NAME

Glib::BookmarkFile – Parser for bookmark files

SYNOPSIS

```
use Glib;

$date .= $_ while (<DATA>);

$b = Glib::BookmarkFile->new;
$b->load_from_data($data);
$uri = 'file:///some/path/to/a/file.txt';
if ($b->has_item($uri)) {
      $title = $b->get_title($uri);
      $desc  = $b->get_description($uri);

      print "Bookmark for `$uri' ($title):\n";
      print "  $desc\n";
}
0;

__DATA__
<?xml version="1.0" encoding="UTF-8"?>
<xbel version="1.0"
      xmlns:bookmark="http://www.freedesktop.org/standards/desktop-bookmarks"
      xmlns:mime="http://www.freedesktop.org/standards/shared-mime-info">
  <bookmark href="file:///tmp/test-file.txt" added="2006-03-22T18:54:00Z" modif
    <title>Test File</title>
    <desc>Some test file</desc>
    <info>
      <metadata owner="http://freedesktop.org">
        <mime:mime-type type="text/plain"/>
        <bookmark:applications>
          <bookmark:application name="Gedit" exec="gedit %u" timestamp="1143053
        </bookmark:applications>
      </metadata>
    </info>
  </bookmark>
</xbel>
```

DESCRIPTION

Glib::BookmarkFile lets you parse, edit or create files containing lists of bookmarks to resources pointed to by URIs, with some meta-data bound to them, following the Desktop Bookmark Specification. The recent files support inside GTK+ uses this type of files to store the list of recently used files.

The syntax of bookmark files is described in detail in the Desktop Bookmarks Specification, here is a quick summary: bookmark files use a subclass of the XML Bookmark Exchange Language (XBEL) document format, defining meta-data such as the MIME type of the resource pointed by a bookmark, the list of applications that have registered the same URI and the visibility of the bookmark.

METHODS

bookmarkfile = Glib::BookmarkFile->new

`$bookmark_file`**->add_application ($uri,** `$name`**,** `$exec`**)**

- `$uri` (string)
- `$name` (string or undef)
- `$exec` (string or undef)

Adds the application with `$name` and `$exec` to the list of applications that have registered a bookmark for

$uri into $bookmark_file.

Every bookmark inside a Glib::BookmarkFile must have at least an application registered. Each application must provide a name, a command line useful for launching the bookmark, the number of times the bookmark has been registered by the application and the last time the application registered this bookmark.

If $name is undef, the name of the application will be the same returned by **Glib::get_application_name()**; if $exec is undef, the command line will be a composition of the program name as returned by **Glib::get_prgname()** and the "%u" modifier, which will be expanded to the bookmark's URI.

This function will automatically take care of updating the registrations count and timestamping in case an application with the same $name had already registered a bookmark for $uri inside the bookmark file. If no bookmark for $uri is found one is created.

$bookmark_file->add_group ($uri, $group**)**

- $uri (string)
- $group (string)

Adds $group to the list of groups to which the bookmark for $uri belongs to. If no bookmark for $uri is found one is created.

unix timestamp = $bookmark_file**->get_added ($uri)**

- $uri (string)

$bookmark_file**->set_added ($uri,** $value**)**

- $uri (string)
- $value (unix timestamp)

Sets the time the bookmark for $uri was added. If no bookmark for $uri is found one is created.

($exec, $count**,** $stamp**) =** $bookmark_file**->get_app_info ($uri,** $name**)**

- $uri (string)
- $name (string)

Gets the registration information of $name for the bookmark for $uri. See **Glib::BookmarkFile::set_app_info()** for more information about the returned data.

May croak with a Glib::Error in $@ on failure.

$bookmark_file**->set_app_info ($uri,** $name**,** $exec**,** $count**,** $stamp**)**

- $uri (string)
- $name (string)
- $exec (string)
- $count (integer)
- $stamp (unix timestamp)

Sets the meta-data of application $name inside the list of applications that have registered a bookmark for $uri inside $bookmark_file.

You should rarely use this method; use **Glib::BookmarkFile::add_application()** and **Glib::BookmarkFile::remove_application()** instead.

$name can be any UTF-8 encoded string used to identify an application. $exec can have one of these two modifiers: "%f", which will be expanded as the local file name retrieved from the bookmark's URI; "%u", which will be expanded as the bookmark's URI. The expansion is done automatically when retrieving the stored command line using the **Glib::BookmarkFile::get_app_info()** method. $count is the number of times the application has registered the bookmark; if it is < 0, the current registration count will be increased by one, if it is 0, the application with $name will be removed from the list of registered applications. $stamp is the Unix time of the last registration, as returned by **time()**; if it is −1, the current

time will be used.

If you try to remove an application by setting its registration count to zero, and no bookmark for `$uri` is found, `%FALSE` is returned and an exception is fired.

May croak with a Glib::Error in $@ on failure.

list = `$bookmark_file`**->get_applications ($uri)**

- `$uri` (string)

Retrieves the names of the applications that have registered the bookmark for `$uri`.

May croak with a Glib::Error in $@ on failure.

`$bookmark_file`**->get_description ($uri)**

- `$uri` (string)

Gets the description of the bookmark for `$uri`.

May croak with a Glib::Error in $@ on failure.

`$bookmark_file`**->set_description ($uri,** `$description`**)**

- `$uri` (string)
- `$description` (string)

Sets the description of the bookmark for `$uri`. If no bookmark for `$uri` is found one is created.

list = `$bookmark_file`**->get_groups ($uri)**

- `$uri` (string)

Retrieves the list of group names of the bookmark for `$uri`.

May croak with a Glib::Error in $@ on failure.

`$bookmark_file`**->set_groups ($uri, ...)**

- `$uri` (string)
- ... (list) one or more group names

Sets a list of group names for the item with URI `$uri`. Each previously set group name list is removed. If no bookmark for `$uri` is found one is created.

boolean = `$bookmark_file`**->has_application ($uri,** `$name`**)**

- `$uri` (string)
- `$name` (string)

Checks whether the bookmark for `$uri` inside `$bookmark_file` has been registered by application `$name`.

May croak with a Glib::Error in $@ on failure.

boolean = `$bookmark_file`**->has_group ($uri,** `$group`**)**

- `$uri` (string)
- `$group` (string)

Checks whether `$group` appears in the list of groups to which the bookmark for `$uri` belongs to.

May croak with a Glib::Error in $@ on failure.

boolean = `$bookmark_file`**->has_item ($uri)**

- `$uri` (string)

Looks whether the bookmark file has a bookmark for `$uri`.

($href,** `$mime_type`**) =** `$bookmark_file`**->get_icon ($uri)

- `$uri` (string)

Gets the icon of the bookmark for `$uri`.

May croak with a Glib::Error in $@ on failure.

`$bookmark_file->set_icon ($uri, $href, $mime_type)`

- `$uri` (string)
- `$href` (string or undef)
- `$mime_type` (string or undef)

Sets the icon for the bookmark for `$uri`. If `$href` is undef, unsets the currently set icon.

boolean = `$bookmark_file->get_is_private ($uri)`

- `$uri` (string)

May croak with a Glib::Error in $@ on failure.

`$bookmark_file->set_is_private ($uri, $is_private)`

- `$uri` (string)
- `$is_private` (boolean)

`$bookmark_file->load_from_data ($buf)`

- `$buf` (scalar)

Parses a string containing a bookmark file structure.

May croak with a Glib::Error in $@ on failure.

($full_path) = `$bookmark_file->load_from_data_dirs ($file)`

- `$file` (localized file name)

Parses a bookmark file, searching for it inside the data directories. If a file is found, it returns the full path.

May croak with a Glib::Error in $@ on failure.

`$bookmark_file->load_from_file ($file)`

- `$file` (localized file name)

Parses a bookmark file.

May croak with a Glib::Error in $@ on failure.

string = `$bookmark_file->get_mime_type ($uri)`

- `$uri` (string)

Gets the MIME type of the bookmark for `$uri`.

May croak with a Glib::Error in $@ on failure.

`$bookmark_file->set_mime_type ($uri, $mime_type)`

- `$uri` (string)
- `$mime_type` (string)

Sets the MIME type of the bookmark for `$uri`. If no bookmark for `$uri` is found one is created.

unix timestamp = `$bookmark_file->get_modified ($uri)`

- `$uri` (string)

`$bookmark_file->set_modified ($uri, $value)`

- `$uri` (string)
- `$value` (unix timestamp)

Sets the time the bookmark for `$uri` was last modified. If no bookmark for `$uri` is found one is created.

`$bookmark_file->move_item ($old_uri, $new_uri)`

- `$old_uri` (string)
- `$new_uri` (string or undef)

Changes the URI of a bookmark item from `$old_uri` to `$new_uri`. Any existing bookmark for

`$new_uri` will be overwritten. If `$new_uri` is undef, then the bookmark is removed.

May croak with a Glib::Error in $@ on failure.

`$bookmark_file`**->remove_application ($uri,** `$name`**)**

- `$uri` (string)
- `$name` (string)

Removes application registered with `$name` from the list of applications that have registered a bookmark for `$uri` inside `$bookmark_file`.

May croak with a Glib::Error in $@ on failure.

`$bookmark_file`**->remove_group ($uri,** `$group`**)**

- `$uri` (string)
- `$group` (string)

Removes `$group` from the list of groups to which the bookmark for `$uri` belongs to.

May croak with a Glib::Error in $@ on failure.

`$bookmark_file`**->remove_item ($uri)**

- `$uri` (string)

Removes the bookmark for `$uri` from the bookmark file.

May croak with a Glib::Error in $@ on failure.

integer = `$bookmark_file`**->get_size**

Gets the number of bookmarks inside the bookmark file.

`$bookmark_file`**->get_title ($uri,** `$title`**)**

- `$uri` (string)

Gets the title of the bookmark for `$uri`.

May croak with a Glib::Error in $@ on failure.

`$bookmark_file`**->set_title ($uri,** `$title`**)**

- `$uri` (string)
- `$title` (string)

Sets the title of the bookmark for `$uri`. If no bookmark for `$uri` is found one is created.

string = `$bookmark_file`**->to_data**

Returns the bookmark file as a string.

May croak with a Glib::Error in $@ on failure.

`$bookmark_file`**->to_file ($file)**

- `$file` (localized file name)

Saves the contents of a bookmark file into a file. The write operation is guaranteed to be atomic by writing the contents of the bookmark file to a temporary file and then moving the file to the target file.

May croak with a Glib::Error in $@ on failure.

list = `$bookmark_file`**->get_uris**

Returns the URI of all the bookmarks in the bookmark file.

unix timestamp = `$bookmark_file`**->get_visited ($uri)**

- `$uri` (string)

`$bookmark_file`**->set_visited ($uri,** `$value`**)**

- `$uri` (string)
- `$value` (unix timestamp)

Sets the time the bookmark for `$uri` was last visited. If no bookmark for `$uri` is found one is created.

SEE ALSO

Glib

COPYRIGHT