

 **This is a working draft. It is here for guidelines. Upon posting final guidelines, please remove this caption.** 

---

# SVN Guide Lines (Working Draft)

## 0. Definitions

**feature:** A feature is a new introduced distinguishing characteristic of a software item. It will be called feature as long as it is not part of the code base.

**trunk:** trunk is the main development branch for the project. It underlies several rules, stated in [section 1](#).

**branch:** A branch is a development part that is used for the creation of modified code (e.g. features). Rules for branches are [section 2](#).

**tag:** A tag is a final branch where a release version is stored. Everything related to tags is stored in [section 3](#).

## 1. The main development branch (trunk)

### 1.1 Commit

**a.** The CHANGELOG file should be updated on every commit. Changelog entries should look like this:

```
YYYY-MM-DD FIRSTNAME SURNAME  
* [PART] Fixed #NUMBER: TEXT
```

Where PART is one of the following: \* CONFIGS \* DATABASE \* DISTS \* DOCS \* ENGINE \* GUI \* SETUP

Multiple entries should be ordered alphabetically. A line should not contain more than ~120 chars.

**b.** The date of commit must be the date in UTC.

**c.** Every commit should belong to a single change: Correction of a bug, addition of a new feature (*branch reintegration*) or any other task. This facilitates the work of the quality assurance tasks. Change reverts can be performed more easy if necessary.

Commits can be done with the following command:

```
$ svn WORKING_COPY
```

```
$ svn up
$ svn ci -m " * [PART] Fixed #NUMBER: TEXT"
$ svn up
```

but better is to use an IDE such as PhpStorm, NetBean, eclipse...

Note: If someone else is the author of a patch the name of the author should be added in round brackets.

## 1.2 Merge

**a.** Developers should never perform partial merge of a branch into the trunk, only complete reintegrations.

Reintegration can be done with the following command:

```
$ svn WORKING_COPY
$ svn up
$ svn merge --reintegrate
http://i-mscp.svn.sourceforge.net/svnroot/i-mscp/trunk
$ svn up
```

# 2. Development branches

## 2.1 General

**a.** The branches should be used to develop of new features or to maintain a version already released.

**b.** The feature branch developer should keep their branch up to date. He/she has to perform regular synchronization with the trunk and need to resolve conflicts properly.

## 2.2 Creation

**a.** A release branch must be named after a release and will be created with the following command, where x.y is the number of the release.:

```
$ svn copy http://i-mscp.svn.sourceforge.net/svnroot/i-mscp/trunk
http://i-mscp.svn.sourceforge.net/svnroot/i-mscp/branches/mscp-x.y
```

**b.** Each feature should have its own branch and must be initialized from the trunk. Use the following command for this:

```
$ svn copy http://i-mscp.svn.sourceforge.net/svnroot/i-mscp/trunk
```

```
http://i-mscp.svn.sourceforge.net/svnroot/i-mscp/branches/FEATURE-NAME-dev
```

## 2.2 Commit

The same rules as for [1.1](#) are valid.

## 2.3 Synchronizations

As specified in paragraph [2.1.c](#) developers must perform regular synchronization of their branch with the trunk. The Synchronization should be made with the following commands:

```
$ svn WORKING_COPY
$ svn merge http://isp-control.net/ispcp_svn/trunk
$ svn ci -m "Synchronization with main development branch"
$ svn up
```

## 2.4 Merge from branch to trunk

Partial merges from one branch into the trunk are not permitted.

## 2.5 Re-integration

**a.** After the development of a feature is completed, the branch developers need to ask the team for approval to re-integrate the branch into the trunk with the following commands:

```
$ svn WORKING_COPY
$ svn merge http://i-mscp.svn.sourceforge.net/svnroot/i-mscp/trunk
$ svn ci -m "Synchronization with main development branch"
$ svn up
$ svn merge --reintegrate
http://i-mscp.svn.sourceforge.net/svnroot/i-mscp/trunk
```

**b.** After successful re-integration, the branch developer has to delete the branch. This can be done by using the following command:

```
$ svn rm
http://i-mscp.svn.sourceforge.net/svnroot/i-mscp/branches/YOUR_BRANCH
```

## 3. Tags

### 3.1 General

- a. Tags are the storage of releases. Once set, tags are immutable.
- b. Tags should be only created by the **\_person preparing the release\_**
- c. Every tag has to follow the name convention **i-mscp-x.y.z**, where x.y.z is the number of the release.

### 3.2 Tagging

The tagging has to be done on the day of a new version with the following command:

```
$ svn move  
http://i-mscp.svn.sourceforge.net/svnroot/i-mscp/branches/mscp-x.y.z  
http://i-mscp.svn.sourceforge.net/svnroot/i-mscp/tags/mscp-x.y.z
```

From:

<https://wiki.i-mscp.net/> - **i-MSCP Documentation**

Permanent link:

<https://wiki.i-mscp.net/doku.php?id=svn&rev=1290290682>

Last update: **2011/07/04 06:40**

