



Documentation x-Desktop Release 1.0.5

<http://www.x-desktop.org>

## Documentation Index

-- QuickJump --

## Introduction

Prepare files & directories [1. Create Frameset](#) [2. Create base loader](#)  
[3. Create first application file](#)

## Methods

**Global stuff**[Desktop initialization](#)[Online Update Check](#)[Onsite documentation](#)[Version check](#)**General stuff**[xDT.alert](#)[xDT.confirm](#)[xDT.quit](#)[xDT.sysMessage](#)[xDT.language](#)[xDT.resPath](#)[xDT.playSound](#)[xDT.frameName](#)[xDT.preloadImages](#)**Taskbar stuff**[xDT.taskbar](#)[xDT.taskbarColor](#)[xDT.taskbarStatus](#)**Single window stuff**[xDT.addWindow](#)[xDT.show](#)[xDT.hide](#)[xDT.popupWindow](#)[xDT.url](#)[xDT.html](#)[xDT.closeWindow](#)[xDT.deleteWindow](#)[xDT.refreshWindow](#)[xDT.positionWindow](#)[xDT.maximizeWindow](#)[xDT.minimizeWindow](#)[xDT.resizeWindow](#)[xDT.window.name](#)[xDT.window.find](#)[xDT.window.skin](#)[xDT.window.pos](#)[xDT.window.title](#)[xDT.window.icon](#)[xDT.window.onClose](#)[xDT.window.properties](#)**Multiple window stuff**[xDT.minimizeAllWindows](#)[xDT.restoreAllWindows](#)[xDT.arrangeAllWindows](#)[xDT.closeAllWindows](#)[xDT.deleteAllWindows](#)[xDT.positionAllWindows](#)[xDT.innerWindows](#)**Desktop stuff**[xDT.addSkin](#)[xDT.dSkin](#)[xDT.desktop.init](#)[xDT.desktop.skin](#)[xDT.hideDesktop](#)**Other stuff**[xDT.prop](#)[xDT.cbe](#)

## Creating skins

[6 steps to your own skin](#)[The desktop FUN\(\) function](#)[The window skin FUN\(wName\) function](#)

## Others

[Release Notes](#)

## Introduction

[top](#)

Those are the steps to follow in order to start a new x-Desktop project:

**Prepare files & directories**

After download and installation we recommend copying the minimum set of files & directories into an new project

directory. If you don't want any other desktop style than DEFAULT, all x-desktop\_skin\_???.js files are optional, as well as their corresponding xDT/skins/??? directory. The sounds directory is not yet in use, we may add sound events later. The cbe\_slide.js file is not currently in use. However we plan to add animated windows as well, therefore it is included.

```

index.html
startup.html
x-desktop.css
x-desktop.html
cbe_core.js
cbe_event.js
cbe_slide.js
x-desktop_core.js
x-desktop_skin_AEON.js
x-desktop_skin_GNOME.js
x-desktop_skin_KDE.js
x-desktop_skin_NONE.js
x-desktop_skin_OSX.js
x-desktop_skin_OSXA.js
x-desktop_skin_REDHAT.js
x-desktop_skin_SBO.js
x-desktop_skin_WIN.js
x-desktop_skin_XP.js
xDT
!-- docs
    !--- styles.css
    !--- docs.html
!-- files
    !--- alert.html
    !--- confirm.html
    !--- update.html
    !--- sysmsg.html
!-- images
    !-- alert.gif
    !-- confirm.gif
    !-- opac.gif
    !-- blank.gif
!-- skins
    !-- DEFAULT
        !-- DEFAULT.css
        !-- wallpaper_default.jpg
        !-- winclose.gif
        !-- winclose_over.gif
        !-- winleft.gif
        !-- winleft_over.gif
        !-- winmax.gif
        !-- winmax_over.gif
        !-- winmin.gif
        !-- winmin_over.gif
        !-- wintitlebgr.gif
        !-- wintitleleft.gif

```

## 1. Create a Frameset (i.e. index.html)

[top](#)

The reason for using a frameset is because of the different resizing behaviours. Some browsers do reserve a scroll bar, some not. It's a nightmare, but solveable with an outer frameset. However, if you are already in a frameset - go ahead and open the x-desktop.html in it.

```

<HTML>
<HEAD><TITLE>My Desktop</TITLE></HEAD>
<FRAMESET ROWS="100%">
  <FRAME SRC="x-desktop.html" noresize="noresize" scrolling="no">
  <NOFRAMES><BODY></BODY></NOFRAMES>
</FRAMESET>
</HTML>

```

## 2. Create the "base loader" (i.e. x-desktop.html)

[top](#)

You must load all .js except the x-desktop\_skin\_???.js files. Those are optional and depending on the number of desktops you want to use (if any). If you want to start with a different desktop than DEFAULT and jump into another start funtion than "function start()" - make sure you supply certain parameters to the main object initialization function

```

<HTML>
<HEAD>
  <TITLE>Top</TITLE>
<LINK REL="stylesheet" TYPE="text/css" HREF="x-desktop.css">
<script type='text/javascript' src='cbe_core.js'></script>

```

```

<script type='text/javascript' src='cbe_event.js'></script>
<script type='text/javascript' src='cbe_slide.js'></script>
<script type='text/javascript' src='x-desktop_skin_WIN.js'></script>
<script type='text/javascript' src='x-desktop_skin_OSX.js'></script>
<script type='text/javascript' src='x-desktop_skin_OSXA.js'></script>
<script type='text/javascript' src='x-desktop_skin_KDE.js'></script>
<script type='text/javascript' src='x-desktop_skin_GNOME.js'></script>
<script type='text/javascript' src='x-desktop_skin_SBO.js'></script>
<script type='text/javascript' src='x-desktop_skin_XP.js'></script>
<script type='text/javascript' src='x-desktop_skin_AEON.js'></script>
<script type='text/javascript' src='x-desktop_skin_REDHAT.js'></script>
<script type='text/javascript' src='x-desktop_skin_NONE.js'></script>
<script type='text/javascript' src='x-desktop_core.js'></script>

</HEAD>
<BODY>
<script>
var xDT = new xDesktop();
function start() {
  xDT.desktop.init();
  xDT.addWindow('test','Desktop Startup Window: x-desktop.html',500,350,'center',0);
  xDT.url("test",'startup.html');
  xDT.show("test");
}
}
</script>
</BODY>
</HTML>

```

### 3. Create the first application file (i.e. startup.html)

[top](#)

This is the basic outline of any file loaded into a x-Desktop window. However this a recommendation - but loading the appropriate style sheet and using the common classes ensures the compatibility when switching between different desktops.

```

<HTML>
<HEAD><TITLE>Basic Outline</TITLE>
<SCRIPT>
  var xDT = parent.xDT; // reference to the parent xDT object
  document.write('<link rel="stylesheet" type="text/css" href="' +
    xDT.resPath() + 'skins/' + xDT.dSkin() + '/' +
    xDT.dSkin() + '.css' + '>'); // the skin css
</SCRIPT>
</HEAD>
<BODY LEFTMARGIN="0" TOPMARGIN="0" MARGINWIDTH="0" MARGINHEIGHT="0">
  ...put your stuff here ...
</BODY>
</HTML>

```

## Methods

### Desktop initialization

[top](#)

You must name the main x-Desktop object xDT ! You can supply 4 optional parameters, if none are supplied, default is used

1. Resource Path: could be local or remote (http://), don't forget the trailing '/'
2. Default startup desktop: i.e. 'XP', if you want to start with XP
3. Number of internal maximum window objects: default is 50 (min: 20 max: 500)
4. Name of function which will be executed after desktop initialization
5. First call must be xDT.desktop.init();

```
var xDT = new xDesktop('./xDT/', 'DEFAULT', 50, start);
```

### Example default startup

```

var xDT = new xDesktop();
function start() {
  xDT.desktop.init();
  //... now you can add windows, apply methods etc...
  win = xDT.addWindow('windowname','window title',360,190,'center');
  xDT.url(win,'anyfile.html');
  xDT.show(win);
}

```

### xDT.addWindow

[top](#)

The main method to add a new window to the desktop. It may take several parameters and returns the name of the

**successfully created window or 0 in case of error**

```
var win = xDT.addWindow(wName,wTitle,wWidth,wHeight,wPos,wSkin)
windowName: only characters and numbers, no spaces,
             default: "w" + number (in case not name given)
windowTitle: as you like
windowWidth: 1 .... 10000, default 300
windowHeight: 1 .... 10000, default 200
windowPos: "x,y" or
           "center","n","s","e","w","ne","nw","se","sw" or
           'windowname:T|B|L|R:Offset' in case you want the position
             relative to another existing window
           windowname: must be an existing window
           dock to T(op), B(ottom), L(eft) or R(ight) side of window
           Offset in px: space between borders of the 2 windows
windowSkin: if not specified uses same name as Desktop Skin
```

**Example windows**

```
//centered window
win = xDT.addWindow('main','This is the main window',360,190,'center');
//north-east window
win = xDT.addWindow('main','This is the main window',360,190,'ne');
//absolut positioned window, 100px from left and 200 px from top
win = xDT.addWindow('main','This is the main window',360,190,'100,200');
//dock to Left side of window 'menu' with 5 px space
win = xDT.addWindow('main','This is the main window',360,190,'menu:L:5');
```

**xDT.docs**[top](#)

This method allows you to access the local documentation

```
xDT.docs(anker)
anker: direct anker inside the document
```

**Examples**

```
xDT.docs(); // opens top of docu
xDT.docs('addWindow'); // jumps to addWindow doc
```

**xDT.checkUpdate**[top](#)

This method will check your current version against the public release.

```
xDT.checkUpdate()
```

**Example**

```
<INPUT CLASS="xF_button" TYPE="button" VALUE="Update" ONCLICK="xDT.checkUpdate()">
```

**xDT.show**[top](#)

This method will make an existing window visible, sets visibility to "true" and returns "true"

```
xDT.show(windowName)
windowName: Name of existing window
```

**Examples**

```
win = xDT.addWindow('main','This is the main window',360,190,'center');
xDT.show(win)
// or hardcoded name
xDT.addWindow('test','This is a test window',360,190,'center');
xDT.show('test');
if ( xDT.show('test') ) xDT.alert('test is visible');
```

**xDT.hide**[top](#)

This method will make an existing window invisible, sets visibility to "false" and returns "true"

```
xDT.hide(windowName)
windowName: Name of existing window
```

**Examples**

```
win = xDT.addWindow('main','This is the main window',360,190,'center');
xDT.show(win); // show window
```

```
xDT.show(win); // show window
xDT.hide(win); // hide window
// or hardcoded name
xDT.addWindow('test','This is a test window',360,190,'center');
xDT.show('test');
xDT.hide('test');
```

**xDT.popupWindow**[top](#)

This method will make an existing window visible and sets the z-Index to the highest value, keeping it therefor on top of all other windows. Good use to bring a window on top of all others if clicked anywhere in the <body></body> area. Returns the window name.

```
xDT.popupWindow(windowName)
windowName: Name of existing window
```

**Examples**

```
win = xDT.addWindow('main','This is the main window',360,190,'center');
xDT.popupWindow(win)
// or hardcoded name
xDT.addWindow('test','This is a test window',360,190,'center');
xDT.popupWindow('test');
// HTML Example - tricky: create, popup and assign url in one statement
<input type="button" value="google"
  onclick="xDT.url(xDT.popupWindow(xDT.addWindow()),'http://www.google.de')">
// in Body Tag of an application html file to bring window on top
// strip of "xDTiF_" which is the iFrame name,
// followed by the window name, with method xDT.window.name()
<BODY onclick="xDT.popupWindow(xDT.window.name(window.name))">
```

**xDT.url**[top](#)

Load an URL into a created window. If you load a URL of which you have no control, make sure it will not break out of your frame using target "\_top" anywhere. Check external URLs before open them in a x-Desktop window.

```
xDT.url(windowName,URL)
windowName: Name of existing window
URL: any URL or file (if local to relative path)
```

**Examples**

```
xDT.addWindow('google','google.com',360,190,'center');
xDT.url('google','http://www.google.com');
// local file
xDT.url('local','test.html');
```

**xDT.html**[top](#)

Writes any HTML code into an existing window. Please note, that all code will be kept in your "base loader" file, so make sure you load any extra css files etc. in case you are refereing to any specific style sheet classes

```
xDT.html(windowName,HtmlCode)
windowName: Name of existing window
HtmlCode: any valid HTML Code, which could reside inside the <body></body> tag.
```

**Examples**

```
xDT.addWindow('test','This is a test window',360,190,'center');
xDT.html('test','<table align="center"><tr><td>This is a cell</td></tr></table>');
```

**xDT.taskbar**[top](#)

When called with an existing window name it will put the window to the taskbar. If called without an window name it will re-arrange the taskbar. After minimizing a window the taskbar will be automatically re-arranged, so there is normally no need to call this method without argument.

```
xDT.taskbar(windowName)
windowName: Name of existing window
or
xDT.taskbar()
```

**Examples**

```
xDT.addWindow('test','This is a test window',360,190,'center');
xDT.taskbar('test'); // minimize this window and put it in taskbar
```

```
.....
xDT.taskbar(); //re-arrange taskbar
```

**xDT.alert**[top](#)

This Method is an replacement for the message box created by javascript: alert() function. The main difference is that program execution will not stop ! The screen will darken and the user can only push the 'OK' button. Because opacity does not work correctly or stable in other browsers than IE so far, we "fake" the opacity with an image, making the screen "shatter" a little bit, but still readable.

```
xDT.alert(message)
message: any string
```

**Example**

```
xDT.alert('This is a message');
```

**xDT.confirm**[top](#)

This Method is an replacement for the confirm box created by javascript: confirm() function. The main difference is that program execution will not stop ! Therefore you can pass a function to the method in case the user hits 'OK' or 'Cancel'. The screen will darken and the user can only push the 'OK' or 'Cancel' button. Because opacity does not work correctly or stable in other browsers than IE so far, we "fake" the opacity with an image, making the screen "shatter" a little bit, but still readable.

```
xDT.confirm(message,cOK,cCANCEL)
message: any string
cOK: confirm OK function supplied as string or function
cCANCEL: confirm CANCEL function supplied as string or function
```

**Examples**

```
// use string if passing arguments
xDT.confirm('Close this application ?', "CloseApp('now')", "ContinueApp(1)");
// use function if no arguments
xDT.confirm('Close this application ?',parent.CloseApp,parent.ContinueApp);
```

**xDT.quit**[top](#)

This method executes whatever function is passed to an xDT.confirm method. Normally there is no need to call this method, it will be called automatically from the alert/confirm.html file in the resource files/ directory.

```
xDT.quit(param)
param: 'cOK' or 'cCANCEL'
```

**Examples**

```
xDT.quit('cOK'); // executes confirm cOK function, if any
xDT.quit('cCANCEL'); // executes confirm cCANCEL function, if any
```

**xDT.language**[top](#)

Sets or gets the current language. We recommend to use the ISO abbreviations. Currently only "" = default and GERMAN 'de' is being used (alert/confirm.html).

```
xDT.language(language)
language: any string (iso abbreviation recommended)
```

**Examples**

```
xDT.language('de'); // sets the language to 'de'
xDT.language(); // returns 'de'
if (xDT.language() == 'de') xDT.alert('Language is German');
```

**xDT.resPath**[top](#)

Sets or gets the resource path. Don't forget to put a trailing '/' (slash) as last character when changing the resource path.

```
xDT.resPath(resourcePath)
resourcePath: any URL or local/relative path
```

**Examples**

```
//display current resource path
xDT.alert('Current resource path is ' + xDT.resPath());
xDT.resPath('http://intranet/xDT/');
xDT.resPath('./xDT/'); // default
// sets the path to local drive e:/home/...
xDT.resPath('file:///e:/home/xdesktop/xDT/');
```

**xDT.deleteWindow** or **xDT.closeWindow**[top](#)

Deletes/closes a window, freeing the resources for a new one.

```
xDT.deleteWindow(windowName,windowCommand)
windowName: Name of existing window
windowCommand: optionally 'KILL' (any assigned closing function will be ignored)
```

**Examples**

```
// deletes window test
xDT.deleteWindow('test');
// deletes window test does not execute any closing function (fClose property)
xDT.deleteWindow('test','KILL');
```

**xDT.refreshWindow**[top](#)

Refreshes a window (reloads the content) and optionally changes the window skin.

```
xDT.refreshWindow(windowName,windowSkin)
windowName: Name of existing window
windowSkin: a valid skin
```

**Examples**

```
xDT.addWindow('test','This is a test window',360,190,'center','XP');
xDT.refreshWindow('test'); // refresh
xDT.refreshWindow('test',OSXA); // refresh & set skin to OS X Aqua
```

**xDT.positionWindow**[top](#)

Positions a window within the defined borders of the desktop. Normally there's no need to call this method. During drag&drop and/or resize operations it will be done automatically.

```
xDT.positionWindow(windowName)
windowName: Name of existing window
```

**Example**

```
xDT.positionWindow('test'); //position Window 'test'
```

**xDT.maximizeWindow**[top](#)

Maximizes a window to full screen.

```
xDT.maximizeWindow(windowName)
windowName: Name of existing window
```

**Example**

```
xDT.maximizeWindow('test'); //put Window 'test' to full screen
```

**xDT.minimizeWindow**[top](#)

Same as xDT.taskbar

```
xDT.minimizeWindow(windowName)
windowName: Name of existing window
```

**Example**

```
xDT.minimizeWindow('test'); //put Window 'test' to the taskbar
```

**xDT.resizeWindow**[top](#)

Does resize and (optionally) re-position window

```
xDT.resizeWindow(windowName,windowWidth,windowHeight,windowPos)
windowName: Name of existing window
windowWidth: new window width
windowHeight: new window height
windowPos: new window position (optional)
```

**Example**

```
xDT.resizeWindow('test',200,400,'center'); // does resize and reposition window test
```

**xDT.innerWindows**[top](#)

This method hides or shows all windows except the window passed as first argument. This is useful for window moving operations in order to avoid disruption with mouseover events etc.. Except you write a special function, you probably don't need to call this method.

```
xDT.innerWindows(windowName,status)
windowName: Name of existing window
status: 'hide' or 'show'
```

**Examples**

```
xDT.innerWindows('test','hide'); // hide all windows except window 'test'
xDT.innerWindows('test','show'); // show all windows,
                                   leave 'test' as is (normally shown as well)
```

**xDT.positionAllWindows**[top](#)

Positions all windows. Invoked after/during resizing of the x-Desktop by the system itself. Except you write a special function, you probably don't need to call this method.

```
xDT.positionAllWindows
```

**Example**

```
xDT.positionAllWindows(); //re-positions all Windows
```

**xDT.minimizeAllWindows**[top](#)

Does minimize all windows to the taskbar except the one passed as argument

```
xDT.minimizeAllWindows(windowName)
windowName: Name of existing window (optional)
```

**Examples**

```
xDT.minimizeAllWindows(); //minimizes all windows
xDT.minimizeAllWindows('test'); //minimizes all windows except window 'test'
```

**xDT.closeAllWindows** or **xDT.deleteAllWindows**[top](#)

Closes / deletes all windows except the one passed as argument

```
xDT.closeAllWindows(windowName,option)
windowName: name of existing window (optional)
option: could be 'KILL' or any value evaluating true (i.e. 1)
        in order avoid that any closing function will be executed
```

**Examples**

```
// does close all windows with no attached closing function
xDT.closeAllWindows();
// does close all windows with no attached closing function, skipping window 'test'
xDT.closeAllWindows('test');
// does close all windows, skipping window 'test'
xDT.closeAllWindows('test', 1);
```

```
xDT.closeAllWindows( 0000 ,1);
// both do close all windows
xDT.closeAllWindows('', 'KILL');
xDT.closeAllWindows('',1);
```

**xDT.restoreAllWindows**[top](#)

Does restore all minimized windows from taskbar to their previous position

```
xDT.restoreAllWindows
```

**Example**

```
xDT.restoreAllWindows(); // does restore all windows
```

**xDT.arrangeAllWindows**[top](#)

Does arrange all windows except the optional window passed as last argument

```
xDT.arrangeAllWindows(startX,startY,offsetX,offsetY,windowName)
startX: number of px from left (for first window)
startY: number of px from top (for first window)
offsetX: accumulating offset of px from left for following windows...
offsetY: accumulating offset of px from top for following windows...
windowName: ame of existing window (optional), which will be skipped
```

**Examples**

```
// arrange all Windows with a 20x20 offset starting at 0,0 of usable desktop area
xDT.arrangeAllWindows();
xDT.arrangeAllWindows(0,20,0,20); // same 1st example (default values)

// arrange all windows except 'qrmenu', all windows from left 250, no offset and
// from top 20, then accumulating offset of 30 px (250/0 and 20/30)
// this is a vertical alignment of all windows
xDT.arrangeAllWindows(250,20,0,30,'qrmenu')

// do a horizontal alignment, skip window 'qrmenu'
xDT.arrangeAllWindows(250,20,30,0,'qrmenu')
```

**xDT.frameName**[top](#)

Returns the Inline Window Name (frame name) of a x-Desktop window to be used as a target attribute for Forms / and HREFS. The inline window frame is always "xDTiF\_" + windowName. However if you are using this as a target instead of the xDT.url method, xDT.refreshWindow will not work, because the content has not been created through an xDT method.

```
xDT.frameName
```

**Example**

```
xDT.alert(xDT.frameName('test')); // outputs iFrame name 'xDTiF_test'
```

**xDT.preloadImages**[top](#)

Preloads any image supplied as argument. Adds x-desktop resource path if supplied argument is not full qualified ('://') or does not begin with '/'

```
xDT.preloadImages(image1,image2);
```

**Example**

```
xDT.preloadImages('skins/SUSE/wallpaper.jpg',
                  'http://www.yourserver.com/images/test.jpg',
                  '/test.jpg');
```

1st argument will convert to: xDT.resourcePath + 'skins/SUSE/wallpaper.jpg'  
2nd & 3rd image paths remain as they are

**xDT.cbe**[top](#)

This method is a direct interface to the underlying CBE library. You should not use it unless you are familiar with CBE

This method is a direct interface to the underlying CBE library. You should not use it unless you are familiar with CBE. All methods are applied to the named window. Using this function also does increase the risk of many changes in case the CBE lib will change.

```
xDT.cbe(windowName).<cbeMethod>
windowName: Name of existing window
<cbeMethod>: any valid CBE method applicable to a cbe element
```

#### Example

```
xDT.cbe('test').show(); //make window 'test' visible
xDT.cbe('test').hide(); //make window 'test' invisible
```

### xDT.prop

[top](#)

Change any property of a window. Use it very carefully until you are not familiar with the different window properties. Better use the xDT.window.<..> methods instead.

Returns the current property value. Changes to the properties will not execute anything. Values will only be applied by appropriate methods.

```
xDT.prop(windowName,propertyName,propertyValue)
windowName: Name of existing window
propertyName:
    !-- cbe          : object
    !-- innercbe    : object if window content is URL
    !-- wName       : window name
    !-- wTitle      : window title
    !-- wWidthOrg   : original window width (when created)
    !-- wWidth      : current window width
    !-- wHeightOrg  : original window height (when created)
    !-- wHeight     : current window height
    !-- wPos        : window position (i.e. center)
    !-- wX          : window X position
    !-- wY          : window Y position
    !-- wSkin       : window skin
    !-- wIcon       : window icon stat (I1/0, M1/0, X1/0 C1/0)
    !-- wUrl        : window url
    !-- wScroll     : window scrollable (content)
    !-- wHtml       : window html content (if not URL)
    !-- zIndex     : window zIndex
    !-- wIndex      : internal window index (number)
    !-- wStat       : window status (min)
    !-- fClose      : function executed before window closes (if any)
    !-- wVisible    : window visibility status
propertyValue: if passed to the function, Value will be set
```

#### Examples

```
xDT.prop('test','wX'); //returns X coordinate
xDT.prop('test','wTitle'); //returns window Title
//shows alert when 'test' will be closed
xDT.prop('test','fClose',"xDT.alert('Closing test is not allowed')");
```

### xDT.taskbarColor

[top](#)

Sets the taskbar colors

```
xDT.taskbarColor(backgroundcolor,foregroundcolor,bordercolor)
```

#### Example

```
xDT.taskbarColor("#6B8CCE","#ffffff","#000000");
```

### xDT.taskbarStatus

[top](#)

Returns taskbar status

```
xDT.taskbarStatus
```

#### Example

```
if (xDT.taskbarStatus()) xDT.alert('There is a taskbar');
else xDT.alert('There is no taskbar');
```

**xDT.version**[top](#)

Returns the current version

```
xDT.version
```

**Example**

```
xDT.alert(xDT.version());
```

**xDT.addSkin**[top](#)

explanation

```
xDT.addSkin(skinName,distanceTop,distanceBottom)
skinName: name of the skin (x-desktop_skin_*.js)
distanceTop: borderheight top
distanceBottom: borderheight bottom
```

**Examples**

```
xDT.addSkin('XP',0,0); // XP skin no borderheights
// SAP Business One skin with 70px top and 40px bottom distance
xDT.addSkin('SBO',70,40);
```

**xDT.dSkin**[top](#)

Returns the name of the current desktop skin

```
xDT.dSkin
```

**Example**

```
xDT.alert('Current desktop skin is ' + xDT.dSkin());
```

**xDT.playSound**[top](#)

Not yet really implemented

```
xDT.playSound(soundFile)
soundFile: URL to any soundfile (mp3, wav, mid)
```

**Example**

```
xDT.playSound('intro.mp3');
```

**xDT.hideDesktop**[top](#)

Hides Desktop background when doing moving/resizing operations. Needs to be set to true if desktop is a live webpage.

```
xDT.hideDesktop(status)
status: true or false
```

**Examples**

```
xDT.hideDesktop(true); // hides the desktop during window operations
xDT.hideDesktop(false); // leaves the desktop visible during window operations
```

**xDT.sysMessage**[top](#)

Writes a text string to the system message window and displays it.

```
xDT.sysMessage(message,hide)
message: any (HTML) string
hide: optional, true or false
```

**Example**

```
xDT.sysMessage('This is a message'); // does show the message
xDT.sysMessage('This is a message',true); // does not show the message
```

```
xDT.sysmessage('This is a message',true); // does not display the message
```

### xDT.window.name

[top](#)

Returns the window name of the passed window name. Mainly useful for getting the x-Desktop window name from the browser's 'window.name' object property  
Returns 0 if window not found.

```
xDT.window.name(windowName)
windowName: either a existing window name or the browser object 'window.name'
```

#### Examples

```
// strips 'xDTiF_' of the iFrame (window.name) and returns the x-Desktop name
xDT.alert(xDT.window.name(window.name));
//any click in window pops it on top
<BODY onclick="xDT.popupWindow(xDT.window.name(window.name))">
```

### xDT.window.find

[top](#)

Checks if a window is there (regardless if visible or hidden).  
Returns 1 if window has been found otherwise 0

```
xDT.window.find(windowName)
windowName: Name of existing window
```

#### Examples

```
xDT.addWindow('mywin','This is a test window',360,190,'center');
if (xDT.window.find('mywin') ) xDT.alert('window mywin found');
if (xDT.window.find('xxx') ) xDT.alert('window xxx does not exists');
```

### xDT.window.skin

[top](#)

Gets or sets the window skin property. Normally no need to set (is done during addWindow()).  
Returns 0 if no window found, otherwise returns skin name

```
xDT.window.skin(windowName,windowSkin)
windowName: Name of existing window
windowSkin: skin name of loaded x-desktop_skin_???.js
```

#### Example

```
//returns the skin name of the current window
xDT.alert(xDT.window.skin(xDT.window.name(window.name)));
```

### xDT.window.pos

[top](#)

Gets or sets the window position property. Normally no need to set (is done during addWindow()).

```
xDT.window.pos(windowName,windowPos)
windowName: Name of existing window
windowPos: 'center', 'n', etc.. or 'x,y'
```

#### Example

```
xDT.alert(xDT.window.pos('test')); // displays position of window 'test'
```

### xDT.window.title

[top](#)

Gets or sets the window title property. Normally no need to set (is done during addWindow()).

```
xDT.window.title(windowName,windowTitle)
windowName: Name of existing window
windowTitle: any string
```

#### Example

```
xDT.alert(xDT.window.title('test')); // display window title of window 'test'
```

### xDT.window.icon

[top](#)

Gets or sets the window icon property. Is of use for mouseover/mouseout function for the supported window actions. Use in desktop\_SKIN creation function. Does return the window icon property

```
xDT.window.icon(windowName,iconStatus)
windowName: Name of existing window
iconStatus:
  !- info icon      : I1 - mouseover, IO - mousout
  !- minimize icon : M1 - mouseover, MO - mousout
  !- maximize icon : X1 - mouseover, XO - mousout
  !- close icon    : C1 - mouseover, CO - mousout
```

#### Example

```
xDT.alert(xDT.window.icon('test')); // display window icon property value
// by the way this would be the same as xDT.prop('test','wIcon');
```

### xDT.window.onClose

[top](#)

Assigns a function to the window close event. In order to close window, function must return true.

```
xDT.window.onClose(windowName,closeFunction)
windowName: Name of existing window
closeFunction: either a string to eval or function name
```

#### Example

```
//assigns an alert when user tries to close window
//window will not be closed because xDT.alert() does not return true
xDT.window.onClose(xDT.window.name(window.name),
    "xDT.alert('hakuna matata - do not close me')");
//define a closing function OUTSIDE of the content of the closing window
top.onCloseWindow = function() {xDT.alert('I am sad to say goodbye'); return true}
xDT.window.onClose(xDT.window.name(window.name),"top.onCloseWindow()");
// same as
xDT.window.onClose(xDT.window.name(window.name),top.onCloseWindow);
```

### xDT.window.properties

[top](#)

Does display the info window with all properties for any or all windows. If given window name can not be found all window properties will be shown

```
xDT.window.properties(windowName)
windowName: optional existing window name
```

#### Examples

```
//display infos about current window
xDT.window.properties(xDT.window.name(window.name));
//display infos about all windows (including system windows)
xDT.window.properties();
```

### xDT.desktop.init

[top](#)

Must be first called in the start function after creating the object

```
xDT.desktop.init
```

#### Examples

```
xDT.desktop.init(); // initializes the desktop

// complete init routine in base loading file
var xDT = new xDesktop();
function start() {
  xDT.desktop.init();
  //... now you can add windows, apply methods etc...
  ..
  ..
}
```

**xDT.desktop.skin**[top](#)

Does set a new skin and reloads all windows with the new skin. Does change only those window skins equal to the (previous) desktop skin in order to leave any extra windows like selection windows, etc. as they were. Make sure the appropriate x-desktop\_skin\_???.js file is loaded in your baseloading file and the resources (at least the ????.css in ./xDT/skins/???) are available.

```
xDT.desktop.skin(skinName)
skinName: any skin name where a desktop .js exists
```

**Example**

```
xDT.desktop.skin('XP'); // sets the current skin to Win XP style
```

**Creating desktop & window skins****6 steps to create your own skin**[top](#)

You need at least the script file and the skin stylesheet. After you decide for a name (we suggest using capital letters) following following steps:

Let's take an example, we create a FUN skin

- Create the resource directory (default ./xDT/skins/FUN)
- Copy i.e. the ./xDT/skins/DEFAULT/DEFAULT.css to ./xDT/skins/FUN/FUN.css
- Create the script file ./x-desktop\_skin\_FUN.js
- Create at least the 2 mandatory functions "skin\_FUN(wName) " & "desktop\_FUN()"
- Load the script in your "base loader" file (i.e. x-desktop.html):  
<script type='text/javascript' src='x-desktop\_skin\_FUN.js'></script>
- Use the skin when creating the xDT object or later with xDT.desktop.skin('FUN')

**The desktop\_FUN() function**[top](#)

Whatever you do here it will become the desktop background. All operations are done

```
function desktop_FUN() {
  // register the skin
  xDT.addSkin('FUN',0,0);
  //grey bgr, black fg, yellow border
  xDT.taskbarColor("#E6E6E6","#0000FF","#FFFF00");
  //resize to fullscreen
  xDT.cbe("dDesktop").resizeTo(document.cbe.width(),document.cbe.height());
  // here you can do what you want, any <BODY> HTML code allowed
  xDT.cbe("dDesktop").innerHTML(
    '<table width="100%" height="100%"><tr><td></td></tr></table>'
  );
  // set background color and/or background image (2nd argument)
  xDT.cbe("dDesktop").background('#0000FF','');
  //always leave that unchanged
  xDT.cbe("dDesktop").zIndex(0);
  xDT.show("dDesktop");
}
```

**The window skin\_FUN(wName) function**[top](#)

When designing skins keep in mind that you do need some border on top and bottom otherwise the window is not "clickable" for drag and resize operations. First example is without any window icon functions.

```
function skin_FUN(wName) {
  return (
    '<table bgcolor="#0066FF" cellpadding="0" ' +
      ' cellspacing="0" border="0" height="100%" width="100%">' +
    '<tr><td height="20" bgcolor="#FFFFFF">' + // the top border
    '</td></tr>' +
    '<tr><td align="left" bgcolor="#FFFF00" color="#000000" valign="top" ' +
      'width="100%" height="100%" id="' + wName + 'iTD' + '"></td></tr>' +
    '<tr><td height="10" bgcolor="#FFFFFF"></td></tr>' + // the bottom border
    '</table>'
  );
}
```

Now we add the basic icons (I)nfo, (M)inimize, ma(X)imize and (C)lose as links. In addition we add a function that those windows cannot be closed (which may not make sense because it will apply to all windows based on that skin ! In most examples we use xDT.prop(wName,'wIcon','I1') etc..., which is the same, but different method. We recommend using the xDT.window.icon method.

```

function skin_FUN(wName) {
  xDT.window.onClose(wName,"xDT.alert('Please do not close me')");
  return (
    '<table bgcolor="#0066FF" cellpadding="0" ' +
      'cellspacing="0" border="0" height="100%" width="100%">' +
      '<tr><td height="20" bgcolor="#FFFFFF">' + // the top border
      '</td></tr>' +

      "<a href=\"javascript: void(0)\" onmouseover=\"xDT.window.icon('" +
        wName + "','I1')\" onmouseout=\"xDT.window.icon('" +
        wName + "','I0')\">&nbsp;i&nbsp;</a>" +
      "<a href=\"javascript: void(0)\" onmouseover=\"xDT.window.icon('" +
        wName + "','M1')\" onmouseout=\"xDT.window.icon('" +
        wName + "','M0')\">&nbsp;m&nbsp;</a>" +
      "<a href=\"javascript: void(0)\" onmouseover=\"xDT.window.icon('" +
        wName + "','X1')\" onmouseout=\"xDT.window.icon('" +
        wName + "','X0')\">&nbsp;M&nbsp;</a>" +
      "<a href=\"javascript: void(0)\" onmouseover=\"xDT.window.icon('" +
        wName + "','C1')\" onmouseout=\"xDT.window.icon('" +
        wName + "','C0')\">&nbsp;X&nbsp;</a>" +

      '<tr><td align="left" bgcolor="#FFFF00" color="#000000" valign="top" ' +
        'width="100%" height="100%" id="' + wName + 'iTD' + '"></td></tr>' +
      '<tr><td height="10" bgcolor="#FFFFFF"></td></tr>' + // the bottom border
    '</table>'
  );
}

```

## Others

### Release Notes

[top](#)

#### 1.0.5

- new Method `resizeWindow()`
- new Method `preloadImages()`
- Galeon supported
- Camino supported
- `xDT.html()` supports scrolling using current desktop stylesheet
- overcome Mozilla 1.3 bug (minimum window height for iframes)

#### 1.0.2 Initial Development Release