X Display Power Management Signaling (DPMS) Extension

X Consortium Standard

Rob Lembree

X Display Power Management Signaling (DPMS) Extension: X Consortium Standard

by Rob Lembree

X Version 11, Release 6.

Version 1.0

Table of Contents

Chapter 1. Overview

This extension provides X Protocol control over the VESA Display Power Management Signaling (DPMS) characteristics of video boards under control of the X Window System.

Traditionally, the X Window System has provided for both blanking and non-blanking screen savers. Timeouts associated with these built-in screen saver mechanisms are limited to idle (dwell) time, and a change timeout that specifies the change interval for non-blanking screen savers.

The United States' Environmental Protection Agency (EPA) Energy Star program requires that monitors power down after some idle time by default. While it is possible to simply overload the existing screen saver timeouts, this solution leaves the non-privileged user little to no control over the DPMS characteristics of his or her system. For example, disabling DPMS would require some unintended side effect in the core screen saver, such as disabling the changing of a non-blanking screen saver. Providing clients with this control requires an extension to the core X Window System Protocol, and this extension seeks to fill this gap.

There are four power levels specified by the Video Electronics Standards Association (VESA) Display Power Management Signaling (DPMS) standard. These are mapped onto the X DPMS Extension like this:

`DPMS Extension Power Levels`

0	DPMSModeOn	In use
1	DPMSModeStandby	Blanked, low power
2	DPMSModeSuspend	Blanked, lower power
3	DPMSModeOff	Shut off, awaiting activity

Chapter 2. DPMS Functions

`Bool` **`DPMSQueryExtention`**`(`*`*display`*`, `*`event_base`*`, `*`error_base`*`);`

**display*	Specifies the connection to the X server.
event_base	Specifies the return location for the assigned base event
error_base	Specifies the return location for the assigned base error

The DPMSQueryExtension function queries the X server to determine the availability of the DPMS Extension. If the extension is available, the return value is TRUE, and *event_base* and *error_base* are set to the base event number and base error number for the extension, respectively. Otherwise, the return value is FALSE, and the values of *event_base* and *error_base* are undefined.

`Status` **`DPMSGetVersion`**`(`*`*display`*`, `*`*major_version`*`, `*`*minor_version`*`);`

display	Specifies the connection to the X server.
major_version	Specifies the return location for the extension major version.
minor_version	Specifies the return location for the extension minor version.

The DPMSGetVersion function returns the version of the DPMS extension implemented by the X server. The version is returned in *major_version* and *minor_version*. The major version and minor version for this specification are '1' and '1', respectively. The major version will be incremented for protocol incompatible changes, and the minor version will be incremented for small, upwardly compatible changes.

`Bool` **`DPMSCapable`**`(`*`*display`*`);`

display	Specifies the connection to the X server.

The DPMSCapable function returns the DPMS capability of the X server, either TRUE (capable of DPMS) or FALSE (incapable of DPMS). The capability of an X server is implementation defined. For example, if a multi-headed X server is capable of DPMS on one head, and incapable on another, the truth value of this function is defined by the X server implementation.

`Status` **`DPMSSetTimeouts`**`(`*`*display`*`, `*`standby`*`, `*`suspend`*`, `*`off`*`);`

display	Specifies the connection to the X server.
standby	Specifies the new standby timeout in seconds.
suspend	Specifies the new suspend timeout in seconds.
off	Specifies the new off timeout in seconds.

The DPMSSetTimeouts function permits applications to set the timeout values used by the X server for DPMS timings.

The value *standby* is the amount of time of inactivity in seconds before standby mode is invoked. The actual effects of this mode are implementation defined, but in the case of DPMS compliant hardware, it is implemented by shutting off the horizontal sync signal, and pulsing the vertical sync signal. Standby mode provides the quickest monitor recovery time. Note also that many monitors implement this mode identically to suspend mode. A value of zero disables this mode.

The value *suspend* is the amount of time of inactivity in seconds before the second level of power savings is invoked. Suspend mode's physical and electrical characteristics are implementation defined, but in DPMS compliant hardware, results in the pulsing of the horizontal sync signal, and shutting off of the vertical sync signal. Suspend mode recovery is considered to be slower than standby mode, but faster than off mode, however this is monitor dependent. As noted above, many monitors implement this mode identically to standby mode. A value of zero disables this mode.

The value *off* is the amount of time of inactivity in seconds before the third and final level of power savings is invoked. Off mode's physical and electrical characteristics are implementation defined, but in DPMS compliant hardware, is implemented by shutting off both horizontal and vertical sync signals, resulting in the power-down of the monitor. Recovery time is implementation dependant, but frequently is similar to the power-up time of the monitor. A value of zero disables this mode.

Chronologically, standby mode occurs before or simultaneously with suspend mode, and suspend mode must occur before or simultaneously with off mode. Therefore, non-zero mode timeout values must be greater than or equal to the timeout values of earlier modes. If inconsistent values are supplied, a BadValue error will result.

`Status` **`DPMSGetTimeouts`**`(`*`*display`*`, `*`*standby`*`, `*`*suspend`*`, `*`*off`*`);`

display	Specifies the connection to the X server.
standby	Specifies the new standby timeout in seconds.
suspend	Specifies the new suspend timeout in seconds.
off	Specifies the new off timeout in seconds.

The DPMSGetTimeouts function retrieves the timeout values used by the X server for DPMS timings.

The value *standby* is the amount of time of inactivity in seconds before standby mode is invoked. A value of zero indicates that this mode has been disabled.

The value *suspend* is the amount of time of inactivity in seconds before the second level of power savings is invoked. A value of zero indicates that this mode has been disabled.

The value *off* is the amount of time of inactivity in seconds before the third and final level of power savings is invoked. A value of zero indicates that this mode has been disabled.

`Status` **`DPMSEnable`**`(`*`*display`*`);`

display	Specifies the connection to the X server.

The DPMSEnable function enables DPMS on the specified display. When enabled, DPMS will use the currently saved timeout values, and will invoke the DPMS power mode appropriate for the amount of time that the workstation input devices have been idle. If DPMSEnable is invoked on a display with DPMS already enabled, no change is made, and no error is returned. If DPMSEnable is invoked on a display without support for DPMS, no change is made and no error is returned.

`Status DPMSDisable(*display);`

display	Specifies the connection to the X server.

The DPMSDisable function disables DPMS on the specified display. When disabled, DPMS returns the display to DPMSModeOn. If DPMSDisable is invoked on a display with DPMS already disabled, no change is made, and no error is returned. If DPMSDisable is invoked on a display without support for DPMS, no change is made and no error is returned.

`Status DPMSForceLevel(*display, level);`

display	Specifies the connection to the X server.
level	Specifies the level to force power to.

The DPMSForceLevel function forces a DPMS capable display into the specified power level. The *level* must be one of DPMSModeOn, DPMSModeStandby, DPMSModeSuspend, or DPMSModeOff. Values other than these will result in a BadValue error. If DPMS is disabled on the display, a BadMatch protocol error will result.

Status DPMSInfo(*display, power_level, state*)

`Status DPMSInfo(*display, power_level, *state);`

display	Specifies the connection to the X server.
power_level	Specifies the current power level.
state	Specifies the current DPMS state.

The DPMSInfo function returns information about the current DPMS state. The *state* return parameter indicates whether or not DPMS is enabled (TRUE) or disabled (FALSE). The *power_level* return parameter indicates the current power level (one of DPMSModeOn, DPMSModeStandby, DPMSModeSuspend, or DPMSModeOff.)