

# Subversion

# Introduction

This extension implements PHP bindings for [» Subversion](#) (SVN), a version control system, allowing PHP scripts to communicate with SVN repositories and working copies without direct command line calls to the `svn` executable.

<b>Warning</b>
This extension is <i>EXPERIMENTAL</i> . The behaviour of this extension?including the names of its functions and any other documentation surrounding this extension?may change without notice in a future release of PHP. This extension should be used at your own risk.

# Installing/Configuring

## Requirements

The Subversion binaries are not necessary to use this extension. However, when compiling the extension, libsvn (the Subversion headers) must be available.

## Installation

Information for installing this PECL extension may be found in the manual chapter titled [Installation of PECL extensions](#). Additional information such as new releases, downloads, source files, maintainer information, and a CHANGELOG, can be located here:   
» <http://pecl.php.net/package/svn>

If `./configure` is having trouble finding the SVN files (for example, Subversion was installed with a different prefix directory), use `./configure --with-svn=$USR_PATH` to specify the directory where the `include/subversion-1/` folder is located.

The DLL for this PECL extension may be downloaded from either the [» PHP Downloads](#) page or from [» http://pecl4win.php.net/](http://pecl4win.php.net/)

Warning
If the extension is compiled against libsvn 1.3, functions that work with working copies will fail when used on working copies created by Subversion 1.4.

## Runtime Configuration

This extension has no configuration directives defined in `php.ini`.

## Resource Types

This extension has no resource types defined.

# Predefined Constants

The constants below are defined by this extension, and will only be available when the extension has either been compiled into PHP or dynamically loaded at runtime.

**SVN\_REVISION\_HEAD** ( [integer](#) )

Magic number (-1) specifying the HEAD revision

Constants usable with [svn\\_auth\\_set\\_parameter\(\)](#)

**SVN\_AUTH\_PARAM\_DEFAULT\_USERNAME** ( [string](#) )

Property for default username to use when performing basic authentication

**SVN\_AUTH\_PARAM\_DEFAULT\_PASSWORD** ( [string](#) )

Property for default password to use when performing basic authentication

**SVN\_AUTH\_PARAM\_NON\_INTERACTIVE** ( [string](#) )

**SVN\_AUTH\_PARAM\_DONT\_STORE\_PASSWORDS** ( [string](#) )

**SVN\_AUTH\_PARAM\_NO\_AUTH\_CACHE** ( [string](#) )

**SVN\_AUTH\_PARAM\_SSL\_SERVER\_FAILURES** ( [string](#) )

**SVN\_AUTH\_PARAM\_SSL\_SERVER\_CERT\_INFO** ( [string](#) )

**SVN\_AUTH\_PARAM\_CONFIG** ( [string](#) )

**SVN\_AUTH\_PARAM\_SERVER\_GROUP** ( [string](#) )

**SVN\_AUTH\_PARAM\_CONFIG\_DIR** ( [string](#) )

**PHP\_SVN\_AUTH\_PARAM\_IGNORE\_SSL\_VERIFY\_ERRORS** ( [string](#) )

Custom property for ignoring SSL cert verification errors

**Filesystem constants**

**SVN\_FS\_CONFIG\_FS\_TYPE** ( [string](#) )

Configuration key that determines filesystem type

**SVN\_FS\_TYPE\_BDB** ( [string](#) )

Filesystem is Berkeley-DB implementation

**SVN\_FS\_TYPE\_FSFS** ( [string](#) )

Filesystem is native-filesystem implementation

**Reserved property constants**

**SVN\_PROP\_REVISION\_DATE** ( [string](#) )

svn:date

**SVN\_PROP\_REVISION\_ORIG\_DATE** ( [string](#) )

svn:original-date

**SVN\_PROP\_REVISION\_AUTHOR** ( [string](#) )

svn:author

**SVN\_PROP\_REVISION\_LOG** ( [string](#) )

svn:log

**Working copy status constants**

**svn\_wc\_status\_none** ( [int](#) )

Status does not exist

**svn\_wc\_status\_unversioned** ( [int](#) )

Item is not versioned in working copy

**svn\_wc\_status\_normal** ( [int](#) )

Item exists, nothing else is happening

**svn\_wc\_status\_added** ( [int](#) )

Item is scheduled for addition

**svn\_wc\_status\_missing** ( [int](#) )

Item is versioned but missing from the working copy

**svn\_wc\_status\_deleted** ( [int](#) )

Item is scheduled for deletion

**svn\_wc\_status\_replaced** ( [int](#) )

Item was deleted and then re-added

**svn\_wc\_status\_modified** ( [int](#) )

Item (text or properties) was modified

**svn\_wc\_status\_merged** ( [int](#) )

Item's local modifications were merged with repository modifications

**svn\_wc\_status\_conflicted** ( [int](#) )

Item's local modifications conflicted with repository modifications

**svn\_wc\_status\_ignored** ( [int](#) )

Item is unversioned but configured to be ignored

**svn\_wc\_status\_obstructed** ( [int](#) )

Unversioned item is in the way of a versioned resource

**svn\_wc\_status\_external** ( [int](#) )

Unversioned path that is populated using svn:externals

**svn\_wc\_status\_incomplete** ( [int](#) )

Directory does not contain complete entries list

### **Node type constants**

**svn\_node\_none** ( [int](#) )

Absent

**svn\_node\_file** ( [int](#) )

File

**svn\_node\_dir** ( [int](#) )

Directory

**svn\_node\_unknown** ( [int](#) )

Something Subversion cannot identify

# SVN Functions

# svn\_add

svn\_add -- Schedules the addition of an item in a working directory

## Description

bool **svn\_add** ( string \$path [, bool \$recursive [, bool \$force ] ] )

Adds the file, directory or symbolic link at *path* to the working directory. The item will be added to the repository the next time you call [svn\\_commit\(\)](#) on the working copy.

## Parameters

*path*

Path of item to add.

Note
Relative paths will be resolved as if the current working directory was the one that contains the PHP binary. To use the calling script's working directory, use <a href="#">realpath()</a> or <a href="#">dirname(__FILE__)</a> .

*recursive*

If item is directory, whether or not to recursively add all of its contents. Default is **TRUE**

*force*

If true, Subversion will recurse into already versioned directories in order to add unversioned files that may be hiding in those directories. Default is **FALSE**

## Return Values

Returns **TRUE** on success or **FALSE** on failure.

## Notes

Warning
This function is <i>EXPERIMENTAL</i> . The behaviour of this function, its name, and surrounding documentation may change without notice in a future release of PHP. This function should be used at your own risk.

## Examples



### Example #1 - [svn\\_add\(\)](#) example

In a working directory where **svn status** returns:

```
$ svn status
?      foobar.txt
```

...this code:

```
<?php
svn_add( 'foobar.txt' );
?>
```

...will schedule *foobar.txt* for addition into the repository.

## See Also

- [» SVN documentation on svn add](#)

# svn\_auth\_get\_parameter

svn\_auth\_get\_parameter -- Retrieves authentication parameter

## Description

string **svn\_auth\_get\_parameter** ( string \$key )

Retrieves authentication parameter at *key*. For a list of valid keys and their meanings, consult the [authentication constants list](#).

## Parameters

*key*

String key name. Use the [authentication constants](#) defined by this extension to specify a key.

## Return Values

Returns the string value of the parameter at *key*; returns **NULL** if parameter does not exist.

## Notes

Warning
This function is <i>EXPERIMENTAL</i> . The behaviour of this function, its name, and surrounding documentation may change without notice in a future release of PHP. This function should be used at your own risk.

## See Also

- [svn\\_auth\\_set\\_parameter\(\)](#)
- [Authentication constants](#)

# svn\_auth\_set\_parameter

svn\_auth\_set\_parameter -- Sets an authentication parameter

## Description

**void** **svn\_auth\_set\_parameter** ( string \$key, string \$value )

Sets authentication parameter at *key* to *value*. For a list of valid keys and their meanings, consult the [authentication constants list](#).

## Parameters

*key*

String key name. Use the [authentication constants](#) defined by this extension to specify a key.

*value*

String value to set to parameter at key. Format of value varies with the parameter.

## Return Values

No value is returned.

## Notes

### Warning

This function is *EXPERIMENTAL*. The behaviour of this function, its name, and surrounding documentation may change without notice in a future release of PHP. This function should be used at your own risk.

## Examples

### Example #2 - Default authentication example

This example configures SVN so that the default username to use is 'Bob' and the default password is 'abc123':

```
<?php
svn_auth_set_parameter(SVN_AUTH_PARAM_DEFAULT_USERNAME, 'Bob');
svn_auth_set_parameter(SVN_AUTH_PARAM_DEFAULT_PASSWORD, 'abc123');
?>
```

## See Also

- [svn\\_auth\\_get\\_parameter\(\)](#)
- [Authentication constants](#)

# svn\_cat

svn\_cat -- Returns the contents of a file in a repository

## Description

string **svn\_cat** ( string \$repos\_url [, int \$revision\_no ] )

Returns the contents of the URL *repos\_url* to a file in the repository, optionally at revision number *revision\_no*.

## Parameters

*repos\_url*  
String URL path to item in a repository.

*revision\_no*  
Integer revision number of item to retrieve, default is the HEAD revision.

## Return Values

Returns the string contents of the item from the repository on success, and **FALSE** on failure.

## Notes

### Warning

This function is *EXPERIMENTAL*. The behaviour of this function, its name, and surrounding documentation may change without notice in a future release of PHP. This function should be used at your own risk.

## Examples

### Example #3 - Basic example

This example retrieves the contents of a file at revision 28:

```
<?php
$contents = svn_cat('http://www.example.com/svnroot/calc/gui.c', 28)
?>
```

## See Also

- `svn_list()`
- [» SVN documentation on svn cat](#)

# svn\_checkout

svn\_checkout -- Checks out a working copy from the repository

## Description

bool **svn\_checkout** ( string \$repos, string \$targetpath [, int \$revision ] )

Checks out a working copy from the repository at *repos* to *targetpath* at revision *revision*.

## Parameters

*repos*

String URL path to directory in repository to check out.

*targetpath*

String local path to directory to check out in to

Note
Relative paths will be resolved as if the current working directory was the one that contains the PHP binary. To use the calling script's working directory, use <a href="#">realpath()</a> or <code>dirname(__FILE__)</code> .

*revision*

Integer revision number of repository to check out. Default is HEAD, the most recent revision.

## Return Values

Returns **TRUE** on success or **FALSE** on failure.

## Notes

Warning
This function is <i>EXPERIMENTAL</i> . The behaviour of this function, its name, and surrounding documentation may change without notice in a future release of PHP. This function should be used at your own risk.

## Examples

#### Example #4 - Basic example

This example demonstrates how to check out a directory from a repository to a directory named calc:

```
<?php
svn_checkout('http://www.example.com/svnroot/calc/trunk', dirname(__FILE__)
. '/calc');
?>
```

The *dirname(\_\_FILE\_\_)* call is necessary in order to convert the calc relative path into an absolute one. If calc exists, you can also use [realpath\(\)](#) to retrieve an absolute path.

#### See Also

- [svn\\_add\(\)](#)
- [svn\\_commit\(\)](#)
- [svn\\_status\(\)](#)
- [svn\\_update\(\)](#)
- [» SVN documentation on svn checkout](#)



# svn\_cleanup

svn\_cleanup -- Recursively cleanup a working copy directory, finishing incomplete operations and removing locks

## Description

bool **svn\_cleanup** ( string \$workingdir )

Recursively cleanup working copy directory *workingdir*, finishing any incomplete operations and removing working copy locks. Use when a working copy is in limbo and needs to be usable again.

## Parameters

*workingdir*

String path to local working directory to cleanup

Note
Relative paths will be resolved as if the current working directory was the one that contains the PHP binary. To use the calling script's working directory, use <a href="#">realpath()</a> or <code>dirname(__FILE__)</code> .

## Return Values

Returns **TRUE** on success or **FALSE** on failure.

## Notes

Warning
This function is <i>EXPERIMENTAL</i> . The behaviour of this function, its name, and surrounding documentation may change without notice in a future release of PHP. This function should be used at your own risk.

## Examples

### Example #5 - Basic example

This example demonstrates clean up of a working copy in a directory named help-me:

```
<?php
svn_cleanup(realpath('help-me')) ;
?>
```

The [realpath\(\)](#) call is necessary due to SVN's quirky handling of relative paths.

### See Also

- [update\(\)](#)
- [» SVN documentation on svn cleanup](#)

# svn\_client\_version

svn\_client\_version -- Returns the version of the SVN client libraries

## Description

string **svn\_client\_version** ( void )

Returns the version of the SVN client libraries

## Return Values

String version number, usually in form of x.y.z.

## Notes

Warning
This function is <i>EXPERIMENTAL</i> . The behaviour of this function, its name, and surrounding documentation may change without notice in a future release of PHP. This function should be used at your own risk.

## Examples

Example #6 - Basic example
<pre>&lt;?php echo svn_client_version(); ?&gt;</pre> <p>The above example will output something similar to:</p> <pre>1.3.1</pre>

# svn\_commit

svn\_commit -- Sends changes from the local working copy to the repository

## Description

array **svn\_commit** ( string *\$log*, array *\$targets* [, bool *\$dontrecurse* ] )

Commits changes made in the local working copy files enumerated in the *targets* array to the repository, with the log message *log*. Directories in the *targets* array will be recursively committed unless *dontrecurse* is set to true.

### Note

This function does not have any parameters for specifying authentication, so a username and password must be set using [svn\\_auth\\_set\\_parameter\(\)](#)

## Parameters

*log*

String log text to commit

*targets*

Array of local paths of files to be committed

### Warning

This parameter must be an array, a string for a single target is not acceptable.

### Note

Relative paths will be resolved as if the current working directory was the one that contains the PHP binary. To use the calling script's working directory, use [realpath\(\)](#) or [dirname\(\\_\\_FILE\\_\\_\)](#).

*dontrecurse*

Boolean flag to disable recursive committing of directories in the *targets* array. Default is **FALSE**.

## Return Values

Returns array in form of:

```
array(  
    0 => integer revision number of commit  
    1 => string ISO 8601 date and time of commit  
    2 => name of committer  
)
```

Returns **FALSE** on failure.

## Notes

### Warning

This function is *EXPERIMENTAL*. The behaviour of this function, its name, and surrounding documentation may change without notice in a future release of PHP. This function should be used at your own risk.

## Examples

### Example #7 - Basic example

This example commits the calculator directory to a repository, using the username Bob and the password abc123 (hopefully, his password is stronger):

```
<?php  
svn_auth_set_parameter(SVN_AUTH_PARAM_DEFAULT_USERNAME, 'Bob');  
svn_auth_set_parameter(SVN_AUTH_PARAM_DEFAULT_PASSWORD, 'abc123');  
var_dump(svn_commit('Log message of Bob\'s commit',  
    array(realpath('calculator'))));  
?>
```

The above example will output:

```
array(  
    0 => 1415,  
    1 => '2007-05-26T01:44:28.453125Z',  
    2 => 'Bob'  
)
```

## See Also

- [svn\\_auth\\_set\\_parameter\(\)](#)
- [» SVN documentation on svn commit](#)

# svn\_diff

svn\_diff -- Recursively diffs two paths

## Description

array **svn\_diff** ( string \$path1, int \$rev1, string \$path2, int \$rev2 )

Recursively diffs two paths, *path1* and *path2*.

Note
This is not a general-purpose diff utility. Only local files that are versioned may be diffed: other files will fail.

## Parameters

*path1*

First path to diff. This can be a URL to a file/directory in an SVN repository or a local file/directory path.

Note
Relative paths will be resolved as if the current working directory was the one that contains the PHP binary. To use the calling script's working directory, use <a href="#">realpath()</a> or <code>dirname(__FILE__)</code> .

Warning
If a local file path has only backslashes and no forward slashes, this extension will fail to find the path. Always replace all backslashes with forward slashes when using this function.

*rev1*

First path's revision number. Use **SVN\_REVISION\_HEAD** to specify the most recent revision.

*path2*

Second path to diff. See *path1* for description.

*rev2*

Second path's revision number. See *rev1* for description.

## Return Values

Returns an array-list consisting of two streams: the first is the diff output and the second contains error stream output. The streams can be read using [fread\(\)](#). Returns **FALSE** or **NULL** on error.

The diff output will, by default, be in the form of Subversion's custom unified diff format, but an [» external diff engine](#) may be used depending on Subversion's configuration.

## Notes

### Warning

This function is *EXPERIMENTAL*. The behaviour of this function, its name, and surrounding documentation may change without notice in a future release of PHP. This function should be used at your own risk.

## Examples

### Example #8 - Basic example

This example demonstrates the basic usage of this function, and the retrieval of contents from the stream:

```
<?php
list($diff, $errors) = svn_diff(
    'http://www.example.com/svnroot/trunk/foo', SVN_REVISION_HEAD,
    'http://www.example.com/svnroot/branches/dev/foo', SVN_REVISION_HEAD
);
if (!$diff) exit;
$contents = '';
while (!feof($diff)) {
    $contents .= fread($diff, 8192);
}
fclose($diff);
fclose($errors);
var_dump($contents);
?>
```

The above example will output:

```
Index: http://www.example.com/svnroot/trunk/foo
=====
--- http://www.example.com/svnroot/trunk/foo      (.../foo) (revision 23)
+++ http://www.example.com/svnroot/branches/dev/foo (.../foo) (revision 27)
// further diff output
```

### Example #9 - Diffing two revisions of a repository path

This example implements a wrapper function that allows a user to easily diff two revisions of the same item using an external repository path (the default syntax is somewhat verbose):

```
<?php
function svn_diff_same_item($path, $rev1, $rev2) {
    return svn_diff($path, $rev1, $path, $rev2);
}
?>
```

### Example #10 - Portably diffing two local files

This example implements a wrapper function that portably diffs two local files, compensating for the [realpath\(\)](#) fix and the backslashes bug:

```
<?php
function svn_diff_local($path1, $rev1, $path2, $rev2) {
    $path1 = str_replace('\\', '/', realpath($path1));
    $path2 = str_replace('\\', '/', realpath($path2));
    return svn_diff($path1, $rev1, $path2, $rev2);
}
?>
```

## See Also

- [» SVN documentation on svn diff](#)



# svn\_fs\_abort\_txn

svn\_fs\_abort\_txn -- Abort a transaction, returns true if everything is ok, false otherwise

## Description

bool **svn\_fs\_abort\_txn** ( resource \$txn )

Warning
This function is currently not documented; only its argument list is available.

Abort a transaction, returns true if everything is ok, false otherwise

## Notes

Warning
This function is <i>EXPERIMENTAL</i> . The behaviour of this function, its name, and surrounding documentation may change without notice in a future release of PHP. This function should be used at your own risk.

# svn\_fs\_apply\_text

svn\_fs\_apply\_text -- Creates and returns a stream that will be used to replace

## Description

resource **svn\_fs\_apply\_text** ( resource \$root, string \$path )

<b>Warning</b>
This function is currently not documented; only its argument list is available.

Creates and returns a stream that will be used to replace

## Notes

<b>Warning</b>
This function is <i>EXPERIMENTAL</i> . The behaviour of this function, its name, and surrounding documentation may change without notice in a future release of PHP. This function should be used at your own risk.

# svn\_fs\_begin\_txn2

svn\_fs\_begin\_txn2 -- Create a new transaction

## Description

resource **svn\_fs\_begin\_txn2** ( resource \$repos, int \$rev )

<b>Warning</b>
This function is currently not documented; only its argument list is available.

Create a new transaction

## Notes

<b>Warning</b>
This function is <i>EXPERIMENTAL</i> . The behaviour of this function, its name, and surrounding documentation may change without notice in a future release of PHP. This function should be used at your own risk.

# svn\_fs\_change\_node\_prop

svn\_fs\_change\_node\_prop -- Return true if everything is ok, false otherwise

## Description

bool **svn\_fs\_change\_node\_prop** ( resource \$root, string \$path, string \$name, string \$value )

<b>Warning</b>
This function is currently not documented; only its argument list is available.

Return true if everything is ok, false otherwise

## Notes

<b>Warning</b>
This function is <i>EXPERIMENTAL</i> . The behaviour of this function, its name, and surrounding documentation may change without notice in a future release of PHP. This function should be used at your own risk.

# svn\_fs\_check\_path

svn\_fs\_check\_path -- Determines what kind of item lives at path in a given repository  
fsroot

## Description

int **svn\_fs\_check\_path** ( resource \$fsroot, string \$path )

<b>Warning</b>
This function is currently not documented; only its argument list is available.

Determines what kind of item lives at path in a given repository fsroot

## Notes

<b>Warning</b>
This function is <i>EXPERIMENTAL</i> . The behaviour of this function, its name, and surrounding documentation may change without notice in a future release of PHP. This function should be used at your own risk.

# svn\_fs\_contents\_changed

svn\_fs\_contents\_changed -- Return true if content is different, false otherwise

## Description

bool **svn\_fs\_contents\_changed** ( resource \$root1, string \$path1, resource \$root2, string \$path2 )

<b>Warning</b>
This function is currently not documented; only its argument list is available.

Return true if content is different, false otherwise

## Notes

<b>Warning</b>
This function is <i>EXPERIMENTAL</i> . The behaviour of this function, its name, and surrounding documentation may change without notice in a future release of PHP. This function should be used at your own risk.

# svn\_fs\_copy

svn\_fs\_copy -- Copies a file or a directory, returns true if all is ok, false otherwise

## Description

bool **svn\_fs\_copy** ( resource \$from\_root, string \$from\_path, resource \$to\_root, string \$to\_path )

<b>Warning</b>
This function is currently not documented; only its argument list is available.

Copies a file or a directory, returns true if all is ok, false otherwise

## Notes

<b>Warning</b>
This function is <i>EXPERIMENTAL</i> . The behaviour of this function, its name, and surrounding documentation may change without notice in a future release of PHP. This function should be used at your own risk.

# svn\_fs\_delete

svn\_fs\_delete -- Deletes a file or a directory, return true if all is ok, false otherwise

## Description

bool **svn\_fs\_delete** ( resource \$root, string \$path )

Warning
This function is currently not documented; only its argument list is available.

Deletes a file or a directory, return true if all is ok, false otherwise

## Notes

Warning
This function is <i>EXPERIMENTAL</i> . The behaviour of this function, its name, and surrounding documentation may change without notice in a future release of PHP. This function should be used at your own risk.



# svn\_fs\_dir\_entries

svn\_fs\_dir\_entries -- Enumerates the directory entries under path; returns a hash of dir names to file type

## Description

array **svn\_fs\_dir\_entries** ( resource \$fsroot, string \$path )

<b>Warning</b>
This function is currently not documented; only its argument list is available.

Enumerates the directory entries under path; returns a hash of dir names to file type

## Notes

<b>Warning</b>
This function is <i>EXPERIMENTAL</i> . The behaviour of this function, its name, and surrounding documentation may change without notice in a future release of PHP. This function should be used at your own risk.

# svn\_fs\_file\_contents

svn\_fs\_file\_contents -- Returns a stream to access the contents of a file from a given version of the fs

## Description

resource **svn\_fs\_file\_contents** ( resource \$fsroot, string \$path )

Warning
This function is currently not documented; only its argument list is available.

Returns a stream to access the contents of a file from a given version of the fs

## Notes

Warning
This function is <i>EXPERIMENTAL</i> . The behaviour of this function, its name, and surrounding documentation may change without notice in a future release of PHP. This function should be used at your own risk.

# svn\_fs\_file\_length

svn\_fs\_file\_length -- Returns the length of a file from a given version of the fs

## Description

int **svn\_fs\_file\_length** ( resource \$fsroot, string \$path )

<b>Warning</b>
This function is currently not documented; only its argument list is available.

Returns the length of a file from a given version of the fs

## Notes

<b>Warning</b>
This function is <i>EXPERIMENTAL</i> . The behaviour of this function, its name, and surrounding documentation may change without notice in a future release of PHP. This function should be used at your own risk.

# svn\_fs\_is\_dir

svn\_fs\_is\_dir -- Return true if the path points to a directory, false otherwise

## Description

bool **svn\_fs\_is\_dir** ( resource \$root, string \$path )

Warning
This function is currently not documented; only its argument list is available.

Return true if the path points to a directory, false otherwise

## Notes

Warning
This function is <i>EXPERIMENTAL</i> . The behaviour of this function, its name, and surrounding documentation may change without notice in a future release of PHP. This function should be used at your own risk.

# svn\_fs\_is\_file

svn\_fs\_is\_file -- Return true if the path points to a file, false otherwise

## Description

bool **svn\_fs\_is\_file** ( resource \$root, string \$path )

Warning
This function is currently not documented; only its argument list is available.

Return true if the path points to a file, false otherwise

## Notes

Warning
This function is <i>EXPERIMENTAL</i> . The behaviour of this function, its name, and surrounding documentation may change without notice in a future release of PHP. This function should be used at your own risk.

# svn\_fs\_make\_dir

svn\_fs\_make\_dir -- Creates a new empty directory, returns true if all is ok, false otherwise

## Description

bool **svn\_fs\_make\_dir** ( resource \$root, string \$path )

<b>Warning</b>
This function is currently not documented; only its argument list is available.

Creates a new empty directory, returns true if all is ok, false otherwise

## Notes

<b>Warning</b>
This function is <i>EXPERIMENTAL</i> . The behaviour of this function, its name, and surrounding documentation may change without notice in a future release of PHP. This function should be used at your own risk.

# svn\_fs\_make\_file

svn\_fs\_make\_file -- Creates a new empty file, returns true if all is ok, false otherwise

## Description

bool **svn\_fs\_make\_file** ( resource \$root, string \$path )

<b>Warning</b>
This function is currently not documented; only its argument list is available.

Creates a new empty file, returns true if all is ok, false otherwise

## Notes

<b>Warning</b>
This function is <i>EXPERIMENTAL</i> . The behaviour of this function, its name, and surrounding documentation may change without notice in a future release of PHP. This function should be used at your own risk.

# svn\_fs\_node\_created\_rev

svn\_fs\_node\_created\_rev -- Returns the revision in which path under fsroot was created

## Description

int **svn\_fs\_node\_created\_rev** ( resource \$fsroot, string \$path )

<b>Warning</b>
This function is currently not documented; only its argument list is available.

Returns the revision in which path under fsroot was created

## Notes

<b>Warning</b>
This function is <i>EXPERIMENTAL</i> . The behaviour of this function, its name, and surrounding documentation may change without notice in a future release of PHP. This function should be used at your own risk.



# svn\_fs\_node\_prop

svn\_fs\_node\_prop -- Returns the value of a property for a node

## Description

string **svn\_fs\_node\_prop** ( resource \$fsroot, string \$path, string \$propname )

<b>Warning</b>
This function is currently not documented; only its argument list is available.

Returns the value of a property for a node

## Notes

<b>Warning</b>
This function is <i>EXPERIMENTAL</i> . The behaviour of this function, its name, and surrounding documentation may change without notice in a future release of PHP. This function should be used at your own risk.

# svn\_fs\_props\_changed

svn\_fs\_props\_changed -- Return true if props are different, false otherwise

## Description

bool **svn\_fs\_props\_changed** ( resource \$root1, string \$path1, resource \$root2, string \$path2 )

Warning
This function is currently not documented; only its argument list is available.

Return true if props are different, false otherwise

## Notes

Warning
This function is <i>EXPERIMENTAL</i> . The behaviour of this function, its name, and surrounding documentation may change without notice in a future release of PHP. This function should be used at your own risk.

# svn\_fs\_revision\_prop

svn\_fs\_revision\_prop -- Fetches the value of a named property

## Description

string **svn\_fs\_revision\_prop** ( resource *\$fs*, int *\$revnum*, string *\$propname* )

<b>Warning</b>
This function is currently not documented; only its argument list is available.

Fetches the value of a named property

## Notes

<b>Warning</b>
This function is <i>EXPERIMENTAL</i> . The behaviour of this function, its name, and surrounding documentation may change without notice in a future release of PHP. This function should be used at your own risk.

# svn\_fs\_revision\_root

svn\_fs\_revision\_root -- Get a handle on a specific version of the repository root

## Description

resource **svn\_fs\_revision\_root** ( resource `$fs`, int `$revnum` )

<b>Warning</b>
This function is currently not documented; only its argument list is available.

Get a handle on a specific version of the repository root

## Notes

<b>Warning</b>
This function is <i>EXPERIMENTAL</i> . The behaviour of this function, its name, and surrounding documentation may change without notice in a future release of PHP. This function should be used at your own risk.

# svn\_fs\_txn\_root

svn\_fs\_txn\_root -- Creates and returns a transaction root

## Description

resource **svn\_fs\_txn\_root** ( resource \$txn )

<b>Warning</b>
This function is currently not documented; only its argument list is available.

Creates and returns a transaction root

## Notes

<b>Warning</b>
This function is <i>EXPERIMENTAL</i> . The behaviour of this function, its name, and surrounding documentation may change without notice in a future release of PHP. This function should be used at your own risk.

# svn\_fs\_youngest\_rev

svn\_fs\_youngest\_rev -- Returns the number of the youngest revision in the filesystem

## Description

int **svn\_fs\_youngest\_rev** ( resource *\$fs* )

<b>Warning</b>
This function is currently not documented; only its argument list is available.

Returns the number of the youngest revision in the filesystem

## Notes

<b>Warning</b>
This function is <i>EXPERIMENTAL</i> . The behaviour of this function, its name, and surrounding documentation may change without notice in a future release of PHP. This function should be used at your own risk.

# svn\_import

svn\_import -- Imports an unversioned path into a repository

## Description

bool **svn\_import** ( string \$path, string \$url, bool \$nonrecursive )

Commits unversioned *path* into repository at *url*. If *path* is a directory and *nonrecursive* is **FALSE**, the directory will be imported recursively.

## Parameters

*path*

Path of file or directory to import.

### Note

Relative paths will be resolved as if the current working directory was the one that contains the PHP binary. To use the calling script's working directory, use [realpath\(\)](#) or [dirname\(\\_\\_FILE\\_\\_\)](#).

*url*

Repository URL to import into.

*nonrecursive*

Whether or not to refrain from recursively processing directories.

## Return Values

Returns **TRUE** on success or **FALSE** on failure.

## Notes

### Warning

This function is *EXPERIMENTAL*. The behaviour of this function, its name, and surrounding documentation may change without notice in a future release of PHP. This function should be used at your own risk.

## Examples

### Example #11 - Basic example

This example demonstrates a basic use-case of this function. To import a directory named new-files into the repository at <http://www.example.com/svnroot/incoming/abc>, use:

```
<?php
svn_import(realpath('new-files'),
'http://www.example.com/svnroot/incoming/abc', false);
?>
```

### See Also

- [svn\\_add\(\)](#)
- [» SVN documentation for svn import](#)



# svn\_log

svn\_log -- Returns the commit log messages of a repository URL

## Description

array **svn\_log** ( string \$repos\_url [, int \$revision\_no ] )

[svn\\_log\(\)](#) returns the complete history of the item at the repository URL *repos\_url*, or the history of a specific revision if *revision\_no* is set. This function is equivalent to **svn log --verbose -r \$revision\_no \$repos\_url**.

### Warning

For repositories with large histories, the output may be quite large (one array item for every revision of the item). This function does not support the **--limit NUM** flag, nor does it support revision ranges ( **revision\_no** must be an integer).

## Parameters

*repos\_url*

Repository URL of the item to retrieve log history from.

*revision\_no*

Revision number of the log to retrieve. Use **SVN\_REVISION\_HEAD** to retrieve the log for the most recent revision.

## Return Values

On success, this function returns an array file listing in the format of:

```
[0] => Array, ordered most recent (highest) revision first
(
    [rev] => integer revision number
    [author] => string author name
    [msg] => string log message
    [date] => string date formatted per ISO 8601, i.e. date('c')
    [paths] => Array, describing changed files
    (
        [0] => Array
        (
            [action] => string letter signifying change
            [path] => absolute repository path of changed file
        )
        [1] => ...
    )
)
[1] => ...
```

### Note

The output will always be a numerically indexed array of arrays, even when there are none or only one log message(s).

The value of *action* is a subset of the [» status output in the first column](#), where possible values are:

### Actions

Letter	Description
M	Item/props was modified
A	Item was added
D	Item was deleted
R	Item was replaced

If no changes were made to the item, an empty array is returned.

### Notes

#### Warning

This function is *EXPERIMENTAL*. The behaviour of this function, its name, and surrounding documentation may change without notice in a future release of PHP. This function should be used at your own risk.

### Examples

#### Example #12 - [svn\\_log\(\)](#) example

```
<?php
print_r( svn_log('http://www.example.com/', 23) );
?>
```

The above example will output something similar to:

```
Array
(
    [0] => Array
        (
```

```

[rev] => 23
[author] => 'joe'
[msg] => 'Add cheese and salami to our sandwich.'
[date] => '2007-04-06T16:00:27-04:00'
[paths] => Array
(
    [0] => Array
        (
            [action] => 'M'
            [path] => '/sandwich.txt'
        )
)
)
)

```

### Example #13 - Simulating --limit with svn and [svn\\_log\(\)](#)

This sample function simulates the **--limit** switch by using the SVN executable to return a list of revisions, which are then losslessly accessed using [svn\\_log\(\)](#).

#### Note

This function will perform a total of *limit* + 1 requests: the first request to pull the required revisions, and each one after to retrieve the log item for that request.

```

<?php
/**
 * Retrieves the last $limit log entries.
 * @param $repos_url Repository URL of item to get logs for
 * @param $limit Integer limit of items
 */
function svn_log_limit($repos_url, $limit) {
    $limit = (int) $limit;
    if ($limit <= 0) return array();
    // -q flag used to prevent server from sending log messages
    $output = shell_exec("svn log -q --limit $limit $repos_url");
    preg_match_all('/^r(\d+) /m', $output, $matches);
    $ret = array();
    foreach ($matches[1] as $rev) {
        $log = svn_log($repos_url, (int) $rev);
        $ret[] = $log[0]; // log is only one item long
    }
    return $ret;
}
?>

```

**See Also**

- [» SVN documentation on svn log](#)

# svn\_ls

svn\_ls -- Returns list of directory contents in repository URL, optionally at revision number

## Description

array **svn\_ls** ( string \$repos\_url [, int \$revision\_no ] )

This function queries the repository URL and returns a list of files and directories, optionally from a specific revision. This is equivalent to **svn list \$repos\_url[@\$revision\_no]**

### Note

This function does not work with working copies. *repos\_url* *must* be a repository URL.

## Parameters

*url*

URL of the repository, eg. **http://www.example.com/svnroot**. To access a local Subversion repository via filesystem, use the file URI scheme, eg. **file:///home/user/svn-repos**

*revision*

Integer revision number to retrieve listing of. When omitted, the HEAD revision is used.

## Return Values

On success, this function returns an array file listing in the format of:

```
[0] => Array
(
    [created_rev] => integer revision number of last edit
    [last_author] => string author name of last edit
    [size] => integer byte file size of file
    [time] => string date of last edit in form 'M d H:i'
              or 'M d Y', depending on how old the file is
    [time_t] => integer unix timestamp of last edit
    [name] => name of file/directory
    [type] => type, can be 'file' or 'dir'
)
[1] => ...
```

## Notes

## Warning

This function is *EXPERIMENTAL*. The behaviour of this function, its name, and surrounding documentation may change without notice in a future release of PHP. This function should be used at your own risk.

## Examples

### Example #14 - [svn\\_ls\(\)](#) example

```
<?php
print_r( svn_ls('http://www.example.com/svnroot/') );
?>
```

The above example will output something similar to:

```
Array
(
    [0] => Array
        (
            [created_rev] => 20
            [last_author] => Joe
            [size] => 0
            [time] => Apr 02 09:28
            [time_t] => 1175520529
            [name] => tags
            [type] => dir
        )
    [1] => Array
        (
            [created_rev] => 23
            [last_author] => Bob
            [size] => 0
            [time] => Apr 02 15:15
            [time_t] => 1175541322
            [name] => trunk
            [type] => dir
        )
)
```

## See Also

- [» SVN documentation on svn list](#)

# svn\_repos\_create

svn\_repos\_create -- Create a new subversion repository at path

## Description

resource **svn\_repos\_create** ( string \$path [, array \$config [, array \$fsconfig ] ] )

<b>Warning</b>
This function is currently not documented; only its argument list is available.

Create a new subversion repository at path

## Notes

<b>Warning</b>
This function is <i>EXPERIMENTAL</i> . The behaviour of this function, its name, and surrounding documentation may change without notice in a future release of PHP. This function should be used at your own risk.

# svn\_repos\_fs\_begin\_txn\_for\_commit

svn\_repos\_fs\_begin\_txn\_for\_commit -- Create a new transaction

## Description

resource **svn\_repos\_fs\_begin\_txn\_for\_commit** ( resource \$repos, int \$rev, string \$author, string \$log\_msg )

<b>Warning</b>
This function is currently not documented; only its argument list is available.

Create a new transaction

## Notes

<b>Warning</b>
This function is <i>EXPERIMENTAL</i> . The behaviour of this function, its name, and surrounding documentation may change without notice in a future release of PHP. This function should be used at your own risk.



# svn\_repos\_fs\_commit\_txn

svn\_repos\_fs\_commit\_txn -- Commits a transaction and returns the new revision

## Description

int **svn\_repos\_fs\_commit\_txn** ( resource \$txn )

Warning
This function is currently not documented; only its argument list is available.

Commits a transaction and returns the new revision

## Notes

Warning
This function is <i>EXPERIMENTAL</i> . The behaviour of this function, its name, and surrounding documentation may change without notice in a future release of PHP. This function should be used at your own risk.

# svn\_repos\_fs

svn\_repos\_fs -- Gets a handle on the filesystem for a repository

## Description

resource **svn\_repos\_fs** ( resource \$repos )

<b>Warning</b>
This function is currently not documented; only its argument list is available.

Gets a handle on the filesystem for a repository

## Notes

<b>Warning</b>
This function is <i>EXPERIMENTAL</i> . The behaviour of this function, its name, and surrounding documentation may change without notice in a future release of PHP. This function should be used at your own risk.

# svn\_repos\_hotcopy

svn\_repos\_hotcopy -- Make a hot-copy of the repos at repospath; copy it to destpath

## Description

bool **svn\_repos\_hotcopy** ( string \$repospath, string \$destpath, bool \$cleanlogs )

<b>Warning</b>
This function is currently not documented; only its argument list is available.

Make a hot-copy of the repos at repospath; copy it to destpath

## Notes

<b>Warning</b>
This function is <i>EXPERIMENTAL</i> . The behaviour of this function, its name, and surrounding documentation may change without notice in a future release of PHP. This function should be used at your own risk.

# svn\_repos\_open

svn\_repos\_open -- Open a shared lock on a repository.

## Description

resource **svn\_repos\_open** ( string `$path` )

<b>Warning</b>
This function is currently not documented; only its argument list is available.

Open a shared lock on a repository.

## Notes

<b>Warning</b>
This function is <i>EXPERIMENTAL</i> . The behaviour of this function, its name, and surrounding documentation may change without notice in a future release of PHP. This function should be used at your own risk.

# svn\_repos\_recover

svn\_repos\_recover -- Run recovery procedures on the repository located at path.

## Description

bool **svn\_repos\_recover** ( string `$path` )

Warning
This function is currently not documented; only its argument list is available.

Run recovery procedures on the repository located at path.

## Notes

Warning
This function is <i>EXPERIMENTAL</i> . The behaviour of this function, its name, and surrounding documentation may change without notice in a future release of PHP. This function should be used at your own risk.

# svn\_status

svn\_status -- Returns the status of working copy files and directories

## Description

array **svn\_status** ( string \$path [, bool \$recursive [, bool \$get\_all [, bool \$update [, bool \$no\_ignore ]]] ] )

Returns the status of working copy files and directories, giving modifications, additions, deletions and other changes to items in the working copy.

## Parameters

*path*

Local path to file or directory to retrieve status of.

Note
Relative paths will be resolved as if the current working directory was the one that contains the PHP binary. To use the calling script's working directory, use <a href="#">realpath()</a> or <code>dirname(__FILE__)</code> .

*recursive*

Whether or not to recursively descend into directories to retrieve status. Default is **TRUE**.

*get\_all*

Whether or not to return all items, regardless of modification status. Default is **FALSE**.

*update*

Whether or not to return information from the server on whether or not the working copy is up-to-date (specifically, entries will be added for items that are out-of-date along with the regular modification checks). Default is **FALSE**.

*no\_ignore*

Whether or not to disregard svn:ignore properties when scanning for new files. Default is **FALSE**.

## Return Values

Returns a numerically indexed array of associative arrays detailing the status of items in the repository:

Array (

```

[0] => Array (
    // information on item
)
[1] => ...
)

```

The information on the item is an associative array that can contain the following keys:

*path*

String path to file/directory of this entry on local filesystem.

*text\_status*

Status of item's text. Refer to [status constants](#) for possible values.

*repos\_text\_status*

Status of item's text in repository. Only accurate if *update* was set to **TRUE**. Refer to [status constants](#) for possible values.

*prop\_status*

Status of item's properties. Refer to [status constants](#) for possible values.

*repos\_prop\_status*

Status of item's property in repository. Only accurate if *update* was set to **TRUE**. Refer to [status constants](#) for possible values.

*locked*

Whether or not the item is locked. (Only set if **TRUE**.)

*copied*

Whether or not the item was copied (scheduled for addition with history). (Only set if **TRUE**.)

*switched*

Whether or not the item was switched using the switch command. (Only set if **TRUE** )

These keys are only set if the item is versioned:

*name*

Base name of item in repository.

*url*

URL of item in repository.

*repos*

Base URL of repository.

*revision*

Integer revision of item in working copy.

*kind*

Type of item, i.e. file or directory. Refer to [type constants](#) for possible values.

*schedule*

Scheduled action for item, i.e. addition or deletion. Constants for these magic numbers are not available, they can be emulated by using:

```
<?php
if (!defined('svn_wc_schedule_normal')) {
    define('svn_wc_schedule_normal', 0); // nothing special
    define('svn_wc_schedule_add', 1); // item will be added
    define('svn_wc_schedule_delete', 2); // item will be deleted
    define('svn_wc_schedule_replace', 3); // item will be added and deleted
}
?>
```

#### *deleted*

Whether or not the item was deleted, but parent revision lags behind. (Only set if **TRUE**.)

#### *absent*

Whether or not the item is absent, that is, Subversion knows that there should be something there but there isn't. (Only set if **TRUE**.)

#### *incomplete*

Whether or not the entries file for a directory is incomplete. (Only set if **TRUE**.)

#### *cmt\_date*

Integer Unix timestamp of last commit date. (Unaffected by *update*.)

#### *cmt\_rev*

Integer revision of last commit. (Unaffected by *update*.)

#### *cmt\_author*

String author of last commit. (Unaffected by *update*.)

#### *prop\_time*

Integer Unix timestamp of last up-to-date time for properties

#### *text\_time*

Integer Unix timestamp of last up-to-date time for text

## Notes

Warning
This function is <i>EXPERIMENTAL</i> . The behaviour of this function, its name, and surrounding documentation may change without notice in a future release of PHP. This function should be used at your own risk.

## Examples



## Example #15 - Basic example

This example demonstrates a basic, theoretical usage of this function.

```
<?php
print_r(svn_status(realpath('wc')));
?>
```

The above example will output something similar to:

```
Array (
  [0] => Array (
    [path] => /home/bob/wc/sandwich.txt
    [text_status] => 8 // item was modified
    [repos_text_status] => 1 // no information available, use update
    [prop_status] => 3 // no changes
    [repos_prop_status] => 1 // no information available, use update
    [name] => sandwich.txt
    [url] => http://www.example.com/svnroot/deli/trunk/sandwich.txt
    [repos] => http://www.example.com/svnroot/
    [revision] => 123
    [kind] => 1 // file
    [schedule] => 0 // no special actions scheduled
    [cmt_date] => 1165543135
    [cmt_rev] => 120
    [cmt_author] => Alice
    [prop_time] => 1180201728
    [text_time] => 1180201729
  )
)
```

## See Also

- [svn\\_update\(\)](#)
- [svn\\_log\(\)](#)
- [» SVN documentation for svn status](#)

# svn\_update

svn\_update -- Update working copy

## Description

int **svn\_update** ( string *\$path* [, int *\$revno* [, bool *\$recurse* ] ] )

Update working copy at *path* to revision *revno*. If *recurse* is true, directories will be recursively updated.

## Parameters

*path*

Path to local working copy.

Note
Relative paths will be resolved as if the current working directory was the one that contains the PHP binary. To use the calling script's working directory, use <a href="#">realpath()</a> or <a href="#">dirname(__FILE__)</a> .

*revno*

Revision number to update to, default is **SVN\_REVISION\_HEAD**.

*recurse*

Whether or not to recursively update directories, default is **TRUE**.

## Return Values

Returns new revision number on success, returns **FALSE** on failure.

## Notes

Warning
This function is <i>EXPERIMENTAL</i> . The behaviour of this function, its name, and surrounding documentation may change without notice in a future release of PHP. This function should be used at your own risk.

## Examples

### Example #16 - Basic example

This example demonstrates basic usage of this function:

```
<?php
echo svn_update(realpath('working-copy')) ;
?>
```

The above example will output something similar to:

234

### See Also

- [svn\\_checkout\(\)](#)
- [svn\\_commit\(\)](#)
- [» SVN documentation for svn update](#)