

Visual Basic Scripting

See Also

Commence integrates Visual Basic Scripting (VBScript). It is now possible to create scripts that will guide the user through data entry for a form. Field validation criteria can now be defined so that newly-entered data is checked immediately.

Commence permits VBScript code to be written for each form within a database. This enables such operations as setting default values, filling in fields, and navigating within the form, as well as performing validation checks before an item is saved.

VBScript code can be written to handle various events, such as the loading of a form, entering or leaving tabs, pressing the **Save** button, etc.

While processing these events, VBScript code can read and write field values, move the cursor to different fields and/or tabs, display messages, abort a requested save, etc.

One benefit of this new feature is faster data input, since scripts are able to fill in field values based on other field values, or even skip over some fields. Better data validation is achieved since the data is now able to be checked prior to saving an item.

Previously an item would have to be edited in order to correct any invalid data entries.

An additional benefit for users in a workgroup environment is the significant reduction of workgroup traffic. Since values set by VB Script code are in place prior to the **Add** or **Edit** operation, only a single transaction needs to be distributed over the network.

Commence 3.1 typically performed these operations using Agents which operate after an item has been saved. This generates additional **Edit** transactions which must then also be distributed over the network.

What is VBScript

See Also

VBScript is a programming subset of Visual Basic from Microsoft. Originally, VBScript was developed to assist programmers in the design of visually appealing and complex Internet home pages. However, it has been used by programmers to develop various applications for other software packages as well. The commands included in VBScript are restricted to the extent that there is no low-level programming interaction with either application software, system hardware, or the Operating System. What this means is that it is not possible to accidentally format the system hardware by running script software. However, VBScript is a very powerful tool when used to automate many "behind the scenes" processes, such as data validation and automatically filling in known information.

VBScript is a full-featured programming language which supports a wide range of data types, as well as other features typical of most programming languages in use today. This includes **If - Then - Else** constructs, looping (using **Do...Loop**, **While...Wend**, and **For...Next** loops), variable definitions (both scalar and array), the definition of constants (both numeric and string), and a full range of operators, including arithmetic, logical, comparison, and concatenation operators.

For a complete guide to VBScript and its uses, refer to the Microsoft Home Page, located at <http://www.microsoft.com> for the latest information and support.

What Can A VBScript Do?

See Also

When VBScript is used to develop scripts for an application, the application defines to VBScript the scope within which it is to operate. VBScript is provided very specific instructions as to:

- on what it may operate (OBJECTS),
- what functionality will be available to the programmer from within the script (METHODS), and
- under what circumstances it will be called (EVENTS).

Commence limits the available operations that are able to be performed within a script to those that affect Detail Form navigation, data entry, and saving of forms.

Detail Forms

Detail Form Scripting

Script Sample

Commence VBScript Objects

[See Also](#)

<u>Form</u>	Default object referenced by VBScript methods.
<u>Application</u>	Commence application object used for settings that are global to the application and all forms.
<u>Field</u>	Object that represents a single field on the form.
<u>Connection</u>	Object that represents a single connection on a form.

Form Object

[See Also](#)

[Properties](#)

[Methods](#)

[Events](#)

This object represents the detail form and allows access to the tabs and fields it displays. This is also the default object that is referenced by a script

Represents the Detail Form. The Form object includes properties and methods that allow you to navigate and control Detail forms.

Commence Objects

Detail Forms

Detail Form Scripting

Script Sample

Form Object Properties

Relates To

All the Form properties are read-only unless otherwise noted.

All properties are read-only unless otherwise noted.

Name	String, name of the form, ie. "Initial Form"
CategoryName	String, name of the category the form is associated with, ie. "Person"
ItemName	String, name of the current item. May be empty when adding an item and the name field has not yet been entered.
TabName	String, Name of the current Tab
FieldName	String, Name of the current field (this field has the focus)
Application	Object, this property contains the Application object. This property can be used directly or copied to a variable and used in a script
IsAdd	Boolean, set to TRUE if the form is displaying a new item, else FALSE if the form is displaying an existing item
IsShared	Boolean, set to TRUE if the item "Share" button on the toolbar is pressed (indicating the item is shared or will be shared after a save command).
FieldValue	VARIANT, returns the value of the current field with the focus, usually as a string

Form Object

Detail Forms

Detail Form Scripting

Script Sample

Form Object Methods

Relates To

MoveToTab	Switch to the tab whose name is passed as an argument.
MoveToField	Switch to the field whose name is passed as an argument.
SetShared	Sets the state of the "Share" button on the detail toolbar.
Save	Saves the item displayed by the form. Subject to errors detected by Commence, so the item may not be saved. Will not generate a Form_OnSave event to prevent infinite looping.
Cancel	Cancels the item detail window and discards any changes. Will not generate a Form_OnCancel event to prevent infinite looping.
Abort	Aborts the current event. Ie., if called from Form_OnSave it will cancel the Save operation. If called from Form_OnEnterField it will cancel the field navigation and return the focus to the original field.
SetValue	sets the current field's value to the passed in VARIANT, which is usually a string.
Field(name)	where name is a string. Gets an OLE object (Idispatch interface) for a field object representing the named field. On error, the ERR object will be set with 'invalid arg' error code.
Connection(name, ToCategory)	where name and ToCategory are strings. Gets an OLE object for a connection object representing the named connection. On error, the ERR object will be set with 'invalid arg' error code.

Form Object

Detail Forms

Detail Form Scripting

Script Sample

Form Object Events

Relates To

Form_OnLoad	Occurs when the detail form is opened.
Form_OnSave	Occurs when the detail form is saved
Form_OnCancel	Occurs when the detail form cancelled (not saved)
Form_OnEnterTab	Occurs when a detail form tab is opened
Form_OnLeaveTab	Occurs when leaving a detail form tab.
Form_OnEnterField	Occurs when the focus moves into a field.
Form_OnLeaveField	Occurs when the focus leaves a field

Application Object

[See Also](#)

[Properties](#)

[Methods](#)

Represents the Commence application. The Application object allows access to settings that are valid for Commence as well as all forms.

[Commence Objects](#)

[Detail Forms](#)

[Detail Form Scripting](#)

[Script Sample](#)

Application Object Properties

Relates To

This object represents the Commence application and allows access to settings that are valid for the application and all forms.

All properties are read-only unless otherwise noted.

CurrentScriptLevel	Number, contains the script level set by the script. To change the script level (which controls the VBScript objects supported by Commence), assign the desired level to this property. The level must be greater than 0 and less than or equal to DefaultScriptLevel
Database	Accesses the Database API for managing other Commence items
DatabaseDirectory	String, DOS path of the active Commence database, ie., "c:\commence\data\"
DatabaseName	String, Name of the active Commence database, ie., "Main Database"
DefaultScriptLevel	Number, contains the maximum script level supported by this version of Commence
Name	String, application name, usually "Commence"
ProgramDirectory	String, Commence program directory, often "c:\commence"
ProgramName	String, Commence .exe file name, often "c:\commence\commence.exe"

Application Object
Detail Forms
Detail Form Scripting
Script Sample

Application Object Methods

Relates To

GetVersion

returns the Commence version number in the form "4.1.x.y"

IsScriptLevelSupported

returns 1 if the passed in script level is supported, else returns 0 if it is not supported

Application Object
Detail Forms
Detail Form Scripting
Script Sample

Field Object

[See Also](#)

[Properties](#)

Represents the Commence field. The Field object allows access to settings that are valid for Commence as well as all forms.

Commence Objects

Detail Forms

Detail Form Scripting

Script Sample

Field Object Properties

Relates To

This object represents a single field value from the item displayed by the form object.

All properties are read-only unless otherwise noted.

Name	String, field name
Label	String, field's label on the current form tab
Value	VARIANT; contains number for number and calculation fields, string for most other fields. This property can be read and set; check the Err object for any errors on setting a field value. Note: Image fields are not supported in this release.

Field Object

Detail Forms

Detail Form Scripting

Script Sample

Connection Object

[See Also](#)

[Properties](#)

[Methods](#)

This object represents a connection field displayed by the form object. The state represented by this object is consistent with the state of the connection field as it is displayed on the form. If the connection has a filter and it is currently in effect, the object will return values consistent with the filtered items displayed by the connection. If there is no filter or if it has been turned off, then this object will return values consistent with no applied filter.

Commence Objects

Detail Forms

Detail Form Scripting

Script Sample

Connection Properties

Relates To

All properties are read-only unless otherwise noted.

Name	String, connection name, e.g. "Attends"
ToCategory	String, name of the connected 'To' category, e.g., "Appointment"
ConnectedItemCount	Number; count of connected items in this connection; takes into account the current state of the connection's filter
CurrentSelection;	Number; read-write property; index of connected item that is selected for use with this object's methods; reading this property will return the currently selected item; to select a different connected item, assign its 1-based index to this property. If there are no connected items, this property is initialized to 0, otherwise it is initialized to 1 (representing the first connected item)
ItemName	String; name field of the selected item, if any
ItemClarifyField	String; clarify field of the selected item, if any

[Connection Object](#)
[Detail Forms](#)
[Detail Form Scripting](#)
[Script Sample](#)

Connection Methods

Relates To

Clear()	Disconnects all connected items that meet the connection filter. If no filter is defined (via Customize-Database-Connection-Advanced) then it functions the same as ClearAll().
ClearAll()	Disconnects all connected items regardless of the connection filter
ClearConnection(ItemName, Clarify)	Disconnects a single, specific item identified by the ItemName and Clarify strings. Note that the "(-Me-)" keyword can be used for the ItemName parameter.
SetConnection(ItemName, Clarify)	Connects the item displayed on the form to a single, specific item identified by the ItemName and Clarify strings. Note that the "(-Me-)" keyword can be used for the ItemName parameter.
FieldValue(FieldName);	VARIANT; gets the field value for the named field in the selected, connected item. See description of the Value property for Field objects.

Connection Object

Detail Forms

Detail Form Scripting

Script Sample

VBScripting Error Object

The **Err** contains information about run-time errors generated during the execution of a VBScript. This is an intrinsic object to VBScript with a global scope. Consequently, there is no need to create an instance of it within any code that is written.

The **Properties** of the Err Object are set by the generator of an error, such as Visual Basic, an OLE object, or the VBScript programmer. The default property of an Error Object is **Number**. When a run-time error occurs, the properties of the Error Object are filled with information that uniquely identifies the error, as well as information that can be used to handle the error. In order to generate a run-time error within VBScript code, use the **Raise** method.

The Err Object's properties are reset to zero (0) or to a zero-length string (""), after encountering an **On Error Resume Next** statement, an **Exit Sub** statement, or an **Exit Function** statement within an error-handling routine. The **Clear** method can be used to explicitly reset the Error Object.

Some of the causes of run-time errors are listed below:

- Changes to field names with a form's category. If a script references invalid or outdated field names, the script will fail and generate a run-time error.
- Changing fields that are included on a form. If a script attempts to navigate to field names that are not displayed on the form, the script will fail.

NOTE

A script can set values for fields that are not displayed on a form, but it cannot navigate to these fields.

- Navigating or referencing invalid form tab names within a form. This may be a result of renaming or deleting tabs from a form.
- Attempting to set object properties that are identified as read-only properties.

To allow VBScript to continue processing a script after a run-time error has been detected, include the VBScript command **On Error Resume Next** in the script. To check for an error, examine the properties of the VBScript **Err** Object. A non-zero **Err.Number** error code will identify the run-time error.

