

Sakai 2.x branch management cheat sheet

These instructions presumes a familiarity with Subversion conventions, commands and status codes.

Prerequisites

Apache Subversion (SVN) 1.6.x.

[Packaged binaries](#) [Source code](#) [Release notes](#)

Sakai client-side configuration file

Install the Sakai SVN config file on your local machine. This will help to ensure commit consistency amongst Sakai committers.

[config file](#)

Copy the configuration file to `~/.subversion/config` (Unix) or `%APPDATA%\Subversion\config` (Windows).

Jira account

Sakai uses Jira as its issue tracking system. Sign up for a Jira account at <http://jira.sakaiproject.org/secure/Signup!default.jspa>. After you create an account you will need to be added (at a minimum) to the Jira "sakai-dev" group in order to work with Jira tickets. Contact jira-admins@collab.sakaiproject.org for more information.

Commit access

Before you can commit code to the Sakai SVN server, you must obtain commit access. You can request commit access by sending an email to svn@collab.sakaiproject.org. Obtaining commit access is not automatic and requires for core committers, at a minimum, agreeing to the terms of the Sakai Contributor Licensing Agreement (CLA). Contact Mary Miles at mmiles@sakaifoundation.org to obtain a copy of the CLA.

Workflow

🚫 Fixes should be merged to stable maintenance branches only after they have been tested and verified locally by QA. Currently, fixes that are ready for merging are those that have been 1) closed and have a branch status set "to merge".

1. Review Jira tickets that have been 1) closed and 2) have a branch status set "to merge". The ticket assignee is responsible for determining whether or not the fix should be applied to other Sakai versions (e.g., 2.5, 2.6). However, the branch manager should also consider the range of a fix and update the ticket's affects and fix versions appropriately or seek clarification from the assignee.
2. Perform merge or apply patch.
 - If merging, it is recommended that a dry run be performed first.
 - If conflicts are encountered during the merge attempt contact ticket assignee and request that they resolve the conflict.
3. If dry run is successful, perform merge.
4. Run a mvn clean install before committing fix to ensure that the merge did not break the build.
5. Commit fix. Avoid overly descriptive merge messages. `-m "TICKET-number merge trunk r12345"` is sufficient. Feel free to cross-reference other Jira tickets in the log message if appropriate.
6. Update Jira ticket.
 - Set `fix version` to upcoming release(s).
 - Set maintenance branch status to "closed".
 - Add short comment noting the branch merge and the merge revision number(s) (e.g., 2.7.x, r56789)
 - If necessary, close issue.
7. Repeat steps above if fix needs to be merged to other maintenance branches.

Basic actions

svn merge

Merge changes (comparing revisions N to M) in a working copy (an explicit commit is required thereafter to apply the merge). If the working copy path is omitted a default value of `""` is assumed. The `--change (-c)` switch is equivalent to `-r <M-1>:M`.

Format

```
svn merge sourceURL1[@N] sourceURL2[@M] [WCPATH]
svn merge sourceWCPATH1@N sourceWCPATH2@M [WCPATH]
svn merge [[-c M]... | [-r N:M]...] [SOURCE[@REV]] [WCPATH]]
```

Example `--change (-c)`

```

sakaiger$ cd sakai-2.7.x/syllabus
sakaiger$ svn update
At revision 73650.
sakaiger$ svn merge --dry-run -c 72260 https://source.sakaiproject.org/svn/syllabus/trunk/
--- Merging r72260 into '.':
U    syllabus-app/src/webapp/syllabus/edit.jsp
sakaiger$ svn merge -c 72260 https://source.sakaiproject.org/svn/syllabus/trunk/
--- Merging r72260 into '.':
U    syllabus-app/src/webapp/syllabus/edit.jsp
sakaiger$ svn commit -m "SAK-16537 merge trunk r72260"
Sending          .
Sending          syllabus-app/src/webapp/syllabus/edit.jsp
Transmitting file data .
Committed revision 73651.

```

Example (--revision (-r))

```

sakaiger$ cd sakai-2.7.x/gradebook
sakaiger$ svn update
At revision 73695.
sakaiger$ svn merge --dry-run -r73487:73488 https://source.sakaiproject.org/svn/gradebook/trunk/
--- Merging r73488 into '.':
U    app/ui/src/java/org/sakaiproject/tool/gradebook/ui/EnrollmentTableBean.java
sakaiger$ svn merge -r73487:73488 https://source.sakaiproject.org/svn/gradebook/trunk/
--- Merging r73488 into '.':
U    app/ui/src/java/org/sakaiproject/tool/gradebook/ui/EnrollmentTableBean.java
sakaiger$ svn status
M    .
M    app/ui/src/java/org/sakaiproject/tool/gradebook/ui/EnrollmentTableBean.java
sakaiger$ svn commit -m "SAK-16604 merge trunk r73488"
Sending          .
Sending          app/ui/src/java/org/sakaiproject/tool/gradebook/ui/EnrollmentTableBean.java
Transmitting file data .
Committed revision 73696.

```

Example (reverse merge)

You can use `svn merge` to undue a previous commit by specifying a reverse difference between changesets (e.g. `-r70787:70784`). After rolling back your working copy be sure to commit your changes.

```

sakaiger$ svn update
At revision 70787.
sakaiger$ svn merge --dry-run -r70787:70784 https://source.sakaiproject.org/svn/content/branches/sakai_2-5-x/
--- Reverse-merging r70787 through r70785 into '.':
U    content-api/api/src/java/org/sakaiproject/content/cover/ContentHostingService.java
U    content-impl/impl/src/java/org/sakaiproject/content/impl/CollectionAccessFormatter.java
U    content-impl/impl/src/java/org/sakaiproject/content/impl/BaseContentService.java
sakaiger$ svn merge -r70787:70784 https://source.sakaiproject.org/svn/content/branches/sakai_2-5-x/
--- Reverse-merging r70787 through r70785 into '.':
U    content-api/api/src/java/org/sakaiproject/content/cover/ContentHostingService.java
U    content-impl/impl/src/java/org/sakaiproject/content/impl/CollectionAccessFormatter.java
U    content-impl/impl/src/java/org/sakaiproject/content/impl/BaseContentService.java
sakaiger$ svn commit -m "SAK-14351, SAK-13693, reverse merge r70785-87".
Sending          content-api/api/src/java/org/sakaiproject/content/cover/ContentHostingService.java
Sending          content-impl/impl/src/java/org/sakaiproject/content/impl/BaseContentService.java
Sending          content-impl/impl/src/java/org/sakaiproject/content/impl/CollectionAccessFormatter.java
Transmitting file data ...
Committed revision 70788.

```

svn diff

Displays the difference between two revisions or paths. Useful for checking changes before final commit.

Format

```
diff [-c M | -r N[:M]] [TARGET[@REV]...]
diff [-r N[:M]] --old=OLD-TGT[@OLDREV] [--new=NEW-TGT[@NEWREV]] [PATH...]
```

Example

```
sakaiger$ svn diff

Property changes on: .
-----
Modified: svn:mergeinfo
   Merged /gradebook/trunk:r73488

Index: app/ui/src/java/org/sakaiproject/tool/gradebook/ui/EnrollmentTableBean.java
=====
--- app/ui/src/java/org/sakaiproject/tool/gradebook/ui/EnrollmentTableBean.java      (revision 73477)
+++ app/ui/src/java/org/sakaiproject/tool/gradebook/ui/EnrollmentTableBean.java      (working copy)
@@ -23,6 +23,7 @@
 package org.sakaiproject.tool.gradebook.ui;

 import java.io.Serializable;
+import java.text.Collator;
 import java.util.ArrayList;
 import java.util.Collections;
 import java.util.Comparator;
@@ -62,7 +63,7 @@
     */
     static final Comparator<EnrollmentRecord> ENROLLMENT_NAME_COMPARATOR = new Comparator<EnrollmentRecord>() {
-        public int compare(EnrollmentRecord o1, EnrollmentRecord o2) {
+        return o1.getUser().getSortName().compareToIgnoreCase(o2.getUser().getSortName());
+        return Collator.getInstance().compare(o1.getUser().getSortName(), o2.getUser().
getSortName());
     }
 };
```

svn log

Useful for reviewing working copy commit history. `svn log` will return all log messages in the current working copy if no arguments are supplied.

Format

```
svn log [PATH]
svn log URL[@REV] [PATH...]
```

Example

```
sakaiger$ svn log -r73170 --verbose
-----
r73170 | arwhyte@umich.edu | 2010-02-05 09:33:21 -0500 (Fri, 05 Feb 2010) | 1 line
Changed paths:
   M /gradebook/branches/sakai-2.7.x/app/ui/src/java/org/sakaiproject/tool/gradebook/ui/AssignmentDetailsBean.java
   M /gradebook/branches/sakai-2.7.x/app/ui/src/java/org/sakaiproject/tool/gradebook/ui/InstructorViewBean.java

SAK-17937 apply Zach patch.
-----
```

Example

```
sakaiger$ svn log -r{2010-02-01}:{2010-03-01} --xml
<?xml version="1.0"?>
<log>
<logentry
  revision="73170">
<author>arwhyte@umich.edu</author>
<date>2010-02-05T14:33:21.448513Z</date>
<msg>SAK-17937 apply Zach patch.</msg>
</logentry>
<logentry
  revision="73696">
<author>arwhyte@umich.edu</author>
<date>2010-02-19T14:21:39.086598Z</date>
<msg>SAK-16604 merge trunk r73488</msg>
</logentry>
</log>
```

svn propset (editing a commit message)

If you need to change a commit message, you can issue an `svn propset` with the `--revprop` switch enabled to update the log message. If you issue the log revision from a working copy you can omit the path.

Format

```
svn propset -r N --revprop svn:log "new log message" URL
```

Example

```
sakaiger$ svn propset --revprop -r 71566 svn:log "SAK-17728 merge trunk r71535"
property 'svn:log' set on repository revision 71566
```

svn status

If you need to check all file and tree changes you have made to your working copy before committing changes use `svn status`. You can also include a file path or specify the `--verbose -v` option to review the current status of every item in your working copy.

Format

```
svn status [PATH...]
```

Example

```
sakaiger$ cd sakai-2.7.x/reference
sakaiger$ svn status
M      demo/start-sakai.bat
M      demo/start-sakai.sh
```

Codes (partial listing)

'A' Item is scheduled for addition.

'C' The contents (as opposed to the properties) of the item conflict with updates received from the repository.

'D' Item is scheduled for deletion.

'L' Item is locked.

'M' Item has been modified.

'?' Item is not under version control.

'!' Item is missing (e.g., you moved or deleted it without using svn). This also indicates that a directory is incomplete (a checkout or update was interrupted).

Patch Management

ADD TEXT

Applying patches

```
$ patch -p0 < patchfile
```

ADD EXAMPLES

More info

Apache Subversion home: <http://subversion.apache.org/>

SVN documentation: *Version Control with Subversion* (includes links to SVN 1.5 and 1.6 versions (nightly) of the online book): <http://svnbook.red-bean.com/>