Infor Mongoose
About Form Synchronization

Publication Information

Release: Infor Mongoose 9.00

Publication date: November 5, 2013

Contents

Chapter 1: About FormSync

1

Overview

In Design Mode, WinStudio serves as a development tool for customizing forms in Mongoose-based applications. FormSync is an aid to preserving your customizations when you upgrade to a newer version of Mongoose or your Mongoose-based application; when you apply a service pack or a single issue fix; or when you distribute and apply your own forms, form modifications and other global objects in Mongoose. The process of handling these changes during an update or upgrade is referred to as *synchronization*.

Note: Mongoose also incorporates much of the functionality described in this guide within the **Form Sync** form. But the reader should be aware that not everything described in this guide applies equally to that form. For more information about the differences, see the online help for that form.

FormSync also provides a set of development and administrative utilities with which you can:

- Gather information about customizations.
- Copy customizations from one database to another.
- Delete objects.
- Compare objects.
- Generate SQL scripts to apply customizations to a remote database.
- Generate "pseudo code" for forms and objects to use later in evaluating and updating customizations.

Although we assume in this document that you are familiar with WinStudio form development, we emphasize here some basic WinStudio concepts to show the role of FormSync in the development process.

Some Terminology

The following terms are used throughout this document:

- **Forms database** – A SQL Server database in which specifications and metadata that define forms and form-related objects are stored. When a user opens a form, WinStudio retrieves these specifications and metadata, and creates the functional form.

- **Form object** – An item such as a form, form component, or variable defined in WinStudio Design Mode. Specifications that define objects are stored in the forms database. They are the basic data that FormSync processes. (Object types and attributes are listed in Chapter 3, “Default Synchronization.")
- **Global object** – An element, such as a script or variable, that is not specific to a single form but which can be used by more than one form.
- **Vendor version** – The default version of an object supplied by Infor or one of Infor’s business partners. Vendor versions provide the delivered functionality of Mongoose forms. Those users who have normal editing permissions (Basic, Full User, or Site Developer) cannot change or delete the vendor version of an object in WinStudio Design Mode.
- **Update** – A version upgrade, service pack, or single fix to be applied to the Mongoose-based application. The objects in an update are typically vendor versions, but updates can also be provided by business partners and even in-house customizations and modifications.
- **Customized version (or customizations)** – An object changed in an application installation by a user with normal editing permissions (Basic, Full User, or Site Developer). WinStudio saves a changed object in the forms database as an object based on the vendor version (or on another customized version). In effect, an edited copy of a vendor version is inserted in the forms database to create the customized version.

  New objects created by a customer developer are also classified as customizations, as are copies of vendor versions saved under a new name.
- **Source database or configuration** – The database or configuration that contains new data to be merged with or to replace the data in an existing database or configuration.
- **Target database or configuration** – The database or configuration that contains existing data that can potentially be merged with or replaced by new data.

Remarks on Customizations and Synchronization

This section presents some basic concepts associated with customizations and synchronization.

The Nature of Customizations

FormSync looks for differences between a source version and a target version. Any type of change that has been made to an attribute value registers as a customization in synchronization, whether or not the change affects functionality.

For example, if a customer developer adds a comment to a script, FormSync considers the script to be customized, even though the functionality of the script has not changed.

A set of specifications for a complex object, such as a component class or a script, is treated as a unit in synchronization. Mongoose does not attempt to selectively synchronize individual specifications for the object.

The Relationship Between Global Objects and Forms During Customization

Global objects are listed as attributes of forms and components in the **Merge Options** dialog box. The values of these attributes are named references to global objects. A customization of a form component with regard to global objects consists only of selection of a different named object in the component specifications.

For example, if a component uses the right-click menu named **StdDefault** and a customer developer changes the value for a component to **StdDetailsAddFind**, FormSync processes the change as a customization of the component, not of the form that uses it. Customizations of global objects themselves are processed separately, as specified in the **Synchronization Options** dialog box. If you change the attributes of the menu StdDefault, the menu is processed as a customized version of the global object, but form components that *use* the customized menu are not processed as customized.

The Basic Synchronization Process

During synchronization, FormSync compares new versions of forms and objects (in the Source database) with existing versions of those forms and objects (in the Target database). FormSync also compares the Source versions with any existing customizations in the Target database. Depending on your synchronization settings, FormSync then either keeps the customizations, merging them with new data from the Source database, or removes and replaces customizations with the new data. Finally, FormSync replaces the existing vendor versions in the Target database with the new vendor versions from the Source database. For more information, see Chapter 2, “Schematic Overview of Synchronization."

Conflicts Between Customizations and New Versions

FormSync does not attempt to resolve conflicts between the size and position of customized components and revised settings for size and position in a new base version. Where you might have moved, resized, or added components on a form, those components can display overlapped with a base version of a component in a merged form.

Automatic Conversions of Scripts

One of the major changes for FormSync Version 8.0 (which has been carried forward to subsequent verions) was the conversion of VBA code to .NET code. During synchronization, FormSync evaluates both form and global scripts to determine whether they are VBA code, which was used in earlier versions of applications (such as SyteLine) built on Mongoose.

If the code is determined to be VBA, then FormSync attempts to convert the code automatically to VB.NET. After converting the code, FormSync prompts you to keep, remove, or edit the new version of the script.

If you choose to:

- **Keep** the converted code, FormSync commits the automatically converted code to the Target database.
- **Remove** the converted code, FormSync removes the customized script and replaces it with the script from the new (Source) version.
- **Edit** the converted code, FormSync opens the file comparison utility with the new Source version of the script in one pane and the customized code that FormSync converted to VB.NET in another pane. This allows you to compare versions, evaluate the new converted version, and make changes in the customized version if necessary, before FormSync commits the changes to the Target version.

  Note: Whether you keep the converted code as is or edit and then save it, there are no guarantees that the resulting code is fully functional. Plan to test the script before placing it into a production environment.

  FormSync does not automatically convert inline scripts. If you have inline scripts, FormSync simply removes them, replacing them with the new Source version. If you want to continue using the customized version, you must manually convert it after synchronization.

FormSync Users

The degree of development experience required to use FormSync depends on the level of customization in a forms database.

- If customized versions contain only those changes allowed to users with *Basic* editing permission, FormSync automatically handles all customizations. In this case, experience with form development in WinStudio is not required. (See Chapter 3, “Default Synchronization" for Basic customizations.)
- If customized versions contain changes other than Basic customizations, you should be prepared to reply to prompt messages pertaining to specific attributes of forms. Interpreting prompt messages, testing results, and manually changing merged forms all require an understanding of form development in WinStudio. You should also be familiar with your organization’s development plan and policies.

Note: If an installation contains no customizations, you should not run FormSync in connection with updates. But be aware that even the simplest user change to a form constitutes a customization, so if you have any customizations at all on your system, you should consider running FormSync.

Scope of Synchronization

FormSync processes customizations made to the client tier only, that is, forms and global objects. Such customizations are made in WinStudio Design Mode. Common customizations include actions like:

- Resizing a form
- Adding a text-box to a form
- Changing the source for a drop-down list

FormSync does not process or examine changes made to the middle (IDO) tier or the application-database tier. Synchronization and the operations performed by FormSync utilities do not apply to objects listed in this table.

Objects Not Processed by FormSync

Tier	Object
Middle	IDO metadata Report definition files
Application database	Table columns Stored procedures Application Event System metadata Triggers Functions Report background-task definitions

Planning Customization and Synchronization

As a development project, customizing WinStudio forms requires planning, documentation, and application of development policies. FormSync cannot anticipate the consequences of all customizations or guarantee the functionality of both customizations and base-level functionality after an update is applied. FormSync is most effective in maintaining functionality when it is used in conjunction with a development plan, documentation supporting the plan, and testing.

FormSync Administration Utilities

The functionality of FormSync in the update process is different from its functionality as a development tool for administering forms databases.

In an update, synchronization handles vendor versions of forms in both the revised and the unrevised versions of a Mongoose-based application. FormSync compares any customized version of a form

with the two vendor versions of it. FormSync then allows you to update the customized version with specifications from the new vendor version, while selectively keeping customizations. Alternatively, synchronization can remove customized versions entirely or keep them intact. In all cases, FormSync replaces old vendor versions with new vendor versions.

FormSync's utilities for copying, deleting, and scripting objects do not affect vendor versions of forms or global objects. These utilities access only customized versions. These utilities are useful primarily to those who customize forms using a development forms database and subsequently copy the customizations to a production forms database. Utility operations do not alter the application version number or service-pack number in a forms database.

The utilities can also serve an informational role in updates. You can:

- Generate lists of customizations currently in a forms database.
- Compare customizations with vendor versions.
- Store customizations in script files before you synchronize databases.
- Generate "pseudo code" to use in evaluating and comparing objects.

For more information, see Chapter 6, “FormSync Utilities" and the online help for "Utilities."

Note: These utilities are available only with the standalone version of FormSync, and not with the **Form Sync** form.

Synchronizing Site Default or Group Versions

In addition to synchronizing customized versions with new vendor versions in the update process, you can synchronize customized versions in a production forms database with customized versions in a development forms database. This feature of FormSync allows customer developers to conveniently deploy customizations made at Site Default scope and Group scope. See Chapter 5, “Synchronizing Site Default or Group Versions."

For More Information

For more information about how FormSync works, including information about how to use FormSync effectively, see the online help for FormSync. We especially recommend that you read the following topics before using FormSync:

- "Understanding WinStudio Customizations"
- "About Synchronization" (and subtopics)
- "Before You Use FormSync" (and all subtopics)
- "Using FormSync" (and subtopics)

Chapter 2: Schematic Overview of Synchronization

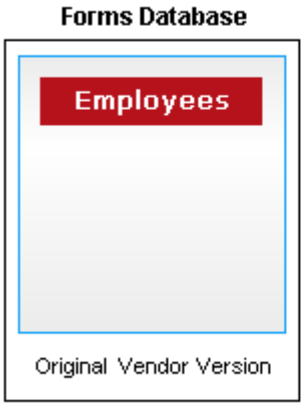
2

This chapter shows schematically the results of running FormSync with various synchronization options. Because it is an overview of functionality, this chapter provides a minimum of technical or procedural detail.

Default synchronization is covered in detail in Chapter 3, “Default Synchronization."

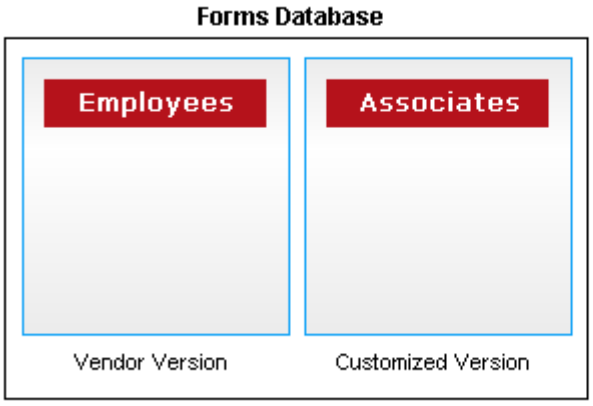
Synchronizing Forms

The following illustration represents a vendor version of a form that has a button labeled **Employees**.



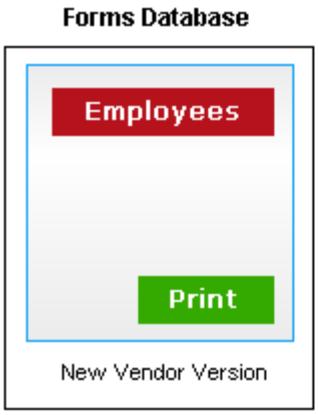
A customer developer changes the button label by specifying a different string in the **Caption** field of the **Component** properties sheet. The old caption was **Employees**; the new caption is **Associates**. The developer assigns **User** scope to the customization and assigns it to the user name **B_Smith**.

The customization results in the addition of a customized version of the form to the forms database, as in this illustration:



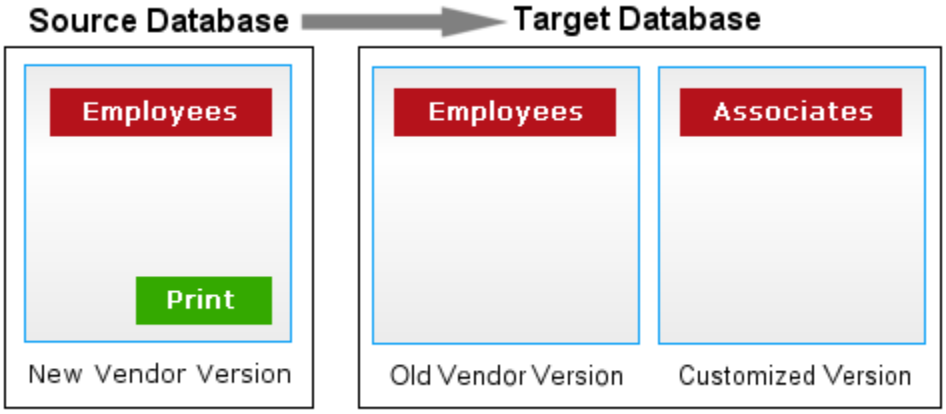
Only the user B_Smith has access to this customized version; other users can only access the vendor version. The vendor version remains unchanged and is not accessible to B_Smith.

A new forms database from a service pack or version upgrade contains a revised vendor version of the form. The revision includes a new command button labeled **Print** as in this illustration:



To synchronize the customized version of the form with the new vendor version, the first step is to select the databases in the **Configurations** dialog box. The forms database containing the customized form must be in the *Target* configuration, and the database containing the new vendor

version must be in the *Source* configuration. In synchronization, data from the Source database updates information in the Target database.



During synchronization of forms, FormSync compares all three of these versions and uses the selected customization option when determining what actions to take.

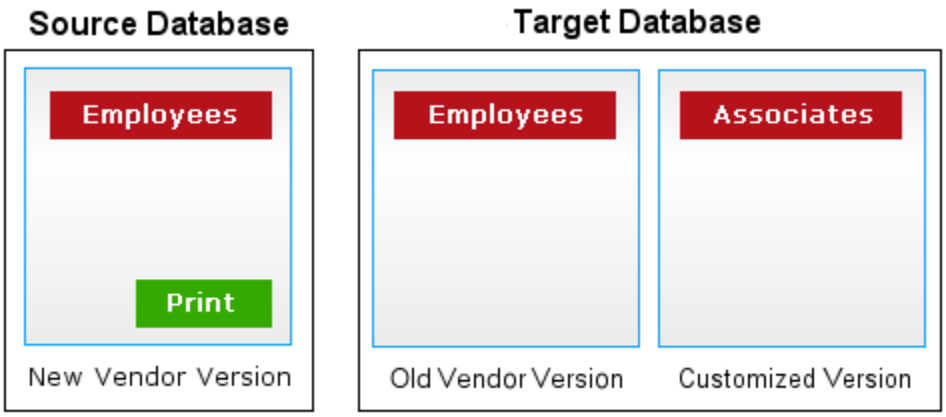
FormSync provides four synchronization options for customized forms. These options are illustrated in the following sections.

Option 1: Merge Customized Forms with Base-Level Forms

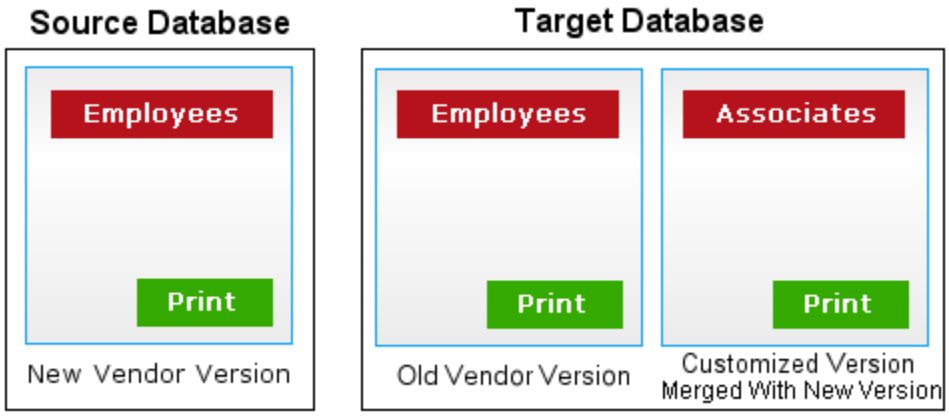
For each customized version of a form in the Target database, FormSync integrates (or prompts you to integrate) specific revisions contained in the new vendor version with the customized version of the form. FormSync then replaces old vendor versions with new vendor versions.

This is the default option.

Before Synchronization



After Synchronization



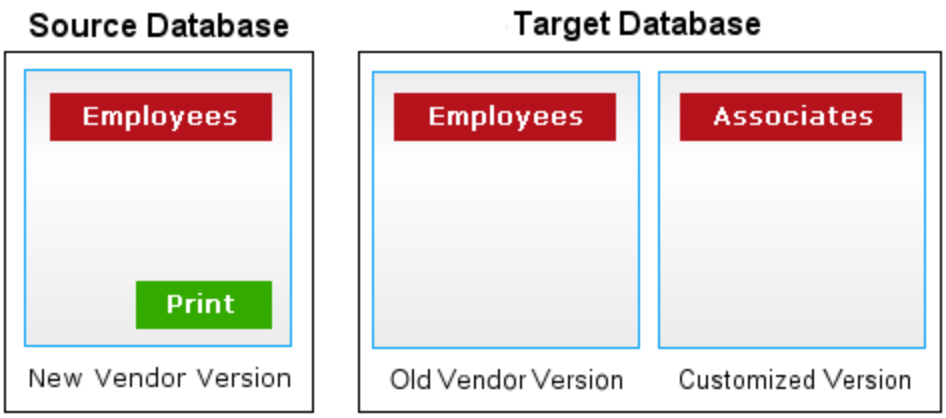
Results

- The old vendor version in the Target database is removed.
- The new vendor version is copied into the Target database.
- The customized version is kept in the Target database and incorporates both the pre-synchronization customization and the revised specifications from the new vendor version.
- The new vendor version has the **Print** button and the original caption **Employees**. The customized version has the new **Print** button and keeps the custom caption **Associates**.

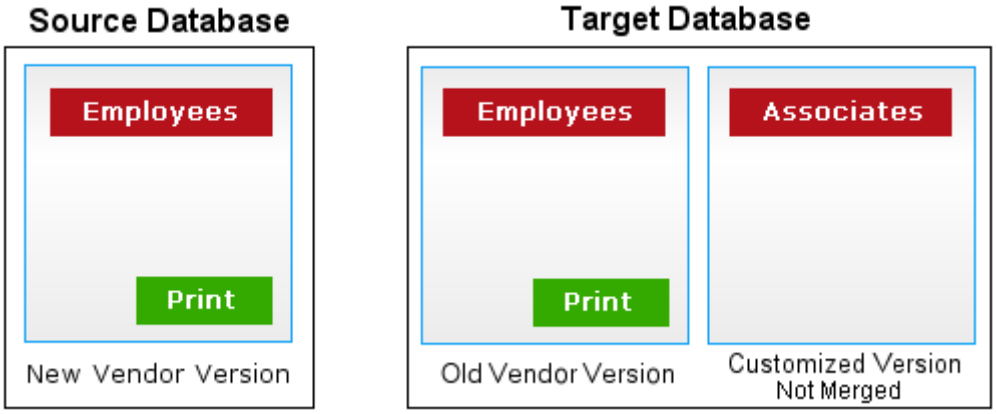
Option 2: Leave Customized Forms Untouched

FormSync keeps any customized versions in the Target database, independent of new vendor versions. FormSync replaces old vendor versions with new vendor versions.

Before Synchronization



After Synchronization



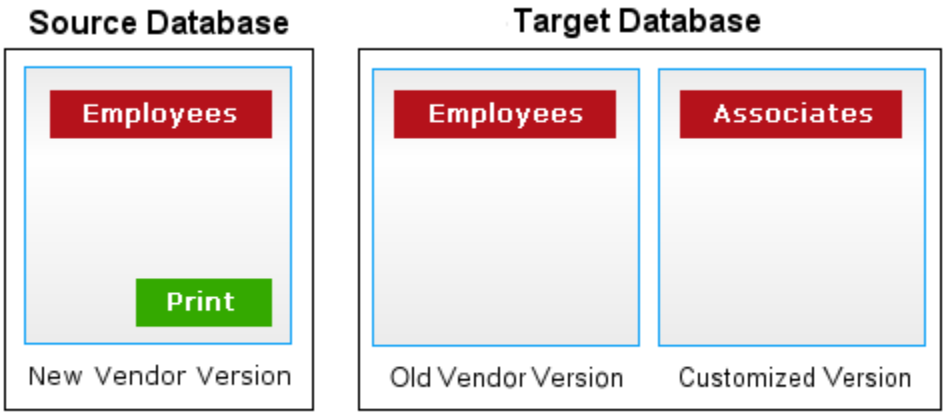
Results

- The old vendor version in the Target database is removed.
- The new vendor version is copied into the Target database.
- The customized version, which is based on the old vendor version, is kept in the Target database and is unchanged. The customized version does not incorporate revisions contained in the new vendor version.
- The customized version keeps the custom caption **Associates** but does not have the new **Print** button. The vendor version has the new **Print** button and the original caption **Employees**.

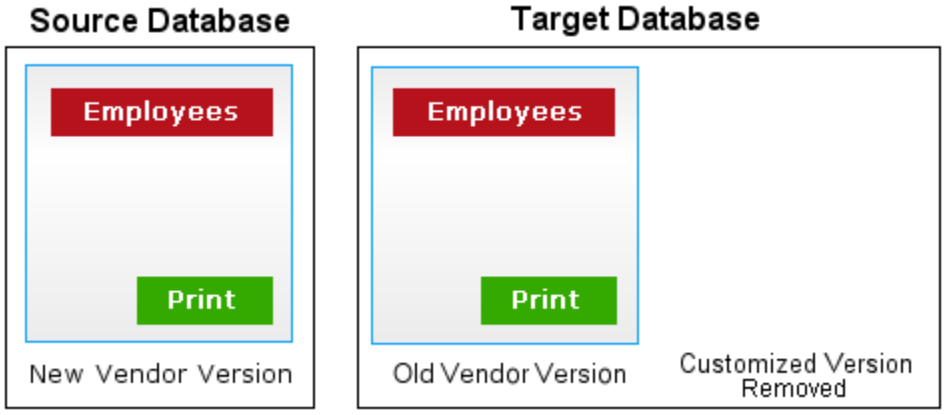
Option 3: Remove Customized Forms

FormSync removes customized versions entirely from the Target database. FormSync replaces old vendor versions with new vendor versions.

Before Synchronization



After Synchronization



Results

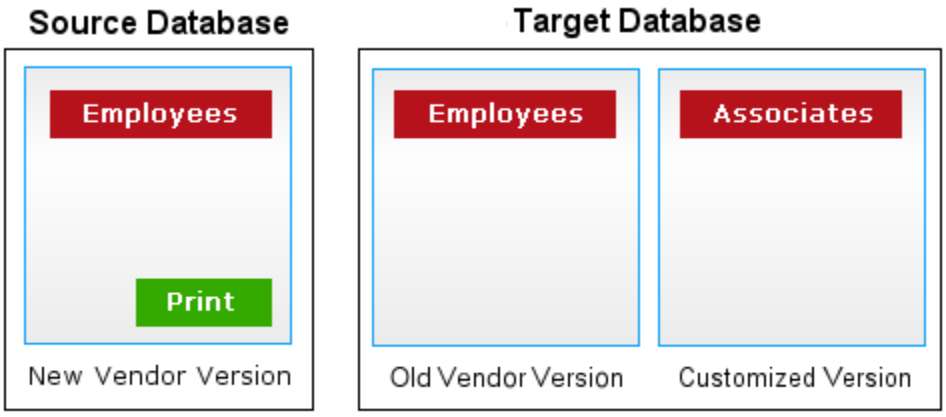
- The old vendor version in the Target database is removed.
- The new vendor version is copied into the Target database.
- The customized version is removed from the Target database.
- The new vendor version has the **Print** button. The custom caption **Associates** is not in use.
- The user **B_Smith** now accesses only the new vendor version, like everyone else.

Option 4: Do Not Process Forms

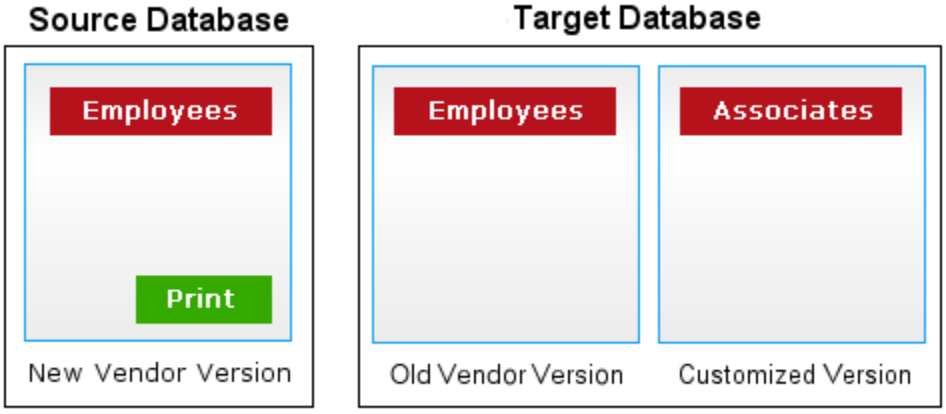
FormSync disregards form specifications in both databases.

This option is normally used when you want to synchronize only global objects or the Explorer.

Before Synchronization



After Synchronization



Results

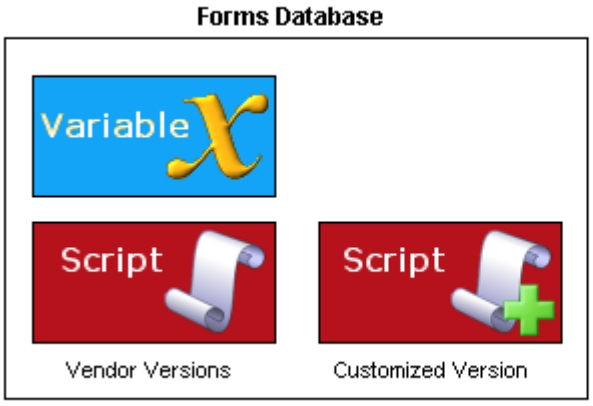
- The old vendor version and the customized version are unchanged.

Synchronizing Global Objects

A global object, such as a variable or a script, is not specific to a form but can be referenced in more than one form. FormSync handles customized global objects separately from customized forms.

In the following illustration, a forms database contains the vendor versions of a global variable and a global Visual Basic .NET script.

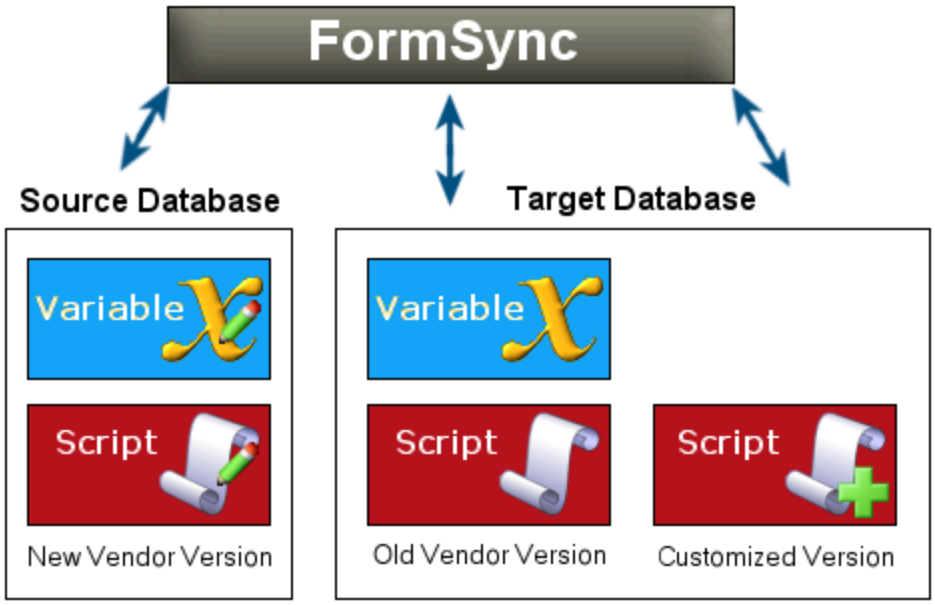
A customer developer changes a line of code in the script and assigns **User** scope to the customization, with user name **B_Smith**. The change results in the addition of a customized object to the forms database.



When B_Smith uses a form that references the script, the customized version is called, rather than the vendor version. The vendor version remains unchanged and is not accessible to B_Smith. Other users do not have access to the customized script.

Some time later, a new forms database from a service pack or version upgrade is issued containing revised vendor versions of the variable and the script.

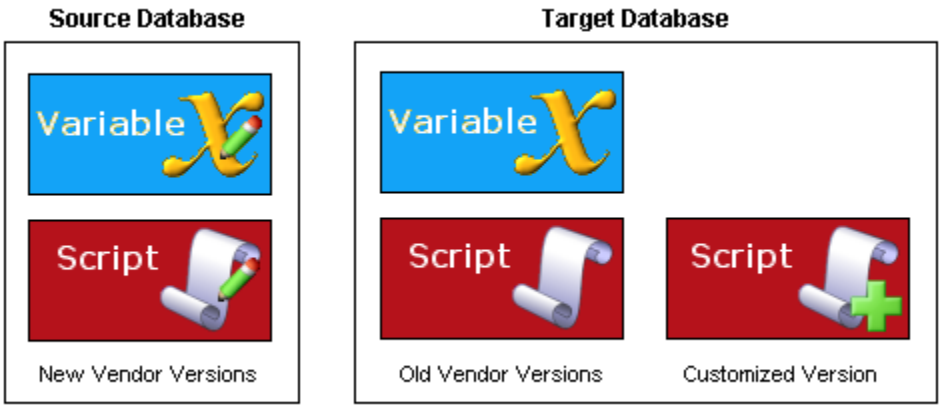
The selection of configurations for synchronizing global objects is similar to the selection for synchronizing forms, with the forms database containing the customized objects in the Target configuration and the new forms database in the Source configuration:



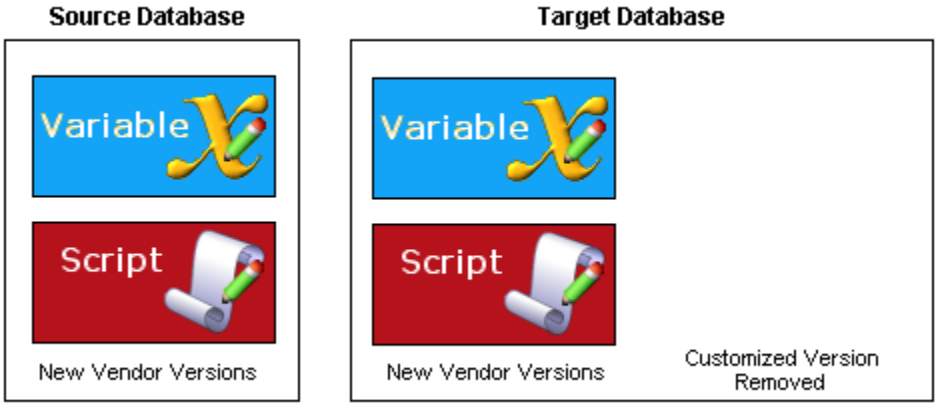
In general, a global object is treated as a unit in synchronization, unlike a form, whose multiple specifications for components and form properties can be handled individually in the merge process. A global script is an exception to this rule, and FormSync provides the user with an option to edit the script as well as to remove it or replace it. Synchronization options for global objects are **Remove customizations** (default) and **Keep customizations**. The two options are illustrated below.

Option 1: Remove customizations

Before Synchronization



After Synchronization



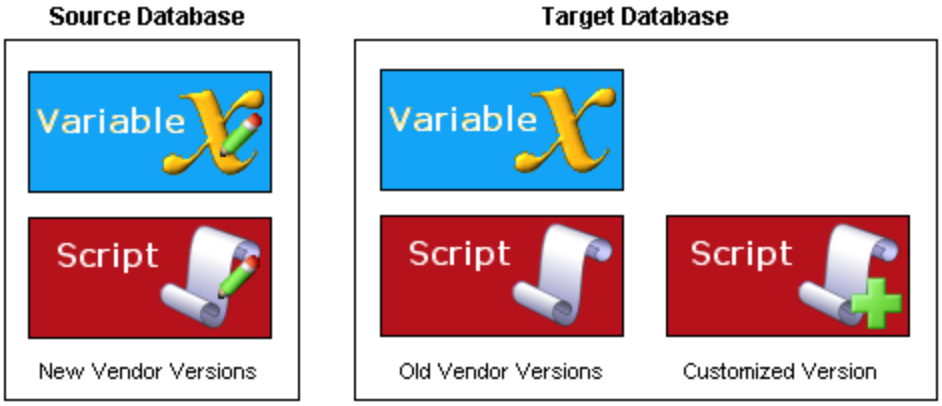
Results

- The old vendor versions of the variable and the script are removed from the Target database.
- The new vendor versions of the variable and the script are copied into the Target database.
- By default, FormSync prompts the user to remove a customized version of an object or to keep it. Here the user removed the object. If the user had kept the customized object, the result would have been the same as the result of **Keep Customizations**.
- When user B_Smith accesses a form that references the script now, the form calls the new vendor version of the script.

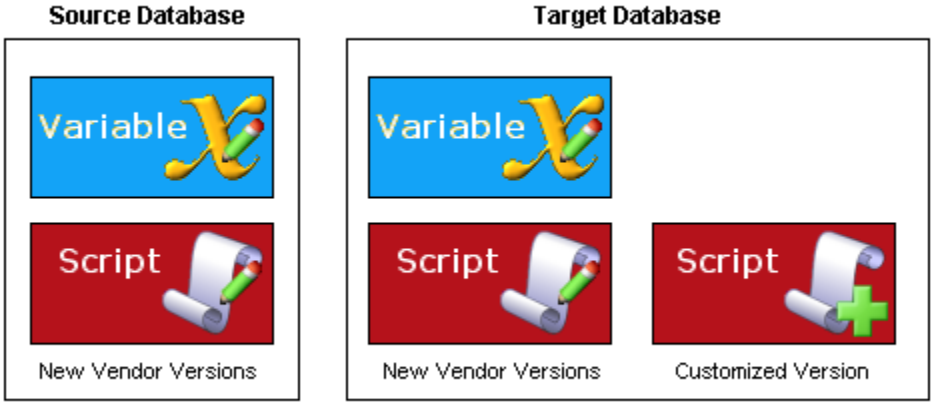
TIP: If you have multiple customizations to different global objects in your system, and you are not totally sure you want to keep them all during synchronization, select the **Remove Customizations** option and make sure you have **Prompt before removal** selected. This gives you the option to decide on a case-by-case basis.

Option 2: Keep customizations

Before Synchronization



After Synchronization



Results

- The old vendor versions of the variable and the script are removed from the Target database.
- The new vendor versions of the variable and the script are copied into the Target database.
- The customized version of the script, which is based on the old vendor version, is kept in the Target database and is unchanged. The customized version does not incorporate revisions contained in the new vendor version, unless you choose, in response to a prompt message, to edit the customized version and then incorporate the changes.
- When user B_Smith accesses a form that references the script, the customized version is still called, rather than the new vendor version.

Chapter 3: Default Synchronization

3

This chapter describes how FormSync handles customizations to forms, global objects, and the Explorer using default synchronization with a vendor-supplied update.

Form Object Specifications

Form object specifications are attribute values that define a form and components on the form. Values consist of WinStudio keywords, parameters, code for scripts, names of referenced global objects, and other specifications that are typically set in the WinStudio Design Mode, using the **Form**, **Component**, and **Object** property sheets.

A *form* in the synchronization process is the entire set of attributes of a form and components on the form. Sample attributes are listed in this table:

Form Attributes

Attribute Type	Attribute	
Form	Caption Data Source Event handler Filter form Form script (distinct from a global script)	Form variable (distinct from a global variable) Size Splitter settings Standard operations

Attribute Type	Attribute	
Component*	ActiveX control name	Hidden
	Autocomplete	List height
	Caption	List Source
	Center justification	Menu (short cut)
	Check box	Message on status line
	Component class	Multiple selection
	Containment	Need right click
	Data Source	No clear on new
	Data type/format	No colon
	Default	No tab stop
	Default on copy	Password
	Disable for existing Object	Radio button selected Value
	Disable for new object	Read only
	Disable in filter-in-place	Required
	Enable on nonmodified objects	Right justification
	Enable on required Data Supplied	Second splitter pane
	Events to generate	Size and position
	Find/maintain form	Solid
	Frame	Sort
	Glue Top	TBD for new objects
	Glue Bottom	Type
	Glue Left	Uppercase
	Glue Right	Validate Immediately
	Grid column frozen	Validators
	Group	Value is list index

* This is not an exhaustive list. To view all the component attributes for a given component, see the **Component** property sheet for that component.

Version Identification

A vendor version can serve as the basis for multiple customized versions created by, or for, individual users and groups. FormSync identifies versions of forms and global objects by a combination of:

- Form or global object name
- Scope type
- Scope name

These identifiers, in turn, appear throughout FormSync prompts and messages to identify customized versions to the user.

Scope type and *scope name* specify who can use a given version of a customized form or object:

Scope type	Scope name
User	A user name (a WinStudio user ID)
Group	The name of an administrative group defined on the **Groups** form
Site Default	Empty when **Site Default** is selected. Internally, in the database, ScopeType = **1** and ScopeName = **"[NULL]"** Note that **"[NULL]"** is a literal value.

For more information on scope, see the WinStudio help.

Example: The following is a logged message: "Processing property class extension VendNum User version for user B_Smith."

- The object name is **VendNum**.
- The scope type is **User**.
- The scope name is **B_Smith**.

In FormSync, a *customized version* is any form or global object with a scope type of **User**, **Group**, or **Site Default**. A *vendor version* is any form or global object that has not been customized and so has a scope type of Vendor (ScopeType = **0** and ScopeName = **"[NULL]"**). In the synchronization process, FormSync compares any customized version of a form with vendor versions of the same name in both the Source and Target databases.

All forms have scope attributes. Customer developers with Site Developer editing permissions specify scope attributes when they customize forms.

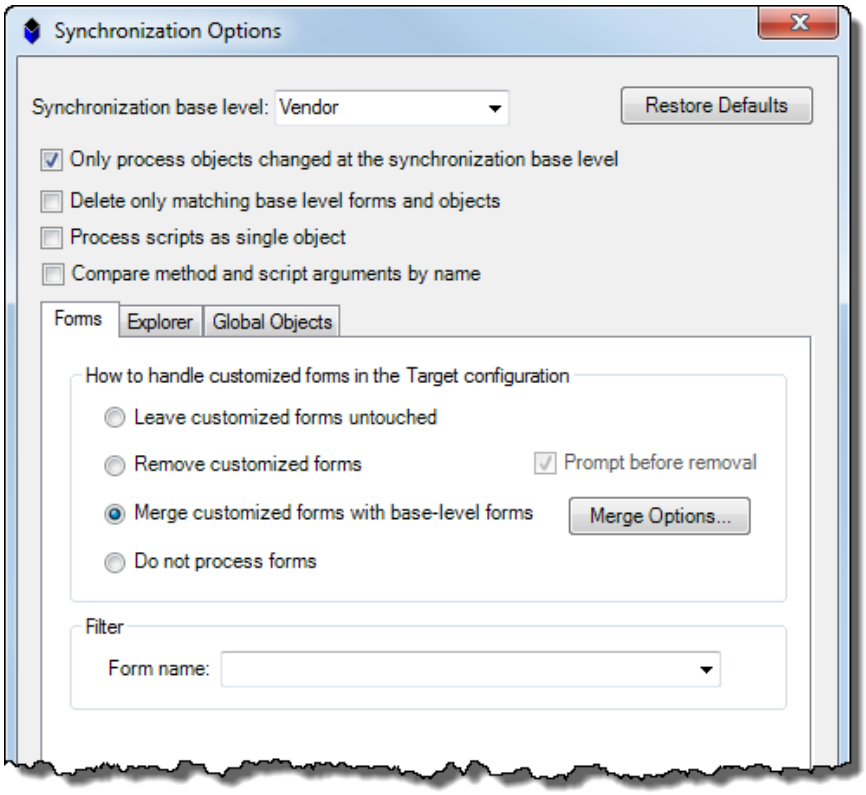
TIP: You can set the scope attributes either when you:

- First enter Design Mode.
- In Design Mode, select **Form > Definition > Site Developer Impersonation**.

When developers with lower-level editing permission customize forms, WinStudio sets scope attributes automatically.

Default Settings

These are the default settings for forms in the **Synchronization Options** dialog box:



Basic Settings

These settings affect how FormSync processes requests for all objects, whether forms, the Explorer, or global objects.

- **Synchronization base level** – Specifies the level of synchronization that FormSync is to use as the base level.

  Options include:

  - **Vendor**
  - **Site**
  - **Group**

  FormSync processes only those forms and objects that have been created or customized at the specified level and below.

For example, if you specify **Site** as the synchronization base level, FormSync first processes forms and objects created or customized at the User or Group level. FormSync then processes updates and changes made at the Site level. FormSync makes no attempt, in this case, to update any forms or objects created or customized at the Vendor level.

The value **Vendor** specifies that synchronization applies to vendor versions of forms in an update (version upgrade, service pack, or single fix).

- **Only process objects changed at the synchronization base level** – When selected, specifies that FormSync should only process those forms and objects that have actually been changed at the base level specified in the **Synchronization base level** field.
- **Delete only matching base level forms and objects** – When selected, specifies that FormSync is only to delete forms or other objects at the base level that have matching names. Forms and objects in the Target configuration that do not have corresponding items (that is, forms or objects with matching names) in the Source configuration are to be left alone.

  This allows for forms and objects created by business partners or other developers at the base level (as specified in the **Synchronization base level** field) to be preserved during synchronization.
- **Process scripts as single objects** – When selected, specifies that FormSync is to treat all scripts as single objects during synchronization.

  This is required for Visual Basic scripts that include multiple namespace directives and all scripts created in C#, because FormSync is unable to correctly parse these types of scripts.
- **Compare method and script arguments by name** – When selected, specifies that FormSync is to compare all calls to IDO methods and global scripts by name, as opposed to comparing them only by signature (which is the default).
- **Restore defaults** – When clicked, restores all values in this dialog box to their system default values.

Forms Tab

These are the default settings for the **Forms** tab:

- **Leave customized forms untouched** – When selected, specifies that FormSync is not to process any customized forms during synchronization. Only forms that have not been customized are processed.

  This leaves all customized forms just as they were before synchronization, which means that any changes to the base level versions are not incorporated.
- **Remove customized forms** – When selected, specifies that FormSync is to delete customized forms and replace them with the new base-level versions.

  Unless you also select the **Prompt before removal** option, this means that all base-level customizations are lost during synchronization.

- **Prompt before removal** – When selected, specifies that FormSync is to prompt you whether to remove or keep each customized version of a form. This option is enabled only if the **Remove customized forms** option is selected.

  This option allows you to decide on a case-by-case basis, which customized versions of forms to keep and which to remove and replace.

- **Merge customized forms with base-level forms** – When selected, specifies that FormSync is to merge any forms that have been customized with the new base-level versions.

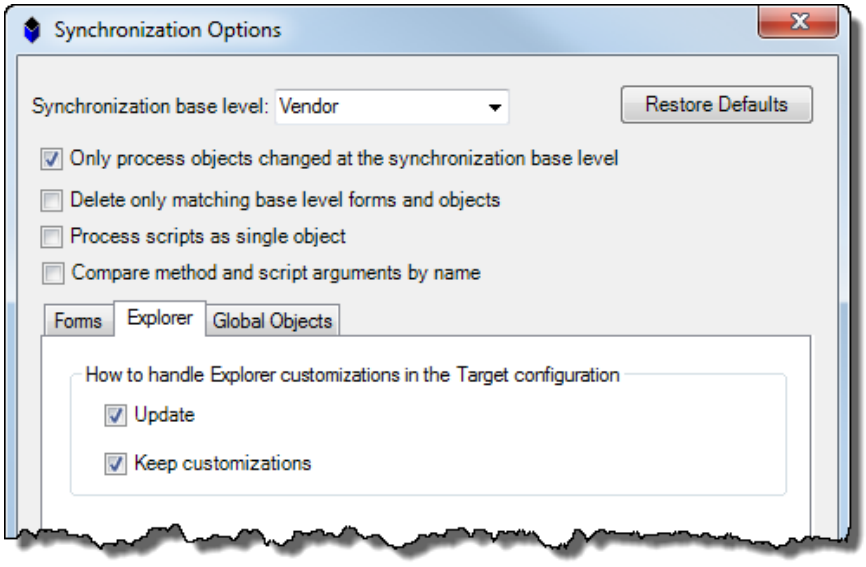
  This option keeps your customizations while at the same time incorporating any base-level updates to the form.

  To further specify which form and form component customizations should (or should not) be processed, click **Merge Options** and use the **Merge Options** dialog box.

- **Merge Options** button – Opens the **Merge Options** dialog box, which provides settings for specific attributes of matching components, non-matching components, and form attributes. For more information, see these sections:
  - “Matching Components Tab” on page 31
  - “Non-matching Components Tab” on page 32
  - “Customized Form Attributes Tab” on page 33
- **Filter: Form Name** – (Optional) Use this field to specify the names of one or more forms to include in the synchronization process. To process all forms, leave this field blank. For more information, see .

Explorer Tab

These are the default settings for the **Explorer** tab:



- **Update** – When selected, allows FormSync to update the Explorer with any changes in the new base-level version. In this case, whether your customizations are kept or discarded depends on the **Keep customizations** option.

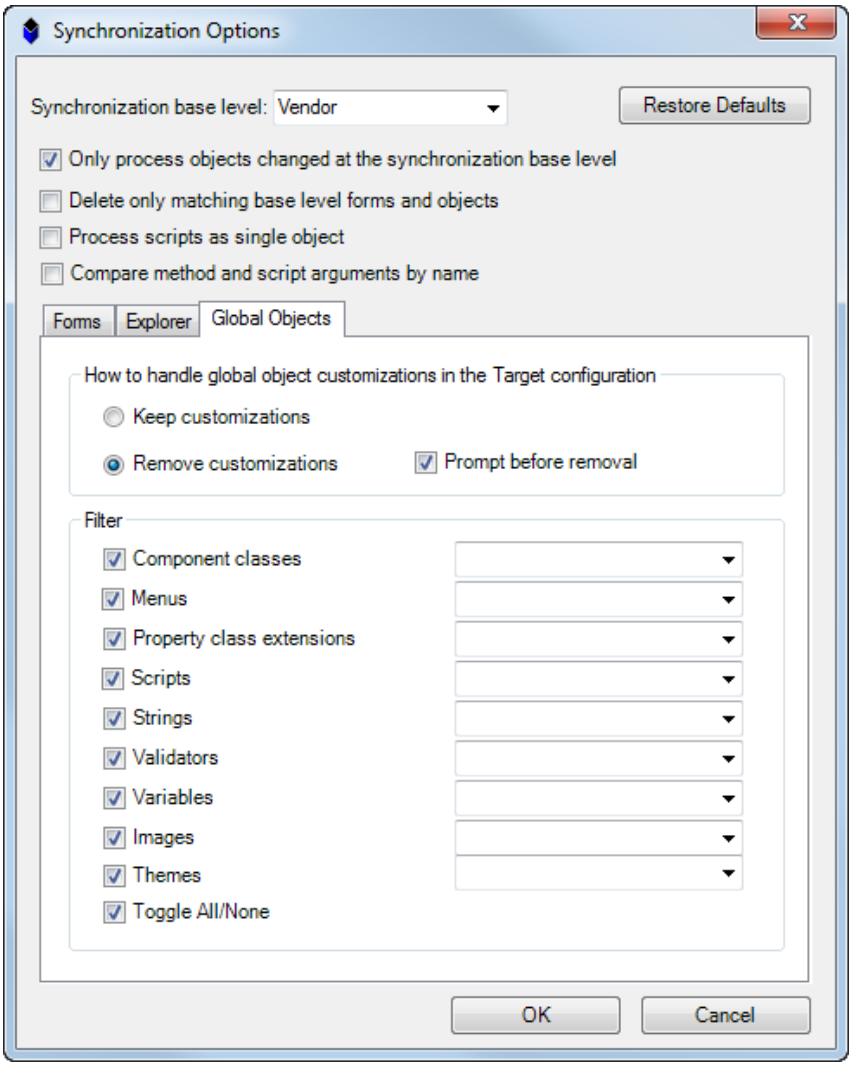
  If this option is cleared, FormSync does not update the Explorer at all.

- **Keep customizations** – When selected, FormSync keeps your customizations. When cleared, Form Sync replaces your customizations with the new base-level version.

  This option is enabled only when the **Update** option is selected.

Global Objects Tab

These are the default settings on the **Global Objects** tab:



- **Keep customizations** – When selected, specifies that FormSync is to keep all your global object customizations. This means that global objects are not processed at all during synchronization.

- **Remove customizations** – When selected, specifies that FormSync is to delete customized global objects and load the new base-level versions.

  Unless you also select the **Prompt before removal** option, this means that all base-level customizations are lost during synchronization.

- **Prompt before removal** – When selected, specifies that FormSync is to prompt you whether to remove or keep each customized version of a global object. This option is enabled only if the **Remove customizations** option is selected.

  This option allows you to selectively decide which customized versions of forms and objects to keep and which to remove and replace.

- **Filter** – When selected, tells FormSync to process the global objects for the specified type.

  If you select the check box for a type of object, FormSync keeps customized objects of the type in the Target configuration or removes them from it, depending on whether you select **Keep customizations** or **Remove customizations**. In both cases, FormSync replaces uncustomized base versions of objects of the selected type in the Target configuration with base versions from the Source configuration.

  If you clear the check box for a type of object, FormSync does not process objects of that type. In this case, FormSync ignores customized objects and does not replace base versions in the Target configuration with base versions from the Source configuration.

  You can restrict the processing of global objects to only those objects that match filter criteria based on object name. Specify a filter in the drop-down list next to a selected type:

  - To process all objects, clear the field.
  - To process a single object, select a name from the list.
  - To process multiple objects with similar names, use the percent sign (%) as a wildcard character.

    For example, to process all objects whose names begin with “Acct”, type **Acct%**.

Merge Options Dialog Box

The **Merge Options** dialog box provides options that allow you to further specify or narrow down the course that FormSync takes during synchronization. This dialog box is accessed byclicking the **Merge Options** button on the **Forms Tab** of the **Synchronization Options** dialog box.

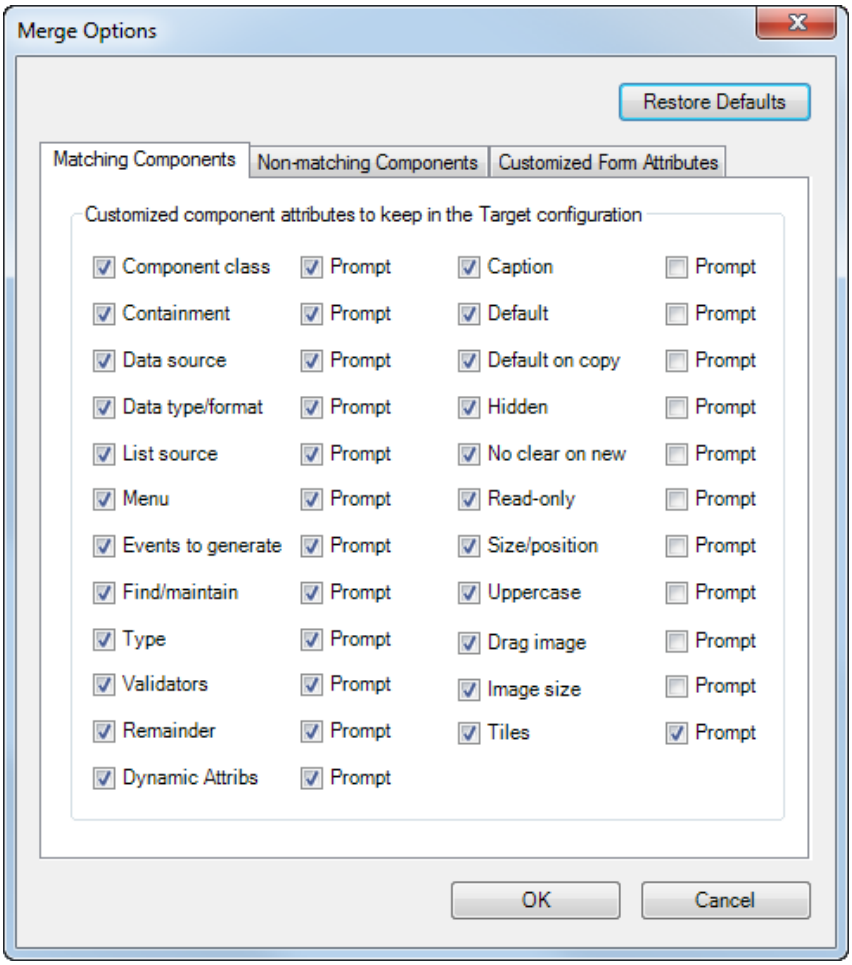
This dialog box has three tabs:

- Matching Components Tab
- Non-matching Components Tab
- Customized Form Attributes Tab

Matching Components Tab

A matching component is a customized component in the Target database for which there is a corresponding component on the same form in the Source database. The components match in the sense that they have the same name in the Target and Source. Possible matching components are listed on the **Matching Components** tab of the **Merge Options** dialog box.

These are the default options that specify what FormSync is to do with specific attributes in cases where there are matching components in the Source and Target configurations.



During synchronization, Form Sync checks the selected attributes for each component. For any attribute that is not selected, Form Sync ignores that attribute during synchronization and automatically keeps any customizations.

The **Prompt** option for each attribute is enabled only if that attribute option is selected. This option determines what FormSync is to do in situations where differences exist. If the **Prompt** option is:

- Selected, then FormSync prompts you as to whether you want to keep each customization.
- Cleared, then FormSync does not prompt you and automatically keeps the customization.

Selecting an attribute name means that FormSync will keep a customized value of the attribute in the Target database or prompt the user to keep it, depending on whether the corresponding **Prompt** check box is also selected.

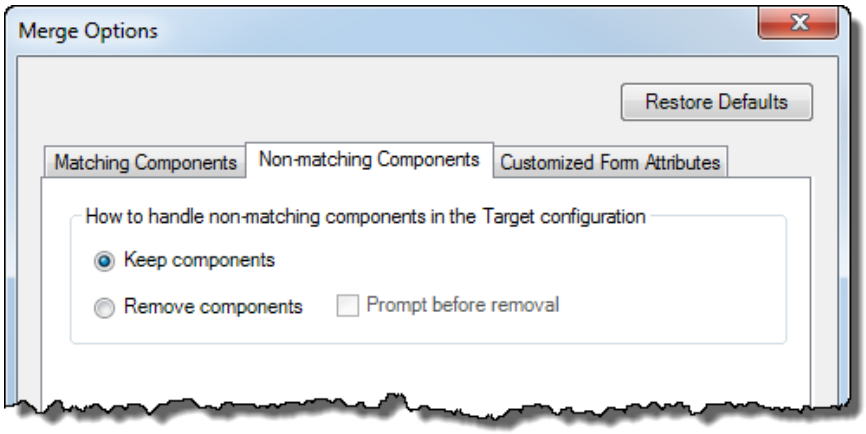
Note: **Prompt** check boxes are selected by default for most attributes. Attributes that are not selected (in the illustration above) correspond to customizations allowed to a user with Basic editing permissions. These attributes typically pertain to the user interface and do not affect data or system integrity. Changes to the values of these attributes are referred to as *Basic customizations*. For information on levels of editing permissions, see the WinStudio help.

Effects of the default **Prompt** settings are described in “Results of Default Synchronization” on page 34.

Non-matching Components Tab

A non-matching component is a customized component in the Target database for which there is no corresponding component on the same form in the Source database. A non-matching component is a new component added to the customized form by a customer developer with Full User or Site Developer editing permissions.

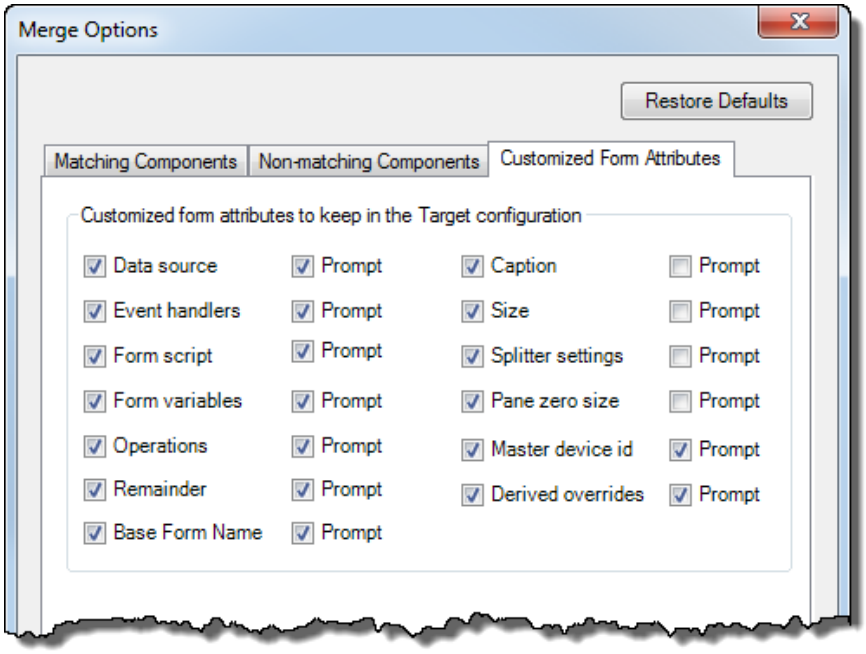
These are the default options that specify what FormSync is to do when it finds form components that exist in the Target configuration but not the Source configuration.



- **Keep components** – Tells FormSync to keep the non-matching component in the Target configuration.
- **Remove components** – Tells FormSync to remove the non-matching component from the Target configuration.
- **Prompt before removal** – Enabled only if the **Remove components** option is selected. This option tells FormSync to ask before removing a non-matching component. This allows you to select which non-matching components are removed from the Target configuration and which are not.

Customized Form Attributes Tab

These are the default settings for options that specify what FormSync is to do when it finds form-level attributes in cases where there are differences between the Source and Target forms.



During synchronization, FormSync checks the selected attributes for each form. For any attribute that is not selected, FormSync ignores that attribute during synchronization and automatically keeps any customizations.

The **Prompt** option for each attribute is enabled only if that attribute option is selected. This option determines what FormSync is to do in situations where differences exist. If the **Prompt** option is:

- Selected, then FormSync prompts you as to whether you want to keep each customization.
- Cleared, then FormSync does not prompt you and automatically keeps the customization.

To revert all options to their default settings—perhaps so that you can start over—, click **Restore Defaults**.

Results of Default Synchronization

This section describes the results when the default settings are used for synchronization of forms, of the Explorer, and of global objects.

Forms

With default settings, FormSync merges customized versions and vendor versions like this:

- **Syncnronizing attributes** – Synchronization automatically updates an attribute of a customized version in the Target database if the attribute was not changed by a customer developer. Developers typically change only a few attributes in a customized version of a form. FormSync updates any of the other attributes.

  ***Example*:** In a customized form, a customer developer changes the list source for a combo box. The developer does not change event handlers in the customized form. The new vendor version of the form in the Source database contains a modification to one of the event handlers. Synchronization replaces the event handler in the customized form with the modified event handler from the Source version, while keeping the list source change for the combo box.
- **Merging new components** – Synchronization automatically inserts in the customized version any new component found in the Source vendor version. (A new component in the Source version is one added to a form as a product enhancement. It has no counterpart in the vendor version of the Target database.)

  ***Example*:** The vendor version of a form in the Source database contains a new text field. Synchronization automatically inserts the text field into any customized versions of the form in the Target database.
- **Resolving differences in Source, Target, and customized versions** – Synchronization does not automatically update an attribute that is changed in the customized version and revised in the vendor version in the Source database. In this situation, the three versions of the form—the customized version, the Target vendor version, and the Source vendor version—each have different values for the same attribute. FormSync prompts the user to keep the changed attribute in the customized form or to replace it with the revision from the Source database.

  Replying appropriately to these prompts requires an understanding of the three versions of the form. The Compare utility in FormSync, which allows you to view specifications for two objects side by side, can be an important aid in replying to prompts (see the online help for FormSync). Changes in the updated version might be incompatible with the customized version. Testing, redevelopment, and manual application of revisions might be required. For more information, see

Chapter 4, “Prompts and Testing."

Note: An exception to prompting behavior: The default settings override prompting behavior for those attributes that correspond to Basic customizations. FormSync automatically keeps the customized versions of these attributes in the Target database. In the Merge Options dialog box, the **Prompt** check boxes for the attributes are cleared.

Example: A customer developer changes a validator for a component in a customized version. Specifications for the validator are also revised in the vendor version in the Source database. Because the **Prompt** check box for **Validator** is selected, FormSync prompts the user to keep the customization or to replace it with new specifications from the Source database.

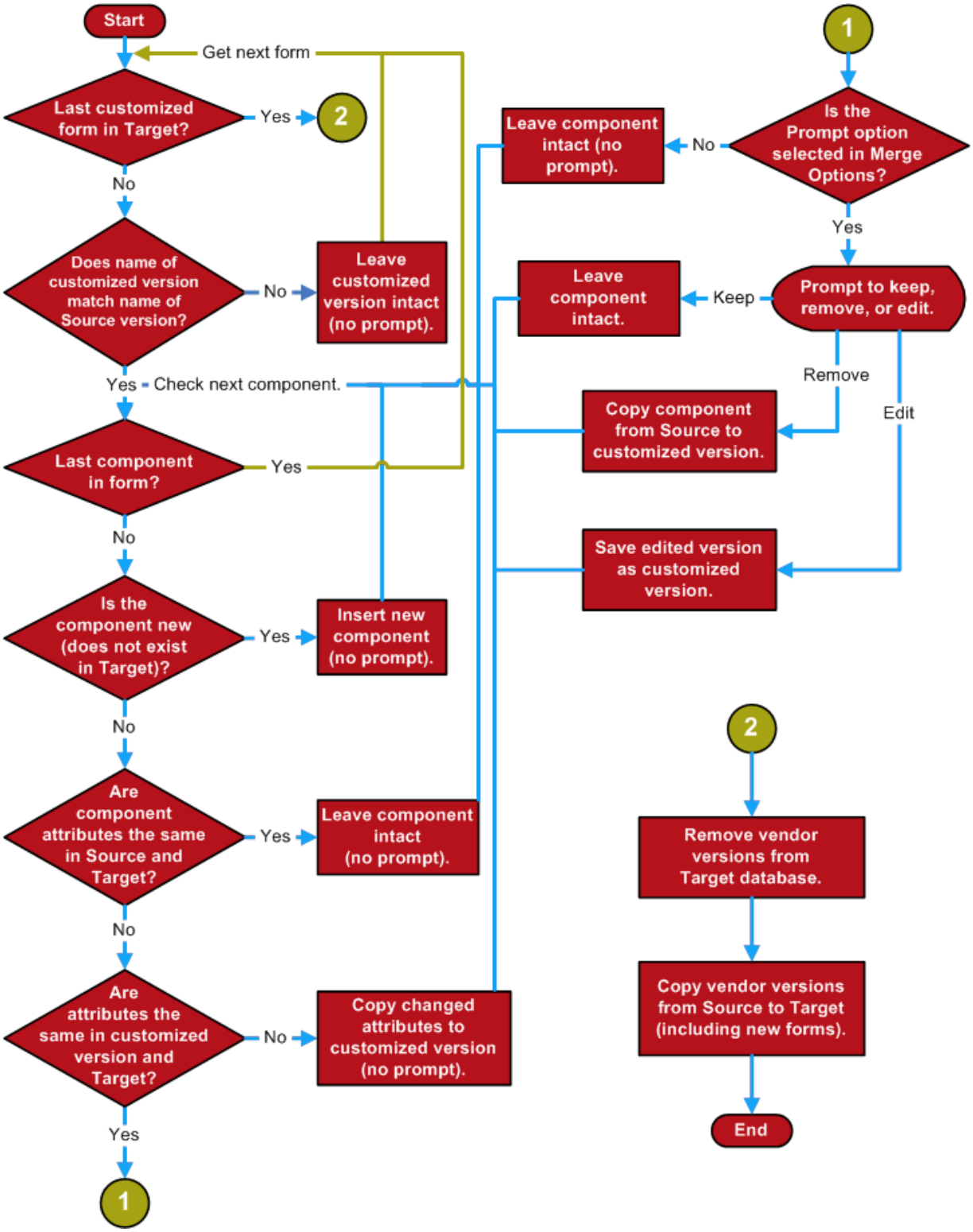
Example: A customer developer increases the width of a component in a customized version. The width setting has also been changed in the vendor version in the Source database. Because the **Prompt** check box for **Size Pos** is not selected, FormSync automatically keeps the customized width setting in the customized version.

- **Replacing vendor versions** – Synchronization automatically removes all vendor versions of forms from the Target database and replaces them with the revised vendor versions from the Source database.

Unless the **Delete only matching base level forms and objects** option is selected, default synchronization removes all base level forms and components, including new forms or components, in the Target database before replacing them with the new Source forms and components. A new form or component is one saved in the Target database under a name specified by a customer developer. Default synchronization also deletes customized versions from the Target database.

The program flow for default synchronization of forms is shown in the diagram on the next page. Following that is a text description of the flow.

Program Flow Diagram



Description of the Program Flow

- FormSync loops through the customized versions of forms in the Target database. If the name of a customized version of a form:
  - Does *not* match the name of a vendor version in the Source database, then FormSync:
    - Leaves the customized version of the form intact, without prompting you.
    - Goes to the next customized form (loops back).
  - *Does* match the name of a vendor version in the Source database, then FormSync compares each component of the Source vendor version of the form with the corresponding component of the Target vendor version of the form. If the name of the component in the Source version:
    - Does *not* match the name of a component in the Target version (that is, it is a new component), then FormSync inserts the new component into the customized version in the Target database, without prompting you.
    - *Does* match the name of a component in the Target version (that is, already exists), then FormSync compares attributes of the component in the Source version with attributes in the Target version. If an attribute in the Source version is:
      - The *same* as the corresponding attribute of the Target version, FormSync leaves the attribute of the customized version of the form intact, without prompting you.
      - *Different* from the attribute of the Target version *and* the corresponding attribute of the customized version is:
        - The *same* as the attribute of the Target version, FormSync copies the attribute of the Source version over the attribute of the customized version, without prompting you.
        - *Different* from the attribute of the Target version, FormSync checks the status of the **Prompt** option in the **Merge Options** dialog box. If the **Prompt** check box is:
          - *Cleared*, FormSync leaves the attribute of the customized version of the form intact, without prompting you.
          - *Selected*, FormSync prompts you whether to:
            - **Keep** the attribute of the customized version. This option tells FormSync to leave the attribute of the customized version of the form intact.
            - **Replace** the attribute of the customized version. This option tells FormSync to copy the attribute of the Source version over the corresponding attribute of the customized version.
            - **Edit** the customized version before proceeding (in cases where a script or event handler is involved). If you select this option, FormSync saves the edited version as the new customized version.
- FormSync removes all vendor versions of forms from the Target database.
- FormSync copies the vendor versions of forms, including any new forms, from the Source database to the Target database.

The Explorer

With the default settings, FormSync:

- Replaces the **Master Explorer** in the Target database with the **Master Explorer** from the Source database.

- Keeps all customized **Public** folders and **My Folders** folders in the **Explorer** in the Target database.

The **All Forms** folder is dynamically created from the forms database, so it automatically includes any new vendor versions of forms after the Target database is synchronized.

Global Objects

With the default settings, FormSync:

- Prompts you to remove a customized global object or to keep it. (See Chapter 4, “Prompts and Testing.”)
- Removes vendor versions of global objects from the Target database and replaces them with revised vendor versions from the Source database.
- Does not remove or change new global objects in the Target database. (A new object is one saved under a developer-specified name in the Target database.)

Chapter 4: Prompts and Testing

4

This chapter contains information and recommendations regarding customizations that cannot be merged automatically with a high degree of confidence.

Note: Most of the information in this chapter does not apply to synchronization using the **Form Sync** form in WinStudio. The form version in WinStudio does provide prompts, but all at once at the end of FormSync's first pass through the configuration being updated. Prompts at that point are limited to "Keep" and "Remove" (replace) options; no merge options are supported.

Overview

If your forms database contains customizations other than basic changes, you should plan to compare customized form attributes with new vendor versions of the attributes. You can then decide whether to keep a customization or to replace it with the new version and then to reapply your changes to the new version.

We recommend that you select FormSync's option to record prompts for replay later. This mode provides information about customizations and new vendor versions, which you can consider before you commit changes through the prompt mechanism. In this mode also, you can run FormSync unattended. (For prompting options and procedures, see the FormSync online help.)

After using FormSync, you should plan to test:

- Customized forms in which you kept a customization in response to a FormSync prompt.
- Any new forms you or your organization might have created.
- Forms that use a customized global object that you kept in response to a FormSync prompt.

If you use the option to record and replay prompts later, you must then replay the FormSync prompts and respond to them before changes are committed to the forms database.

Prompts

Synchronization prompts for most customizations allow you to keep a customization intact in the Target database or to replace it with a new version from the Source database. Prompts for event handlers and scripts provide an additional option to edit customizations. With this option, you can view the two versions side by side and edit the customized version in the Target database.

A prompt to keep, replace, or edit an attribute value indicates that the value has been:

- Customized and
- Changed in the new vendor version relative to the old vendor version

Note: By default, Basic customizations do not generate prompts, though they might meet these two conditions. For more information about Basic customizations, see the FormSync online help.

A customization that was functional in the old version of the application will not necessarily be functional in the new version. Also, if you keep a customization, any enhancement or fix implemented in the new version will not be applied to the Target version.

> ***Example***: A customer developer changes an event handler of the Run Script type so that it invokes a different script. The script depends on getting the value of a component on the form. In the new vendor version, the data type of the component is changed. FormSync changes the data type in the customized form without prompting the user because the component was not customized in the old version. FormSync prompts the user to keep or replace the changed event handler. If the user keeps the customized script, the event handler might fail.

When the automated part of synchronization is finished, you can review the content of prompt messages recorded in the FormSync log. Messages show which attributes are changed in both the customized version and the new vendor version. For most attributes, the messages show the customized value and the new value of a changed attribute. (For tips on using the log, see the FormSync online help.)

You can base a plan on information from the log and on your organization’s development policies and procedures. Plans for handling a specific attribute will likely resemble one of the following:

- Replace the customization with the new version.
- Replace the customization with the new version and subsequently reapply changes manually in WinStudio.
- Keep the customization.
- Edit the customization.

You should consider dependencies within the form and within the system when you choose to keep a customization. An update can modify:

- An IDO or other resource on which the customized object is dependent. (See the “Objects Not Processed by FormSync" table on page 9.)
- A vendor version of a form or global object on which the customized object is dependent.
- A vendor version of a form or global object that depends on the vendor version of the customized object.

After any needed manual editing in WinStudio, you should test customized forms. A test environment must be parallel to the updated production environment, with forms accessing the updated IDOs, application database, and other applicable parts of the application.

New Objects

Objects based on a default version and saved under a new name in the Target database are considered new objects, as are newly created objects that you save under a name you specify. Because FormSync does not examine new objects in the Target database during synchronization, you should consider system dependencies related to customizations, and you should test the objects as noted in the “Prompts" section.

> ***Example*:** A customized form based on a default vendor version is saved under a new name. A button on the form uses the vendor version of an event handler. The event handler is of the type that calls an IDO method, which takes a certain number of parameters. In the updated middle-tier, which FormSync does not examine, the IDO method is modified to require an additional parameter. Because synchronization does not replace the old vendor version of the event handler with the new vendor version, the method call fails in the updated version.
>
> ***Example*:** A customer developer adds a button to a form, creating a new component. The developer creates a new event handler, which uses an IDO method, for the button. In the updated middle-tier, the IDO method is modified to require a different type of parameter. Synchronization does not alter the new component, and so, the event handler fails when it calls the modified IDO.

You should adopt naming conventions for new objects to distinguish them from other customized versions. FormSync does not generate prompts regarding new objects.

Chapter 5: Synchronizing Site Default or Group Versions

5

This chapter provides information specifically for synchronizing forms in a multi-site environment or where groups have customized forms.

Overview

You can use FormSync to deploy revisions to a form customized at a Site Default or Group scope type, so that other system users' customizations are synchronized with the revision. This approach simplifies the customization process because those users do not have to revert their customized forms or recode them. You can similarly deploy revisions of customized global objects.

Synchronization with Site Default or Group versions follows the same logic as synchronization with vendor versions in a vendor-supplied application upgrade. The *Source* database contains the revised versions of forms customized with a scope type of Site Default or Group. The *Target* database contains customized versions that are to receive the revisions.

Note: Customer developers should revise forms in a development environment. This development environment then serves as the Source for synchronization. For more information, see “Using a Development Environment” on page 44.

To synchronize Site Default or Group versions, in the **Synchronization Options** dialog box, the **Synchronization base level** box, designate a base-level scope type of either **Site Default** or **Group**. (For more information, see the FormSync online help.) Forms and global objects with the designated scope type then serve as the base-level versions for synchronization, just as vendor versions serve as base-level versions in an upgrade.

FormSync then merges definitions of forms with the designated scope type in the Source (development) database with any customized forms that are based on forms with the same scope type in the Target (production) database. Forms with the designated scope type in the Source database replace forms with the same scope type in the Target database.

> ***Example***: A customer developer with Site Developer editing permissions adds a component named **DueNow** to a form. The developer specifies **Group** as the scope type and **AccountsReceivable** as the scope (group) name.

A member of the **AccountsReceivable** group with the user ID **JJones** changes the caption of the component. This change is at a scope type of **User**, and the scope name is **JJones**.

In a separate development configuration, the developer adds an event handler to the form.

In FormSync, the developer specifies **Group** as the synchronization base level, **DueNow** as the form name, the development configuration as the Source configuration, and the production configuration as the Target configuration. The developer then synchronizes the two configurations.

Following synchronization:

- JJones's version of the form keeps the customized caption and also contains the new event handler.
- The revised version of **DueNow** with scope type **Group** and scope name **AccountsReceivable** replaces the first customized version of the form with the same scope in the Target database.
- The vendor version of the form is unaffected.

Using a Development Environment

To synchronize customizations made at a scope type of Site Default or Group with customizations in the production environment, you must have a development environment separate from the production environment.

We recommend that you make a copy of the production forms database and create a Source configuration that allows you and other developers in your organization to access that database. To make customizations at the Site Default or Group level, developers must have Site Developer editing permissions.

Developers can then make changes in the new development environment and test them safely without disrupting business operations in the production forms database. Only after you and your developers are satisfied that everything is working as you want should you synchronize the development environment with your production environment.

When you synchronize objects customized at Site Default or Group scope type with objects in the production environment, specify the:

- Development environment as the Source configuration.
- Production environment as the Target configuration.

Chapter 6: FormSync Utilities

6

In addition to its synchronization capabilities, FormSync also offers a number of other form and global object administration utilities. This chapter provides an overview of these utilities, along with recommendations for when and why to use them.

Note: The information in this chapter does not apply to the synchronization capabilities available using the **Form Sync** form in WinStudio.

Overview

The FormSync utilities provide you the capability to directly perform actions from a Source configuration database to a customized object in a Target configuration database. These actions do not affect the vendor versions of the objects in the Target database, only the customized versions.

These utilities are useful primarily to those who customize forms using a development forms database and subsequently copy the customizations to a production forms database. Utility operations do not alter the Mongoose-based application version number or service-pack number in a forms database.

These utilities can also be useful when synchronizing, to help you determine what changes to make, what versions to keep, and so forth.

Accessing the FormSync Utilities

To access all FormSync utilities, you must open the **Utilities** dialog box. To open the **Utilities** dialog box, open the main **Form Synchronization** window, and then do one of the following actions:

- From the **View** menu, select **Utilities**.
- On the toolbar, click the Utilities  button.

Setting the Filter

Most of the FormSync utilities require you to first set the filter so that only those forms and global objects that you want to process are affected.

When the **Utilities** dialog box first opens, the **Filter** tab is automatically selected. This tab is where you set the options and select the forms and objects you want to:

- Copy (**Copy** tab)
- Delete (**Delete** tab)
- Generate and save a list for (**Save List** tab)
- Generate a SQL script for (**Script** tab)

You do not need to make settings on this tab if you want to compare objects (**Compare** tab) or generate "pseudo code" (**Generate Pseudo Code** tab).

To make filter settings:

1 In the **Configuration** fields, verify that you have the correct **Source** and **Target** configurations selected.

   If not, use the **Change Configurations** button to select other configurations.

2 In the selection group box, select the object types you want to perform the action on. Clear the check boxes for object types you do not want to process.

3 Click **Refresh List**.

4 In the **Custom Objects** list, select the **Include** check box for all custom objects you want to process. For objects that you do not want to process, clear the check boxes.

5 (Only if you are deleting objects) Select the **Delete** check box for all custom objects you want to delete.

For more information about these options, see the context-sensitive online help for the desired option.

About the Utilities

The following sections describe briefly what each utility does and when or why you would want to use it.

Delete Utility

The Delete utility allows you to delete custom objects from the designated Target configuration database. This utility can be useful when you want to get rid of test or obsolete objects without having to synchronize your Target database with a Source database.

For the procedure to use this utility, see "Deleting Custom Objects" in the FormSync online help.

Save List Utility

The Save List utility allows you to generate and save a list of custom objects in a Source or a Target database. This utility is especially useful if you are not sure whether you have any customized objects in your configuration or not, and when you want to see a list of all customized objects so that you can prepare for a synchronization.

In addition to the name of the object, this list includes:

- The object type (form, script, string, and so on)
- The scope type (**Vendor**, **Site**, **Group**, or **User**)
- The scope name, if any (**Group** and **User** scope type only)

FormSync saves this list as a comma-separated-values (*.csv) file to a location you designate. You can view this file in any spreadsheet program that handles .csv files.

For the procedure to use this utility, see "Saving a List of Custom Objects" in the FormSync online help.

Script Utility

The Script utility allows you to generate and save a SQL script that contains definitions of custom objects existing in the Target forms database. You can then use a tool like SQL Query Analyzer to import the script into another configuration database.

For the procedure to generate SQL scripts, see "Generating SQL Scripts for Custom Objects" in the FormSync online help.

Copy Utility

The Copy utility allows you to selectively copy forms and objects from a Source database to a Target database, replacing only selected objects. This utility does not make any attempt to merge contents of existing versions or preserve any customizations in the Target database. It simply replaces selected existing versions with the new versions from the Source.

For the procedure to copy objects from one database to another using this utility, see "Copying Custom Objects Between Databases" in the FormSync online Help.

Compare Utility

The Compare utility allows you to compare different versions of objects, usually a new one from a Source database and a customized one from a Target database. This utility allows you only to view the differences between the two versions. You can then either use WinStudio or the merge options in FormSync to make any desired changes to the custom object.

This utility is especially useful when you are trying to synchronize different versions. If, while synchronizing, FormSync finds a difference between the Source and your customized version, you can open a new session of FormSync and use this utility to compare the new and the custom version. If you then want or need to make changes to the customized version, you can make them (not with this utility, though) before proceeding with synchronization.

For the procedure to compare objects, see "Comparing Objects" in the FormSync online help.

Generate Pseudo Code Utility

The Generate Pseudo Code utility allows you to create files containing "pseudo code" for selected objects in a database. You can generate this pseudo code for as many objects as you want. You can later use this pseudo code to compare the contents of different versions of objects, which can in turn help you identify where an object in the Target database might need to be edited to work with the upgraded files.

For the procedure to generate pseudo code files, see "Generating Pseudo Code" in the FormSync online help.

Generate License Records for Customer-Created Forms Utility

Use this utility to generate a "CustomerForms" license record for your customer-created forms that aren't currently licensed. This utility validates that the specified forms are not already licensed to an existing module. The generated license records are based on the "UserForms" modules specified in the current license in effect and are inserted into the ModuleMembers table in the application database of the specified configuration.

For the procedure to generate license records for customer-created forms, see "Generating License Records for Customer-Created Forms" in the FormSync online help.

Using Multiple Sessions of FormSync

When you are synchronizing databases or performing other types of database updates, it can be helpful to run multiple sessions of FormSync using these utilities to help you evaluate customizations and troubleshoot problems.

Index

G

T

U

V