RSA/P Best Practice, StarTeam

<u>Application</u>

StarTeam is not a tool developed with only Ericsson's business in mind, it is a global system used all over the world. StarTeam can be used in many, many ways. This document will guide the developer in the usage of StarTeam with regards to Ericsson standards.

This document is divided into two parts. The first part of this document is a step-by-step guide with examples and explanations. The guide includes creation of projects, setting up of views and revision control with the help of labels.

The second part is a reference document containing the RSA/P specific structure. Developers already familiar with StarTeam should use this reference part as a template for creating and maintaining projects in StarTeam.

<u>Document coverage</u>

This document does not, nor is it intended, to cover all aspects of StarTeam usage. In order to get familiar to general StarTeam functionality you should attain a StarTeam course. There is also other reading materials available at RSA/P including:

Printable manuals (PDF format, located using the SW-platform interface)

- StarTeam Getting Started

- StarTeam User Guide

- Best Practice Documents from StarBase

Books (each NPI-section has two copies)

- Practical Software Configuration Management

ERICSSON ⩘

1 PART I: USING STARTEAM ACCORDING TO RSA/P NORMS (STEP-BY-STEP INSTRUCTION).

1.1 INTRODUCTION

StarTeam uses a standard check-in/check-out system for revision handling. Ericsson companies using StarTeam all have a central server commonly referred to as a repository. Technically, this repository is a database where all files (and their different revisions) are stored.

StarTeam provides an easy to use interface between this database and the developers local hard-drive. A default working directory is specified for each database by the StarTeam administrator. At RSA/P this directory is C:\PROJECTS.

Simply speaking, StarTeam takes care of the synchronisation between your local C:\PROJECTS folder and the global database. If you change a file in your local working folder, and would like other people to access it, you must CHECK-IN the file to the database. If you find a file you want in the database, you should use the CHECK-OUT function. This function retrieves the file from the database and stores it in your local working folder.

1.2 GETTING STARTED

Make sure you have StarTeam correctly installed and configured. Refer to the StarTeam installation instructions in this guide for more information. If you have installed StarTeam correctly the screen should look like this:
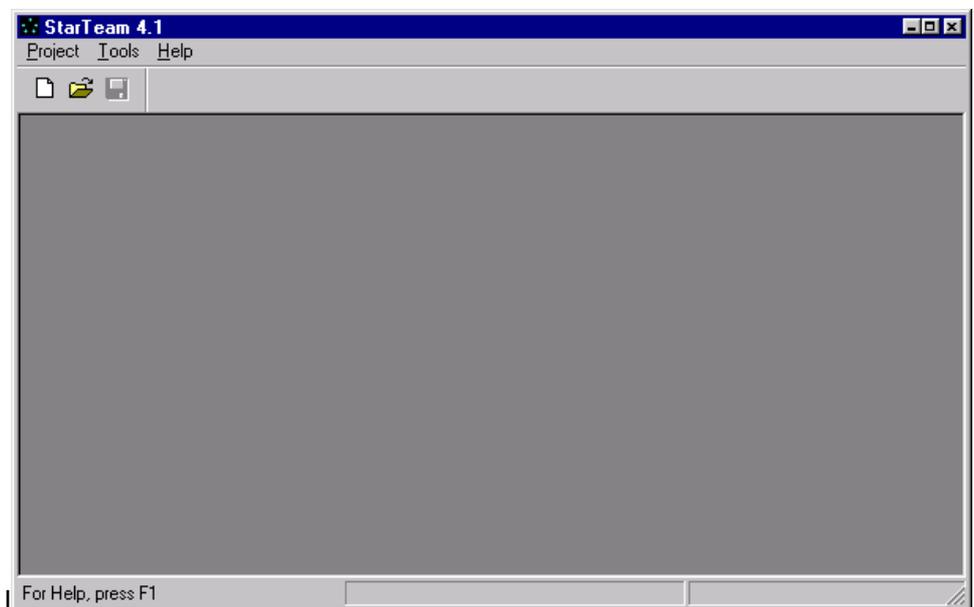
1.3 CREATING A STARTEAM PROJECT

Projects are the most essential part of StarTeam. Most of the RSA/P specific norms are applied at the project level.

1.3.1   Choosing the right StarTeam-project implementation

RSA/P uses StarTeam in special way that is not fully supported in the current StarTeam version. StarTeam is designed to manage a few very large projects, but here at RSA/P we have many small/medium sized projects and even project-independent programs. In order to avoid "overflow" in the StarTeam-project selector, one StarTeam-project is created for each department within the RSA/P. Subprojects are then created within each department in the form of floating views.

For large projects a separate StarTeam-project can be created. Contact your local StarTeam administrator for more information on this subject.

1.3.2   Create a new small/medium size project

Select Open… from the Project menu. If you have configured the server settings correctly a server name (StarTeam Gavle) should appear in the Open dialog. Select you department and click Next.
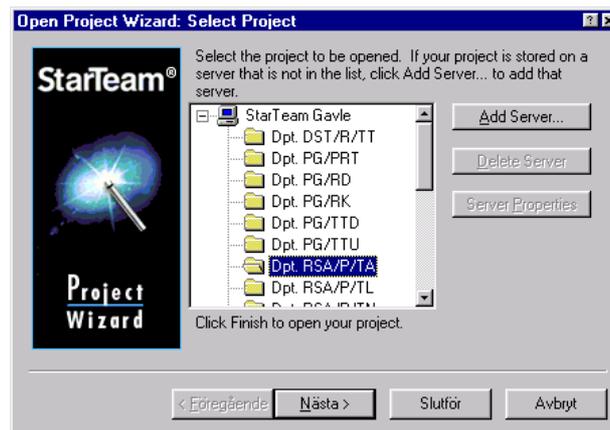
In the next dialog box you are presented with a number of so called Views. Each view is a subset of the complete database. If you would like to create a new project, select the RSA/P/TA view (the root) and click Finish.
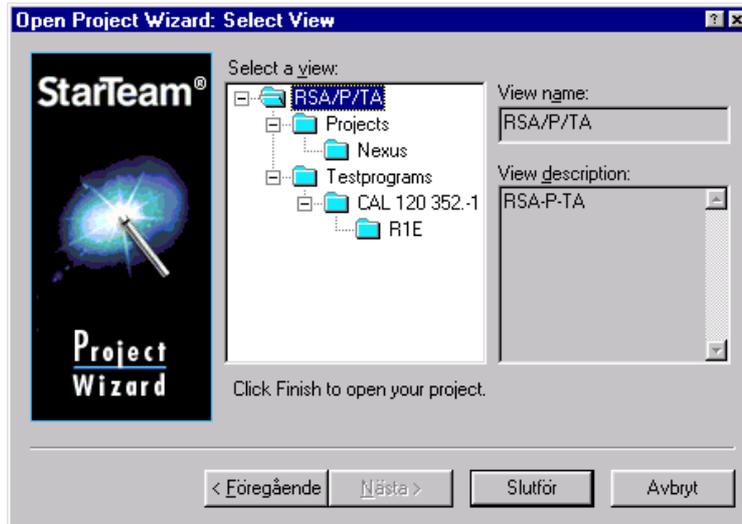
NOTE: The Projects and TestPrograms views/folders are part of the StarTeam RSA/P standardisation and should always exist within each department.

The next step is to create a new folder (and a new view) for the project. Make sure the Projects folder is marked. Then select the New option from the Folder menu. In the first dialog box, click Next, then enter the name of the Project and a description:

Click Next> and then Finish.

1.3.3   Creating your project View

This step is very important with regards to the RSA/P StarTeam norms. Select the newly created folder and access the popup-menu with a right mousebutton click. From the popup-menu select New View.

Now, make sure the Permit items to branch item is selected and that the default item branching is set to Branch none. All project views should have these properties, since they are floating configurations (in order words, if you later change something in the subviews of the project, this view will be updated as well). Click Next.

In the next dialog, enter a name (enter the project name) and a description (enter the project description). Click Next until the following dialog is displayed:
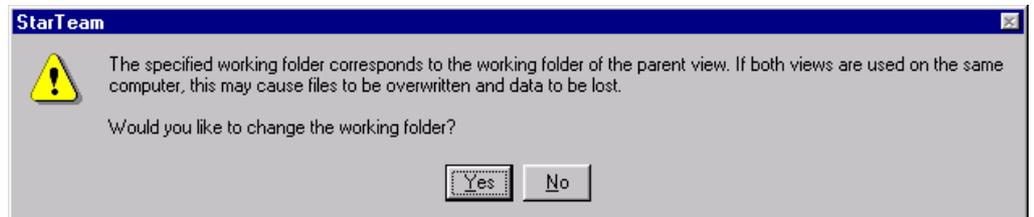
Select No at this point since we will not do any individual branching within this view. In the Next dialog the most important view property is configured. The Specified configuration should be Floating Configuration:

Click Finish. Your project view is now created. When you reopen this project, ALWAYS select the project view. All changes and labels associated with this project should be attached to the project view.

NOTE: You can switch views at any time by using the Select View option found in the View menu.

1.3.4 Create standard folders

Within each project tree standard folders should always be created: Doc, Install and Source. Create these by using the New Folder option. Within Install there should also be two folders called Files and InstallShield.

In order to simplify readability and access all project should have the same base folder structure. It is recommended to use the standard folder, however extensions and modifications must be allowed to ensure flexibility.

### 1.4 CREATE A NEW PROJECT-INDEPENDENT TEST-PROGRAM

Follow the instructions for creating a new small/medium sized project but use the Testprograms folder/view instead of the Projects folder/view were applicable.

All test-programs in the Testprograms folder should have a separate view (just as a project has a separate view). When naming test-program views, you cannot use the /-character, instead use the .- character combination. Example: CAL 120 352/1 becomes CAL 120 352.-1

You should always name the folder and view according to the standard Ericsson program enumeration (CAL). Standard folders within a CAL folder are the same as for project folders.

### 1.5 REVISION CONTROL STEP-BY-STEP

One of StarTeam's most important features is revision control. StarTeam supports a variety of labelling methods for revision control: view-, build- and revision-labels.
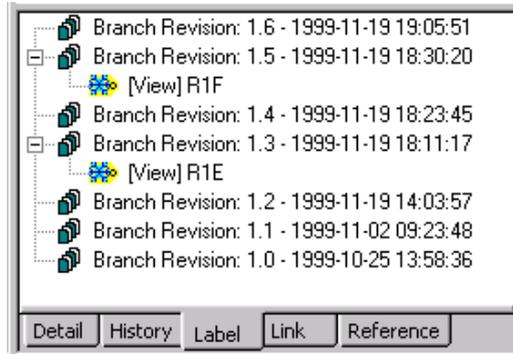
Every time a file is checked in, StarTeam updates the internal revision information for that file. The Labels tab provides the connection between internal revisions and labels. Internal revisions is displayed in the format 1.1, 1.2 etc, while labels should be used to specify the Ericsson revision (R1B etc).

1.5.1 Using View Labels

The correct way to use StarTeam in the RSA/P software environment is to use View Labels to manage revisions. Build and Revision labels are not used since they disable StarTeam's branching capabilities.
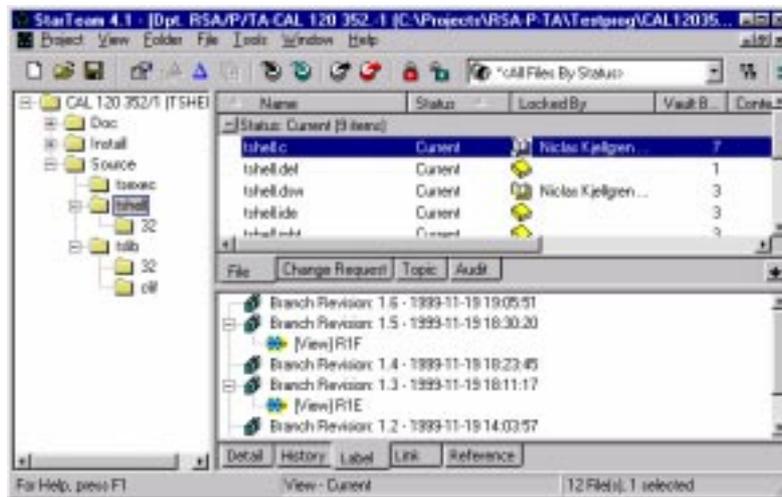
Since all types of labels only appear in the view they were created, it is imperative that all work on a project or test-program is done in the appropriate view (for example the CAL 120 352.-1 or Nexus view).

1.5.2 Attaching a new View label

View labels are attached by selecting the Labels option from the View menu. The most common practice is to attach a new view label when a major milestone is passed. For example the test-program is promoted from R1F to R1G.
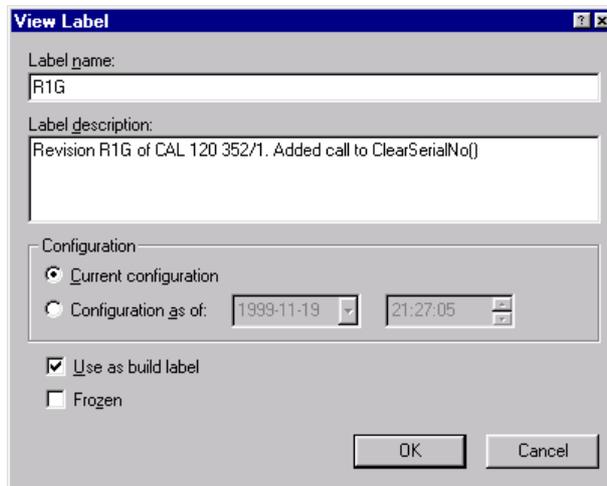
Make sure the Current configuration option is marked if you wish to attach the labels to the newest (current) version of the files.

View labels can be moved as long as they are not frozen. Moving a label is done by usage of the drag-and-drop method in the labels tab (lower right pane in StarTeam).

1.5.3 Branching a current or old revision

One of StarTeam's most useful features is its ability to branch a view based on view labels. Take this example: the current revision of CAL 120 352/1 is R1F and the development department wished to add special functions to revision R1E of the same program. The changes should be made WITHOUT affecting the R1F release.

This is accomplished by creating a new view based on the view labels designated R1E. Make sure the root folder is selected (the CAL or project name). Select New.. from the View menu. Since we don't want the changes to R1E to affect the current (R1F) revision, we select the Permit items to branch option. We also have to set the Default Item Branching to Branch All:

Click Next and give the new view the same name as the revision labels (in this example R1E). Click Next. Make sure the root folder is selected. Click Next.

This next step is important to avoid merge problems. A new local working directory must be entered. In compliance with RSA/P norms the new folder should have "_<revision>" added. In this example "_R1E". When fully printed "C:\Projects\RSA-P-TA\Testprog\CAL120352.-1\" becomes "C:\Projects\RSA-P-TA\Testprog\CAL120352.-1_R1E". Click Next.

In the next dialog you have to specify the view label that the new view should be based on. In this example we select R1E:
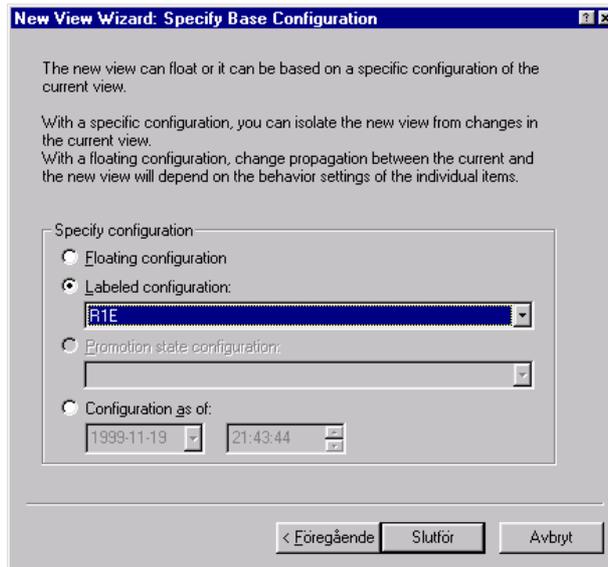
Click Next.

Now you are working in a separate "R1E" view. Changes made to files here will not affect the previous view (R1B, R1F etc). You have now branched your project.

2 PART II: QUICK STRUCTURE REFERENCE

Since StarTeam is a tool used by many people at the same time, a common structure must be established. This standard structure will ensure that all StarTeam projects will look and behave in the same way. This is very important since StarTeam can be used in a number of different ways.
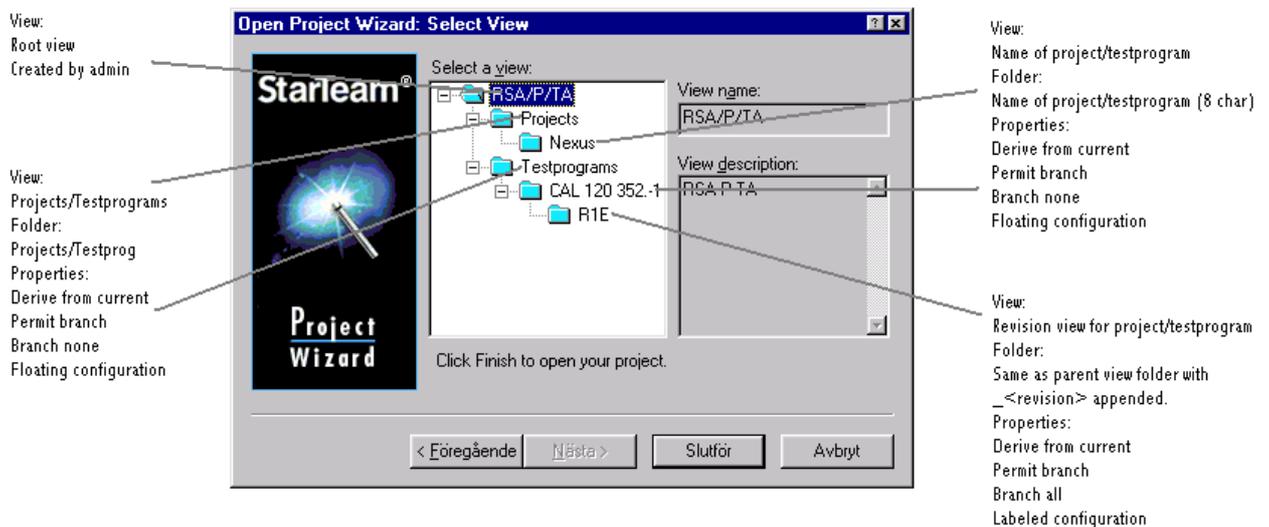
2.1 PROJECTS

For small/moderate projects, the local department name should be used as StarTeam-project title. Larger projects must acquire their own StarTeam-project title. Contact your local StarTeam administrator for more information on creating a separate StarTeam project.

2.2 FILE TYPES

Documents, drawing and source code relating to a project should be checked into StarTeam.

2.3 STANDARD VIEWS:

The Project and Test-program views should always be created. Configuration accordingly:

View:
Root view
Created by admin

View:
Projects/Testprograms
Folder:
Projects/Testprog
Properties:
Derive from current
Permit branch
Branch none
Floating configuration

View:
Name of project/testprogram
Folder:
Name of project/testprogram (8 char)
Properties:
Derive from current
Permit branch
Branch none
Floating configuration

View:
Revision view for project/testprogram
Folder:
Same as parent view folder with
_<revision> appended.
Properties:
Derive from current
Permit branch
Branch all
Labeled configuration

2.4 STANDARD FOLDERS

In order to simplify readability and access all project should have the same base folder structure. It is recommended to use the standard folder, however extensions and modifications must be allowed to ensure flexibility. Recommended standard folders:

Doc folder.
Should contain
documents relating to
the project. Can contain
suitable subfolder.

Install folder.
Contains to
subfolders: Files and
Installshield.
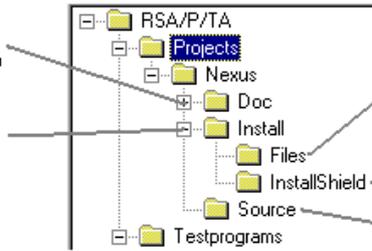
Files folder.
Should contain all files needed to
install software. Contains subfolders
if more than one program belongs to
the project.

Contains installshield packages of files
located in the Files folder.

Source folder.
Contains source code associated with the
projects. Contains subfolders if more than
one program belongs to the project.

RSA/P/TA
  Projects
    Nexus
      Doc
      Install
        Files
        InstallShield
      Source
  Testprograms

2.5 LABELS

The only type of label that should be used is View Labels. These must be attached at the project or test-program level in the view hierarchy. In the View example above, labels should only be attached to the Nexus and CAL 120 352/1 views.