

**NAME**

Xsession - initialize X session

**SYNOPSIS**

Xsession [ *session-type* ]

**DESCRIPTION**

*/etc/X11/Xsession* is a Bourne shell (**sh(1)**) script which is run when an X Window System session is begun by **startx(1)** or a display manager such as **xdm(1)**. (Some display managers only invoke *Xsession* when specifically directed to so by the user; see the documentation for your display manager to find out more.) Administrators unfamiliar with the Bourne shell will likely find the **Xsession.options(5)** configuration file easier to deal with than *Xsession* itself.

*Xsession* is not intended to be invoked directly by the user; to be effective it needs to run in a special environment associated with X server initialization. **startx**, **xdm**, **xinit(1)**, and other similar programs handle this.

By default on a Debian system, *Xsession* is used by both common methods of starting the X Window System, **xdm** (or another X display manager) and **startx**. To change this for **xdm**, edit the ‘DisplayManager\*session’ resource in the */etc/X11/xdm/xdm-config* file — for other display managers, consult their documentation. To stop **startx** from using *Xsession* by default, replace the contents of the */etc/X11/xinit/xinitrc* file.

The *Xsession* script is quite flexible, and extensive customization of the X startup procedure is possible without modifying the script itself. See “CUSTOMIZING THE STARTUP PROCEDURE” below.

**SESSION TYPES**

*Xsession* may optionally be passed a single argument indicating the type of X session to be started. It is up to the display manager to set the argument. To pass *Xsession* an argument from **startx** or **xinit**, */etc/X11/Xsession* (or */etc/X11/xinit/xinitrc*) must be called explicitly with a path, as in **startx /etc/X11/Xsession failsafe**. By default, three different arguments are supported:

**failsafe** invokes a session consisting solely of an **x-terminal-emulator(1)** (no window manager is launched). If the **x-terminal-emulator program cannot** be found, the session exits. The ‘fail-safe’ argument is ignored if there is no ‘allow-failsafe’ line in *Xsession.options*.

**default** produces the same behavior as if no session type argument had been given at all.

*program*

starts *program* if it can be found in the \$PATH. This is usually a session manager or a very featureful window manager. If *program* is not found, the *Xsession* script proceeds with its default behavior. This argument is ignored if there is no ‘allow-user-xsession’ line in *Xsession.options*. (If the administrator does not want users writing their own *.xsession* files, it makes little sense to permit them to specify the names of arbitrary programs to run.) Note that the restriction may be easy to bypass, e.g. by using a *.gnomerc* file instead.

**DEFAULT STARTUP PROCEDURE**

Initially, *Xsession* performs some housekeeping. It declares a set of built-in functions (see “BUILT-IN SHELL FUNCTIONS” below) and variables, then attempts to create a log file for the X session, or append to an existing one. Historically this is called an ‘error’ file, but it catches all sorts of diagnostic output from various X clients run in the user’s session, not just error messages. If it is impossible to write to an error file, the script (and thus the X session) aborts. For convenience, once the error file is successfully opened, *Xsession* reports the fact that the session has started, the invoking username, and the date to the error file. This makes it easier to discern which X session produced a particular line of output in the file.

*Xsession* next confirms that its script directory, *Xsession.d*, exists. If it does not, the script aborts. After the script directory is confirmed to be present, *Xsession* uses **run-parts(1)** to identify files in that directory that should be sourced (executed) in the shell’s environment. Only files named in a certain way are sourced; see the **run-parts** manual page for a description of valid characters in the filename. (This restriction enables the administrator to move experimental or problematic files out of the way of the script but keep them in an obvious place, for instance by renaming them with ‘.old’ or ‘.broken’ appended to the filename.)

## SUPPLIED SCRIPTS

Five shell script portions are supplied by default to handle the details of the session startup procedure.

*/etc/X11/Xsession.d/20x11-common\_process-args*

Arguments are processed as described in “SESSION TYPES” above. The startup program, if one is identified at this point, is merely stored for later reference, and not immediately executed.

*/etc/X11/Xsession.d/30x11-common\_xresources*

X resources are merged. **run-parts** is again used, this time to identify files in the */etc/X11/Xresources* directory that should be processed with ‘*xrdb -merge*’. Next, if the line ‘allow-user-resources’ is present in *Xsession.options*, the user’s *\$HOME/.Xresources* file is merged in the same way.

*/etc/X11/Xsession.d/35x11-common\_xhost-local*

Give access to the X server to the same user on the local host. If the *xhost* command is available, it will use it to allow any process of the same user running on the local host to access the X server.

*/etc/X11/Xsession.d/40x11-common\_xsessionrc*

Source global environment variables. This script will source anything in *\$HOME/.xsessionrc* if the file is present. This allows the user to set global environment variables for their X session, such as locale information.

*/etc/X11/Xsession.d/50x11-common\_determine-startup*

Determine startup program. The X client to launch as the controlling process (the one that, upon exiting, causes the X server to exit as well) is determined next. If a program or failsafe argument was given and is allowed (see above), it is used as the controlling process. Otherwise, if the line ‘allow-user-xsession’ is present in *Xsession.options*, a user -specified session program or script is used. In the latter case, two historically popular names for user X session scripts are searched for: *\$HOME/.xsession* and *\$HOME/.Xsession* (note the difference in case). The first one found is used. If the script is not executable, it is marked to be executed with the Bourne shell interpreter, **sh**. Finally, if none of the above succeeds, the following programs are searched for: */usr/bin/x-session-manager*, */usr/bin/x-window-manager*, and */usr/bin/x-terminal-emulator*. The first one found is used. If none are found, *Xsession* aborts with an error.

*/etc/X11/Xsession.d/90x11-common\_ssh-agent*

Start **ssh-agent(1)**, if needed. If the line ‘use-ssh-agent’ is present in *Xsession.options*, and no SSH agent process appears to be running already, **ssh-agent** is marked to be used to execute the startup program determined previously. **Note:** this functionality may move to the *ssh* package in the future.

*/etc/X11/Xsession.d/99x11-common\_start*

Start the X session. The startup program is executed, inside a Bourne shell if it is not executable, and inside an *ssh-agent* if necessary. The shell’s **exec** command is used to spare a slot in the process table.

## CUSTOMIZING THE STARTUP PROCEDURE

Of course, any of the existing files can be edited in place.

Because the order in which the various scripts in */etc/X11/Xsession.d* are executed is important, files to be added to this directory should have a well-formed name. The following format is recommended:

- \* a two-digit number denoting sequence;
- \* the name of the package providing the script (or ‘custom’ for locally-created scripts);
- \* an underscore;
- \* a description of the script’s basic function, using only characters allowed by **run-parts**.

Here is an example of how one might write a script, named *40custom\_load-xmodmap*, to invoke **xmodmap(1)**:

```
SYSMODMAP="/etc/X11/Xmodmap"
```

```

USRMODMAP="$HOME/.Xmodmap"
if [ -x /usr/bin/X11/xmodmap ]; then
if [ -f "$SYSMODMAP" ]; then
xmodmap "$SYSMODMAP"
fi
fi

if [ -x /usr/bin/X11/xmodmap ]; then
if [ -f "$USRMODMAP" ]; then
xmodmap "$USRMODMAP"
fi
fi

```

Those writing scripts for *Xsession* to execute should avail themselves of its built-in shell functions, described below.

### BUILT-IN SHELL FUNCTIONS

**message** is used for communicating with the user. It is a wrapper for the [echo\(1\)](#) command and relies upon **echo** for its argument processing. This function may be given an arbitrarily long message string, which is formatted to the user's terminal width (breaking lines at whitespace) and sent to standard error. If the *DISPLAY* environment variable is set and the [xmessage\(1\)](#) program is available, **xmessage** is also used to display the message.

**message\_nonl** is used for communicating with the user when a trailing newline is undesirable; it omits a trailing newline from the message text. It otherwise works as **message**.

**errmsg** is used for indicating an error condition and aborting the script. It works as **message**, above, except that after displaying the message, it will exit *Xsession* with status 1.

### ENVIRONMENT

The following environment variables affect the execution of *Xsession*:

#### HOME

specifies the user's home directory; various files are searched for here.

#### TMPDIR

names a default directory for temporary files; if the standard X session error file cannot be opened, this variable is used to locate a place for one.

#### COLUMNS

indicates the width of terminal device in character cells. This value is used for formatting diagnostic messages.

### INPUT FILES

*/etc/X11/Xsession.d/*

is a directory containing Bourne shell scripts to be executed by *Xsession*. Files in this directory are matched using **run-parts** and are **sourced**, not executed in a subshell.

*/etc/X11/Xresources/*

is a directory containing files corresponding to Debian package names, each of which contains system-wide X resource settings for X clients from the corresponding package. The settings are loaded with **xrdb -merge**. Files in this directory are matched using **run-parts**.

*/etc/X11/Xsession.options*

contains configuration options for the */etc/X11/Xsession* script. See [Xsession.options\(5\)](#) for more information.

*\$HOME/.Xresources*

contains X resources specific to the invoking user's environment. The settings are loaded with **xrdb -merge**. Note that *\$HOME/.Xdefaults* is a relic from X Version 10 (and X11R1) days, before app-defaults files were implemented. It has been deprecated for over ten years at the time of this writing. *.Xresources* should be used instead.

*\$HOME/.xsession*

is a sequence of commands invoking X clients (or a session manager such as **xsm(1)**). See the manual page for **xinit** for tips on writing an *.xsession* file.

## OUTPUT FILES

*\$HOME/.xsession-errors*

is where standard output and standard error for *Xsession* script and all X client processes are directed by default.

*\$TMPDIR/filename*

is where the X session error file is placed if *\$HOME/.xsession-errors* cannot be opened. For security reasons, the exact filename is randomly generated by [tempfile\(1\)](#).

## AUTHORS

Stephen Early, Mark Eichin, and Branden Robinson developed Debian's X session handling scripts. Branden Robinson wrote this manual page.

## SEE ALSO

[Xsession.options\(5\)](#), [X\(7\)](#), [run-parts\(1\)](#), [ssh-agent\(1\)](#), [startx\(1\)](#), [tempfile\(1\)](#), [xdm\(1\)](#), [xmessage\(1\)](#), [xmodmap\(1\)](#), [xrdp\(1\)](#), [sh\(1\)](#)