



- ▶ OLE Automation and COM
- ▶ Writing an OLE Server

## Writing an OLE Server

The following steps are required to create an OLE Automation Server in Dyalog APL/W:

1. Create a workspace containing a namespace. This namespace will represent an OLE Object and must contain at least one monadic result returning function.
2. Tell Dyalog APL that you wish to register the namespace as an OLE object by changing its Type to OLEServer. You do this by executing `⊞WC` inside the namespace.
3. Tell Dyalog APL about the functions and variables you wish your object to expose as methods and properties. You can do this using the COM Property pages or by invoking the `SetMethodInfo`, `SetPropertyInfo` methods.
4. )Save the workspace. At this point, Dyalog APL automatically updates the Windows registry and type libraries with all of the information needed to make your object accessible by COM.

Naturally, the namespace may contain as many functions and variables as you want to provide the functionality you require.

The OLEServer namespace itself corresponds to a COM object. Each function exported by the namespace represents a method and each function a property of that object. Non exported functions and variables, sub-namespaces and defined operators may be used internally, but are not available directly to an OLE Automation client.

## In-Process and Out-of-Process OLE Servers

OLE Servers created by Dyalog were be *out-of-process* OLE Servers that run in a separate process from the client application, or *in-process* OLE Servers that run in the same address space as the client application.

The main **advantage** of an *in-process* OLE Server is that communication between the client application and the OLE Server is fast. Communication between clients and *out-of-process* OLE Servers has to go through a separate OLE layer in Windows that incurs a certain overhead. However, if in *your* application, the amount of computation swamps the extra communication overhead, you may not necessarily see any significant performance gains from an in-process server.

Another **advantage** is that *in-process* OLE Servers are simpler to administer and simpler to install.

The main **disadvantages** of *in-process* OLE Servers is that there can only be one client per server and they do not support DCOM.

### Dyalog Ltd

South Barn  
Minchens Court  
Minchens Lane  
Bramley  
Hampshire  
RG26 5BH  
United Kingdom

**Phone:**

+ 44 (0) 1256 830 030

**Fax:**

+ 44 (0) 1256 830 031

**e-mail:**

sales@dyalog.com



- ▶ OLE Automation and COM
- ▶ Writing an OLE Server

## Creating an in-process OLE Server

An in-process OLE Server is implemented as a Dynamic Link Library (DLL). To create an *in-process* OLE Server, you establish a workspace containing one or more OLEServer namespaces, in the usual way, then from the File menu, select **Make OCX/DLL....**

Alternatively, check the File menu option **Make OCX/DLL on Save** and use `)SAVE` as you would for an out-of-process server.

When you make an in-process OLE Server, Dyalog saves your workspace as a DLL. The name of the file is the same as the name of your workspace, with the extension .DLL. Your OLE Server(s) contained in the workspace are also registered in the normal way.

## Using an in-process OLE Server

When a client application requests an instance of an OLEServer object, Windows loads your DLL into the client's own address space. Then, as your DLL initialises, it loads DYALOG.DLL (Also see below), the dynamic link library version of Dyalog. So both your DLL and the DYALOG.DLL are loaded into the client application. If the same client loads a second instance of the same OLEServer, or another OLEServer contained in the same workspace, no additional code is loaded.

However, if another client requests an instance of your in-process OLEServer object, it too loads your DLL and DYALOG.DLL into its own address space.

## Registering an in-process OLE Server

You can register and unregister an in-process OLE Server using the Windows program REGSVR32.EXE. This program normally resides in Windows\System and is usually on your path.

To **register** an in-process OLE Server named `c:\dyalog90\myapp.dll`:

```
regsvr32 c:\dyalog90\myapp.dll
```

To **unregister** the same OLE Server:

```
regsvr32 /u c:\dyalog90\myapp.dll
```

## Licensing and Distribution of DYALOG.DLL

Subject to the conditions of the *Dyalog for Windows RUN-TIME SYSTEM LICENCE AMENDMENT to DYADIC SOFTWARE LICENCE AGREEMENT*, DYALOG.DLL may be distributed as part of an application.

### Dyalog Ltd

South Barn  
Minchens Court  
Minchens Lane  
Bramley  
Hampshire  
RG26 5BH  
United Kingdom

**Phone:**

+ 44 (0) 1256 830 030

**Fax:**

+ 44 (0) 1256 830 031

**e-mail:**

sales@dyalog.com