

Visual M Configuration

Visual M Server systems are Open M systems that can run one or more Visual M Server processes. The Visual M Server allows Visual M client applications to obtain data or run routines located on the Open M system. A Visual M client application is one built with a graphical development tool, such as Visual Basic or Delphi, running on a Windows 3.1, Windows 95, or Windows NT platform. The Visual M Server and Open M can be located on the local computer running the Visual M application (client computer) or on a remote computer (Open M server computer).

Thus, with Visual M, you can create distributed client-server applications, combining ease of GUI development with the power of Open M for database manipulation.

Open M Platforms Supporting Visual M Server

Visual M server capabilities are available on all Visual M client platforms, as well as non-Windows Open M platforms. The Visual M Server is currently available with the following implementations of Open M:

Platform	Version	TCP Connection
Open M [DTM] for DOS	6.3+	FTP or Wollongong
Open M [DTM] for Windows 3.1	6.3+	FTP or Wollongong
Open M [ISM] for Unix	6.1	Built-in
Open M [ISM] for OpenVMS Alpha	6.1	UCX
Open M [ISM] for Digital UNIX	6.1	Built-in
Open M for Windows 95 and NT	7.0	Built-in
Open M [DSM] for OpenVMS VAX	6.4+	UCX or Wollongong
Open M [DSM] for OpenVMS Alpha	6.4+	UCX or Wollongong

Visual M Clients

Visual M client systems are systems on which you can create Visual M applications. Currently, there are three Open M implementations that include Visual M client software:

Platform	Version	TCP Connection
Open M [DTM] for Windows 95 and NT	7.0	Microsoft
Open M [DTM] for Windows 3.1	6.3+	Microsoft

Visual M Client Functionality

From a Visual M client, you can:

- Use the Open M DeskTop, a powerful visual development environment, to create, edit and manage standard M routines.
- Write Visual M applications that allow you to:
 - Run Open M routines locally or remotely
 - Access local or remote Open M data

You can write the GUI interface portion of a Visual M application in two ways:

- With Visual Basic, using one or both of the two Visual M custom controls in a Visual Basic project:
 - The M/VB control lets you access M routines and globals from a Visual Basic application.
 - The M/DeskTop control lets you access Visual Basic controls from M or access the Open M DeskTop from Visual Basic in order to write event code in M.
- With non-VB GUI tools, such as Delphi, using a .DLL that provides access to the functionality in the M/VB control.

Clients Talk with Open M Via a Visual M Server

You can configure client applications to run using either a local or a remote Visual M Server.

Connecting to a Remote Visual M Server

You can run the Visual M Server on a remote computer, using TCP network connections.

If all your Visual M applications will connect to a Visual M Server on a remote computer, then:

- You only need to load the client portion of Visual M on client computers, saving valuable resources on these computers.
- The remote Visual M Server must be running before a client application attempts to connect to it.

Connecting to a Local Visual M Server

You can choose to load Open M and Visual M Server capabilities on Visual M client computers. Do this if:

- The client computer will also act as an M Server for routines and possibly data.
- You want to get maximum speed for Visual M Server-Open M communications during your development of Visual M applications by connecting to a local Visual M Server.

Configuration of Visual M Client Systems

You control these aspects of Visual M client configuration:

- Visual M Servers connected to this client
- Which Visual M Server or group of Servers specific Visual M applications use
- Which Visual M Server or group of Servers Visual M applications use when you haven't defined a specific Server or Servers for them.

Choosing a Visual M Server

If you define more than one Visual M Server as a default server, or as available for use by a specific Visual M application, then when the Visual M application runs, the user sees the Choose Connection dialog box.

At this box, you select the Visual M Server from those in the combo box list. The contents of this list contains all the Visual M Servers you defined in the Client Manager or a subset, depending on how you defined the (default) or application-specific connection in the Client Manager. You can also click on the Edit button to enter the Client Manager and define another connection. See the Visual M component of the Open M help file for more information.

To avoid the display of this box, define only one specific Visual M Server as the default Server, or for a specific application.

Default Configuration

Upon a new installation, Open M for Windows defines a default Visual M client configuration. If you have already set up a configuration, Visual M uses that configuration.

The table below describes the default configuration for each Visual M client system.

Client System	Visual M Server Connection	Starts a Visual M Server	Default Visual M Server
Open M [DTM] for Windows 3.1	none	When a Visual M application attempts to access a local Visual M Server and one is not already running at the requested port number.	Choose Connection dialog box with all Visual M Servers defined in the Visual M Client Manager.
Open M or Windows 95 and NT	LOCALTCP	When Open M starts.	

Edit Your Client Configuration with the Visual M Client Manager

If desired, you can customize this configuration to meet your needs. You can define additional remote Visual M Servers in the Client Manager. You can cause an application to automatically use a specific Visual M Server. When an application connects to a Visual M Server, that Server must already be running, unless it is a local Open M for Windows 3.1 server.

Run the Visual M Client Manager to define connections to all servers that client may use. When you define a connection, be sure you specify the same port number for a server that you specify in the start up call for that Visual M Server. If the port numbers are not the same, the two machines will be unable to communicate.

On Open M for Windows 95 and NT, a local TCP connection is predefined with the name "LOCALTCP".

For more information, select Help from the Visual M Client Manager menu bar.

Install TCP/IP if using Remote Visual M Server

Be sure that TCP/IP is correctly installed and configured on the system that will run the Visual M client.

Configuration of Visual M Server Systems

If you use a remote Visual M Server or a local server that connects via TCP, you will need to start the server whenever you start M. It should be running at all times, since client Visual M applications cannot start it themselves and will fail if it is not running.

Procedure To set up a Visual M server:

1. Install and configure TCP/IP on your system.

In the future, other forms of remote connection may be supported as well.

2. Install Open M for your system.
3. Run Open M.
4. If necessary, start a Visual M Server.

For information about Visual M Server startup, see “Starting the Visual M Server” below.

Some Open M systems start a Visual M Server process when Open M starts. The table below describes these defaults.

Note: On Open M [DTM] for Windows 3.1 and Open M [DTM] for DOS systems, Visual M starts the local Visual M Server if it is not running when a Visual M client application attempts to access it.

5. On a Visual M client system, run the Visual M Server Manager to set up users and password and security levels on applications. If the Visual M Server is an Open M [DSM] system, you also define namespace translations, converting namespace values to a volume set,UCI.

For information about the Visual M Server Manager, select Help from the Server Manager menu bar.

Starting the Visual M Server

When you start a Visual M Server, you start a master server process. If you start the server with spawning on, then the master server starts a slave server each time a Visual M client application connects.

There are two entry points to the Visual M server, *Fore* and *Back*, which start the master server in the foreground or background, respectively.

Note: The older entry points, *fore* and *back*, are still available for local server usage, though the new ones can serve either local or remote clients.

Table 2: ^%mnb Command Parameters

Parameter	Description
autohalt	Autohalt is not possible on non-DTM systems. Always set this argument to 0. Although it will be ignored if you set it to another number, future upgrades may catch you by surprise.
spawn	<p>Indicates whether the Server will spawn slave servers:</p> <ul style="list-style-type: none"> ■ 0 - spawning is off. ■ 1 - spawning is on. <p>Remote server: Behavior is controlled by the TCP connection rules. TCP does not allow a single port to connect to multiple clients at the same time, so a single server job reads from only a single port (and serves only a single client application).</p> <p>Remote foreground server: You may turn off spawning, but then it will only support a single client at a time (using the master server).</p> <p>Remote background server: The given value is ignored; spawning is always on. A separate port is provided by each spawned slave server.</p>

Serving Multiple Clients with Spawning On

When spawning is on, a Visual M Server starts a slave server for each client application.

Port Allocation Under TCP Connections

When a Visual M Server connects to Visual M clients via TCP, each slave server is given a TCP/IP port using a variation of the Least Recently Used algorithm, starting with the initial communication port you specify in the Server's startup call. As clients request connection, the server assigns them the next port available. For example, the client requests connection from the server using the listening port, 6001. The server responds with a port number for the client. In this case, port 6002 is assigned since this is the first port request.

The client drops connection to 6001 and connects to the server at 6002. When the next client sends a request on port 6001 for connection to the server, the server sends that client a command to connect to port 6003. As clients disconnect, those ports are available, but will not be assigned til less recently used ports are used.

Since only one connection can be established on any single port, it is important to make sure that no other jobs will connect to the range of ports that your Visual M clients use. Although the server checks a port before assignment to make sure that another Visual M client is not using the port, it does not check to see if another type of process is using the port. If the assigned port is already in use, the Visual M Server keeps checking all ports available to it (determined by the *number of ports* argument in the startup call). If none are available, the the client can not connect to the server.

Starting Multiple Visual M Servers

You can run more than one master Visual M Server simultaneously. You may want to do this to run applications in separate contexts on the same server. To do this, you must:

- Start each Visual M Server in background mode.
- Define each Visual M Server's initial port and maximum number of slave servers so port numbers cannot overlap, either with each other or with other applications on your server that use TCP.

You can start additional Visual M Servers with the startup call you want by issuing that call from the M programmer's prompt or adding that call to your user startup routine (such as ZSTU in ISM and Open M for Windows 95 and NT).

Stopping a Visual M Server

On all systems, you stop all Visual M Servers running on your system, and all their slave servers if spawning is on, by issuing this command from the server's console:

```
do Stop^%mnb()
```

Note: You can still use the old entry point, stop^%mnb().

In addition, see specific Open M Server platforms below.

Open M [DTM] for Windows 3.1

On an Open M [DTM] for Windows 3.1 system, there are a number of ways to stop a local Visual M Server. In all cases, you are stopping the master Visual M Server. This automatically halts all slave servers connected to that master as well.

Set Server to Stop Automatically in Startup Call

When you start a local Visual M Server, you can cause DTM to autohalt when the last DTM process halts.

Stopping a Foreground Server

1. Restore the icon for the foreground Visual M Server you want to stop.
2. Press the <Escape> key.

Open M for Windows 95 and NT

ShutDown Open M

You can stop a Visual M Server by selecting Stop Open M from the Open M toolbar. You see the Shutdown dialog box.

If you click the Shutdown button, you shut down the local Open M and any Visual M Servers you have started from it. You can restart this Server by selecting the “Restart after shutdown” checkbox.

While Open M is shutting down, you see the “Stopping Open M” splash box:

Stopping a Specific Slave Server

There is now a way to stop an individual slave server:

```
do StopJob^%mvb(jobnumber)
```

where jobnumber is the number of the slave server job.

This entry point stops the specified slave server job on the next read timeout cycle. A read timeout cycle lasts about 2 seconds.

Restarting a Visual M Server

You can elect to restart the Visual M Server, whether it is local or remote, by selecting the “Restart after shutdown” checkbox in the Shutdown dialog box on an Open M for Windows 95 and NT system.

Open M [DSM] Version 6.4

From the M prompt, issue the command:

```
do STOP^%mvb
```