

Archived content. No warranty is made as to technical accuracy. Content may contain URLs that were valid when originally published, but now link to sites or pages that no longer exist.

Visual Basic 4.0 Technical Articles

Mapping the Microsoft Project for Windows 95 Object Library: MSProject 4.1

Ken Lassesen

Microsoft Developer Network Technology Group

January 12, 1996

Abstract

This article presents a graphic map of the Microsoft® Project for Windows® 95 object, showing its properties, methods, and child objects. This server is available from Visual C++® and many products based on Visual Basic® for Applications, including Microsoft Excel version 5.0 and later, Microsoft Access 95, Visual Basic version 4.0, and Visual C++ version 4.0.

Microsoft Project for Windows 95 Object

The Microsoft® Project® for Windows® 95 object (MSProject) enables the developer to manipulate Microsoft Project project files from other applications.

This object is accessible from Visual C++® version 4.0 and many products that contain Visual Basic® for Applications, including Microsoft Excel version 5.0 and later, Microsoft Access 95, Visual Basic version 4.0, and Microsoft Project. For further information on using this object, see the documentation for these products and the articles in the bibliography.

Figure 1 shows the relationship between objects as described by the Object Viewer and online Help files. A map serves as a good learning aid and as a quick reference for developers. After producing a map, I found that it was easy to work with the object: I just post the appropriate map on my wall for quick reference, which is a lot faster than clicking objects in a Help file one by one to discover their properties and methods.

Because Figure 1 is difficult to read online, I have included the two most common graphics formats—encapsulated PostScript™ (.EPS) and Windows metafile (.WMF)—as well as a copy of my original ShapeWare™ Visio™ version 4.0 file (.VSD). The first two formats can be opened and the maps printed across multiple pages using Microsoft Excel or a commercial graphics application such as Adobe™ PageMaker™, CorelDRAW®, or Microsoft Publisher. The original Visio file is included for those who wish to modify the diagram.

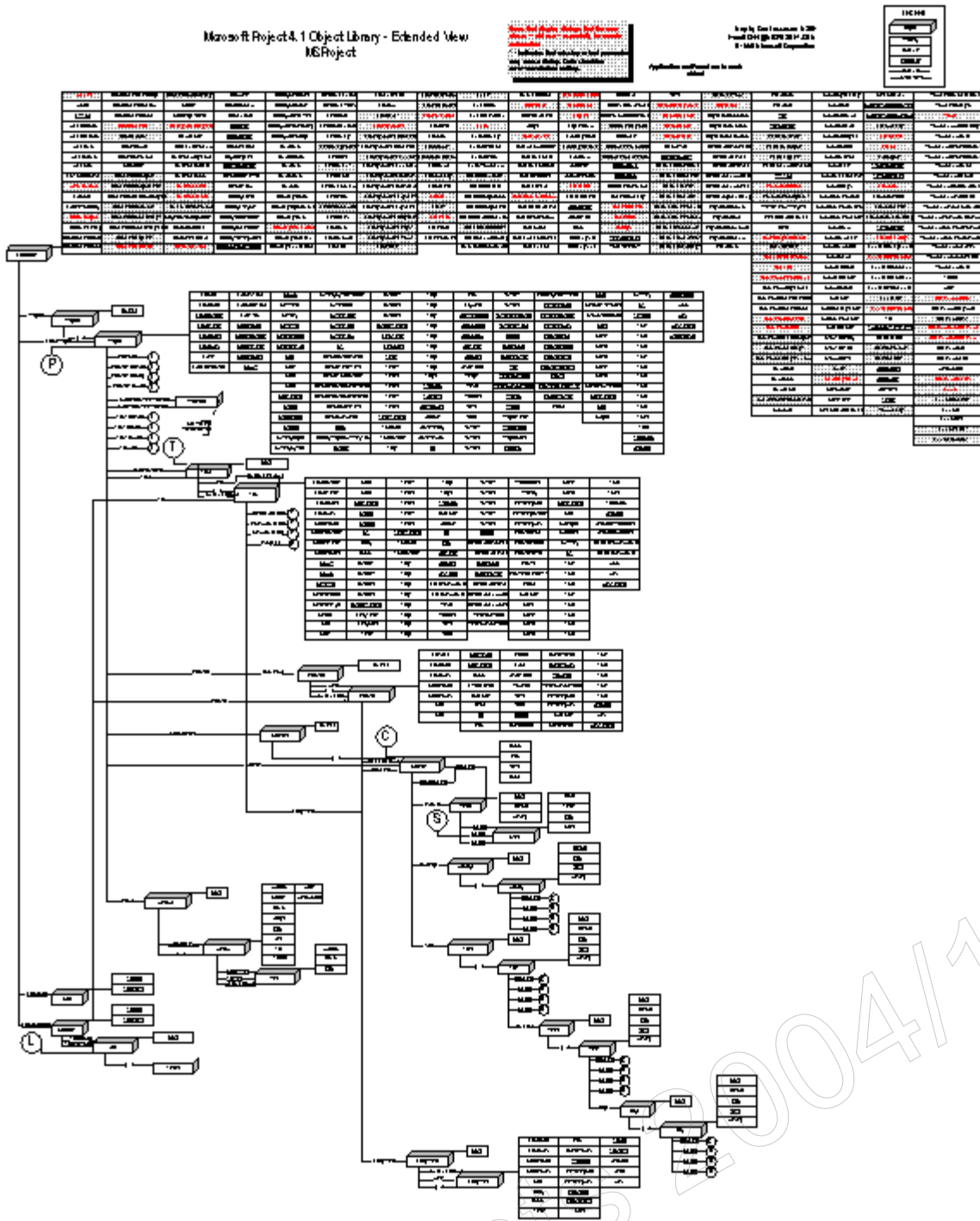


Figure 1. Map of the Microsoft Project for Windows 95 object library

True Confessions

The developers of the Microsoft Project OLE Automation server were faced with the problem of backward compatibility with the version 3.0 macro language. (As the Danish programmer Hamlet put it, "To break or not to break; that is the question!") The decision to support backward compatibility meant including some unusual

methods that may cause problems. These methods are shown in the diagram and are discouraged for new development. The `Application.About` method, which causes the About dialog box to be displayed, is an example of one such method. This dialog box hangs the controller applications until the user dismisses it. The following problems can occur:

- The dialog box becomes hidden behind a dialog box of another application.
- The controller displays a time-out message if the user doesn't dismiss the dialog box quickly enough.
- The dialog box appears on a screen at an unknown location if Remote OLE Automation is used. The user must then gain physical access and logon permission to dismiss the dialog box.

The `Application.About` method, included to support backward compatibility, violates the rule that an OLE Automation server never be put into a state where it is not possible to close the server using only OLE Automation.

The Microsoft Project for Windows 95 OLE Automation server type library is inconsistent with the documentation in several cases. Some read-only properties are not marked read only in the type library. For example, many collections have a `Count` element. The `Count` element cannot be set, although a developer can write code that will compile. For example:

```
Dim x As MSPProject.Projects
x.Count = 5 'Should produce error message at compile time but does not
```

Many of the elements referred to as `Application.properties` in the documentation are defined in the type library as being `Application.methods` when examined by the OLE 2 Object View application. I underlined in Figure 1 all of the elements that are documented to be properties and are defined to be methods. These items appear as Property instead of `Property` or Property. Similarly, a few elements are documented as methods but are defined as properties; these elements appear as defined rather than as documented.

One element defined in the documentation does not exist in the type library—`Application.Left`. However, because `MSPProject.Application` is late bound in the type library, the following code works:

```
Dim appMSPProject As MSPProject.Application
Set appMSPProject = CreateObject("MSPProject.Application")
appMSPProject.Top = 100
appMSPProject.Left = 0
```

Object Definitions

The objects and collections in Figure 1 are defined in Table 1.

Table 1. Object and Collection Definitions

Object or Collection	Definition
Application object	Represents the active Microsoft Project application. The Application object is the parent of the Cell, Project, Selection, and Window objects and of the Projects and Windows collections.
Projects collection	Contains all open projects.
Project object	Represents one open project. The Project object is the parent of the Tasks, Resources, and Calendars collections.
Tasks collection	Contains all of the tasks in a project.
Task object	Represents a task.
Resources collection	Contains all of the resources for a task or a project.
Resource object	Represents a single resource.
Calendars collection	Contains all of the calendars for a project.
Calendar object	Represents a calendar for a project or a resource.
Period object	Represents a period in a calendar.
Shift object	Represents a work shift for a day, month, period, weekday, or year.
Weekdays collection	Contains the weekdays in a calendar.
Weekday object	Represents a weekday in a calendar.

Years collection	Contains all of the years in a calendar.
Year object	Represents a year in a calendar.
Months collection	Contains all of the months in a year.
Month object	Represents a month of a year.
Days collection	Contains all of the days in a month.
Day collection	Represents a day of a month.
Assignments collection	Contains all of the assignments for a task or a resource.
Assignment object	Represents an assignment for a task or a resource.
Windows collection	Contains all of the windows open in an application or project.
Window object	Represents a window open in an application or project.
Properties object	Is a DocumentProperties object as described in "Mapping the Microsoft Office 95 Object Library: MicrosoftOffice."
Cell object	Represents the active cell of the project.
Selection object	Represents the active selection.
List collection	Contains strings or numbers, field names, resource filters, resource tables, resource views, task filters, task tables, or task views.

Bibliography

Lassesen, Ken. "Using Microsoft OLE Automation Servers to Develop Solutions." (MSDN Library, Technical Articles)

Lassesen, Ken. "[Mapping the Microsoft Office 95 Object Library: MicrosoftOffice.](#)" (MSDN Library, Technical Articles)

"Microsoft Project Visual Basic Reference" (VBA_PJ.HLP) is installed with Visual Basic 4.0 and Microsoft Project for Windows 95.

[Manage Your Profile](#) | [Legal](#) | [Contact Us](#) | [MSDN Flash Newsletter](#)

© 2007 Microsoft Corporation. All rights reserved. [Terms of Use](#) | [Trademarks](#) | [Privacy Statement](#)



Requirements 2004/11/18