

# TermMon ActiveX Control

The TermMon ActiveX Control can be used to collect information about a terminal and perform terminal functions. For the purposes of this document Control will refer to the TermMon ActiveX Control.

This section is not meant as a tutorial on ActiveX controls but is a datasheet on the properties and functions of the TermMon ActiveX control.

---

## Registering the Control

The TermMon ActiveX Control can be found on the ThinManager CD as **termmon.ocx**.

The Control must be registered before it can be used. Copy the file **termmon.ocx** to the computer where you want to use it. Register the OCX by executing **regsvr32 <path\termmon.ocx>**.

---

## Using the Control

### TermMon ActiveX Configuration Module

If running the Control in the terminal's terminal services session, no special configuration of the terminal in ThinManager is required. The **TermMon ActiveX Control Configuration Module** is not required.

If the Control is not run in the terminal's terminal services session, the **TermMon ActiveX Control Configuration Module** must be added to the terminal configuration in ThinManager. In the module configuration, **Allow ActiveX Connections** must be set to **YES** and **Only Allow Connections from Session** must be set to **NO**.

The **TermMon ActiveX Control Configuration Module** can be used to prevent a connection from an application running in the terminal's terminal services session. This is accomplished by adding the **TermMon ActiveX Control Configuration Module** to the terminal in ThinManager and setting the **Allow ActiveX Connections** option to **NO**.

### Read-Only Properties

The following properties are read only strings. An event will be generated any time one of these properties changes. The Enable method must be invoked prior to reading these properties.

- **TerminalName** - This is the name of the terminal.
- **TerminalModel** - This is the terminal model number.
- **TerminalIP** - This is the terminal IP address.
- **TerminalMAC** - This is the terminal MAC Address.
- **TerminalBootLoaderVersion** - This is the terminal network boot loader version.
- **TerminalFirmwareVersion** - This is the firmware version that the terminal is running.
- **TerminalWindowsUsername** - This is the Windows Username that is specified in the terminal's ThinManager configuration.

- **TermSecureUsername** - This is the TermSecure username of the TermSecure user currently logged onto the terminal. If no TermSecure user is logged on, this value will be blank.
- **TermSecureWindowsUsername** - This is the Windows Username associated with a TermSecure user. This is the Windows Username for all TermSecure user sessions. If no TermSecure user is logged on, this value will be blank.
- **TerminalServerGroupList** - This is a comma-separated list of Display Clients currently running on the terminal.
- **ConnectionState** - This is the Control's connection state with the terminal.
- **CurrentTerminalServerGroup** - This is the Display Client that is currently being displayed on the terminal.
- **CurrentWindowsUsername** - This is the Windows Username of the session where the Control has been executed. This property is not available when the RunInSession property is set to **False**.
- **TerminalServerName** - This is the name of the Terminal Server where the Control is running. This property is not available when the RunInSession property is set to **False**.

## Read-Write Properties

These properties can be set by the application.

- **RunInSession** - When the RunInSession property is set to True, the Control will be running in the terminal's terminal services session. The terminal IP address will be determined automatically by the control.
- **OverrideIP** - If the RunInSession property is set to False, the OverrideIP property specifies the IP Address of the terminal that the Control will connect to.

---

**Note:** To use the OverrideIP property, the TermMon ActiveX Control Configuration Module must be added to the terminal configuration in ThinManager. In the module configuration, **Allow ActiveX Connections** must be set to **YES**, and **Only Allow Connections from Session** must be set to **NO**.

---

- **WatchdogTime** – This is the number of seconds before the watchdog will reset the terminal session. Once this property is set to a non-zero value, the property must be updated before the watchdog time reaches zero. To disable the watchdog, set this property to zero. The watchdog is disabled by default.

---

**Note:** The **Enable Method** does not need to be called for watchdog operation. Watchdog operation is independent of the **Enable** and **Disable Methods**.

---

- **ActiveScreen** – For MultiMonitor configurations, this is the active screen number. A value of zero (default) will set the active screen to the screen the mouse pointer is on when a method or command is executed. A non-zero value will set the Active Screen to the screen number specified. All methods and commands will be executed on the specified screen.

## Events

When a property value changes, an event will be generated by the Control. When an Event occurs the event code can be used to determine the property that changed. The Enable method must be invoked in order to receive events (except for WatchdogTime). The event code is provided by the Control as follows:

- TermMonEvent.TerminalName
- TermMonEvent.TerminalModel
- TermMonEvent.TerminalIP

- TermMonEvent.TerminalMAC
- TermMonEvent.TerminalBootLoaderVersion
- TermMonEvent.TerminalFirmwareVersion
- TermMonEvent.TerminalWindowsUsername
- TermMonEvent.TermSecureUsername
- TermMonEvent.TermSecureWindowsUsername
- TermMonEvent.TerminalServerGroupList
- TermMonEvent.ConnectionState
- TermMonEvent.CurrentTerminalServerGroup
- TermMonEvent.CurrentWindowsUsername
- TermMonEvent.TerminalServerName
- TermMonEvent.WatchdogTime

## Methods

- **Enable** - Invoking this method will enable the Control. The Control will attempt to connect to the terminal and generate events to update the Control Properties. The Control will maintain a connection to the terminal as long as it is enabled.
- **Disable** - Invoking this method will cause the Control to break the connection with the terminal. Events will be generated to clear the Control Properties.
- **Command** - The Command method can be used to send terminal action commands. The Command method requires one parameter which is the terminal command to be performed. The Enable method must be invoked before these commands can be executed (except for noted exceptions). The supported commands are:
  - **Reboot** - This command will initiate a terminal reboot.
  - **Restart** - This command will initiate a terminal restart.
  - **Calibrate** - This command will initiate a touch screen calibration.
  - **GotoMainMenu** - This command will cause the Main Menu to be displayed.
  - **SwitchToNextGroup** - This command will switch to the next Display Client.
  - **SwitchToPrevGroup** - This command will switch to the previous Display Client.
  - **SwitchInstFailover** - This command will switch the instant failover group.
  - **ChangeTermSecureUser** - This command will disconnect any current TermSecure user sessions and then display the TermSecure Log On menu.
  - **LogOffAndChangeTermSecureUser** - This command will log off any current TermSecure user sessions and then display the TermSecure Log On menu.
  - **LogOffTermSecureUser** - This command will log off any current TermSecure user sessions and will return to a Display Client which is assigned to the terminal. If no Display Clients have been configured on the terminal, the TermSecure Log On menu will be displayed.
  - **DisconnectTermSecureUser** - This command will disconnect any current TermSecure user sessions and will return to a Display Client which is assigned to the terminal. If no Display Clients have been configured on the terminal, the TermSecure Log On menu will be displayed.

- **DisconnectSession** - This command will disconnect the Terminal Services Session running on the terminal. This command does not require that the Enable Method be invoked prior to execution.
- **LogOffSession** - This command will log off the Terminal Services Session running on the terminal. This command does not require that the Enable Method be invoked prior to execution.
- **TileStart** - This command will tile the Display Clients on the current Screen.
- **TileEnd** - This command will untile the Display Clients on the current Screen.

The Command Method constants are provided by the Control as follows:

- TermMonCommand.Reboot
  - TermMonCommand.Restart
  - TermMonCommand.Calibrate
  - TermMonCommand.GotoMainMenu
  - TermMonCommand.SwitchToNextGroup
  - TermMonCommand.SwitchToPrevGroup
  - TermMonCommand.SwitchInstFailover
  - TermMonCommand.ChangeTermSecureUser
  - TermMonCommand.LogOffAndChangeTermSecureUser
  - TermMonCommand.LogOffTermSecureUser
  - TermMonCommand.DisconnectTermSecureUser
  - TermMonCommand.DisconnectSession
  - TermMonCommand.LogOffSession
  - TermMonCommand.TileStart
  - TermMonCommand.TileEnd
- **ChangeTerminalServerGroup** - This method can be used to change the Display Client currently displayed on the terminal. This method requires one parameter which is the name of the Display Client that the terminal should switch to.
  - **TermSecureCheckAccess** - This method can be used to query the access rights of a TermSecure user. This method requires two parameters. The first parameter is the name of the user. The second parameter is the name of the Access Group. This method returns the result of the query as follows:
    - **TermMonConst.Timeout** - The request timed out.
    - **TermMonConst.Busy** - The Control is busy with another request.
    - **TermMonConst.InvalidMember** - The user is not a member of the specified TermSecure Access Group.
    - **TermMonConst.ValidMember** - The user is a member of the specified TermSecure Access Group.
    - **TermMonConst.UserNotFound** - The TermSecure Username was not found.
    - **TermMonConst.GroupNotFound** - The Access Group Name was not found.

- **GetGroupScreen** - This method can be used to determine which screen the specified Display Client is currently on for MultiMonitor configurations. This method requires one parameter which is the name of the Display Client.
- **TermSecureLogonUser** - This method can be used to Log On a specified TermSecure user. This method requires two parameters. The first parameter is the name of the TermSecure user. The second parameter is the password of the TermSecure user. The password will be encrypted before being sent to the terminal. This method returns a result as follows:
  - **TermMonConst.Success** - The TermSecure user was successfully logged on.
  - **TermMonConst.Timeout** - The request timed out.
  - **TermMonConst.Busy** - The Control is busy with another request.
  - **TermMonConst.UserNotFound** - The TermSecure username was not found.
  - **TermMonConst.BadPassword** - The TermSecure password was invalid.
  - **TermMonConst.NoPermission** - The TermSecure user does not have permission to use the terminal.
  - **TermMonConst.PasswordChangeReq** - The TermSecure user is required to change his password.
  - **TermMonConst.NoWindowsUsername** - This TermSecure user does not have a Windows Username specified in the TermSecure user configuration. This is only required for Terminal Services Display Clients assigned to the TermSecure User.
  - **TermMonConst.NoWindowsPassword** - This TermSecure user does not have a Windows Password specified in the TermSecure user configuration. This is only required for Terminal Services Display Clients assigned to the TermSecure User.
- **CameraOverlayEnable** - This method is used to enable a camera overlay. This method requires two parameters. The first parameter is the name of the Display Client the overlay is on. The second parameter is the name of the overlay.
- **CameraOverlayDisable** - This method is used to disable a camera overlay. This method requires two parameters. The first parameter is the name of the Display Client the overlay is on. The second parameter is the name of the overlay.
- **CameraOverlayCycleStart** - This method is used to start camera cycling for a camera overlay. This method requires two parameters. The first parameter is the name of the Display Client the overlay is on. The second parameter is the name of the overlay.
- **CameraOverlayCycleStop** - This method is used to stop camera cycling for a camera overlay. This method requires two parameters. The first parameter is the name of the Display Client the overlay is on. The second parameter is the name of the overlay.
- **CameraOverlaySwitchNext** - This method is used to switch to the next camera in a camera overlay list. This method requires two parameters. The first parameter is the name of the Display Client the overlay is on. The second parameter is the name of the overlay.
- **CameraOverlaySwitchPrev** - This method is used to switch to the previous camera in a camera overlay list. This method requires two parameters. The first parameter is the name of the Display Client the overlay is on. The second parameter is the name of the overlay.
- **CameraOverlayFullscreenEnter** - This method is used to make the current camera in a camera overlay enter full screen. This method requires two parameters. The first parameter is the name of the Display Client the overlay is on. The second parameter is the name of the overlay.
- **CameraOverlayFullscreenExit** - This method is used to make the current camera in a camera overlay exit full screen. This method requires two parameters. The first parameter is the name of the Display Client the overlay is on. The second parameter is the name of the overlay.

- **CameraOverlaySwitchByName** - This method is used to change cameras in a camera overlay. This method requires three parameters. The first parameter is the name of the Display Client the overlay is on. The second parameter is the name of the overlay. The third parameter is the name of the camera. The camera name must include the full path if the camera is in a camera group.
- **CameraOverlayMove** - This method is used to change the position of a camera overlay. This method requires four parameters. The first parameter is the name of the Display Client the overlay is on. The second parameter is the name of the overlay. The third parameter is the x location. The fourth parameter is the y position.
- **CameraOverlayResize** - This method is used to change the size of a camera overlay. This method requires four parameters. The first parameter is the name of the Display Client the overlay is on. The second parameter is the name of the overlay. The third parameter is the width. The fourth parameter is the height.
- **CameraOverlayResizeMove** - This method is used to change the size and position of a camera overlay. This method requires six parameters. The first parameter is the name of the Display Client the overlay is on. The second parameter is the name of the overlay. The third parameter is the x position. The fourth parameter is the y position. The fifth parameter is the width. The sixth parameter is the height.

## Control Constants

Constant values provided by the Control are as follows:

### TermMonEvent

- TerminalName 1
- TerminalModel 2
- TerminalIP 3
- TerminalMAC 4
- TerminalBootLoaderVersion 5
- TerminalFirmwareVersion 6
- TerminalWindowsUsername 7
- TermSecureUsername 8
- TermSecureWindowsUsername 9
- TerminalServerGroupList 10
- ConnectionState 11
- CurrentTerminalServerGroup 12
- CurrentWindowsUsername 13
- TerminalServerName 14
- WatchdogTime 15

### TermMonCommand

- Reboot 100
- Restart 101
- Calibrate 102
- GotoMainMenu 103
- SwitchToNextGroup 104
- SwitchToPrevGroup 105
- SwitchInstFailover 106
- ChangeTermSecureUser 107
- LogOffAndChangeTermSecureUser 108
- LogOffTermSecureUser 109
- DisconnectTermSecureUser 110
- DisconnectSession 111
- LogOffSession 112
- TileStart 113
- TileEnd 114

## TermMonConst

- Success 0
- Fail 1
- Disconnected 2
- Connected 3
- Timeout 4
- Busy 5
- Updating 6
- RequestFailed 7
- InvalidMember 8
- ValidMember 9
- UserNotFound 10
- GroupNotFound 11
- BadPassword 12
- NoPermission 13
- PasswordChangeReq 14
- NoWindowsUsername 15
- NoWindowsPassword 16

---

## TermMon ActiveX Demo Application

The TermMon ActiveX Demo Application can be used to demonstrate the features of the Control. The demo application is found on the ThinManager CD in the **Terminal ActiveX** folder as **TermMon.exe**.

To start the application in graphical mode, run **TermMon.exe** in a terminal's terminal services session. This will allow the Control functionality to be demonstrated.

The demo application can also be run non-graphically using command line options. The format is as follows:

```
TermMon -c <command> -f <output path and filename> -d <data> -a <ip address>
```

The following commands are terminal action commands:

- Reboot
- Restart
- Calibrate
- GotoMainMenu
- SwitchToNextGroup
- SwitchToPrevGroup
- SwitchInstFailover



- ChangeTermSecureUser
- LogOffAndChangeTermSecureUser
- LogOffTermSecureUser
- DisconnectTermSecureUser
- ChangeTerminalServerGroup -d <groupname>

The following commands return the result in the output filename.

- TerminalName -f <output filename>
- TerminalModel -f <output filename>
- TerminalIP -f <output filename>
- TerminalMAC -f <output filename>