

Даденова Гулхан Кенесбаевна
старший преподаватель Ташкентского
университета информационных технологий

***СОВРЕМЕННЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ
ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА И ПРОГРАММИРОВАНИЕ***

PROGRAMMING TECHNOLOGIES

Nowadays, computer technologies are developing. For this programming are servicing. We are using computer programs in every sphere. So, we can say that computer programming is the piece of our life. In this article we will write about some programming technologies.

Visual Basic for Applications

Automation is the Component Object Model (COM) technology that makes Microsoft Project programmable, which makes creating an integrated Microsoft Project solution possible. Automation (formerly called OLE Automation) includes the following features.

- Automation exposes features in a hierarchy of programmable objects Applications, dynamic-link libraries (DLLs), and ActiveX controls that support the appropriate Automation interfaces expose their features as a set of programmable objects. Any application or service that supports these interfaces is called a COM component. The set of programmable objects that a COM component exposes is organized into a set of hierarchical relationships that is called an object model, or a programming model.
- Automation allows applications to share features with other applications a COM component can be either an Automation server that shares its component objects with other applications or an Automation client that uses the component objects of other applications.

- Visual Basic for Applications provides features that make it easy to use Automation. When you use Microsoft Visual Basic for Applications (VBA) code, you can establish a reference to the programming model of an Automation server by using the References dialog box (Tools menu) in the Visual Basic Editor. The reference allows your solution to allocate memory efficiently to work with the objects that the Automation server exposes. The reference also allows the Visual Basic Editor to assist you when you enter VBA statements by automatically listing relevant objects, properties, and methods from the referenced programming model. And you can use the Object Browser command (View menu) to examine how the objects that are exposed by the Automation server are related to each other, and to find more information about how to program the objects.

Automation Servers and Automation Clients

Microsoft Project developers can take advantage of all Automation features when developing custom solutions. Microsoft Project can function as COM components that expose most of their features to other applications that support Automation as programmable objects. Microsoft Project serves as both Automation server and Automation client. Microsoft Project incorporates the VBA programming environment that makes working with Automation easy.

Programmable Applications

The Visual Basic for Applications programming language and the Visual Basic Editor are incorporated into Microsoft Project to make it programmable.

In addition, many other applications incorporate the Visual Basic for Applications programming environment and are written to expose their functionality to VBA programmers through Automation interfaces. This means that developers can use Visual Basic for Applications to create solutions not only for Microsoft Project, but also for other applications that support VBA and Automation.

Programming across applications

Visual Basic for Applications and Automation allow you to program Microsoft Project. Developers can also use Automation to run other applications from within a client application. For example, a VBA solution developed in Microsoft Project can

run a hidden instance of Microsoft Excel to perform mathematical and analytical operations on Microsoft Project data.

Working with programmable services

Automation also allows developers to use Visual Basic for Applications code to work with programming models that function as freestanding services and are independent of individual applications. For example, Microsoft Project installs the Microsoft Data Access Components (MDAC), which consists of OLE DB components that provide low-level access to a variety of data sources and components that make up the ActiveX Data Objects (ADO) programming model. Because the ADO programming model supports Automation, you can use ADO from Visual Basic for Applications code running in Microsoft Project to gain access to and work with any data source that is available through an OLE DB provider or an Open Database Connectivity (ODBC) driver.

New programming features

Extensive changes have been made to the Microsoft Project object model and events to support new and improved features in the application. Many objects, properties, and methods have been replaced. To provide backward compatibility, most of the replaced components have been hidden rather than removed. This means that they do not appear in the object browser by default, however, old code that uses the hidden components will still work correctly without modification. When you write new code, however, you should use the new objects, properties, and methods.

New objects

The following objects have been added to the Microsoft Project 2002 object model:

- Event Info parameter added to events to provide cancel information.
- Filter and Filters to provide access to filters in Microsoft Project.
- Profile and Profiles to provide access to profiles in Microsoft Project.
- Table and Tables to provide access to tables and table formats in Microsoft Project.

- Table Field and Table Fields to provide access to table fields and their formatting in Microsoft Project.

- View and Views to provide access to views in Microsoft Project.

- View Combination and Views Combination to provide access to combination views in Microsoft Project.

- View Single and Views Single to provide access to single-pane views in Microsoft Project.

New events, properties, and methods

Many new events, properties, and methods have been added for numerous objects in Microsoft Project. For a complete list of all new events, properties, and methods, see Visual Basic for Applications Help.

Hidden properties and methods

Most of the hidden properties and methods have been replaced by new functionality in Microsoft Project. These properties and methods are supported only for backward compatibility; for new code, you should use the replacement functionality provided in Microsoft Project. Visual Basic for Applications in Microsoft Project contains the following hidden properties and methods for visible application objects for which no replacements are available:

- Help Quick Preview;

- Mail Update Project.

Methods with new, changed, or deleted arguments

Visual Basic for Applications in Microsoft Project provides several methods for which arguments have been added, changed, or removed. Any code that uses these items may produce errors and should be modified accordingly. For a complete list of methods with new, changed, or deleted arguments, see Visual Basic for Applications Help.

For more information about the Visual Basic for Applications programming language and applications that support it, see the Microsoft Visual Basic for Applications Web site.

For more information about developing custom solutions by using Office applications and the Visual Basic for Applications programming language, see the Office XP Developer Web site, and the Microsoft Office /Visual Basic Programmer's Guide.

Instead of using macros and Visual Basic for Applications scripts to customize Microsoft Project, you can create a custom Microsoft Project solution by developing and distributing an add-in. An add-in extends an application by adding functionality that is not in the core product. For example, an add-in might add new menu commands or toolbar buttons that display custom forms to add new features to an application.

Microsoft Project supports a COM add-in architecture that allows developers to create a single add-in that can run in any application. VB developers can create COM add-ins by using the Building COM Add-ins for Office Applications article on MSDN.

COM add-ins can be created by developers who are using one of the following programming languages:

- Microsoft Visual Basic versions 5.0 or later
- Microsoft Visual C++
- Microsoft Visual J++
- Any language that can create COM components

This wide support for COM add-ins means that developers can now use multiple development environments to create custom solutions in Microsoft Project. The only requirement to connect COM add-ins to Microsoft Project is for the add-in to implement the `IDExtensibility2` interface. COM add-ins can be loaded when the host application starts, or they can be loaded on demand. Support has also been added to allow the On Action property of a custom command bar button to load a COM add-in.

Microsoft Project provides the following add-ins:

- Adjust Dates
- Analyze Timescaled Data

- PERT Analysis

The following COM add-ins for Microsoft Project are available at the Microsoft Download Center:

- Compare Project Versions
- Database Upgrade Utility
- Euro Currency Converter
- Visio WBS Chart Wizard
- XML Reporting Wizard

Using the following Microsoft Project object model properties on the Project object, you can define a custom Project Guide for a project:

- **Project Guide Use Default Function all Layout Page** This is a Boolean property. If set to True, the default HTML page, referenced as gbui://mainpage.htm, defines the layout and functionality of the user interface for the Project Guide. If set to False, the HTML page referenced by the property Project Guide Functional Layout Page is used.

- **Project Guide Functional Layout Page** This is a Read/write string property on the Project object. It returns or sets the HTML page that should be used to define the layout and functionality of the Project Guide for the specified project. Note that this property is used only if Project Guide Use Default Functional Layout Page is set to False.

- **Project Guide Use Default Content** This is a Boolean property. If True, the default XML document, referenced as gbui://gbui.xml, defines the content for the Project Guide. If False, the XML document referenced by the property Project Guide Content is used.

- **Project Guide Content** This is a Read/write string property on the Project object. It returns or sets the XML document that should be used to define the content of the Project Guide for the specified project. This property is used only if Project Guide Use Default Content is set to False.

Each project can have its own Project Guide with different layout, functionality, and content. Potentially, a company or solution provider could create

one Project Guide for finance projects, another for engineering projects, and so on. There could also be different Project Guides based on the role of the user. Project Guides based on user roles would involve checking a user's role when he or she opens a project and dynamically changing the properties that define the Project Guide for that project. When the properties defining the Project Guide for a project are changed, the HTML pages displayed in that project are refreshed, so the new Project Guide is displayed immediately. You do not have to close and open the project again for the changes to take effect.

Most organizations will typically only customize the content properties of the Project Guide, Project Guide Use Default Content, and Project Guide Content. The default functionality and layout page (defined by the properties Project Guide Use Default Functional Layout Page and Project Guide Functional Layout Page) should work for most custom Project Guides. Only consider changing the functionality and layout page to change the default layout of the Project Guide, for instance, to have the side pane displayed on the right side of the view area instead of the left, or to tile the side pane horizontally with the Project window instead of vertically. Some of the default behavior of the Project Guide, such as the way it responds to view changes, can also be customized. The default functionality and layout page (gbui://mainpage.htm) has been written generically, so it should work for most custom scenarios.

The properties defining the Project Guide are displayed in the Interface tab of the Options dialog box (Tools menu) in Microsoft Project. A Microsoft Project user accesses this tab to change the Project Guide for a project or to set the default Project Guide for all projects (using the Set as Default button). Instead of allowing individual users to change Project Guide settings, it is expected that organizations will set up the Project Guide for the entire organization or group so that these changes apply to all users.

For more information about custom Project Guide solutions, see the Microsoft Project Project Guide Architecture and Extensibility or Introduction to Customizing the Microsoft Project Guide articles on MSDN.

You can use the Microsoft Project OLE DB providers to access project data. If the data is stored in a Microsoft Access or Microsoft SQL Server™ database, you can also use the Jet 4.0 OLE DB provider. If the data is stored in a SQL Server or Microsoft Data Engine (MSDE) database, it can be accessed using the SQL OLE DB provider.

Data access pages in Microsoft Access provide a versatile and powerful method for generating reports using data from Microsoft Project. They also provide a convenient way to view the Microsoft Project OLE DB table structure. For more information on data access pages, see Microsoft Access Help.

Dadenova Gulxan Kenesbaevna
Uzbekistan, Tashkent

REFERENCES

1. Peter Seibel “Coders at work”
2. Andrew Hunt, David Thomas “The Pragmatic Programmer”
3. Gayle McDowell “Cracking the Coding interview”
4. Cormen, Leiserson “Algorithms”