

Menus for Visual Studio Tools for Microsoft Dynamics GP 2010 (build 9)

Introduction

Menus for Visual Studio® Tools for Microsoft Dynamics® GP 2010 is a Dexterity® add-on software tool which contains an Application Programming Interface (API) which can be called from Visual Studio development system using Visual Basic® or Visual C#® development systems to add their own menu items into menu structure for the Microsoft Dynamics GP.

Compatibility information

The tool is only available for Microsoft Dynamics GP 2010.

Contents of the Menus for Visual Studio Tools for Microsoft Dynamics GP 11.0 Archive

Below is a table of the files and folders included in the VSTM_1100.zip archive file:

| 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. |
| 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 |
| VSTMenusCSharp folder | Sample Visual C# project which add menus to Cards >> Sales menu |
| VSTMenusVB folder | Sample Visual Basic project which add menus to Cards >> Purchasing menu |

Installing Menus for Visual Studio Tools for Microsoft Dynamics GP 2010

1. From the VSTM_1100.zip archive, extract the VSTMenus.cnk, Application.MenusForVisualStudioTools.dll and Application.MenusForVisualStudioTools.xml files to the Microsoft Dynamics GP Application folder (default location: C:\Program Files\Microsoft Dynamics\GP).
2. Launch Microsoft Dynamics GP.
3. When the “New code must be included in the Dynamics.set dictionary. Do you wish to included new code now?” dialog is displayed, click Yes.

Installing Menus for Visual Studio Tools for Microsoft Dynamics GP 2010 sample projects

1. From the VSTM_1100.zip archive, extract the VSTMenuCSharp folder and/or VSTMenuVB 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 (VSTMenuCSharp.sln or VSTMenuVB.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 VSTMenuCSharp.dll or VSTMenuVB.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 Menus for Visual Studio Tools for Microsoft Dynamics GP 2010 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 Menus for Visual Studio Tools for Microsoft Dynamics GP 2010, 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 Menus for Visual Studio Tools (GP 2010 Syntax)

Next you need to add into the `Initialize()` script the commands to register for the two function events used by the tool. This syntax now uses the function events added in GP 2010.

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(VSTMCommandFormRegister);
    // Register Event to handle callbacks from menu entries
    MenusForVisualStudioTools.Functions.EventHandler.InvokeAfterOriginal += new
    EventHandlerFunction.InvokeEventHandler(VSTMCommandFormCallback);
}

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

In these samples the `VSTMCommandFormRegister()` script would contain the function calls to add the menu items and record their returned menu tags. The `VSTMCommandFormCallback()` script would then use the first parameter `e.inParam1` from `EventHandlerFunction.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 Visual Basic and Visual C# examples included in the archive for more information.

Registering Events for Menus for Visual Studio Tools (GP 10.0 Syntax still supported)

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

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

```
MenusForVisualStudioTools.Forms.VstmCommandForm.Api.RegisterButton.ClickAfterOriginal
```

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

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

```
MenusForVisualStudioTools.Forms.VstmCommandForm.Api.CmdId.ValidateAfterOriginal
```

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.Forms.VstmCommandForm.Api.RegisterButton.ClickAfterOriginal +=
    new EventHandler(VSTMCommandFormRegister);
    // Register Event to handle callbacks from menu entries
    MenusForVisualStudioTools.Forms.VstmCommandForm.Api.CmdId.ValidateAfterOriginal += new
    EventHandler(VSTMCommandFormCallback);
}
```

```
Sub Initialize() Implements IDexterityAddIn.Initialize
    ' Register Event to add menu entries
    AddHandler
    MenusForVisualStudioTools.Forms.VstmCommandForm.Api.RegisterButton.ClickAfterOriginal,
    AddressOf VSTMCommandFormRegister
    ' Register Event to handle callbacks from menu entries
    AddHandler
    MenusForVisualStudioTools.Forms.VstmCommandForm.Api.CmdId.ValidateAfterOriginal,
    AddressOf VSTMCommandFormCallback
End Sub
```

In these samples the `VSTMCommandFormRegister()` script would contain the function calls to add the menu items and record their returned menu tags. The `VSTMCommandFormCallback()` script would then use the `MenusForVisualStudioTools.Functions.Callback.Invoke()` 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 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 Menus for Visual Studio Tools for Microsoft Dynamics GP 2010:

| | |
|-------------------|---|
| 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. |

Limitations

Menus for Visual Studio Tools for Microsoft Dynamics GP 2010 has the following limitations:

- There are a maximum of 100 menu items shared among all of the Visual Studio Tools products using the 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 Menus for Visual Studio Tools for Microsoft Dynamics GP 2010 can be obtained through the standard Dynamics GP support channels. Information on contacting support can be found in the Support area of the MSDN Dynamics GP Developer Center.

<http://msdn.microsoft.com/en-us/dynamics/gp/bb468042.aspx>

For customers or partners on a Dynamics specific support plan, please use the following link from CustomerSource or PartnerSource to log a Visual Studio Tools case.

<https://mbs.microsoft.com/support>

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

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 menu 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. | | |

Name: RegisterWithSecurity

Description: This function is used to create menu entries with security inherited from a 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

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.

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 (Short) Menu Tag of Menu Entry
IN_Checked (Boolean) 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 (Short) Menu Tag of Menu Entry
IN_Hidden (Boolean) 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 (Short) Menu Tag of Menu Entry
IN_Disabled (Boolean) 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.

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 11.00.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.

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

This document is for informational purposes only. MICROSOFT 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 Microsoft Corporation.

Microsoft 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 Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

© 2010 Microsoft Corporation. All rights reserved. Microsoft, Dexterity, Microsoft Dynamics, Visual Basic, Visual C#, and Visual Studio are trademarks of the Microsoft group of companies.

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

** End of document - VSTMenus.doc - DM - 15 June 2010 **