Visual Studio Integration Toolkit for Microsoft Dynamics GP (build 21)

Introduction

Visual Studio Integration Toolkit for Microsoft Dynamics® GP is a Dexterity® add-on software tool which contains an Application Programming Interface (API) that can be called from Visual Studio development system using Visual Basic® or Visual C# ® languages to integrate with functionality normally only available in Microsoft Dynamics GP via the Dexterity development environment, such as menu navigation.

Registration information

The basic functionality of Visual Studio Integration Toolkit is free but does require registration. It is David Musgrave's gift to the GP Community. All we ask is that the code and customer contact details are kept up to date. The advanced functionality of Visual Studio Integration Toolkit, such as the Custom Forms features, require a paid annual subscription to use.

Compatibility information

The tool was previously known as Menus for Visual Studio Tools. File and object names have not been changed to maintain compatibility with existing installations and projects. Each build of the tool is only compatible with the version of Microsoft Dynamics GP it was created for.

Contents of the Visual Studio Integration Toolkit for Microsoft Dynamics GP Archive

Below is a table of the files included with the installer:

File or Folder Name	Description
VSTMenus.cnk	Tool chunk file (self installing application Dexterity dictionary)
VSTMenus.pdf	User guide documentation (this document)
VSTMenus.txt	Quick usage guide with example C# and VB scripts and version history.
VSTMenus_License.doc	End User License Agreement
Application.MenusForVisualStudioTools.dll	Visual Studio Tools Assembly for Tool
Application.MenusForVisualStudioTools.xml	Visual Studio Tools XML descriptions for Intellisense
WinthropDC.VisualStudioIntegrationToolkit.dll	Visual Studio Tools Addin for Registration and system features (in Addins folder)
VSTMenus_Examples.zip	VSTMenusCSharp sample Visual C# project which add menus to Cards >> Sales menu VSTMenusVB sample Visual Basic project which add menus to Cards >> Purchasing menu

Installing Visual Studio Integration Toolkit for Microsoft Dynamics GP

1. Visual Studio Integration Toolkit is installed by downloading the installer and executing it. Follow the onscreen instructions to install the product files into the Microsoft Dynamics GP application folder.

Uninstalling Visual Studio Integration Toolkit for Microsoft Dynamics GP

1. Visual Studio Integration Toolkit is uninstalled from its About window (Help >> About Dynamics GP >> Additional >> About Visual Studio Integration. Then it can be uninstalled from Windows.

Installing Visual Studio Integration Toolkit for Microsoft Dynamics GP sample projects

1. From the VSTM_Examples.zip archive, extract the VSTMenusCSharp folder and/or VSTMenusVB folder to your desired location in your development system. There is no specific path recommended.
2. For each of the projects, perform the steps 3 to 13 below.
3. Open the project solution file (VSTMenusCSharp.sln or VSTMenusVB.sln) with Visual Studio.
4. In the Solution Explorer, expand References. To see References, you might need to click on the "Show All Files" button (second button from the left on the Solutions Explorer toolbar). For Visual Basic you will also need to expand the "My Project" node.
5. Right click on the reference for Application.MenusForVisualStudioTools and click Remove.
6. Right click on the References folder and click Add Reference.
7. On the Add Reference dialog, select the Browse tab.
8. Select the Application.MenusForVisualStudioTools.dll file installed previously the Microsoft Dynamics GP Application folder (default location: C:\Program Files\Microsoft Dynamics\GP).
9. Click OK.
10. Click to select the newly added Application.MenusForVisualStudioTools reference.
11. In the Properties window, set the Copy Local property to False.
12. From the Build menu, build the project.
13. Copy the resulting VSTMenusCSharp.dll or VSTMenusVB.dll file from the project's bin/Debug folder to the AddIns folder in the Microsoft Dynamics GP Application folder.
14. Launch Microsoft Dynamics GP.
15. Look at Cards >> Sales for the "VS C# Test" menu items, or Cards >> Purchasing for the "VS VB Test" menu items.

Using Visual Studio Integration Toolkit for Microsoft Dynamics GP in your own projects

Note: The following section is assuming that you have an existing Visual Studio Tools for Microsoft Dynamics GP project created.

To use the Visual Studio Integration Toolkit for Microsoft Dynamics GP, you will need to add a reference from your Visual Studio application to the Application.MenusForVisualStudioTools.dll file you installed in the Microsoft Dynamics GP application folder. Below are the steps:

1. In the Solution Explorer, expand References. To see References, you might need to click on the "Show All Files" button (second button from the left on the Solutions Explorer toolbar). For Visual Basic you will also need to expand the "My Project" node.
2. Right click on the References folder and click Add Reference.
3. On the Add Reference dialog, select the Browse tab.
4. Select the Application.MenusForVisualStudioTools.dll file installed previously the Microsoft Dynamics GP Application folder (default location: C:\Program Files\Microsoft Dynamics\GP).
5. Click OK.
6. Click to select the newly added Application.MenusForVisualStudioTools reference.
7. In the Properties window, set the Copy Local property to False.

Registering Events for Visual Studio Integration Toolkit

Next you need to add into the `Initialize()` function the commands to register for the two function events used by the tool for menus.

Register the following function event to call the script to register/create your menu entries:

`MenusForVisualStudioTools.Functions.EventRegister.InvokeAfterOriginal`

The event handler called from this event then contains the commands to create the menus as needed by your application.

Then, register the following function event to call the script to handle the callback when a menu entry is selected:

`MenusForVisualStudioTools.Functions.EventHandler.InvokeAfterOriginal`

The event handler called from this event then checks that the callback is caused by one of the menu items from this application and if so, performs the desired action, such as opening a Visual Studio Tools form.

Below are examples of the required event registrations for Visual C# and Visual Basic:

```
public void Initialize()
{
    // Register Event to add menu entries
    MenusForVisualStudioTools.Functions.EventRegister.InvokeAfterOriginal += new
EventRegisterFunction.InvokeEventHandler(VSTMEventRegister);
    // Register Event to handle callbacks from menu entries
    MenusForVisualStudioTools.Functions.EventHandler.InvokeAfterOriginal += new
EventHandlerFunction.InvokeEventHandler(VSTMEventHandler);
}
```

```
Sub Initialize() Implements IDexterityAddIn.Initialize
    ' Register Event to add menu entries
    AddHandler MenusForVisualStudioTools.Functions.EventRegister.InvokeAfterOriginal,
AddressOf VSTMEventRegister
    ' Register Event to handle callbacks from menu entries
    AddHandler MenusForVisualStudioTools.Functions.EventHandler.InvokeAfterOriginal,
AddressOf VSTMEventHandler
End Sub
```

In this sample the `VSTMEventRegister()` script would contain the function calls to add the menu items and record their returned menu tags. The `VSTMEventHandler()` script would then use the first parameter `e.inParam1` from `EventRegisterFunction.InvokeEventArgs e` argument object returned by the function call to obtain the menu tag of the menu item selected. This tag can then be compared to the stored menu tags obtained during registration and if a match is found, the desired actions can be executed.

Please see the VSTMenus.txt file or the Visual Basic and Visual C# examples included in the archive for more information.

Registering Area Page Events for Visual Studio Integration Toolkit

Build 15 or above introduces support for Area Pages, if you wish to use this functionality you need to add into the `Initialize()` function the commands to register for the two new function events used by the tool for Area Pages.

Register the following function event to call the script to register/create your Area Page:

`MenusForVisualStudioTools.Functions.EventRegisterAreaPage.InvokeAfterOriginal`

The event handler called from this event then contains the commands to create the Area Page button and command list.

Then, register the following function event to call the script to handle the callback when the Area Page is selected:

`MenusForVisualStudioTools.Functions.EventHandlerAreaPage.InvokeAfterOriginal`

The event handler called from this event then contains the commands to create the Area Page content.

Below are examples of the required event registrations for Visual C# and Visual Basic:

```
public void Initialize()
{
    // Register Event to add Area Page Button
    MenusForVisualStudioTools.Functions.EventRegisterAreaPage.InvokeAfterOriginal += new
EventRegisterAreaPageFunction.InvokeEventHandler(VSTMEventRegisterAreaPage);
    // Register Event to add Area Page Content
    MenusForVisualStudioTools.Functions.EventHandlerAreaPage.InvokeAfterOriginal += new
EventHandlerAreaPageFunction.InvokeEventHandler(VSTMEventHandlerAreaPage);
}
```

```
Sub Initialize() Implements IDexterityAddIn.Initialize
    ' Register Event to add Area Page Button
    AddHandler
MenusForVisualStudioTools.Functions.EventRegisterAreaPage.InvokeAfterOriginal, AddressOf
VSTMEventRegisterAreaPage
    ' Register Event to add Area Page Content
    AddHandler
MenusForVisualStudioTools.Functions.EventHandlerAreaPage.InvokeAfterOriginal, AddressOf
VSTMEventHandlerAreaPage
End Sub
```

In this sample the `VSTMEventRegisterAreaPage()` script would contain the function calls to add the Area Page button and record the returned tag. The `VSTMEventHandlerAreaPage()` script would then use the first parameter `e.inParam1` from `EventRegisterFunction.InvokeEventArgs e` argument object to obtain the tag of the Area Page selected. Then the call to add the Area Page content can be made.

Please see the VSTMenus.txt file or the Visual Basic and Visual C# examples included in the archive for more information.

Registering Form Events for Visual Studio Integration Toolkit

Build 18 or above introduces support for Custom Forms, if you wish to use this functionality you need to add into the `Initialize()` function the commands to register for the two new function events used by the tool for Custom Forms.

Register the following function event to call the script to register your Custom Forms:

`MenusForVisualStudioTools.Functions.EventRegisterForm.InvokeAfterOriginal`

The event handler called from this event then contains the commands to register your Developer ID and your Custom Form IDs and be assigned a Form Number for the Form Type for each form. The assigned form can then be modified with the Modifier to add additional fields and controls.

Then, register the following function event to call the script to handle the callback when Custom Form events are triggered:

`MenusForVisualStudioTools.Functions.EventHandlerForm.InvokeAfterOriginal`

The event handler called from this event then contains whatever functionality is desired for the various Form, Window, Field and Table events.

Below are examples of the required event registrations for Visual C# and Visual Basic:

```
public void Initialize()
{
    // Register Event to Register Custom Forms
    MenusForVisualStudioTools.Functions.EventRegisterForm.InvokeAfterOriginal += new
EventRegisterFormFunction.InvokeEventHandler(VSTMEventRegisterForm);
    // Register Event to handle Custom Forms Events
    MenusForVisualStudioTools.Functions.EventHandlerForm.InvokeAfterOriginal += new
EventHandlerFormFunction.InvokeEventHandler(VSTMEventHandlerForm);
}
```

```
Sub Initialize() Implements IDexterityAddIn.Initialize
    ' Register Event to Register Custom Forms
    AddHandler MenusForVisualStudioTools.Functions.EventRegisterForm.InvokeAfterOriginal,
AddressOf VSTMEventRegisterForm
    ' Register Event to handle Custom Forms Events
    AddHandler MenusForVisualStudioTools.Functions.EventHandlerForm.InvokeAfterOriginal,
AddressOf VSTMEventHandlerForm
End Sub
```

In this sample the `VSTMEventRegisterForm()` script would contain the function calls to register the Developer ID and Custom Forms and obtain the Form Number assigned. The `VSTMEventHandlerForm()` script would be called with the parameters for each event when a custom form is used.

Please see the VSTMenus.txt file or the Visual Basic and Visual C# examples included in the archive for more information.

Glossary of Terms

Below is a list terms used in this documentation and with Visual Studio Integration Toolkit for Microsoft Dynamics GP:

Dictionary ID:	Short integer uniquely identifying a Dexterity Dictionary, also known as a Product ID. The Dictionary ID for a product can be obtained from the Dynamics.set launch file.
Resource ID:	Short integer uniquely identifying an individual resource within a Dexterity Dictionary.
Form:	A container object that holds windows, scripts and commands.
Form Name:	The name of a form within a Dictionary.
Form Resource ID:	The Resource ID for a form within a Dictionary
Command:	A Dexterity resource used for creating Menus and Toolbars. Commands are contained on a form. To uniquely identify a command requires a
Menu Tag:	Short integer assigned to a command when it is added to the application menu structure.
Area Page:	Navigation page in the application main window containing multiple menus. Supported from Build 15 or above.

Limitations

Visual Studio Integration Toolkit for Microsoft Dynamics GP has the following limitations:

- There is a maximum of 100 menu items shared among all Visual Studio addins using API.
- There is a maximum of 10 area pages shared among all Visual Studio addins using API.
- There is a maximum of 10 custom forms for each form type shared among all Visual Studio addins using API.
- Menu items defined cannot be added to toolbars
- The menus added with this tool will not be visible to other Dexterity based applications.

Product Support

Technical support for Visual Studio Integration Toolkit for Microsoft Dynamics GP can be obtained by emailing support@winthropdc.com.

While all efforts will be made to respond in a timely manner, due to time zone differences a response may take up to two business days.

Menu structure in Microsoft Dynamics GP

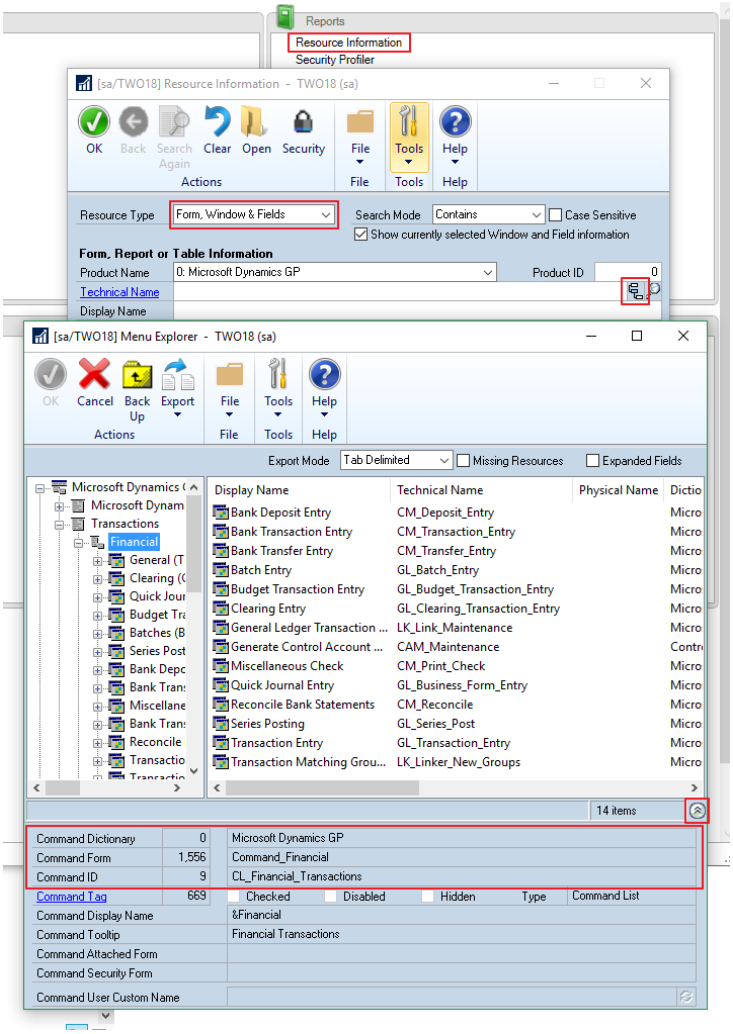
To add your menu items to the set of menus in Microsoft Dynamics GP, you need to know how the menus are structured. The following table shows the menu structure for the core application. These are the menus you see in the Area Pages, in the Microsoft Dynamics GP menu, and the "Main" toolbar. You will use this information when you add the menu items for your integration.

Top-level menu	Submenu	Form	Command
Tools*		Command_System	CL_Tools
Tools >> Setup		Command_System	CL_Setup
	System	Command_System	CL_System_Setup
	Company	Command_System	CL_Company_Setup
	Posting	Command_System	CL_Posting_Setup
	Financial	Command_Financial	CL_Financial_Setup
	Sales	Command_Sales	CL_Sales_Setup
	Purchasing	Command_Purchasing	CL_Purchasing_Setup
	Inventory	Command_Inventory	CL_Inventory_Setup
	Payroll	Command_Payroll	CL_Payroll_Setup
Tools >> Utilities		Command_System	CL_Utilities
	System	Command_System	CL_System_Utilities
	Company	Command_System	CL_Company_Utilities
	Financial	Command_Financial	CL_Financial_Utilities
	Sales	Command_Sales	CL_Sales_Utilities
	Purchasing	Command_Purchasing	CL_Purchasing_Utilities
	Inventory	Command_Inventory	CL_Inventory_Utilities
	Payroll	Command_Payroll	CL_Payroll_Utilities
Tools >> Routines		Command_System	CL_Routines
	Company	Command_System	CL_Company_Routines
	Financial	Command_Financial	CL_Financial_Routines
	Sales	Command_Sales	CL_Sales_Routines
	Purchasing	Command_Purchasing	CL_Purchasing_Routines
	Inventory	Command_Inventory	CL_Inventory_Routines
	Payroll	Command_Payroll	CL_Payroll_Routines
Transactions		Command_System	CL_Transactions
	Financial	Command_Financial	CL_Financial_Transactions
	Sales	Command_Sales	CL_Sales_Transactions
	Purchasing	Command_Purchasing	CL_Purchasing_Transactions
	Inventory	Command_Inventory	CL_Inventory_Transactions
	Payroll	Command_Payroll	CL_Payroll_Transactions
Inquiry		Command_System	CL_Inquiry
	System	Command_System	CL_System_Inquiry
	Financial	Command_Financial	CL_Financial_Inquiry
	Sales	Command_Sales	CL_Sales_Inquiry
	Purchasing	Command_Purchasing	CL_Purchasing_Inquiry
	Inventory	Command_Inventory	CL_Inventory_Inquiry
	Payroll	Command_Payroll	CL_Payroll_Inquiry

Top-level menu	Submenu	Form	Command

Reports		Command_System	CL_Reports
	System	Command_System	CL_System_Reports
	Company	Command_System	CL_Company_Reports
	Financial	Command_Financial	CL_Financial_Reports
	Sales	Command_Sales	CL_Sales_Reports
	Purchasing	Command_Purchasing	CL_Purchasing_Reports
	Inventory	Command_Inventory	CL_Inventory_Reports
	Payroll	Command_Payroll	CL_Payroll_Reports
Cards		Command_System	CL_Cards
	System	Command_System	CL_System_Cards
	Financial	Command_Financial	CL_Financial_Cards
	Sales	Command_Sales	CL_Sales_Cards
	Purchasing	Command_Purchasing	CL_Purchasing_Cards
	Inventory	Command_Inventory	CL_Inventory_Cards
	Payroll	Command_Payroll	CL_Payroll_Cards
Help*		Command_System	CL_Help

*We recommend not adding items directly to this top-level menu



For other Microsoft and third-party product dictionaries, you can find details of the menu structures and commands using Winthrop Development Consultants **GP Power Tools**.

From the GP Power Tools area page, select the Resource Information window (GP Power Tools >> Reports >> Resource Information).

Make sure the window has the Resource Type as Form, Window & Fields, then click on the Menu Lookup to open the Menu Explorer.

On the Menu Explorer window use the button at the bottom right corner of the window to expand the window.

Now you can see the command dictionary, form and ID for any menu item selected in the left-hand pane of the Menu Explorer.

See the highlighted sections on the screenshot.

Functions Available

Below is a list of the functions available to be called from the MenusForVisualStudioTools.Functions object.

Registration Functions:

Name: Register

Description: This function is used to create menu entries or parent menus entries for sub menus. Error codes are shown below.

Qualified Name: MenusForVisualStudioTools.Functions.Register

Returns: (Short) The Menu Tag of the registered menu or error code.

Parameters:

IN_Parent_Tag	(Short)	Menu Tag for Parent
IN_Caption	(String)	Menu Caption, cannot be empty
IN_Tooltip	(String)	Menu Tooltip (optional)
IN_Key	(Integer)	Menu Shortcut, ASCII value of key
IN_Modifier	(Integer)	Menu Shortcut Modifier (optional)
IN_Checked	(Boolean)	Set to True to create menu as checked
IN_Disabled	(Boolean)	Set to True to create menu as disabled
IN_Hidden	(Boolean)	Set to True to create menu as hidden
IN_Below_Tag	(Short)	Menu Tag for menu entry to add below
IN_Separator	(Boolean)	Set to True to add Separator above
IN_List	(Boolean)	Set to True to add Menu as Command List

Related: GetTag, GetTagByName, Unregister

Error Codes:
-1 : Attempting to register more than the supported number of menus.
-2 : No Parent Tag provided.
-3 : No Caption provided.
-4 : Parent Tag was provided, unable to identify Parent Command from Tag.
-17: Visual Studio Integration Toolkit is not registered.

Notes: Leave IN_Below_Tag as 0 for bottom of menu or set to -1 for top of menu. Constants for use with IN_Key and IN_Modifier shown after GetAccelerator SetAccelerator functions.

Name: RegisterWithSecurity

Description: This function is used to create menu entries with security inherited from an application form. Error codes are shown below.

Qualified Name: MenusForVisualStudioTools.Functions.RegisterWithSecurity

Returns: (Short) The Menu Tag of the registered menu or an error code.

Parameters:

IN_Parent_Tag	(Short)	Menu Tag for Parent
IN_Caption	(String)	Menu Caption, cannot be empty
IN_Tooltip	(String)	Menu Tooltip (optional)
IN_Key	(Integer)	Menu Shortcut, ASCII value of key
IN_Modifier	(Integer)	Menu Shortcut Modifier (optional)
IN_Checked	(Boolean)	Set to True to create menu as checked
IN_Disabled	(Boolean)	Set to True to create menu as disabled
IN_Hidden	(Boolean)	Set to True to create menu as hidden
IN_Below_Tag	(Short)	Menu Tag for menu entry to add below
IN_Separator	(Boolean)	Set to True to add Separator above
IN_List	(Boolean)	Set to True to add Menu as Command List
IN_SecurityDictID	(Short)	Dictionary ID for Security Form
IN_SecurityResID	(Short)	Resource ID for Security Form

Related: GetTag, GetTagByName, GetFormResID, Unregister

Error Codes: Validate TagError Codes
ValidateForm Error Codes
-1 : Attempting to register more than the supported number of menus.
-2 : No Parent Tag provided.
-3 : No Caption provided.
-4 : Parent Tag was provided, unable to identify Parent Command from Tag.
-5 : Security Form not applied due to table error
-17: Visual Studio Integration Toolkit is not registered.

Notes: Leave IN_Below_Tag as 0 for bottom of menu or set to -1 for top of menu.

Name: RegisterSimple

Description: This function provides a simplified registration syntax to add menu entries to the end of any existing menu in core Dynamics GP, Project Accounting, Field Service, Manufacturing and Human Resources.

The Command List Name for the parent menu can be any command list command from any of the command forms in the products listed above. The table provided earlier in the documentation includes the command lists for core Dynamics GP in the final column. Command lists for the other products can be identified using Dexterity from the following forms:

Project Accounting	Command_PA
Field Service	SVC_Command_* (6 forms)
Manufacturing	Command_MFG
Human Resources	Command_HR

The command form and product dictionary do not need to be specified and parent menu tag does not need to be identified first. The Error codes are shown below.

Qualified Name: MenusForVisualStudioTools.Functions.RegisterSimple

Returns: (Short) The Menu Tag of the registered menu or error code.

Parameters:

IN_Parent_Menu	(String)	Command List Name for Parent
IN_Caption	(String)	Menu Caption, cannot be empty

Related: GetTagSimple, Unregister

Error Codes: ValidateFormByName Error Codes
-1 : Attempting to register more than the supported number of menus.
-2 : No Parent Tag provided.
-3 : No Caption provided.
-4 : Parent Tag was provided, unable to identify Parent Command from Tag.
-13: Command Resource not found.
-17: Visual Studio Integration Toolkit is not registered.

Name:	RegisterAreaPage (build 15 or above)		
Description:	This function is used to create an Area Page Button and associated Command List. Error codes are shown below.		
Qualified Name:	MenusForVisualStudioTools.Functions.RegisterAreaPage		
Returns:	(Short) The Area Page Tag of the registered Area Page or error code.		
Parameters:	IN_Caption	(String)	Area Page Caption, cannot be empty
	IN_Tooltip	(String)	Area Page Tooltip (optional)
Related:	RegisterAreaPageContent		
Error Codes:	-3 : No Caption provided. -17: Visual Studio Integration Toolkit is not registered. -19 : Attempting to register more than the supported number of Area Pages.		
Notes:	Generic images are used for the buttons as the commands to change the images did not work for the Area Page buttons.		

Name: RegisterAreaPageContent (build 15 or above)

Description: This function is used to create menu entries or parent menus entries for sub menus. Error codes are shown below.

Qualified Name: MenusForVisualStudioTools.Functions.RegisterAreaPageContent

Returns: (Short) The Menu Tag of the registered menu or error code.

Parameters:

IN_Tag	(Short)	Area Page Button Tag
IN_Caption	(String)	Area Page Caption, cannot be empty
IN_Tag1	(Short)	Menu Tag for Transaction Menu (optional)
IN_Pos1	(Short)	Column for Transaction Menu (optional)
IN_Tag2	(Short)	Menu Tag for Inquiry Menu (optional)
IN_Pos2	(Short)	Column for Inquiry Menu (optional)
IN_Tag3	(Short)	Menu Tag for Reports Menu (optional)
IN_Pos3	(Short)	Column for Reports Menu (optional)
IN_Tag4	(Short)	Menu Tag for Cards Menu (optional)
IN_Pos4	(Short)	Column for Cards Menu (optional)
IN_Tag5	(Short)	Menu Tag for Setup Menu (optional)
IN_Pos5	(Short)	Column for Setup Menu (optional)
IN_Tag6	(Short)	Menu Tag for Routines Menu (optional)
IN_Pos6	(Short)	Column for Routines Menu (optional)
IN_Tag7	(Short)	Menu Tag for Utilities Menu (optional)
IN_Pos7	(Short)	Column for Utilities Menu (optional)

Related: RegisterAreaPage

Error Codes: -3 : No Caption provided.
-17: Visual Studio Integration Toolkit is not registered.

Notes: Tag left as 0 will be skipped and not added to the Area Page.
Column Numbers can be 1 or 2 for v11.0 and 1 to 3 for V12.0 onwards. Leave the Column Number as 0 for default column location.

Name: Unregister

Description: This function is used to unregister menu entries. This is only provided to allow menus to be removed due to an exception during the registration process. Ensure all child menu entries are removed before removing the parent menu entry.

This function will not return the unregistered menus to the pool of available menus entries. If you require the ability to dynamically control menus, use the Hide and Disable functions.

Qualified Name: MenusForVisualStudioTools.Functions.Unregister

Returns: (Short) Returns zero for no error or an error code

Parameters:			
	IN_Parent_Tag	(Short)	Menu Tag for Parent
	IN_Tag	(Short)	Menu Tag to link Security Form to

Related: Register, RegisterWithSecurity

Error Codes: ValidateTag Error Codes
-2 : No Parent Tag provided.
-4 : Parent Tag was provided, unable to identify Parent Command from Tag.
-16 : Unregistering Menu Tag failed.

Security Functions:

Name:	ApplySecurity		
Description:	This function is used to add/change security to an existing menu. Security is inherited from an application form. Error codes are shown below.		
Qualified Name:	MenusForVisualStudioTools.Functions.ApplySecurity		
Returns:	(Short) Returns zero for no error or an error code		
Parameters:	IN_Tag	(Short)	Menu Tag to link Security Form to
	IN_DictID	(Short)	Dictionary ID for Security Form
	IN_ResID	(Short)	Resource ID for Security Form
Related:	GetFormResID		
Error Codes:	ValidateTag Error Codes ValidateForm Error Codes -5 : Security Form not applied due to table error.		

Name:	RemoveSecurity		
Description:	This function is used to remove security from a menu. Error codes are shown below.		
Qualified Name:	MenusForVisualStudioTools.Functions.RemoveSecurity		
Returns:	(Short) Returns zero for no error or an error code		
Parameters:	IN_Tag	(Short)	Menu Tag to remove Security Form from
Related:	GetFormResID		
Error Codes:	ValidateTag Error Codes -5 : Security Form not applied due to table error.		

Name:	CheckSecurity		
Description:	This function is used to check the application level security access of a Form given its ID. Error codes are shown below.		
Qualified Name:	MenusForVisualStudioTools.Functions.CheckSecurity		
Returns:	(Short) Returns zero for no error or an error code		
Parameters:	IN_Dict	(Short)	Dictionary ID for Form
	IN_Form	(Short)	Form Resource ID for Form
	IN_Verbose	(Boolean)	Display Security Error when Access Denied
Related:	CheckSecurityByName, DisplaySecurityError		
Error Codes:	ValidateForm Error Codes 0 : OKAY – Access Granted. 1 : REJECT_RECORD – Access Denied. 2 : REJECT_SCRIPT – Form not loaded.		

Name:	CheckSecurityByName		
Description:	This function is used to check the application level security access of a Form given its Name. Error codes are shown below.		
Qualified Name:	MenusForVisualStudioTools.Functions.CheckSecurityByName		
Returns:	(Short) Returns zero for no error or an error code		
Parameters:	IN_DictID	(Short)	Dictionary ID for Form
	IN_FormName	(String)	Form Name for Form
	IN_Verbose	(Boolean)	Display Security Error when Access Denied
Related:	CheckSecurity, DisplaySecurityError		
Error Codes:	ValidateFormByName Error Codes 0 : OKAY – Access Granted. 1 : REJECT_RECORD – Access Denied. 2 : REJECT_SCRIPT – Form not loaded.		

Name:	DisplaySecurityError
Description:	This function is used to display the “You don't have security privileges to open this window” security error dialog.
Qualified Name:	MenusForVisualStudioTools.Functions.DisplaySecurityError
Returns:	(Short) Returns 1
Parameters:	None
Related:	CheckSecurity, CheckSecurityByName

Helper Functions:

Name:	GetFormResID		
Description:	This function is used to obtain the Resource ID of a form from a string containing the form's technical name. Error codes are shown below.		
Qualified Name:	MenusForVisualStudioTools.Functions.GetFormResID		
Returns:	(Short) Resource ID for Form or an error code		
Parameters:	IN_DictID	(Short)	Dictionary ID for Form
	IN_FormName	(String)	Form Name for Form
Related:	RegisterWithSecurity, ApplySecurity		
Error Codes:	ValidateFormByName Error Codes		

Name:	GetIDs		
Description:	This function is used to obtain the Dictionary ID, Form ID and Command ID for a given Menu Tag. Error codes are shown below.		
Qualified Name:	MenusForVisualStudioTools.Functions.GetIDs		
Returns:	(Short) Returns zero for no error or an error code		
Parameters:	IN_Tag	(Short)	Menu Tag to retrieve
	OUT_Dict	(Short)	Dictionary ID of Menu Entry
	OUT_Form	(Short)	Form Resource ID of Menu Entry
	OUT_Cmd	(Short)	Command Resource ID of Menu Entry
Related:	GetTag		
Error Codes:	ValidateTag Error Codes		

Name:	GetTag		
Description:	This function is used to obtain the Tag ID of a menu entry from the Dictionary ID, Form ID and Command ID. Error codes are shown below.		
Qualified Name:	MenusForVisualStudioTools.Functions.GetTag		
Returns:	(Short) Menu Tag of specified Menu Entry or an error code		
Parameters:	IN_Dict	(Short)	Dictionary ID of Menu Entry
	IN_Form	(Short)	Form Resource ID of Menu Entry
	IN_Cmd	(Short)	Command Resource ID of Menu Entry
Related:	GetIDs, GetTagByName, GetTagSimple		
Error Codes:	ValidateCmd Error Codes		

Name:	GetTagByName		
Description:	This function is used to obtain the Tag ID of a menu entry from the Dictionary ID, Form Name and Command Name. Error codes are shown below.		
Qualified Name:	MenusForVisualStudioTools.Functions.GetTagByName		
Returns:	(Short) Menu Tag of specified Menu Entry or an error code		
Parameters:	IN_Dict	(Short)	Dictionary ID of Menu Entry
	IN_Form	(String)	Form Name of Menu Entry
	IN_Cmd	(String)	Command Name of Menu Entry
Related:	GetTag, GetTagSimple, Register, RegisterWithSecurity		
Error Codes:	ValidateCmdByName Error Codes		

Name: GetTagSimple

Description: This function provides a simplified syntax to obtain the Tag ID of any existing menu in core Dynamics GP, Project Accounting, Field Service, Manufacturing and Human Resources.

The Command List Name for the parent menu can be any command list command from any of the command forms in the products listed above. The table provided earlier in the documentation includes the command lists for core Dynamics GP in the final column. Command lists for the other products can be identified using Dexterity from the following forms:

Project Accounting	Command_PA
Field Service	SVC_Command_* (6 forms)
Manufacturing	Command_MFG
Human Resources	Command_HR

The command form and product dictionary do not need to be specified and parent menu tag does not need to be identified first. The Error codes are shown below.

Qualified Name: MenusForVisualStudioTools.Functions.GetTagSimple

Returns: (Short) Menu Tag of specified Menu Entry or an error code

Parameters: IN_Parent_Menu (String) Command List Name for Parent

Related: GetTag, GetTagByName, Register, RegisterWithSecurity, RegisterSimple

Error Codes: ValidateCmdByName Error Codes
-13: Command Resource not found.

Callback Function:

Name:	Callback
Description:	This function is used to obtain the Tag ID of the just selected menu entry to be compared with the values returned during registration.
Qualified Name:	MenusForVisualStudioTools.Functions.Callback
Returns:	(Short) Menu Tag of just selected Menu Entry
Parameters:	None
Related:	Register, RegisterWithSecurity

Boolean Property Functions:

Name:	Checked		
Description:	This function is used to get the checked state of the specified Menu Entry. Error codes are shown below.		
Qualified Name:	MenusForVisualStudioTools.Functions.Checked		
Returns:	(Short) Returns 0 (False) or 1 (True) or an error code.		
Parameters:	IN_Tag	(Short)	Menu Tag of Menu Entry
Related:	Check		
Error Codes:	ValidateTag Error Codes		

Name:	Check		
Description:	This function is used to set the checked state of the specified Menu Entry. Error codes are shown below.		
Qualified Name:	MenusForVisualStudioTools.Functions.Check		
Returns:	(Short) Returns zero for no error or an error code		
Parameters:	IN_Tag IN_Checked	(Short) (Boolean)	Menu Tag of Menu Entry Set to True to check Menu Entry
Related:	Checked		
Error Codes:	ValidateTag Error Codes -15 : Menu Property change failed.		

Name:	Hidden		
Description:	This function is used to get the hidden state of the specified Menu Entry. Error codes are shown below.		
Qualified Name:	MenusForVisualStudioTools.Functions.Hidden		
Returns:	(Short) Returns 0 (False) or 1 (True) or an error code.		
Parameters:	IN_Tag	(Short)	Menu Tag of Menu Entry
Related:	Hide		
Error Codes:	ValidateTag Error Codes		

Name:	Hide		
Description:	This function is used to set the hidden state of the specified Menu Entry. Error codes are shown below.		
Qualified Name:	MenusForVisualStudioTools.Functions.Hide		
Returns:	(Short) Returns zero for no error or an error code		
Parameters:	IN_Tag IN_Hidden	(Short) (Boolean)	Menu Tag of Menu Entry Set to True to hide Menu Entry
Related:	Hidden		
Error Codes:	ValidateTag Error Codes -15 : Menu Property change failed.		

Name:	Disabled		
Description:	This function is used to get the disabled state of the specified Menu Entry. Error codes are shown below.		
Qualified Name:	MenusForVisualStudioTools.Functions.Disabled		
Returns:	(Short) Returns 0 (False) or 1 (True) or an error code.		
Parameters:	IN_Tag	(Short)	Menu Tag of Menu Entry
Related:	Disable		
Error Codes:	ValidateTag Error Codes		

Name:	Disable		
Description:	This function is used to set the disabled state of the specified Menu Entry. Error codes are shown below.		
Qualified Name:	MenusForVisualStudioTools.Functions.Disable		
Returns:	(Short) Returns zero for no error or an error code		
Parameters:	IN_Tag IN_Disabled	(Short) (Boolean)	Menu Tag of Menu Entry Set to True to disable Menu Entry
Related:	Disabled		
Error Codes:	ValidateTag Error Codes -15 : Menu Property change failed.		

String Property Functions:

Name:	GetCaption		
Description:	This function is used to get the caption value of the specified Menu Entry. Error codes are shown below.		
Qualified Name:	MenusForVisualStudioTools.Functions.GetCaption		
Returns:	(Short) Returns zero for no error or an error code		
Parameters:	IN_Tag	(Short)	Menu Tag of Menu Entry
	OUT_Caption	(String)	Current Name of Menu Entry
Related:	SetCaption		
Error Codes:	ValidateTag Error Codes		

Name:	SetCaption		
Description:	This function is used to set the caption value of the specified Menu Entry. Error codes are shown below.		
Qualified Name:	MenusForVisualStudioTools.Functions.SetCaption		
Returns:	(Short) Returns zero for no error or an error code		
Parameters:	IN_Tag	(Short)	Menu Tag of Menu Entry
	IN_Caption	(String)	Value to change Menu Name to
Related:	GetCaption		
Error Codes:	ValidateTag Error Codes -3 : No Caption provided. -15 : Menu Property change failed.		

Name:	GetTooltip		
Description:	This function is used to get the tooltip value of the specified Menu Entry. Error codes are shown below.		
Qualified Name:	MenusForVisualStudioTools.Functions.GetTooltip		
Returns:	(Short) Returns zero for no error or an error code		
Parameters:	IN_Tag	(Short)	Menu Tag of Menu Entry
	OUT_Tooltip	(String)	Current Tooltip of Menu Entry
Related:	SetTooltip		
Error Codes:	ValidateTag Error Codes		

Name:	SetTooltip		
Description:	This function is used to set the tooltip value of the specified Menu Entry. Error codes are shown below.		
Qualified Name:	MenusForVisualStudioTools.Functions.SetTooltip		
Returns:	(Short) Returns zero for no error or an error code		
Parameters:	IN_Tag	(Short)	Menu Tag of Menu Entry
	IN_Tooltip	(String)	Value to change Tooltip to
Related:	GetTooltip		
Error Codes:	ValidateTag Error Codes -15 : Menu Property change failed.		

Name:	GetNamedProperty (build 15 or above)		
Description:	This function is used to get the value of a Named Property for the specified Menu Entry. Error codes are shown below.		
Qualified Name:	MenusForVisualStudioTools.Functions.GetNamedProperty		
Returns:	(Short) Returns zero for no error or an error code		
Parameters:	IN_Tag	(Short)	Menu Tag of Menu Entry
	IN_Property	(String)	Name of Property for Menu Entry
	OUT_Value	(String)	Current Value of Menu Entry Property
Related:	SetNamedProperty		
Error Codes:	ValidateTag Error Codes -18 : No Property Name provided.		

Name:	SetNamedProperty (build 15 or above)		
Description:	This function is used to set the value of a Named Property for the specified Menu Entry. Error codes are shown below.		
Qualified Name:	MenusForVisualStudioTools.Functions.SetNamedProperty		
Returns:	(Short) Returns zero for no error or an error code		
Parameters:	IN_Tag	(Short)	Menu Tag of Menu Entry
	IN_Property	(String)	Name of Property for Menu Entry
	IN_Value	(String)	Value to change Menu Entry Property to
Related:	GetNamedProperty		
Error Codes:	ValidateTag Error Codes -15 : Menu Property change failed. -18 : No Property Name provided.		

Numeric Property Functions:

Name:	GetAccelerator (build 15 or above)		
Description:	This function is used to get the accelerator shortcut key of the specified Menu Entry. Error codes are shown below.		
Qualified Name:	MenusForVisualStudioTools.Functions.GetAccelerator		
Returns:	(Short) Returns zero for no error or an error code		
Parameters:	IN_Tag	(Short)	Menu Tag of Menu Entry
	OUT_Key	(Integer)	ASCII value of key or constant
	OUT_Modifier	(Integer)	Optional constant value
Related:	SetAccelerator		
Error Codes:	ValidateTag Error Codes		

Name:	SetAccelerator (build 15 or above)		
Description:	This function is used to set the accelerator shortcut key of the specified Menu Entry. Error codes are shown below.		
Qualified Name:	MenusForVisualStudioTools.Functions.SetAccelerator		
Returns:	(Short) Returns zero for no error or an error code		
Parameters:	IN_Tag	(Short)	Menu Tag of Menu Entry
	IN_Key	(Integer)	ASCII value of key or constant
	IN_Modifier	(Integer)	Optional constant value
Related:	GetAccelerator		
Error Codes:	ValidateTag Error Codes -15 : Menu Property change failed.		

Constants for use with Accelerator Shortcut Key for Visual C# and Visual Basic .Net:

```
// Shortcut Key Modifier Constants
const int COMMAND_SHORTCUT_CTRL = 65536;
const int COMMAND_SHORTCUT_CTRLSHIFT = 327680;
const int COMMAND_SHORTCUT_CTRLALT = 196608;
const int COMMAND_SHORTCUT_ALT = 131072;
const int COMMAND_SHORTCUT_ALTSHIFT = 393216;
const int COMMAND_SHORTCUT_CTRLALTSHIFT = 458752;

// Shortcut Key Function Key Constants, can be used instead of ASCII value
const short COMMAND_SHORTCUT_KEY_F1 = 112;
const short COMMAND_SHORTCUT_KEY_F2 = 113;
const short COMMAND_SHORTCUT_KEY_F3 = 114;
const short COMMAND_SHORTCUT_KEY_F4 = 115;
const short COMMAND_SHORTCUT_KEY_F5 = 116;
const short COMMAND_SHORTCUT_KEY_F6 = 117;
const short COMMAND_SHORTCUT_KEY_F7 = 118;
const short COMMAND_SHORTCUT_KEY_F8 = 119;
const short COMMAND_SHORTCUT_KEY_F9 = 120;
const short COMMAND_SHORTCUT_KEY_F10 = 121;
const short COMMAND_SHORTCUT_KEY_F11 = 122;
const short COMMAND_SHORTCUT_KEY_F12 = 123;
```

```
' Shortcut Key Modifier Constants
Const COMMAND_SHORTCUT_CTRL As Integer = 65536
Const COMMAND_SHORTCUT_CTRLSHIFT As Integer = 327680
Const COMMAND_SHORTCUT_CTRLALT As Integer = 196608
Const COMMAND_SHORTCUT_ALT As Integer = 131072
Const COMMAND_SHORTCUT_ALTSHIFT As Integer = 393216
Const COMMAND_SHORTCUT_CTRLALTSHIFT As Integer = 458752

' Shortcut Key Function Key Constants, can be used instead of ASCII value
Const COMMAND_SHORTCUT_KEY_F1 As Short = 112
Const COMMAND_SHORTCUT_KEY_F2 As Short = 113
Const COMMAND_SHORTCUT_KEY_F3 As Short = 114
Const COMMAND_SHORTCUT_KEY_F4 As Short = 115
Const COMMAND_SHORTCUT_KEY_F5 As Short = 116
Const COMMAND_SHORTCUT_KEY_F6 As Short = 117
Const COMMAND_SHORTCUT_KEY_F7 As Short = 118
Const COMMAND_SHORTCUT_KEY_F8 As Short = 119
Const COMMAND_SHORTCUT_KEY_F9 As Short = 120
Const COMMAND_SHORTCUT_KEY_F10 As Short = 121
Const COMMAND_SHORTCUT_KEY_F11 As Short = 122
Const COMMAND_SHORTCUT_KEY_F12 As Short = 123
```

Version Functions:

Name:	GetVersion
Description:	This function is used to obtain the version and build number for the tool.
Qualified Name:	MenusForVisualStudioTools.Functions.GetVersion
Returns:	(String) Version Number in form XX.00.XXXX
Parameters:	None
Related:	DisplayAbout

Name:	DisplayAbout
Description:	This function is used to open the About window.
Qualified Name:	MenusForVisualStudioTools.Functions.DisplayAbout
Returns:	(Boolean) Set True if the window opened
Parameters:	None
Related:	GetVersion

Validation Functions:

Name:	SetProtectedMode		
Description:	This function is used to turn off and on Protected Mode. Protected Mode is on by default which will prevent menu tags from other products being addressed. If you turn off Protected Mode to allow access to a menu from another product, ensure you turn it back on afterwards.		
Qualified Name:	MenusForVisualStudioTools.Functions.SetProtectedMode		
Returns:	(Boolean) Returns True		
Parameters:	IN_Mode	(Boolean)	Set to False to turn Protected Mode off
Related:	ValidateTag		

Name:	ValidateTag		
Description:	This function is used check if a Menu Tag is valid. If Protected Mode is on, a Menu Tag from another product will return an error code. Error codes are shown below.		
Qualified Name:	MenusForVisualStudioTools.Functions.ValidateTag		
Returns:	(Short) Returns zero for no error or an error code		
Parameters:	IN_Tag	(Short)	Menu Tag of Menu Entry
Related:	SetProtectedMode		
Error codes:	-6 : No Menu Tag provided. -7 : Menu Tag not found. -8 : Menu Tag found, but from another product and Protected Mode is on. -9 : Product Dictionary not loaded. -10 : No Form Name or Form ID provided. -11 : Form Resource not found. -12 : No Command Name or Command ID provided. -14 : Command Resource not loaded in memory.		

Name:	ValidateCmd		
Description:	This function is used check if a Command on a form is valid given its ID. Error codes are shown below.		
Qualified Name:	MenusForVisualStudioTools.Functions.ValidateCmd		
Returns:	(Short) Returns zero for no error or an error code		
Parameters:	IN_Dict	(Short)	Dictionary ID of Menu Entry
	IN_Form	(Short)	Form Resource ID of Menu Entry
	IN_Cmd	(Short)	Command Resource ID of Menu Entry
Related:	ValidateCmdByName		
Error codes:	-9 : Product Dictionary not loaded. -10 : No Form Name or Form ID provided. -11 : Form Resource not found. -12 : No Command Name or Command ID provided. -13 : Command Resource not found. -14 : Command Resource not loaded in memory.		

Name:	ValidateCmdByName		
Description:	This function is used check if a Command on a form is valid given its name. Error codes are shown below.		
Qualified Name:	MenusForVisualStudioTools.Functions.ValidateCmdByName		
Returns:	(Short) Returns zero for no error or an error code		
Parameters:	IN_Dict	(Short)	Dictionary ID of Menu Entry
	IN_Form	(String)	Form Name of Menu Entry
	IN_Cmd	(String)	Command Name of Menu Entry
Related:	ValidateCmd		
Error codes:	-9 : Product Dictionary not loaded. -10 : No Form Name or Form ID provided. -11 : Form Resource not found. -12 : No Command Name or Command ID provided. -13 : Command Resource not found. -14 : Command Resource not loaded in memory.		

Name:	ValidateForm		
Description:	This function is used check if a Form is valid given its ID. Error codes are shown below.		
Qualified Name:	MenusForVisualStudioTools.Functions.ValidateForm		
Returns:	(Short) Returns zero for no error or an error code		
Parameters:	IN_Dict	(Short)	Dictionary ID for Form
	IN_Form	(Short)	Form Resource ID for Form
Related:	ValidateFormByName		
Error codes:	-9 : Product Dictionary not loaded. -10 : No Form Name or Form ID provided. -11 : Form Resource not found.		

Name:	ValidateFormByName		
Description:	This function is used check if a Form is valid given its Name. Error codes are shown below.		
Qualified Name:	MenusForVisualStudioTools.Functions.ValidateFormByName		
Returns:	(Short) Returns zero for no error or an error code		
Parameters:	IN_DictID	(Short)	Dictionary ID for Form
	IN_FormName	(String)	Form Name for Form
Related:	ValidateForm		
Error codes:	-9 : Product Dictionary not loaded. -10 : No Form Name or Form ID provided. -11 : Form Resource not found.		

Error Code Summary:

This is a list of the standard error codes returned by the functions.

-1	Attempting to register more than the supported number of menus.
-2	No Parent Tag provided.
-3	No Caption provided.
-4	Parent Tag was provided, unable to identify Parent Command from Tag.
-5	Security Form not applied due to table error.
-6	No Menu Tag provided.
-7	Menu Tag not found.
-8	Menu Tag found, but from another product and Protected Mode is on.
-9	Product Dictionary not loaded.
-10	No Form Name or Form ID provided.
-11	Form Resource not found.
-12	No Command Name or Command ID provided.
-13	Command Resource not found.
-14	Command Resource not loaded in memory.
-15	Menu Property change failed.
-16	Unregistering Menu Tag failed.
-17	Visual Studio Integration Toolkit is not registered.
-18	No Property Name provided.
-19	Attempting to register more than the supported number of Area Pages.

Error codes for Custom Forms module:

-20	The Developer ID has not been registered for use.
-21	The Developer ID is still in use by forms and cannot be deleted.
-22	There is data missing from the function call parameters.
-23	There are no more forms of the request Form Type available, or form not open.
-24	The Form ID has not been registered for use.
-25	The Form Type is not the correct type for the requested action.
-26	An internal Compiler Error has occurred, please report to support@winthropdc.com.
-27	Visual Studio Integration Toolkit does not have its tables installed correctly.
-28	The Form ID has data associated with it and can only be deleted via the user interface.

Custom Forms Module

Build 18 of Visual Studio Integration Toolkit adds the Custom Forms Module. There are 10 of each type of form available. The forms range from blank forms to dialog forms to complex forms with much of the standard Microsoft Dynamics GP functionality already built in.

The available form types are:

1 = Blank Form (CF_TYPE_BLANK, 0)
2 = Dialog Form (CF_TYPE_DIALOG, 0)
3 = Setup Form (linked to Setup Table) (CF_TYPE_SETUP, 0)
4 = Maintenance Form (Linked to Header Table). Includes Report and Sub Forms:
    0 = Maintenance Form (CF_TYPE_MAINT, 0)
    1 = Lookup Form (CF_TYPE_MAINT, 1)
    2 = Inquiry Form (CF_TYPE_MAINT, 2)
5 = Maintenance Form with Lines (Linked to Header and Line Tables). Includes Report and Sub Forms:
    0 = Maintenance Form (CF_TYPE_MAINT_LINES, 0)
    1 = Lookup Form (CF_TYPE_MAINT_LINES, 1)
    2 = Inquiry Form (CF_TYPE_MAINT_LINES, 2)

The Setup and Maintenance forms are linked with tables. You can create your own tables to store any additional data desired using the same primary key fields.

Using the event registered against the EventRegisterForm() function, you can then execute the DeveloperRegister() function to register your Developer ID (a user definable string of up to 15 characters and is usually the initials of the company). Then you can execute the FormRegister() function to register the Form IDs (user definable string of up to 25 characters) for each of the forms you would like to use. When registering a Form, you will be provided with the Form Number assigned for the selected form type, this can be used to identify the actual Dexterity form reserved for your use. You can then use Modifier to change the appearance of the window and add any extra fields you need. You can use events to handle the CRUD (Create, Read, Update and Delete) actions for your own table data.

Using the event registered against the EventHandlerForm() function, you can pick up all the events triggered from the forms and take actions based on the parameters provided.

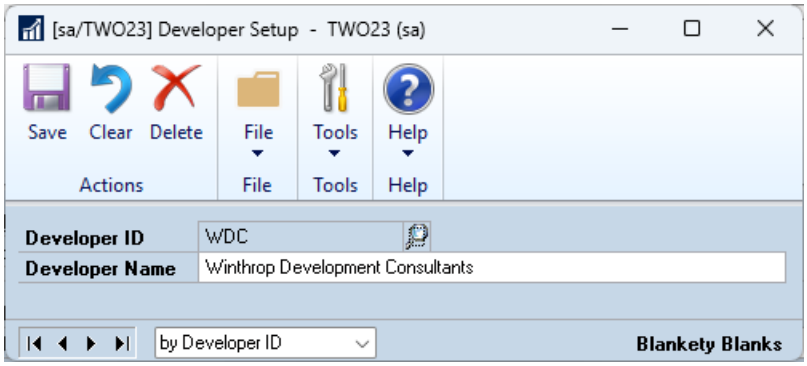
Parameters:	IN_DeveloperID	(String)	Developer ID when registered
	IN_FormID	(String)	Form ID when registered
	IN_SubForm	(Short)	Sub Form Number (0 when no sub forms)
	IN_Mode	(Short)	Event Mode (see table below)
	IN_FieldID	(String)	Name or Caption of field or table key value
	IN_Event	(Short)	Event Type (see table below)

See Sample code in the VSTMenus.txt file or the Visual Basic and Visual C# examples for recommended approach to handling the parameters passed in.

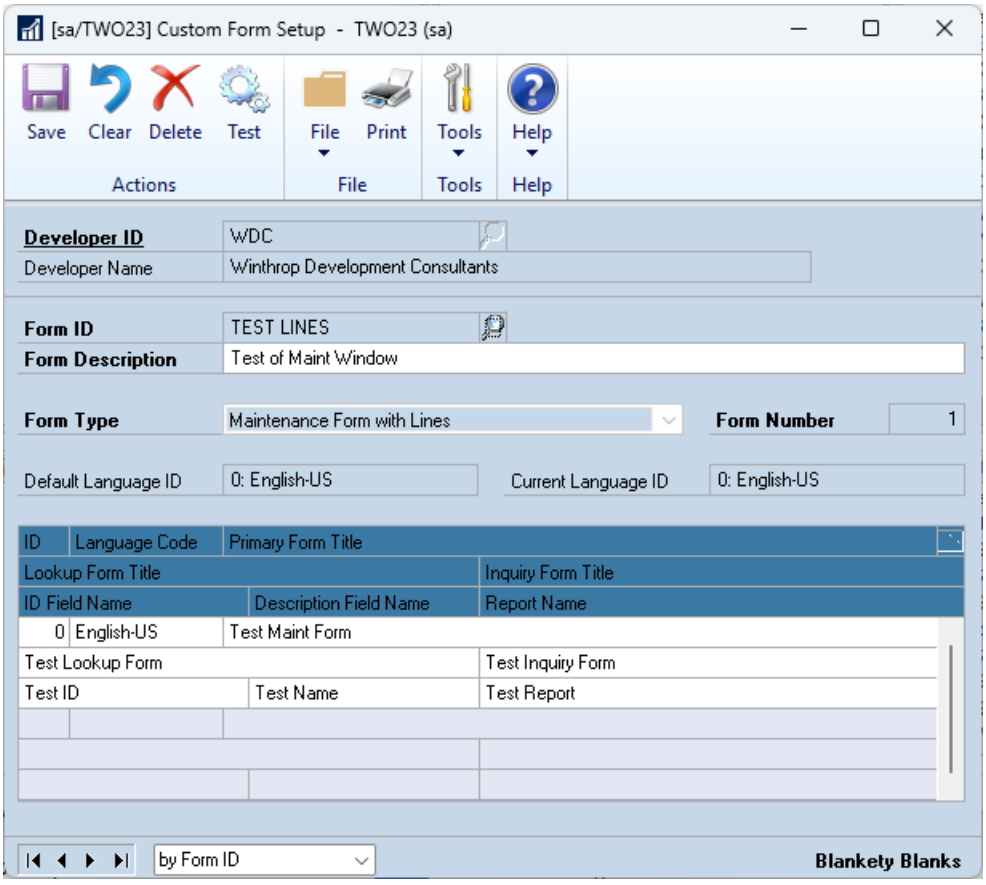
You can also use the windows under Administration >> System >> Cards to look at Developer IDs and Form IDs registered as well as test the Custom Forms.

Note: It is recommended to use code to register your Developer ID and Form IDs, but for testing purposes you can use these windows.

The Developer Setup window is used to maintain and view Developer IDs.

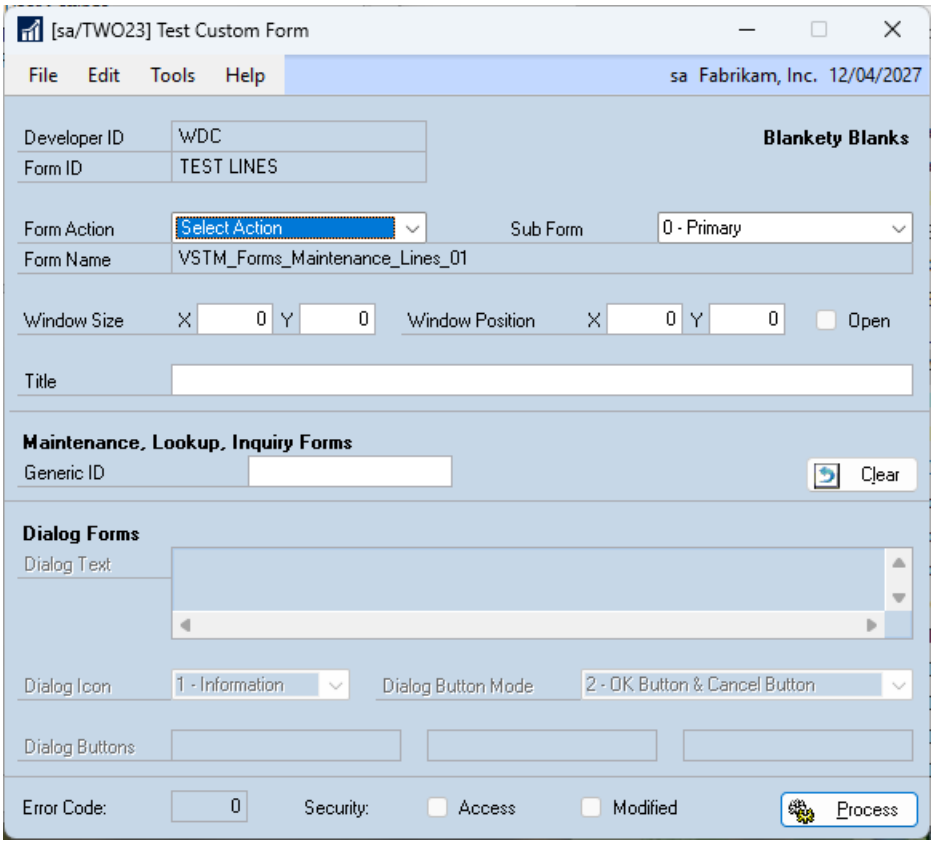


The Custom Form Setup window is used to maintain, view and test Form IDs



Note: Terminology for Form Names, Field Names and Report Names support multiple languages using the Language IDs in Microsoft Dynamics GP.

Clicking the Test button from the Custom Form Setup window when a Form ID is displayed will open the Test Custom Form window.



This window can be used to test the different Form Types with most of the functions available. Make sure to Open the form before using any “Get” or “Set” functions.

Note: It is recommended to execute the “Get” functions to retrieve the current values before using the “Set” Functions to change them.

The testing window also displays the Technical Name of the form in the Visual Studio Integration Toolkit dictionary (Product ID: 5986) that is used to provide the custom form.

Details of the functions available are listed in the section below.

Events Modes and Event Types for each Mode

1 = Form
- 0 = TRIGGER_FOCUS_PRE
- 2 = TRIGGER_FOCUS_POST

2 = Window
- 0 = TRIGGER_FOCUS_PRE
- 2 = TRIGGER_FOCUS_POST — use with FormAbortClose(), if needed
- 3 = TRIGGER_FOCUS_PRINT
- 4 = TRIGGER_FOCUS_ACTIVATE

3 = Scrolling Window
- 0 = TRIGGER_FOCUS_PRE
- 1 = TRIGGER_FOCUS_CHANGE — use with FormRejectScript(), if needed
- 2 = TRIGGER_FOCUS_POST
- 5 = TRIGGER_FOCUS_FILL — use with FormRejectRecord(), if needed
- 6 = TRIGGER_FOCUS_INSERT
- 7 = TRIGGER_FOCUS_DELETE

4 = Field
- 0 = TRIGGER_FOCUS_PRE
- 1 = TRIGGER_FOCUS_CHANGE — use with FormRejectScript(), if needed
- 2 = TRIGGER_FOCUS_POST

5 = Scrolling Field
- 0 = TRIGGER_FOCUS_PRE
- 1 = TRIGGER_FOCUS_CHANGE — use with FormRejectScript(), if needed
- 2 = TRIGGER_FOCUS_POST

6 = Table — Field Parameter will contain ID value for Header Table
- 4 = TRIGGER_ON_DB_ADD
- 5 = TRIGGER_ON_DB_UPDATE
- 16 = TRIGGER_ON_DB_DELETE

7 = Scroll Table — Field Parameter will contain ID & Sequence for Line Table
- 4 = TRIGGER_ON_DB_ADD — use with KeySplitLine() to split ID & Sequence
- 5 = TRIGGER_ON_DB_UPDATE — use with KeySplitLine() to split ID & Sequence
- 16 = TRIGGER_ON_DB_DELETE — use with KeySplitLine() to split ID & Sequence

8 = Report
- 0 = rw_ReportStart() — use with ReportSetString()
- 1 = rw_ReportEnd() — use with ReportSetString()
- 2 = rw_TableHeaderString() — use with ReportGetTableHeader() & ReportSetString()
- 3 = rw_TableHeaderCurrency() — use with ReportGetTableHeader()) & ReportSetCurrency()
- 4 = rw_TableLineString() — use with ReportGetTableLine() & ReportSetString()
- 5 = rw_TableLineCurrency() — use with ReportGetTableLine() & ReportSetCurrency()

Custom Forms Functions Available

Below is a list of the functions available to be called from the MenusForVisualStudioTools.Forms.VstmForms.Functions object.

Registration Functions:

Name:	DeveloperRegister		
Description:	This function is used to register a Developer ID for use with Custom Forms. Error codes are shown below.		
Qualified Name:	MenusForVisualStudioTools.Forms.VstmForms.Functions.DeveloperRegister		
Returns:	(Short) The error code.		
Parameters:	IN_DeveloperID	(String)	Developer ID to register
	IN_DeveloperName	(String)	Name for Developer ID
Related:	DeveloperUnregister		
Error Codes:	-17: Visual Studio Integration Toolkit is not registered. -20: The Developer ID has not been registered for use. -22: There is data missing from the function call parameters. -27: Visual Studio Integration Toolkit does not have its tables installed correctly.		
Notes:	A Developer ID must be registered before Form IDs can be registered.		

Name:	DeveloperUnregister		
Description:	This function is used to remove a Developer ID. Error codes are shown below.		
Qualified Name:	MenusForVisualStudioTools.Forms.VstmForms.Functions.DeveloperUnregister		
Returns:	(Short) The error code.		
Parameters:	IN_DeveloperID	(String)	Developer ID to unregister
Related:	DeveloperRegister		
Error Codes:	-21: The Developer ID is still in use by forms and cannot be deleted. -22: There is data missing from the function call parameters. -27: Visual Studio Integration Toolkit does not have its tables installed correctly.		
Notes:	A Developer ID cannot be removed if there are still Form IDs using it.		

Name: FormRegister

Description: This function is used to register a Form ID for the desired form type. Error codes are shown below.

Qualified Name: MenusForVisualStudioTools.Forms.VstmForms.Functions.FormRegister

Returns: (Short) The error code.

Parameters:

IN_DeveloperID	(String)	Developer ID to use
IN_FormID	(String)	Form ID to register
IN_FormDescription	(String)	Description for Form
IN_FormType	(Short)	Form Type to register
IN_FormName	(String)	Name to use for Window Title
OUT_FormNumber	(Short)	Form Number assigned to Form ID

Related: FormRegisterMaint, FormUnregister

Error Codes:
-17: Visual Studio Integration Toolkit is not registered.
-20: The Developer ID has not been registered for use.
-22: There is data missing from the function call parameters.
-23: There are no more forms of the request Form Type available.
-24: The Form ID has not been registered for use.
-25: The Form Type is not the correct type for the requested action.
-27: Visual Studio Integration Toolkit does not have its tables installed correctly.

Notes: A Developer ID must be registered before Form IDs can be registered.
Use FormRegisterMaint for Maintenance Forms and Maintenance Forms with Lines.

Name: FormRegisterMaint

Description: This function is used to register a Maintenance Form ID for the desired Maintenance Form or Maintenance Form With Lines form type. Error codes are shown below.

Qualified Name: MenusForVisualStudioTools.Forms.VstmForms.Functions.FormRegisterMaint

Returns: (Short) The error code.

Parameters:

IN_DeveloperID	(String)	Developer ID to use
IN_FormID	(String)	Form ID to register
IN_FormDescription	(String)	Description for Form
IN_FormType	(Short)	Form Type to register
IN_FormName	(String)	Name to use for Maintenance Window Title
IN_FormLookupName	(String)	Name to use for Lookup Window Title
N_FormInquiryName	(String)	Name to use for Inquiry Window Title
IN_FieldIDName	(String)	Name to use for ID Field
IN_FieldDescName	(String)	Name to use for Description Field
IN_ReportName	(String)	Name to use for Report Name
OUT_FormNumber	(Short)	Form Number assigned to Form ID

Related: FormRegister, FormUnregister

Error Codes:
-17: Visual Studio Integration Toolkit is not registered.
-20: The Developer ID has not been registered for use.
-22: There is data missing from the function call parameters.
-23: There are no more forms of the request Form Type available.
-24: The Form ID has not been registered for use.
-25: The Form Type is not the correct type for the requested action.
-27: Visual Studio Integration Toolkit does not have its tables installed correctly.

Notes: A Developer ID must be registered before Form IDs can be registered.
Use FormRegister for non-Maintenance Forms.

Name: FormRegisterReserve

Description: This function is used to register a Form ID for the desired form type. Providing a Form Number to the function will attempt to register that form number if it is available. This allows the number of the previously modified forms and reports to be requested. Error codes are shown below.

Qualified Name: MenusForVisualStudioTools.Forms.VstmForms.Functions.FormRegister

Returns: (Short) The error code.

Parameters:

IN_DeveloperID	(String)	Developer ID to use
IN_FormID	(String)	Form ID to register
IN_FormDescription	(String)	Description for Form
IN_FormType	(Short)	Form Type to register
IN_FormName	(String)	Name to use for Window Title
IN_FormNumber	(Short)	Form Number requested, if available
OUT_FormNumber	(Short)	Form Number assigned to Form ID

Related: FormRegisterMaint, FormUnregister

Error Codes:
-17: Visual Studio Integration Toolkit is not registered.
-20: The Developer ID has not been registered for use.
-22: There is data missing from the function call parameters.
-23: There are no more forms of the request Form Type available.
-24: The Form ID has not been registered for use.
-25: The Form Type is not the correct type for the requested action.
-27: Visual Studio Integration Toolkit does not have its tables installed correctly.

Notes: A Developer ID must be registered before Form IDs can be registered.
Use FormRegisterMaint for Maintenance Forms and Maintenance Forms with Lines.

Name: FormRegisterReserveMaint

Description: This function is used to register a Maintenance Form ID for the desired Maintenance Form or Maintenance Form With Lines form type. Providing a Form Number to the function will attempt to register that form number if it is available. This allows the number of the previously modified forms and reports to be requested. Error codes are shown below.

Qualified Name: MenusForVisualStudioTools.Forms.VstmForms.Functions.FormRegisterMaint

Returns: (Short) The error code.

Parameters:

IN_DeveloperID	(String)	Developer ID to use
IN_FormID	(String)	Form ID to register
IN_FormDescription	(String)	Description for Form
IN_FormType	(Short)	Form Type to register
IN_FormName	(String)	Name to use for Maintenance Window Title
IN_FormLookupName	(String)	Name to use for Lookup Window Title
N_FormInquiryName	(String)	Name to use for Inquiry Window Title
IN_FieldIDName	(String)	Name to use for ID Field
IN_FieldDescName	(String)	Name to use for Description Field
IN_ReportName	(String)	Name to use for Report Name
IN_FormNumber	(Short)	Form Number requested, if available
OUT_FormNumber	(Short)	Form Number assigned to Form ID

Related: FormRegister, FormUnregister

Error Codes:
-17: Visual Studio Integration Toolkit is not registered.
-20: The Developer ID has not been registered for use.
-22: There is data missing from the function call parameters.
-23: There are no more forms of the request Form Type available.
-24: The Form ID has not been registered for use.
-25: The Form Type is not the correct type for the requested action.
-27: Visual Studio Integration Toolkit does not have its tables installed correctly.

Notes: A Developer ID must be registered before Form IDs can be registered.
Use FormRegister for non-Maintenance Forms.

Name:	FormUnregister		
Description:	This function is used to remove a Form ID. Error codes are shown below.		
Qualified Name:	MenusForVisualStudioTools.Forms.VstmForms.Functions.FormUnregister		
Returns:	(Short) The error code.		
Parameters:	IN_DeveloperID	(String)	Developer ID to use
	IN_FormID	(String)	Form ID to unregister
Related:	FormRegister, FormRegisterMaint		
Error Codes:	-21: The Developer ID is still in use by forms and cannot be deleted. -22: There is data missing from the function call parameters. -27: Visual Studio Integration Toolkit does not have its tables installed correctly. -28: The Form ID has data associated with it and can only be deleted via the user interface.		
Notes:	A Form ID cannot be removed via the API if there data stored against it (for Setup or Maintenance Forms).		

Opening and Closing Form Functions:

Name:	FormClose		
Description:	This function is used to close a form that is currently open. Error codes are shown below.		
Qualified Name:	MenusForVisualStudioTools.Forms.VstmForms.Functions.FormClose		
Returns:	(Short) The error code.		
Parameters:	IN_DeveloperID	(String)	Developer ID to use
	IN_FormID	(String)	Form ID to open
	IN_SubForm	(Short)	Sub Form to open, leave as 0 for primary
Related:	FormOpen, FormOpenDialog, FormOpenInquiry, FormOpenLookup, FormOpenLookupReturn, FormOpenMaint, FormOpenReport, FormOpenReportDestination, FormIsOpen		
Error Codes:	-17: Visual Studio Integration Toolkit is not registered. -22: There is data missing from the function call parameters. -24: The Form ID has not been registered for use. -26: An internal Compiler Error has occurred, please report to support. -27: Visual Studio Integration Toolkit does not have its tables installed correctly.		

Name: FormOpen

Description: This function is used to open a form that has already been registered. There are also specific functions for different form types which have additional functionality. Error codes are shown below.

Qualified Name: MenusForVisualStudioTools.Forms.VstmForms.Functions.FormOpen

Returns: (Short) The error code.

Parameters:

IN_DeveloperID	(String)	Developer ID to use
IN_FormID	(String)	Form ID to open
IN_SubForm	(Short)	Sub Form to open, leave as 0 for primary

Related: FormClose, FormOpenDialog, FormOpenInquiry, FormOpenLookup, FormOpenLookupReturn, FormOpenMaint, FormOpenReport, FormOpenReportDestination, FormIsOpen

Error Codes:
-17: Visual Studio Integration Toolkit is not registered.
-22: There is data missing from the function call parameters.
-24: The Form ID has not been registered for use.
-26: An internal Compiler Error has occurred, please report to support.
-27: Visual Studio Integration Toolkit does not have its tables installed correctly.

Name: FormOpenDialog

Description: This function is used to open a Dialog type form that has already been registered. Error codes are shown below.

Qualified Name: MenusForVisualStudioTools.Forms.VstmForms.Functions.FormOpenDialog

Returns: (Short) The error code.

Parameters:

IN_DeveloperID	(String)	Developer ID to use
IN_FormID	(String)	Form ID to open
IN_Dialog_Icon	(Short)	Icon 1=Information, 2=Warning, 3=Error
IN_Dialog_Text	(String)	Text to display in the dialog
IN_Dialog_Button	(Short)	Mode, 1=OK, 2=OK & Cancel, 3=Custom
IN_Dialog_Button1	(String)	Button 1 Caption for Custom Button Mode
IN_Dialog_Button2	(String)	Button 2 Caption for Custom Button Mode
IN_Dialog_Button3	(String)	Button 3 Caption for Custom Button Mode

Related: FormClose, FormOpen, FormOpenInquiry, FormOpenLookup, FormOpenLookupReturn, FormOpenMaint, FormOpenReport, FormOpenReportDestination, FormIsOpen

Error Codes:
-17: Visual Studio Integration Toolkit is not registered.
-22: There is data missing from the function call parameters.
-24: The Form ID has not been registered for use.
-25: The Form Type is not the correct type for the requested action.
-26: An internal Compiler Error has occurred, please report to support.
-27: Visual Studio Integration Toolkit does not have its tables installed correctly.

Notes: IN_Dialog_Text can contain up to 32,000 characters including carriage return characters (ASCII 13) 0x0D to start new lines.

Name: FormOpenInquiry

Description: This function is used to open an Inquiry Sub Form of Maintenance type form that has already been registered. Error codes are shown below.

Qualified Name: MenusForVisualStudioTools.Forms.VstmForms.Functions.FormOpenInquiry

Returns: (Short) The error code.

Parameters:

IN_DeveloperID	(String)	Developer ID to use
IN_FormID	(String)	Form ID to open
IN_Sort	(Short)	Sort by Index 1=ID Field, 2=Description
IN_ID	(String)	Optional value of ID Field to open to

Related: FormClose, FormOpen, FormOpenDialog, FormOpenLookup, FormOpenLookupReturn, FormOpenMaint, FormOpenReport, FormOpenReportDestination, FormIsOpen

Error Codes:
-17: Visual Studio Integration Toolkit is not registered.
-22: There is data missing from the function call parameters.
-24: The Form ID has not been registered for use.
-25: The Form Type is not the correct type for the requested action.
-26: An internal Compiler Error has occurred, please report to support.
-27: Visual Studio Integration Toolkit does not have its tables installed correctly.

Name: FormOpenLookup

Description: This function is used to open a Lookup Sub Form of Maintenance type form that has already been registered. Error codes are shown below.

Qualified Name: MenusForVisualStudioTools.Forms.VstmForms.Functions.FormOpenLookup

Returns: (Short) The error code.

Parameters:

IN_DeveloperID	(String)	Developer ID to use
IN_FormID	(String)	Form ID to open
IN_Sort	(Short)	Sort by Index 1=ID Field, 2=Description
IN_ID	(String)	Optional value of ID Field to open to
IN_Name	(String)	Optional value for Description to open to
OUT_Return	(String Field)	Field to return Lookup data to

Related: FormClose, FormOpen, FormOpenDialog, FormOpenInquiry, FormOpenLookupReturn, FormOpenMaint, FormOpenReport, FormOpenReportDestination, FormIsOpen

Error Codes:
-17: Visual Studio Integration Toolkit is not registered.
-22: There is data missing from the function call parameters.
-24: The Form ID has not been registered for use.
-25: The Form Type is not the correct type for the requested action.
-26: An internal Compiler Error has occurred, please report to support.
-27: Visual Studio Integration Toolkit does not have its tables installed correctly.

Notes: This function is only available for use from Dexterity or GP Power Tools as it uses an anonymous field parameter not supported by Visual Studio.

Name: FormOpenLookupReturn

Description: This function is used to open a Lookup Sub Form of Maintenance type form that has already been registered. Error codes are shown below.

Qualified Name: MenusForVisualStudioTools.Forms.VstmForms.Functions.FormOpenLookupReturn

Returns: (Short) The error code.

Parameters:

IN_DeveloperID	(String)	Developer ID to use
IN_FormID	(String)	Form ID to open
IN_Sort	(Short)	Sort by Index 1=ID Field, 2=Description
IN_ID	(String)	Optional value of ID Field to open to
IN_Name	(String)	Optional value for Description to open to
IN_Dict	(Short)	Dictionary Product ID
IN_Form	(String)	Form Name
IN_Window	(String)	Window Name
IN_Field	(String)	Field Name
IN_Modified	(Bool)	Modifier Added Field flag

Related: FormClose, FormOpen, FormOpenDialog, FormOpenInquiry, FormOpenLookup, FormOpenMaint, FormOpenReport, FormOpenReportDestination, FormIsOpen

Error Codes:
-17: Visual Studio Integration Toolkit is not registered.
-22: There is data missing from the function call parameters.
-24: The Form ID has not been registered for use.
-25: The Form Type is not the correct type for the requested action.
-26: An internal Compiler Error has occurred, please report to support.
-27: Visual Studio Integration Toolkit does not have its tables installed correctly.

Notes: The Field to return data to is passed in as values for Dictionary ID, Form Name, Window Name and Field Name as well as a flag to indicate if the field is a Modifier Added Field. This method will work for Dexterity and for Visual Studio Tools.

Name: FormOpenMaint

Description: This function is used to open a Maintenance Sub Form of Maintenance type form that has already been registered. Error codes are shown below.

Qualified Name: MenusForVisualStudioTools.Forms.VstmForms.Functions.FormOpenMaint

Returns: (Short) The error code.

Parameters:

IN_DeveloperID	(String)	Developer ID to use
IN_FormID	(String)	Form ID to open
IN_Sort	(Short)	Sort by Index 1=ID Field, 2=Description
IN_ID	(String)	Optional value of ID Field to open to

Related: FormClose, FormOpen, FormOpenDialog, FormOpenInquiry, FormOpenLookup, FormOpenLookupReturn, FormOpenReport, FormOpenReportDestination, FormIsOpen

Error Codes:
-17: Visual Studio Integration Toolkit is not registered.
-22: There is data missing from the function call parameters.
-24: The Form ID has not been registered for use.
-25: The Form Type is not the correct type for the requested action.
-26: An internal Compiler Error has occurred, please report to support.
-27: Visual Studio Integration Toolkit does not have its tables installed correctly.

Name: FormOpenReport

Description: This function is used to open a Report for a Maintenance type form that has already been registered. Error codes are shown below.

Qualified Name: MenusForVisualStudioTools.Forms.VstmForms.Functions.FormOpenReport

Returns: (Short) The error code.

Parameters:

IN_DeveloperID	(String)	Developer ID to use
IN_FormID	(String)	Form ID to open
IN_ID	(String)	Optional value of ID Field to print report for

Related: FormClose, FormOpen, FormOpenDialog, FormOpenInquiry, FormOpenLookup, FormOpenLookupReturn, FormOpenMaint, FormOpenReportDestination, FormIsOpen

Error Codes:
-17: Visual Studio Integration Toolkit is not registered.
-22: There is data missing from the function call parameters.
-24: The Form ID has not been registered for use.
-25: The Form Type is not the correct type for the requested action.
-26: An internal Compiler Error has occurred, please report to support.
-27: Visual Studio Integration Toolkit does not have its tables installed correctly.

Name: FormOpenReportDestination

Description: This function is used to open a Report for a Maintenance type form that has already been registered. Error codes are shown below.

Qualified Name: MenusForVisualStudioTools.Forms.VstmForms.Functions.FormOpenReportDestination

Returns: (Short) The error code.

Parameters:

IN_DeveloperID	(String)	Developer ID to use
IN_FormID	(String)	Form ID to open
IN_ID	(String)	Optional value of ID Field to print report for
IN_Screen	(Bool)	Print to Screen
IN_Printer	(Bool)	Print to Printer
IN_Type	(Short)	Export Type (see notes)
IN_Path	(String)	Path to Export Report to

Related: FormClose, FormOpen, FormOpenDialog, FormOpenInquiry, FormOpenLookup, FormOpenLookupReturn, FormOpenMaint, FormOpenReport, FormIsOpen

Error Codes:
-17: Visual Studio Integration Toolkit is not registered.
-22: There is data missing from the function call parameters.
-24: The Form ID has not been registered for use.
-25: The Form Type is not the correct type for the requested action.
-26: An internal Compiler Error has occurred, please report to support.
-27: Visual Studio Integration Toolkit does not have its tables installed correctly.

Notes: Export Types 1=Text file, 2=Tab delimited, 3=Comma delimited, 4=HTML

Name: FormIsOpen

Description: This function is used to check if a form is currently open. Error codes are shown below.

Qualified Name: MenusForVisualStudioTools.Forms.VstmForms.Functions.FormIsOpen

Returns: (Short) The error code.

Parameters:

IN_DeveloperID	(String)	Developer ID to use
IN_FormID	(String)	Form ID to open
IN_SubForm	(Short)	Sub Form to open, leave as 0 for primary
OUT_IsOpen	(Bool)	Set to true if form is currently open

Related: FormClose, FormOpen, FormOpenDialog, FormOpenInquiry, FormOpenLookup, FormOpenLookupReturn, FormOpenMaint, FormOpenReport, FormOpenReportDestination

Error Codes:
-17: Visual Studio Integration Toolkit is not registered.
-22: There is data missing from the function call parameters.
-24: The Form ID has not been registered for use.
-26: An internal Compiler Error has occurred, please report to support.
-27: Visual Studio Integration Toolkit does not have its tables installed correctly.

Getting and Setting Form Properties Functions:

Name:	FormGetNames		
Description:	This function is used to get the Dexterity Form and Window names of a form. Error codes are shown below.		
Qualified Name:	MenusForVisualStudioTools.Forms.VstmForms.Functions.FormGetNames		
Returns:	(Short) The error code.		
Parameters:	IN_DeveloperID	(String)	Developer ID to use
	IN_FormID	(String)	Form ID to use
	IN_SubForm	(Short)	Sub Form to use, leave as 0 for primary
	OUT_Form	(String)	Dexterity Form Name
	OUT_Window	(String)	Dexterity Window Name
Related:	FormGetNumber		
Error Codes:	-17: Visual Studio Integration Toolkit is not registered. -22: There is data missing from the function call parameters. -23: There are no more forms of the request Form Type available, or form not open. -24: The Form ID has not been registered for use. -27: Visual Studio Integration Toolkit does not have its tables installed correctly.		

Name: FormGetNumber

Description: This function is used to get the Custom Form Type and Number of a form. Error codes are shown below.

Qualified Name: MenusForVisualStudioTools.Forms.VstmForms.Functions.FormGetNumber

Returns: (Short) The error code.

Parameters:

IN_DeveloperID	(String)	Developer ID to use
IN_FormID	(String)	Form ID to use
OUT_Type	(Short)	Custom Form Type
OUT_Number	(Short)	Custom Form Number

Related: FormGetNames

Error Codes:
-17: Visual Studio Integration Toolkit is not registered.
-22: There is data missing from the function call parameters.
-23: There are no more forms of the request Form Type available, or form not open.
-24: The Form ID has not been registered for use.
-27: Visual Studio Integration Toolkit does not have its tables installed correctly.

Name: FormGetPosition

Description: This function is used to get the Position of an open form. Error codes are shown below.

Qualified Name: MenusForVisualStudioTools.Forms.VstmForms.Functions.FormGetPosition

Returns: (Short) The error code.

Parameters:

IN_DeveloperID	(String)	Developer ID to use
IN_FormID	(String)	Form ID to use
IN_SubForm	(Short)	Sub Form to use, leave as 0 for primary
OUT_HPos	(Short)	Horizontal Position in Pixels
OUT_VPos	(Short)	Vertical Position in Pixels

Related: FormGetSize, FormGetTitle, FormResize, FormSetPosition, FormSetSize, FormSetTitle

Error Codes:
-17: Visual Studio Integration Toolkit is not registered.
-22: There is data missing from the function call parameters.
-23: There are no more forms of the request Form Type available, or form not open.
-24: The Form ID has not been registered for use.
-26: An internal Compiler Error has occurred, please report to support.
-27: Visual Studio Integration Toolkit does not have its tables installed correctly.

Notes: The form must be open for this function to work.

Name: FormGetSize

Description: This function is used to get the Size of an open form. Error codes are shown below.

Qualified Name: MenusForVisualStudioTools.Forms.VstmForms.Functions.FormGetSize

Returns: (Short) The error code.

Parameters:

IN_DeveloperID	(String)	Developer ID to use
IN_FormID	(String)	Form ID to use
IN_SubForm	(Short)	Sub Form to use, leave as 0 for primary
OUT_HSize	(Short)	Horizontal Size in Pixels
OUT_VSize	(Short)	Vertical Size in Pixels

Related: FormGetPosition, FormGetTitle, FormResize, FormSetPosition, FormSetSize, FormSetTitle

Error Codes:
-17: Visual Studio Integration Toolkit is not registered.
-22: There is data missing from the function call parameters.
-23: There are no more forms of the request Form Type available, or form not open.
-24: The Form ID has not been registered for use.
-26: An internal Compiler Error has occurred, please report to support.
-27: Visual Studio Integration Toolkit does not have its tables installed correctly.

Notes: The form must be open for this function to work.

Name: FormGetTitle

Description: This function is used to get the Title of an open form. Error codes are shown below.

Qualified Name: MenusForVisualStudioTools.Forms.VstmForms.Functions.FormGetTitle

Returns: (Short) The error code.

Parameters:

IN_DeveloperID	(String)	Developer ID to use
IN_FormID	(String)	Form ID to use
IN_SubForm	(Short)	Sub Form to use, leave as 0 for primary
OUT_Title	(String)	Title of window

Related: FormGetPosition, FormGetSize, FormResize, FormSetPosition, FormSetSize, FormSetTitle

Error Codes:
-17: Visual Studio Integration Toolkit is not registered.
-22: There is data missing from the function call parameters.
-23: There are no more forms of the request Form Type available, or form not open.
-24: The Form ID has not been registered for use.
-26: An internal Compiler Error has occurred, please report to support.
-27: Visual Studio Integration Toolkit does not have its tables installed correctly.

Notes: The form must be open for this function to work.

Name: FormResize

Description: This function is used to resize an open form. Error codes are shown below.

Qualified Name: MenusForVisualStudioTools.Forms.VstmForms.Functions.FormResize

Returns: (Short) The error code.

Parameters:

IN_DeveloperID	(String)	Developer ID to use
IN_FormID	(String)	Form ID to use
IN_SubForm	(Short)	Sub Form to use, leave as 0 for primary
IN_HSize	(Short)	Horizontal Size in Pixels
IN_VSize	(Short)	Vertical Size in Pixels

Related: FormGetPosition, FormGetSize, FormGetTitle, FormSetPosition, FormSetSize, FormSetTitle

Error Codes:
-17: Visual Studio Integration Toolkit is not registered.
-22: There is data missing from the function call parameters.
-23: There are no more forms of the request Form Type available, or form not open.
-24: The Form ID has not been registered for use.
-26: An internal Compiler Error has occurred, please report to support.
-27: Visual Studio Integration Toolkit does not have its tables installed correctly.

Notes: The form must be open for this function to work.
This function will resize the window and cause fields to resize as well.

Name: FormSetPosition

Description: This function is used to set the Position of an open form. Error codes are shown below.

Qualified Name: MenusForVisualStudioTools.Forms.VstmForms.Functions.FormSetPosition

Returns: (Short) The error code.

Parameters:

IN_DeveloperID	(String)	Developer ID to use
IN_FormID	(String)	Form ID to use
IN_SubForm	(Short)	Sub Form to use, leave as 0 for primary
IN_HPos	(Short)	Horizontal Position in Pixels
IN_VPos	(Short)	Vertical Position in Pixels

Related: FormGetPosition, FormGetSize, FormGetTitle, FormResize, FormSetSize, FormSetTitle

Error Codes:
-17: Visual Studio Integration Toolkit is not registered.
-22: There is data missing from the function call parameters.
-23: There are no more forms of the request Form Type available, or form not open.
-24: The Form ID has not been registered for use.
-26: An internal Compiler Error has occurred, please report to support.
-27: Visual Studio Integration Toolkit does not have its tables installed correctly.

Notes: The form must be open for this function to work.

Name: FormSetSize

Description: This function is used to set the Size of an open form. Error codes are shown below.

Qualified Name: MenusForVisualStudioTools.Forms.VstmForms.Functions.FormSetSize

Returns: (Short) The error code.

Parameters:

IN_DeveloperID	(String)	Developer ID to use
IN_FormID	(String)	Form ID to use
IN_SubForm	(Short)	Sub Form to use, leave as 0 for primary
IN_HSize	(Short)	Horizontal Size in Pixels
IN_VSize	(Short)	Vertical Size in Pixels

Related: FormGetPosition, FormGetSzie, FormGetTitle, FormResize, FormSetPosition, FormSetTitle

Error Codes:
-17: Visual Studio Integration Toolkit is not registered.
-22: There is data missing from the function call parameters.
-23: There are no more forms of the request Form Type available, or form not open.
-24: The Form ID has not been registered for use.
-26: An internal Compiler Error has occurred, please report to support.
-27: Visual Studio Integration Toolkit does not have its tables installed correctly.

Notes: The form must be open for this function to work.
This function will not cause existing fields to resize, it changes the windows base size.

Name: FormSetTitle

Description: This function is used to set the Title of an open form. Error codes are shown below.

Qualified Name: MenusForVisualStudioTools.Forms.VstmForms.Functions.FormSetTitle

Returns: (Short) The error code.

Parameters:

IN_DeveloperID	(String)	Developer ID to use
IN_FormID	(String)	Form ID to use
IN_SubForm	(Short)	Sub Form to use, leave as 0 for primary
OUT_Title	(String)	Title of window

Related: FormGetPosition, FormGetSize, FormGetTitle, FormResize, FormSetPosition, FormSetSize

Error Codes:
-17: Visual Studio Integration Toolkit is not registered.
-22: There is data missing from the function call parameters.
-23: There are no more forms of the request Form Type available, or form not open.
-24: The Form ID has not been registered for use.
-26: An internal Compiler Error has occurred, please report to support.
-27: Visual Studio Integration Toolkit does not have its tables installed correctly.

Notes: The form must be open for this function to work.

Security Checking Function:

Name:	FormCheckSecurity		
Description:	This function is used to check the Security access to a form and whether access is to the modified version of the form. Error codes are shown below.		
Qualified Name:	MenusForVisualStudioTools.Forms.VstmForms.Functions.FormCheckSecurity		
Returns:	(Short) The error code.		
Parameters:	IN_DeveloperID	(String)	Developer ID to use
	IN_FormID	(String)	Form ID to use
	IN_SubForm	(Short)	Sub Form to use, leave as 0 for primary
	OUT_Access	(Bool)	Access to the form is granted
	OUT_Modified	(Bool)	Access to the modified version is granted
Related:	None		
Error Codes:	-17: Visual Studio Integration Toolkit is not registered. -22: There is data missing from the function call parameters. -23: There are no more forms of the request Form Type available, or form not open. -24: The Form ID has not been registered for use. -27: Visual Studio Integration Toolkit does not have its tables installed correctly.		

Combining and Splitting Key Fields Functions:

Name:	KeyCombineForm		
Description:	This function is used to combine the Developer ID and Form ID into a single string.		
Qualified Name:	MenusForVisualStudioTools.Forms.VstmForms.Functions.KeyCombineForm		
Returns:	(String) The combined string.		
Parameters:	IN_DeveloperID	(String)	Developer ID to use
	IN_FormID	(String)	Form ID to use
Related:	KeyCombineLine, KeySplitForm, KeySplitLine		
Error Codes:	None		

Name:	KeyCombineLine		
Description:	This function is used to combine the key fields ID and Sequence Number into a single string.		
Qualified Name:	MenusForVisualStudioTools.Forms.VstmForms.Functions.KeyCombineLine		
Returns:	(String) The combined string.		
Parameters:	IN_ID	(String)	ID Value to use
	IN_Sequence	(Integer)	Sequence Number to use
Related:	KeyCombineForm, KeySplitForm, KeySplitLine		
Error Codes:	None		

Name:	KeySplitForm		
Description:	This function is used to split a combined Developer ID and Form ID back into separate values.		
Qualified Name:	MenusForVisualStudioTools.Forms.VstmForms.Functions.KeySplitForm		
Returns:	(Short) Zero value.		
Parameters:	IN_Combined	(string)	Combined string to split
	OUT_DeveloperID	(String)	Returned Developer ID
	OUT_FormID	(String)	Returned Form ID
Related:	KeyCombineForm, KeyCombineLine, KeySplitLine		
Error Codes:	None		

Name:	KeySplitLine		
Description:	This function is used to split a combined field ID and Sequence Numbers into separate values.		
Qualified Name:	MenusForVisualStudioTools.Forms.VstmForms.Functions.KeySplitLine		
Returns:	(Short) Zero value.		
Parameters:	IN_Combined	(string)	Combined string to split
	OUT_ID	(String)	Returned ID Value
	OUT_Sequence	(Integer)	Returned Sequence Number
Related:	KeyCombineForm, KeyCombineLine, KeySplitForm		
Error Codes:	None		

Report Writer Functions:

Name:	ReportGetTableHeader		
Description:	This function is used to get the key fields passed into the rw_TableHeaderString() and rw_TableHeaderCurrency() report writer functions.		
Qualified Name:	MenusForVisualStudioTools.Forms.VstmForms.Functions.ReportGetTableHeader		
Returns:	(Short) Zero value.		
Parameters:	OUT_Number	(String)	Number, often used for SOP Number
	OUT_Type	(Short)	Type, often used for SOP Type
	OUT_Control	(Short)	Control, used to specify returned data
Related:	ReportGetTableLine, ReportSetCurrency, ReportSetString		
Error Codes:	None		

Name:	ReportGetTableLine		
Description:	This function is used to get the key fields passed into the rw_TableLineString() and rw_TableLineCurrency() report writer functions.		
Qualified Name:	MenusForVisualStudioTools.Forms.VstmForms.Functions.ReportGetTableLine		
Returns:	(Short) Zero value.		
Parameters:	OUT_Number	(String)	Number, often used for SOP Number
	OUT_Type	(Short)	Type, often used for SOP Type
	OUT_SequenceOne	(Decimal)	Sequence One, often used for Line Item
	OUT_SequenceTwo	(Decimal)	Sequence Two, often used for Component
	OUT_Control	(Short)	Control, used to specify returned data
Related:	ReportGetTableHeader, ReportSetCurrency, ReportSetString		
Error Codes:	None		

Name:	ReportSetCurrency
Description:	This function is used to set the currency value returned by the rw_TableHeaderCurrency() and rw_TableLineCurrency() report writer functions.
Qualified Name:	MenusForVisualStudioTools.Forms.VstmForms.Functions.ReportSetCurrency
Returns:	(Short) Zero value.
Parameters:	IN_Currency (Decimal) Currency value returned to RW function
Related:	ReportGetTableHeader, ReportGetTableLine, ReportSetString
Error Codes:	None

Name:	ReportSetString
Description:	This function is used to set the string value returned by the rw_ReportStart(), rw_ReportEnd(), rw_TableHeaderString() and rw_TableLineString() report writer functions.
Qualified Name:	MenusForVisualStudioTools.Forms.VstmForms.Functions.ReportSetString
Returns:	(Short) Zero value.
Parameters:	IN_String (String) String value returned to RW function
Related:	ReportGetTableHeader, ReportGetTableLine, ReportSetCurrency
Error Codes:	None

Controlling Form Behavior from the Event Handler:

Name:	FormAbortClose
Description:	This function is used to set the Abort Close flag from the Window Post Event Handler to stop the window closing. Error codes are shown below.
Qualified Name:	MenusForVisualStudioTools.Forms.VstmForms.Functions.FormAbortClose
Returns:	(Short) The error code.
Parameters:	None
Related:	FormRejectRecord, FormRejectScript
Error Codes:	-17: Visual Studio Integration Toolkit is not registered. -27: Visual Studio Integration Toolkit does not have its tables installed correctly.
Notes:	This function only works for the Window Post event.

Name:	FormRejectRecord
Description:	This function is used to set the Reject Record flag from the Scrolling Window Fill Event Handler to prevent a record being displayed. Error codes are shown below.
Qualified Name:	MenusForVisualStudioTools.Forms.VstmForms.Functions.FormRejectRecord
Returns:	(Short) The error code.
Parameters:	None
Related:	FormAbortClose, FormRejectScript
Error Codes:	-17: Visual Studio Integration Toolkit is not registered. -27: Visual Studio Integration Toolkit does not have its tables installed correctly.
Notes:	This function only works for the Scrolling Window Fill event.

Name:	FormRejectScript
Description:	This function is used to set the Reject Script flag from the Event Handler to stop the current script completing. Error codes are shown below.
Qualified Name:	MenusForVisualStudioTools.Forms.VstmForms.Functions.FormRejectScript
Returns:	(Short) The error code.
Parameters:	None
Related:	FormAbortClose, FormRejectRecord
Error Codes:	-17: Visual Studio Integration Toolkit is not registered. -27: Visual Studio Integration Toolkit does not have its tables installed correctly.
Notes:	This function works for Field events as well as Save record and Delete events.

Visual Studio Helper Functions

Build 15 of Visual Studio Integration Toolkit adds over 260 helper functions which expose Dexterity commands, functions and function libraries as well as some useful Dynamics GP functions. These functions allow a Visual Studio developer access to features which were previously only available using Dexterity or passthrough sanScript.

Some of the functions are not included in the Dexterity documentation and some functions have been altered to make them multi dictionary capable and accessible from Visual C# or Visual Basic .Net.

All the functions are form level Dexterity functions on the VSTM_Helper form in the Visual Studio Integration Toolkit dictionary. A shortcut to access the functions is to define a helper variable, for example:

```
var helper = MenusForVisualStudioTools.Forms.VstmHelper.Functions;
short response = helper.Ask.Invoke("Do you want to save this record?", "&Save",
"&Discard", "&Cancel");
```

Below is a list of the functions grouped by functional areas with a brief description of the functionality:

Types: D = Dexterity function wrapped, G = Dynamics GP function wrapped, C = Custom wrapper or code

Helper Function Registration and Version

Function Name	Type	Description	Version
HelperGetVersion	C	Return full product version details	v11.0+
HelperRegistered	C	Check if the Helper Functions Module is registered	v11.0+
HelperVersionInteger	C	Return the Helper Functions Version Number as Integer	v11.0+
HelperVersionString	C	Return the Helper Function Version Number as String	v11.0+

Messages and Constants

Function Name	Type	Description	Version
ConstantGetInteger	C	Get an Integer Constant from a Dictionary	v11.0+
ConstantGetIntegerForm	C	Get an Integer Constant from a Dictionary Form	v11.0+
ConstantGetString	C	Get a String Constant from a Dictionary	v11.0+
ConstantGetStringForm	C	Get a String Constant from a Dictionary Form	v11.0+
Getmsg	D	Get Message from core Dynamics dictionary	v11.0+
GetmsgEx	CD	Get Message from any installed dictionary	v11.0+

Clearing and Filling Variables

Function Name	Type	Description	Version
ClearBoolean	C	Set a Boolean variable to its empty or cleared state, false	v11.0+
ClearDate	C	Set a Date variable to its empty or cleared state, 00/00/0000	v11.0+
ClearDateTime	C	Set a DateTime variable to its empty or cleared state, 00/00/0000 00:00:00	v11.0+
ClearDecimal	C	Set a Decimal variable to its empty or cleared state, 0.00000	v11.0+
ClearInteger	C	Set an Integer variable to its empty or cleared state, 0	v11.0+
ClearShort	C	Set a Short variable to its empty or cleared state, 0	v11.0+
ClearString	C	Set a String variable to its empty or cleared state, ""	v11.0+
ClearText	C	Set a Text variable to its empty or cleared state, ""	v11.0+
ClearTime	C	Set a Time variable to its empty or cleared state, 00:00:00	v11.0+
EmptyBoolean	C	Check a Boolean variable for its empty or cleared state, false	v11.0+
EmptyDate	C	Check a Date variable for its empty or cleared state, 00/00/0000	v11.0+
EmptyDateTime	C	Check a DateTime variable for its empty or cleared state, 00/00/0000 00:00:00	v11.0+
EmptyDecimal	C	Check a Decimal variable for its empty or cleared state, 0.00000	v11.0+
EmptyInteger	C	Check an Integer variable for its empty or cleared state, 0	v11.0+
EmptyShort	C	Check a Short variable for its empty or cleared state, 0	v11.0+
EmptyString	C	Check a String variable for its empty or cleared state, ""	v11.0+
EmptyText	C	Check a Text variable for its empty or cleared state, ""	v11.0+
EmptyTime	C	Check a Time variable for its empty or cleared state, 00:00:00	v11.0+
FillBoolean	C	Set a Boolean variable to its filled or max state, true	v11.0+
FillDate	C	Set a Date variable to its filled or max state, 31/12/9999	v11.0+
FillDateTime	C	Set a DateTime variable to its filled or max state, 31/12/9999 23:59:59	v11.0+
FillDecimal	C	Set a Decimal variable to its filled or max state, maximum value for datatype	v11.0+
FillInteger	C	Set an Integer variable to its filled or max state, maximum value for datatype	v11.0+

FillShort	C	Set a Short variable to its filled or max state, maximum value for datatype	v11.0+
FillString	C	Set a String variable to its filled or max state, ASCII code 255 for length of variable	v11.0+
FillText	C	Set a Text variable to its filled or max state, ASCII code 255 for length of variable	v11.0+
FillTime	C	Set a Time variable to its empty or cleared state, 23:59:59	v11.0+
FilledBoolean	C	Check a Boolean variable for its filled or max state, true	v11.0+
FilledDate	C	Check a Date variable for its filled or max state, 31/12/9999	v11.0+
FilledDateTime	C	Check a DateTime variable for its filled or max state, 31/12/9999 23:59:59	v11.0+
FilledDecimal	C	Check a Decimal variable for its filled or max state, maximum value for datatype	v11.0+
FilledInteger	C	Check an Integer variable for its filled or max state, maximum value for datatype	v11.0+
FilledShort	C	Check a Short variable for its filled or max state, maximum value for datatype	v11.0+
FilledString	C	Check a String variable for its filled or max state, ASCII code 255 for length of variable	v11.0+
FilledText	C	Check a Text variable for its filled or max state, ASCII code 255 for length of variable	v11.0+
FilledTime	C	Check a Time variable for its empty or cleared state, 23:59:59	v11.0+

Clearing and Filling Fields

Function Name	Type	Description	Version
FieldClear	C	Clear a Field, see FieldEmpty	v11.0+
FieldEmpty	C	Check if Field is cleared, see FieldClear	v11.0+
FieldFill	C	Fill a Field, see FieldFilled	v11.0+
FieldFilled	C	Check if Field is filled, see FieldFill	v11.0+

Field Hiding, Locking, Disabling

Function Name	Type	Description	Version
FieldDisable	C	Disable a Field, see FieldEnable	v11.0+
FieldEnable	C	Enable a Field, see FieldDisable	v11.0+
FieldHide	C	Hide a Field, see FieldShow	v11.0+
FieldLock	C	Lock a Field, see FieldUnlock	v11.0+
FieldShow	C	Show a Field, see FieldHide	v11.0+
FieldUnlock	C	Unlock a Field, See FieldLock	v11.0+

Field Size and Position

Function Name	Type	Description	Version
FieldGetPosition	CD	Get Position of a Field	v11.0+
FieldGetSize	CD	Get Size of a Field	v11.0+
FieldSetPosition	C	Set Position of a Field	v11.0+
FieldSetSize	C	Set Size of a Field	v11.0+

Field Control

Function Name	Type	Description	Version
FieldClearForceChange	C	Clear Force Change Event when leaving Field	v11.0+
FieldFocus	C	Change Focus to specified field, takes effect when control is returned to the user interface. See FieldSetFocus	v11.0+
FieldForceChange	C	Force Change Event when leaving Field	v11.0+
FieldMap	CD	Enter data into Field as though entered by user. Window must have focus before issuing command, see WindowFocus and FieldSetFocus	v14.0+
FieldRedraw	CD	Force specified field to redraw	v11.0+
FieldSetFocus	CD	Change Focus to specified field, takes effect immediately. See FieldFocus	v11.0+

Field Properties

Function Name	Type	Description	Version
FieldGetBooleanProperty	CD	Get a Boolean Property of a field, properties available: 157 = FIELD_PROP_AUTOCOMPLETE 146 = FIELD_PROP_BDL_DROP_INDICATOR 11 = FIELD_PROP_CANCEL 30 = FIELD_PROP_DEFAULT 152 = FIELD_PROP_DISABLED 88 = FIELD_PROP_EDITABLE 150 = FIELD_PROP_FOCUS 45 = FIELD_PROP_HYPERSPACE 151 = FIELD_PROP_LOCKED 65 = FIELD_PROP_REQUIRED 72 = FIELD_PROP_SURROUNDING_BOX 102 = FIELD_PROP_TAB_STOP 74 = FIELD_PROP_VISIBLE	v11.0+
FieldGetCaption	CD	Get the Caption of a field	v11.0+

FieldGetIntegerProperty	CD	Get an Integer Property of a field, properties available: 147 = FIELD_PROP_BDL_DROP_POS_X 148 = FIELD_PROP_BDL_DROP_POS_Y 149 = FIELD_PROP_BUTTON_STYLE 0 = BUTTON_STYLE_TEXT_ONLY 1 = BUTTON_STYLE_TEXT_ON_LEFT 2 = BUTTON_STYLE_TEXT_ON_RIGHT 3 = BUTTON_STYLE_TEXT_ON_TOP 4 = BUTTON_STYLE_TEXT_ON_BOTTOM 5 = BUTTON_STYLE_GRAPHIC_ONLY	v11.0+
FieldGetStringProperty	CD	Get a String Property of a field, properties available: 158 = FIELD_PROP_AUTOCOMPLETE_LOOKUP_ITEM 12 = FIELD_PROP_CAPTION	v11.0+
FieldGetToolTip	CD	Get the ToolTip of a field	v11.0+
FieldSetBooleanProperty	CD	Set a Boolean Property of a field, properties available: 157 = FIELD_PROP_AUTOCOMPLETE 146 = FIELD_PROP_BDL_DROP_INDICATOR 11 = FIELD_PROP_CANCEL 30 = FIELD_PROP_DEFAULT 152 = FIELD_PROP_DISABLED 88 = FIELD_PROP_EDITABLE 150 = FIELD_PROP_FOCUS 45 = FIELD_PROP_HYPERSPACE 151 = FIELD_PROP_LOCKED 65 = FIELD_PROP_REQUIRED 72 = FIELD_PROP_SURROUNDING_BOX 102 = FIELD_PROP_TAB_STOP 74 = FIELD_PROP_VISIBLE	v11.0+
FieldSetCaption	CD	Set the Caption of a field	v11.0+
FieldSetIntegerProperty	CD	Set an Integer Property of a field, properties available: 147 = FIELD_PROP_BDL_DROP_POS_X 148 = FIELD_PROP_BDL_DROP_POS_Y 149 = FIELD_PROP_BUTTON_STYLE 0 = BUTTON_STYLE_TEXT_ONLY 1 = BUTTON_STYLE_TEXT_ON_LEFT 2 = BUTTON_STYLE_TEXT_ON_RIGHT 3 = BUTTON_STYLE_TEXT_ON_TOP 4 = BUTTON_STYLE_TEXT_ON_BOTTOM 5 = BUTTON_STYLE_GRAPHIC_ONLY	v11.0+
FieldSetStringProperty	CD	Set a String Property of a field, properties available: 158 = FIELD_PROP_AUTOCOMPLETE_LOOKUP_ITEM 12 = FIELD_PROP_CAPTION	v11.0+

FieldSetToolTip	CD	Set the ToolTip of a field	v11.0+

Field Colors

Function Name	Type	Description	Version
FieldGetColor	CD	Get a Field's colors (RGB value) and pattern settings. Colors: 131072 = COLOR_SYSTEM 262144 = COLOR_TRANSPARENT 0 = COLOR_BLACK 16777215 = COLOR_WHITE 16711680 = COLOR_RED 65280 = COLOR_BRIGHT_GREEN 255 = COLOR_BLUE 65535 = COLOR_TURQUOISE 16711935 = COLOR_PINK 16776960 = COLOR_YELLOW 32896 = COLOR_TEAL 32768 = COLOR_GREEN 8421376 = COLOR_DARK_YELLOW 8388736 = COLOR_VIOLET 8388608 = COLOR_DARK_RED 128 = COLOR_DARK_BLUE 8421504 = COLOR_MEDIUM_GRAY 12632256 = COLOR_LIGHT_GRAY 16777088 = COLOR_LIGHT_YELLOW 33554498 = SYSTEM_WINDOW_BACKGROUND 33554499 = SYSTEM_WINDOW_TEXT Patterns: 2 = PATTERN_NONE 4 = PATTERN_25_SHADING 3 = PATTERN_50_SHADING 5 = PATTERN_75_SHADING 6 = PATTERN_HORIZONTAL 7 = PATTERN_VERTICAL 8 = PATTERN_UP_DIAGONAL 9 = PATTERN_DOWN_DIAGONAL 10 = PATTERN_GRID 11 = PATTERN_TRELLIS	v11.0+

FieldSetBackColor	CD	Set a Field's back color (RGB value) Colors: 131072 = COLOR_SYSTEM 262144 = COLOR_TRANSPARENT 0 = COLOR_BLACK 16777215 = COLOR_WHITE 16711680 = COLOR_RED 65280 = COLOR_BRIGHT_GREEN 255 = COLOR_BLUE 65535 = COLOR_TURQUOISE 16711935 = COLOR_PINK 16776960 = COLOR_YELLOW 32896 = COLOR_TEAL 32768 = COLOR_GREEN 8421376 = COLOR_DARK_YELLOW 8388736 = COLOR_VIOLET 8388608 = COLOR_DARK_RED 128 = COLOR_DARK_BLUE 8421504 = COLOR_MEDIUM_GRAY 12632256 = COLOR_LIGHT_GRAY 16777088 = COLOR_LIGHT_YELLOW 33554498 = SYSTEM_WINDOW_BACKGROUND	v11.0+
FieldSetFont	CD	Set a Field's font setting Fonts: 1 = FONT_COURIER 2 = FONT_HELVETICA 4 = FONT_TIMES 3 = FONT_SYSTEM	v11.0+

FieldSetFontColor	CD	Set a Field's font color (RGB value) Colors: 131072 = COLOR_SYSTEM 262144 = COLOR_TRANSPARENT 0 = COLOR_BLACK 16777215 = COLOR_WHITE 16711680 = COLOR_RED 65280 = COLOR_BRIGHT_GREEN 255 = COLOR_BLUE 65535 = COLOR_TURQUOISE 16711935 = COLOR_PINK 16776960 = COLOR_YELLOW 32896 = COLOR_TEAL 32768 = COLOR_GREEN 8421376 = COLOR_DARK_YELLOW 8388736 = COLOR_VIOLET 8388608 = COLOR_DARK_RED 128 = COLOR_DARK_BLUE 8421504 = COLOR_MEDIUM_GRAY 12632256 = COLOR_LIGHT_GRAY 16777088 = COLOR_LIGHT_YELLOW 33554499 = SYSTEM_WINDOW_TEXT	v11.0+
FieldSetPattern	CD	Set a Field's pattern setting. Patterns: 2 = PATTERN_NONE 4 = PATTERN_25_SHADING 3 = PATTERN_50_SHADING 5 = PATTERN_75_SHADING 6 = PATTERN_HORIZONTAL 7 = PATTERN_VERTICAL 8 = PATTERN_UP_DIAGONAL 9 = PATTERN_DOWN_DIAGONAL 10 = PATTERN_GRID 11 = PATTERN_TRELLIS	v11.0+

FieldSetPatternColor	CD	Set a Field's pattern color (RGB value) Colors: 131072 = COLOR_SYSTEM 262144 = COLOR_TRANSPARENT 0 = COLOR_BLACK 16777215 = COLOR_WHITE 16711680 = COLOR_RED 65280 = COLOR_BRIGHT_GREEN 255 = COLOR_BLUE 65535 = COLOR_TURQUOISE 16711935 = COLOR_PINK 16776960 = COLOR_YELLOW 32896 = COLOR_TEAL 32768 = COLOR_GREEN 8421376 = COLOR_DARK_YELLOW 8388736 = COLOR_VIOLET 8388608 = COLOR_DARK_RED 128 = COLOR_DARK_BLUE 8421504 = COLOR_MEDIUM_GRAY 12632256 = COLOR_LIGHT_GRAY 16777088 = COLOR_LIGHT_YELLOW	v11.0+

Field Fonts

Function Name	Type	Description	Version
FieldGetFont	CD	Get a Field's font, font size and font style settings Fonts: 1 = FONT_COURIER 2 = FONT_HELVETICA 4 = FONT_TIMES 3 = FONT_SYSTEM Styles: 0 = FONT_STYLE_PLAIN 1 = FONT_STYLE_BOLD 2 = FONT_STYLE_ITALIC 4 = FONT_STYLE_UNDERLINE Styles can be added to combine	v11.0+
FieldSetFontSize	CD	Set a Field's font side setting	v11.0+
FieldSetFontStyle		Set a Field's font style setting. Styles: 0 = FONT_STYLE_PLAIN 1 = FONT_STYLE_BOLD 2 = FONT_STYLE_ITALIC 4 = FONT_STYLE_UNDERLINE Styles can be added to combine	

Text Fields

Function Name	Type	Description	Version
FieldGetSelection	CD	Get the string value, position and length of selection from a text field.	v11.0+
FieldInsertStringInText	CD	Insert string into text field replacing current selection	v11.0+
FieldParseText	CD	Parse text field into strings of specified length or to end of line. Does not break words	v11.0+
FieldSetSelection	CD	Set the position and length of selection in a text field.	v11.0+

Dates and Times

Function Name	Type	Description	Version
MkDate	D	Create a Date variable based on Month, Day, Year	v11.0+
MkDateTime	C	Create a DateTime variable based on Month, Day, Year and Hours, Minutes, Seconds	v12.0+
MkTime	D	Create a Time variable based on Hours, Minutes, Seconds	v11.0+
SetDate	D	Update a Date variable based on Month, Day, Year. Leave Month, Day, Year as zero to leave unchanged	v11.0+
Sysdate	D	Get System Date, use parameter true for server	v11.0+
Sysdatetime	D	Get System Date and Time	v11.0+
Systime	D	Get System Time, use parameter true for server	v11.0+

Formatting Fields

Function Name	Type	Description	Version
Format	D	Format currency to a string Negative Types: 1 = SYSTEMNEG 2 = MINUSNEG 3 = PARENNEG 4 = CRNEG + 100 for % at the end	v11.0+
FormatPhoneNumber	G	Format Phone Number as per Message 6055	v11.0+
FormatResource	C	Format Dexterity Form, Window and Field names for use in runtime executed code.	v11.0+
FormatSocialSecurityNumber	G	Format Social Security Number as per Message 19311	v11.0+

List Fields

Function Name	Type	Description	Version
ListAddItem	C	Add an Item to the bottom of a List Field	v11.0+
ListChangeItem	C	Change the name of an item in a List Field	v11.0+
ListCountItems	C	Count the number of items in a List Field	v11.0+
ListDeleteItem	C	Delete an item in a List Field	v11.0+
ListFindData	C	Find an item in a List Field by its Data value	v11.0+
ListFindItem	C	Find an item in a List Field by its Name value	v11.0+
ListGetInsertPosFromVisualPos	CD	Get the Insert Position of an item in a List Field from its Visual Position	v11.0+
ListGetVisualPosFromInsertPos	CD	Get the Visual Position of an item in a List Field from its Insert Position	v11.0+
ListInsertItem	C	Insert an item into a List Field	v11.0+
ListItemData	C	Get the Data value of an item in a List Field	v11.0+
ListItemName	C	Get the Name value of an item in a List Field	v11.0+

System Dialogs

Function Name	Type	Description	Version
Ask	D	Display a system dialog with a string prompt & one, two or three buttons	v11.0+
AskText	D	Display a system dialog with a text prompt one two or three buttons	v11.0+
Beep	D	Play a sound: 1=ERRORSOUND, 2=WARNSOUND, 3=INFOSOUND	v11.0+
Debug	D	Display a system dialog with a string prompt & an OK button. Only displayed when Debug >> Show Debug Messages is enabled	v11.0+
DebugText	D	Display a system dialog with a text prompt & an OK button. Only displayed when Debug >> Show Debug Messages is enabled	v11.0+
Error	D	Display an error system dialog with a string prompt & an OK button.	v11.0+
ErrorText	D	Display an error system dialog with a text prompt & an OK button.	v11.0+
Getfile	D	Open a dialog to select a file to open	v11.0+
Getstring	D	Open a dialog and prompt for a string or password	v11.0+
GetValidSystemPassword	G	Prompt for a valid Dynamics GP System Password	v11.0+
PathSelectPathname	D	Open a dialog to select a folder location	v11.0+
Savefile	D	Open a dialog to select a file to sale	v11.0+
Warning	D	Display a warning system dialog with a string prompt & an OK button.	v11.0+
WarningText	D	Display a warning system dialog with a text prompt & an OK button.	v11.0+

Launching External Applications

Function Name	Type	Description	Version
FileLaunch	D	Open or Print specified File. On Web Client it will transfer the file to client first. Actions: 1 = FILE_LAUNCH_ACTION_OPEN 2 = FILE_LAUNCH_ACTION_PRINT 0 = FILE_LAUNCH_ACTION_NONE (Web Client only)	v12.0+
RunApplication	D	Run an external application with parameters. Leave application blank to launch default program based on extension. For Web Client runs on server	v11.0+
RunMacro	D	Run a Macro file	v11.0+
RunMacroFast	C	Run a Macro file with fast screen refreshing	v12.0+
UtilityLaunchUrl	D	Open browser window to display URL	v12.0+
WinHelpLaunchUrl	D	Re-use open browser window to display URL	v12.0+

Dictionary Resources

Function Name	Type	Description	Version
ResourceGetDisplayName	D	Get the Display Name for a dictionary resource Types: 2 = FORMTYPE 1 = TABLETYPE 42 = GROUPTYPE 23 = REPORTTYPE	v11.0+
ResourceGetID	D	Get the Resource ID for a Dictionary Resource Types: 1 = Tables 2 = Forms 6 = Data types 7 = Composites 8 = Formats 9 = Strings 15 = Generic pictures 18 = Fields 21 = Procedure scripts 23 = Reports 26 = Global variables 28 = Metafiles 29 = Mac Pictures 30 = Messages 42 = Table Groups 48 = Constants 62 = Functions	v11.0+

ResourceGetNthResource	D	Get the Resource ID, Name and Series for a specified Index value for a Dictionary Resource Types: 1 = Tables 2 = Forms 6 = Data types 7 = Composites 8 = Formats 9 = Strings 15 = Generic pictures 18 = Fields 21 = Procedure scripts 23 = Reports 26 = Global variables 28 = Metafiles 29 = Mac Pictures 30 = Messages 42 = Table Groups 48 = Constants 62 = Functions Series: 1 = Financial 2 = Sales 3 = Purchasing 4 = Inventory 5 = Payroll 6 = Project 7 = System	v11.0+
ResourceGetPhysicalName	C	Get the Physical Name for a Table or Field by Resource ID Types: 1 = Tables 18 = Fields	v12.0+
FieldGetPhysicalName	C	Get the Physical Name for a Field by Resource Name	v12.0+
TableGetPhysicalName	C	Get the Physical Name for a by Table Resource Name	v12.0+

ResourceGetResourceName	D	Get the Resource Name for a Dictionary Resource Types: 1 = Tables 2 = Forms 6 = Data types 7 = Composites 8 = Formats 9 = Strings 15 = Generic pictures 18 = Fields 21 = Procedure scripts 23 = Reports 26 = Global variables 28 = Metafiles 29 = Mac Pictures 30 = Messages 42 = Table Groups 48 = Constants 62 = Functions	v11.0+
ResourceGetSubResourceName	D	Get the Resource Name for a specified Dictionary Sub Resource Type: 2 = MT_FORM 3 = MT_REPORT Sub Type: 71 = DT_COMMAND	v11.0+

Working with SQL

Function Name	Type	Description	Version
SqlClear	D	Clear the SQL Connection and Result Sets	v11.0+
SqlConnect	D	Create a SQL Connection using current SQL login	v11.0+
SqlDate	G	Format Date as String for SQL Server	v11.0+
SqlDescribeColumn	D	Retrieve details of specified column. Types: 20 = DATATYPE_BOOLEAN 28 = DATATYPE_VCURRENCY 18 = DATATYPE_DATE 0 = DATATYPE_INTEGER 1 = DATATYPE_LONG_INTEGER 4 = DATATYPE_STRING 5 = DATATYPE_TEXT 19 = DATATYPE_TIME	v11.0+
SqlExecute	D	Execute SQL Statements via SQL Connection	v11.0+
SqlFetchNext	D	Fetches the next row from the data set. Returned value of 31 indicates that there is no more data	v11.0+
SqlFormatStrings	G	Format String for SQL Server	v11.0+
SqlGetData	D	Get the data from the specified column in the current data row	v11.0+
SqlGetError	D	Return details of last error on SQL Connection. Internal Error Numbers: 0 = No Error 41 = Cannot find specified column 56 = Cannot find specified table 58 = Syntax error in SQL statement	v11.0+
SqlGetNextResults	D	Move to next result set. Returned value of 31 indicates that there is no more data	v11.0+
SqlGetNumCols	D	Returns the number of columns in the current result set	v11.0+
SqlGetNumRows	D	Returns the number of rows in the current result set	v11.0+
SqlTerminate	D	Closes the SQL Connection	v11.0+
SqlTime	G	Format Time as String for SQL Server	v11.0+

Executing Dexterity sanScript

Function Name	Type	Description	Version
Execute	CD	Execute Dexterity sanScript script in the context of specified Dictionary	v11.0+
ExecuteEx	CD	Execute Dexterity sanScript script in the context of specified Dictionary and return compiler errors	v11.0+
ExecuteEx1	CD	Execute Dexterity sanScript script in the context of specified Dictionary with 1 parameter and return compiler errors. Parameter format: inout text INOUT_TextX ; { where X = 1 }	v11.0+
ExecuteEx2	CD	Execute Dexterity sanScript script in the context of specified Dictionary with 2 parameters and return compiler errors. Parameter format: inout text INOUT_TextX ; { where X = 1 to 2 }	v11.0+
ExecuteEx3	CD	Execute Dexterity sanScript script in the context of specified Dictionary with 3 parameters and return compiler errors. Parameter format: inout text INOUT_TextX ; { where X = 1 to 3 }	v11.0+
ExecuteEx4	CD	Execute Dexterity sanScript script in the context of specified Dictionary with 4 parameter and return compiler errors. Parameter format: inout text INOUT_TextX ; { where X = 1 to 4 }	v11.0+
ExecuteEx5	CD	Execute Dexterity sanScript script in the context of specified Dictionary with 5 parameters and return compiler errors. Parameter format: inout text INOUT_TextX ; { where X = 1 to 5 }	v11.0+
ExecuteEx6	CD	Execute Dexterity sanScript script in the context of specified Dictionary with 6 parameters and return compiler errors. Parameter format: inout text INOUT_TextX ; { where X = 1 to 6 }	v11.0+
ExecuteEx7	CD	Execute Dexterity sanScript script in the context of specified Dictionary with 7 parameter and return compiler errors. Parameter format: inout text INOUT_TextX ; { where X = 1 to 7 }	v11.0+
ExecuteEx8	CD	Execute Dexterity sanScript script in the context of specified Dictionary with 8 parameters and return compiler errors. Parameter format: inout text INOUT_TextX ; { where X = 1 to 8 }	v11.0+
ExecuteEx9	CD	Execute Dexterity sanScript script in the context of specified Dictionary with 9 parameters and return compiler errors. Parameter format: inout text INOUT_TextX ; { where X = 1 to 9 }	v11.0+
ExecuteEx10	CD	Execute Dexterity sanScript script in the context of specified Dictionary with 10 parameters and return compiler errors. Parameter format: inout text INOUT_TextX ; { where X = 1 to 10 }	v11.0+

Executing Dexterity sanScript against Modified Dictionary Forms (Requires Forms module registered)

Use these functions when needing to execute Dexterity sanScript code against a modified form when actually referencing a modifier added field. There is no need to use these functions if only referencing window fields that exist on the original version of the form.

Note: These functions required the Forms module added in Build 18 to be registered.

Function Name	Type	Description	Version
ExecuteModified	CD	Execute Dexterity sanScript script in the context of specified Modified Dictionary	v18.0+
ExecuteModifiedEx	CD	Execute Dexterity sanScript script in the context of specified Modified Dictionary and return compiler errors	v18.0+
ExecuteModifiedEx1	CD	Execute Dexterity sanScript script in the context of specified Modified Dictionary with 1 parameter and return compiler errors. Parameter format: local text INOUT_TextX ; { where X = 1 } OLE_GetProperty("INOUT_ParameterX", INOUT_TextX); { Read Parameter at start of script } OLE_SetProperty("INOUT_ParameterX", INOUT_TextX); { Write Parameter at end of script }	v18.0+
ExecuteModifiedEx2	CD	Execute Dexterity sanScript script in the context of specified Modified Dictionary with 2 parameters and return compiler errors. Parameter format: local text INOUT_TextX ; { where X = 1 to 2 } OLE_GetProperty("INOUT_ParameterX", INOUT_TextX); { Read Parameter at start of script } OLE_SetProperty("INOUT_ParameterX", INOUT_TextX); { Write Parameter at end of script }	v18.0+
ExecuteModifiedEx3	CD	Execute Dexterity sanScript script in the context of specified Modified Dictionary with 3 parameters and return compiler errors. Parameter format: local text INOUT_TextX ; { where X = 1 to 3 } OLE_GetProperty("INOUT_ParameterX", INOUT_TextX); { Read Parameter at start of script } OLE_SetProperty("INOUT_ParameterX", INOUT_TextX); { Write Parameter at end of script }	v18.0+
ExecuteModifiedEx4	CD	Execute Dexterity sanScript script in the context of specified Modified Dictionary with 4 parameter and return compiler errors. Parameter format: local text INOUT_TextX ; { where X = 1 to 4 } OLE_GetProperty("INOUT_ParameterX", INOUT_TextX); { Read Parameter at start of script } OLE_SetProperty("INOUT_ParameterX", INOUT_TextX); { Write Parameter at end of script }	v18.0+

ExecuteModifiedEx5	CD	Execute Dexterity sanScript script in the context of specified Modified Dictionary with 5 parameters and return compiler errors. Parameter format: local text INOUT_TextX ; { where X = 1 to 5 } OLE_GetProperty("INOUT_ParameterX", INOUT_TextX); { Read Parameter at start of script } OLE_SetProperty("INOUT_ParameterX", INOUT_TextX); { Write Parameter at end of script }	v18.0+
ExecuteModifiedEx6	CD	Execute Dexterity sanScript script in the context of specified Modified Dictionary with 6 parameters and return compiler errors. Parameter format: local text INOUT_TextX ; { where X = 1 to 6 } OLE_GetProperty("INOUT_ParameterX", INOUT_TextX); { Read Parameter at start of script } OLE_SetProperty("INOUT_ParameterX", INOUT_TextX); { Write Parameter at end of script }	v18.0+
ExecuteModifiedEx7	CD	Execute Dexterity sanScript script in the context of specified Modified Dictionary with 7 parameter and return compiler errors. Parameter format: local text INOUT_TextX ; { where X = 1 to 7 } OLE_GetProperty("INOUT_ParameterX", INOUT_TextX); { Read Parameter at start of script } OLE_SetProperty("INOUT_ParameterX", INOUT_TextX); { Write Parameter at end of script }	v18.0+
ExecuteModifiedEx8	CD	Execute Dexterity sanScript script in the context of specified Modified Dictionary with 8 parameters and return compiler errors. Parameter format: local text INOUT_TextX ; { where X = 1 to 8 } OLE_GetProperty("INOUT_ParameterX", INOUT_TextX); { Read Parameter at start of script } OLE_SetProperty("INOUT_ParameterX", INOUT_TextX); { Write Parameter at end of script }	v18.0+
ExecuteModifiedEx9	CD	Execute Dexterity sanScript script in the context of specified Modified Dictionary with 9 parameters and return compiler errors. Parameter format: local text INOUT_TextX ; { where X = 1 to 9 } OLE_GetProperty("INOUT_ParameterX", INOUT_TextX); { Read Parameter at start of script } OLE_SetProperty("INOUT_ParameterX", INOUT_TextX); { Write Parameter at end of script }	v18.0+
ExecuteModifiedEx10	CD	Execute Dexterity sanScript script in the context of specified Modified Dictionary with 10 parameters and return compiler errors. Parameter format: local text INOUT_TextX ; { where X = 1 to 10 } OLE_GetProperty("INOUT_ParameterX", INOUT_TextX); { Read Parameter at start of script } OLE_SetProperty("INOUT_ParameterX", INOUT_TextX); { Write Parameter at end of script }	v18.0+

Files and Paths

Function Name	Type	Description	Version
FileDelete	D	Delete a File, always returns true even file deletion failed, see FileProbe	v11.0+
FileGetSize	D	Retrieve size of a File in bytes	v11.0+
FileGetTempDirectory	D	Retrieve path to workstation's Temp directory	v11.0+
FileProbe	D	Confirm if specified File exists	v11.0+
PathCreateFolder	D	Create a new folder	v11.0+
PathDoesExist	D	Check if Path exists	v11.0+
PathGetForApp	D	Return Pathname for system components Options: 1 = PATH_EXEFOLDER 2 = PATH_SETFOLDER 3 = PATH_DATAFOLDER 4 = PATH_DEXINIFILE 5 = PATH_SETFILE	v11.0+
PathMakeGeneric	D	Convert Native Pathname to Generic format	v11.0+
PathMakeNative	D	Convert Generic Pathname to Native format	v11.0+
PathParseFileFromPath	D	Return File name from a full pathname	v11.0+
PathParsePathFromPath	D	Return Folder path from a full pathname	v11.0+

Launch File

Function Name	Type	Description	Version
LaunchCountProds	D	Get Number of Products in the Launch File	v11.0+
LaunchDictionaryInstalled	C	Confirm is a Dictionary is installed in the Launch File	v11.0+
LaunchGetFileName	D	Get Path Name of the Launch File	v11.0+
LaunchGetProdID	D	Get the Product ID from the Position in the Launch File	v11.0+
LaunchGetProdName	D	Get the Product Name from the Product ID in the Launch File	v11.0+
LaunchGetProdPosition	D	Get the Position from the Product ID in the Launch File	v11.0+

Window Size and Position

Function Name	Type	Description	Version
WindowGetPosition	CD	Get the Position of a Window	v11.0+
WindowGetSize	CD	Get the Size of a Window	v11.0+
WindowSetBaseSize	CD	Set the Base Size of a Window, does not affect fields	v11.0+
WindowSetPosition	C	Set the Position of a Window	v11.0+
WindowSetSize	C	Set the Size of a Window, fields may be affected	v11.0+

Window Control

Function Name	Type	Description	Version
WindowChanged	CD	Returns true if data in the window has changed. Leave Window parameter blank to perform function against all windows on the Form. See also WindowClearChanges and WindowForceChanges	v11.0+
WindowClear	CD	Clears the specified window, can be used on scrolling windows. Leave Window parameter blank to perform function against all windows on the Form. See WindowFill	v11.0+
WindowClearChanges	CD	Clears the changed flag from the window. Leave Window parameter blank to perform function against all windows on the Form. See also WindowChanged and WindowForceChanges	v11.0+
WindowFocus	C	Bring focus to window by re-opening it	v11.0+
WindowForceChanges	CD	Sets the changed flag from the window. Leave Window parameter blank to perform function against all windows on the Form. See also WindowChanged and WindowClearChanges	v11.0+
WindowForceRedraw	CD	Force window to be updated immediately before control is returned to the user interface. Call once when window is opened.	v11.0+
WindowGetTitleByProduct	D	Get the Title of the specified window	v11.0+
WindowSetTitleByProduct	D	Set the Title of the specified window	v14.0+
WindowIsOpen	CD	Returns true if the specified window is open. Leave Window parameter blank to perform function against the Form	v11.0+
WindowPullFocus	CD	Pull the focus from the window to force event scripts to execute	v11.0+
WindowRequired	CD	Returns true if all required fields on the window have data. Leave Window parameter blank to perform function against all windows on the Form	v11.0+
WindowReturnMapped	CD	Return Mapped Fields on child window to parent window	v14.0+

Scrolling Window Control

Function Name	Type	Description	Version
WindowClear	CD	Clears the specified window, can be used on scrolling windows. Leave Window parameter blank to perform function against all windows on the Form. See WindowFill	v11.0+
WindowExpand	CD	Expand or Shrink a Scrolling window	v11.0+
WindowFill	CD	Fill scrolling window with data using current table Modes: 0 = Default 1 = From Top 2 = From Current 3 = From Bottom 4 = Redraw 5 = Redraw Up 6 = Redraw Down See WindowClear	v11.0+
WindowGetLineTag	CD	Get current Tag value for selected Scrolling window line	v14.0+
WindowScrollScrollingWindow	CD	Move record in Scrolling window, designed for single line scrolling windows. Must have focus on the line. Scroll Type: true = SCROLLTYPE_NEXT false = SCROLLTYPE_PREV Results: 0 = FOCUS_OK 1 = FOCUS_DIVERTED 2 = FOCUS_RESTART 3 = FOCUS_DENIED 4 = FOCUS_NOEXIST 5 = FOCUS_NOEXIST_EOF 6 = FOCUS_NOEXIST_SOF 7 = FOCUS_NOT_IN_SWIN	v11.0+
WindowSetFocusLine	CD	Set the focus to the specified Line value.	v14.0+
WindowSetFocusTag	CD	Set the focus to the specified Tag value. To go to new line, use WindowClear, WindowFill from bottom and WindowSetFocusTag to -1	v14.0+
WindowSetLineMode	CD	Change the editable Mode for a Scrolling Window Modes: 0 = LINEMODE_EDITABLE 1 = LINEMODE_ADDS_ALLOWED 2 = LINEMODE_BROWSE_ONLY	v11.0+

Runtime Engine and Dictionary Versions

Function Name	Type	Description	Version
RuntimeGetModuleInfo	D	Get Dictionary Module Major Version, Minor Version and Build Number	v11.0+
RuntimeGetVersionNum	D	Get version number from runtime engine	v11.0+

WebClient and Service Based Architecture

Function Name	Type	Description	Version
IsServiceMode	G	Check if in Service Based Architecture Mode	v14.0+
IsWebClient	G	Check if in Web Client mode	v12.0+
RuntimeGetClientType	D	Return Client Type from Runtime Engine Type: 1 = DEX_CLIENT_TYPE_STANDARD 2 = DEX_CLIENT_TYPE_WEB 0 = DEX_CLIENT_TYPE_NO_UI	v12.0+
RuntimeGetWebClientTrustLevel	D	Check the trust level on the Silverlight Web Client Level: 1 = TRUST_LEVEL_FULL 0 = TRUST_LEVEL_SANDBOXED	v12.0+
RuntimeWriteToWebClientLog	G	Write entries to the Web Client Logs	v12.0+

User Functions

Function Name	Type	Description	Version
syUserHasAccessToAllCompanies	G	Check if user has access to all companies in the system	v11.0+
syUserInRole	G	Check if user has SQL Roles, Roles: "sysadmin" = ROLE_SYSADMIN "securityadmin" = ROLE_SECURITYADMIN "dbcreator" = ROLE_DBCREATOR "db_owner" = ROLE_DBOWNER "db_accessadmin" = ROLE_DBACCESSADMIN "db_securityadmin" = ROLE_DBSECURITYADMIN "db_backupoperator" = ROLE_DBBACKUPOPERATOR	v11.0+
syUserIsDBO	G	Check if user has db_owner status	v11.0+
syUserIsPowerUser	G	Check if user is a POWERUSER for current company	v11.0+
syUserIsPowerUserForCompany	CG	Check if user is a POWERUSER for specified company	v11.0+

Encoding and Encryption

Function Name	Type	Description	Version
Hex	D	Convert an integer into a hexadecimal string using lower case letters and prefix, see UtilityConvertLongTo8CharHex	v11.0+
UtilityConvertLongTo8CharHex	D	Convert an integer into a hexadecimal string using uppercase letters and no prefix, see Hex	v11.0+
UtilityDecodeDateTimeString	D	Decode a string to datetime, See also UtilityEncodeDateTimeString	v11.0+
UtilityDecodeString	D	Decode a string, see also UtilityEncodeString	v11.0+
UtilityDecryptTableString	D	Decrypt a string stored encrypted in a table, see also UtilityEncryptTableString	v11.0+
UtilityEncodeDateTimeString	D	Encode a datetime to a string, See also UtilityDecodeDateTimeString	v11.0+
UtilityEncodeString	D	Encode a string, see also UtilityDecodeString	v11.0+
UtilityEncryptTableString	D	Encrypt a string to be stored encrypted in a table, see also UtilityDecryptTableString	v11.0+

Miscellaneous

Function Name	Type	Description	Version
GetNextNoteIndex	G	Get the next Note Index for the current company and increment the stored value	v11.0+
IsAlpha	D	Check if string contains only alphabetic characters	v11.0+
TimerSleep	D	Pause for the specified number of milliseconds	v11.0+

Legal Disclaimer

The information contained in this document represents the current view of Winthrop Development Consultants (Winthrop) on the issues discussed as of the date of publication. Because Winthrop must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Winthrop, and Winthrop cannot guarantee the accuracy of any information presented after the date of publication.

This document is for informational purposes only. WINTHROP MAKES NO WARRANTIES, EXPRESS OR IMPLIED, AS TO THE INFORMATION IN THIS DOCUMENT.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Winthrop Development Consultants.

Winthrop may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Winthrop, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

Copyright © 2014-2027, Winthrop Development Consultants. All rights reserved.

The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

** End of document - VSTMenus.doc - DM - 21 April 2026 **