKey_** mapping in the OnBase Client. **Caution:**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 will be 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**, etc. **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="OBKey__###_1">
```

Multiple Selection List:

```
<select size="1" name="OBKey_KeywordName_1" multiple>
```

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 will display 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.

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.

7

<OPTION>	This tag specifies what operators are available for each **OBOperKey** instance. The following operators and their corresponding codes are available for use: **=** (equals) **<** (less than) **<=** (less than or equal to) **>** (greater than) **>=** (greater than or equal to) **<>** (Not equal to) There is not space between the codes. **" "** (Literal) There is a space between each " Example: <OPTION><</OPTION> <OPTION><=</OPTION> <OPTION>></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. **Note:**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 'Use OR for duplicates' is not selected, the OBOperator tag is respected, and users can refine their searches using the AND or OR operator.

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

Note: 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:

```
<HTML><HEAD>
</HEAD>
<BODY class=rcmclass>
<FORM name=form action="" method=post>
<TABLE class=rcmclass>
  <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>
    <TD align=middle
      colSpan=4><INPUT type=submit value=Query
name=OBBtn_Yes>  <INPUT type=submit value=Cancel
name=OBBtn_Cancel><BR></TD></TR></TBODY></TABLE><BR><!-- </
FORM></BODY></HTML>
```

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". 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

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.

Note: OBDocumentDate and **OBFromDate**/**OBToDate** should not exist on the same Custom Query.

Note: 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.

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.

Document Property	Description
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.
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:**An EDM Services license is required for revising.

Note: 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.

7

Note: 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. When using the Desktop in Remote Mode, the server's system locale is used. When using the Desktop in Local Mode, the system locale of the workstation is used.

Note: These property names are not case sensitive.

User Properties

The following properties reflect 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 an E-Form, they will not be added to the E-Form upon creation. All properties will 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_CurrentUserID	Reflects the User ID of the currently logged in user.
OBProperty_CurrentUserName	Reflects the OnBase user name of the currently logged in user.
OBProperty_CurrentUserRealName	Reflects 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 will be left blank.
OBProperty_CurrentUserDisplayName	Reflects 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 will display the OnBase user name.

Note: These property names are not case-sensitive.

Buttons

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

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.

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

Button	Description
OBBtn_CrossReference	Selecting a button of this type will cause 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">** **Note:** This button type only functions after a form has been submitted. **Note:** Only one cross-reference button can be configured from an E-Form. **Note:** 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. **Note:** 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. **Caution:** This button is not supported in the Java Web Client.

7

7

Button	Description
OBBtn_Keyset	**Note:**The **OBBtn_Keyset** button is for backward compatibility and should not be used to configure new forms. It is recommended that new forms use the **OBBtn_ExpandKS###** button. Selecting a button of this type causes all Keyword Sets on a newly created E-Form to be expanded (when one or more AutoFill Keyword Sets are properly configured and populated). This button must be **type=submit**. When a primary keyword is entered on the form and the **OBBtn_Keyset** button is pushed, the corresponding keyword values are filled in on the form. When a keyword type that has not been mapped to form fields is populated, the keyword value is not applied to the form and is not saved to the database when the **Save** or **Yes** button is pressed. 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_Keyset">** **Note:**The primary Keyword Type of the AutoFill Keywords Set should not be of the date or date & time format if this button is on an E-Form to be used in the Web Client. **Note:**This button will not function when used in conjunction with the following Workflow actions: Form - Display HTML Form for This Document, Form - Display HTML Form for Related Document, and Form - Display E-Form for Input. **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.

Button	Description
OBBtn_Keyset (cont.)	**Caution:** In the Web Client, OBBtn_KS###, OBBtn_ExpandKS###, and OBBtn_Keyset 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:** This functionality cannot be used in conjunction with Multi-Instance Keyword Type Groups. You may only specify one instance of each AutoFill Keyword Set to be displayed on an E-Form. **Note:** Despite the AutoFill Keyword Set configuration, only a single instance of an AutoFill Keyword Set may be selected to be associated with an E-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. **Caution:** This button is not supported in the Java Web Client. **Caution:** This button is not supported in the Workflow module.

Button	Description
OBBtn_KS### **(Where ### represents an AutoFill Keyword set number)**	Selecting a button of this 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 pushed, the corresponding Keyword Values for the specified keyword set are filled in on the form. For example, pressing 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 MIKGs, 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: Form - Display HTML Form for This Document, Form - Display HTML Form for Related Document, and Form - Display E-Form for Input. **Caution:** In the Web Client, OBBtn_KS###, OBBtn_ExpandKS###, and OBBtn_Keyset 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.

Button	Description
OBBtn_ExpandKS### (cont.)	**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: Form - Display HTML Form for This Document, Form - Display HTML Form for Related Document, and Form - Display E-Form for Input. **Note:** 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###, OBBtn_ExpandKS###, and OBBtn_Keyset 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 **Form - Display HTML Form for This Document** and **Form - Display HTML Form for Related Document** Workflow actions. In addition, this button is not supported when using it in the Java Workflow Client when using **Form - Display E-Form for Input** as an ad hoc task. **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.

7

Button	Description
OBBtn_Yes **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.

Button	Description
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.

7

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 **SYS - 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.

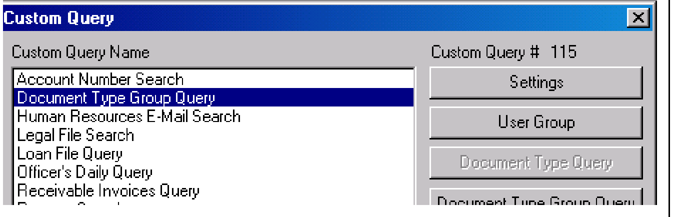
7

Button	Description
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:** This button is not supported in the Java Web Client.

Button	Description
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. At least one AutoFill Keyword Set must be defined on the form. (See “OBBtn_Keyset” on page 42 or See “OBBtn_KS###” on page 44.) 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:** If more than one AutoFill Keyword Set type exists on the E-Form, they cannot share Keyword Types. **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 Java Web Client.

7

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 (e.g. **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.  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. **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. **Caution:** This button is not supported in the Java Web Client.

Button	Description
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 or Java Web Client.

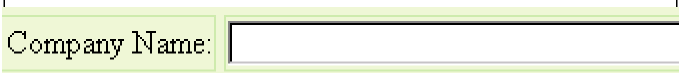
Reset Buttons	Description
Type = reset	Any button of the **type=Reset** will clear all changes made to the form fields and restores the form fields to their previously saved values. The E-Form remains open. This button can be named anything. **Note:**Once an AutoFill Keyword Set is initiated, the reset 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**.

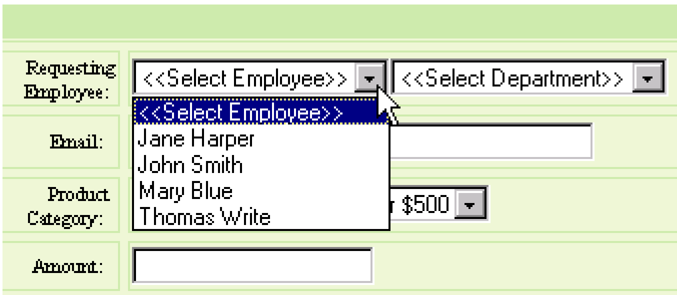
Form Fields

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.

Note: Form fields cannot be named with the OBKey convention.

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. 
Radio Button	Use radio buttons when only one option should be selected from a group.  **Note:**A Keyword Type that uses a radio group can only appear once per E-Form.
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 select 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 exampe: <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>
 	

Form Field	Description
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.
Check Box	Use check boxes for optional items. The E-Form user can select and clear multiple check boxes. 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:** Date & Time formatted Keyword Types are not supported for check box fields.
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.

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 OnBase on HTML Documents

If an image is already stored in OnBase, you can include it on OnBase HTML documents, Custom Queries, and E-Forms that are displayed in the OnBase Client and Web Client. This feature lets you maintain an image on a document or form by updating the image in OnBase. 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 OnBase images on HTML documents by modifying the Keyword Values on the referenced images.

Note: This feature is not supported in the Java Web Client. HTML documents, Custom Queries, and E-Forms accessed from this application will not display images referenced within OnBase.

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.

Here is an 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.

7

Additional Parameters

You can use an image's document date and page number to further refine a reference to an image. A **KeyTypeNum** of **-6** indicates that the **KeyValue** represents the Document Date of the target document. The date must be in the form of YYYYMMDD. Separators are optional. For example, February 7, 2009 could be represented as 20090207, 2009/02/07, or 2009-02-07. The following example shows how an image could be referenced using a document date:

<img alt="mz:101;212:LOGO;**-6:20090207;**">

A **KeyTypeNum** of **-10** indicates that the **KeyValue** represents the page number of the image to be displayed. If a **KeyTypeNum** of **-10** is not specified, the first page of the referenced document is displayed by the default. Page numbering starts at 0, so the second page in a document is represented by the number 1. The following example shows how the second page of a document could be referenced:

<img alt="mz:101;212:LOGO;**-10:1;**">

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 an HTML form for Signature Capture

HTML forms must be properly configured for capturing signatures using Signature Pad Interface. The user must sign in an image placeholder, in a pre-designated area. When creating the HTML form, ensure the following syntax to designate the signing area is used.

To create a form with one Signature Placeholder, insert this syntax in the form where the signature placeholder will reside.

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

To create a form with two Signature Placeholders, insert this syntax in the form where the signature placeholder will 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.

You can also create a user friendly label for the placeholder. To create a user friendly label, you must place the value you want to be displayed in the **alt** value for the placeholder. 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 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 on the form. If the label value has a space in it, the entire value for the **alt** attribute must be surrounded by quotation marks, as shown in the second example. If the label does not have a space in it, the quotation marks are optional.

E-Form Validator

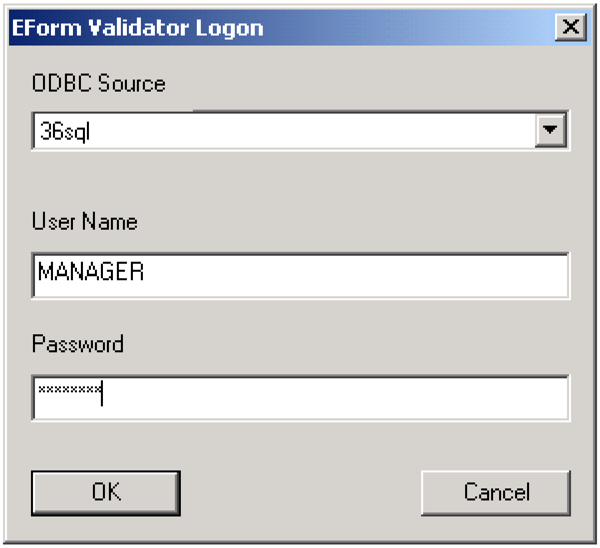
The E-Form Validator is an external program initiated outside of the Configuration module using the OnBase Client API. It checks for proper Keyword Type configuration on any HTML form that is intended for use in OnBase. The E-Form Validator also validates the following properties:

- OBProperty_CurrentUserLocale
- OBProperty_CurrentUserID
- OBProperty_CurrentUserName
- OBProperty_CurrentUserRealName

- OBProperty_CurrentUserDisplayName
- OBProperty_DateStored
- OBProperty_TimeStored
- OBProperty_DocumentDate
- OBProperty_UserName
- OBProperty_ItemNum

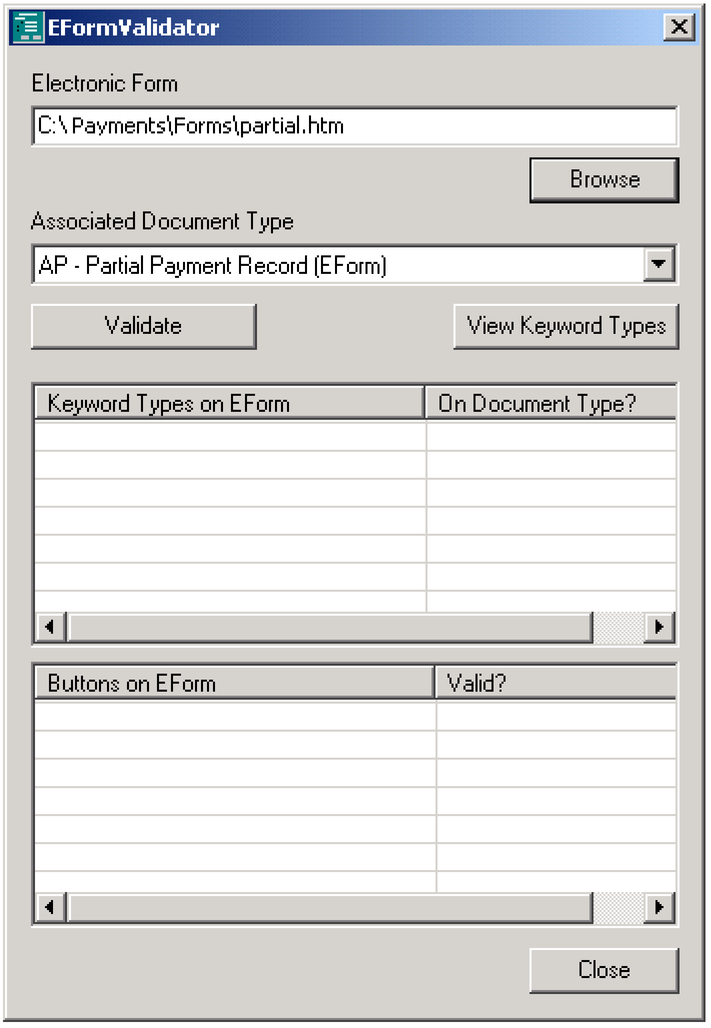
Performing E-Form Validation

1. Double-click the EFValid.exe icon on the desktop to initiate the E-Form Validator.
2. The **EForm Validator Logon** dialog box displays. This menu requires the user to log onto the database containing the E-Form Document Types to be used for comparison when checking for proper Keyword Type configuration on the E-Form.



3. Enter the appropriate values, then click **OK**.
    a. Choose the database that contains the Document Types/keywords for validation from the **ODBC Source** pull-down. If the OnBase Client is already running on the computer against a database, E-Form Validator will use this database and not the one selected.
    b. Enter a **User Name** from any valid user account for the selected database.
    c. Enter the appropriate **Password** for the user account. The **EForm Validator** dialog box displays. The parameters supplied at this dialog box identify both the E-Form that will be validated and the database parameters that will be used for comparison.

7



Identify the E-Form to be validated by supplying the path to the .htm file, or using the **Browse** button to navigate to the location.

4. Select the Document Type that contains the Keyword Types to be used for comparison when validating the E-Form content. Make the appropriate selection at the **Associate Document Type** pull-down.

   If desired, click **View Keyword Types** to display the Keyword Types currently configured for the selected Document Type. This ensures that the expected Keyword Types will be used in the comparison. Only currently configured Keyword Types will be used in the comparison, and the validation results will only include these Keyword Types.

5. To initiate the validation, click **Validate**. The bottom of the dialog box displays the results of the validation.

   The **Keyword Types on EForm** section of the dialog box lists all Keyword Types detected as configured for the E-Form. The results of how these keywords are mapped to the Document Type are reflected in the **On Document Type?** section of the dialog box. The **Button on EForm** section of the dialog box lists all buttons that have been configured for use on the E-Form. The results of how the buttons are mapped for use with OnBase are reflected in the **Valid?** section of the dialog box.

   The following results can appear:

On Document Type?	Description
Yes	This Keyword Type is properly configured on the E-Form and also exists on the Document Type.
No	This Keyword Type is configured on the E-Form, but is not defined for the Document Type.
Error	This Keyword Type is improperly configured on the E-Form. The syntax, as stored for the keyword in the .htm file, is echoed in the E-Form Validator dialog box. Review this syntax and make the appropriate changes. Refer to Mapping Form Fields to Keyword Types on page 24. **Note:** E-Form Validator only looks for words starting with **OBKey** or **OBBtn**. It does not look for **OBBKey**, **OBtn**, etc. In case of such errors, the misspelled Keyword Type does not display in the dialog box.
Suspect	Indicates that a syntax error exists on the E-Form. For example, a single underscore in a keyword number, instead of a double underscore.

Valid?	Description
Yes	This button is properly configured on the E-Form.
Error	This button is improperly configured on the E-Form. The syntax, as stored for this button in the .htm file, is echoed in the E-Form Validator dialog box. Review this syntax and make the appropriate changes. Refer to Using Multi-Instance Keyword Type Groups on page 29Using Multi-Instance Keyword Type Groups.

Note: E-Form Validator is not guaranteed to find all possible errors. The E-Form Validator helps identify the most common configuration errors. The only true test of whether an E-Form is properly configured is that it works as expected.

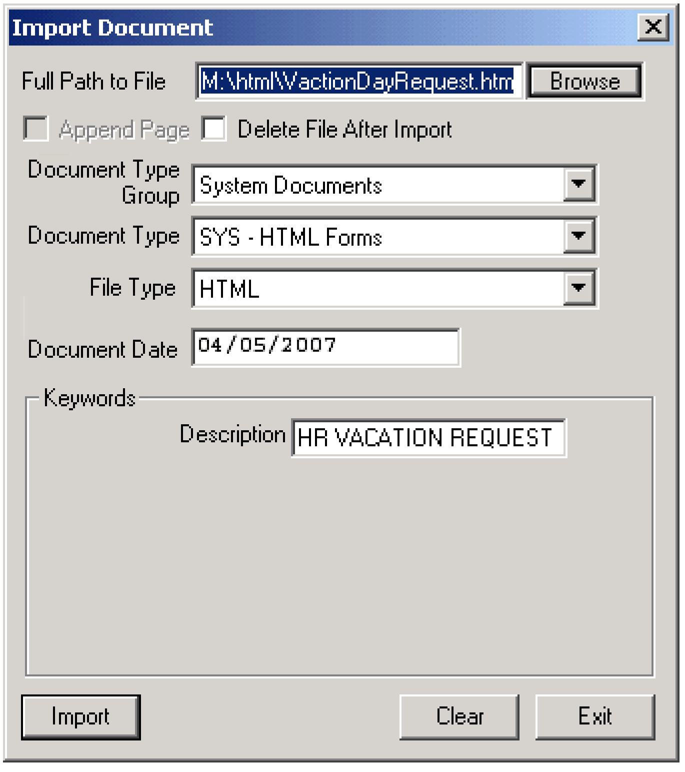
Note: E-Forms do not validate keywords as the keywords are entered into the form. The E-Form Validator is used to make sure the keywords are configured properly for the form template, not that each keyword is entered correctly on individual E-Forms.

7

Importing The HTML E-Form Template into OnBase

To use an HTML E-Form template in OnBase, it must be imported through the Client module. The HTML form is stored into the system **SYS HTML Forms** Document Type. The form will not change once it has been imported.

1. In the Client module select **File | Import** from the menu bar. The **Import Document** dialog box is displayed.



2. In the **Full Path to File** text box, enter the location of the HTML form or click **Browse** to search.
3. Optionally, select the **Delete File After Import** check box to delete the file from its original location after it is imported.
4. From the **Document Type** drop-down list, select the **SYS HTML Forms** Document Type. Documents store in the **SYS HTML Form** Document Type can be associated with multiple Document Types. When the Document Type has been selected, the **File Type** and **Document Date** text boxes are filled in automatically.
5. 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.
6. Click **Import** to store the HTML form into the **SYS HTML Forms** Document Type.
7. Once an HTML form has been imported into the system, it can be applied to a Document Type.

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 submittance of an E-Form, the system's default behavior is to prompt the user with a **Do you wish to create another new document?** message. If you do not wish for this message to be displayed for this Document Type, check the **Don't prompt for new form after submit** check box. This is particularly useful for organizations that have no need to submit multiples of the same form at the same time.

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 69 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 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.

To create a Document Type revision:

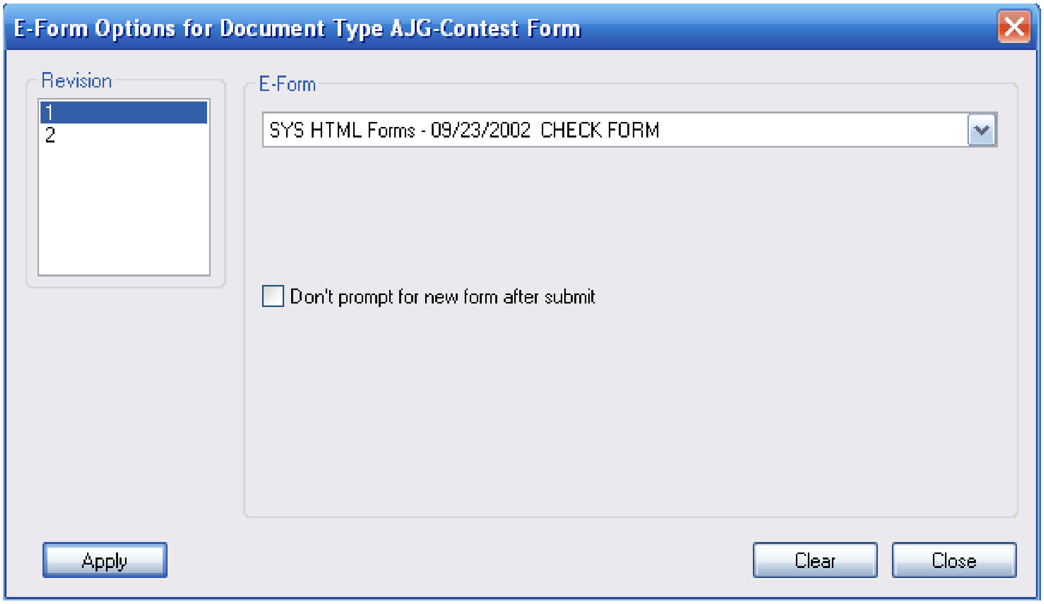
1. In the Configuration module, select **Document | Document Types**. The **Document Types** dialog box displays.
2. Select a **Document Type** and click **Add Revision**.
3. A warning dialog box displays. Click **OK** to create a revision of that Document Type. Click **Cancel** to cancel.

   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.
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.

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



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

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 " "> ]>
```

An example of an XML declaration is:

```
<?xml version="1.0" encoding="UTF-8" ?>
```

Associating XML Style Sheets at the Document Type Level

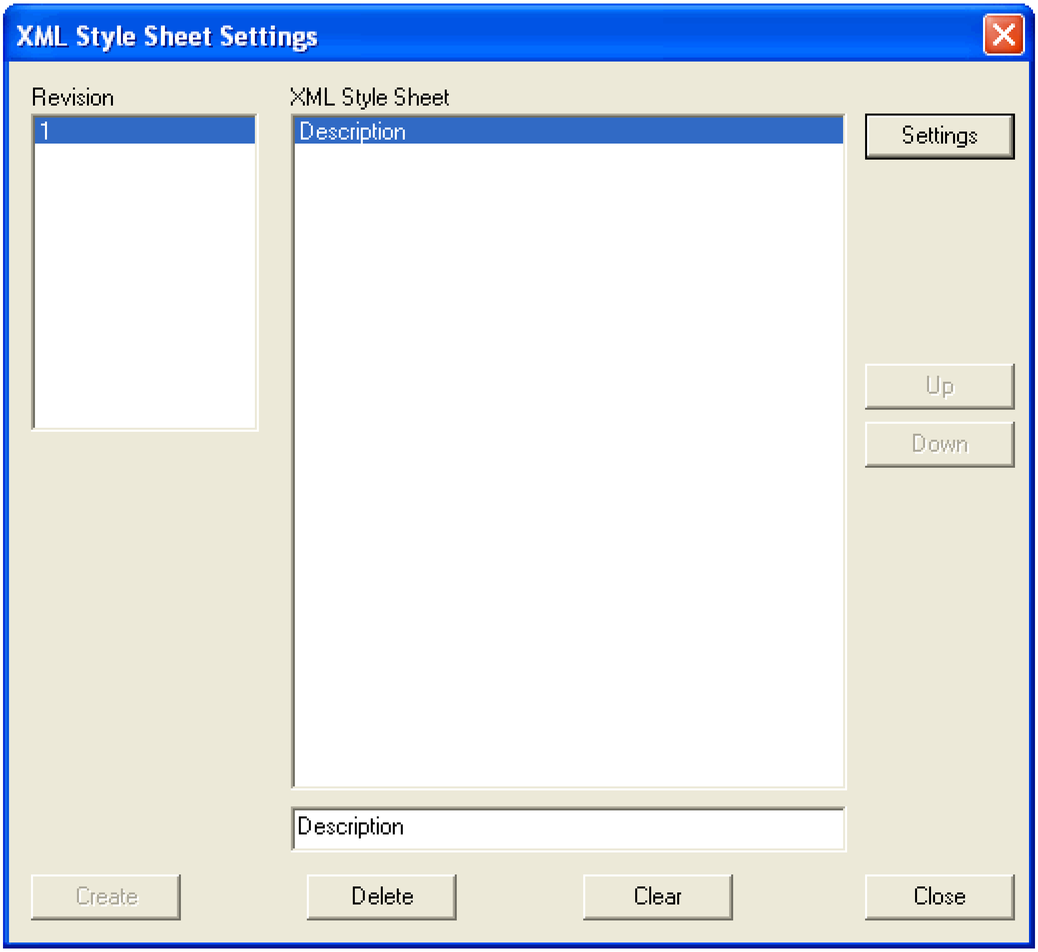
You can associate multiple style sheets with a single Document Type, providing the you with the ability to select the style sheet they want to use when viewing, printing or e-mailing the document.

To associate a style sheet with a Document Type:

1. From the OnBase Configuration module, click **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**.

Note: The **XML Style Sheet** button is only enabled when your OnBase solution is licensed for one of the following: E-Forms, Tag Import Processor, EDI 835 EOB Processor, EDI 837 Processor, EDI 810 Processor, Integration for IBM Lotus Notes, Fax Import Integration for Captaris RightFax, or Integration for Microsoft InfoPath.

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 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**.

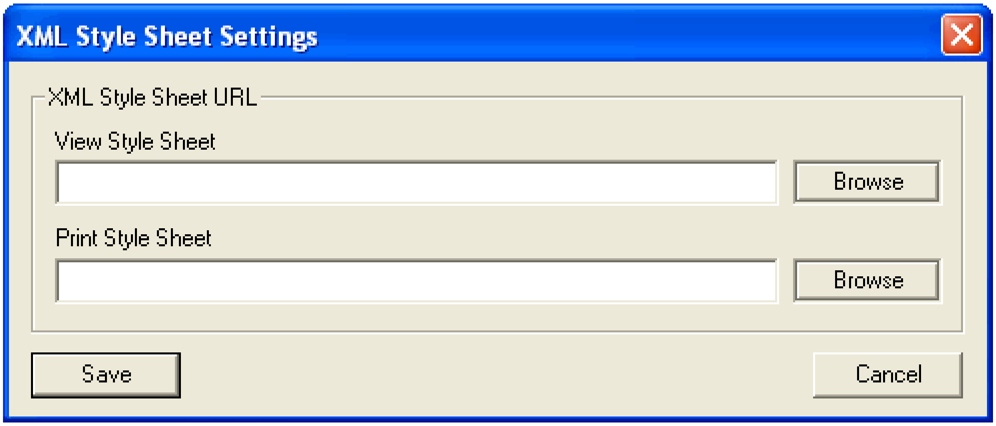
Tip: OnBase allows you to assign different style sheets to different revisions of the document to account for changes in the appearance of the document, such as if a style sheet is modified or updated.

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 URL** dialog box 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 3-8 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.

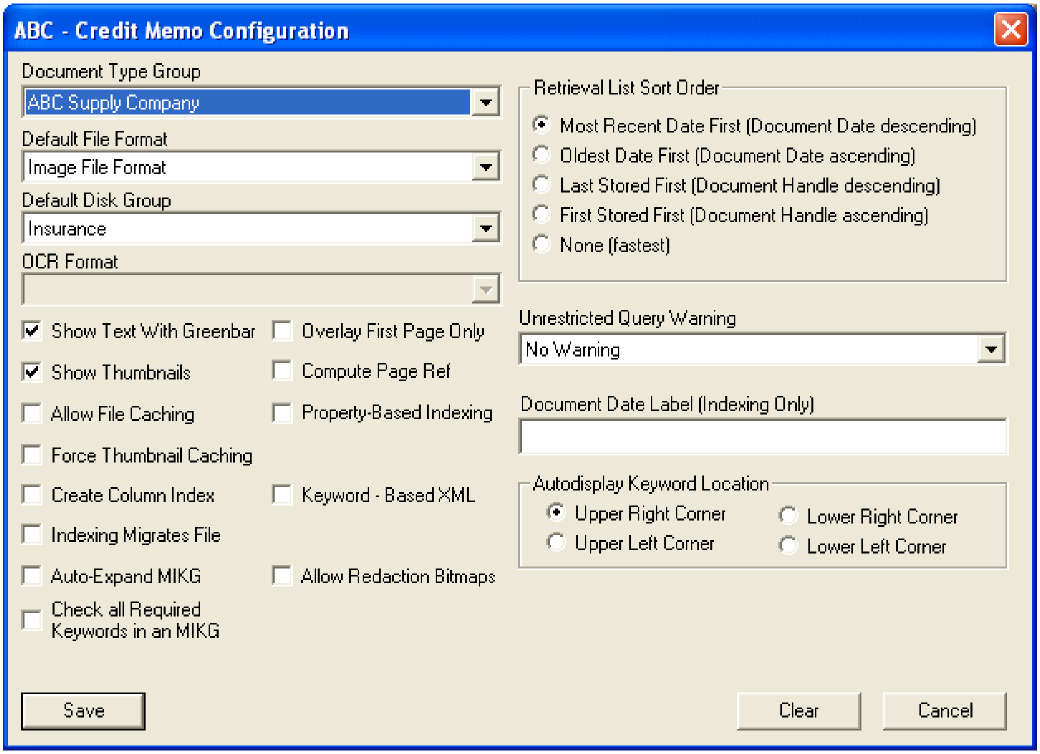
For example, an Accounts Receivable department prints and mails color-coded invoices depending on how past due an invoice is. To enable this, they added a **Status** Keyword Type with four different possible Keyword Values (e.g., **Current**, **30 Days Past Due**, **60 Past Due**, **90 Days Past Due**) 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 the OnBase Configuration module, 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.

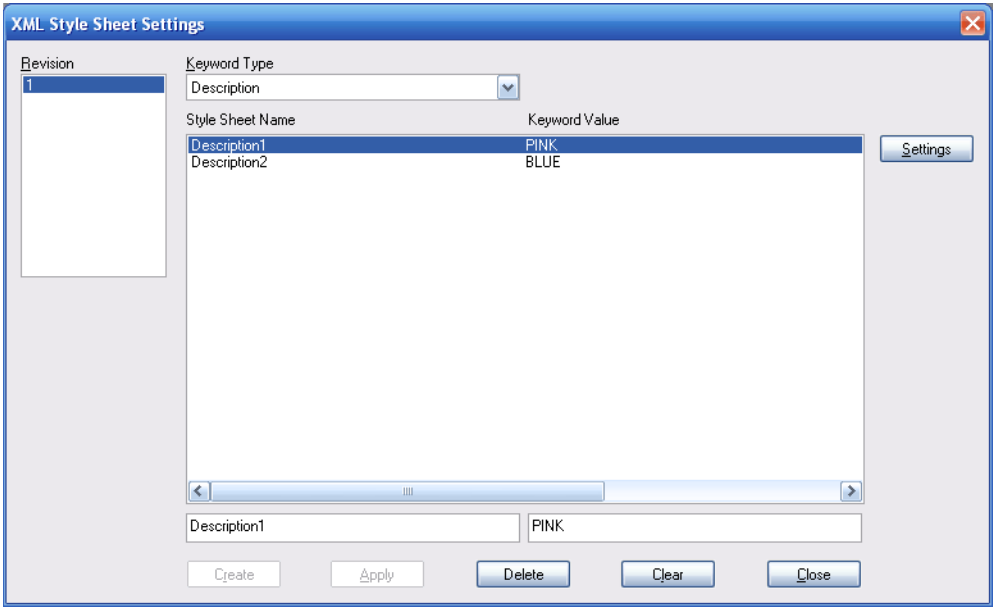
The **<Document Type Name> Configuration** dialog box is displayed



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**.

Note: The **XML Style Sheet** button is only enabled when your OnBase solution is licensed for one of the following: E-Forms, Tag Import Processor, EDI 835 EOB Processor, EDI 837 Processor, EDI 810 Processor, Integration for IBM Lotus Notes, Fax Import Integration for Captaris RightFax, or Integration for Microsoft InfoPath.

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



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 Associate XML Style Sheet by Document Type above 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**.

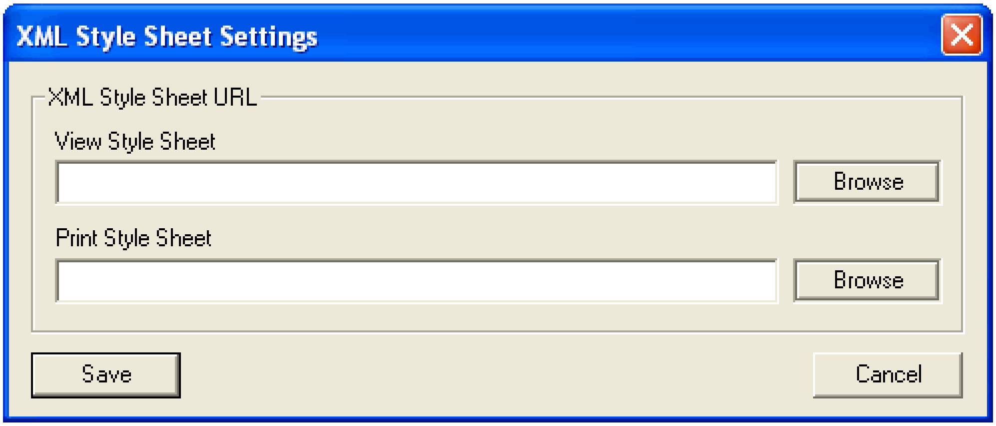
Tip: OnBase allows you to assign different style sheets to different revisions of the document to account for changes in the appearance of the document, such as if a style sheet is modified or updated.

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 URL** dialog box 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 7-12 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

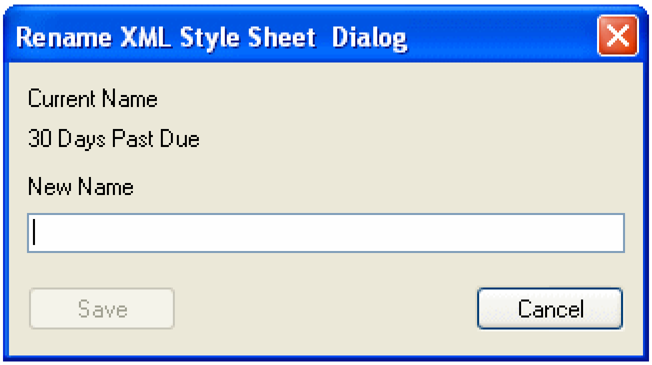
To rename an XML style sheet:

1. From the OnBase Configuration module, click **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**.

Note: The **XML Style Sheet** button is only enabled when your OnBase solution is licensed for one of the following: E-Forms, Tag Import Processor, EDI 835 EOB Processor, EDI 837 Processor, EDI 810 Processor, Integration for IBM Lotus Notes, Fax Import Integration for Captaris RightFax, or Integration for Microsoft InfoPath.

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

3. Double-click on 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

Tip: 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">
```


Tip: Use a text editor, like Notepad®, to edit and save the revised file as plain text. HTML editors, like FrontPage®, are not recommended because they can add extraneous information.

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**.

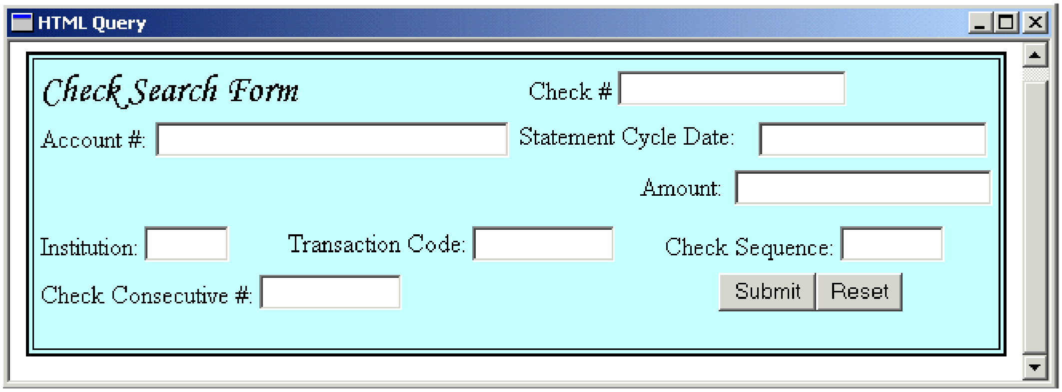
Tip: To strip out non-essential data from forms before storage, re-configure the original E-Form Document Type to use a default file format of Virtual Electronic Form.

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.

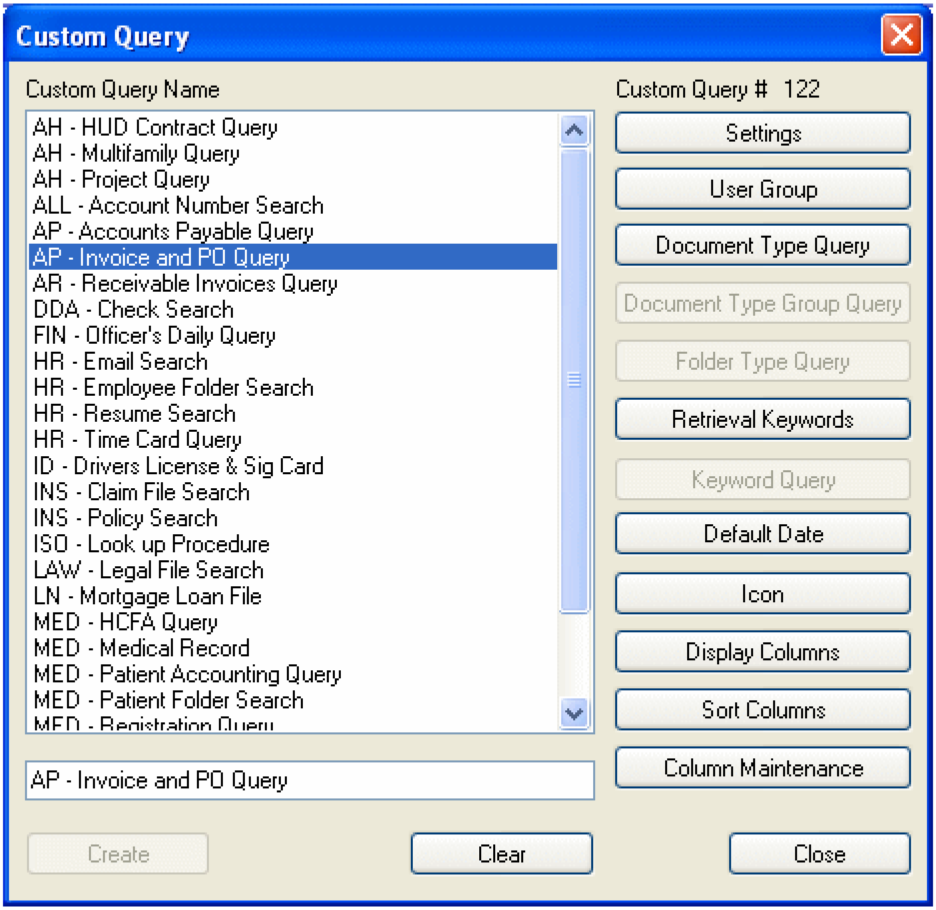


Before you begin configuring an HTML Custom Query, please note the following requirements and limitations

- 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 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.

1. Create and configure an HTML form with appropriate fields for document or folder retrieval. Ensure you follow the guidelines listed above. For more information on creating and configuring forms, see the E-Forms documentation.
2. In the Configuration module, select **Queries | Custom Queries**.

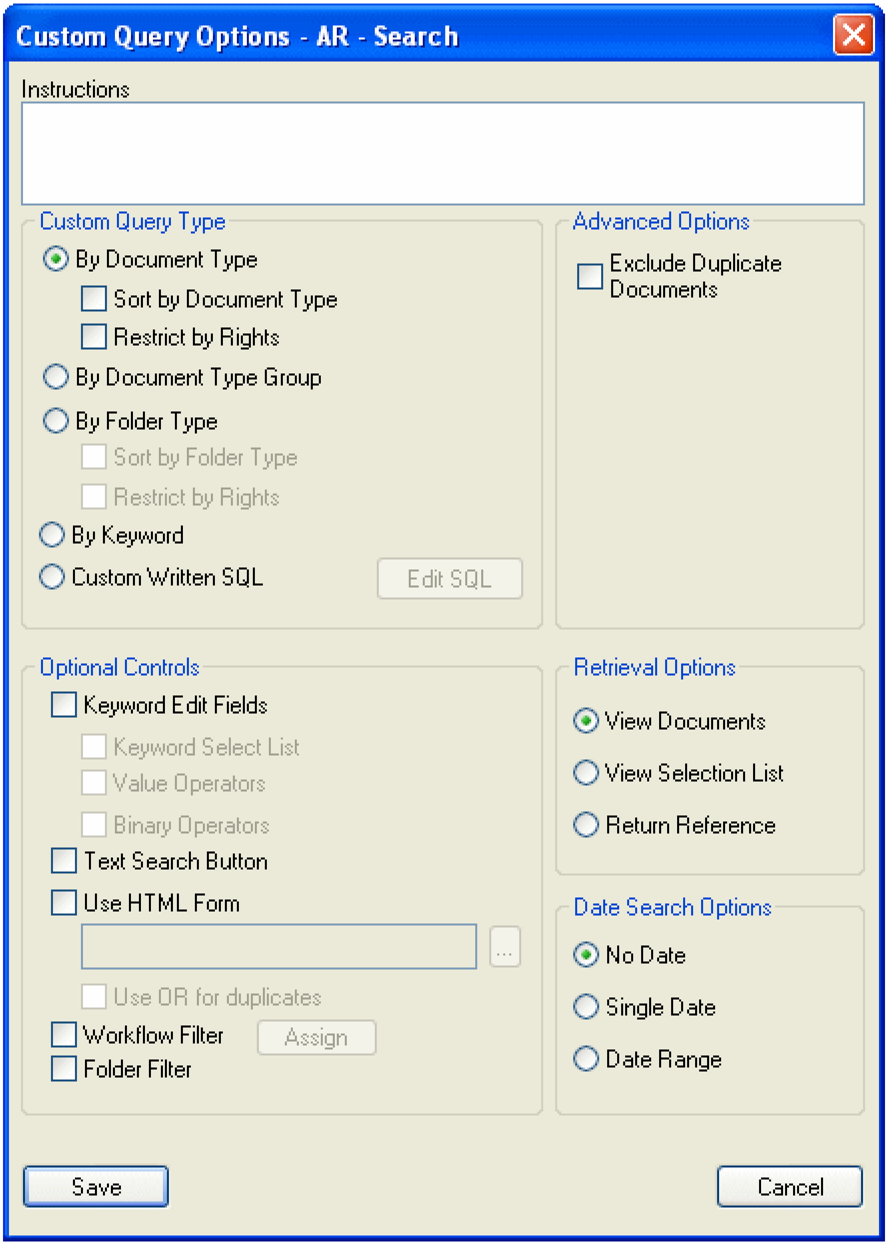
3. Type a name for the Custom Query in the **Custom Query** dialog box.



4. Click **Create**.

5. Assign **User Group(s)** to the Custom Query. See Add to Selected/Remove from Selected on page 722 for more information about adding and removing selected items from lists.

6. In the **Custom Query Options** dialog box, 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** and browse the form file by clicking [...].
8. 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.

9. 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.
10. Click **Save**.
11. 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 Keywords

Auto Number Keywords are used to show how many documents of a Document Type appear within the database system.

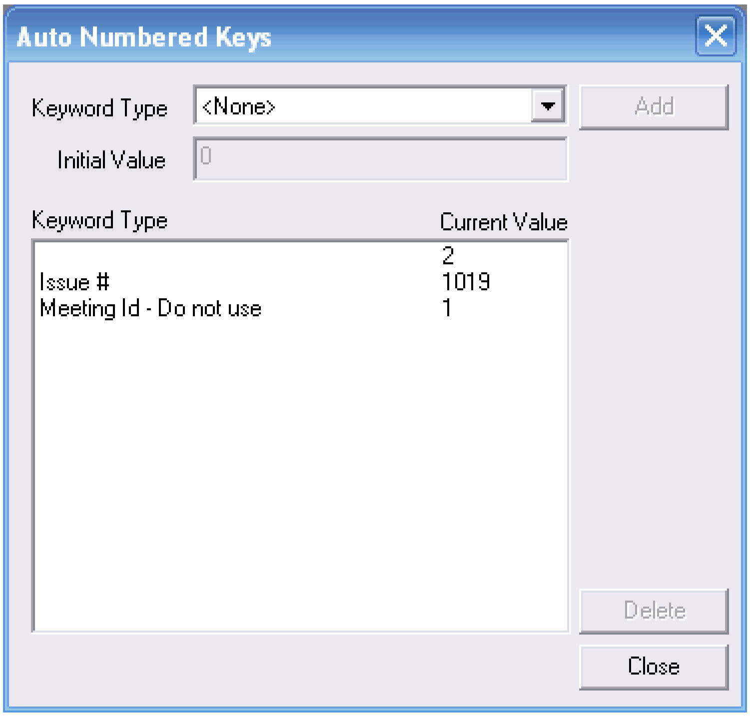
Auto Number Keywords are used with E-Forms. When a new E-Form is created in an OnBase client, this number is incremented by one. If during the process of E-Form creation, the user decides to cancel or close the E-form without submitting it, the Auto Number Keyword is still incremented. Even though the last E-Form was not submitted, the next E-Form created increments the Auto Number Keyword 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's value is incremented 2 from the first submitted E-Form.

Since user's may cancel documents during creation, the Auto Number should not be used to determine an accurate count of documents within a specific Document Type. For a more accurate count of submitted E-Form documents into the system, a Workflow solution could be created to capture E-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

1. In the Configuration module, select **Keyword | Auto Number Keywords Types**.
2. In the **Auto Numbered Keys** dialog box, select the counter **Keyword Type** from the **Keyword Type** drop-down list.



3. Type 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, 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 select list.

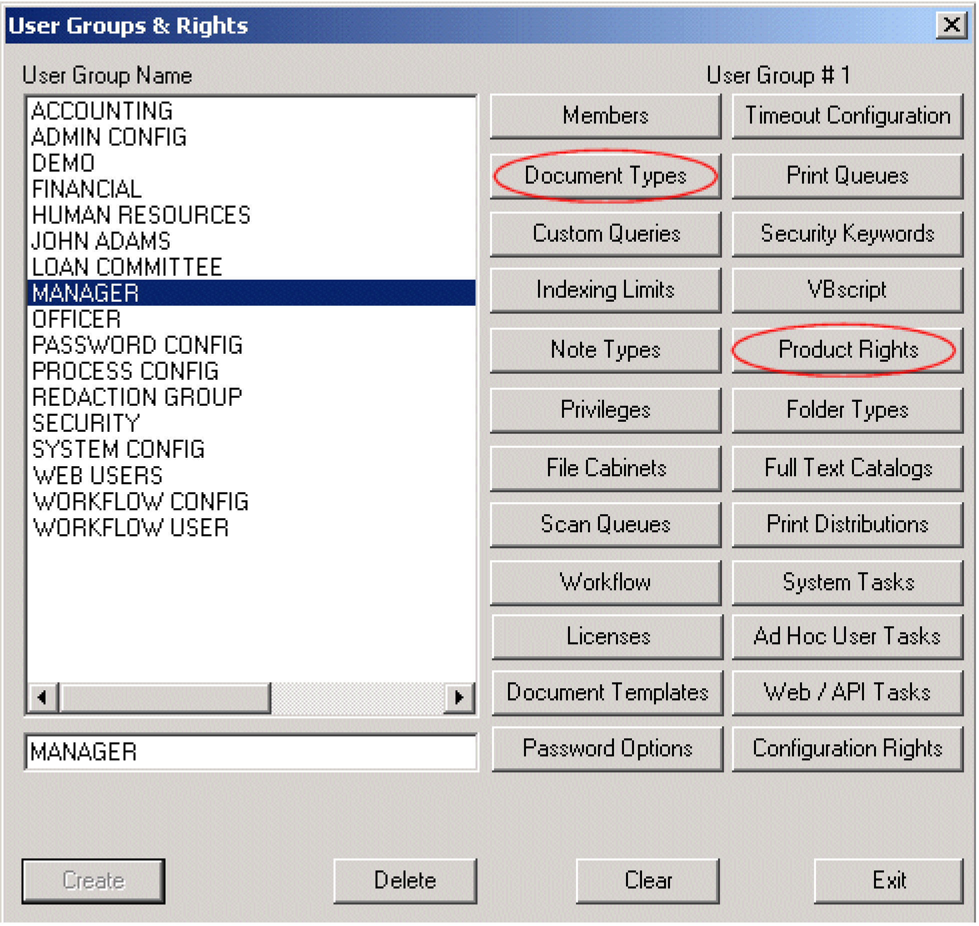
Every time this Keyword Value is brought into the system on a Document Type, its value increments to indicate the number of documents that are now in the Document Type. The current value will increase as the number of documents are increased in the system.

User Groups & Rights

In order to create and fill out forms, users must have product rights to **HTML Forms**. To assign product rights, in the Configuration module choose **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 the Configuration module, choose **Users**, **User Groups/Rights**. Select the appropriate **User Group** name, and click **Document Types**.

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

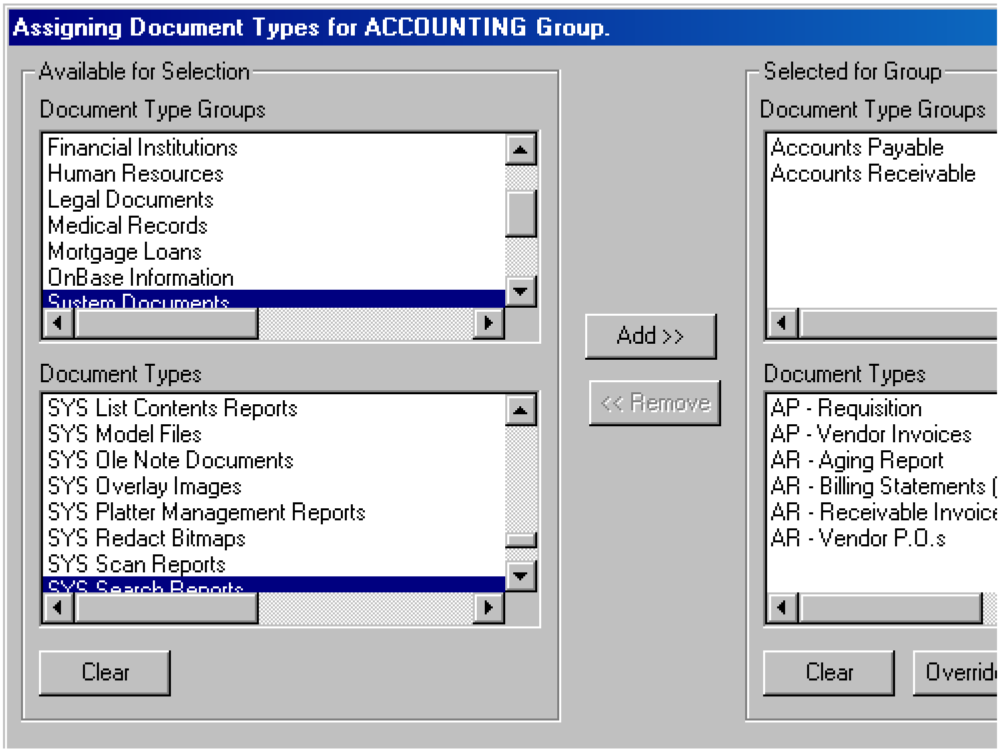


Privileges

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

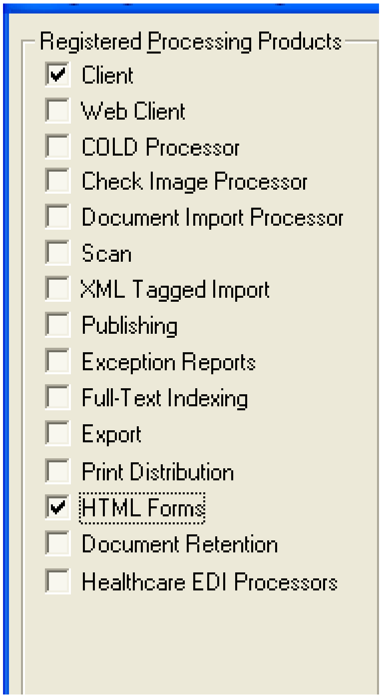
In order to process documents as specific Document Types, as well as view the documents that have been processed into the OnBase system, the user group must have rights to the Document Types. This is done under the **Document Types** button. Highlight the Document Types or groups on the left and click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



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 on the **Product Rights** button and select the **Client** and **HTML Forms** options under **Registered Processing Products**.



SYSTEM INTERACTION

Workflow

When changes are made to an E-Form in Workflow, the E-Form becomes a separate document.

Note: In order to have the newly created E-Form update the original document, a new action must be created for each keyword that the original E-Form and the new E-Form will share. The new E-Form must be associated with the original document. Select the action key **Set related documents keywords equal to this document**. Do this for each keyword to be updated on the original document.

Digital Signatures

Signing E-Forms

You can sign an E-Form the same way you would sign an image or text document in the OnBase Client.

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

Document Knowledge Transfer

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

Document Composition for Microsoft Word and Image Document Composition

When used in conjunction with Document Composition for Microsoft Word and 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 Document Composition for Microsoft Word or Image Document Composition to submit the request.

INSTALLATION

Requirements

If you will use E-Forms with the Web Client, refer to the Web Server Module Reference Guide for current Web Server requirements.

If you will use E-Forms with the Unity Client, refer to the Unity Client Module Reference Guide for current Unity Client requirement.

The following sections list the requirements for using E-Forms with the Client module and the Desktop.

Client Module Supported Operating Systems

- Windows XP SP3 or later service pack
- Windows Server 2003 SP2 or later service pack
- Windows Vista SP1 or later service pack
- Windows Server 2008
- Windows Server 2008 R2
- Windows 7

8

Note: As of release 7.2, OnBase no longer supports the use of the Windows® NT 4.0 and Windows® 98 operating systems. Microsoft retired its support of Windows NT 4.0 on January 1, 2005 and Windows® 98 on July 11, 2006. Microsoft no longer offers technical support or security updates for these operating systems.

Note: As of release 10.0.0, OnBase no longer supports the use of the Windows 2000 Professional Edition and Windows 2000 Server operating systems. Microsoft's extended support for these operating systems was retired on July 13, 2010. Microsoft will no longer offer technical support or security updates for these operating systems after that date. Additionally, these operating systems do not support Microsoft .NET Framework 4.0, which is an OnBase requirement as of version 10.0.0. If you are using either Windows 2000 operating system, you should not upgrade to OnBase 10.0.1 until you have upgraded to a Windows operating system supported by OnBase.

Operating System Requirements

The above Client module supported operating systems require both:

- Microsoft .NET Framework 4.0 (Client Profile)
- Microsoft Visual C++ 2010 Redistributable Package (x86)

Note: The Microsoft Visual C++ 2010 Redistributable Package (x86) is installed using the **vcredist_x86.exe** delivered with OnBase.

Desktop Supported Operating Systems

- Windows XP SP3 or later service pack
- Windows Server 2003 SP2 or later service pack
- Windows Vista SP1 or later service pack
- Windows Server 2008 or later service pack
- Windows Server 2008 R2
- Windows 7

Note: As of release 7.2, OnBase no longer supports the use of the Windows® NT 4.0 and Windows® 98 operating systems. Microsoft retired its support of Windows NT 4.0 on January 1, 2005 and Windows® 98 on July 11, 2006. Microsoft no longer offers technical support or security updates for these operating systems.

Note: As of release 10.0.0, OnBase no longer supports the use of the Windows 2000 Professional Edition and Windows 2000 Server operating systems. Microsoft's extended support for these operating systems was retired on July 13, 2010. Microsoft will no longer offer technical support or security updates for these operating systems after that date. Additionally, these operating systems do not support Microsoft .NET Framework 4.0, which is an OnBase requirement as of version 10.0.0. If you are using either Windows 2000 operating system, you should not upgrade to OnBase 10.0.1 until you have upgraded to a Windows operating system supported by OnBase.

Desktop Requirements

Desktop Component	Minimum	Recommended
CPU	1 GHz	
Memory (RAM)	512 MB	1 GB or greater (2 GB or greater for Windows Vista and later versions of Windows)
Free Hard Disk Space	200 MB	500 MB
Screen Resolution	1024 x 768 (1280 x 800) **Note:**Using a lower resolution may result in a loss of functionality.	1280 x 1024 (1440 x 900 widescreen)
Microsoft .NET Framework	Local mode - Microsoft .NET Framework 4.0 (Full Profile/ Extended) Remote mode - Microsoft .NET Framework 4.0 (Client Profile)	
Microsoft MSXML	MSXML 4.0 SP2 or later service pack	
E-mail Platform	MAPI 1.1 Compliant E-mail Client connection and supporting Active Messaging DLLs	

8

Databases Supported

The following tables list the databases supported in OnBase 10.0.1.

Microsoft SQL Server	Additional Information
Microsoft SQL Server™ v7.0 **Microsoft SQL Server 2000 (SP4 recommended)** **Microsoft SQL Server 2005 (SP2 or later recommended)** **Microsoft SQL Server 2008 (RTM and SP1; SP1 recommended)** **Microsoft SQL Server 2008 R2**	Microsoft SQL Server 7, 2000, and 2005 must be running in compatibility mode 7 or greater. Running in compatibility mode 6.5 or lower will result in errors during the upgrade process. SQL Server 2005 drivers must be upgraded to the Feature Pack for Microsoft SQL Server 2005 - December 2008 or a later feature pack. **Note:**You must ensure that your SQL Server database client software version matches or exceeds the database server version. For example, if your database server is SQL Server 2005, verify that the database client is SQL Server 2005 (or later). Running a previous client version, such as SQL Server 2000, will result in system instability and memory issues. For instructions on determining your server and client versions, see "Database Client / Server Version Compatibility" on page 95.

8

Oracle	Additional Information
Oracle® v 8.0.5.0 or later	Oracle version 8.0.5.0 can be used, but it is not recommended due to potential memory leaks. If Oracle 8.0.5.0 is used, a third-party ODBC driver is recommended. **Note:**It is strongly recommended that you have a certified Oracle Database Administrator on staff if you are using an Oracle Database.
Oracle 8i: 8.1.7.7 or later	ODBC drivers should be 8.1.7 or later. 8.1.6.x drivers have known issues and are not supported. **Note:**It is strongly recommended that you have a certified Oracle Database Administrator on staff if you are using an Oracle Database.

Oracle	Additional Information
Oracle 9i: Release 1 and Release 2 (9.2) **Oracle 10g: Release 1 and Release 2**	Oracle driver version 10.2.0.3 is recommended. An issue has been observed with the Oracle ODBC drivers where Unicode characters (e.g., Japanese characters) retrieved from a CLOB data type will be truncated, and could potentially cause errors in OnBase. The data remains intact in the database; however the results when retrieving the data will be truncated. This has specifically been observed in two areas of OnBase that use the CLOB data type to store large amounts of data: VBScripts and License Certificates. This behavior may apply to other areas of the software that use this data type as well. To ensure that Unicode characters retrieved from a CLOB data type are not truncated, the Oracle 10g R2 ODBC drivers (which are backward compatible) should be installed, as well as the latest patchset (version 10.2.0.3) for these drivers. **Note:**It is strongly recommended that you have a certified Oracle Database Administrator on staff if you are using an Oracle Database.
Oracle 11g: Release 1 and Release 2	All Oracle 11g drivers can be used. **Note:**It is strongly recommended that you have a certified Oracle Database Administrator on staff if you are using an Oracle Database.

Sybase SQL Anywhere	Additional Information
Sybase SQL Anywhere™ 5.5.04, 7.0.0-7.0.3, 8.0.1, 9.0, 10 and 11	No 6.0.X releases. **Note:**Sybase Adaptive Server Enterprise (ASE) is not supported.

Database Client / Server Version Compatibility

Due to critical issues that have been reported to Hyland Software, Hyland Software strongly recommends that:

- your database client software version matches or exceeds the database server version and

8

- you are running the most recent version of the database client.

This will help to reduce compatibility issues and minimize troubleshooting time when issues do occur.

Your database administrator can determine the database server version and identify the most-recent version of the database client software. The ODBC driver number indicates which version of the database client software you are using. For example, if your database server software is Oracle 10 Release 2, verify that the Oracle Client software is Oracle 10 Release 2 (or later). The same is true of SQL databases. For example, if your database server is SQL Server 2005, verify that the database client is SQL Server 2005 (or later).

To check your database client version, perform the following steps from the workstation or server where the ODBC connection is configured:

1. Open your ODBC Data Source Administrator, and click on the **Drivers** tab.
2. Select the driver you are using to connect to your OnBase database.
   - If your database server software is Oracle 10 Release 2, the version number should appear as **10.2.[#.#.#]** (or later), where **10.2** is the version number and **[#.#.#]** represents the service pack.
   - If your database server software is SQL Server 2005, the version number should appear as **2005.[##.####.##]** (or later), where **2005** is the version number and **[##.####.##]** represents the service pack.

The above descriptions are examples of two commonly used database version numbering schemes. Ensure that the supported database that you use adheres to the database client/server recommendation.

Database/File Servers

Server requirements are site-specific. Database/file servers should be dedicated purpose servers; i.e., not used as a domain controller, e-mail server, print server, proxy server, etc. Network and disk I/O hardware should be optimized for performance and redundancy. Multiple network interface cards on servers are often required to minimize network bottlenecks.

Hardware

Client Retrieval Workstation Hardware Requirements

Hardware	Minimum	Recommended
CPU	1 GHz	2 GHz or faster
Memory (RAM)	512 MB	1 GB or greater (2 GB or greater for Windows Vista and later versions of Windows)
Free Hard Disk Space	200 MB	500 MB
Screen Resolution	1024 x 768 (1280 x 800) **Note:** Using a lower resolution may result in a loss of functionality.	1280 x 1024 (1440 x 900 widescreen)

About Virtual Environments

Hyland Software develops, tests, and supports the OnBase suite of products on specific Operating Systems, not specific hardware configurations. When OnBase is operated in a virtual environment (such as Citrix, VMware, Hyper-V, or Windows Terminal Server) there may be limitations or subtle differences imposed by the environment. The customer and the virtual environment vendor are responsible for any interactions or issues that arise at the Hardware or Operating System layer as a result of their use of a virtual environment.

When it appears that an OnBase performance-related issue is either caused by (or is unique to) the virtual environment, organizations may be asked to validate that the issue occurs in a non-virtual environment. Hyland Software will make this request if there is reason to believe that the virtual environment is a contributing factor to the issue.

Each OnBase site is unique. Hyland Software depends on the customers who deploy OnBase in virtual environments to do so only after careful design and adequate planning (that takes into account the workloads of your organization), and in accordance with recommendations provided by the virtual environment's vendor. As with any implementation, Hyland Software strongly recommends that any customer deploying an OnBase solution in a virtual environment thoroughly test the solution before putting it into production.

For information about using OnBase in a Citrix and Terminal Server environment, please see the **OnBase in a Citrix and Terminal Server Environment** reference guide, available from your solution provider.

64-Bit Support Statement

The OnBase suite of products is tested on 64-bit systems and is capable of being deployed on 64-bit systems using the Windows 32-bit on Windows 64-bit Emulator (WOW64) layer.

Supported database versions that are deployed on a 64-bit database server are also supported. For more information, contact your solution provider.

Windows User Account Control Statement

Hyland Software is dedicated to ensuring that OnBase is compatible with Windows User Account Control (UAC). UAC is a feature of Windows operating systems that was introduced with Windows Vista. It limits the ability of standard users to make global system changes to a workstation and prevents malicious software from making unauthorized changes to protected areas.

For details on UAC, refer to your Microsoft support information or see http://technet.microsoft.com/en-us/library/cc709691(WS.10).aspx.

You may encounter UAC in OnBase when:

- Installing or uninstalling OnBase, an OnBase module, or OnBase ActiveX controls.
- Copying, moving, or saving files to the Program Files directory, Windows directory, or another protected location.
- Modifying system-wide settings, such as the registry.

If Windows UAC is enabled, the above operations will prompt for administrator privileges, even if an administrator is currently logged on.

Third-Party Software

The ability to create and edit HTML documents is required for the E-Forms module. In many cases an HTML editor, such as Microsoft FrontPage™ 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.

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 FrontPage).

Registration

Workstation registration is not required for E-Forms.

Installation

There is no installation process for E-Forms.

To install EForm Validator the EFValid.exe file should be copied to the computer. Also, the OnBase API for OnBase Client should be installed. Before running the E-Form Validator, the mzapi.dll and mzasp.dll must be registered using the following commands:

- regsvr32 mzapi.dll
- regsvr32 mzasp.dll

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.

To backup EForm Validator, the EFValid.exe file should be backed up.

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.

To restore EForm Validator the EFValid.exe file should be restored from its backup location. Also, the OnBase API for OnBase Client should be present on the computer.

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.

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?

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

This causes Internet Explorer documents and OnBase E-Forms to print without headers or footers.

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 (Example: 10.0.1.571) and/or the Core Services version and build (Example: 10.0.1.6).
- The type and version of the connected database, such as Microsoft SQL Server 2008 or Oracle 11g, and any Service Packs that have been installed.
- The operating system that the workstation is running on, such as Windows XP or Windows Server 2008, and any Service Packs that have 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 Packs that have 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.

E-FORMS BEST PRACTICES

A

The following best practice recommendations have been 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

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

Use a text editor, like Notepad®, to edit and save the revised file as plain text. HTML editors, like FrontPage®, are not recommended because they can add extraneous information.

Some HTML editors store extra spaces as ** ** 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 field still takes 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 the E-Form Validator to verify the configuration of OnBase items on a form.

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 select 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-Shft-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. 'form.input1.attachEvent("onclick", fnEventHandler)'. However, If you are developing for a non-Internet Explorer browser, this approach will not work.

Store stylesheets 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.

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 "OBBtn_Yes" when an E-Form is used in Workflow and a button function is not specified otherwise. Previous to 9.2 of OnBase, buttons that were not configured specifically as submission buttons would function as cancel buttons. In 9.2 and beyond, you must configure appropriate cancel buttons, as button not specifically configured will be processed as a "yes" submit button.

OFFLINE E-FORM DESIGN

Offline E-Forms are used in conjunction with the Unity Briefcase and Integration for Allscripts 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.
- Electronic and Digital Signatures 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.

Index