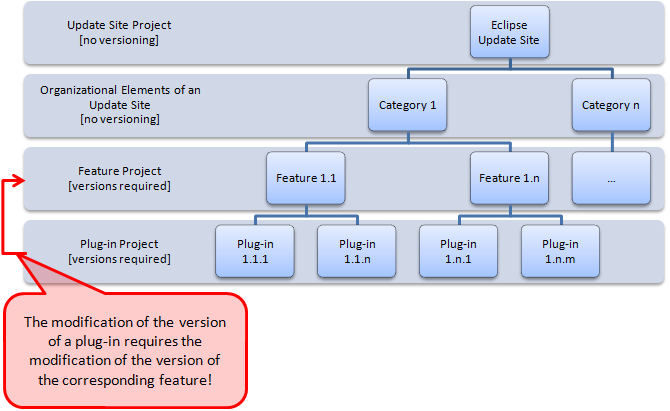
# EASy-Producer Release Preparation

## Overview Eclipse Update Site

The figure below illustrates the general structure of an Eclipse update site:



Typically, “modification” of a version means increasing the current version number (plug-in and feature). As decreasing of versions is in general possible, this does not have any effect on the user installation of EASy (The Eclipse update mechanism does not support any kind of roll-back to an earlier version; it only detects new versions checking the version numbers of the installed features and the available features of an update site).

## Overview EASy-Producer Update Sites

EASy-Producer is managed in a set of different update sites. Besides a core update site[[1]](#footnote-1) which provides the basic capabilities of EASy-Producer, the following project-specific update sites exist:

EASy-Producer Update Site 🡺 core update site *[including INDENICA-features]*1

EASy-Producer Update Site (Bosch) 🡺 update site for Bosch GmbH (Project “EASyCar”)

EASy-Producer Update Site (KlugIS) 🡺 update site for KlugIS (Project “ScaleLog”)

Thus, a new release depends on the features that have to be updated. In other words, if new versions of ScaleLog-specific features (plug-ins) have to be released, focus on the EASy-Producer Update Site (KlugIS), while EASy core features like reasoning affect the EASy-Producer Update Site (core update site).

## Release Process

The release process is the same of each of the update sites introduced above.

### When release is prepared for the first time:

* Create new workspace for the projects (Update site, features, plug-ins) included in the release. Keep this workspace separate from your programming workspace as it probably will contain more projects than you need and you can maintain the release version as well as a possibly different development version.
* Add projects from the repository (plug-ins are in trunk folder, features and update sites are root folders).

### When release is prepared:

* Inform all contributors by mail that no git submission shall be done until the release notification is sent.
* For every project that does not end with “feature“[[2]](#footnote-2):
  + Right-click on the project and open the history view of the “Team” context menu: Team 🡪 Show History.
  + If there are any critical changes (indicated by history comments) after the last release (committed files with comment “Release x.x.x“): Update the project (plug-in) and increase the version of the plug-in in the manifest file, e.g. to “y.y.y+1“.
  + **Important**: for each plug-in whose version is changed, the version of the parent feature has to be changed accordingly (see next bullet)
* Check features (projects that end with “feature”):
  + Open the Feature.xml file in the feature-project and open the Plug-ins tab
  + For each plug-in change the version number (in Plug-in Details in the upper right corner of the menu) to **the same version** previously defined for the plug-in (previous bullet)
  + If the plug-in versions were updated, than the feature version must be changed to „z.z.z+1“.
  + Name of the feature and the new version number are copied to the change list.
* Update the EASy help bundle
  + Update the most recent versions of the IVML and the VIL language spec with the respective bundle version numbers on the front page. (and commit the changed versions)
  + Generate PDFs from the IVML language spec, the VIL language spec as well as the user’s guide and the developers guide. Thereby, add the current date to title and version history, temporarily remove “Preview” from the title to create the release PDFs.
  + Replace the old PDFs in the folder html\pdf of the EASy help bundle with the newly generated ones. **Important:** Ensure that the names of the new files match the names of the old files (explicitly replace them). Otherwise, the link in the HTML-pages will brake and the help is not usable anymore.
  + Copy the PDFS also into doc/web/docRelease and commit the result for automated publishing to the EASy update sites.
* Update site:
  + Open the site.xml file in the update site project and open the site.xml tab
  + Search for features that are registered in the change list and edit their version number accordingly to the versions defined in the above steps.
  + **Hint**: you can check whether all features of an update site are updated correctly by switching to the site map tab of the site.xml file and searching for features (in the categories) if they are grayed-out. This indicates that the feature in version defined in the site.xml cannot be found and thus has to be correct.
  + Finally, build the update site by pressing the “Build All” button in the site map tab of the site.xml file.
* Build and commit:
  + After ”Build All“ 2 folders (plugins and features), 2 jar files (artifacts and content),and the site.xml file should be generated; these files have to be uploaded to the server.
  + **Error:** In case that an additional **log.zip** was generated, the build was not successful. In this case, unzip the log.zip and search in the various logs (there is one log-file for each bundle) for error-messages. Correct the specific error(s), delete the log.zip from the update site bundle and rebuild the entire update site using the “Build all” button. If the log.zip will be generated again, follow the instructions of this bullet ;-)
  + Further all changes (files, versions, etc.) should be committed with the comment „Release x.x.x+1“. **Build.xml files must not be committed!**
* Publish the release.
  + **Requirements:**
    - Access to the projects-server via WinSCP or similar tools.
    - Add the following information to access the server:
      * Server-address: projects.sse.uni-hildesheim.de
      * User: webmaster
      * PW: <ask Holger>
  + Navigate to one of following server-directories:
    - *WWW-Folder/eclipse/update-sites/easy/easy* – for an EASy-core release (this means an update of the EASy-core bundles which are typically not related to any specific project or extension)
    - *WWW-Folder/eclipse/update-sites/easy/easy\_extensions/  
      {ExtensionOrProjectName}* --- for a release of project- or partner-specific extensions
  + Copy the following files and folders of the newly created release to one of the folders described above:
    - Folder: features (including all nested files)
    - Folder: plugins (including all nested files)
    - File: artifact.jar
    - File: content.jar
    - File site.xml
* Testing the release:
  + Typically, you should have an Eclipse-instance with an installed EASy-Producer release (after the new release, this should be outdated)
  + Start this Eclipse-instance and go to *Help 🡪 Check for Updates*. If the release was successfully build with increased version numbers for the respective bundles and parent-features (and published correctly), the features that include the updated bundles will be displayed as “ready for update”.
  + Perform this update and **test your new or update functionalities.** In particular, test also the UI functionality including the content assists of IVML, VIL and VTL.
* Create a Maven release
  + Change all POM files from x.x.x-SNAPSHOT to x.x.x, i.e., keep the same version number. The x.x.x-SNAPSHOT versions were used to create the x.x.x release. Also change the POM files in EASy-Standalone/poms as they are used for deploying the bundled versions. Since EASy version 1.3.0-SNAPSHOT, versions are just in POM headers.
  + Commit the changes and let Jenkins do a full build.
  + Check deploying/deployCentral.bat whether all headless components are included. If needed, perform a dry run, i.e., comment out all Maven deployment commands and check the tmp folder whether all Jars have been downloaded correctly. Finally, check the prerequisites for a Maven Central deployment (pgp installed, default key set, etc. as stated in deploy.Central.bat) and run deployCentral.bat with all Maven deployment commands.
  + Go for <https://oss.sonatype.org> as described in deployCentral.bat, close/check the deployment and, if checks are successful, deploy the release to Maven central.
  + Change all POM files to y.y.y-SNAPSHOT, with y.y.y being the next version number of EASy. Also change the POM files in EASy-Standalone/poms as they are used for deploying the bundled versions. Since EASy version 1.3.0-SNAPSHOT, versions are just in POM headers.
  + Commit the changes and let Jenkins do a full build for the new snapshot versions.
* Update EASyCommandLine.zip from Jenkins job Standalone in /var/www/easy on projects.
* For QM:
  + With new version numbers commited, wait until Jenkins built the *Standalone* job
  + Change the version numbers in all pom files of *QualiMaster.EASyRuntime* to the (new) EASy release version number (without –SNAPSHOT). Please consider also the Eclipse dependencies. Commit the changes (causes a Maven release).
  + Wait until Jenkins build job of *QualiMaster.EASyRuntime* is done.
  + Change all EASy-/Eclipse-related version numbers to the next expected EASy release version number and add the –SNAPSHOT prefix.
  + Commit again (causes start of new snapshot series).
  + Inform SSE-QM team about new version so that the infrastructure can be updated accordingly.
* **Inform about new release**:
  + If testing the new release was successful (no [new] errors found), the new release should be announced, typically via mail. Mention explicitly the **new features** or the **specific reason for the release**.
    - Internal:
      * Subject: [Release: EASy] Version {ReleaseVersionNumber}
      * Content: Summary of major changes and/or extensions in simple bullet-form, no long text!
    - External (e.g. for specific partners or projects): this depends on the conventions for using, e.g., the project-specific mailing-list.

1. Due to historical reasons the core update site of EASy-Producer also provides the features for the INDENICA project. We will restructure this update site in the future to decouple the INDENICA-specific features from the core features of EASy-Producer. [↑](#footnote-ref-1)
2. We decided to include the term „feature“ in the name of a feature project (at the end of the name) for easily differentiation of plug-in-projects and feature-project in Eclipse. [↑](#footnote-ref-2)