

XDG Base Directory Specification

Waldo Bastian

[<bastian@kde.org>](mailto:bastian@kde.org)

Version 0.6

Table of Contents

[Introduction](#)

[Basics](#)

[Environment variables](#)

[Referencing this specification](#)

Introduction

Various specifications specify files and file formats. This specification defines where these files should be looked for by defining one or more base directories relative to which files should be located.

Basics

The XDG Base Directory Specification is based on the following concepts:

- There is a single base directory relative to which user-specific data files should be written. This directory is defined by the environment variable `$XDG_DATA_HOME`.
- There is a single base directory relative to which user-specific configuration files should be written. This directory is defined by the environment variable `$XDG_CONFIG_HOME`.
- There is a set of preference ordered base directories relative to which data files should be searched. This set of directories is defined by the environment variable `$XDG_DATA_DIRS`.
- There is a set of preference ordered base directories relative to which configuration files should be searched. This set of directories is defined by the environment variable `$XDG_CONFIG_DIRS`.
- There is a single base directory relative to which user-specific non-essential (cached) data should be written. This directory is defined by the environment variable `$XDG_CACHE_HOME`.

Environment variables

`$XDG_DATA_HOME` defines the base directory relative to which user specific data files should be stored. If `$XDG_DATA_HOME` is either not set or empty, a default equal to `$HOME/.local/share` should be used.

`$XDG_CONFIG_HOME` defines the base directory relative to which user specific configuration files should be stored. If `$XDG_CONFIG_HOME` is either not set or empty, a default equal to `$HOME/.config` should be used.

`$XDG_DATA_DIRS` defines the preference-ordered set of base directories to search for data files in addition to the `$XDG_DATA_HOME` base directory. The directories in `$XDG_DATA_DIRS` should be separated with a colon `:`.

If `$XDG_DATA_DIRS` is either not set or empty, a value equal to `/usr/local/share/:/usr/share/` should be used.

`$XDG_CONFIG_DIRS` defines the preference-ordered set of base directories to search for configuration files in addition to the `$XDG_CONFIG_HOME` base directory. The directories in `$XDG_CONFIG_DIRS` should be separated with a colon `:`.

If `$XDG_CONFIG_DIRS` is either not set or empty, a value equal to `/etc/xdg` should be used.

The order of base directories denotes their importance; the first directory listed is the most important. When the same information is defined in multiple places the information defined relative to the more important base directory takes precedent. The base directory defined by `$XDG_DATA_HOME` is considered more important than any of the base directories defined by `$XDG_DATA_DIRS`. The base directory defined by `$XDG_CONFIG_HOME` is considered more important than any of the base directories defined by `$XDG_CONFIG_DIRS`.

`$XDG_CACHE_HOME` defines the base directory relative to which user specific non-essential data files should be stored. If `$XDG_CACHE_HOME` is either not set or empty, a default equal to `$HOME/.cache` should be used.

Referencing this specification

Other specifications may reference this specification by specifying the location of a data file as `$XDG_DATA_DIRS/subdir/filename`. This implies that:

- Such file should be installed to `$datadir/subdir/filename` with `$datadir` defaulting to `/usr/share`.
- A user specific version of the data file may be created in `$XDG_DATA_HOME/subdir/filename`, taking into account the default value for `$XDG_DATA_HOME` if `$XDG_DATA_HOME` is not set.
- Lookups of the data file should search for `./subdir/filename` relative to all base directories specified by `$XDG_DATA_HOME` and `$XDG_DATA_DIRS`. If an environment variable is either not set or empty, its default value as defined by this specification should be used instead.

Specifications may reference this specification by specifying the location of a configuration file as `$XDG_CONFIG_DIRS/subdir/filename`. This implies that:

- Default configuration files should be installed to `$_sysconfdir/xdg/subdir/filename` with `$_sysconfdir` defaulting to `/etc`.
- A user specific version of the configuration file may be created in `$_xdg_config_home/subdir/filename`, taking into account the default value for `$_xdg_config_home` if `$_xdg_config_home` is not set.
- Lookups of the configuration file should search for `./subdir/filename` relative to all base directories indicated by `$_xdg_config_home` and `$_xdg_config_dirs`. If an environment variable is either not set or empty, its default value as defined by this specification should be used instead.

If, when attempting to write a file, the destination directory is non-existent an attempt should be made to create it with permission `0700`. If the destination directory exists already the permissions should not be changed. The application should be prepared to handle the case where the file could not be written, either because the directory was non-existent and could not be created, or for any other reason. In such case it may chose to present an error message to the user.

When attempting to read a file, if for any reason a file in a certain directory is unaccessible, e.g. because the directory is non-existent, the file is non-existent or the user is not authorized to open the file, then the processing of the file in that directory should be skipped. If due to this a required file could not be found at all, the application may chose to present an error message to the user.

A specification that refers to `$_xdg_data_dirs` or `$_xdg_config_dirs` should define what the behaviour must be when a file is located under multiple base directories. It could, for example, define that only the file under the most important base directory should be used or, as another example, it could define rules for merging the information from the different files.