

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 Schedule+ 7.0 Object Library: SPL 7.0

Ken Lassesen

Microsoft Developer Network Technology Group

December 18, 1995

Abstract

This article presents a graphic map of the Microsoft® Schedule+ 7.0 Object Library (SPL), showing its properties, methods, and child objects. This type library manipulates the Schedule+ OLE Automation server. Any Microsoft Visual Basic®-based language (Access Basic, Visual Basic, and Visual Basic for Applications) and Visual C++® can access the SPL server.

The Microsoft Schedule+ 95 Type Library

The Microsoft® Schedule+ 95 Object Library (SPL) allows the developer to manipulate the Schedule+ OLE Automation server. This server can be viewed in three different ways. The view described by the type library is outlined in this article; the other views are described in the other two mapping articles listed in the bibliography.

The SPL assumes that the Microsoft Exchange Server Beta 2 or later version of Schedule+ is installed. The SPL will not bind with the earlier versions of Schedule+ (including the version shipped with Microsoft Office 95). Whatever version of Schedule+ you are using, you should include the SPL because it provides constants for Schedule+ OLE Automation and templates for methods and properties in the Object Browser.

The SPL does not show Schedule+ tables such as Contacts, Appointments, Tasks, or Events. You can get these tables through the Schedule object's Properties method. Note that you cannot use the SPL as the sole source of information (as can be done with other type libraries); you must refer to the Schedule+ Software Development Kit (SDK) documentation and other sources. My other extended mapping articles on the Schedule+ server, listed in the bibliography, are a good starting point. For a general introduction to OLE Automation, see "Using Microsoft OLE Automation Servers to Develop Solutions." For an introduction to Schedule+ OLE Automation, see "Schedule+ OLE Automation" in the Microsoft Developer Network News and "An Extended Introduction to Schedule+ OLE Automation Programming."

Figure 1 shows the relationship between objects as described by the Object Browser in Microsoft Visual Basic®. A map is a good learning aid and a quick reference when developing an application. After producing a map, I found that it was easy to work with the objects: I just post the three appropriate extended maps for Schedule+ 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 the figure 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 the original Shapeware® Visio™ version 4.0 file (.VSD). The first two formats can be opened and the figure 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 easily.

Schedule Plus 95 Type Library- Extended View SPL 7.0

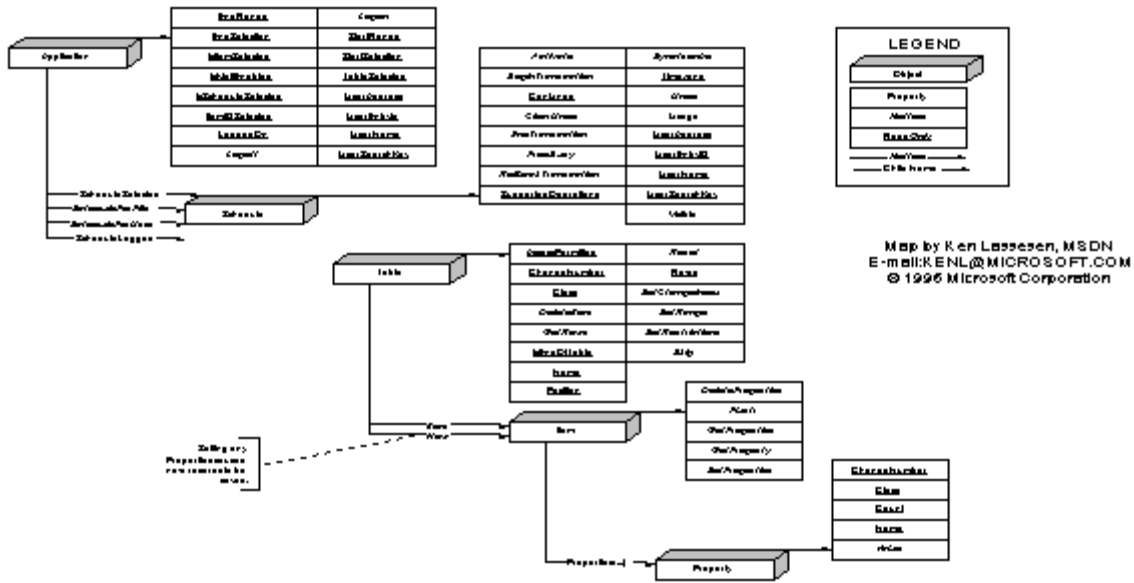


Figure 1. The Microsoft Schedule+ 7.0 Object Library

Following are some of the characteristics of Schedule+ OLE Automation that are different from other OLE Automation servers.

- Schedule+ is a single-document interface (SDI) and not a multiple-document interface (MDI). The Visible property applies to each document (Schedule object) and not to the application.
- The Application object does not have a graphical user interface (GUI) equivalent.
- There are no localized versions of the SPL, nor is Schedule+ OLE Automation localized. Even if you are using a localized version of Schedule+, you must use the English OLE Automation words (with or without a type library).
- Properties that are not initialized may be reported as error 450, "Wrong number of arguments or invalid property assignment", or as an "OLE Automation error". The following code shows the error:

```
Dim SPlusApplication as Object
On Error Resume Next
Set SPlusApplication = CreateObject("SchedulePlus.Application")
Debug.Print SPlusApplication.UserName 'Not logged on
Debug.Print Err.Number,Err.Description
'Reports: -2147219351 OLE Automation error
SPlusApplication.Logon
Debug.Print SPlusApplication.UserName 'Logged on
'Reports KenL
```

- Item objects have many Property objects. The Property objects that are initialized are enumerated by the Properties method. The Properties method is overloaded.

- Item.Properties returns a count of the number of initialized properties.
- Item.Properties(idx%) returns a Property object.

The following routine illustrates the use of Table, Item, and Property objects. An Item object (or Schedule object) is passed to the routine, which then enumerates the Property objects, Table objects, and the first Item object of each Table object. This process is done by recursion to each Item object found. Three tables (Appointments, Events, and Tasks) are merged Table objects that require special handling. A merged Table object requires the object to be set to a range before all of the properties of the Table object are valid.

```

Sub DebugItem(AnItem As Object, Optional indent)
Dim i%, j%, ind%
If IsMissing(indent) Then
    ind% = 0
Else
    ind% = Val(indent)
End If
If ind < 1 Then ind = 1
If TypeName(AnItem) <> "Item" Then Debug.Print "Not an 'Item'": Exit Sub
For i% = 0 To AnItem.Properties - 1
    Debug.Print Space(ind); AnItem.Properties(i%).Name
    If TypeName(AnItem.Properties(i%)) = "Table" Then
        If AnItem.Properties(i%).Name = "Appointments" _
            Or AnItem.Properties(i%).Name = "Events" _
            Or AnItem.Properties(i%).Name = "Tasks" Then
            AnItem.Properties(i%).SetRange Now - 365, Now + 365
        End If
        On Error Resume Next
        AnItem.Reset
        Debug.Print Space(ind); "Table with "; _
            AnItem.Properties(i%).Rows; " rows"
        If AnItem.Properties(i%).Rows = 0 Then
            Else
                On Error GoTo 0
                Debug.Print Space(ind); String(20, "-")
                DebugItem AnItem.Properties(i%).Item, (ind% + 2)
            End If
            On Error GoTo 0
        Else
            If AnItem.Properties(i%).Count = 1 Then
                Debug.Print Space(ind); _
                    "="; AnItem.Properties(i%).Value
            Else
                For j% = 0 To AnItem.Properties(i%).Count - 1
                    Debug.Print Space(ind); "="; _
                        AnItem.Properties(i%).Value(j%)
                Next j%
            End If
        End If
    Next i%
End Sub

```

Value is the default property of the Property object. The properties shown in Figure 1 are data members on the objects and not Property objects. When you use SPL, Property objects may be obtained only by using the Properties method. The TypeName function can be used to illustrate this confusion, as shown in the examples in Table 1.

Table 1. Examples of the Confusion Possible with the Properties Property

Command	Reports
Schedule.properties(1)	11/30/95 8:32:02 PM
TypeName(Schedule.properties(1))	Property

```
Schedule.properties(1).value          11/30/95 8:32:02 PM
TypeName(Schedule.properties(1).value)  Date
Schedule.properties(1).class          7 (SPTY_DATETIME)
```

The objects in Figure 1 are defined in Table 2. The objects are listed in the same sequence as they appear on the map.

Table 2. Object Definitions

Object	Definition
Application object	Reports information about the SDI application.
Schedule object	Represents one schedule in the Application object. This is also an Item object.
Table object	Represents a table of Item records storing information. It is not a collection.
Item object	Represents an object that may be a Property or a Table.
Property object	Represents information about a characteristic of an object. Multiple values may be stored in one Property object (see Count to determine number).

Bibliography

Lassenen, Ken. "An Extended Introduction to Schedule+ OLE Automation Programming." (MSDN Library, Technical Articles)

Lassenen, Ken. "[Mapping the Schedule+ OLE Automation Server: Internal Objects.](#)" (MSDN Library, Technical Articles)

Lassenen, Ken. "[Mapping the Schedule+ OLE Automation Server: Programming Model.](#)" (MSDN Library, Technical Articles)

Lassenen, Ken. "Schedule+ OLE Automation." Developer Network News 5 (January/February 1996) (MSDN Library, Periodicals, Microsoft Developer Network News, 1996 Volume 5)

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

Microsoft Exchange Server SDK. Microsoft Schedule+ Programmer's Reference. (MSDN Library, Platform, SDK, and DDK Documentation)

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

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

