

**Root Window Properties (and Related Messages)**[Prev](#)[Next](#)**Root Window Properties (and Related Messages)**

Whenever this spec speaks about “sending a message to the root window”, it is understood that the client is supposed to create a ClientMessage event with the specified contents and send it by using a SendEvent request with the following arguments:

```
destination      root
propagate        False
event-mask       (SubstructureNotify|SubstructureRedirect)
event            the specified ClientMessage
```

**\_NET\_SUPPORTED**

```
_NET_SUPPORTED, ATOM[]/32
```

This property MUST be set by the Window Manager to indicate which hints it supports. For example: considering \_NET\_WM\_STATE both this atom and all supported states e.g. \_NET\_WM\_STATE\_MODAL, \_NET\_WM\_STATE\_STICKY, would be listed. This assumes that backwards incompatible changes will not be made to the hints (without being renamed).

**\_NET\_CLIENT\_LIST**

```
_NET_CLIENT_LIST, WINDOW[]/32
_NET_CLIENT_LIST_STACKING, WINDOW[]/32
```

These arrays contain all X Windows managed by the Window Manager. \_NET\_CLIENT\_LIST has initial mapping order, starting with the oldest window. \_NET\_CLIENT\_LIST\_STACKING has bottom-to-top stacking order. These properties SHOULD be set and updated by the Window Manager.

**\_NET\_NUMBER\_OF\_DESKTOPS**

```
_NET_NUMBER_OF_DESKTOPS, CARDINAL/32
```

This property SHOULD be set and updated by the Window Manager to indicate the number of virtual desktops.

A Pager can request a change in the number of desktops by sending a \_NET\_NUMBER\_OF\_DESKTOPS message to the root window:

```
_NET_NUMBER_OF_DESKTOPS
message_type = _NET_NUMBER_OF_DESKTOPS
format = 32
data.l[0] = new_number_of_desktops
other data.l[] elements = 0
```

The Window Manager is free to honor or reject this request. If the request is honored \_NET\_NUMBER\_OF\_DESKTOPS MUST be set to the new number of desktops, \_NET\_VIRTUAL\_ROOTS MUST be set to store the new number of desktop virtual root window IDs and \_NET\_DESKTOP\_VIEWPORT and \_NET\_WORKAREA must also be changed accordingly. The \_NET\_DESKTOP\_NAMES property MAY remain unchanged.

If the number of desktops is shrinking and \_NET\_CURRENT\_DESKTOP is out of the new range of available desktops, then this MUST be set to the last available desktop from the new set. Clients that are still present on desktops that are out of the new range MUST be moved to the very last desktop from the new set. For these \_NET\_WM\_DESKTOP MUST be updated.

**\_NET\_DESKTOP\_GEOMETRY**

```
_NET_DESKTOP_GEOMETRY width, height, CARDINAL[2]/32
```

Array of two cardinals that defines the common size of all desktops (this is equal to the screen size if the Window Manager doesn't support large desktops, otherwise it's equal to the virtual size of the desktop). This property SHOULD be set by the Window Manager.

A Pager can request a change in the desktop geometry by sending a \_NET\_DESKTOP\_GEOMETRY client message to the root window:

```
_NET_DESKTOP_GEOMETRY
```

```

message_type = _NET_DESKTOP_GEOMETRY
format = 32
data.l[0] = new_width
data.l[1] = new_height
other data.l[] elements = 0

```

The Window Manager MAY choose to ignore this message, in which case `_NET_DESKTOP_GEOMETRY` property will remain unchanged.

## **`_NET_DESKTOP_VIEWPORT`**

`_NET_DESKTOP_VIEWPORT` `x`, `y`, `CARDINAL[][2]/32`

Array of pairs of cardinals that define the top left corner of each desktop's viewport. For Window Managers that don't support large desktops, this MUST always be set to (0,0).

A Pager can request to change the viewport for the current desktop by sending a `_NET_DESKTOP_VIEWPORT` client message to the root window:

```

_NET_DESKTOP_VIEWPORT
message_type = _NET_DESKTOP_VIEWPORT
format = 32
data.l[0] = new_vx
data.l[1] = new_vy
other data.l[] elements = 0

```

The Window Manager MAY choose to ignore this message, in which case `_NET_DESKTOP_VIEWPORT` property will remain unchanged.

## **`_NET_CURRENT_DESKTOP`**

`_NET_CURRENT_DESKTOP` `desktop`, `CARDINAL/32`

The index of the current desktop. This is always an integer between 0 and `_NET_NUMBER_OF_DESKTOPS - 1`. This MUST be set and updated by the Window Manager. If a Pager wants to switch to another virtual desktop, it MUST send a `_NET_CURRENT_DESKTOP` client message to the root window:

```

_NET_CURRENT_DESKTOP
message_type = _NET_CURRENT_DESKTOP
format = 32
data.l[0] = new_index
data.l[1] = timestamp
other data.l[] elements = 0

```

Note that the timestamp may be 0 for clients using an older version of this spec, in which case the timestamp field should be ignored.

## **`_NET_DESKTOP_NAMES`**

`_NET_DESKTOP_NAMES`, `UTF8_STRING[]`

The names of all virtual desktops. This is a list of NULL-terminated strings in UTF-8 encoding [\[UTF8\]](#). This property MAY be changed by a Pager or the Window Manager at any time.

Note: The number of names could be different from `_NET_NUMBER_OF_DESKTOPS`. If it is less than `_NET_NUMBER_OF_DESKTOPS`, then the desktops with high numbers are unnamed. If it is larger than `_NET_NUMBER_OF_DESKTOPS`, then the excess names outside of the `_NET_NUMBER_OF_DESKTOPS` are considered to be reserved in case the number of desktops is increased.

Rationale: The name is not a necessary attribute of a virtual desktop. Thus the availability or unavailability of names has no impact on virtual desktop functionality. Since names are set by users and users are likely to preset names for a fixed number of desktops, it doesn't make sense to shrink or grow this list when the number of available desktops changes.

## **`_NET_ACTIVE_WINDOW`**

`_NET_ACTIVE_WINDOW`, `WINDOW/32`

The window ID of the currently active window or None if no window has the focus. This is a read-only property set by the Window Manager. If a Client wants to activate another window, it MUST send a `_NET_ACTIVE_WINDOW` client message to the root window:

```

_NET_ACTIVE_WINDOW
    window = window to activate
    message_type = _NET_ACTIVE_WINDOW
    format = 32
    data.l[0] = source indication
    data.l[1] = timestamp
    data.l[2] = requestor's currently active window, 0 if none
    other data.l[] elements = 0

```

Source indication should be 1 when the request comes from an application, and 2 when it comes from a pager. Clients using older version of this spec use 0 as source indication, see [the section called “Source indication in requests”](#) for details. The timestamp is Client's last user activity timestamp (see `_NET_WM_USER_TIME`) at the time of the request, and the currently active window is the Client's active toplevel window, if any (the Window Manager may be e.g. more likely to obey the request if it will mean transferring focus from one active window to another).

Depending on the information provided with the message, the Window Manager may decide to refuse the request (either completely ignore it, or e.g. use `_NET_WM_STATE_DEMANDS_ATTENTION`).

## **`_NET_WORKAREA`**

```

_NET_WORKAREA, x, y, width, height CARDINAL[][4]/32

```

This property **MUST** be set by the Window Manager upon calculating the work area for each desktop. Contains a geometry for each desktop. These geometries are specified relative to the viewport on each desktop and specify an area that is completely contained within the viewport. Work area **SHOULD** be used by desktop applications to place desktop icons appropriately.

The Window Manager **SHOULD** calculate this space by taking the current page minus space occupied by dock and panel windows, as indicated by the [\\_NET\\_WM\\_STRUT](#) or [\\_NET\\_WM\\_STRUT\\_PARTIAL](#) properties set on client windows.

## **`_NET_SUPPORTING_WM_CHECK`**

```

_NET_SUPPORTING_WM_CHECK, WINDOW/32

```

The Window Manager **MUST** set this property on the root window to be the ID of a child window created by himself, to indicate that a compliant window manager is active. The child window **MUST** also have the `_NET_SUPPORTING_WM_CHECK` property set to the ID of the child window. The child window **MUST** also have the `_NET_WM_NAME` property set to the name of the Window Manager.

Rationale: The child window is used to distinguish an active Window Manager from a stale `_NET_SUPPORTING_WM_CHECK` property that happens to point to another window. If the `_NET_SUPPORTING_WM_CHECK` window on the client window is missing or not properly set, clients **SHOULD** assume that no conforming Window Manager is present.

## **`_NET_VIRTUAL_ROOTS`**

```

_NET_VIRTUAL_ROOTS, WINDOW[]/32

```

To implement virtual desktops, some Window Managers reparent client windows to a child of the root window. Window Managers using this technique **MUST** set this property to a list of IDs for windows that are acting as virtual root windows. This property allows background setting programs to work with virtual roots and allows clients to figure out the window manager frame windows of their windows.

## **`_NET_DESKTOP_LAYOUT`**

```

_NET_DESKTOP_LAYOUT, orientation, columns, rows, starting_corner CARDINAL[4]/32

```

```

#define _NET_WM_ORIENTATION_HORZ 0
#define _NET_WM_ORIENTATION_VERT 1

#define _NET_WM_TOPLEFT 0
#define _NET_WM_TOPRIGHT 1
#define _NET_WM_BOTTOMRIGHT 2
#define _NET_WM_BOTTOMLEFT 3

```

*This property is set by a Pager, not by the Window Manager.* When setting this property, the Pager must own a manager selection (as defined in the ICCCM 2.8). The manager selection is called `_NET_DESKTOP_LAYOUT_Sn` where n is the screen number. The purpose of this property is to allow the Window Manager to know the desktop layout displayed by the Pager.

`_NET_DESKTOP_LAYOUT` describes the layout of virtual desktops relative to each other. More specifically, it describes the layout used by the owner of the manager selection. The Window Manager may use this layout information or may choose to ignore it. The property contains four values: the Pager orientation, the number of desktops in the X direction, the number in the Y direction, and the starting corner of the layout, i.e. the corner containing the first desktop.

Note: In order to inter-operate with Pagers implementing an earlier draft of this document, Window Managers should accept a `_NET_DESKTOP_LAYOUT` property of length 3 and use `_NET_WM_TOPLEFT` as the starting corner in this case.

The virtual desktops are arranged in a rectangle with rows `rows` and columns `columns`. If `rows times columns` does not match the total number of desktops as specified by `_NET_NUMBER_OF_DESKTOPS`, the highest-numbered workspaces are assumed to be nonexistent. Either rows or columns (but not both) may be specified as 0 in which case its actual value will be derived from `_NET_NUMBER_OF_DESKTOPS`.

When the orientation is `_NET_WM_ORIENTATION_HORZ` the desktops are laid out in rows, with the first desktop in the specified starting corner. So a layout with four columns and three rows starting in the `_NET_WM_TOPLEFT` corner looks like this:

```
+---+---+---+
| 0| 1| 2| 3|
+---+---+---+
| 4| 5| 6| 7|
+---+---+---+
| 8| 9|10|11|
+---+---+---+
```

With starting\_corner `_NET_WM_BOTTOMRIGHT`, it looks like this:

```
+---+---+---+
|11|10| 9| 8|
+---+---+---+
| 7| 6| 5| 4|
+---+---+---+
| 3| 2| 1| 0|
+---+---+---+
```

When the orientation is `_NET_WM_ORIENTATION_VERT` the layout with four columns and three rows starting in the `_NET_WM_TOPLEFT` corner looks like:

```
+---+---+---+
| 0| 3| 6| 9|
+---+---+---+
| 1| 4| 7|10|
+---+---+---+
| 2| 5| 8|11|
+---+---+---+
```

With starting\_corner `_NET_WM_TOPRIGHT`, it looks like:

```
+---+---+---+
| 9| 6| 3| 0|
+---+---+---+
|10| 7| 4| 1|
+---+---+---+
|11| 8| 5| 2|
+---+---+---+
```

The numbers here are the desktop numbers, as for `_NET_CURRENT_DESKTOP`.

## **`_NET_SHOWING_DESKTOP`**

`_NET_SHOWING_DESKTOP` desktop, CARDINAL/32

Some Window Managers have a "showing the desktop" mode in which windows are hidden, and the desktop background is displayed and focused. If a Window Manager supports the `_NET_SHOWING_DESKTOP` hint, it MUST set it to a value of 1 when the Window Manager is in "showing the desktop" mode, and a value of zero if the Window Manager is not in this mode.

If a Pager wants to enter or leave the mode, it MUST send a `_NET_SHOWING_DESKTOP` client message to the root window requesting the change:

```
_NET_SHOWING_DESKTOP
message_type = _NET_SHOWING_DESKTOP
```

```
format = 32
data.l[0] = boolean 0 or 1
other data.l[] elements = 0
```

The Window Manager may choose to ignore this client message.

---

[Prev](#)

Non-ICCCM features

[Up](#)

[Home](#)

[Next](#)

Other Root Window Messages