Visual Basic 4.0 Technical Articles

# Mapping the Schedule+ OLE Automation Server: Internal Objects

Ken Lassesen
Microsoft Developer Network Technology Group

December 14, 1995

## Abstract

This article consists primarily of a graphic map of the internal objects of the Microsoft® Schedule+ OLE Automation server, showing its properties, methods, and child objects. This is the third in a series of extended maps describing different views of the Schedule+ server. This object manipulates the completed Schedule+ OLE Automation server. Any Visual Basic®–based language (Access Basic, Visual Basic, and Visual Basic for Applications) and Visual C++™ can access the Schedule+ OLE Automation server.

## The Microsoft Schedule+ OLE Automation Server: Internal Objects

The map of the internal objects of the Microsoft® Schedule+ OLE Automation server describes the complete Schedule+ OLE Automation server. SPlus, my name for the view of these internal objects, includes the information visible to the user through the Schedule+ graphical user interface (GUI) and information not visible through the GUI. SPlus illustrates how the developer may use child tables of existing items. This server implements recursive objects that are not available in most OLE Automation servers.

SPlus shows that many objects (Access, Alarm, Appointment, Event, Task, Contact, Project, DeletedItem, and so on) are not separate object types; rather, these objects are the same Item objects. These Item objects have the same set of properties and the same set of methods. The difference between them is the value of the Name Property object. The Schedule object is an exception because it is a subclass of the Item object. The Schedule object has all of the methods and properties of the Item object by name, though some of the Property objects have been replaced with a data item. The Schedule object also has additional properties and methods!

Figure 1 shows the relationship between objects as discovered in conversations with the Schedule+ developers and some traditional hacking of the Schedule+ OLE Automation server. 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 Schedule+: I just posted 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 Figure 1 is difficult to read online, I have included the two most common graphics formats—encapsulated PostScript™ (.EPS) and Microsoft Windows® metafile (.WMF)—as well as a copy of my original Shapeware® Visio™ version 4.0 file (.VSD). The first two formats can be printed across multiple pages using any of the commercial graphics applications—such as Adobe™ PageMaker™, Corel® Draw, or Microsoft Publisher—or using Microsoft Excel. The original Visio file is included for those who have a copy of Visio and want to modify the diagram easily. For further information, see the bibliography at the end of this article.

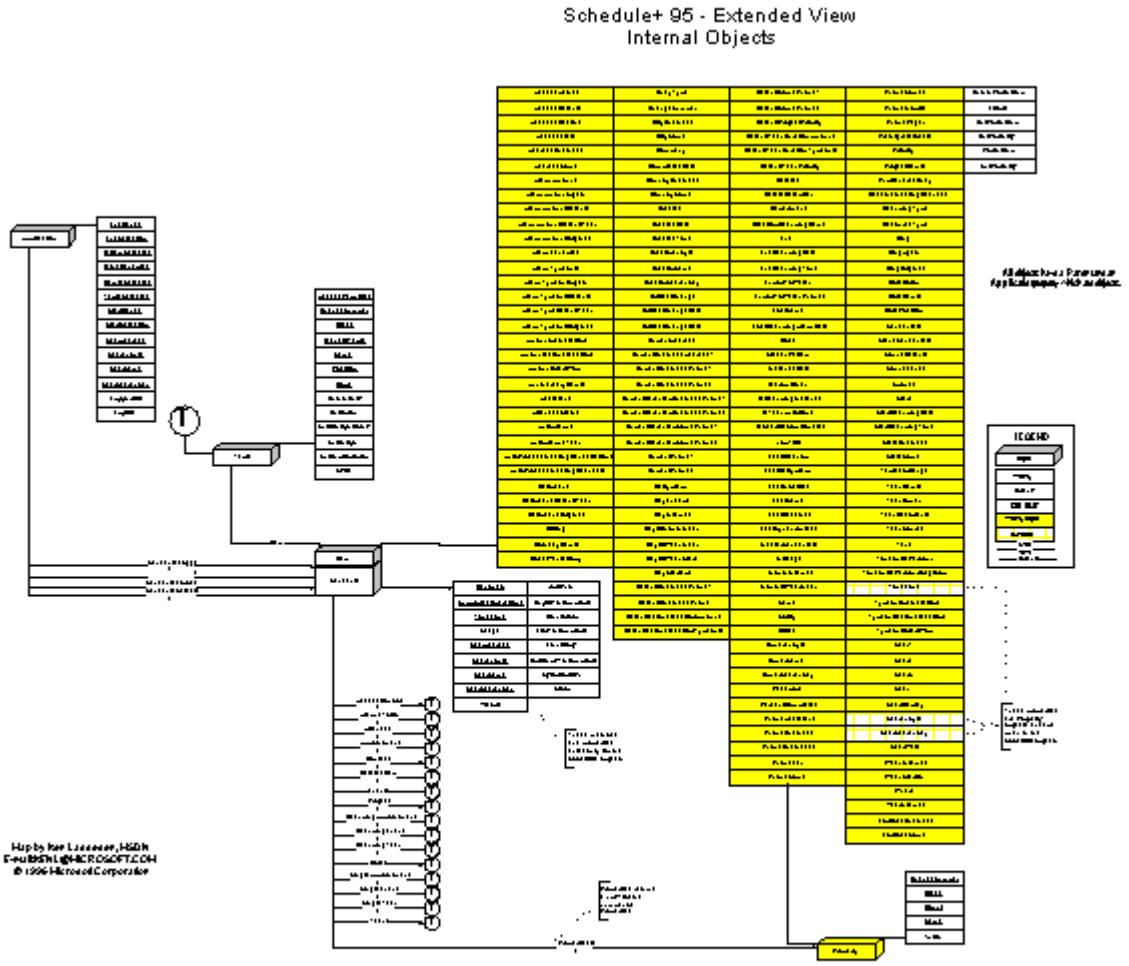


Figure 1. The Microsoft Schedule+ OLE Automation Server: Internal Objects

Some characteristics of the Schedule+ OLE Automation server are different from the common OLE Automation servers. SPlus illustrates the extensibility of this server and offers a different view than those presented in my adjunct articles, "Mapping the Schedule+ 7.0 Object Library: SPL 7.0" and "Mapping the Schedule+ OLE Automation Server: Programming Model," which are available in the MSDN Library. The key extensibility features are listed below.

- All of the properties on an Item may be used, regardless of the name of the item. For example, a DeletedItem object in the Programming Model extended map has no Property objects listed. In reality, a DeletedItem object has all of the properties of an Item object available. This is not intuitive, so a simple example may better explain this concept.

```
Debug.Print DeletedItems.rows
 5
While Not DeletedItems.IsEndofTable
   Debug.print DeletedItems.item.properties
   DeletedItems.skip
Wend
  6
```

```
10
 8
11
10
```

- Because the Schedule object is an Item object, many of the Table objects are available on any Item object. For example:

```
Debug.Print DeletedItems.Item.Contacts.rows,DeletedItems.Item.Projects.rows
 0              0
```

These features enable the Schedule+ OLE Automation server to do complex activities. Consider the following case study illustrated in Figure 2.

I have a bunch of people (Contact objects) who I am working with on different software projects (Project objects). Each software project is made up of different job types (Role objects), for example, programmer, analyst, writer, and tester. Each software project also consists of a series of jobs (also Project objects) done by one or more persons (Contact objects).

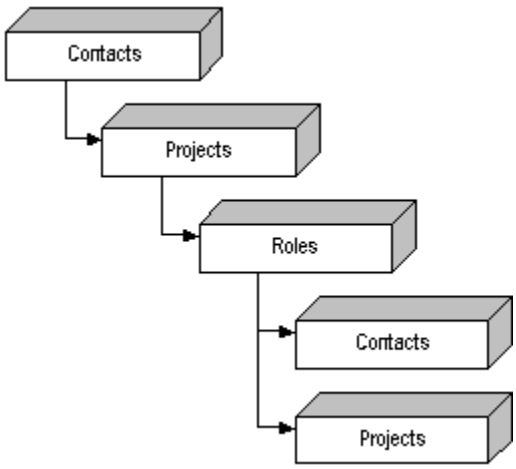


Figure 2. A sample of recursive Schedule+ tables

Many developers who look at the various extended maps would start to look for ways to place the ItemID value of each Project Item object in various Contact properties and hope they could avoid recursive relationships if more functionality is needed!

I take a different approach because I understand the power of this server's design. I would enter the information exactly as described above. The code sample below illustrates how I would do this.

```
Dim NewItem As Object
Dim NewProject As Object
Dim NewContact As Object
Dim NewRoles As Object

Set NewItem = Contacts.New
NewItem.LastName = "Gates"
NewItem.FirstName = "Bill"
NewItem.Flush

Set NewProject = NewItem.Projects.New
NewProject.Text = "Visual Basic for the Internet"
```

```
NewProject.Flush

Set NewRoles = NewProject.Roles.New
NewRoles.JobTitle = "Writer"
NewRoles.Flush

Set NewContact = NewRoles.Contacts.New
NewContact.LastName = "Lassesen"
NewContact.FirstName = "Ken M"
NewContact.Flush

Set NewProject = NewRoles.Projects.New
NewProject.JobTitle = "MSDN Library Articles"
NewProject.Flush
```

The code above shows that I can add items to tables that are part of items. Tables that are part of a merged table structure do not support the creation of new Item objects, except for the Schedule object. The support for the New method on the various tables is shown in Table 1.

Table 1. Tables Support for the New Method

| New (Valid for Schedule object only) | New (Valid for any Item object) |
|---|---|
| Appointments | AccessControls |
| SingleAppointments | AlarmsToRing |
| RecurringAppointments | Attendees |
| Tasks | Contacts |
| SingleTasks | Exceptions |
| RecurringTasks | Projects |
| Events | Roles |
| SingleEvents | |
| RecurringEvents | |

- The GetProperty and GetProperties methods do not return Property objects; rather, they return Value properties of the Property object in a two-dimension variant array.

These examples show that the Schedule+ OLE Automation server is quite different from most servers. These differences make it a powerful and flexible tool in the Microsoft Solutions Development Kit.

## Object Definitions

The objects and collections in Figure 1 are defined in Table 2. These objects are listed in the same sequence as they appear in the map. The Roles table contains the Role object that is used to support the Covey Seven Habits tool included with the Microsoft Exchange Server Software Development Kit (SDK). Use of the Role object may result in future compatibility problems.

Table 2. Object Definitions

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

## Bibliography

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

Lassesen, Ken. "Mapping the Schedule+ 7.0 Object Library: SPL 7.0." (MSDN Library, Technical Articles)

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

Lassesen, Ken. "Schedule+ OLE Automation." Developer Network News, January/February 1996, Number 1. (MSDN Library, Periodicals, Microsoft Developer Network News)

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

Microsoft Exchange Server SDK. Microsoft Schedule+ Programmer's Guide. (MSDN Library, Product Documentation)