

Process Dashboard Users Manual

Table of Contents

1. Quick Overview

1.1. [A quick look at the tool](#)

1.2. [Overall goals](#)

1.2.1. [What is the PSP\(SM\)?](#)

1.2.2. [What is the TSP\(SM\)?](#)

1.2.3. [Why build a tool?](#)

1.3. [How To Install](#)

1.3.1. [System Requirements](#)

1.3.2. [Installation Instructions:](#)

1.3.3. [Add-on Process Sets](#)

1.4. [Using help](#)

1.4.1. [The Table of Contents view](#)

1.4.2. [The Index view](#)

1.4.3. [The Search Engine view](#)

1.4.4. [The navigation buttons](#)

1.4.5. [The print buttons](#)

1.4.6. [Context sensitive help](#)

1.5. [Quick start example: a PSP2.1 project](#)

1.5.1. [Step 1: Create a project in the dashboard](#)

1.5.2. [Step 2: Planning the project](#)

1.5.3. [Step 3: Designing the program](#)

1.5.4. [Step 4: Writing the code](#)

1.5.5. [Step 5: Compiling the program](#)

1.5.6. [Step 6: Testing the program](#)

1.5.7. [Step 7: Postmortem](#)

1.6. [Change History](#)

1.6.1. [Features new in version 2.0](#)

1.6.2. [Features new in version 1.15](#)

1.6.3. [Features new in version 1.14](#)

1.6.4. [Features new in version 1.13](#)

1.6.5. [Features new in version 1.12](#)

1.6.6. [Features new in version 1.11](#)

1.6.7. [Features new in version 1.10](#)

1.6.8. [Features new in version 1.9](#)

1.6.9. [Features new in version 1.8](#)

1.6.10. [Features new in version 1.7](#)

1.6.11. [Features new in version 1.6 \(released September 1, 2003\)](#)

1.6.12. [Features new in version 1.5 \(released October 3, 2002\)](#)

1.6.13. [Features new in version 1.4 \(released February 22, 2002\)](#)

1.6.14. [Features new in version 1.3.1 \(released August 14, 2001\)](#)

1.6.15. [Features new in version 1.3 \(released July 16, 2001\)](#)

- [1.6.16. Features new in version 1.2 \(released April 20, 2001\)](#)
- [1.6.17. Features new in version 1.1 \(released February 9, 2001\)](#)

[1.7. License Information](#)

- [1.7.1. SEI/CMU Materials](#)
- [1.7.2. GNU General Public License](#)
- [1.7.3. GNU Lesser General Public License](#)
- [1.7.4. Apache License, Version 2.0](#)
- [1.7.5. MIT License](#)
- [1.7.6. Berkeley Software Distribution \(BSD\) License](#)
- [1.7.7. Java Tutorial Code License](#)
- [1.7.8. DOM4J License](#)
- [1.7.9. Colt License](#)
- [1.7.10. overLIB License](#)
- [1.7.11. License Agreement for Extensions from Tuma Solutions](#)
- [1.7.12. PSP\(SM\) Process Distribution](#)
 - [1.7.12.1. License Agreement for PSP\(SM\) Materials](#)

[2. Planning your project](#)

[2.1. Using the hierarchy editor](#)

- [2.1.1. The File menu](#)
- [2.1.2. The Edit menu and toolbar](#)
 - [2.1.2.1. Deleting, moving, and renaming hierarchy nodes](#)
 - [2.1.2.2. Organizing your hierarchy](#)
 - [2.1.2.3. Creating projects that follow a defined process](#)
- [2.1.3. Process templates](#)
- [2.1.4. Processes PSP0 - PSP1.1](#)
- [2.1.5. The PSP3 process](#)

[2.2. The process scripts](#)

- [2.2.1. The PSP Process Scripts](#)
- [2.2.2. The "generic" process template](#)
- [2.2.3. The "timer" process script](#)
- [2.2.4. Custom process scripts](#)
- [2.2.5. Custom process vs Generic process?](#)
- [2.2.6. The Team Process](#)

[2.3. Entering process data](#)

- [2.3.1. The nature of the dashboard forms](#)
- [2.3.2. Frozen Calculations](#)
- [2.3.3. Read-only data fields](#)
- [2.3.4. Exporting form data](#)

[2.4. Using the Size Estimating Template](#)

- [2.4.1. Base Parts](#)
- [2.4.2. Parts Additions](#)
- [2.4.3. Reused Parts](#)
- [2.4.4. Performing the size/time estimation](#)
- [2.4.5. Entering Actual data during Postmortem](#)

[2.5. Using the PROBE Wizard](#)

- 2.5.1. [PROBE Wizard Step 1](#)
 - 2.5.2. [PROBE Wizard Step 2](#)
 - 2.5.3. [PROBE Wizard Step 3](#)
 - 2.5.4. [PROBE Wizard Step 4](#)
 - 2.5.5. [PROBE Wizard Step 5](#)
 - 2.5.6. [A sanity check](#)
 - 2.6. [Using the PROBE tool](#)
 - 2.6.1. [Filtering your data set](#)
 - 2.6.2. [Measures of correlation](#)
 - 2.6.2.1. [The correlation measurement](#)
 - 2.6.2.2. [The significance measurement](#)
 - 2.6.3. [Making a projection](#)
 - 2.6.4. [Visualizing the data](#)
 - 2.7. [Using the Task & Schedule Tool](#)
 - 2.7.1. [Columns in the Task List](#)
 - 2.7.2. [Columns in the Schedule List](#)
 - 2.7.3. [Potential mistakes](#)
 - 2.7.4. [Creating a schedule step by step](#)
 - 2.7.4.1. [Step 1: Add tasks to the task list](#)
 - 2.7.4.2. [Step 2: Estimate task time](#)
 - 2.7.4.3. [Step 3: Estimate calendar time](#)
 - 2.7.4.4. [Step 4: Track your progress](#)
 - 2.7.5. [Earned Value Date Calculations](#)
 - 2.7.5.1. [Plan Date](#)
 - 2.7.5.2. [Replan Date](#)
 - 2.7.5.3. [Forecast Date](#)
 - 2.7.5.4. [Optimized Forecast Date](#)
 - 2.7.6. [Earned Value Charts](#)
 - 2.7.6.1. [Cumulative Direct Time](#)
 - 2.7.6.2. [Cumulative Earned Value](#)
 - 2.7.6.3. [Earned Value vs. Cost](#)
 - 2.7.7. [Tracking Task Dependencies](#)
 - 2.7.7.1. [Misordered Dependencies](#)
 - 2.7.7.2. [Dependency Dates](#)
 - 2.7.8. [Prediction Ranges in the Task & Schedule Tool](#)
 - 2.7.8.1. [When are Prediction Ranges Calculated?](#)
 - 2.7.9. [Task & Schedule Collaboration](#)
 - 2.7.9.1. [Publishing Schedules](#)
 - 2.7.9.2. [Rolling Up Schedules](#)
 - 2.7.9.3. [Constraints and Limitations](#)
 - 2.8. [Data Freezing](#)
- 3. [Running the dashboard](#)
 - 3.1. [Selecting the active task](#)
 - 3.1.1. [Navigating Tasks Hierarchically](#)

- 3.1.2. [Navigating Within a Task List](#)
 - 3.1.3. [Selecting a Task Directly](#)
 - 3.2. [Using the play/pause button](#)
 - 3.2.1. [The play/pause button pull-down control](#)
 - 3.3. [The Percent Spent Indicator](#)
 - 3.3.1. [Editing the Time Estimate](#)
 - 3.4. [Accessing the Process Scripts](#)
 - 3.4.1. [Custom Hyperlinks](#)
 - 3.4.2. [The timing icons](#)
 - 3.4.3. [External documents](#)
 - 3.4.4. [Entering data into the forms](#)
 - 3.5. [Entering defects](#)
 - 3.5.1. [Defect Type](#)
 - 3.5.2. [Defect Date](#)
 - 3.5.3. [Phase Injected and Phase Removed](#)
 - 3.5.4. [Timing the Defect Fix](#)
 - 3.5.5. [Fix Count](#)
 - 3.5.6. [Fix Defect](#)
 - 3.5.7. [Entering a description](#)
 - 3.5.8. [Fixing multiple defects at a time](#)
 - 3.6. [The Task Note Indicator](#)
 - 3.6.1. [Linking to External Websites](#)
 - 3.7. [Using the completion checkbox](#)
 - 3.8. [The "C" \(configuration\) menu](#)
 - 3.9. [Using the time log editor](#)
 - 3.9.1. [Project hierarchy](#)
 - 3.9.2. [Date filter](#)
 - 3.9.3. [Time entries](#)
 - 3.9.4. [Control buttons](#)
 - 3.10. [Using the Defect Log Editor](#)
 - 3.10.1. [Configuring Defect Type Standards](#)
 - 3.11. [The System Tray Icon](#)
- 4. [Postmortem analysis](#)
 - 4.1. [Counting LOC](#)
 - 4.1.1. [The LOC Counter Tool](#)
 - 4.1.1.1. [Counting LOC in regular files](#)
 - 4.1.1.2. [Counting LOC changes from Subversion](#)
 - 4.1.2. [The Quick LOC Counter](#)
 - 4.1.3. [The LOC Counter Report](#)
 - 4.1.4. [Entering PSP Size Data](#)
 - 4.1.5. [Advanced Usage](#)
 - 4.2. [Data charts and reports](#)
 - 4.2.1. [The Project Plan Summary](#)
 - 4.2.2. [Marking your project complete](#)
 - 4.2.3. [The Time Log and Defect Log](#)

- 4.2.4. [Graphical charts](#)
- 4.3. [Exporting your data](#)
 - 4.3.1. [Exporting Team Project Data](#)
 - 4.3.2. [Advanced: Exporting Non-Project Data](#)
- 4.4. [Importing data](#)
- 4.5. [Exporting data to Excel \(PC only\)](#)
- 4.6. [Importing Historical PSP Data](#)
- 5. [Frequently Asked Questions](#)
 - 5.1. [Installation issues](#)
 - 5.2. [Online forms](#)
 - 5.3. [Files](#)
 - 5.4. [Other questions](#)
- 6. [Advanced Topics](#)
 - 6.1. [Team Use](#)
 - 6.2. [Backing up Dashboard Data](#)
 - 6.2.1. [Manual Data Backups](#)
 - 6.2.1.1. [Viewing the data in a manual data backup](#)
 - 6.2.1.2. [Restoring data from manual data backups](#)
 - 6.2.2. [Automatic, External Backups](#)
 - 6.2.2.1. [Additional Uses](#)
 - 6.2.3. [Automatic, Internal Backups](#)
 - 6.2.3.1. [Restoring data from automatic, periodic backups](#)
 - 6.3. [Localization Tool](#)
 - 6.4. [Editing Preferences](#)
 - 6.4.1. [User Interface](#)
 - 6.4.2. [Add-ons](#)
 - 6.4.3. [Backups](#)
 - 6.4.4. [Miscellaneous](#)
 - 6.4.5. [Advanced](#)
 - 6.4.6. [Advanced Configuration Settings](#)
 - 6.5. [Database Support in the Process Dashboard](#)
 - 6.6. [Using Multiple "To Date" Data Sets](#)
 - 6.6.1. [Creating Multiple "To Date" Data Sets](#)
 - 6.6.2. [Customizing "To Date" Data Sets](#)
 - 6.6.3. [Viewing Rolled Up Data](#)
 - 6.6.4. [Using Alternate "To Date" Data Sets](#)
 - 6.7. [Making a change to a script](#)
 - 6.8. [How the Dashboard finds process files](#)
 - 6.8.1. [Sharing changes with team members](#)
 - 6.8.2. [Some files are only read at start up](#)
 - 6.9. [Defining a Custom LOC Counter](#)
 - 6.9.1. [Step 1: Decide what lines of code should be counted](#)
 - 6.9.2. [Step 2: Create an XML definition for your language filter](#)
 - 6.9.3. [Step 3: Save your XML language filter definition](#)
 - 6.9.4. [Step 4: Restart the dashboard](#)

6.10. [Defining your own processes for use in the Process Dashboard](#)

6.10.1. [Step 1: Identify your process structure](#)

6.10.2. [Step 2: Create an XML definition for your process](#)

6.10.3. [Step 3: Save your XML process descriptor](#)

6.10.4. [Step 4: Restart the dashboard](#)

6.10.5. [Advanced/Optional: Defining HTML scripts](#)

6.10.6. [Additional Advanced Topics](#)

6.11. [Size Metrics API](#)

6.11.1. [HTTP POST](#)

6.11.1.1. [Request Parameters](#)

6.11.1.2. [Errors](#)

6.11.1.3. [Results](#)

1. Quick Overview

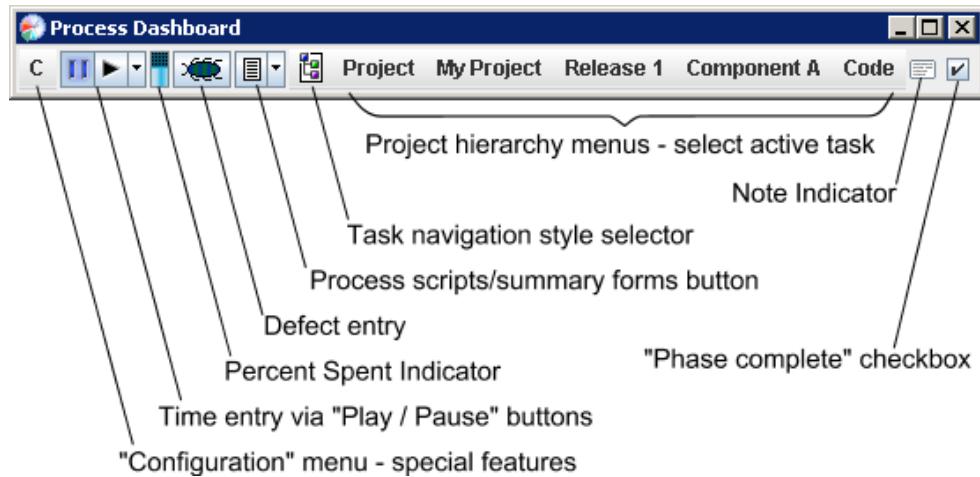
We feel that the PSP^(SM) and the TSP^(SM) are remarkable technologies that can change the face of the software industry, and we share the SEI's zeal to promote their widespread use. We feel that a freely available, powerful support tool could help to remove one of the most significant barriers to PSP / TSP adoption. We therefore aim to develop a world-class tool under the open-source model, and distribute it freely to anyone using the PSP and/or TSP. We feel that this is the least we can do to thank the SEI for developing and distributing these remarkable processes.

You have installed an optional module that contains PSP^(SM) scripts, forms, and other materials. [Special permission](#) to use "PSP Materials.doc" copyright © 2006 by Carnegie Mellon University, in this module is granted by the Software Engineering Institute. The use of these materials is subject to [license](#).

Personal Software Process^(SM), PSP^(SM), Team Software Process^(SM), and TSP^(SM) are service marks of Carnegie Mellon University. The open source team that writes the Process Dashboard is not affiliated with Carnegie Mellon University.

1.1. A quick look at the tool

The personal Process Dashboard displays a streamlined user interface that helps individuals to edit plans, collect metrics, and analyze their data.



The Process Dashboard attempts to keep itself as small as possible to save valuable screen real estate for the actual work of software development. In a very small space the dashboard gives access to a stopwatch-like timer for timing activities, a defect entry dialog for capturing defect information, scripts and forms for following defined processes, a handy "completion checkbox" that allows a quick and easy way to tell the tool that a phase of development is complete, menus that allow navigation through the project work breakdown structure hierarchy, and a configuration menu to allow access to other features of the tool.

Pick an item for further help:

[Configuration menu](#)

[Play/Pause button](#)

[Percent Spent indicator](#)

[Defect entry](#)

[Process scripts and tools](#)

[Project hierarchy menus](#)

[Note indicator](#)

[Completion checkbox](#)

1.2. Overall goals

We feel that the PSP and the TSP are remarkable technologies that can change the face of the software industry, and we share the SEI's zeal to promote their widespread use. We feel that a freely available, powerful support tool could help to remove one of the most significant barriers to PSP / TSP adoption. We therefore aim to develop a world-class tool under the open-source model, and distribute it freely to anyone using the PSP and/or TSP. We feel that this is the least we can do to thank the SEI for developing and distributing these remarkable processes.

1.2.1. What is the PSP^(SM)?

The Personal Software Process^(SM), created by Watts Humphrey of the Software Engineering Institute, introduces software engineers to a disciplined process for software engineering. Engineers using the PSP to develop software follow defined processes and collect detailed metrics on the time required to produce a product, the defects that were injected and removed at various stages in development, and the size of the finished product.

These metrics are then analyzed using statistical methods, enabling engineers to produce highly accurate estimates based on historical data, track progress and quality of a project in progress, predict schedule impacts, and predict the quality of a finished software product. The PSP encourages engineers to quantitatively determine ways to improve their process.

For more information on the PSP, see the official PSP website at <http://www.sei.cmu.edu/tsp>, maintained by Carnegie Mellon University.

1.2.2. What is the TSP^(SM)?

The Team Software Process^(SM), also created by Watts Humphrey, is a process framework for teams of PSP-trained engineers. The TSP scales well and can be used by teams of 3 to 20 people to develop software products of significant size and complexity.

Teams of engineers using both the PSP and the TSP to develop software have consistently observed remarkable improvements in their work:

- End-to-end productivity improvements of 20% - 150%
- 99% reductions in system test defects, and virtually defect-free released software.
- Schedule estimates accurate to within 10% (4% error on average)

1.2.3. Why build a tool?

Both the PSP and the TSP require the collection and analysis of metrics at a *very* fine-grained level. Further, TSP requires teams to roll-up individual metrics to produce team metrics. Once data are collected at this level, statistical analyses of the data permit remarkable planning, tracking, prediction, and control of software products and projects.

These metrics collection and analysis processes, however, are not trivial. In any real-world project, tool support for the PSP and TSP become important considerations. Although studies have demonstrated that people can maintain their productivity when using the PSP *without* tool support, the "frustration factor" inherent with such an approach tries the patience of all but the most disciplined engineers, making PSP behaviors difficult to sustain.

Ideally, PSP/TSP practitioners would like to have a support tool that:

- Allows data at the personal level to be collected quickly and easily, with minimal frustration.
- Can be integrated with existing development environments and project management tools.
- Allows individuals to collaborate on the execution of a process (even if they are geographically distributed).
- Allows data at the individual level to be rolled up to produce team-level or organizational-level metrics.
- Protects the privacy of individuals, and prevents unauthorized people from seeing or using their data.
- Supports powerful analyses of data at the individual, team, organizational, and enterprise levels, and allows existing (external) applications to access the data (while still maintaining the security mentioned above).
- Supports arbitrary processes (including processes that have yet to be written), and arbitrary new process tools.

This initiative aims to develop an open-source tool to meet all of these needs.

Development on the dashboard is ongoing. To obtain the latest version of the dashboard, visit our web site at: <http://www.processdash.com>.

For more information on the TSP, see the official TSP website at <http://www.sei.cmu.edu/tsp>, maintained by Carnegie Mellon University.

1.3. How To Install

1.3.1. System Requirements

The Process Dashboard is 100% Java, and will run on any system with a Java 2 compliant virtual machine. This includes Windows, Linux, Unix, and Macintosh systems. For access to forms and scripts, a supported web browser is also required.

To use the process dashboard, you must **first** install the following software:

- The Java Runtime Environment, version 1.5 or greater (available from Oracle at <http://www.java.com/>)
- Make certain that JavaScript is enabled in your web browser. (This is typically the default setting, so you usually don't need to worry about enabling these unless you explicitly disabled them in the past.)
- If you connect to the internet via a proxy server, you may need to tell your browser to bypass the proxy for the address "localhost". The instructions below may not apply to your browser word-for-word, but should hopefully get you pointed in the right direction:
 - For Internet Explorer, choose "Tools → Internet Options". Click on the "Connections" tab, then on the "LAN Settings" button. Make certain the "Bypass proxy server for local addresses" box is **checked**.
 - For Firefox, no changes are typically necessary. If you have trouble viewing dashboard scripts and forms, open the Options dialog. On the Advanced pane, choose the Network tab and click the button to configure settings for how Firefox connects to the Internet. On that Connection Settings dialog, ensure that the "No Proxy for" box contains "localhost".

1.3.2. Installation Instructions:

1. Before attempting to install the dashboard, make certain you have installed the required software listed above. In particular, the Process Dashboard installer program will not run if you do not have the [Java Runtime Environment](#) installed on your computer.
2. Download the Process Dashboard installer program from <http://www.processdash.com/download> and follow the directions found there.
3. Please consider subscribing to our announcements mailing list. We will send email to this mailing list when new versions of the tool are available. You can subscribe to the list at <http://lists.sourceforge.net/lists/listinfo/processdash-announce>

1.3.3. Add-on Process Sets

The Process Dashboard was written to allow various processes to be followed, not just the PSP processes. As such, we can offer add-on process sets that can be installed into the dashboard. Check the downloads section of our website at <http://www.processdash.com/download> for the latest add-on process sets.

To install an add-on process set:

- Choose "Help → About" from the "C" menu, and click the "Configuration" tab. Just above the list of add-ons, a paragraph will tell you where the Process Dashboard is installed. Find that directory on your computer. It should contain a small number of files, including one called "pspdash.jar".
- Download the add-on process set from our website. When your browser asks you whether you want to save the file or open it; choose **Save**. Save the file into the directory you found in the previous step. (In other words, the *pspdash.jar* file and the add-on process file should be side-by-side in the same directory. **Do not** overwrite the *pspdash.jar* file, just save the process file into the same directory.)
- If the dashboard is running, you will need to restart it for the new processes to be found.

1.4. Using help

Starting the help system is as easy as clicking "help" on the "C" (configure) menu. This online help is available via two methods:

- If you have JavaHelp™ installed on your computer, the dashboard will use it to provide context-sensitive help with an index and full-text search capabilities.
- If JavaHelp is not present on your computer, the dashboard will display help topics in your web browser.

The full-text search capability is a very useful feature; for this reason, we strongly recommend installing JavaHelp. The [installer program](#) for the Process Dashboard can automatically download and install the JavaHelp software if you select that option. The paragraphs below describe the behavior of the help when JavaHelp has been installed. (*JavaHelp is a trademark of Oracle.*)

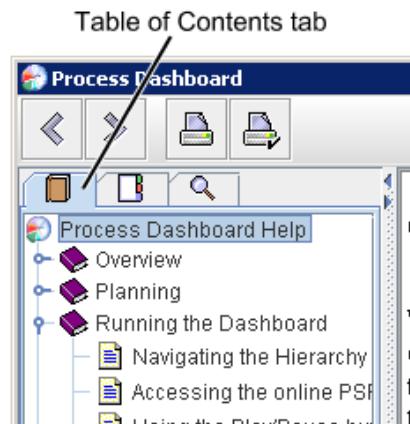
Currently the help system provides three ways of accessing help topics:

- A table of contents view
- An index view
- A search engine view

These three views are accessed via the three tabs at the top of the left screen pane.

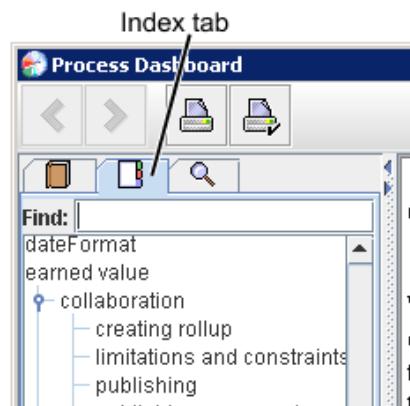
1.4.1. The Table of Contents view

The table of contents provides a hierarchical organization of the help topics. Items that are shown as books represent logical groupings of help topics and can be clicked to reveal more detailed help.



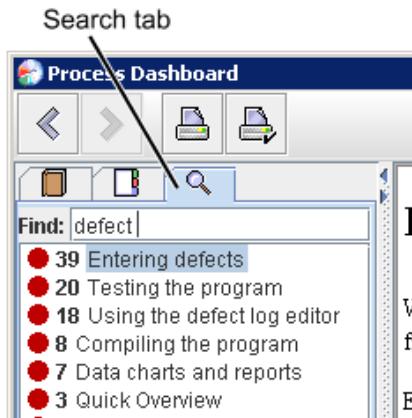
1.4.2. The Index view

The index view provides an alphabetical listing of various keywords on which to find help.



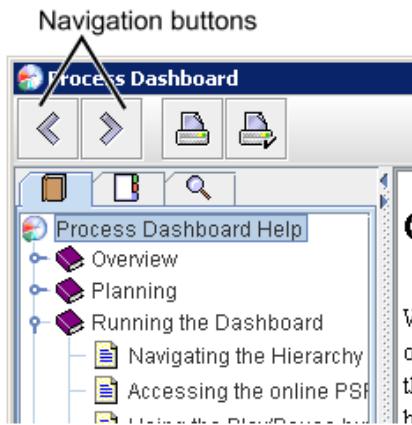
1.4.3. The Search Engine view

The search engine view provides an entry blank where you can enter words you would like the help system to search for within the help topics. After entering search criteria you will be presented with a list of help topics that contained the word or words entered. To the left of each item will be a number indicating how many times the word(s) entered were found in that particular help topic.



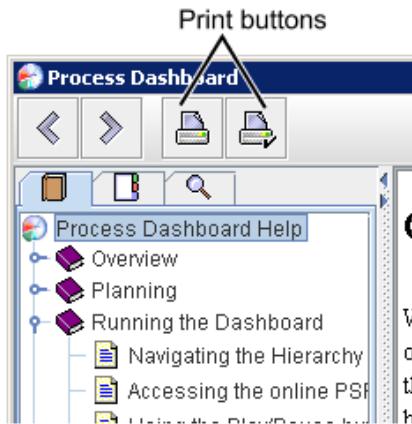
1.4.4. The navigation buttons

The help viewer also provides two navigation buttons at the top left. They allow for the familiar "back" and "forward" functionalities that are so common in modern web browsers.



1.4.5. The print buttons

The two buttons to the right of the navigation buttons are a "Print Topic" button and a "Page Setup" button. These buttons allow you to print a help topic for later reading.



1.4.6. Context sensitive help

There is also some context sensitive help available. If you press the F1 key while viewing the dashboard itself or one of its tools (PROBE, time log editor, etc.), the dashboard will startup a help screen pertinent to that tool.

1.5. Quick start example: a PSP2.1 project

This "Quick Start" example is provided to illustrate the most commonly used features in the Process Dashboard, and walk you through their use on a typical project. PSP2.1 is used for this example, because most PSP-trained engineers will choose to use PSP2.1 for their daily work.

Note: The PSP2.1 process helps developers improve the quality of their plans by analyzing data from previously completed projects. To illustrate how this functionality works in the Process Dashboard, the following quick start example illustrates dashboard behavior for a user who has already completed several historical projects.

When you first begin using the dashboard, you won't have any historical project data in the dashboard. As a result, brand new users of the dashboard should not expect to be able to follow these steps verbatim and see the same results. (For example, the PROBE wizard will behave very differently for an individual with no historical data.)

If you have been using PSP for some time, you may be able to use the "[PSP Data Import](#)" template to enter your historical data into the dashboard. This is not strictly required, but it can help you to benefit immediately from the wealth of historical data you have collected in the past.

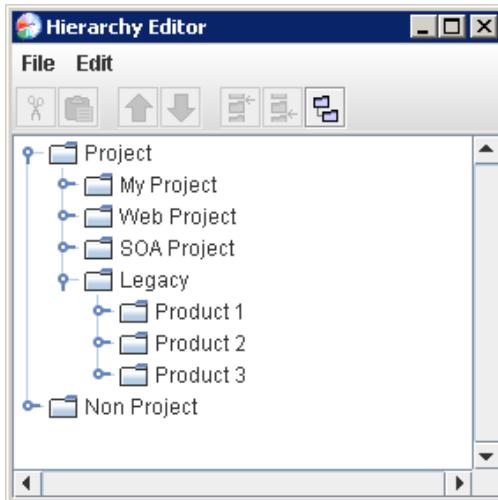
If you have never used PSP before, you are strongly encouraged to use the PSP0 process for your first project, and the PSP0.1 process for your second and third projects. You should also take a PSP course to learn how to use the PSP processes.

Here are the steps that we will take in this example:

1. [Creating the project](#)
2. [Planning the project](#)
3. [Designing the program](#)
4. [Writing the code](#)
5. [Compiling the program](#)
6. [Testing the program](#)
7. [Postmortem](#)

1.5.1. Step 1: Create a project in the dashboard

The first thing we need to do to get started is to create a new PSP2.1 project in the dashboard. To do that, we will start the [hierarchy editor](#) by choosing Hierarchy from the [Configure menu](#).

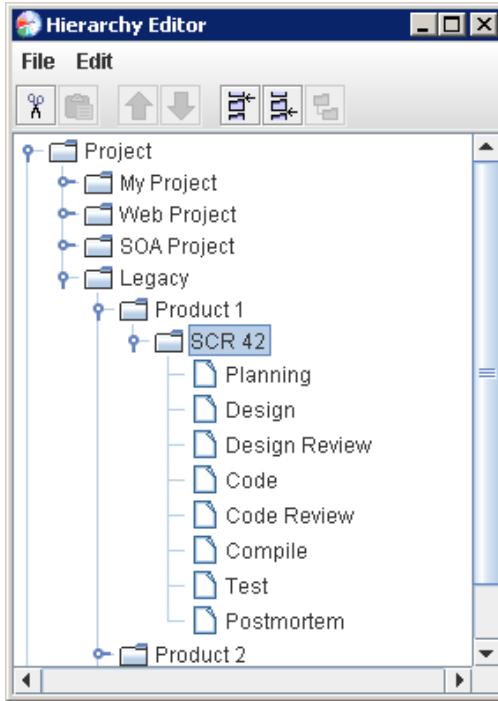


We need to decide where to place this project in our work breakdown structure. For this example, we will imagine that we are responsible for supporting a legacy application called "Product 1". Our users have identified a problem, and issued "Software Change Request (SCR) #42".

As you can see in the image above, this user has created "nodes" or "folders" in their hierarchy to keep things nicely organized. We will

place this change request under the "Project / Legacy / Product 1" node in our work hierarchy. To do that we will select the "Product 1" node by clicking it once. Next we will pick the "Edit" menu and choose "Add Template" followed by "PSP2.1". This will add a PSP2.1-based project to the project hierarchy as a child of "Product 1". When the new project is added, it is given a default name. We can change the name to better describe what we are working on. In this case, we will choose a helpful name, like "SCR 42". For more information on using the hierarchy editor, and creating both "nodes" and projects, see the [Using the hierarchy editor](#) help topic.

Once the project has been created and renamed, click on the little expansion control to the left of the "SCR 42" project to expand its child nodes in the hierarchy tree. There you can see that the dashboard has placed all the process phases for a PSP2.1 project.



Now we can save our changes and close the hierarchy editor via the File menu.

1.5.2. Step 2: Planning the project

Now that the project exists, we can start the planning of our work. The PSP provides scripts for planning, development, postmortem work, etc., so let's go look at the planning script for our new PSP2.1 project. This is done by selecting our newly created project via the [hierarchy menus](#) (the dashboard will automatically select the planning phase -- the first development phase of a PSP2.1 project), then clicking the scripts button. We should also click the [play button](#) on the dashboard to start timing our planning activities. When the play button is pressed, a quiet "clicking" sound will be heard. This sound will be heard anytime the dashboard switches phases of activity while the clock is running.



The dashboard responds by bringing up the PSP2.1 planning script in a web browser. Reading through the script, we see that once we understand the requirements for this program, we need to do a size estimate. To guide our estimation, we can use the PROBE method. Following the [PROBE method](#) link on the planning script brings up the PROBE Estimating Script.

Reading through this script we see that once we have produced a conceptual design, we need to fill out the Size Estimating Template. The dashboard provides both the template itself and the instruction script for its use. Following the link for the [Size Estimating Template](#) brings up a dynamic HTML version of the template. For ready reference, the template has a link near the top to display the Size Estimating Template instructions form. If you click on this link, a second browser will be started so that you can have the instructions and the template side by side on your screen. For more information on the Size Estimating Template, see the [Size Estimating Template](#) help topic.

Project Name /Project/My Project/Release 1/Component A
 Project Owner My Name
 Size Measure LOC

BASE PARTS				Estimated				Actual			
BASE	DEL.	MOD.	ADDED	BASE	DEL.	MOD.	ADDED	BASE	DEL.	MOD.	ADDED
0	0	0	0	0	0	0	0	0	0	0	0

Instructions link on the Size Estimating Template

The Size Estimating Template Instructions direct us to count the base LOC and enter it in the Base Parts section if this is an enhancement of a previous program. We're modifying an existing program, so we use the [LOC counter](#) to measure the size of the components we plan to modify, and enter that number in this section.

The Parts Additions section will be the places for most of our estimating. Based on our conceptual design, we enter several lines describing the work we plan to do. For each item, we estimate the type, number of methods, and LOC estimates. The "add more rows for new objects" link will expand the form with more entries if chosen.

We don't anticipate reusing any code, so we will skip that section.

Moving on to the size calculation, we need to calculate the estimated added & modified LOC using PROBE. There are at least two ways to do this: with the PROBE Wizard or with the PROBE tool. (The PROBE Wizard automates the standard PSP PROBE process, and is simpler to use; the PROBE tool is more complex to use but provides more power for custom statistical analyses.) Most people will want to use the PROBE Wizard for all their planning needs, so we will use the wizard for this example. After you complete the top portion of the Size Estimating Template, you can launch the PROBE wizard by clicking on the hyperlink halfway down the page.

REUSED PARTS

[add more rows for reused parts...](#)

Total: Estimated SIZE: 0 Actual SIZE: 0 [0]

Instructions: During the planning phase, complete the "Estimated" columns in the form above, then use the [PROBE Wizard](#) to fill out the fields below.

Added Size (A):	A = BA + PA
Estimated Proxy Size (E):	E = PA + PA + M

SIZE	TIME
69.3	69.3

PROBE Wizard link on the Size Estimating Template

The PROBE Wizard consists of a series of dialogs that guide you through the PROBE process.

 **PROBE Wizard**
/Project/Legacy/Product 1/SCR 42

Size and time estimating are complex skills, best developed through experience. As you strive to improve your estimating skills, your primary goal should be estimating accuracy.

The Proxy Based Estimating process (PROBE) includes statistical methods that can help you to improve your estimating accuracy over time. These methods analyze your past estimating performance, and attempt to mathematically remove any consistent estimating bias found.

This wizard will help you to perform this portion of the PROBE process. Keep in mind that:

- PROBE is not magic - you must actively exercise your engineering judgement to ensure that you produce sensible estimates.
- The PROBE methods implicitly assume that the project you are currently estimating is similar to the projects you have completed in the past. If this is not true, use PROBE-generated estimates with caution.
- PROBE works best when you are consistent in your use of the Size Estimating Template.

To begin, click the "Continue" button.

[Continue](#)

The PROBE script gives several ways to calculate an estimate. The PROBE Wizard displays these methods side by side and ranks them, advising us which method may be the most reliable. By simply following the prompts, we decide in this example to select method C2 for size and then choose C2 for time. As we make these choices, the resulting values are automatically written into the appropriate fields of the Size Estimating Template.

After our planning efforts on this form it looks like the following:

Size Estimating Template - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost:2468/Project/Legacy/Product+1/SCR+42//psp2.1/sizeest.class

Size Estimating Template [\(instructions...\)](#)

Project Name	/Project/Legacy/Product 1/SCR 42
Project Owner	My Name
Size Measure	LOC

BASE PARTS				Estimated				Actual				
BASE	DEL.	MOD.	ADDED	BASE	DEL.	MOD.	ADDED	BASE	DEL.	MOD.	ADDED	
345	0	0	10.9	345	0	0	10.9	0	0	0	0	
add more rows for base parts...				Total:								

PARTS ADDITIONS				Estimated				Actual			
TYPE	ITEMS	REL SIZE	SIZE	NR	SIZE	ITEMS	NR	SIZE	ITEMS	NR	
Calculation	1	1	1	1	1	1	1	1	1	1	

[http://localhost:2468/help/book.html?](#)

Logic for new calculation
Presentation templates
Customization UI
Customization data structures
add more rows for parts additions...

Calculation	Value	Very Large	54			
I/O	2	Large	43.2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I/O	1	Very Small	28.9	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Data	3	Medium	26.5	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
		Total:	153	0		

REUSED PARTS	Estimated SIZE	Actual SIZE
add more rows for reused parts...	0	0
	Total: 0	0

Instructions: During the planning phase, complete the "Estimated" columns in the form above, then use the [PROBE Wizard](#) to fill out the fields below.

Added Size (A):
Estimated Proxy Size (E):
PROBE estimating basis used: (A, B, C, or D)
Correlation: (r^2)
Regression Parameters:
Projected Added and Modified Size (P):
Estimated Total Size (T):
Estimated Total New Reusable (NR):
Estimated Total Development Time:
Prediction Range:
Upper Prediction Interval:
Lower Prediction Interval:
Prediction Interval Percent:

$A = BA + PA$
 $E = BA + PA + M$
A, B
 B_0 (size and time)
 B_1 (size and time)
 $P = B_0 size + B_1 size * E$
 $T = P + B - D - M + R$
(sum of NR items)
 $Time = B_0 time + B_1 time * E$

SIZE	TIME
164	
164	
A	B
0.87	0.84
26	1.4
1.08	5.42
202	
547	
0	

Range	20:36
36.4	4:08
239	24:44
166	16:28
70%	70%

[View PROBE Report](#)

Adapted from "PSP Materials," copyright © 2005 Carnegie Mellon University. Used by permission.

[Unlock read-only data](#) Export to: [HTML](#) [Excel](#) [Help...](#)

Done



Click the Completion checkbox to finish

Note also that when the dashboard moves into a new phase while the timer is running, a little "clicking" noise is made. This little bit of feedback assures us that the timer is still running.

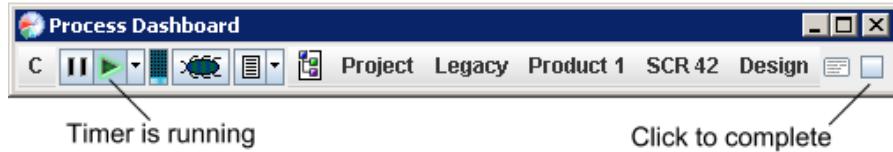
1.5.3. Step 3: Designing the program

The PSP provides Development scripts for all the levels. To access the development script for this program, we can click the back button on the web browser to take us back to the main Process script which contains a link to the Development script, or we can just press the

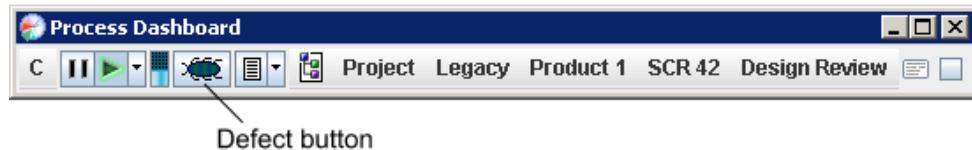
scripts button again. When the scripts button is pressed, the dashboard always opens the web browser on a form that is appropriate for the current process phase. In this case, it will open to the Development script. We can also use the pull-down control on the scripts button. Pressing the pull-down will bring up a menu of the scripts and forms available for the project. Picking a script or form from the menu will open that form in a web browser. In this case, picking "PSP2.1 Development Script" will take us to the development script.

While designing this program, we will be sure to keep the dashboard's timer running. This can be confirmed by noticing the state of the play/pause button. For more information on the play/pause button, see the [play/pause button](#) help topic.

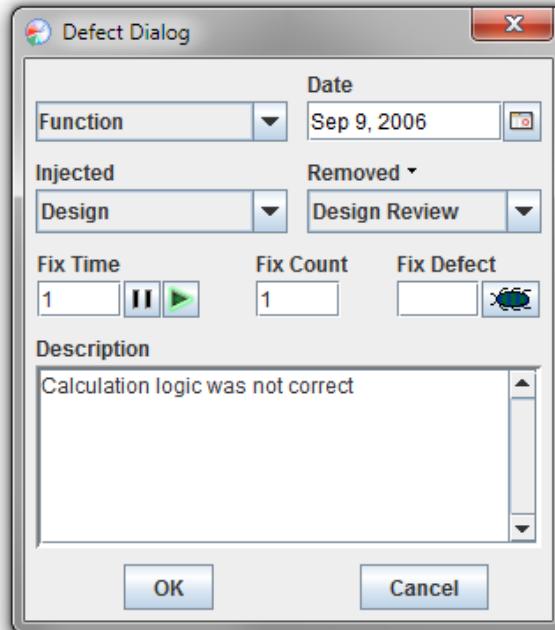
Once finished with the program design, we click the completion checkbox to mark the design phase complete.



The dashboard will move us to the next phase, which is design review. We perform a thorough review of our design, and find several defects. To enter each defect, we click the defect button.



The defect dialog appears, allowing us to enter all the pertinent data about the defect.



After you enter the data for a defect, click the OK button to save it. Then, if you need to enter another defect, click the defect button on the main dashboard window again. For more information about entering defects into the dashboard, see the [Entering Defects](#) help topic.

Of course, several more defects were found during the review, but we won't show them all.

Once the design review is finished, we click the completion checkbox and move along to coding.

1.5.4. Step 4: Writing the code

During software development, you will inevitably spend a sizeable amount of time in the coding phase. Even though you have your head down and your mind focused on code, don't forget to follow the disciplined PSP behaviors!

In particular, get into the habit of faithfully starting and stopping the timer. If you are interrupted during your work, just click the play/pause button. When the interruption ends, press the button again. The dashboard will automatically record rows in your time log to capture the time you are spending on the work.



Remember to start and stop the timer as needed

If you forget to start or stop the timer, you can fix these mistakes with the [time log editor](#).

Also, it is common when coding to discover defects that were injected in earlier phases. For example, you might find an error in your design. Use the defect button to record these defects as you find and fix them.

When we are done coding, we mark the phase complete and move on to code review. Of course, while reviewing the code, we faithfully log our time and defects. Finally, we mark the code review phase complete, and move on to the compile phase.

1.5.5. Step 5: Compiling the program

If your code review was perfect, your code will compile cleanly the first time! Otherwise, you'll need to click the defect button to record each compile defect.



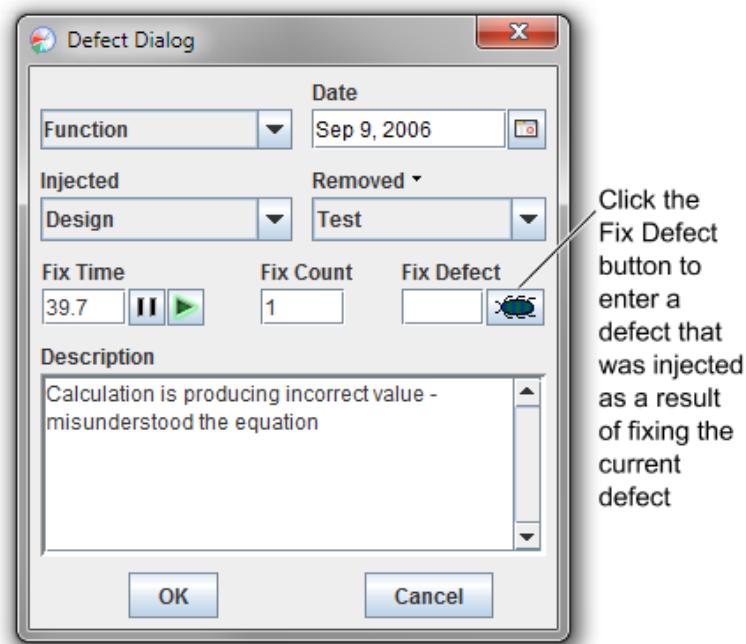
Once all the defects that were found in compile have been entered and the program compiles cleanly, clicking the completion checkbox moves us onward into the testing phase.

1.5.6. Step 6: Testing the program

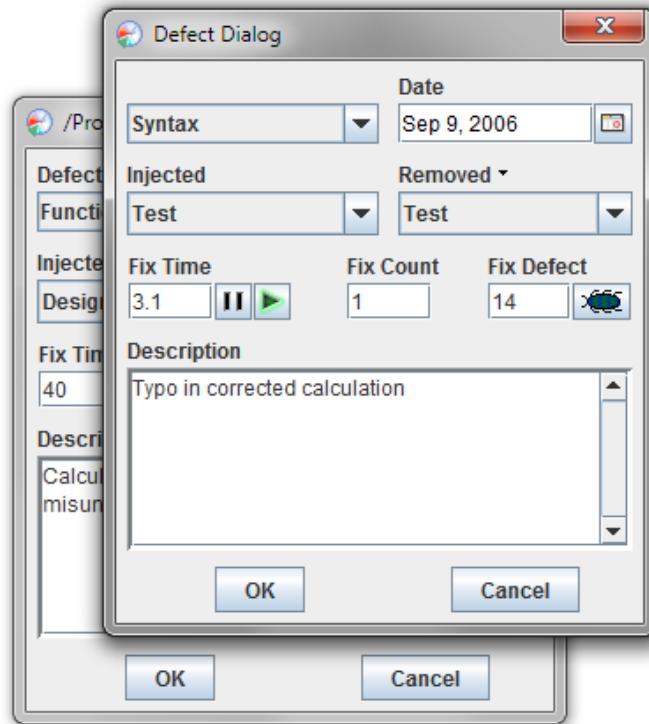
When using the PSP, you should strive to remove defects before the testing phase whenever possible. Nevertheless, you will still encounter defects during testing, and you should log these defects using the defect button. Since we have already seen an entry of a defect, we will take this opportunity to look at a more complicated situation.

Consider the following circumstance: while testing, you find a defect and put in a fix. Upon compiling your fix you discover that you have inadvertently injected another defect (perhaps a syntax error in your coding). This is what the PSP calls a "fix defect" - a defect injected while entering a fix for another defect.

The dashboard has a very handy feature to help in this regard. While the defect dialog for one defect is open, you can click on the little defect button in the corner to enter a new defect for the fix of the current defect.



After clicking the Fix Defect button, a new defect dialog will open for entry of data for this new defect.



While this new defect dialog is open, the timer for the previous defect dialog is paused. Also, notice that the number of the previous defect is already entered in the Fix Defect blank. Once the information for this new defect is entered and the user clicks OK, the previous defect dialog will pop to the front of the screen and the timer will resume. For more information about this behavior, see the [Entering Defects](#) help topic.

Back to our example. Once the program has been sufficiently tested and all the defects have been entered, hit the completion checkbox again to move into Postmortem.

1.5.7. Step 7: Postmortem

The PSP provides Postmortem scripts for all of the PSP levels. To get to the postmortem for the current program, we can click the link at

the bottom of the development script. Also, clicking the script button on the dashboard while in the postmortem phase will bring up the postmortem script, or using the pull-down control on the scripts button will display an option for the postmortem script as well.

After making sure that the entry criteria are met, the next step that must be done is to enter the actual size data for the program. You may find the dashboard's [LOC counter](#) useful for measuring added, deleted, and modified lines of code. Enter the actual values for base, deleted, modified, total, base additions, new and reused LOC on the Size Estimating Template form, which can be reached by clicking the appropriate link from the postmortem script.

Size Estimating Template - Mozilla Firefox

File Edit View History Bookmarks Tools Help

Project Name /Project/Legacy/Product 1/SCR 42
Project Owner My Name
Size Measure LOC

BASE PARTS

BASE	Estimated			ADDED
	DEL.	MOD.		
345	0	0	10.9	
345	0	0	10.9	

[add more rows for base parts...](#) Total: 10.9

PARTS ADDITIONS

TYPE	Estimated			NR
	ITEMS	REL. SIZE	SIZE	
Calculation	1	Very Large	54	<input type="checkbox"/>
I/O	2	Large	43.2	<input type="checkbox"/>
I/O	1	Very Small	28.9	<input type="checkbox"/>
Data	3	Medium	26.5	<input type="checkbox"/>

Total: 153

SIZE	Actual			NR
	ITEMS	SIZE		
0	0	0	<input type="checkbox"/>	
0	0	0	<input type="checkbox"/>	
0	0	0	<input type="checkbox"/>	
0	0	0	<input type="checkbox"/>	

[add more rows for parts additions...](#)

REUSED PARTS

SIZE	Estimated			NR
	ITEMS	SIZE		
0	0	0	<input type="checkbox"/>	
0	0	0	<input type="checkbox"/>	

Total: 0

SIZE	Actual			NR
	ITEMS	SIZE		
0	0	0	<input type="checkbox"/>	
0	0	0	<input type="checkbox"/>	

[add more rows for reused parts...](#)

Enter actual Size data

After you enter actual size information for the various rows on the Size Estimating Template, you will also need to open "Size" section of the Project Plan Summary form. Review the numbers in the Actual column (correcting any if necessary), and enter the Total Actual Size in the appropriate field.

PSP2.1 Project Plan Summary

Program Size

	Plan	Actual	To Date
Base (B)	345	345	0
Deleted (D)	0	5	0
Modified (M)	0	4	0
Added (A)	202	218	020
Reused (R)	0	0	0
Added and Modified (A+M)	202	222	020
Total Size (T)	547	558	020
Total New Reusable	0	0	0
Estimated Proxy Size (E)	164	210	682
<i>Upper Prediction Interval (70%)</i>	239		
<i>Lower Prediction Interval (70%)</i>	166		

Adapted from "PSP Materials," copyright © 2006 Carnegie Mellon University. Used by permission.

Once the actual size information (including measures of reuse) has been entered, we can mark the postmortem phase complete. Once all the phases in the process have been marked complete, the dashboard will mark the project itself complete as well. We can double check that it has been marked by looking at the Project Completion checkbox on the Project Plan Summary.

Project Owner	Sample
Start Date	Jun 25, 2001 09:42:27 AM
Completion Date	Jan 29, 2002 05:59:03 PM
Keywords	
Project completion checkbox	

If the Project Complete checkbox had not been automatically checked, it would indicate that we had already overridden this automatic behavior. You can learn more about the relationship between marking the phases complete and marking the entire project complete by reading the [Using the completion checkbox](#) help topic.

Throughout your project, you can view project metrics in the Project Plan Summary form.

PSP2.1 Project Plan Summary - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost:2468/Project/Legacy/Product+1/SCR+42//cms/psp2.1/summary

PSP2.1 Project Plan Summary

Overall Metrics

Summary	Plan	Actual	To Date
Size/Hour	9.82	11.6	10.2
Planned Time	20:36		94:50
Actual Time		19:05	106:20
CPI (Cost-Performance Index)		1.08	0.89
% Reuse	0%	0%	0%
% New Reusable	0%	0%	0%
Test Defects/KLOC or equivalent	15.7	9.01	15.7
Total Defects/KLOC or equivalent	79.5	67.6	79.5
Yield %	32.1%	85.7%	32.1%
<i>% Appraisal COQ</i>	4.86%	18.4%	4.86%
<i>% Failure COQ</i>	23.7%	20.8%	23.7%
<i>COQ A/F Ratio</i>	0.2	0.89	0.2
<i>PQI</i>	0.0	0.33	0.0

Planned Quality Profile

Quality Profile

Adapted from "PSP Materials," copyright © 2005 Carnegie Mellon University. Used by permission.

[Unlock read-only data](#) Export to: [HTML](#) [Excel](#) [Help...](#)

Done

The Process Dashboard also offers several other aids for analyzing postmortem data. Choosing Data Analysis from the "C" menu will open the web browser to a set of pages that contain many useful graphs of process data.



This Data Analysis page provides several different categories of charts to allow easy analysis of various metrics measured in the PSP. Clicking on one of the chart thumbnails in the right pane will bring up a full-sized view of that chart. For more information on the types of data that can be analyzed via these reports and graphs, see the [Creating Reports and Charts](#) help topic.

That's it for this example. We hope it has been helpful.

1.6. Change History

1.6.1. Features new in version 2.0

- The Process Dashboard now includes a powerful new [Data Warehouse](#) component. This component stores team project data in a relational database, making it possible for external analysis and reporting tools to access project data via SQL queries.
- The various reports, charts, and calculations in the Team Dashboard have been rewritten to take advantage of the Data Warehouse component. As a result, Team Dashboard reports are displayed 100 to 1000 times faster than before. Team Dashboard memory usage has also decreased significantly. Together, these improvements introduce a dramatic change in the scalability of the Team Dashboard for large and long-running project teams.
- For over a decade, the Team Dashboard has reused the same graphical user interface as the "personal" Dashboard. With this release, the Team Dashboard now displays a redesigned user interface that focuses on the needs of coaches, team leaders, team role managers, and other users of consolidated team data. The new interface provides quick access to team tools and reports, improving productivity of existing teams, and reducing the learning curve for teams that are new to the Process Dashboard.
- Historically, some teams have observed extremely long startup times for the Team Dashboard - especially when opening the data over a VPN, WAN, or other slow network. This version introduces a change to the storage strategy for team projects, enabling significantly faster startup times for the Team Dashboard.
- When the Team Dashboard and Process Dashboard shut down, a task runs to export data for various team projects. Historically, as more team projects are added to a particular dashboard, this export step takes longer and longer. In this version, the export task has been streamlined significantly, allowing the dashboard to shut down much faster.
- Significant improvements have been made to the efficiency of the earned value calculation logic, enabling earned value reports to display much faster than before.
- The Defect Log Editor now includes an option to import defects from the Review Board code review tool.
- A new REST API has been provided that makes it possible for external tools (running on the same computer as the personal Process Dashboard) to add entries to the Size Inventory Form for a Team Project.
- The splash screen appears more quickly after clicking an icon to launch the Process Dashboard.

1.6.2. Features new in version 1.15

- It is now possible for multiple individuals to edit the Work Breakdown Structure at the same time:
 - When you save changes to the WBS, the application will check to see if other individuals have recently saved changes as well. If they have, the save operation will merge their changes into your view.
 - If you wish to merge other people's recently saved changes into your view without saving your own in-progress edits, a new "Refresh Data" option is available on the "File" menu for this purpose.
 - In either case, if the merge identifies editing conflicts (where you and another individual have made conflicting edits to the same value), warning messages will be displayed. These warnings include hyperlinks that help you to locate and recover from the editing conflict.
- The Work Breakdown Structure Editor now includes a feature to save data to a ZIP file. This feature can be used to:
 - Create multiple alternative plans during a launch
 - Explore what-if replanning scenarios in the middle of a team project iteration
 - Save edits if the network server is temporarily unreachable
 - Copy components tasks, milestones, team members, and workflows into a new team project iteration
- Many other significant changes were made to the Work Breakdown Structure Editor:
 - The WBS Editor now includes a filtering function. This makes it possible to focus in on the set of components and tasks that contain certain words in the name or notes, that are assigned to various people, that are complete/incomplete, or that have certain labels or milestones.
 - The WBS Editor now includes a search feature. This makes it possible to quickly find components or tasks whose name or notes contain particular words.
 - The colored balancing bars in the WBS Editor have been enhanced to calculate dates that align more closely to the "Replan" dates in the team earned value report.
 - The colored balancing bars can now display the typical number of hours per week each team member has in their schedule. A new menu option makes it possible to toggle this display on and off.
 - On the colored balancing bars, it is now possible to hide the colored diamonds and commit date lines for selected milestones. This can help reduce clutter when many intermediate milestones are present.
 - In the WBS Editor, the Task Details tab now contains a "Data Problems" column. When a coach, planning manager, quality

- manager, or other individual discovers a problem for a particular component or task, they can type a description of the problem in this field. Doing so will display a red highlight behind the component/task, making it easier to bring the item to the attention of another individual.
- Move Up and Move Down buttons are now provided on the toolbar of the WBS Editor Milestones window.
 - The "Task Time" tab of the WBS Editor includes a column called "Task Size." When you create tasks underneath a component, the numbers in this column are automatically inherited from that component. Now, these numbers are editable as well, so you can override the size for a particular task.
 - The WBS Editor's memory footprint has been changed to help improve performance for teams with very large work breakdown structures.
 - Historically, if a Team Dashboard was opened in read-only mode, the read-only flag would also propagate to any WBS Editor windows that were opened. This linkage has been broken. As a result, an individual can now open the Team Dashboard in read-only mode (for example, to view team rollups), and then open the WBS Editor to make changes if problems are noted.
 - During a project launch, teams commonly need to perform high-level capacity planning - for example, to estimate the end-to-end project schedule and to choose the scope for various project iterations. Several changes have been made to facilitate this activity:
 - The WBS balancing panel now includes a "Team" row in addition to the colored bars for each team member. This new team row displays the balanced completion dates for each milestone, making it easier to distribute work across a series of future iterations.
 - In earlier versions of the dashboard, teams would have to create "placeholder" tasks to record time estimates for this future work. Now, rough time estimates (and milestones) can be entered directly on WBS components that have no subtasks, and the vertical black balancing bar will take these time estimates into account.
 - Teams may wish to enter vacation time or other schedule exceptions in the WBS Editor Team Member List months in advance. In the past, this would cause the team earned value charts to extend artificially far into the future (to include the week when the exception was recorded). The earned value charts have been enhanced to avoid this problem.
 - Several changes have been made to improve the usability of the user interface:
 - Historically, clicking the script button has opened the "default" script, form, or report for the currently active task. To view the menu of other available scripts and forms, it was necessary to click the small down-arrow to the right of the script button. In practice, however, opening the script menu is a far more common action. Accordingly, the behavior of the main script button has been changed to open the menu instead of opening the default script. For users who prefer the former behavior, a checkbox is provided in the Tools > Preferences window to revert back.
 - In the Task & Schedule window, it is now possible to choose the set of columns you prefer to see in the Task list. This can reduce clutter and make it easier to focus on the information that is most important to you. In addition, if the table columns are resized or rearranged, these changes will be remembered the next time you open that task list.
 - Several common items on the script menu have been given clearer, simpler names to reduce confusion and to lower the learning curve for people who are beginning their first team project.
 - The Preferences tool now includes an option to enable large fonts throughout the application.
 - Several enhancements have been made to the charts and reports:
 - The Weekly EV Report has always included "Previous" and "Next" links to scroll forward and backward in time. Now, it also includes a link for jumping directly to a specific date.
 - When applying a label filter to the Team Project Rollup Plan Summary, autocomplete support is now provided as you type.
 - The Tasks In Progress chart is now configurable, allowing you to change the red overspent rings to either black or white.
 - When viewing the Kanban chart for a team, you can now filter the list of tasks by assigned individual.
 - Several enhancements have been made to the LOC Counting support:
 - The LOC counter now includes support for counting changes made to files in a subversion repository. The count can include uncommitted changes made to a working copy as well as changes made in past revisions. Multiple past revisions can be listed, and need not be consecutive.
 - The LOC Counter report is often used during the postmortem phase of a PSP project to measure added, deleted, and modified lines of code. Now, individuals can drag data from that report and drop it onto a row in the Size Estimating Template to apply actual size metrics.
 - When creating tasks and workflows in the WBS, it is a best practice to use a PSP Task to represent detailed software development activities. Several improvements have been made to this support:
 - In the WBS, a PSP Task can be followed by Design and Code Inspection tasks to represent the associated peer reviews. When this pattern has been followed, the Task & Schedule "Flat View" will now automatically insert the inspections into the correct order within the phases of the PSP task.

- The dashboard has always allowed individuals to configure multiple collections of "To Date" data that can be used to track different types of work. Now, new options on the Team Project Parameters and Settings page allow individuals to select which "To Date" rollup they would like to use for new PSP tasks.
- Teams sometimes encounter, fix, and log defects that should not technically count against their quality metrics. (For example, they might fix a defect in legacy code that they inherited from some external source.) Individuals can now enter 0 in the "Fix Count" field of the defect dialog to indicate that a defect should not count against quality metrics such as yield or defect density.
- If you drag-and-drop an empty directory onto the Quick Launcher, it will now ask you if you wish to create a new Team or Personal dataset there.
- When saving a data backup file, it has historically been possible to save in either ZIP or PDBK format. Now, a third option is available: "Redacted Process Dashboard Backup." Choosing this format allows you to select various categories of data (for example, names of individuals, projects and tasks) that should be scrambled or removed from the backup. This can be a useful tool to protect privacy or confidentiality.

1.6.3. Features new in version 1.14

- Significant changes were made to the earned value support:
 - It is now possible to customize the set of charts that appear on the earned value report. In addition, clicking on these charts now opens a page displaying a larger chart with helpful instructions on how the chart can be used and interpreted.
 - The Earned Value report now includes a "More Charts" hyperlink. Clicking on this link makes it possible to see all of the EV charts in the web browser. (Previously, many of these charts were only available in the "Chart" dialog of the Task & Schedule window.)
 - New charts have been added to display the Earned Value Trend and the Direct Time Trend.
 - The earned value report now includes a "Kanban View" of the tasks that have been completed recently, the tasks that are in progress, and the tasks that are planned for the near future.
 - By default, the bars on the Gantt charts depict dates from the Forecast column of the task list. Now, a configuration button makes it possible to depict Plan, Replan, or Baseline dates instead.
 - In the past, the Weekly EV report used generic phrases such as "Tasks Completed This Week." This wording was confusing for views of data in the past or the future. These section headings have been altered to be clearer and more descriptive.
 - New topics have been added to the online help for each of the earned value charts. These topics explain how to interpret the data in each chart, and provide helpful analysis tips for people who are new to earned value tracking.
 - A new help topic has been added that explains the difference between Plan, Replan, and Forecast dates. This help topic explains these three calculations in detail, so users can understand and appreciate the differences between the dates that are generated.
- The WBS now has a special "personal editing mode." When the WBS is opened from a personal dashboard instead of the Team Dashboard (and the team leader has not disabled edits by team members), the WBS becomes aware of the individual who opened it, and changes its behavior:
 - Newly created tasks are automatically assigned to the individual who opened the WBS.
 - If the individual makes a change that affects one of their coworkers, the WBS will display a warning and offer to undo the change.
 - The individual is only allowed to edit their own row in the Team Member List.
 - The individual can disable these features if desired by toggling a checkbox in the "Team" menu. (So for example, these new features will not prevent a Planning Manager from making changes to their coworker's tasks.)
- Numerous changes were made to provide better support for the SEI PSP courses:
 - A new Student Profile element has been added to the PSP course assignment sequence, giving students an opportunity to answer questions about their job position, experience, and so on.
 - The Analysis Report exercises now allow the use of the Size Estimating Template and PROBE.
 - "To Date" metrics are no longer reset on the first PSP2 project in a PSP Fundamentals & Advanced course.
 - During the PSP course, PROBE Method D for Size will require students to use their Estimated Proxy Size verbatim; they will not be allowed to edit the number.
 - During the PSP course, the PROBE Wizard will not offer PROBE Method C1 for Size as a selectable option.
 - The final page of the PROBE Wizard has always performed a sanity check on the planned productivity. If it is unrealistic, the wizard prints a warning message and takes the user back to reevaluate their estimates. Unfortunately, that warning message was easy to overlook, resulting in confusion when the PROBE wizard looped back to the earlier pages. The buttons and messages on the sanity check page have been altered to make it more clear that a planning error may be

- present.
- When students are allowed to edit the quality plan and they produce phase times that do not sum up properly, an error message is now displayed on the Project Plan Summary.
- Student data can now be exported to an XML file.
- When capturing defect data, individuals can now:
 - Make a single entry to represent several related defects that were found and fixed simultaneously.
 - Mark a defect as "pending" (i.e. found, but not yet removed).
 - Alter the date associated with the defect entry.
- Several usability enhancements have been made to the WBS Editor and the team project integration features:
 - The Team Project Setup Wizard will now provide suggested default values for the team process, the name of the EV schedule, and the location of the Team Project Network Directory.
 - If an individual enters their initials incorrectly when joining a team project, the "Sync to WBS" operation will detect this error and display a message, helping them to correct it.
 - Actual Size is now displayed in the WBS Editor for each component in the hierarchy.
 - The charts in the Rollup Plan Summary report will now use consistent colors to represent the various phases in the standard metrics collection frameworks. This makes it easier to correlate data between several charts.
 - When a note is attached to the root node of the project in the WBS Editor, that note will now be copied down into the personal plans of each individual. This can be used to record helpful project-specific URLs for team use.
 - In the past, if a node was deleted from the WBS and a new node was created in its place with the same name, the Sync to WBS operation would perform a similar delete/recreate operation. Now, the Sync to WBS attempts to detect this scenario and reuse the existing node instead of deleting/recreating it. (Note: deleting/recreating nodes in the WBS is still discouraged; but now when the mistake occurs, the consequence should be less severe.)
 - In the past, the WBS editor would display a confirmation prompt every time you request to delete a node. Now, the WBS Editor will only display this confirmation prompt when the node to be deleted has actual time associated with it - and the warning message has been altered to describe this new condition.
 - In the past, when a leaf task in the WBS was subdivided, the next "sync to WBS" operation would often result in a "top-down-bottom-up" error in the personal plan of the affected individual. This problem has been corrected.
- Significant changes were made to the installer for the Process Dashboard:
 - The installer can now create application shortcuts for many users on Unix and Linux platforms.
 - On 64-bit Windows, sometimes the installer would not create Process Dashboard shortcuts. This problem has been corrected.
 - If a user has less than 800MB of memory (for example, because they are running in a virtual machine), the Process Dashboard shortcuts would fail to start the application. The installer has been adjusted to create shortcuts that work on systems with limited memory. (Note that the Process Dashboard does not require or use that much memory; the problem was with the shortcut icon itself, not the application.)
- Several changes have been made to improve compatibility with various programs and operating systems:
 - The "Export to Excel" hyperlinks have been tweaked to improve compatibility with a variety of web browsers and with newer versions of Microsoft Office.
 - On some versions of Unix (for example, Solaris and RedHat), the Process Dashboard would sometimes appear to hang at the splash screen. This was occurring when a dialog box was displayed (to prompt for some type of input), because the Unix window manager was improperly stacking the dialog box behind the splash screen. This problem has been corrected.
 - If the dashboard data directory was unreachable (for example, because it was on a network drive that was unavailable), or if other operating-system-specific problems prevented the dashboard from locking the data, the dashboard was incorrectly displaying a message claiming that someone on another computer had locked the data. This error message has been corrected and clarified.
 - The "Import Defects from Code Collaborator" feature was not working against Code Collaborator version 6. This problem has been corrected.
- The "Generic" process template now allows the use of the Size Estimating Template and PROBE. (This functionality will appear for projects created using the Generic process template **after** upgrading to version 1.14.)
- The "C > Tools" menu now includes an "Open Dataset" option, giving all users the ability to open data backup ZIP files.
- When users filter the Time Log Editor to display the current week, they can now easily select which day of the week to use as the starting point.
- In the past, when a user made a change to a data value within the Process Dashboard, that change might take 30 seconds to appear on the Project Plan Summary form in their web browser. Now, these changes will appear in the web browser immediately.
- The LOC counter that is built in to the Process Dashboard will now count lines appropriately even when comment indicators

appear within string literals.

- When all the tasks in a earned value task list share a common path prefix (common for team projects), the Flat View will extract that common prefix, making the display easier to read.
- In the defect log editor, a combo box displays the defect type standard that has been set at each level of the hierarchy. In the past, when a defect type standard had been set at the team project level, nothing was displayed in this combo box for individual, leading people to think that their standard had not taken effect. Now, this combo box will display the name of the defect type standard that was set by the team project, minimizing confusion.

1.6.4. Features new in version 1.13

- Several enhancements have been made in the Work Breakdown Structure Editor:
 - A strikethrough font is now used in the "Assigned To" column to indicate which individuals have completed a multi-person task.
 - "Move Up" and "Move Down" buttons have been added to the toolbar.
 - The Team Member List will now allow you to reorder individual team members using drag-and-drop.
 - If you are starting a new project iteration, you can now use Ctrl-C and Ctrl-V to copy and paste team members from the old team member list to the new team member list.
 - In the past, when individual team members open the WBS from their personal dashboard, the WBS Editor would sometimes unexpectedly open in read-only mode. This problem has been corrected.
 - In the past, the Team Dashboard's Team Project Parameters and Settings page displayed a checkbox allowing the team leader to decide whether team members could edit the WBS. This checkbox has been moved into the Work Breakdown Structure Editor itself, in the "Edit > Preferences" dialog. As a result, if a team leader has locked down the WBS (forbidding edits by team members), they will need to reaffirm this choice in the "Edit > Preferences" dialog of the WBS Editor after upgrading to version 1.13
 - If you insert a workflow underneath a component that has a LOC size estimate, the Code task in that workflow will automatically inherit that size estimate.
 - The selection colors have been modified to produce better readability on Mac OS X.
- Several enhancements have been made in the WBS Common Team Workflows window:
 - It is now possible to define a workflow that distributes time across various phases using percentages, even if your team does not use size estimates or historical productivity rates.
 - The keyboard accessibility of the Common Team Workflows editor has been improved. Keystrokes such as Tab, Enter, Delete, Ctrl-C, Ctrl-V, and Ctrl-X will now assume intuitive, spreadsheet-like behaviors when the focus is not in the first column.
 - The Task Size Units column now supports autocompletion.
- Several enhancements have been made in the Task & Schedule window:
 - When an individual views one of their personal earned value task lists in the Task & Schedule window, they will see a "Notes" column that allows them to view and edit the notes associated with each component or task.
 - When an individual views one of their personal earned value task lists in the Task & Schedule window, the schedule pane (in the lower portion of the window) now contains a "Notes" column. This column can be used to record the reason why a particular week has an unusual number of planned or actual direct task hours.
 - A strikethrough font is now used to indicate tasks that are 100% complete.
 - When you toggle from "Tree View" to "Flat View" and back, the currently selected task will be preserved.
- The Defect Log Editor now provides a feature to Import defects from the system clipboard. This can be used to copy and paste defects from an external source, like an Excel spreadsheet or a web page.
- The "Tools > Preferences" dialog now shows an "Always on Top" option. (This feature will only appear if you are using Java 1.6 or higher.)
- If you pause the timer on the main dashboard toolbar, it will pause the "Fix Time" timer on the currently active defect dialog as well. If you subsequently resume the main dashboard timer, it will resume the defect fix timer as well.
- The dashboard has always allowed you to set an "END" date for a personal earned value schedule. (This feature is typically used to record the date when an individual will unequivocally leave a project team, due to an impending reassignment.) When an individual with an END date is assigned too much work, their personal schedule may project that certain tasks will "never" be completed. Although that information is useful at an individual level, it can result in a frustrating lack of insight at the team level. Now, when one of these schedules is a part of a team schedule, the team rollup schedule will hypothetically rebalance these tasks to the rest of the team, calculate the date when the rest of the team might complete the tasks, and display that completion date in the team's "merged" view. The "Assigned To" column for such a task will indicate that the task has been hypothetically rebalanced

to the team.

1.6.5. Features new in version 1.12

- Since version 1.7, the dashboard has included a team metrics collection framework called "Software Systems Development." To avoid any potential copyright violations, that metrics framework has used a different set of process phases than the TSP^(SM). Beginning with version 1.12, we are happy to announce that the Software Engineering Institute has granted us permission to distribute a TSP-compatible metrics framework. This new framework will reduce the learning curve for TSP teams as they first begin to use the Process Dashboard.
- Process Dashboard installers are now provided to support the assignment sequence of the PSP^(SM) Fundamentals and Advanced courses.
- It is now possible to attach HTTP URLs to the elements in a Common Team Workflow. This can be used to associate organizational process scripts or wiki pages that describe the steps in a workflow. (Note: this feature is only available for team projects that are created using Process Dashboard 1.12 or later.)
- Data from earned value schedules can now be exported to Microsoft Project in the new MS Office XML format.
- The "Status of Team Member Metrics" page has always displayed the date when each team member last exported their metrics. Now, it also displays the date they last performed a "Sync to WBS" operation.
- In the past, the components and tasks in a rolled-up team EV schedule would sometimes not appear in the same order that they appear in the WBS. The reporting logic has been enhanced so the EV schedule will follow the WBS element ordering more closely.
- Charts have been added to the PSP project plan summary forms to display time, defects, and PQI.
- In the WBS Editor Team Member List, the column headers now display dates in a locale-sensitive manner for non-US teams.
- Teams can optionally choose to have the WBS Editor prompt for read-only mode on startup. (This option can be controlled on a project-by-project basis.)
- In the WBS Editor and in the EV reports, when a percent complete number is greater than 99.5% but less than 100%, it will be rounded down to 99%. This helps to avoid confusion, since 100% will now only appear for tasks that are truly complete.
- The EV Report will now display a warning message if tasks have been marked complete in the future, or if time has been logged in the future. This can help in the troubleshooting of problems caused by data entry typos.
- On the Weekly EV Report, the "Tasks Completed Last Week" table now displays a total row.
- Some international dashboard users have seen problems with corruption of accented characters in the Team Dashboard. A new feature is provided in the Preferences dialog to help those teams.
- In the personal dashboard, the Team Project Tools and Settings page has previously included a checkbox to "Include all WBS components and documents in sync operations." This checkbox represented obsolete functionality, and now only causes problems when individuals enable it out of curiosity. To avoid those problems, the feature has been removed.
- Teams that have upgraded to a recent version of the Code Collaborator server were finding that defect types were no longer being transferred by the "Import from Code Collaborator" feature. This problem has been corrected.
- A bug was affecting the date-based filtering of entries in the Time Log (for example, when you request to view time log entries for "Today"). This bug has been corrected.
- The Perforce LOC counter can now count LOC in integration changelists. This functionality is designed to support the workflow where an individual makes changes in a personal sandbox, then integrates those changes back to the main line of development. In that case, the individual can point the LOC counter at the changelist where the files were integrated back to the mainline, and receive a LOC count of the effective changes.
- Installers are now available in EXE format for use on Windows systems. These EXEs can simplify the installation process for users who previously encountered problems with filesystem permissions in Windows Vista.

1.6.6. Features new in version 1.11

- URLs can now be attached to any component or task in the dashboard. Just use the WBS Editor or the dashboard to edit the free-text note associated with a particular item, and enter a URL within the body of the note. The dashboard will discover the embedded URLs and add them to the script menu for easy, one-click access.
- A new Preferences dialog is available on the "C > Tools" menu, providing easy access to the most common configuration settings.
- Several enhancements were made to the usability of the WBS Editor:
 - Pressing the Enter key in the WBS Editor typically inserts a new task. A new button is now provided to toggle this behavior on and off.
 - The WBS Editor now uses a strikethrough font to highlight tasks that are 100% complete.

- A "New Tab" button is now displayed, providing one-click access to the functionality on the WBS Editor "Tabs" menu.
- Significant enhancements have been made to the PSP materials in the Process Dashboard:
 - The PSP scripts and forms in the Process Dashboard have been updated to match the materials in the 2005 book, *A Self-Improvement Process for Software Engineers*. (Up to now, they have been based upon the 1995 book, *A Discipline for Software Engineering*.) With this change, the scripts, forms, and wizards now match the materials and terminology used in the most recent PSP courses from the SEI. (Note: the PSP materials are now released as a separate download from the Process Dashboard.)
 - On the Size Estimating Template, the estimated size of an "added part" will be computed automatically after an engineer estimates its type, relative size, and number of items. Engineers can create their own size-per-item lookup tables in support of this functionality.
 - On the Size Estimating Template, an interactive chart now appears to help engineers visualize the relationship between estimated base, added, deleted, and modified size. Engineers can drag handles on a slider to visually estimate base additions, modifications, and deletions as a percentage of base program size.
 - The PSP Project Plan Summary forms are now customizable, in support of personal process improvement goals.
- Significant changes have been made in support of the PSP for Engineers courses:
 - A new "Grading Helper" has been created for PSP Instructors. After installing a special "PSP Instructor" add-on, this grading helper will appear in the script menu for PSP course assignments. The grading helper automates many of the data analysis tasks and consistency checks that are part of the grading process. Instructors can customize the form to match their grading preferences.
 - The dashboard traditionally baselines "Planned" and "To Date" project data at specific points during a project. Although this strategy is very helpful for the planning and tracking of real-world project work, it becomes an impediment to the quick-turnaround learning objectives of the PSP course, because the "frozen" values prevent students from correcting errors in earlier assignments. To address this problem, this "freezing" mechanism has now been disabled for the assignments in the PSP course.
 - It is now much easier for students to create the quality plan for their first PSP2 assignment.
 - Error messages now appear on the Project Plan Summary form to warn engineers about common mistakes. For example, if students forget to enter the actual size of Program 1, an error message will appear on their Program 2 Project Plan Summary form.
- When you first create a team project, the Team Project Setup Wizard prompts you for the name of a directory where team data should be stored. Later in the project, you may discover that you need to move these files to a different location on the network. A new feature makes this possible.
- The dashboard has always included advanced earned value features to support iterative project planning. Unfortunately, some teams would inadvertently trigger these advanced features by choosing a start date in the future (for example, the week *after* their project launch). To avoid this confusion, the iterative planning feature must now be explicitly enabled in the "Tools > Schedule Options" dialog of the Task & Schedule window.
- Geographically diverse teams - such as teams split between India and the United States - would sometimes see inconsistent data in the earned value reports. New time-zone awareness logic has been added to address these problems.
- The defect log now includes an "Import from Code Collaborator" button. Teams that use Collaborator (<http://smartbear.com/>) can take advantage of this new import feature to reduce double-entry of defect data.

1.6.7. Features new in version 1.10

- The Process Dashboard is now released under version 3 of the GNU General Public License.
- For new team projects, created after the installation of this update, a new task structure will be used:
 - Previously, the process phase of a task (e.g. Planning, Code, Test, etc) was recorded via a final "stub" that appeared at the end of each task in the hierarchy. Now, these stubs are no longer created. This change reduces clutter, makes it easier to subdivide existing tasks, and reduces the memory usage of the Process Dashboard.
 - In a new-style team project, individuals will have a project plan summary report similar to the one on the team side: it can be edited to include custom charts, reports, and tables of data.
 - **Important:** Data from new-style and old-style team projects **can** still be rolled up together into master projects.
- For a new-style team project that is underway, new columns will appear in the Work Breakdown Structure Editor to display actual metrics:
 - Actual time spent on each task
 - Actual completion date
 - Percent complete

- Percent spent

Among other things, this makes it possible to see which tasks have been completed, which are in progress, and which have not yet started.

- When changes are made in the WBS, individuals use the "Sync to WBS" operation to copy those changes into their personal plan. In a [new-style](#) team project, the synchronization is now bidirectional. When individuals make changes to the following data in their personal plan, the changes will be propagated back into the WBS automatically:

- Changes to the planned time for tasks
- Changes to the planned hours in the schedule
- Changes to task notes/comments
- Changes to the phase type of a task (e.g. Planning, Code, etc.)

- The colored bars in the WBS Editor are a useful tool for balancing work between team members. For a [new-style](#) team project that is underway, these bars can now be toggled between two modes:

- Show End-to-End Plan: This mode takes all tasks into account. The dates calculated for each team member will correspond to the "Planned Completion Date" for that individual in their personal earned value schedule.
- Show Remaining Work: This mode subtracts out tasks that have already been completed, as well as effort spent on in-progress work. The dates calculated for each team member will correspond to the "Replanned Completion Date" for that individual in their personal earned value schedule.

The new "Remaining Work" mode is a powerful way to rebalance work during a relaunch. Teams can also use it throughout a project iteration to continually rebalance work.

- The Process Dashboard now supports baselines for earned value data. You can save a baseline for any earned value schedule, then view baseline metrics in the task list and EV charts.

- The WBS Editor now includes support for project milestones:

- It is possible to define a list of project milestones, and optionally enter commit dates
- Each component and/or task in the WBS can be assigned to a particular milestone
- Marks appear on the colored balancing bars to indicate when each individual could potentially finish their work for each milestone
- The colored balancing bars can be configured to balance work for the entire project or just for the next milestone(s)
- Future milestones can be marked as "deferred," and the sync operation will not copy the associated tasks into team member plans.

- A new icon on the main toolbar makes it easy to attach free-text notes to any component or task in the Process Dashboard. On a team project, these notes are shared by the team, and accessible in the WBS Editor.

- Play and pause now appear as separate toggle buttons.

- A new "timing reminder" feature is available. This feature can periodically display pop-up alerts, reminding the user to start or stop their timer or to change the active task. (*This feature is not enabled by default, but can be turned on by individuals who wish to use it. To enable this feature, right-click on the tray icon and choose "Reminder > Enable Reminder."*)

- The WBS Editor now allows you to enter detailed schedules for each team member, to include start dates, end dates, and planned time exceptions on specific weeks. This information is bidirectionally synchronized with each individual's personal earned value schedule.

- Previously, the WBS Editor would not allow you to assign a task to an individual until a time estimate had been entered for the task. Now, it is possible to assign a task to an individual even while the task's time estimate is still zero. Agile project teams may use this feature to support on-the-fly cost estimation.

- The Team Dashboard Users Manual is now included in the dashboard's searchable online help.

- When starting a new project or iteration, you can now bring forward the data from historical earned value schedules. As a result, the dashboard can calculate forecast dates and prediction ranges during a team project launch, even before any actual data has been collected.

- The weekly earned value report now includes a "Tasks In Progress" section. This section displays tasks that have been started but not yet completed, making it easier to spot problems such as overspent tasks.

- Historically, "PSP Tasks" were required if a team wanted to capture planned and actual LOC for their project. Now, LOC can be captured for "code" tasks as well.

- Installation and operation of the dashboard on Mac OS X is greatly improved.

- On Unix/Linux systems, the default web browser has been changed from Netscape to Firefox.

1.6.8. Features new in version 1.9

- The Process Dashboard now displays an icon in the system tray:

- Users can click on the icon to start/stop the timer, and tell from a glance at the icon whether the timer is running.
- The tooltip on the icon displays the active task. Ctrl-click on the icon allows the user to select a new active task.
- The popup menu on the tray icon provides quick access to many commonly used dashboard features.
- In the Team Dashboard, you can now expand the weekly earned value report to show data for each individual team member. This resulting report can be very useful in weekly team meetings.
- The play/pause button now displays new icons that make it easier to determine whether the timer is running or stopped.
- Tasks labels (assigned in the WBS Editor) are now displayed for each task when you open an earned value plan in the Task & Schedule window or view a report in your browser.
- Warning icons now appear when a task is planned to be completed before its dependencies.
- Since version 1.7.1b-9, team members have been able to open the Work Breakdown Structure directly from their personal dashboard. A preference on the Team Project Parameters and Settings page controls whether individuals should be allowed to make changes to the WBS. Previously, this setting defaulted to "read-only," but with this release, the default is to **allow edits**. Teams that wish to keep the WBS read-only will need to adjust the value of this setting after upgrading.
- A new project-specific parameter allows teams to disable (permanently or temporarily) the alerts telling individuals that they need to sync to the work breakdown structure.
- Several common operations have been optimized, and now require much less time. For example, the dashboard may start in half the time, and export data in one-fifth the time.

1.6.9. Features new in version 1.8

- Several enhancements to the team functionality:
 - Individuals can now open the team Work Breakdown Structure directly from their personal dashboard, via a new option on the script menu.
 - If an individual needs to perform a "Synchronize to WBS" operation, they will receive an alert shortly after starting the dashboard.
 - It is now possible to view the rolled up time log for team and master projects.
 - The Team Project Parameters and Settings page now allows each team to select the set of phases they wish to appear in team forms, charts, and reports.
 - It is now possible to define a defect type standard at the team level for shared use by the members of a team project.
- Several enhancements to earned value support:
 - Forecast dates are calculated for every task in the EV schedule.
 - A new "Replan" column has been added to the earned value task list, displaying the planned dates that would result if you were to "replan from today."
 - In addition to the "Chart" button, the Task & Schedule window now contains a "Filtered Chart" button. This button will display a filtered view of the earned value charts and statistics, for the selected portion of the hierarchical task list.
 - On the "Combined" earned value chart, the "Actual Time" line has been replaced with an "Actual Cost" line. With this change, the three lines on that chart are now displaying traditional BCWS, BCWP, and ACWP metrics.
 - Drag-and-drop can be used to reorder tasks in the Flat View of an earned value schedule.
- A new option has been added to the "C > Tools" menu, allowing the user to save a backup of their data directory to a file.
- A new Quick Launcher tool is provided for quickly opening a dashboard instance without making a shortcut first.
- A new visual indicator has been added to the main Process Dashboard toolbar, displaying the percent spent for the current task.
- When a user manually enters interrupt time in their time log, the delta time is now adjusted accordingly.

1.6.10. Features new in version 1.7

- The Process Dashboard now contains extensive, powerful functionality to support project teams, including:
 - Support for team projects (to roll-up metrics and schedule data from several individuals) and master projects (to roll-up metrics and schedule data from several teams)
 - Streamlined project planning tools enable rapid creation of balanced, top-down team project plans
 - Powerful reporting tools allow the creation of custom forms, charts, and reports that analyze project metrics and earned value at the team level
 - Tracking of task dependencies assists with coordination of work within and across project teams
- The Process Dashboard now contains extensive support for internationalization. A Localization Tool is provided to help users translate dashboard messages into their native language, and share those translations with other users.
- Memory usage and startup time have been reduced dramatically for individuals with many historical projects.

- The dashboard contains several enhancements to earned value functionality:
 - A new task navigation option makes it easier to work through the tasks in an earned value plan.
 - The plan and forecast lines on the earned value charts can be toggled off.
 - The earned value report can display your task list hierarchically, and display rolled-up calculations such as "percent complete" at various levels.
 - Export a GANTT chart of an earned value plan to MS Project
 - Export an HTML snapshot of an earned value plan, for status reporting and collaboration
- The Data Analysis center has been enhanced to support custom processes. Users who define a custom process will see customized versions of Report 3, 4, and 5 for their process.
- The Process Dashboard now integrates with a wide variety of web browsers on all platforms. In particular, Unix/Linux users no longer need to use an older version of Netscape to view dashboard forms.
- The dashboard can be opened in read-only mode, making it possible for a mentor/coach to look at dashboard data without worrying about accidentally changing data.
- It is now very safe to use a networked directory for storage of dashboard data; the logic that saves changes will detect and recover from sporadic network outages.
- Time values are displayed on dashboard forms in hours and minutes.

1.6.11. Features new in version 1.6 (released September 1, 2003)

- The dashboard contains several innovative enhancements to earned value functionality:
 - The earned value tool in the dashboard can now calculate estimated prediction intervals for forecast completion date and forecast cost. These intervals compare your planned progress to your actual progress using numerical methods and approximate pivotal statistics, computing a range of dates within which your project is likely to finish. The dashboard can calculate these ranges both for individual earned value schedules and for rolled-up team schedules.
 - Traditional PSP earned value planning addresses only "direct time," omitting time spent on recurring activities (such as weekly meetings) and ongoing activities (such as communicating with customers and management, or participating on committees). Unfortunately, "direct time" is an unfamiliar metric for many organizations, which are accustomed to tracking "project time" instead. As a result, senior managers in some organizations take issue with earned value plans that reflect only 15-20 direct hours per week.

To help development teams avoid this potential misunderstanding, the dashboard now provides an innovative new feature: "level of effort" tasks can be added to earned value task lists. Although these tasks don't affect the earned value calculations, they do allow you to create a schedule that accounts for all of your "project time" - and this can help when you present plans to management for approval. To add "level of effort" tasks to an earned value plan, just enter percentages for these tasks in the "planned time" column instead of durations. (Planned direct time for the periods in the schedule is then calculated automatically by reducing the planned total time by the appropriate percentage.) To help you enhance the accuracy of future estimates, "actual time" for "level of effort" tasks is also displayed as a percentage.

 - Earned value task lists in the dashboard have always been hierarchically organized. When the list of tasks for a particular project differs from the structure of your dashboard hierarchy, new features in the earned value tool allow you to "prune" and reorder tasks to better describe the work that needs to be done.
 - At times in the middle of a project, you may decide that you wish to "re-plan" the remaining tasks and start with a fresh earned value schedule. The dashboard now makes this simple: just change the starting date on your earned value schedule to the current date. All work performed before the starting date of the schedule will be factored out of earned value calculations automatically.
- The "Data Analysis" center now allows you to generate custom charts and reports, simply by selecting options from drop-down lists. The data in these custom charts and reports can also be exported to Excel for further custom analysis.
- The time log now includes a column for the entry of optional comments.
- It is now possible to create and edit custom defect type standards.

1.6.12. Features new in version 1.5 (released October 3, 2002)

- The dashboard is now compatible with version 1.4 and higher of the Java Runtime Environment.
- The dashboard now contains support for Netscape 6 and higher on Windows platforms, and is also compatible with the Sun Java Plug-in in Internet Explorer.
- The task and schedule planning tool now contains built-in support for collaboration! You can share your schedules with your co-workers, your manager, and/or your customer, and keep them up-to-date on your progress. If you are working on a project with

- other people, you can roll up your individual schedules to create an earned value schedule for the project.
- The task and schedule planning tool helps you to avoid several common planning mistakes.
- The hierarchy editor now includes a toolbar to facilitate the editing of your work breakdown structure.
- Numerous user interface enhancements make the hierarchy editor, the time log editor, and the defect log editor easier to use.

1.6.13. Features new in version 1.4 (released February 22, 2002)

- Creating custom processes has never been easier - now you can create your own process definitions for use with the dashboard by writing a simple XML file.
- Task and schedule features help you to track your work using earned value, and calculate forecasts for cost and schedule.
- New "data rollup" templates allow you to roll-up data from multiple projects. This feature makes it possible to:
 - Roll up psp data to the project level.
 - Create multiple subsets of "To Date" data (for example, representing different programming languages). The "To Date" data mechanism now allows you to select, for each project, which subset should be used for automatic calculations.
- External documents are now integrated into the process scripts - click on hyperlinks for documents like the PIP form, the Issue Tracking Log, etc. and the appropriate document will be opened for the project you're working on.
- A new "PROBE Wizard" simplifies the PROBE process, and provides advice on which PROBE method is most appropriate.
- You can now export forms (like the Project Plan Summary or the Size Estimating Template) to Excel for additional, custom analysis. If you want to show your data to others, or save copies of forms for project documentation, you can also export these forms to static HTML documents.
- You can now "unlock" the read-only data on forms if you want to override any of the calculations that the dashboard automatically performs for you.
- The hierarchy editor now allows you to move nodes/projects. Cut and paste operations make it possible to reorganize your work breakdown structure.
- The drop-down list on the script button now displays choices like "Project Plan Summary" instead of listing the names of process phases. In addition, a "script browser" makes it easy to open a script/form for any project in your hierarchy without needing to navigate to that project first.
- Checkmarks in the hierarchy menus make it easy to see which phases / projects have been marked complete.
- The "pspdif" program (which calculates added / deleted / modified physical LOC counts) is now incorporated into the dashboard.
- A major overhaul of the internal data calculation engine allows the dashboard to recalculate data up to 10 times faster.
- If you insert "Design Review" or "Code Review" phases into a PSP0 - PSP1.1 project, these phases will automatically appear on the project plan summary and in the process scripts.
- The text "ERROR" in the dashboard forms has been changed to "#VALUE!" or "#DIV/0!" to behave more like Excel.
- An entry to enter the Total LOC was added to the Size Estimating Template screen. This only makes sense since you have to enter nearly all other actual LOC measures on this form.
- Help was implemented via a sub menu. Other items on the Help sub menu allow you to submit bug reports online and navigate to the user online help forum.

1.6.14. Features new in version 1.3.1 (released August 14, 2001)

- Version 1.3.1 was released to address a high-priority bug present in version 1.3. No significant new features were added to 1.3 for this release.

1.6.15. Features new in version 1.3 (released July 16, 2001)

- The dashboard now includes greatly enhanced online help! Improved help has been the #1 requested feature by you, the user community. We are therefore pleased to offer indexed, searchable online help with ten times the content of version 1.2.
- The dashboard now includes a "Generic" process. When you use the generic process to perform a task, it allows you to:
 - Choose any list of phases, with custom names
 - Decide whether or not it makes sense to track defects for the task
 - Decide whether or not it makes sense to measure size for the task, and choose a custom unit of size measurement.

The dashboard then provides a customized Project Plan Summary Form for your custom process.
- The dashboard now contains a mechanism for freezing baselined data. When you mark the planning phase of a project complete, all the data in the "Plan" column of the Project Plan Summary form will freeze and stop recalculating. Similarly, when you mark the project complete, all the data in the "To Date" column will freeze and stop recalculating. This makes it possible to view the report

again in the future without having the numbers changing in the report.

- The dashboard can now notify you when updates are released, either to the dashboard itself or to the add-on process sets you have installed.
- The PROBE tool now contains a drop-down box for quickly exploring PROBE methods A, B, and C.
- The Play/Pause button now has a drop-down menu for quickly jumping to a task where you have recently logged time. If you often bounce back and forth between tasks for different projects, this feature makes your life easier.
- Numerous bug fixes make the Time Log Editor more robust.
- The Time Log Editor now makes it easy to quickly filter your view of the time log to "Today", "This Week", or "This Month". Forward and backward buttons let you scroll to look at different time periods.
- The play/pause button and the hierarchy menus now provide audible feedback to help you keep track of whether the dashboard is timing or paused.
- The previous version of the dashboard would only write entries to the time log when you changed phases; now, the time log is updated continually. The data in your time log and your project plan summary forms is never more than a minute old.
- When you find yourself fixing several defects at once (which can happen often due to "fix defects" and "related defects"), the dashboard provides better assistance. It automatically ensures that only one defect timer is running at a time, and maintains a stack of "interrupted" defects.
- The Defect Log Editor now allows defects to be deleted. It also supports quick filtering to see the defects which were injected or removed in a given phase.
- Timer icons have been placed on the HTML forms to allow navigation among the phases of development without having to reach all the way up to the hierarchy menus.
- A method was added that will allow driving of the dashboard by another program. This will allow IDE developers to add support for the dashboard from within their programming environments.

1.6.16. Features new in version 1.2 (released April 20, 2001)

- The dashboard now includes support for ALL the standard PSP processes (PSP0, PSP0.1, PSP1.0, PSP1.1, PSP2.0, PSP2.1, and PSP3.0). All of the online PSP process scripts and forms have been edited to conform as closely as possible to the originals in the PSP book.
- The dashboard now displays charts and reports! Choose the "Data Analysis" option on the "C" menu, and dozens of standard charts and reports are available (including ALL the tables required by reports R3, R4, and R5, and ALL the charts currently drawn by the SEI's "Student Workbook" Excel spreadsheet). With a single click, export data to Excel for further analysis.
- The PROBE tool is now simpler to use. Previously, it would allow you to explore the correlation between any two data items; unfortunately, this made it cumbersome for everyday use. Now, by default, it only displays the data items traditionally used by the PSP PROBE methodology.
- The PSP0 - PSP1.1 processes now support optional "Design Review" and "Code Review" phases.
- Errors in several calculations have been corrected. Also, the "To Date" data calculations have been completely revamped and are now working properly.
- Data calculation efficiency has been improved.
- With the addition of the standard PSP processes, the SCR/DCR processes are no longer shipped with the dashboard. If you are a current dashboard user, and you have been using the SCR/DCR processes, your existing SCR/DCR projects WILL still work. If you wish to continue creating projects that use the SCR/DCR processes in the future, please download the SCR/DCR process separately.

1.6.17. Features new in version 1.1 (released February 9, 2001)

- Version 1.1 contained enhancements that made the tool significantly easier to install.

1.7. License Information

Personal Process Dashboard, copyright © 1998-2013 Tuma Solutions, LLC. The Personal Process Dashboard comes with ABSOLUTELY NO WARRANTY; this is free software, and you are welcome to redistribute it under certain conditions; [click here](#) for details.

The Process Dashboard includes software contributions from the following individuals:

- David Tuma
- Ken Raisor
- Rick Reynolds
- Jonathan Tellier
- Ali El Gamal
- Mark Sebern
- Audrey Bartlett
- Max Agapov
- Kai Johnson

This program also contains:

- Java software libraries, © 1995-2012 [Apache Software Foundation](#)
- Spring Framework libraries, © 2001-2012 [SpringSource](#), a division of VMware, Inc.
- Hibernate ORM libraries, © 2001-2012 [Hibernate development team](#)
- JFreeChart charting software © 2000-2003 [David Gilbert, Simba Management Limited](#)
- Miscellaneous code from the Java Tutorial, © 1997-2003 [Sun Microsystems](#)
- Colt statistics library, © 1999 [CERN - European Organization for Nuclear Research](#).
- The prototype JavaScript library, © 2005 [Sam Stephenson](#), and the Scriptaculous JavaScript library, © 2005 [Thomas Fuchs](#).
- The overLIB tooltip library © 1998-2004 [Erik Bosrup](#).
- Additional code copyright © 2006 [Kai Toedter](#); © 2007-2008 [Anthony Garrett](#); © 2006-2007 [Ian McDonagh](#); © 2000 [Justin P. McCarthy](#); © 2002-2005 [Stephen Kelvin Friedrich](#); © 1999 [Fredrik Ehnbom](#); © 1996 [Santeri Paavolainen](#); © 2003-2006 [Greg Luck](#); © 2001-2005 [MetaStuff, Ltd.](#); © 2012 [France Télécom](#); © 2012 [Terence Parr and Sam Harwell](#); © 2004 Robert Harder.
- Additional translations [copyright](#) © 2003 Chihiro Nagata; © 2005 Su Nan; © 2007 Jingkai Su; © 2005 Abraham Bermejo; © 2007 Andres Cifuentes; © 2011 Andres Amaya Diaz; © 2008 Edgar Fernandez; © 2011 Carlos Payan; © 2011 David García Polo; © 2012 Manuel López Sánchez Pineda; © 2012 Pablo Gonzalez; © 2013 Jose Alberto Carrera; © 2013 Hans Peyrot; © 2008 Luciano Rodrigues; © 2007 Robson Alecio Somera; © 2007-2008 Walmir Antunes da Silva; © 2009 Afonso Goncalves; © 2009 Luis Oliveira; © 2011 Lukas; © 2008 Dmitri Pissarenko; © 2009-2013 Pascal Vollmer.

You may have installed optional modules that provide enhanced charting and reporting capabilities and relational database support. These modules are © 2008-2013 [Tuma Solutions, LLC](#).

This software contains unmodified binary redistributions for H2 database engine (<http://www.h2database.com/>), which is dual licensed and available under a modified version of the MPL 1.1 (Mozilla Public License) or under the (unmodified) EPL 1.0 (Eclipse Public License). An original copy of the license agreement can be found at: <http://www.h2database.com/html/license.html>

1.7.1. SEI/CMU Materials

Personal Software Process^(SM), PSP^(SM), Team Software Process^(SM), and TSP^(SM) are service marks of Carnegie Mellon University. **The open source team that writes the Process Dashboard is not affiliated with Carnegie Mellon University.**

You have installed an optional module that contains PSP^(SM) scripts, forms, and other materials. [Special permission](#) to use "PSP Materials.doc" copyright © 2006 by Carnegie Mellon University, in this module is granted by the Software Engineering Institute. The use of these materials is subject to [license](#).

You may have installed an optional module that contains TSP^(SM) forms and other materials. Reference to and use of Carnegie Mellon University's Team Software Process^(SM) materials (including "Using TSP to Improve Performance" by Dan Burton, Copyright 2008

Carnegie Mellon University, and "The Team Software Process^(SM) (TSP^(SM))" by Watts S. Humphrey, CMU/SEI-2000-TR-023, Copyright 2000 Carnegie Mellon University) is done under special permission from the Software Engineering Institute.

If you wish to use any of Carnegie Mellon University's materials contained in these modules you must obtain specific written permission from the Software Engineering Institute by contacting them at permission@sei.cmu.edu.

Special permission to create derivative work based on charts contained in "The Software Quality Profile" by Watts Humphrey, copyright 1997-2008 by Carnegie Mellon University in the Process Dashboard is granted by the Software Engineering Institute.

ANY CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL CONTAINED HEREIN IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

Neither Carnegie Mellon University nor the Software Engineering Institute have reviewed or endorsed the Process Dashboard.

1.7.2. GNU General Public License

Version 3, 29 June 2007

Copyright © 2007 Free Software Foundation, Inc. <<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we

stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding

Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".
- c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.
- d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

- a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.
- b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.
- c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.
- d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.
- e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates

the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- d) Limiting the use for publicity purposes of names of licensors or authors of the material; or
- e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
- f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered

work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>
```

```
This program is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.
```

```
This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.
```

```
You should have received a copy of the GNU General Public License
along with this program. If not, see <http://www.gnu.org/licenses/>.
```

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

```
<program> Copyright (C) <year> <name of author>

This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands 'show w' and 'show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <<http://www.gnu.org/licenses/>>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <<http://www.gnu.org/philosophy/why-not-lgpl.html>>.

Additional Permissions

The following additional permission is granted, under the terms of [section 7](#) above:

If you modify this Program, or any covered work, by linking or combining it with modules that have been produced, distributed or approved by Tuma Solutions, LLC (regardless of the license terms of those modules), the licensors of this Program grant you additional permission to convey the resulting work.

1.7.3. GNU Lesser General Public License

Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts
as the successor of the GNU Library Public License, version 2, hence
the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- **a)** The modified work must itself be a software library.
- **b)** You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.
- **c)** You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

- **d)** If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

- **a)** Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)
- **b)** Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.
- **c)** Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.
- **d)** If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.
- **e)** Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

- **a)** Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.
- **b)** Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

*one line to give the library's name and an idea of what it does.
Copyright (C) year name of author*

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library `Frob' (a library for tweaking knobs) written by James Random Hacker.

*signature of Ty Coon, 1 April 1990
Ty Coon, President of Vice*

That's all there is to it!

1.7.4. Apache License, Version 2.0

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- a. You must give any other recipients of the Work or Derivative Works a copy of this License; and
- b. You must cause any modified files to carry prominent notices stating that You changed the files; and
- c. You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- d. If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.7.5. MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.7.6. Berkeley Software Distribution (BSD) License

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.7.7. Java Tutorial Code License

Copyright © 2002 Sun Microsystems, Inc. All Rights Reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistribution of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistribution in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of Sun Microsystems, Inc. or the names of contributors may be used to endorse or promote products derived from this software without specific prior written permission.

This software is provided "AS IS," without a warranty of any kind. ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE HEREBY EXCLUDED. SUN MICROSYSTEMS, INC. ("SUN") AND ITS LICENSORS SHALL NOT BE LIABLE FOR ANY DAMAGES SUFFERED BY LICENSEE AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THIS SOFTWARE OR ITS DERIVATIVES. IN NO EVENT WILL SUN OR ITS LICENSORS BE LIABLE FOR ANY LOST REVENUE, PROFIT OR DATA, OR FOR DIRECT, INDIRECT, SPECIAL, CONSEQUENTIAL, INCIDENTAL OR PUNITIVE DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF THE USE OF OR INABILITY TO USE THIS SOFTWARE, EVEN IF SUN HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

You acknowledge that this software is not designed, licensed or intended for use in the design, construction, operation or maintenance of any nuclear facility.

1.7.8. DOM4J License

Redistribution and use of this software and associated documentation ("Software"), with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain copyright statements and notices. Redistributions must also contain a copy of this document.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- The name "DOM4J" must not be used to endorse or promote products derived from this Software without prior written permission of MetaStuff, Ltd. For written permission, please contact dom4j-info@metastuff.com.
- Products derived from this Software may not be called "DOM4J" nor may "DOM4J" appear in their names without prior written permission of MetaStuff, Ltd. DOM4J is a registered trademark of MetaStuff, Ltd.
- Due credit should be given to the DOM4J Project - <http://www.dom4j.org>

THIS SOFTWARE IS PROVIDED BY METASTUFF, LTD. AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL METASTUFF, LTD. OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.7.9. Colt License

Permission to use, copy, modify, distribute and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation. CERN makes no representations about the suitability of this software for any purpose. It is provided "as is" without expressed or implied warranty.

1.7.10. overLIB License

Preamble

The intent of this document is to state the conditions under which a Package may be copied, such that the Copyright Holder maintains some semblance of artistic control over the development of the package, while giving the users of the package the right to use and distribute the Package in a more-or-less customary fashion, plus the right to make reasonable modifications.

Definitions:

- "Package" refers to the collection of files distributed by the Copyright Holder, and derivatives of that collection of files created through textual modification.
- "Standard Version" refers to such a Package if it has not been modified, or has been modified in accordance with the wishes of the Copyright Holder.
- "Copyright Holder" is whoever is named in the copyright or copyrights for the package.
- "You" is you, if you're thinking about copying or distributing this Package.
- "Reasonable copying fee" is whatever you can justify on the basis of media cost, duplication charges, time of people involved, and so on. (You will not be required to justify it to the Copyright Holder, but only to the computing community at large as a market that must bear the fee.)
- "Freely Available" means that no fee is charged for the item itself, though there may be fees involved in handling the item. It also

means that recipients of the item may redistribute it under the same conditions they received it.

Terms:

1. You may make and give away verbatim copies of the source form of the Standard Version of this Package without restriction, provided that you duplicate all of the original copyright notices and associated disclaimers.
2. You may apply bug fixes, portability fixes and other modifications derived from the Public Domain or from the Copyright Holder. A Package modified in such a way shall still be considered the Standard Version.
3. You may otherwise modify your copy of this Package in any way, provided that you insert a prominent notice in each changed file stating how and when you changed that file, and provided that you do at least ONE of the following:
 1. place your modifications in the Public Domain or otherwise make them Freely Available, such as by posting said modifications to Usenet or an equivalent medium, or placing the modifications on a major archive site such as [ftp.uu.net](ftp://ftp.uu.net), or by allowing the Copyright Holder to include your modifications in the Standard Version of the Package.
 2. use the modified Package only within your corporation or organization.
 3. rename any non-standard executables so the names do not conflict with standard executables, which must also be provided, and provide a separate manual page for each non-standard executable that clearly documents how it differs from the Standard Version.
 4. make other distribution arrangements with the Copyright Holder.
4. You may distribute the programs of this Package in object code or executable form, provided that you do at least ONE of the following:
 1. distribute a Standard Version of the executables and library files, together with instructions (in the manual page or equivalent) on where to get the Standard Version.
 2. accompany the distribution with the machine-readable source of the Package with your modifications.
 3. accompany any non-standard executables with their corresponding Standard Version executables, giving the non-standard executables non-standard names, and clearly documenting the differences in manual pages (or equivalent), together with instructions on where to get the Standard Version.
 4. make other distribution arrangements with the Copyright Holder.
5. You may charge a reasonable copying fee for any distribution of this Package. You may charge any fee you choose for support of this Package. You may not charge a fee for this Package itself. However, you may distribute this Package in aggregate with other (possibly commercial) programs as part of a larger (possibly commercial) software distribution provided that you do not advertise this Package as a product of your own.
6. The scripts and library files supplied as input to or produced as output from the programs of this Package do not automatically fall under the copyright of this Package, but belong to whomever generated them, and may be sold commercially, and may be aggregated with this Package.
7. C or perl subroutines supplied by you and linked into this Package shall not be considered part of this Package.
8. The name of the Copyright Holder may not be used to endorse or promote products derived from this software without specific prior written permission.
9. THIS PACKAGE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

1.7.11. License Agreement for Extensions from Tuma Solutions

THIS IS A LEGAL AGREEMENT ("Agreement") BETWEEN YOU AND TUMA SOLUTIONS, LLC ("TS, LLC"). PLEASE READ THIS SOFTWARE LICENSE AGREEMENT CAREFULLY BEFORE INSTALLING OR USING THE SOFTWARE. BY INSTALLING AND/OR USING THIS SOFTWARE, YOU ARE CONSENTING TO BE BOUND BY THIS AGREEMENT. IF YOU DO NOT AGREE TO ALL OF THE TERMS OF THIS AGREEMENT, YOU MUST UNINSTALL AND REFRAIN FROM USING THIS SOFTWARE.

1. TS, LLC and/or its affiliates own all files included in this software (this "Software"). This Software is protected by United States copyright laws and international treaty provisions. You may use this Software only as stated in this Agreement.
2. Permitted Uses: Under this Agreement you may install multiple copies of the software for use as a component extension of the Process Dashboard software application. YOU MAY NOT: (1) use, copy, modify, merge, and/or transfer copies of this Software or any component thereof except as provided in this Agreement; (2) reverse engineer, reverse assemble or reverse compile this Software or any

component thereof; (3) sublicense, rent, lease, sell, or assign this Software; (4) distribute this software for profit or fee in excess of duplication or transmission costs; or (5) incorporate this Software or any component thereof as part or all of another product without prior written permission from TS, LLC.

3. Under this Agreement, you may be given a product key to register this Software by TS, LLS or an authorized affiliate. Product keys are uniquely assigned to an individual, a team, or an organization. YOU MAY NOT disclose or transmit your product key to any other party without prior permission from TS, LLC. Unauthorized sharing of your product key constitutes a federal copyright violation. Under this Agreement, you agree to allow this software and associated software to store small amounts of information necessary to protect the software and/or product keys against unauthorized use.

4. LIMITED WARRANTY: THIS SOFTWARE IS LICENSED ON AN "AS-IS" BASIS. TS, LLC MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. TS, LLC DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT. TS, LLC DOES NOT WARRANT THAT THE SOFTWARE IS ERROR-FREE OR THAT ANY ERRORS WILL BE CORRECTED. Some jurisdictions do not allow the exclusion or limitation of implied warranties, so the above exclusions and limitations may not apply to you.

5. TS, LLC and its affiliates entire liability and your exclusive remedy under this Agreement is limited to no more than you paid for this Software. For any claim (including fundamental breach), in any form, related in any way to this Agreement, TS, LLC's liability will be for actual damages only and will be limited to the purchase price, if any, you paid for this Software.

6. TS, LLC and/or its affiliates will not be liable for any lost profits, lost savings, or any incidental or punitive damages or other economic consequential damages (including loss of information), even if TS, LLC or its affiliates have been advised of the possibility of such damages. TS, LLC will not be liable for any damages claimed by you based on any third party claim. This limitation of remedies also applies to any developer of Software supplied to TS, LLC. TS, LLC and the developer's limitation of remedies are not cumulative. Such developer is an intended beneficiary of this Section. This limitation of remedies also applies to any licensor of content supplied to TS, LLC. TS, LLC and the licensor's limitation of remedies are not cumulative. Such licensor is an intended beneficiary of this Section.

7. This Agreement is governed by the laws of the State of Utah. If any portion hereof is found to be void or unenforceable, the remaining provisions of this Agreement shall remain in full force and effect.

If you have further questions regarding this Software, please contact:

Tuma Solutions, LLC
legal@tuma-solutions.com

1.7.12. PSP^(SM) Process Distribution

The Process Dashboard is a general-purpose tool designed to support the enactment of any defined process. It is especially useful for supporting metrics-intensive, high-maturity processes. Although the dashboard is best known for its support of the Personal Software Process^(SM) (PSP), it can be used just as effectively with any defined process. In addition, many features of the dashboard, such as the earned value tracking tool, can be used independently of a defined process.

The first versions of the dashboard were distributed with PSP scripts and forms bundled-in. The dashboard development team was originally under the impression that this was permitted, since the copyright page of the original PSP book contains the statement:

The tables in appendix C of this book or electronic copies thereof may be copied, used, or modified by owners of this book or users of the materials it contains.

The dashboard development team originally interpreted this statement as written permission from the publisher to make electronic copies of the scripts and forms in Appendix C of the book A Discipline for Software Engineering, and incorporate those materials into the dashboard download. (Apparently, other development teams must have made the same assumption, because several other PSP tools already in existence in 1997 also included PSP scripts and forms.)

However, in February of 2002, Watts Humphrey (the author of the PSP book) brought it to our attention that the statement in the book does not grant permission to distribute electronic copies of the PSP scripts and forms. For that, special permission is needed.

Mr. Humphrey worked with Addison-Wesley to arrange permission for the PSP scripts and forms to once again be distributed for use with the dashboard, beginning with version 1.6 of the Process Dashboard. For several years, the installer for the Process Dashboard would provide the option of downloading and installing the PSP materials after individuals consented to Addison-Wesley's license agreement.

In 2005, Watts Humphrey published a new PSP book entitled A Self-Improvement Process for Software Engineers. With the release of this book, the PSP materials were revised. Also, many of the materials were separated from the book and made available as an electronic download from the Software Engineering Institute.

The Process Dashboard development team once again worked with the Software Engineering Institute, and obtained permission to distribute the updated PSP materials. The SEI granted this permission, but with conditions: most significantly, we must collect contact information from each person who downloads the PSP materials, and send this information to the SEI. (This requirement is not specific to the Process Dashboard; people are required to provide the same contact information when they download these materials from the SEI website.)

The Process Dashboard is a freely available open source application. We will not require people to enter contact information before downloading the Process Dashboard. So to meet our legal obligations under the permission statement written by the SEI, it was necessary to completely remove the copyrighted PSP materials from the main Process Dashboard download.

So in Process Dashboard version 1.11 and later, the PSP materials are distributed in the following way:

- When you download the main Process Dashboard application, no copyrighted PSP materials are included in the download.
- It is still possible to create PSP projects, capture data, and analyze your progress.
- If you are a long-time user of the Process Dashboard, you can continue to use the legacy PSP materials that you received when you installed Process Dashboard versions 1.6 through 1.10. (These materials are based upon the old book, A Discipline for Software Engineering.)
- Users who are installing the Process Dashboard for the first time will initially not have access to the PSP scripts or forms:
 - The PSP scripts will not be available until you download the PSP materials. Since the scripts are the primary differentiator between the different PSP levels, the Hierarchy Editor will only give the option of creating PSP2.1 projects. (PSP0 - PSP2 will not be available.)
 - Instead of the standard PSP Project Planning Summary forms, a simple customizable form will be provided. If you do not wish to download the PSP materials, you can instead customize this form as desired.
 - The Size Estimating Template will not be available. As a result, you will not be able to use tools like PROBE to produce estimates unless you download the PSP materials.
- Active users of the PSP are encouraged to download the new PSP materials from <http://www.processdash.com/>. The new materials provide the following benefits:
 - Project plan summary forms are customizable, allowing you to add charts and reports that support your personal process improvement goals.
 - The Size Estimating Template provides new interactive functionality. For example, you can define your own size-per-item tables, and the size of added parts will be computed automatically when you enter their type and relative size.
 - Scripts and forms display the updated terminology that appears in the recent PSP book, A Self-Improvement Process for Software Engineers.

1.7.12.1. License Agreement for PSP^(SM) Materials

PSP materials are distributed in an optional module for the Process Dashboard. Special permission to use "PSP Materials.doc" copyright © 2006 by Carnegie Mellon University, in this module is granted by the Software Engineering Institute.

ANY CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

If you wish to use any of the PSP materials contained herein you must obtain specific written permission from the Software Engineering Institute by contacting them at permission@sei.cmu.edu.

Neither Carnegie Mellon University nor the Software Engineering Institute have reviewed or endorsed Process Dashboard or this module.

The PSP materials found in this module are static and will not be updated by the Software Engineering Institute (SEI). This is not the PSP for Engineers course offered by the SEI. Persons who download and use these materials will not be credited for the SEI's PSP for Engineers I & II.

Use of the PSP materials is subject to the terms of the license set forth below:

Use

In downloading these materials individuals are granted permission to use the PSP scripts, forms, instructions, and standards for personal use only. Any derivatives or modifications to these materials require express written permission from the Software Engineering Institute.

NO WARRANTY

ANY INFORMATION, MATERIALS, SERVICES, INTELLECTUAL PROPERTY OR OTHER PROPERTY OR RIGHTS GRANTED OR PROVIDED BY ANY PARTY PURSUANT TO THIS LICENSE (HEREINAFTER THE "DELIVERABLES") ARE ON AN "AS-IS" BASIS. NO PARTY MAKES ANY WARRANTIES OF ANY KIND, EITHER EXPRESS OR IMPLIED AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, INFORMATIONAL CONTENT, NONINFRINGEMENT, OR ERROR-FREE OPERATION. CARNEGIE MELLON UNIVERSITY SHALL NOT BE LIABLE FOR INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES, SUCH AS LOSS OF PROFITS OR INABILITY TO USE SAID INTELLECTUAL PROPERTY, UNDER THIS LICENSE, REGARDLESS OF WHETHER SUCH PARTY WAS AWARE OF THE POSSIBILITY OF SUCH DAMAGES. LICENSEE AGREES THAT IT WILL NOT MAKE ANY WARRANTY ON BEHALF OF CARNEGIE MELLON UNIVERSITY, EXPRESS OR IMPLIED, TO ANY PERSON CONCERNING THE APPLICATION OF OR THE RESULTS TO BE OBTAINED WITH THE DELIVERABLES UNDER THIS LICENSE.

Licensee hereby agrees to defend, indemnify, and hold harmless Tuma Solutions, the SEI, Carnegie Mellon University, its trustees, officers, employees, and agents from all claims or demands made against them (and any related losses, expenses, or attorney's fees) arising out of, or relating to Licensee's and/or its sub licensees' negligent use or willful misuse of or negligent conduct or willful misconduct regarding the Software, facilities, or other rights or assistance granted by Carnegie Mellon University under this License, including, but not limited to, actions in the form of tort, warranty, or strict liability, any claims of product liability, personal injury, death, damage to property, or violation of any laws or regulations, regardless of whether such action has any factual basis.

Carnegie Mellon University Software Engineering Institute authored documents are sponsored by the U.S. Department of Defense under Contract F19628-00-C-0003. Carnegie Mellon University retains copyrights in all material produced under this contract. The U.S. Government retains a non-exclusive, royalty-free license to publish or reproduce these documents, or allow others to do so, for U.S. Government purposes only pursuant to the copyright license under the contract clause at 252.227.7013.

Personal Software Process and PSP are service marks of Carnegie Mellon University.

2. Planning your project

If you do not know where you are going, any road will do.

- Chinese proverb

If you do not know where you are, a map will not help.

- Watts Humphrey

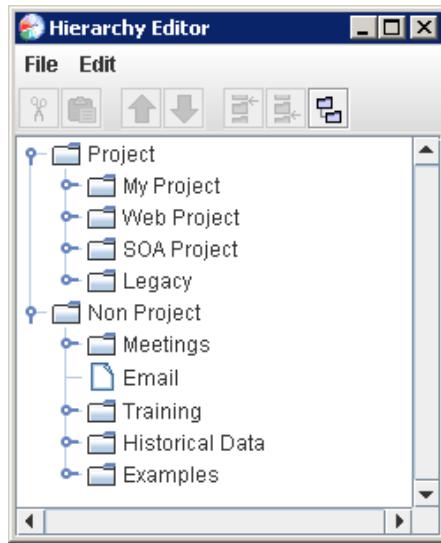
Planning a software project is just as important as planning any other complicated endeavor. In his book *A Discipline for Software Engineering* Watts Humphrey devotes three chapters to planning activities. This help topic will not rehash the ideas put forth there, but will simply serve as a reminder that planning is essential!

The Process Dashboard can provide some help in your planning activities. The [hierarchy editor](#) allows you to enter a work breakdown structure for your activities into the dashboard so that time, size, and defect data can be tracked, and the [PROBE tool](#) and [PROBE Wizard](#) help automate the PROxy Based Estimation method. Finally, the [Task and Schedule Tool](#) allows you to create a schedule and track your progress against it using earned value.

2.1. Using the hierarchy editor

The hierarchy editor allows you to create a work breakdown structure for your work, and also defines the process elements that will be used for a given software project. The hierarchy editor can be opened by selecting the "Hierarchy" option on the [configure menu](#).

Its basic function is to allow creation and deletion of nodes in a project tree. You can create any arbitrary node structure to organize the different types of projects you are working on, and add project templates to the tree from the template files provided.



In this image, some child nodes have already been defined under both "Project" and "Non Project". When the little tree control is pointing to the right, it indicates that the node can be expanded. Clicking the tree control will cause the expansion of the tree. When the control is pointing down, it indicates that the node is showing its expansion and can be collapsed.

2.1.1. The File menu

The File menu provides the choices of Close, Save, and Revert. Their functionality is pretty self-explanatory: Close closes the editor, Save saves the changes to the dashboard project tree, and Revert will revert the project tree back to the state of the last save or to when the editor was opened.

2.1.2. The Edit menu and toolbar

The Edit menu provides some rudimentary editing control on the project hierarchy. It provides the choices of Delete, Rename, Move Up, Move Down, Cut, Paste, Add Node, and Add Template. Several of these choices are duplicated on the Edit toolbar.

2.1.2.1. Deleting, moving, and renaming hierarchy nodes

Delete will allow the deletion of a portion of the project tree. It deletes the selected node and all its children from the tree. Be careful; deleted nodes and projects cannot be revived once you save your changes. Deleted projects will no longer contribute to your historical data. Therefore, you probably don't want to delete projects unless they contain no real data. If you find that many historical projects are cluttering up your hierarchy, your best strategy is to create a node in your hierarchy called "Archived Projects," then use Cut and Paste to move completed projects there.

The **Rename** option will allow you to rename a node in the the project hierarchy. When you choose this option, the name of the selected node will appear as a text entry. You can type a new name and hit return to finish, or escape to cancel the name change. But note: some nodes cannot be renamed (such as the process elements for Planning, Design, Code, etc). Also, node names cannot contain the '/' character.

Move Up and **Move Down** alter the relative position of the selected node in the tree. The selected node will still have the same parent, but it will appear earlier or later in the list of children for that parent. Since the [task and schedule](#) feature assumes that tasks will be performed in hierarchy order, Move Up and Move Down can help rearrange nodes so they appear in the order you plan to work on

them.

Cut and Paste can be used to move a node / project from one part of the tree to another. Simply highlight the node or project you wish to move, and choose Cut. Then highlight the new parent for the node or project, and choose Paste. The cut node (and all its children) will be relocated underneath the new parent. You can then use Move Up and Move Down to reposition the node relative to its siblings.

2.1.2.2. Organizing your hierarchy

The **Add Node** option on the Edit menu allows insertion of nodes into the tree. Nodes created in this way serve two general purposes:

1. They behave like "folders", which can contain other nodes and/or projects. This allows you to add organization to your work projects.
2. They can be used to track time for process-less tasks like "reading email." If your employer requires you to track your time for tasks like these, you can create an entire hierarchy of such nodes for time tracking purposes. These nodes will not interfere with your historical PSP project data - they will not contribute toward To Date calculations and they will not appear in PROBE linear regressions.

Actual projects themselves should *NOT* be added in this way, but should be added via Add Template (see below). The Add Node menu brings up a cascading second menu that provides the choices of Above, Below, and As Child. Again, these are pretty self-explanatory: Above will add a node to the tree as a sibling to the selected node, and will place it above the selected node; Below will also add a sibling node, but will place it below the selected node; As Child will add a node as a child of the selected node.

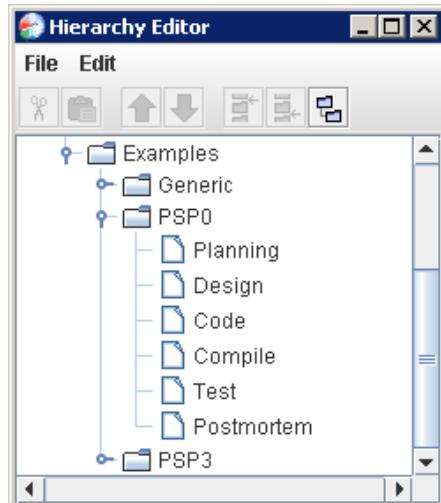
2.1.2.3. Creating projects that follow a defined process

The **Add Template** option on the hierarchy menu allows insertion of a process template into the tree. Use this choice when you are ready to define a project. The Add Template menu brings up a cascading second menu that provides choices among the various defined processes. By default, two simple templates are provided: [Generic](#) and [Timer](#). If you have installed the PSP process definitions, you will also see the 7 PSP process levels, along with [Rollup PSP Data](#). If you have installed any other add-on process sets, other choices will appear as well. For more information about the processes and their scripts, see the [process scripts](#) help topic.

When you insert a project template into the hierarchy, it will initially have a generic name like "PSP2.1". You should feel free to change its name immediately to something more descriptive. (To make dashboard operation clear, this help document sometimes displays projects with names like "PSP0", but such bland project names are neither required nor expected for actual use.)

2.1.3. Process templates

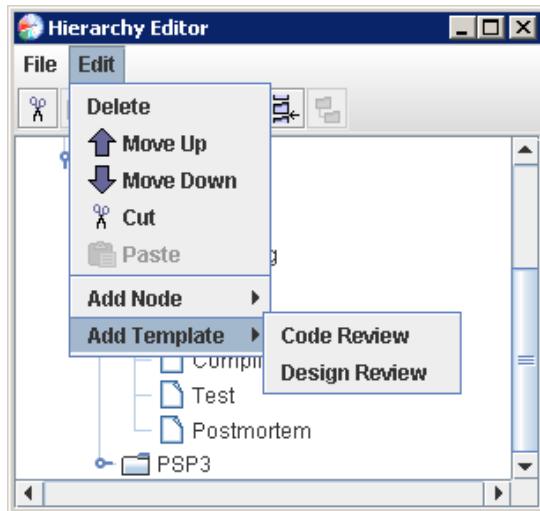
When a process template is added, all the process phase elements that are defined for that process will be automatically added as children.



These process phases cannot be edited in any way (with a couple exceptions). If a process phase is selected and the Edit menu is clicked, the menu will show all entries greyed out and disabled. This is to protect the integrity of the processes themselves.

2.1.4. Processes PSP0 - PSP1.1

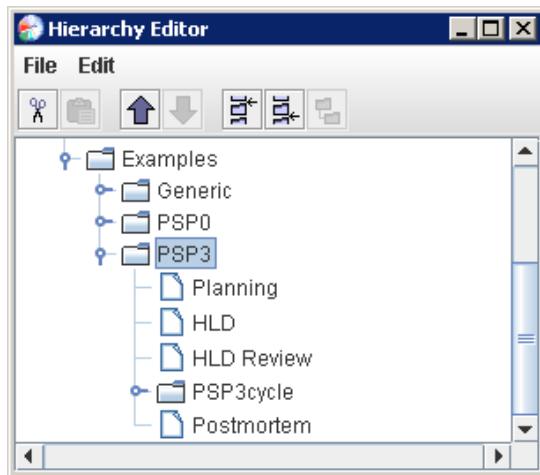
PSP processes PSP0 through PSP1.1 do not provide for design or code reviews, as those improvements are introduced in PSP level 2. However, the dashboard will allow you to add design and or code reviews to these low level PSP processes if you choose. If you pick on a project that was defined from the PSP0, PSP0.1, PSP1, or PSP1.1 process templates and then choose the "Add Template" menu you will see Design Review and Code Review choices. Either of these can be added to the process. They will be automatically inserted in the correct places in the process phase order.



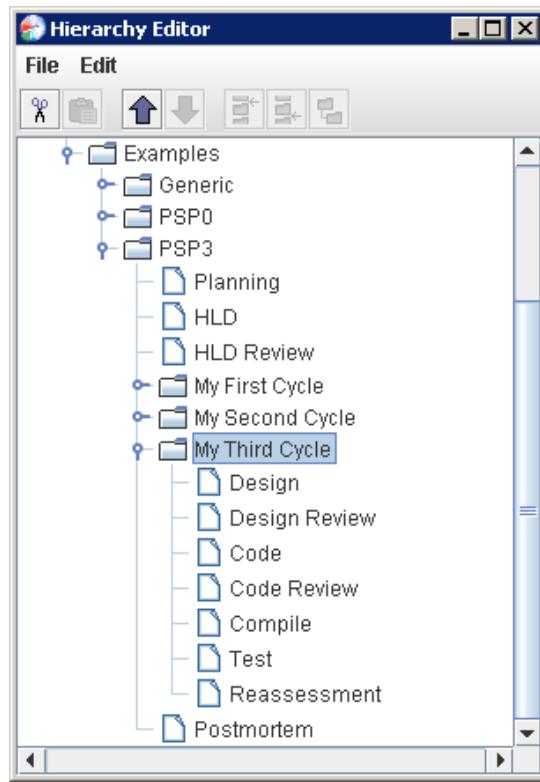
Once you insert a Design Review or Code Review phase, it will appear in the process scripts and forms for the project. If you later change your mind and decide you don't want the review phase after all, you can highlight it and choose Delete from the Edit menu.

2.1.5. The PSP3 process

The PSP3 process is a cyclic development process. As such, it provides a basic structure for the whole project and also provides the ability to add development cycles. When a PSP3 process is added to the hierarchy, it will have the basic skeletal structure defined (Planning, High Level Design, High Level Design Review, and Postmortem phases), along with one PSP3 cycle.



Once you decide upon your cyclical development strategy, you should create additional cycles underneath the PSP3 project. Each of these cycles will contain the other PSP3 process phases (Design, Design Review, Code, Code Review, Compile, Test, and Reassessment). The PSP3 cycles are added just like the PSP3 project itself, via the "Add Template" menu. When a PSP3 based project is selected and the Edit menu is chosen, the Add Template menu item will have only one available choice: PSP3cycle. The cycle will be placed appropriately before the Postmortem step. You can input as many cycles as you would like in this manner.



2.2. The process scripts

Building upon the flexibility of the world-wide-web, the dashboard displays many scripts, forms, and reports in your web browser. See the [Installation](#) help topic for more info about which web browsers are supported and how to install them if necessary.

The connection between the dashboard and the browser is facilitated by a very small web server that is running within the dashboard. It is this web server, combined with some Java scripting, that provides all the live data communication between the dashboard and the browser. One consequence of this is that the various reports and charts that are displayed within the browser will not be available unless the dashboard is running.

2.2.1. The PSP Process Scripts

An add-on module for the Process Dashboard provides nearly all the scripts defined in the book *A Self-Improvement Process for Software Engineers* by Watts Humphrey.

The PSP forms and scripts in this add-on module are designed to be as close to the "real thing" as possible (the "real thing" being the forms and tables in Watts Humphrey's book). Where the online forms and scripts deviate, the text is color coded as follows:

- Where the dashboard makes a certain step unnecessary the text is colored in grey (e.g. writing entries in a time log).
- Where the dashboard changes the way a certain step is performed, the text is colored in green (e.g. using the PROBE tool or the PROBE wizard instead of performing statistical calculations via a calculator or spreadsheet).

Many of the process forms contain links to other forms that are used at that process level, making navigation through the forms simple and intuitive. Also, many of the process scripts contain little stopwatch icons on them. These icons provide another way to navigate and time a process phase. When you click on one of these icons, the current phase being measured will change to the phase indicated by the icon and the timer will start measuring time for that phase.

For information on how to access the forms, see the [Accessing process scripts](#) help topic.

The following lists detail the forms and scripts provided by the dashboard at each PSP level. Items marked by * are "live" forms that either update themselves with process data (e.g. the project summary forms) or can be expanded as necessary to permit multiple data entry (e.g. the size estimating template or the PSP3 cycle summary form).

At PSP0:

- The Process script
- The Planning script
- The Development script
- The Postmortem script
- The Project Plan Summary form*

At PSP0.1:

- The Process script
- The Planning script
- The Development script
- The Postmortem script
- The Project Plan Summary form*

At PSP1:

- The Process script
- The Planning script
- The PROBE estimating script
- The Size estimating template*
- The Size estimating template instructions
- The Development script

- The Postmortem script
- The Project Plan Summary form*

At PSP1.1:

- The Process script
- The Planning script
- The PROBE estimating script
- The Size estimating template*
- The Size estimating template instructions
- The Development script
- The Postmortem script
- The Project Plan Summary form*

At PSP2:

- The Process script
- The Planning script
- The PROBE estimating script
- The Size estimating template*
- The Size estimating template instructions
- The Development script
- The Design Review script
- The Code Review script
- The Postmortem script
- The Project Plan Summary form*

At PSP2.1:

- The Process script
- The Planning script
- The PROBE estimating script
- The Size estimating template*
- The Size estimating template instructions
- The Development script
- The Design Review script
- The Code Review script
- The Postmortem script
- The Project Plan Summary form*

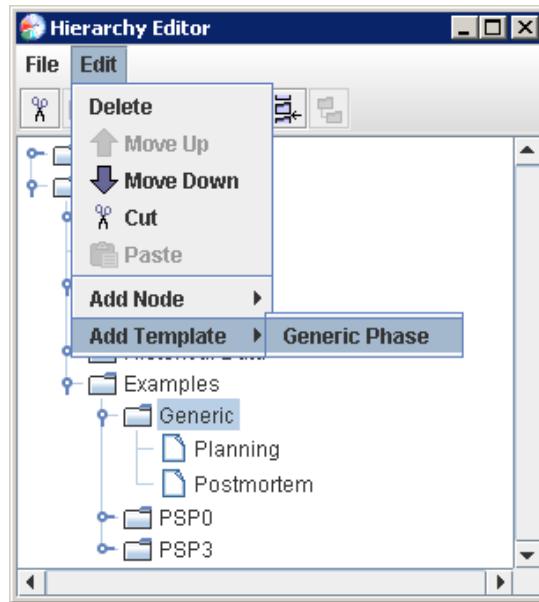
At PSP3:

- The Process script
- The Planning script
- The PROBE estimating script
- The Size estimating template*
- The Size estimating template instructions
- The High Level Design script
- The High Level Design Review script
- The Development script
- The Design Review script
- The Code Review script
- The Postmortem script
- The Project Plan Summary form*
- The Cycle Summary form*

2.2.2. The "generic" process template

Very much of the methodology that is employed by the PSP can be used in other endeavors. Any kind of work that has measurable time, size, and/or defects can probably benefit from a PSP-like process. The generic process template gives estimation, measurement, and defect tracking support for other work efforts.

When you create a project based on the generic process in the [hierarchy editor](#), it will initially contain only a planning phase and a postmortem phase. You can then add additional phases (and give them any names that you want) by selecting the node corresponding to the generic project, then selecting "Generic Phase" from the "Add Template" menu:



The generic process then creates a tailored project plan summary form. It allows entry of time and size estimates, and even lets you enter the unit of size measurement. For example, you could run the generic process to measure and track your progress as you write a technical paper. The size unit would end up being something related to the work of writing (perhaps pages). The generic process would capture your productivity rates and allow you to enter defects. If you have installed the add-on with PSP process materials, the Size Estimating Template and PROBE Wizard will also be available to provide support for the estimation process.

2.2.3. The "timer" process script

The timer process allows for estimating and measuring activities that have only a time aspect to them. If you regularly perform some kind of task that pretty much remains the same each time, you could record estimates and times for that task to attempt to provide some engineering discipline.

2.2.4. Custom process scripts

The dashboard has been designed from the ground up to support custom processes. The PSP scripts and forms that you see are not hard-coded into the tool; instead, they are dynamically loaded from simple HTML and text files. Thus, it is possible to create your own custom process scripts, forms, and data that will be dynamically integrated into the dashboard.

More information on creating custom processes is available in the [Advanced Topics](#) section of the help.

2.2.5. Custom process vs Generic process?

When should you use the generic process, and when should you create a custom process? In general, you should use the generic process:

- If the process for this task is unlikely to be reused in the future (for example, because the task at hand is a departure from the work you normally do).
- If the process for this task is still immature and uncertain. As you begin creating your own custom process definitions, the flexibility of the generic process will allow you to radically change your process from one task to the next in response to process

improvement proposals. Once you decide upon and finalize the [list of phases](#) in your process, you are ready to begin using a custom process script.

The generic script gives you a lot of flexibility - the phase names and other process specifics can change from task to task, or even in the middle of a generic project! This flexibility comes at a price, however: without an unchanging process definition, it isn't possible to compare "apples to apples" across numerous generic processes. Therefore, the generic process cannot calculate "to date" data like a custom process can. On the other hand, a change to your process definition can potentially invalidate all the historical data you've collected, so it is best to wait and define a custom process when you feel you've confidently identified its phases.

2.2.6. The Team Process

The dashboard also includes a sophisticated metrics collection framework designed to support team projects. The team project template allows you and your teammates the ability to define arbitrary hierarchical structures of project subcomponents and tasks. It makes it easy to allocate and reassign work between team members, and it automatically rolls up individual data to the team level. If you plan to collaborate with other individuals, you should use the team process.

For more information on the team project support, see the [Team Use](#) section of the Users Manual.

2.3. Entering process data

There are two main ways of entering data into the Process Dashboard. The first is through various input tools and dialogs provided by the dashboard (e.g. the [defect dialog](#), the [time log editor](#), etc.).

The second is through various process forms. The scripts and forms are provided through an integrated web interface. See the [Accessing Process Scripts](#) help topic for more info on how to access the scripts and forms. To find out more information about the different process scripts and forms that are available, see the [process scripts](#) help topic.

2.3.1. The nature of the dashboard forms

The dashboard forms are implemented via static and sometimes dynamic HTML pages. Unlike the HTML forms you see on the internet, the forms on these HTML pages are "live" at all times, much like a spreadsheet. You can type values in the input fields and other fields will recalculate automatically, even if those fields are in different browser windows. There is no need to press any kind of "submit data" button.

After typing in a value, you must generally "leave" that input field to signal that you are finished entering information. Pressing the "Tab" key (to move to the next input field) or clicking anywhere else on the page is generally sufficient. As soon as the "input cursor" leaves the field in question, the dashboard will automatically pick up your data, recalculate related data items, and save your data for future sessions.

If you enter an invalid value for a data item (for example, typing "foo" in an input field that is expecting a number), the dashboard will reject the invalid value and restore the previous value for that input field.

The dashboard uses various visual cues to show you where data is missing or where a calculated value could not be completed. Here is a partial screen shot of a dashboard form that shows several of the ways the dashboard indicates incomplete or bad data:

Summary	Plan	Actual	Est.Eff.%
Productivity (Pages/Hr)	#DIV/0!	#VALUE!	
Time	0	0	#DIV/0!
Size (Pages)	30	?????	#VALUE!
Defects/Page		#VALUE!	

Divide-by-zero
 Calculated (read-only) fields
 Missing required input value
 Calculated value referencing a missing value

When a required data value is *missing*, the dashboard will display "?????". This is an indication that you still need to enter a value for that data item. Therefore, you should ensure that there are no "?????" symbols in the Plan column of your project plan summary before marking the Planning phase complete. Similarly, make certain you fill in all missing values in the Actual column of your project plan summary before marking the project complete.

If the dashboard cannot recalculate a formula because the formula references a missing ("?????") or otherwise invalid data value, it will display "#VALUE!" as the result of the calculation. When you see "#VALUE!" in a dashboard form, it almost *always* means that there is a "?????" field somewhere that you still need to fill in.

If the dashboard recalculates a formula and encounters a divide-by-zero error, it will display "#DIV/0!" as the result of the calculation. Divide-by-zero errors are common when you first begin using the dashboard, since you will not have any historical data.

To make all this possible, the dashboard software actually contains a very small HTTP server embedded within it. This server allows communication between the HTML pages and the process data that is tracked by the dashboard. If you close the dashboard program while one of the HTML forms is open, the form will reset and display "NO CONNECTION". You can reestablish the connection simply by

restarting the dashboard, then telling your browser to reload the HTML page.

2.3.2. Frozen Calculations

If you edit the value in an input field, and related data items don't recalculate, the data is probably *frozen*. Remember that when you mark the planning phase of a PSP project complete, all the Planned data elements are baselined and stop recalculating. Similarly, when you mark a PSP project complete, the Actual and To Date data elements will freeze and stop recalculating. So if you expect formulas to recalculate and they don't, check to make certain that the planning phase and the entire project are not marked complete. For more information, see the [Data Freezing](#) help topic.

2.3.3. Read-only data fields

The Process Dashboard automatically calculates many data elements for you. Normally, these calculated data values are not directly editable, but rather they will update when other data elements to which they are related get updated. If you are using a recent web browser, these values are displayed with a grey background.

Sometimes you may decide that you want to override one or more of these calculated values. You can "unlock" these read-only data elements by clicking on the "[Unlock read-only data](#)" link at the bottom of the page. Unlocking read-only data will make all the elements on the page editable, allowing you to override calculated values with your own.

Unlocking read-only data is useful for times when you intentionally want to override a calculation, but it is not without risk. Once you have overridden a calculation, it will no longer automatically update in response to changes in related data items. In addition, related calculations may no longer add up like you would normally expect. Therefore, this feature should be used with caution.

After you have overridden a calculation with some value, you may decide that you want the old calculation back. Just type `DEFAULT` in the input field, and when you leave the field the old calculation will be restored.

2.3.4. Exporting form data

When you open a dashboard form, the form connects to the process dashboard to retrieve the most up-to-date values for all the data elements on the page. If you choose the "File → Save" option from your browser, you will find that only the form is saved; the values contained in the form are not. If you want to save an electronic copy of a dashboard form (for example, to email to a coworker, or to save with project documentation), click the "Export to HTML" link at the bottom of the page. The dashboard will create an HTML version of the form containing a *snapshot* of your data. This HTML snapshot can be saved using the "File → Save" option of your browser.

If you would like to perform additional analysis on the data in a dashboard form, use the "Export to Excel" link at the bottom of the page. This will open the form in Excel, where you can reference the data in Excel charts and equations. For more information on the Excel export feature, see the [Exporting your data](#) help topic.

Important: the Excel version of the form contains *exported* data only. If you edit values in the Excel spreadsheet, your changes will not be imported back into the Process Dashboard. To change a data value in the Process Dashboard, you must make the change in a dashboard form.

2.4. Using the Size Estimating Template

The Size Estimating Template is used to help make estimates of how big a new software project will be or how long it will take to create the project (size and time estimation). It is patterned after the Size Estimating Template from Watts Humphrey's book *A Self-Improvement Process for Software Engineers*.

Size Estimating Template - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost:2468/Project/My+Project/Release+1/Component+A//psp2.1/sizeest.class

Size Estimating Template ([instructions...](#))

Project Name /Project/My Project/Release 1/Component A

Project Owner My Name

Size Measure LOC

BASE PARTS				Estimated				Actual			
BASE	DEL.	MOD.	ADDED	BASE	DEL.	MOD.	ADDED	BASE	DEL.	MOD.	ADDED
0	0	0	0	0	0	0	0	0	0	0	0
add more rows for base parts...				Total: 0				Total: 0			

PARTS ADDITIONS				Estimated				Actual			
TYPE	ITEMS	REL.SIZE	SIZE	SIZE	ITEMS	NR	NR	SIZE	ITEMS	NR	NR
add more rows for parts additions...				Total: 0				Total: 0			

REUSED PARTS				Estimated		Actual	
				SIZE	SIZE	SIZE	SIZE
add more rows for reused parts...				0	0	0	0
				Total: 0	Total: 0	Total: 0	Total: 0

Instructions: During the planning phase, complete the "Estimated" columns in the form above, then use the [PROBE Wizard](#) to fill out the fields below.

	SIZE	TIME
Added Size (A):	A = BA + PA	0
Estimated Proxy Size (E):	E = BA + PA + M	0
PROBE estimating basis used: (A, B, C, or D)		D
Correlation: (r^2)		0
Regression Parameters:	B_0 (size and time)	0
Regression Parameters:	B_1 (size and time)	0
Projected Added and Modified Size (P):	P = B_0 size + B_1 size * E	?????
Estimated Total Size (T):	T = P + B - D - M + R	#VALUE!
Estimated Total New Reusable (NR):	(sum of NR items)	0
Estimated Total Development Time:	Time = B_0 time + B_1 time * E	?????
Prediction Range:	Range	0
Upper Prediction Interval:	UPI = P + Range	0:00
Lower Prediction Interval:	LPI = P - Range	0:00
Prediction Interval Percent:		0%

[View PROBE Report](#)

Adapted from "PSP Materials," copyright © 2005 Carnegie Mellon University. Used by permission.

[Unlock read-only data](#) Export to: [HTML](#) [Excel](#) [Help...](#)

Done

(Note: Watts Humphrey released an earlier book called *A Discipline for Software Engineering* which used a very different layout for this form. If you are using the old version of the PSP scripts, you will see the old form layout. The paragraphs below describe the new version of the form.)

Right near the top of the form is a link to bring up the Size Estimating Template instructions form. The dashboard will bring up the instructions in a separate web browser so that the two can be placed side by side on your screen.

Also, there are several links on the form that will add rows to the form. If you need more room than the form provides, just click on the appropriate link and it will expand for you.

The header section contains entries for the name of the project and your name. Both of these fields should be automatically filled in by the dashboard (if your name does not appear, go ahead and enter it -- it will show up on the other forms as well).

The Size Estimating Template can theoretically be used to estimate any type of work, so the official version of this form includes a header field describing the unit of size measurement used for the numbers on the form. However, the PSP process definitions are currently focused on support for software development, so this value will be hardcoded to "LOC" for PSP tasks. If you are using the Size Estimating Template in conjunction with a [generic project](#), you should enter the appropriate size metric in this field.

2.4.1. Base Parts

The Base Parts section is where you enter the information about any preexisting components that may need modifications during the course of your project. If you have several of these components, you can use a separate line for each.

The form has blanks for both Estimated and Actual sets of data. When using this form for planning purposes, the Estimated columns should get filled in. After the project is finished and you are in the postmortem phase, you should enter data in the Actual columns.

The form asks for the base size and the size of deleted, added, and modified elements. In the BASE column, enter the size of the preexisting component. For example if you are creating a modification to an already existing program, the LOC from the existing program count as base LOC.

In the next three columns, you estimate the size of deleted, modified, and added elements. For example, you would enter the number of existing LOC you think you might delete, the number of existing LOC you think you might modify, and some number of new LOC you think you might add. A histogram appears to show you how these numbers relate to the size of the preexisting, base program. You can drag sliders to estimate an approximate percentage.

2.4.2. Parts Additions

Parts Additions is the section where you will record the bulk of new development in most of your projects. This section allows you to estimate the new components you believe you will create during the course of this project. These could be new objects, new files, new functions, new subroutines, new methods, etc. - whatever makes the most sense for your estimating needs.

Blanks are provided to enter the description of each added part, the type of part that is being added, the number of items that are going to be added of this type, the relative size of these items, and the numeric estimate/measure of the size of each added part.

When you are estimating in LOC (for example, for a PSP task), the type field will include a default set of categories: namely Calculation, Data, I/O, Logic, Setup or Text. However, you can define your own categories by clicking on the hyperlink at the top of the column. New PSP users should usually stick to the preset types; as you gain experience, you may use your PSP training to define your own proxy categories. Select the category that best fits the type of each added part.

The number of items can be useful when you are estimating a high-level object. For example, if you are estimating the size of an object or class in an object-oriented language, you could use this field to record the number of methods that will be present in an object you plan to create. If you are using some custom proxy type category, this field could capture the count of proxy elements you plan to produce. If these scenarios don't apply, you can leave this field blank or enter the number "1".

The relative size pull down allows you to choose from subjective size categories such as Very Small, Small, Medium, Large, and Very Large. To produce the most accurate estimates, you should aim for a level of planning granularity that allows most of your planned parts to be of Medium size, with a balance of smaller and larger parts.

After you enter values in the previous three columns, the data in the proxy category table will be used to produce an estimated size. If the generated number seems too big or too small, choose a different relative size.

As you enter new parts, consider whether each new part is sufficiently general purpose that it can be checked into a reuse library. Generally this means more than just code that can probably be reused in the general sense. It usually has to do with proper packaging of the code as well: Are proper interfaces defined? Is the code sufficiently well documented? Etc. If you intend to design the code for reuse, put a check mark in the NR (New Reusable) column.

2.4.3. Reused Parts

Reused parts are objects that are brought into the program from a reuse library and used verbatim. New PSP users can sometimes be confused by the line between a reused part and a base part. The main rule is that you must not count the same component as both base and reused - otherwise the size accounting formulas will not work out correctly. In addition, the form will not let you enter added/deleted/modified size for reused parts, so you should only put items in the reused parts section if they can be reused with no changes.

2.4.4. Performing the size/time estimation

Once you have thought through all the changes that are going to be made to the software, you are ready to calculate a prediction of size and or time. The form will add up numbers from the sections above to produce an Estimated Proxy Size, which the PROBE method will use as your base estimate.

REUSED PARTS		Estimated SIZE	Actual SIZE
add more rows for reused parts...		0	0
		Total: 0	0

Instructions: During the planning phase, complete the "Estimated" columns in the form above, then use the [PROBE Wizard](#) to fill out the fields below.

SIZE	TIME
69.3	69.3

Added Size (A):
Estimated Proxy Size (E):

$A = BA + PA$
 $E = PA + PA + M$

PROBE Wizard link on the Size Estimating Template

The Size Estimation Template offers a link that will allow you to choose the PROBE Wizard to help you make your estimates. If you use the PROBE Wizard, it will read your Estimated Proxy Size off the size estimating template automatically, and will save your chosen estimates back to the size estimating template automatically. For more information about using the PROBE Wizard, see the [Using the PROBE Wizard](#) help topic.

On the other hand, if you use the PROBE tool, you will need to type the Estimated Proxy Size into the Estimate field on the tool. Once you have investigated the relationships among your process data, you should manually copy your final estimates back to the size estimating template, along with the various other measures that come about as a result of the estimation (B_0 , B_1 , UPI, LPI, prediction range, r^2 , probe method selected). If you use PROBE method C or D, the Prediction Range field and the other numeric fields below it can be left blank. For more information about the PROBE tool provided by the dashboard, see the [Using the PROBE tool](#) help topic.

2.4.5. Entering Actual data during Postmortem

Once you reach the Postmortem phase and you have the Actual size data, you should return to this form to enter the actuals for the various measures on the form (base, deleted, modified, and added size, counts for the various objects, etc.).

Size Estimating Template - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost:2468/Project/Legacy/Product+1/SCR+42//psp2.1/sizeest.class

Size Estimating Template [\(instructions...\)](#)

Project Name /Project/Legacy/Product 1/SCR 42

Project Owner My Name

Size Measure LOC

Estimated				Actual			
BASE	DEL.	MOD.	ADDED	BASE	DEL.	MOD.	ADDED
345	0	0	10.9	0	0	0	0
345	0	0	10.9	0	0	0	0

BASE PARTS

Base Program

[add more rows for base parts...](#)

PARTS ADDITIONS

Logic for new calculation

Presentation templates

Customization UI

Customization data structures

[add more rows for parts additions...](#)

Estimated				Actual			
TYPE	ITEMS	REL. SIZE	SIZE	NR	SIZE	ITEMS	NR
Calculation	1	Very Large	54	<input type="checkbox"/>			<input type="checkbox"/>
I/O	2	Large	43.2	<input type="checkbox"/>			<input type="checkbox"/>
I/O	1	Very Small	28.9	<input type="checkbox"/>			<input type="checkbox"/>
Data	3	Medium	26.5	<input type="checkbox"/>			<input type="checkbox"/>

Total: 153

Estimated		Actual	
SIZE	ITEMS	SIZE	ITEMS
0	0	0	0
0	0	0	0

REUSED PARTS

[add more rows for reused parts...](#)

Total:

Enter actual Size data

That's all the data that has to be entered on this form. The dashboard calculates as much of the data for you as possible. The numbers entered and calculated on this form will automatically appear on the Project Plan Summary form.

2.5. Using the PROBE Wizard

The PROBE process is described in the PSP book *A Self-Improvement Process for Software Engineers*, but it can be complicated to follow. Even when the calculations are performed for you (for example, by the [PROBE Tool](#)), there are many cross-checks you must perform to verify the validity of the estimates. For individual who do not thoroughly understand the calculations being performed, it is easy to make a mistake.

The PROBE Wizard recognizes and reduces this complexity by automating as much of the process as possible. The PROBE Wizard can be opened by clicking on the link found on the [Size Estimating Template](#).

When the PROBE Wizard is activated, it starts with a welcoming screen that reminds you of the nature of estimating with the PROBE method.

PROBE Wizard
/Project/Legacy/Product 1/SCR 42

Size and time estimating are complex skills, best developed through experience. As you strive to improve your estimating skills, your primary goal should be estimating accuracy.

The Proxy Based Estimating process (PROBE) includes statistical methods that can help you to improve your estimating accuracy over time. These methods analyze your past estimating performance, and attempt to mathematically remove any consistent estimating bias found.

This wizard will help you to perform this portion of the PROBE process. Keep in mind that:

- PROBE is not magic - you must actively exercise your engineering judgement to ensure that you produce sensible estimates.
- The PROBE methods implicitly assume that the project you are currently estimating is similar to the projects you have completed in the past. If this is not true, use PROBE-generated estimates with caution.
- PROBE works best when you are consistent in your use of the Size Estimating Template.

To begin, click the "Continue" button.

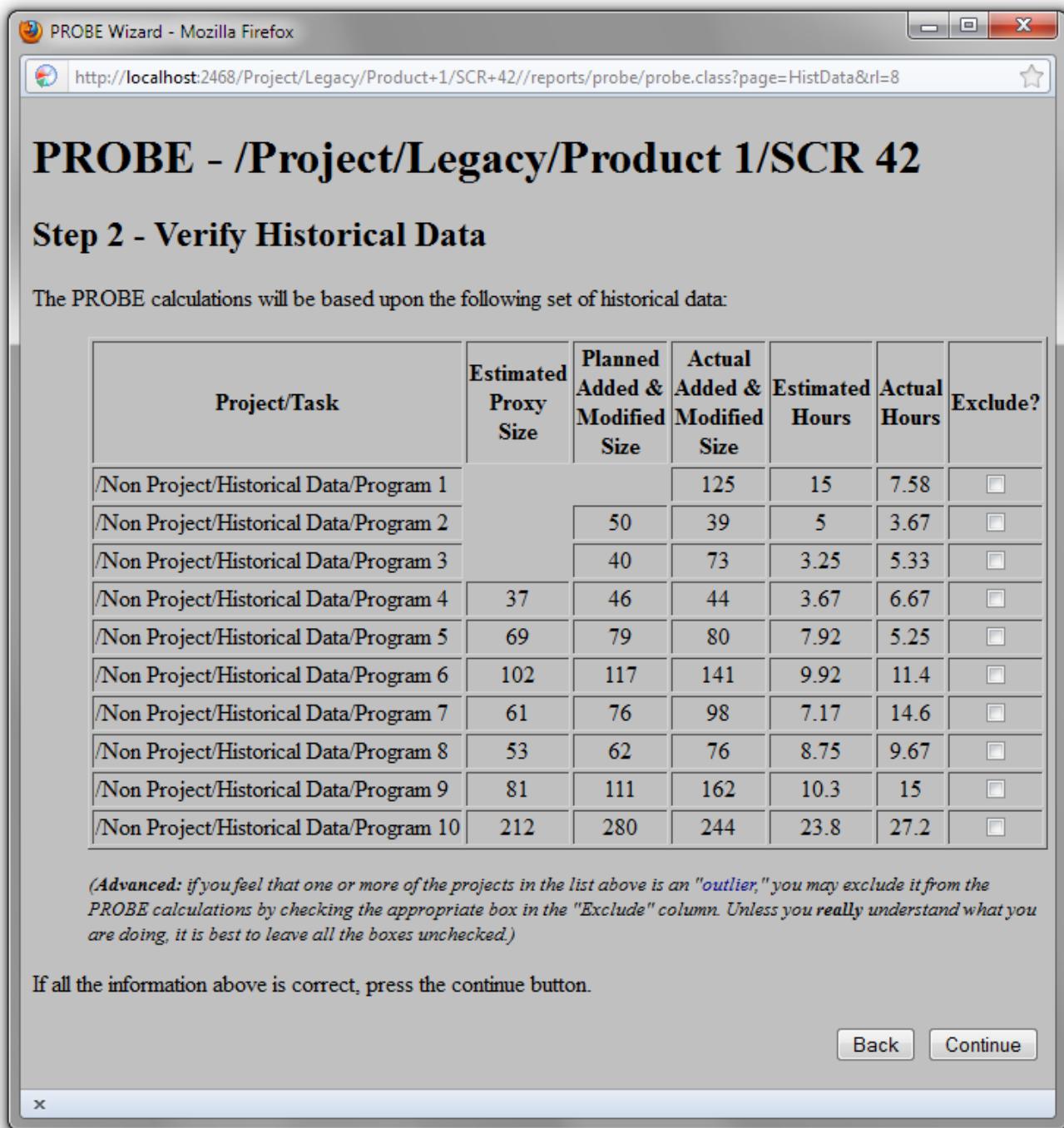
Continue

2.5.1. PROBE Wizard Step 1

Step 1 in the PROBE method is to verify that the Estimated Proxy Size is accurate. The Wizard shows you the number you have input on the Size Estimating Template and asks you to verify that this is indeed a good estimate.

2.5.2. PROBE Wizard Step 2

Step 2 is to verify your historical data. The PROBE Wizard puts up a table of numbers that shows your estimations from previous PSP projects.

A screenshot of a Mozilla Firefox browser window. The title bar says "PROBE Wizard - Mozilla Firefox". The address bar shows the URL "http://localhost:2468/Project/Legacy/Product+1/SCR+42//reports/probe/probe.class?page=HistData&rl=8". The main content area has a large title "PROBE - /Project/Legacy/Product 1/SCR 42" and a section header "Step 2 - Verify Historical Data". Below this, a note states: "The PROBE calculations will be based upon the following set of historical data:". A table follows, showing historical data for 10 programs. The columns are: Project/Task, Estimated Proxy Size, Planned Added & Modified Size, Actual Added & Modified Size, Estimated Hours, Actual Hours, and Exclude?. The data is as follows:

Project/Task	Estimated Proxy Size	Planned Added & Modified Size	Actual Added & Modified Size	Estimated Hours	Actual Hours	Exclude?
/Non Project/Historical Data/Program 1		125	15	7.58	<input type="checkbox"/>	
/Non Project/Historical Data/Program 2		50	39	5	3.67	<input type="checkbox"/>
/Non Project/Historical Data/Program 3		40	73	3.25	5.33	<input type="checkbox"/>
/Non Project/Historical Data/Program 4	37	46	44	3.67	6.67	<input type="checkbox"/>
/Non Project/Historical Data/Program 5	69	79	80	7.92	5.25	<input type="checkbox"/>
/Non Project/Historical Data/Program 6	102	117	141	9.92	11.4	<input type="checkbox"/>
/Non Project/Historical Data/Program 7	61	76	98	7.17	14.6	<input type="checkbox"/>
/Non Project/Historical Data/Program 8	53	62	76	8.75	9.67	<input type="checkbox"/>
/Non Project/Historical Data/Program 9	81	111	162	10.3	15	<input type="checkbox"/>
/Non Project/Historical Data/Program 10	212	280	244	23.8	27.2	<input type="checkbox"/>

(Advanced: if you feel that one or more of the projects in the list above is an "outlier," you may exclude it from the PROBE calculations by checking the appropriate box in the "Exclude" column. Unless you really understand what you are doing, it is best to leave all the boxes unchecked.)

If all the information above is correct, press the continue button.

[Back](#) [Continue](#)

On this pane, you can exclude data points that you know to be "outliers" statistically. By clicking on the checkboxes at the end of each row, you can exclude any data point that you know does not represent your normal work processes. If you are not completely sure about the consequences of excluding outlier data, you should probably leave all of the boxes unchecked.

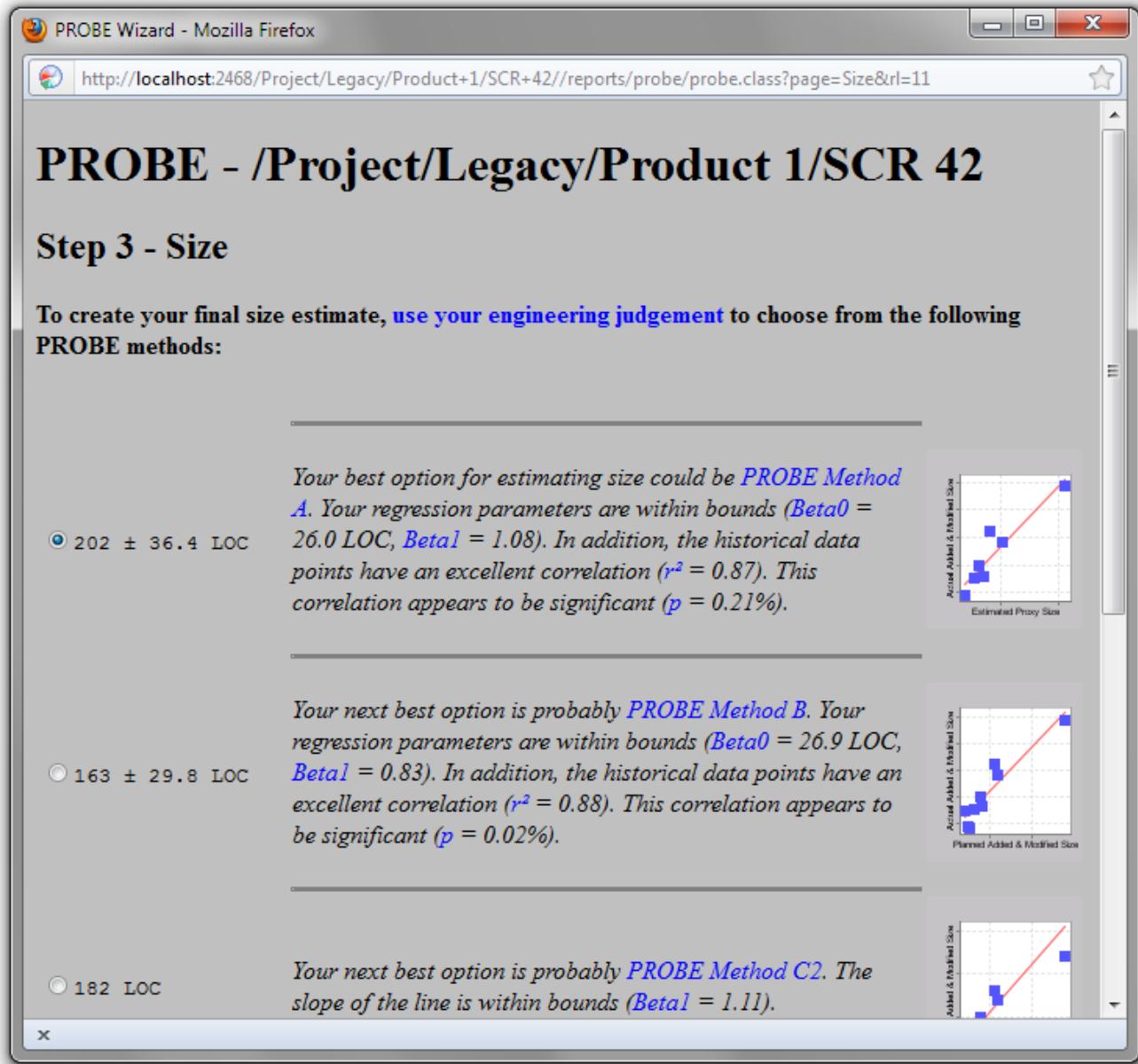
2.5.3. PROBE Wizard Step 3

Step 3 is to estimate the size of the new development effort. The PROBE method (as implemented by the Process Dashboard) outlines 5 ways to estimate size:

- A Estimate size based on a good correlation between previous Estimated Proxy Size data and Actual Added & Modified Size data. This method is the most preferred method, but can only be used if you have sufficient data that correlates well.
- B If Proxy Size estimates do not correlate well enough to use method A, PROBE suggests estimating size based on a good correlation between previous Planned Added & Modified Size data and Actual Added & Modified Size data. Again, this method can only be used if you have sufficient data that correlates well.

- C1 If neither of the data pairings for methods A and B correlate, but you do have Estimated Proxy Size data, you can use that data to do the estimate. This method requires at least some Estimated Proxy Size data be recorded.
- C2 If you do not have any Estimated Proxy Size, but you do have Planned Added & Modified Size data, you can use that data to do the estimate. This method requires at least some Planned Added & Modified Size data be recorded.
- D If you have no historical data, method D just indicates that you should use your best engineering judgement to make the estimate.

The Wizard will present all of these possibilities to you in decreasing order of preference based on how well each method seems to track (or correlates) with your historical data. You just have to choose the method you would like to follow by picking from a set of radio boxes.



The Wizard also has links to explanatory help windows that will pop up and give you more detail about the different types of calculations that it is doing to generate the different estimates. These links are shown in blue text.

2.5.4. PROBE Wizard Step 4

Step 4 is very similar to step 3, except the estimation is for time instead of size. The same basic set of methods is available, and the Wizard will offer each data estimate in decreasing order of preference.

The PROBE method (as implemented by the Process Dashboard) outlines 6 ways to estimate time:

- A Estimate time based on a good correlation between previous Estimated Proxy Size data and Actual time data. This method is the most preferred method, but can only be used if you have sufficient data that correlates well.
- B If Proxy Size estimates do not correlate well enough to use method A, PROBE suggests estimating time based on a good correlation between previous Planned Added & Modified Size data and Actual time data. Again, this method can only be used if you have sufficient data that correlates well.
- C1 If neither of the data pairings for methods A and B correlate, but you do have Estimated Proxy Size data, you can use that data to do the estimate. This method requires at least some Estimated Proxy Size data be recorded.
- C2 If you do not have any Estimated Proxy Size, but you do have Planned Added & Modified Size data, you can use that data to do the estimate. This method requires at least some Planned Added & Modified Size data be recorded.
- C3 If you only have Actual Time and Size data, you can use that data to do the estimate.
- D If you have no historical data, method D just indicates that you should use your best engineering judgement to make the estimate.

Unlike other wizards you may have used in the past, the PROBE Wizard updates the data in the dashboard whenever you click the Continue button to move onward (that is why there is no Cancel button on the Wizard). If you make a mistake while using the Wizard, you'll have to use the Back button to go back and fix it, or make the change manually on the Size Estimating Template. Of course, you can always just run the PROBE Wizard a second time.

2.5.5. PROBE Wizard Step 5

At this point, you are finished making your estimates. Step 5 encourages you to check the estimates to make sure that they make sense.

The screenshot shows a Firefox browser window titled "PROBE Wizard - Mozilla Firefox". The address bar displays the URL: <http://localhost:2468/Project/Legacy/Product+1/SCR+42//reports/probe/probe.class?page=Check&rl=1>. The main content area has a large title "PROBE - /Project/Legacy/Product 1/SCR 42" and a sub-section "Step 5 - Check Estimates". Below this, a message states: "You have estimated that this project will require:" followed by a bulleted list: "• 202 Actual Added & Modified Size" and "• 20.6 Total Hours". A note below says: "Your estimates for size and time translate into a planned productivity of 9.82 LOC/Hr. This is consistent with your 'To Date' productivity of 10.2 LOC/Hr (± 3.57)". At the bottom, a message says: "Congratulations! You have completed the PROBE process. Press the Finish button to close this window." Two buttons are visible: "Back" and "Finish".

Clicking Finish dismisses the PROBE Wizard.

2.5.6. A sanity check

After you use various PROBE methods to independently generate your size estimate and your time estimate, it is always a good idea to "sanity check" those estimates to ensure that they are reasonable and in sync with each other. The final screen of the PROBE Wizard

helps with this task. It uses your size and time estimates to calculate your planned productivity, and then compares this number with your historical productivity to date. When your planned productivity is significantly different than your historical productivity, it is usually a warning flag indicating that either your size estimate or your time estimate (or possibly both) is incorrect. When this occurs, the PROBE wizard strongly encourages you to reexamine your estimates.

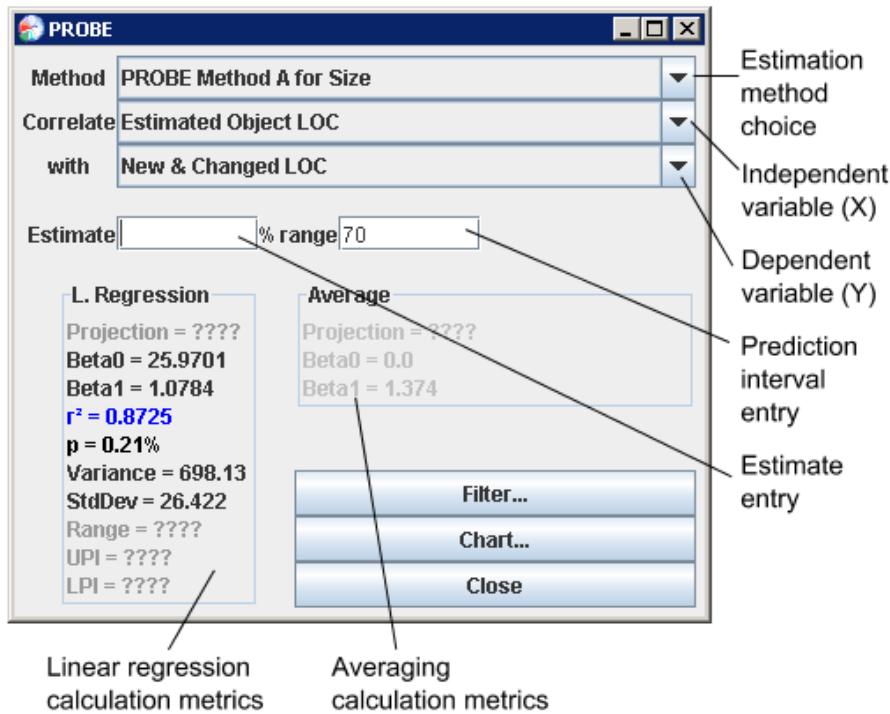
The screenshot shows a Firefox browser window titled "PROBE Wizard - Mozilla Firefox". The URL in the address bar is <http://localhost:2468/Project/Legacy/Product+1/SCR+42//reports/probe/probe.class?page=Check&rl=6>. The main content area displays the title "PROBE - /Project/Legacy/Product 1/SCR 42" and the section "Step 5 - Check Estimates". A message states: "You have estimated that this project will require:" followed by a bulleted list: "• 202 Actual Added & Modified Size" and "• 5000 Total Hours". Below this, a note says: "Your estimates for size and time translate into a planned productivity of 0.04 LOC/Hr. This is significantly less than your "To Date" productivity of 10.2 LOC/Hr (± 3.57). This is usually a warning flag, indicating that you have probably underestimated size and/or overestimated time." At the bottom of the window are two buttons: "Re-evaluate your estimates" and "Keep your (potentially incorrect) estimates".

If you reexamine your estimates and decide not to change them, you can close the PROBE Wizard by clicking on the second button.

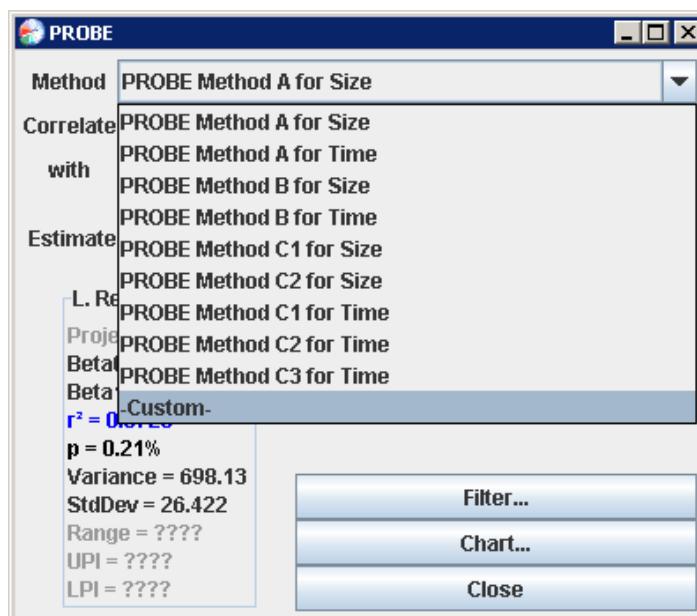
2.6. Using the PROBE tool

When following PSP's PROBE process (as defined in the book *A Self-Improvement Process for Software Engineers*), most people will want to use the [PROBE Wizard](#). It thoroughly automates the PROBE process and helps you to avoid common mistakes.

However, the PROBE Wizard is specifically designed for the correlations described in the PSP process scripts. If you thoroughly understand the linear regression calculations, the PROBE tool can offer more flexibility. The PROBE Tool will allow you to explore the correlations between any pair of metrics, not just the pairs described in the PROBE process. The PROBE tool can be opened by selecting "Tools → PROBE" from the [configuration menu](#).



If you wish to explore the correlations of the standard PROBE data pairs, the Method pull-down allows quick access to them. Clicking on the pull-down gives the following choices:

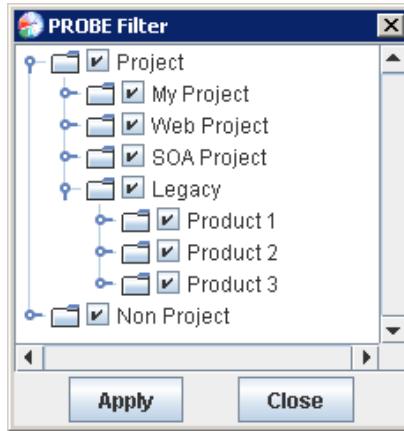


Choosing any of the standard PROBE methods will set the independent and dependent variables to their appropriate values. If you would like to explore correlations between any two other process metric variables, they can be chosen from the independent and dependent variable pull-downs.

Basically the PROBE tool runs a correlation test on the two sets of data chosen. It then reports the various statistical measurements on the dialog. There are two types of measurements shown on the dialog: those that show relationships between the sets of data chosen (r^2 , p, Variance, Standard Deviation), and those that are related to a projection based on the estimate entered (B_0 , B_1 , Range, UPI and LPI). Also there are two columns of data on the tool. The left column shows data based on a linear regression (this is for methods A and B the PROBE estimation method) and the right column shows data based on how well your data seem to predict on the average (this is for the different C methods of the PROBE estimation method).

2.6.1. Filtering your data set

The filter button allows you to restrict the data set of the correlation and projection to certain projects. It will pop up a tree view of the current projects stored and allow you to add or remove data items from the selected set.



Selected items are marked with a graphical "X". By default the tool only shows projects in the tree (nodes). You can expand this view to include the process phases as well by choosing the "Show leaf nodes" box. This will allow you to include or reject data from among different phases of projects. The Apply button applies your filtering selection to the numbers in the PROBE dialog.

2.6.2. Measures of correlation

On a typical estimation, you will want to choose from the different combinations of correlations available to see if any of the data correlate well enough to be used for projections. The PROBE tool will make this quite simple. Just pick the two data sets you would like to correlate, and the numbers on the tool will update to show you the results. The key numbers to examine are the measure of correlation (r^2) and the significance measurement (p).

2.6.2.1. The correlation measurement

As detailed by Watts Humphrey, an r^2 value of over 0.5 indicates that the numbers correlate well enough to use a linear regression method. The degree of confidence as measured by the r^2 variable is shown on the dialog by its color. When the value is too low to be used for predictive purposes, the r^2 value will show up in red. Values that show better and better agreement are shown in different colors. The colors used are: red (very bad), yellow/orange (fair), blue (good), and black (very good).

2.6.2.2. The significance measurement

The significance measurement is an indication of how significant the correlation is. It is expressed in a percentage that represents the probability that this correlation could have happened by chance. Low numbers mean that the data is probably showing a causality and can be used to make projections. The same color scheme to show the relative "goodness" of the data is used here.

Both measures need to be taken into consideration before using a projection from a linear correlation. There are some data combinations that can make one of the measurements look good while the other is bad. For example if you have data that correlate well but the significance of that correlation is not very high, you could be looking at a random occurrence.

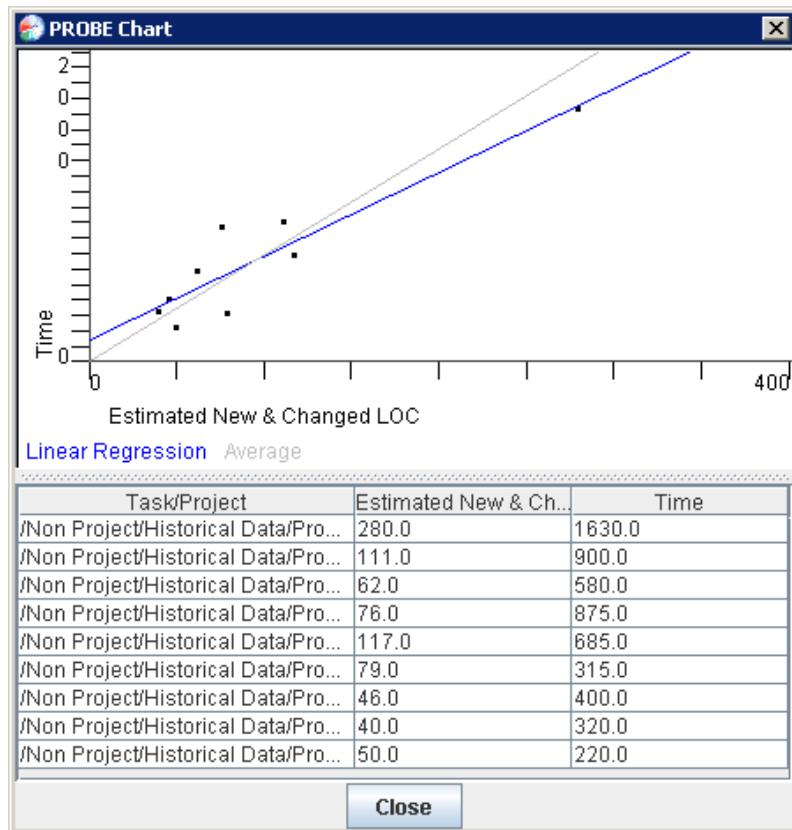
2.6.3. Making a projection

To make a projection, just enter an estimate in the Estimate entry. The estimate should be an estimate of the type of data that is chosen in the top pull down box marked "Correlate". For example, you should put an estimated number of New & Changed LOC in the entry if you are correlating against Estimated New & Changed LOC. The % range entry will allow you to change the confidence interval of the projection. Most PSP work is done at a 70% confidence interval, but you can set this as high or as low as you would like. Setting the confidence interval higher will give you a wider range of possible values, and setting the interval lower will narrow the range of possible values.

As you change the input parameters, the tool will recalculate all the numbers appropriately. Some calculations may display ERROR if the data is poorly correlated, or if there are not enough historical data points to perform the calculation. The projected estimate will show up at the top of both columns (the left column showing the projection using linear regression and the right column showing the projection based on an averaging technique). It is up to you to decide which numbers are appropriate for making the projection.

2.6.4. Visualizing the data

The PROBE tool also offers a scatterplot of your process data. This view is chosen by clicking on the "Chart" button.



The plot will show a scatterplot of the data being used for the regression calculations and two lines passing through the data. A blue line shows the line as projected from B_0 and B_1 calculated using linear regression, and a red line shows the line as projected from the averaging calculation. Only one of these lines will be in color if you've chosen one of the standard PROBE estimating methods. In this case, the other line is shown in grey. If you choose a custom estimation method (i.e. you are comparing different sets of data than what the PROBE method calls for) both lines will show in color. Many times this visual representation of the data can provide insights that straight numbers may not.

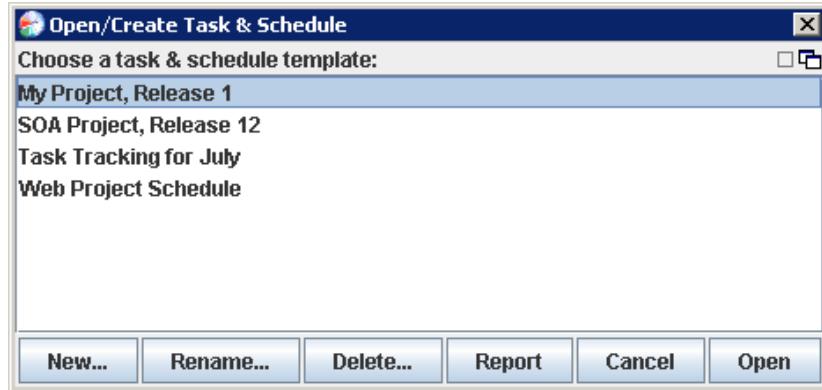
The three columns below the plot display the actual values used for the data plotted. The first column contains the project name, and the second and third columns show the data values for the X and Y axes respectively.

2.7. Using the Task & Schedule Tool

The task and schedule tool allows you to generate a schedule for completing tasks, and track progress against that schedule using earned value. You can open the task and schedule tool by selecting the "Task & Schedule" option on the [configure menu](#).

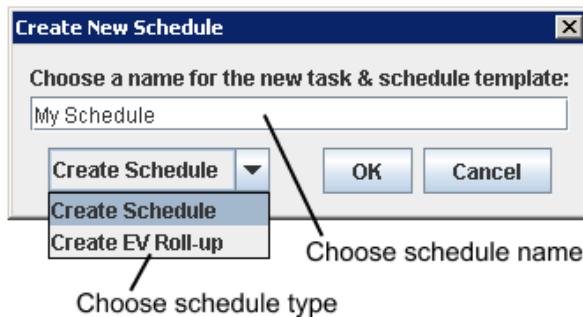
For more guidance in creating an earned value schedule, see the [Creating a schedule step by step](#) help topic.

If you haven't created any schedules, the dashboard will immediately jump to the [Create New Schedule](#) dialog which will allow you to create one. If you have already created at least one schedule, a Task & Schedule chooser window will appear.



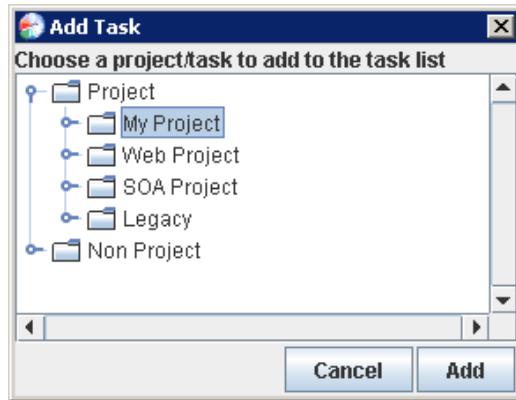
This window displays all the schedules you have created. After you click on a schedule to highlight it, you can use the buttons along the bottom to rename it, delete it, or open it for editing. The Report button will show you the detailed status for the schedule in your web browser.

Clicking the New button will create a new schedule. The dashboard will ask you to enter a name for the new schedule. This allows you to track different schedules from within the dashboard; each schedule must have a unique name.



In addition to choosing a schedule name, you can indicate whether you want to create a regular schedule (for tracking your own work), or an EV Roll-up (for combining data from several schedules into a single report). Most of the time, you will just want to create a regular schedule, so you can just keep the default selection for this item and click OK.

When a new schedule is first created, an empty Task & Schedule window will appear prompting you for the first task you want to add to the schedule.



All the tasks that are to be tracked within the schedule must come from the project hierarchy defined using the [Hierarchy Editor](#). You can track progress against PSP projects, projects you have made using custom process scripts or add-on process sets, or even against just a hierarchy of task names that are not associated with a particular process in the dashboard. The Task & Schedule tool is completely process independent.

The Task & Schedule window combines a project task list and a simple project schedule into one window. The top portion of the window gives a view of the task schedule on a task by task basis and tracks metrics about each task. The bottom portion of the window is concerned with actual calendar time. The time for the schedule is broken out into time periods (e.g. weeks) and the columns track metrics about each time period.

Tasks hierarchy

Task progress information

The screenshot shows a software interface for managing tasks and schedules. At the top, there's a menu bar with File, Edit, View, Tools, and Help. Below the menu is a tree view titled 'ProjectTask' showing a hierarchy of tasks. A specific node, 'Component A', is highlighted with a yellow background. The main area contains a large table with columns for PT (Planned Time), Time, PV (Performance Value), Plan Date, Replan, Forecast, Date, Notes, Dep, %C, %S, EV, and a small icon. The table rows represent various tasks and their details. At the bottom of the main window are buttons for Add Task..., Remove Task, Move Task Up, Move Task Down, and Flat View. Below this is another table for scheduling, with columns for From, To, PT, CPT, CPV, Time, CT, Notes, and EV. It lists several scheduled entries. At the bottom of this table are buttons for Add Schedule Row, Insert Schedule Row, and Delete Schedule Row. Finally, at the very bottom are buttons for Errors, Chart, Report, Cancel, and Save.

Task Schedule

Editable values appear with a yellow background. If you are unfamiliar with the various columns in the task and schedule planning templates, you can hover the pointer over the column headings and a tool tip will tell you what the abbreviated heading stands for.

If you edit the schedule or the list of tasks, you will need to click the Save button to save your changes. If you edit yellow-background values in the task list, however, those changes will be saved transparently; pressing the save button is not necessary.

2.7.1. Columns in the Task List

While this help topic won't get into all the details of earned value tracking, briefly here are the items that are shown in the upper window. (Note: you can customize the columns that are displayed by choosing "Select Task Columns" from the "View" menu.)

Planned Time (PT): This is the amount of time you have planned for the given task. Generally this will be displayed in hours and minutes. However, if this field is editable, you can enter a percentage instead of a duration to mark this task as a "[level of effort](#)" task.

Time This is the actual amount of time spent on a given task.

Planned Value (PV): This is the amount of earned value that will be earned by the given task based on the planned numbers. It is calculated as a percentage of the total planned work.

Plan Date: This is the date the task is planned to complete. The Task & Schedule tool computes this by assuming that the planned time for each task, the planned task order, and the planned hours per time period will be 100% accurate. Then, it determines when your schedule will have enough planned hours to complete the first task, the second task, and so on. (For more details on this calculation, [click here](#).)

Replan: No plan is perfect. When things don't go according to plan, it is helpful to chart a new course ahead based on the most up-to-date information. The dashboard looks at the tasks you've completed so far, the time you've spent on tasks in progress, and the time you have available in your schedule. Then, it recalculates planned dates for each task by assuming assumes that your plan was essentially correct, but has encountered a few "bumps" in the road.

As an oversimplification, you can think of this as calculating how far ahead or behind schedule you are, and adding that delta to the planned date for each task. In reality, the calculation is more intelligent, because it accounts for in-progress work, overspent tasks, and variations in the amount of time your schedule contains per period. (For more details on this calculation, [click here](#).)

Forecast: This is the date that the Task & Schedule tool thinks you might actually finish the given task. After you have completed at least one task in your plan and at least one time period in your schedule, the Task & Schedule tool will compare your actual performance to your planned performance. (For example, are tasks taking twice as long as expected? Are you spending as much time per period as you thought you would?) Then, it assumes that these plan-to-actual relationships will hold true for the rest of the tasks in your plan, and recomputes completion dates accordingly. (For more details on this calculation, [click here](#).)

Date: This is the actual completion date. It will be filled in automatically when you mark the task complete via the [Completion Button](#). Alternately, you can edit completion dates here (for example, if you need to back-date the completion of an item).

Notes: This column displays a note icon for any task that has an [associated note](#). You can place your mouse over the icon to read the note. To create a new note or edit an existing note, just click on a cell in this column.

Dep: This column displays information about [task dependencies](#). If a task has no dependencies, nothing will be displayed. If the task depends on at least one other task which has not yet been completed, a "stop sign" icon will be displayed, indicating that work cannot proceed until the predecessor has been completed. Once all predecessors have been marked complete, the icon will change to a green checkmark.

To edit the list of dependencies for a task, double-click on a cell in this column, and follow the instructions. Generally, this approach is unnecessary, because most projects will choose to define dependencies at the team level. In that scenario, the dependency information will be copied from the team plan automatically, and you won't need to edit the information here.

This column will also display an icon for "reverse dependencies." If other individuals are waiting for you to complete a task, this column will display an indicator to that effect.

Percent Complete (%C): This is the percent complete for the current task. Many people are accustomed to informally estimating percent complete; PSP formalizes this concept:

- If a task has no subtasks, it will either be 0% complete or 100% complete, based upon whether you have marked it complete via the [Completion Button](#).
- If a task has subtasks, its percent complete is calculated by using the planned times of those tasks to compute a weighted average.

At times you might wish for the ability to edit the "percent complete" for a task because it is "90% done." Unfortunately, this often leads to tasks that reach 90% complete and stay there for weeks, slipping the schedule. PSP states that instead, you should break the task down into subtasks and mark some of those subtasks complete. That way, you will get an accurate measure of percent complete.

Percent Spent (%S): This number describes the relationship between planned time and actual time for the given task. If you have spent twice as much time on this task as you planned to spend, it would be 200% spent. If you have only used up half of the planned time, the task would be 50% spent.

Earned Value (EV): This is the actual amount of earned value that this task has earned. *Remember that earned value measures are not counted until the task is marked as complete.*

Note: The columns listed above are sufficient for most simple planning. As a result, the task list will usually only display the columns listed above. For advanced planning, however, some additional information must also be displayed. Therefore, the following columns will automatically appear if:

- You manually prune subtasks out of your hierarchical task list
- You logged time to any of the tasks in your task list before the starting date of your schedule, and you have enabled the [iterative planning](#) schedule option.
- You completed any of the tasks in your task list before the starting date of your schedule, and you have enabled the [iterative planning](#) schedule option.

Planned Direct Time (PDT): The "Planned Time" column (described above) displays the total amount of time you plan to spend on a task, no matter when that time gets spent. In contrast, the "Planned Direct Time" column displays the amount of effort you plan to spend as part of this schedule.

An example can help illustrate the difference. Imagine that one of the tasks on your task list is to write a large software module. It might have subtasks for requirements, design, implementation, and formal testing. The total amount of time needed to complete all of these subtasks would be displayed in the "Planned Time" column. Now suppose you already completed requirements and design, and started on the implementation, as part of an earlier project iteration. Furthermore, you don't plan to begin formal testing until the next project iteration. The "Planned Direct Time" for requirements, design, and formal testing would be zero, since you don't intend to spend any time on them during the current project phase. Also, since you already performed some of the work for the implementation task, only the remaining portion of its planned time would appear in the "Planned Direct Time" column.

The "Planned Direct Time" column will only be displayed if it differs from the "Planned Time" column for at least one task in your task list. When it appears, you will note that the "plan value" for a task is based on its planned direct time, not its planned time.

Actual Direct Time (DTIME): The "Actual Time" column displays the total amount of time you have spent on a task to date, no matter when that time was spent. In contrast, the "Actual Direct Time" column displays the amount of effort you have spent as part of this schedule.

2.7.2. Columns in the Schedule List

Briefly, here are the items that are shown in the bottom window:

From: This is the starting date for a portion of your work schedule. The schedule can start anytime and can be of just about any length (one week in duration, one day, one month, etc.).

To: This is the ending date for a portion of your work schedule. Using the From and To fields, you can break your schedule up into manageable time chunks. The Task & Schedule tool defaults to using time periods of one week in length, but you can edit the To field to create time periods of any length (e.g. months, days, or even hours).

Planned Time (PT): This is the amount of time that will be applied to the tasks in the task list during the calendar time period selected by the From and To fields. It defaults to 20 hours, but can be set to whatever number you think is accurate for the amount of time you will actually spend on task *during the given time period*. Note that this is the total amount of time you plan on spending on *ALL* of the tasks listed in the top window. The default numbers on the Task & Schedule tool work out to 20 hours per week of work on task.

Cumulative Planned Direct Time (CPT): This is the Planned Direct Time of the given time period, plus the times of all previous periods.

Cumulative Planned Value (CPV): This shows cumulatively the amount of earned value that is planned for the given time period.

Actual Time (Time): This shows the actual amount of time spent on the tasks in the schedule during the given time period.

Cumulative Actual Direct Time (CT): This shows the actual amount of direct time spent on task during the given time period cumulatively.

Notes: This column allows you to attach an annotation to a time period in your schedule. These notes could be used, for example, to

record the reason why a week has a different number of planned hours, or the reason why actual hours differed from expectations. A note icon is displayed for any time period that has an annotation. You can place your mouse over the icon to read the note. To create a new note or edit an existing note, just click on a cell in this column.

Actual Cumulative Earned Value (EV): This shows the amount of earned value actually earned during the given time period cumulatively.

Note: The columns listed above are sufficient for most simple planning. As a result, the schedule list will usually only display the columns listed above. However, if your task list contains any "level of effort" tasks, then "time" and "direct time" will no longer be the same. As a result, the following columns will automatically appear:

Planned Direct Time (PDT): The first column, labeled "PT," will display the total amount of time you plan to spend during this period on *ALL* tasks in the task list; this column, however, will show how much time is available after subtracting time for "level of effort" tasks.

Percent Indirect Time (%I): This shows the percentage of your total actual time that you spent on "level of effort" tasks.

Actual Direct Time (DTime): The "Time" column will display the total amount of time you actually spent on *ALL* tasks in the task list. This column, however, will subtract out time spent on "level of effort" tasks to show how much time you spent on tasks that earn value.

2.7.3. Potential mistakes

Be careful to avoid making the following mistakes when creating a schedule:

Duplicate tasks: Nothing currently prevents you from adding the same task to your task list more than once. This could easily happen, for example, if you add a task to your task list, then add its parent as well. If you do this, the dashboard will dutifully create a schedule as if you were performing the task twice.

Conflicts between different schedules: The dashboard lets you create multiple, independent schedules. It does not, however, offer any support for time management when multiple schedules are running concurrently. Nothing prevents you from creating twelve different concurrent schedules, and promising to spend 40 hours per week on each. Also, nothing prevents you from adding one task to multiple different schedules.

Time estimates that don't add up: The Planned Time (PT) column should always add up in a hierarchical manner; that is, the planned time for a parent node should always be equal to the sum of the planned times for all its children. This is almost always the case for the PSP processes that come with the dashboard. However, if you have [unlocked](#) your data and overwritten calculations, times may no longer sum up correctly. This problem can also occur if you are using plain hierarchy "nodes" to describe tasks, and you subdivide a node after estimating its time. When this occurs, the task list will use the children's plan time for calculations rather than the parent's. To correct this error, do one of the following:

- Edit the planned times for the children, so they agree with the planned time for the parent.
- Edit the planned time for the parent, so it agrees with the planned time for the children.
- Highlight the planned time for the parent, and delete it.

Missing time estimates: Earned value forecasts and calculations make heavy use of the the planned times for the tasks in your task list. If you forget to estimate the planned time for a task in your task list, these forecasts will be underestimated.

The dashboard automatically checks for *some* of these errors, and displays an warning if they are found. If errors are present, they will be highlighted with red text. If you hover the mouse over the red text, a tool tip will provide a short description of the problem. For a more complete explanation of the problem, click the red "Errors" button that appears at the bottom of the task & schedule window.

2.7.4. Creating a schedule step by step

The Task & Schedule tool tries to make entering all the data necessary for project planning and tracking as painless as possible. It fills in many of the columns automatically. The following discussion will show how to create a schedule step by step.

2.7.4.1. Step 1: Add tasks to the task list

To create a schedule, you decide what tasks you want included in the schedule, and add them with the Add Task button. You can add any node in your hierarchy to the task list; it will be added, along with all its children, recursively. As an improvement over the traditional task planning template, the dashboard displays your task list in a hierarchical fashion. You can expand and contract this hierarchical view to get coarser- or finer-grained data.

The relationship between the Hierarchy Editor and the Task & Schedule tool

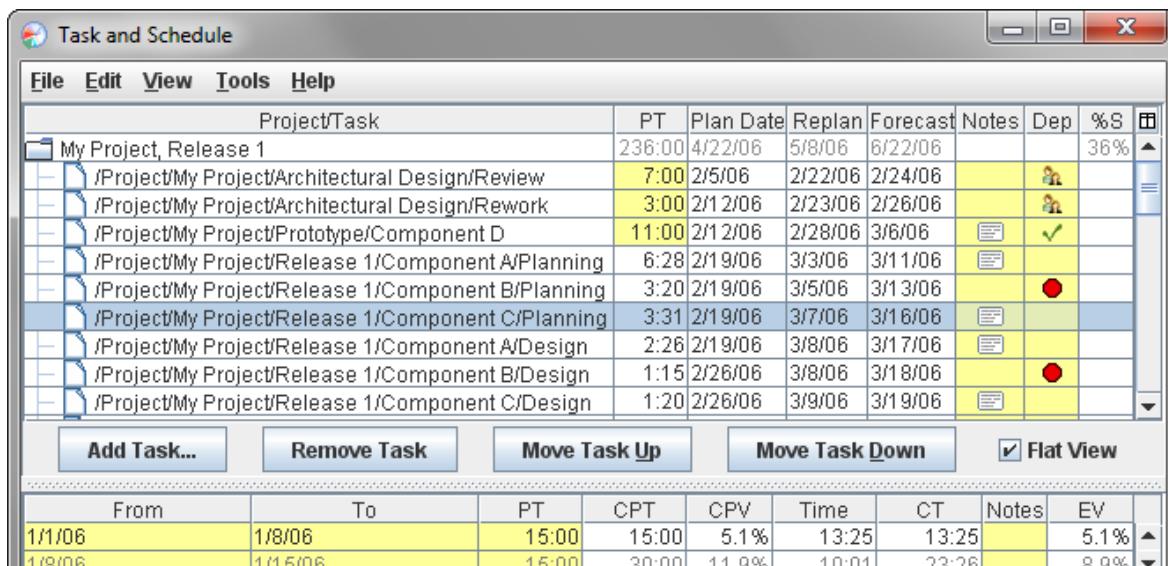
All the tasks that are to be tracked within the schedule must come from your project hierarchy, as defined in the [Hierarchy Editor](#). If you edit your hierarchy in the future, your changes will automatically propagate into affected task lists. For example, if you add a node named "Release 3.0" to a task list, then in the future if you were to create/delete subtasks underneath "Release 3.0", they would automatically appear in the task list / disappear from the task list.

Changes made in the hierarchy automatically get reflected into the schedule, because the schedule is basically a listing of the tasks/projects in the dashboard. However, if you remove a task in the Task & Schedule tool, it does *NOT* get deleted from the task hierarchy of the process dashboard (as defined using the [Hierarchy Editor](#)). All you will have done is removed the task from the schedule, not deleted it entirely from the dashboard. Deletion of tasks or projects from the dashboard itself must be done from the [Hierarchy Editor](#).

Editing the task list

Once you have added tasks to the task list, you can use the Remove Task button to remove them. If you remove a top-level task, it will be removed from the task list entirely. (Top-level tasks are the tasks that you explicitly added using the Add Task button; in the task list, they appear as immediate children of the root of the tree.) If you remove a child of a top-level task, it will be "pruned" out of the task list hierarchy, along with all of its children. You can highlight any pruned task (including children of a pruned task) and click the Restore Task button to add it back to the task list.

The planned completion dates displayed in the task list are based upon the order you plan to complete the tasks. It is unlikely that you plan to complete the tasks in the order they appear in the hierarchy. Therefore, there are two ways to indicate the planned order of tasks. First, you can change the relative order of top-level tasks by highlighting them and using the Move Task Up/Down buttons. For more fine-grained control, check the Flat View checkbox. This will replace the hierarchical view of the task list with a flat view of the "leaf" tasks that have not yet been completed.



After you switch into flat view, you can reorder the list of tasks. Place tasks in approximate chronological order (the first task in the list should be the one you plan to work on first, and so on). There are several ways to reorder the tasks:

- You can highlight a set of tasks, and click the Move Task Up/Down buttons.
- You can highlight a set of tasks, then drag them up and down to move them to a new place in the list.
- You can highlight a set of tasks, and type **ctrl-c** to copy the tasks to the clipboard. Next, highlight another task in the list to

indicate where you would like the tasks to appear. Finally, type **ctrl-v** to move the tasks to the highlighted position.

- For extensive reordering, type **ctrl-a** to select the entire list of tasks, then **ctrl-c** to copy the entire list to the clipboard. Next, open a word processor or spreadsheet program, and paste the list of tasks there. Use that application to move and reorder the tasks as desired, keeping one task per line. When you are finished, highlight the list of tasks in the other application, and copy them to the clipboard. Switch back to the Task and Schedule window and type **ctrl-v** to apply the reordering.

2.7.4.2. Step 2: Estimate task time

Once tasks are added to the task list, the dashboard will look up information like the planned time, actual time, etc. from the appropriate Project Plan Summary forms (if applicable), and display it in the appropriate boxes. You can click on any value with a yellow background and edit it, and the change will appear immediately on the associated Project Plan Summary form. Similarly, if you edit one of these values on a Project Plan Summary form, the new value will automatically appear here in the task list.

Level of Effort Tasks

Often, you may find that you are required to perform "overhead" tasks, like reading email or attending meetings. Or, you may be assigned to some ongoing responsibility like keeping the server running, or monitoring the quality of your team's software modules. These tasks are ongoing, and usually have no beginning and no end. It doesn't make sense to say that you could ever "complete" such a task. In general, you'll spend some percentage of your available time working on these tasks. Because of this, it is generally not possible to estimate how many hours and minutes of your time the task will require. If the project lasts one month, you might spend 10 hours during that time period reading email - but if the project lasts two months, you'll spend 20 hours. Because of this, traditional PSP earned value techniques tell you not to include these tasks on your task list at all.

The Process Dashboard includes an innovative feature that makes it easy to include these "level of effort" tasks on your task list. Add the task to your task list as usual, and in the "Planned Time" column, enter a percentage. For example, if you think you'll spend 5% of your time reading email, you can add a task to your task list called "reading email," and enter "5%" in the planned time column. If you indicate that a particular task is a level of effort task, all of its subtasks will be considered level of effort tasks as well.

"Level of effort" tasks in your task list will display differently from regular tasks. For example, the actual time for these tasks will be displayed as a percentage (so you can quickly learn how much time you really spend in meetings!). Level of effort tasks do not earn any value, since it doesn't make sense to mark them "complete." For the same reason, these tasks will not contain plan/actual completion dates or percent complete/spent values.

2.7.4.3. Step 3: Estimate calendar time

To generate a schedule, the dashboard needs to know how much time you plan to spend working on these tasks, and when you plan to spend it. This information is displayed in the bottom portion of the window. Each row in the table represents a time period. These time periods are consecutive; each time period begins when the previous time period ends.

Think about how much time you have available to work on the tasks listed in your task list. Describe a time period by entering the dates it will start and stop, and in the "PT" (Planned Time) column enter the number of hours you think you can spend during that time period.

Most of the time people have fairly predictable schedules - for example, you might say "I'm going to spend 20 hours a week on this until it gets done." Since this is the most common situation, the schedule will, by default, grow automatically to meet the demand. You don't have to manually add rows to your schedule until there are enough hours to cover the tasks - instead, the schedule template will automatically duplicate the final row in the table until there are enough hours. You can easily see where the automatic portion of the schedule starts because the automatic rows are displayed with a medium gray font. Thus, the black rows in the schedule template are rows that you have manually created/edited, and the gray rows are the ones which have been automatically added to meet the time requirements.

From	To	PT	CPT	CP
Feb 17, 2002	Feb 23, 2002	20:00	20:00	
Feb 24, 2002	Mar 2, 2002	20:00	40:00	
Mar 3, 2002	Mar 9, 2002	20:00	60:00	
Mar 10, 2002	Mar 16, 2002	20:00	80:00	11

You can disable this automatic schedule expansion by entering the word `END` in the planned time column for a particular row. This tells the dashboard that you plan to stop work on a particular date, whether the tasks are completed or not. The dashboard will take this end date into account for its calculations; if the schedule no longer has enough planned hours to finish all of the work, some tasks may be assigned the planned completion date of "never".

As in the task list, editable cells have a yellow background. Since time periods are contiguous, editing the "To" value on one row will cause the "From" value on the following row to change accordingly. Similarly, editing the "From" value on one row will cause the "To" value on the previous row to change. If you make changes to the last black row (which is used as a template and copied to make all the subsequent gray rows), all of the gray rows will be regenerated accordingly. If you edit a value in one of the gray rows, it (and all of the rows before it) will become "real", and the row in question will become the new "template row" upon which subsequent gray rows will be based.

If you need to extend the schedule past its optimal completion date, click the Add Row button to create a new row at the bottom of your schedule. The Insert Row button will split the currently selected time period into two separate rows, while the Delete Row will merge the currently selected time period with the one that follows it. Note that it isn't possible to delete an automatically generated row. If you want to force your schedule to end early, you should do so by entering `END` in the planned time column for the final time period, as described [above](#).

If you have any "level of effort" tasks in your task list, you'll notice that there are two planned time columns - one for total planned time, and one for planned "direct" time. Once you enter how much total time you plan to spend in a period, it will automatically subtract out time for the level of effort tasks, and display the remaining time in the "PDT" column. As your schedule progresses, you'll also be able to see how much time you're spending on these "indirect" tasks by looking in the "%I" column.

Iterative Planning

Many teams use an iterative approach to develop software. To create a schedule for the second or subsequent iteration of your project, you can take these steps:

1. Identify the start date of the current iteration, and enter it in the "From" column on the first schedule line.
2. In the Task & Schedule window, choose "Tools → Schedule Options..."
3. Select the appropriate option to "rezero" the charts and calculations concurrent with the start date of your schedule.

These changes will have the following effect on your EV schedule:

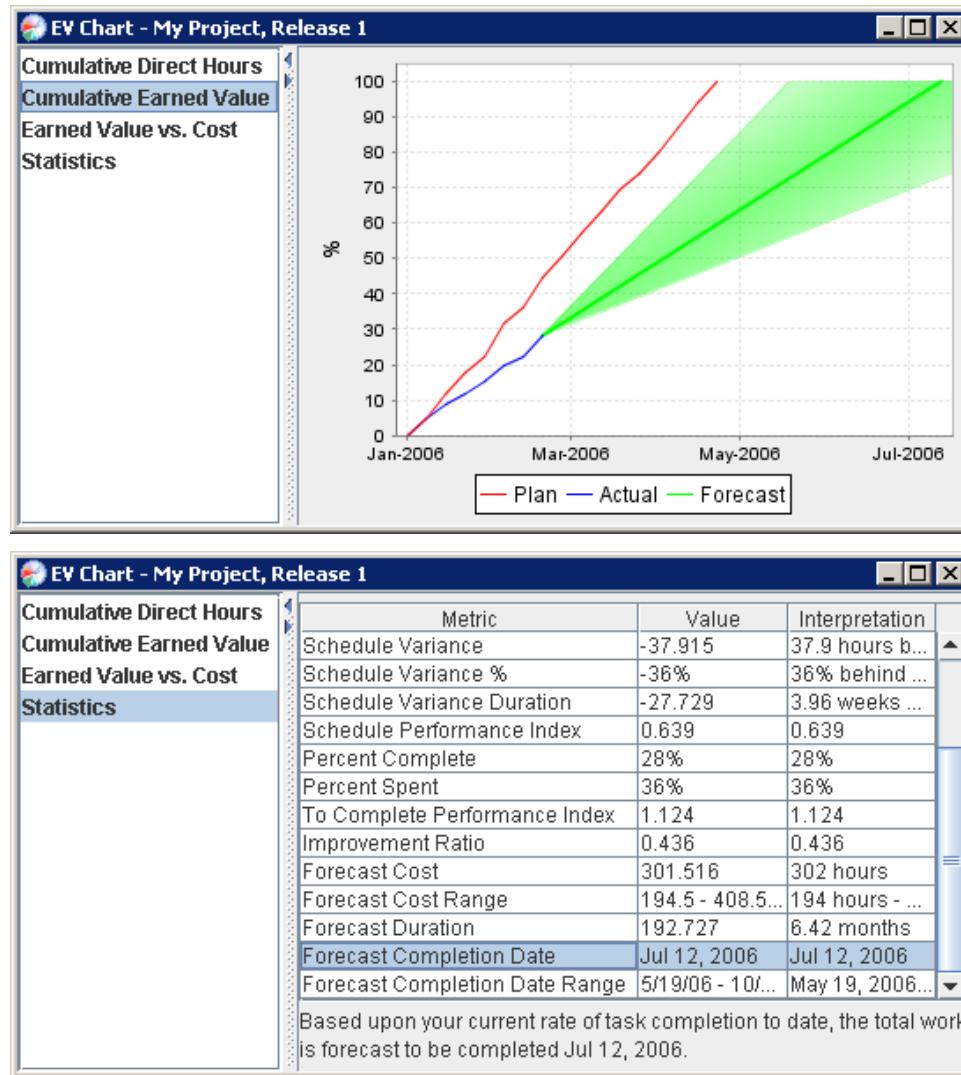
- Tasks that were marked complete before the start date of the schedule will be displayed in sepia tone, and flagged as "Previously completed tasks." They will not be included in the earned value calculations for this schedule.
- Tasks that were in progress when the iteration began will be treated specially. The dashboard will measure the amount of time that was spent on the task before this iteration began, and will subtract that amount from the total planned time for the task. The result is the "planned remaining time" - the amount of time you plan to spend on the task during this iteration. The earned value for the task for this iteration will be based upon its remaining time, rather than its total planned time. (The planned remaining time for the task will be displayed in the ["PDT" column](#) of the task list.)

2.7.4.4. Step 4: Track your progress

With the list of tasks and the calendar of available time in place, the dashboard will calculate planned completion dates for each task. As you work, you should:

- Accurately log time against tasks using the [Play/Pause button](#) and the [Time Log Editor](#). Time logged against tasks in your task list will appear in the Actual Time columns of the task list and schedule list.
- Mark tasks complete using the [Completion Checkbox](#). When tasks are marked complete, the completion date will appear in the task list, and you will earn the planned value for that task.

While your project is in progress, you can click on the "Chart" button for access to [earned value charts](#) and numerical analyses:



If you prefer, you can view this information in HTML, and even export it Excel, by clicking the Report button.

Sometimes, you may be interested in seeing these charts and statistics for a portion of your task list. The "Filtered Chart" button makes this possible. Here are some examples of its use:

- While working on a large project, you might like to see the status of a particular component. You would highlight that component in the task list and press the "Filtered Chart" button. Earned value charts and statistics would be displayed for that component only.
- While looking at the earned value plan for a team, it might be necessary to look at the schedule for a single individual - for example, to see when their work is forecast to complete. With the team schedule open in the Task & Schedule window, you could highlight the row corresponding to a particular individual, and click the filtered chart button.

2.7.5. Earned Value Date Calculations

Earned value is a powerful technique for planning and tracking your work. To get the most value out of the earned value tools in the dashboard, it is helpful to understand how the calculations are performed. The paragraphs below shed light on how various dates are calculated.

2.7.5.1. Plan Date

When you create a Task & Schedule plan to describe your personal work, the dashboard will automatically calculate the date when each task is planned to complete. This is a very simple calculation, and is specified in the PSP book by Watts Humphrey.

Specifically, the Process Dashboard begins by making a list of the "leaf" nodes in your task list. (A "leaf" node is a task that does not have any children underneath.) The dashboard looks up the amount of time that you have planned for each task. Then, it uses the [flat view](#) of your task list to determine the approximate order that you plan to perform the work. The resulting list might look something like this:

Task Name	Planned Time
Task A	1
Task B	5
Task C	3
Task D	8
Task E	4
Task F	7
Task G	2
Task H	5

Next, the dashboard will calculate a running total of the numbers in the Planned Time column:

Task Name	Planned Time	Cumulative Planned Time
Task A	1	1
Task B	5	6
Task C	3	9
Task D	8	17
Task E	4	21
Task F	7	28
Task G	2	30
Task H	5	35

In addition, your task and schedule plan will include a calendar of time that you plan to spend on the tasks in this task list. That calendar might look something like this:

Calendar Date	Planned Time
Week 1	6
Week 2	1
Week 3	6
Week 4	6
Week 5	0
Week 6	6
Week 7	6

Just as before, the dashboard will calculate a running total of the numbers in the Planned Time column of your schedule. If the schedule does not contain enough total hours to cover the tasks in your plan, the final row of the schedule will be repeated until enough hours are available:

Calendar Date	Planned Time	Cumulative Planned Time
Week 1	6	6

Week 2	1	7
Week 3	6	13
Week 4	6	19
Week 5	0	19
Week 6	6	25
Week 7	6	31
Week 8	6	37

Finally, the dashboard will cross-reference the numbers in these two "Cumulative Planned Time" columns. For example, if you were to work on the tasks in the exact order that they appear in the task list, you would finish Task A and Task B after completing 6 hours of work. If you spend exactly as much time per week as you plan to, you will finish 6 hours of work during Week 1. So Task A and Task B would be completed on Week 1. Task C would be completed when you reach the 9 hour point, and that would occur sometime in the middle of Week 3; so Task C would have a planned completion date of Week 3. If we were to finish these cross-references, we would obtain the following planned dates for each task:

Task Name	Planned Time	Cumulative Planned Time	Planned Date
Task A	1	1	Week 1
Task B	5	6	Week 1
Task C	3	9	Week 3
Task D	8	17	Week 4
Task E	4	21	Week 6
Task F	7	28	Week 7
Task G	2	30	Week 7
Task H	5	35	Week 8

Once these values have been calculated for the "leaf" tasks in your plan, the dashboard then takes a look at the other nodes in your hierarchically organized task list. Each parent node is planned to complete when its last child finishes. (The same technique is used to calculate the team rollup: each node in the team rollup will be planned to finish on the date when the last team member plans to finish that item.)

So that explains how the values in the "Plan Date" column are calculated. From this explanation, you can make the following important observations:

- The planned dates are an idealized notion of the date when each task could finish, if everything goes exactly according to plan.
- Calculations are based solely on the planned times for each task and each calendar period. The actual time you spend has no effect on the values in the Plan Date column. (For that, see the description of the [Replan](#) column below.)
- The Flat View ordering is a very important part of the calculation. If you want to obtain useful information for the dates when various tasks might complete, you must periodically visit the flat view and update the task order based on the work you plan to do next. (However, if you neglect to do this, the dashboard will still detect when you mark tasks complete, and will automatically rearrange the completed tasks to the top of your "flat view" ordering.)
- In your schedule, each calendar period has a start date and an end date (as specified in the "From" and "To" columns of the schedule). When calculating the Plan Date for each task, the dashboard simply reads the date in the "To" column of the schedule row when the task is projected to finish. On a typical project, this means that the Plan Dates will be rounded off to the end of the following week. Although useful information is discarded by this rounding process, it is unfortunately necessary to ensure that the planned dates remain faithful to the formulas specified by the PSP process.

2.7.5.2. Replan Date

As discussed above, the [Plan Date](#) column provides an idealized projection for when each task could be completed. But of course, work rarely goes exactly according to plan. So the Replan Date column produces an improved projection that takes into account the amount of time your schedule has slipped.

Specifically, the dashboard will look at your original list of tasks, and will estimate how much planned work is still remaining. To do this, it will make a temporary "working" copy of the original task list. If any tasks have already been marked complete, those completed tasks will be removed from the "working" copy. And if you have started logging time to other tasks in your plan, the actual time you've spent so far will be subtracted from the originally planned time, to calculate the amount of time that is remaining for each task:

Task Name	Planned Time	Actual Time	Completed?	Planned Time Remaining
Task A	1	2	Y	
Task B	5	7	Y	
Task C	3			3
Task D	8	3		5
Task E	4	1		3
Task F	7			7
Task G	2			2
Task H	5			5

Similarly, the dashboard will take a look at your original calendar of available time, and will determine how much planned time is still remaining. Just as before, this is accomplished by making a temporary "working" copy of the original schedule. Any calendar periods that have already passed are removed from the working copy. For the calendar period that is currently in progress, the planned time will be prorated based on the percentage of real time that has already elapsed. The table below shows an example of what the "working" calendar might look like if we are halfway through Week 4 of the original schedule:

Calendar Date	Planned Time	In the Past?	Planned Time Remaining
Week 1	6	Y	
Week 2	1	Y	
Week 3	6	Y	
Week 4	6	50%	3
Week 5	0		0
Week 6	6		6
Week 7	6		6
Week 8	6		6

After making a "working" list of the remaining tasks, and a "working" list of the time planned for future calendar periods, the dashboard now takes these two "working" lists and repeats the calculation that was originally performed to compute planned dates.

Based on this explanation, the following observations about "Replan Date" are worth noting:

- The Replan Date effectively calculates what the plan would look like if we were to stop at the current instant in time and create a new plan for the remaining work.
- The Replan column assumes that the originally planned times will hold true for all of the remaining tasks in the plan, and for all of the remaining periods in the schedule. As such, it is assuming that you had some unforeseen bumps in the road, but that the remaining work will proceed exactly according to the original plan.
- In a simplified sense, the Replan column can be envisioned as having measured the current schedule slip, and adding that difference to each planned date. But in reality the calculation is much more sophisticated, because it accounts for (a) tasks in progress, and (b) variations in the amount of time you plan to spend per calendar period.
- The replan calculation takes the current date and time into account when it calculates how much time is remaining in the current calendar period. As a result, the replan dates in your personal plan are being recalculated continually, every second of every day.
- If a task has been marked complete, the Replan column will show the actual date the task was completed.
- [As mentioned above](#), the Plan Date calculation is rounded off to the nearest week to maintain fidelity to the formulas in the original PSP process. However, since the Replan date is a calculation invented by the Process Dashboard, there is no need to discard precision in this manner. So when cross-referencing the cumulative planned times between the task list and the schedule list, the dashboard will extrapolate into the middle of a calendar period to calculate a more specific date. (Please note that this is a simple

linear extrapolation. It does not take working hours into account, so replan dates may fall on weekends or holidays. Like any other EV projection, these dates are only intended to facilitate tracking and coordination; it would be an error in planning judgement to interpret them literally.)

- Sometimes your team may hold project relaunch meetings where you perform a "replan." However, it is just a coincidence that this column has a similar name. The "Replan" column is **not** displaying a snapshot of data produced during your team relaunch. If you want to retain a snapshot of a team plan, you should save a baseline.

2.7.5.3. Forecast Date

As discussed above, the [Replan Date](#) column observes that your schedule has slipped somewhat, then calculates the dates each task might complete based on an assumption that the remaining work will proceed exactly according to plan. However, if your original plan contained an estimating bias, that assumption will be incorrect. So the Forecast column takes additional steps to adjust for estimating bias.

First, it looks at all of the tasks you have completed so far, and compares the total planned time for those tasks to the total actual time that was spent. It uses this ratio to compute your estimating bias for the tasks in this task list. Then, it assumes that the remaining tasks in your plan will also have this same bias.

For example, in the table shown below, Task A and Task B have been marked complete. Those two tasks had a combined planned time of 6 hours (1 + 5), and a combined actual time of 9 hours (2 + 7). 9 divided by 6 is 1.5, indicating that tasks are taking 50% longer than expected on average. So the dashboard would multiply all of the original planned times in this "working" task list by 1.5:

Task Name	Planned Time	Adjusted Plan Time	Actual Time	Completed?	Planned Time Remaining
Task A	1		2	Y	
Task B	5		7	Y	
Task C	3	4.5			4.5
Task D	8	12	3		9
Task E	4	6	1		5
Task F	7	10.5			10.5
Task G	2	3			3
Task H	5	7.5			7.5

Similarly, the dashboard would examine the portion of the schedule that is in the past. It would add up the total planned time and the total actual time for those past periods, and compute the ratio between these totals. That ratio would indicate whether you over- or underestimated the amount of time you would be able to spend each week. Then, it would make a "working" copy of the schedule, and adjust the planned times for all of the future periods by that ratio.

After having made these corrections to adjust for estimating bias in the task list and the schedule, the calculation would proceed in the same manner as the Replan calculation described above.

Based on this explanation, the following observations about "Forecast Date" are worth noting:

- If your plan was reasonably accurate, the Forecast date can be a powerful tool for automatically measuring and correcting the estimating biases in your plan.
- The forecast date examines completed tasks to calculate estimating bias. However, you may only complete a few tasks during the first week or two of a project. During this time, the dashboard will not have much data to use for its bias calculation, so the forecast date may swing wildly each time you mark a task complete. As a result, the Forecast date should be viewed with caution until you have collected a few weeks worth of data. During those first few weeks of a project, the Replan date may be more useful for tracking purposes.
- The forecast calculation is based upon an assumption that future work will have the same estimating bias as completed work. If you have reason to doubt that assumption, use the Forecast date with caution. For example, if your original plan was extremely poor, and there is no good relationship between the planned and actual times of the tasks in your plan, then the forecast date may not be meaningful.
- Since the forecast date looks at completed work to measure the estimating bias, a forecast date cannot be calculated until you have marked at least one task complete.

- At the team level, the Forecast Date is computed by identifying the chronologically latest forecast date for all of the individual members of the team. As a result, if any single team member does not have a forecast date (because they have not yet marked a task complete), then the team rollup will not have a forecast date either.
- To allow for the most accurate calculation of estimating bias, you should strive to log your task time faithfully. For example, if you routinely forget to log time against your tasks, and then just mark tasks complete with zero time logged, the bias calculation and resulting forecast date will be inaccurate.

2.7.5.4. Optimized Forecast Date

The discussions above describe how dates are calculated in the earned value plan for a single individual. Then, the dashboard rolls these dates up to the team level by identifying the date when all team members will finish. As a result, when you see the green "Forecast" line on a team earned value chart, it is depicting the date when the **last** team member is projected to finish.

Of course, a high-maturity team will monitor the forecast dates for each team member to see whether there is a large discrepancy between the dates when the first and last team members will finish. When a large discrepancy is present, the workload is said to be unbalanced. In that scenario, it is helpful to identify the date when the team could finish if the workload were optimally rebalanced.

The dashboard calculates this Optimized Forecast Date in the following way:

- The Forecast algorithm above is used to create a bias-adjusted "working" task list, and a bias-adjusted "working" schedule for each team member.
- Then, a consolidated task list is created, and the dashboard identifies the number of hours of remaining work that is forecast for the entire team.
- Additionally, the "working" schedules are summed up to produce a single consolidated schedule, showing the number of hours that the entire team is forecast to have available during each week in the future.
- Finally, the remaining hours in the task list are cross-referenced against the forecast hours in the consolidated team schedule. The resulting "optimized forecast date" is the date that the work would finish if the work was balanced perfectly.

This is a very sophisticated calculation. In particular, it is performing independent bias adjustments for each individual team member, based on the actual task and schedule data they have collected so far. And it is taking into account variations in staffing levels - for example, the dates when individual team members will be out of the office, and dates when particular individuals [will leave the team project](#).

The Optimized Forecast Date is displayed on the earned value charts as a yellow line. When the yellow line and the green line are far apart, this is a warning sign that the workload is unbalanced.

2.7.6. Earned Value Charts

When you track your work using the [Task & Schedule tool](#), the Process Dashboard can analyze your data and produce many helpful charts to help you track your progress.

When you have the Task & Schedule window open, you can view these charts by clicking the Charts button. In addition, you can view these charts by clicking the "More Charts" hyperlink on the earned value report.

2.7.6.1. Cumulative Direct Time

This chart shows whether you have been able to devote as much time to your work as you originally expected.

The horizontal axis depicts the passage of time. The lines on the chart convey various measures of direct time:

- As you log time against the tasks in this task list, the dashboard will calculate a running total of the number of hours you have spent, and will display that running total as the blue "**Actual**" line on this chart.
- The red "**Plan**" line will display a running total of the number of hours you planned to spend. (This information comes from the schedule table, which appears on the lower half of the Task and Schedule window.)
- After you collect time and mark at least one task complete, the dashboard may be able to calculate an adjusted estimate of the number of hours that the work will actually require. This projected cost will be displayed on the chart as a green "**Forecast**" line.

If the dashboard is able to compute a 70% prediction interval for that forecast, the interval will be displayed on the chart as a green cone around the forecast line.

Interpretation

You can compare the Plan and Actual lines to determine whether you are devoting as much time to the work as you originally expected. If the Actual line is below the Plan line, you are not getting in enough hours; this could potentially cause you to fall behind schedule. If the Actual line is above the Plan line, you are getting in more hours than you planned.

Remember that this chart only displays the time that you have logged to the tasks in [this task list](#). It does not display all of the time you spend in the office. Accordingly, if your actual time is too low, you may be able to remedy the problem by eliminating workplace distractions and finding ways to devote more of your time to these tasks.

2.7.6.2. Cumulative Earned Value

This chart displays your planned and actual progress at a glance.

The horizontal axis depicts the passage of time, and the vertical axis displays percent complete. The lines on the chart convey various measures of progress:

- The red "**Plan**" line shows your planned progress, beginning at 0% and reaching 100% on the date when you could finish the work if everything were to go according to plan.
- The blue "**Actual**" line shows your actual progress so far. In the PSP earned value technique, you do not receive credit for partially completed tasks; you only receive credit for tasks that have been marked complete. So each time you mark a task complete, the blue line will rise to indicate that you are closer to your goal of completing the project. When you finish all of the work in this task list, the blue line will reach 100%.
- Once you begin collecting data, the dashboard may be able to calculate an adjusted date when the work is most likely to finish, assuming that you make progress in the future at a rate similar to the past. This will be displayed on the chart as a green "**Forecast**" line. Sometimes, the dashboard is also able to [calculate a range](#) of dates that place an upper and lower bound on the completion date forecast with a 70% error margin. If so, a green cone will depict this range of dates.
- When you roll up data for a team, the dashboard will also calculate an "**Optimized**" completion date. This is the date that the project would be forecast to complete, if the workload were rebalanced perfectly so everyone finishes at the same time.
- If you have saved a baseline for this schedule, a brown "**Baseline**" line will appear. This shows what the "Plan" line looked like at the moment you saved the baseline.

Interpretation

The simplest and most useful way to use this chart, is to compare the Actual line to the Plan. If the Actual line is above the plan line, you are ahead of schedule. If it is below the plan line, you are behind schedule.

When you are behind schedule, it could be due to the fact that you are unable to spend as many direct hours as you planned. If so, the [Cumulative Direct Time](#) chart will make this problem clear. Otherwise, you could be behind because tasks are taking longer than expected. The [Earned Value vs. Cost](#) chart will highlight that problem.

Next, the Forecast line can be a helpful tool for strategic planning. But be wary about using the Forecast line during the first few weeks of your schedule. When limited actual data is available, the forecast date may be unstable, changing dramatically each time a task is marked complete. Once you have collected a few weeks' worth of data, the Forecast line will be much more useful. If the forecast line is consistently predicting a date that will not allow you to meet your commitments, you may need to escalate the issue.

On a team rollup, lines will be drawn for the "Forecast" and the "Optimized Forecast." The optimized line shows the date work might complete if the workload is rebalanced perfectly. In contrast, the Forecast line will be displaying the date when the last person is projected to finish, assuming that no tasks are reassigned. When the Optimized line and the Forecast line are far apart, this is an indication that the team workload is unbalanced, and tasks may need to be reassigned.

A forecast line cannot be drawn until you have logged some time against the schedule and marked at least one task complete. When you are looking at a team rollup, the forecast is computed by finding the date when the last team member will finish. If the schedule for any

team member does not have a forecast, then no forecast can be computed for the team either. So if your team rollup doesn't have a forecast line, check to see whether some of your team members are still 0% complete. The easiest way to do this is to open the Task & Schedule window for the team schedule.

Normally, the Plan line will extend to reach 100%. However, if you have set an "[end date](#)" for your schedule, and there is not enough time in your plan to finish all of the work, the Plan line will not reach 100%. In this case, the dashboard's forecast calculation may also decide that the work will never complete, and no forecast line will be drawn on the chart. In these cases, you may need to escalate the issue.

When a Baseline is present, the Baseline line usually will **not** end at the 100% mark. For example, if the planned cost of your project has grown since the baseline was saved, the planned line will reach higher than the baseline. Thus, you can compare the Baseline to the Plan to visualize Baseline Growth.

2.7.6.3. Earned Value vs. Cost

This chart displays planned and actual progress and cost at a glance.

The horizontal axis depicts the passage of time. The lines on the chart convey various measures of progress and cost:

- The red "**Plan Value**" line shows your planned progress. This line corresponds exactly to the Plan line in the [Cumulative Earned Value](#) chart; the only difference is that the vertical axis of this chart is labeled in hours instead of % complete. Thus, the Plan line will start at zero, and end at the total number of hours you plan to spend on the tasks in this task list.
- The blue "**Actual Value**" line shows your actual progress. Once again, this line corresponds exactly to the Actual line in the [Cumulative Earned Value](#) chart. In PSP, the value of a task is determined by its planned time; so each time you mark a task complete, the blue line will rise by the number of hours that you planned to spend on that task.
- The green "**Actual Cost**" line shows the actual time that has been spent on tasks that have been marked complete. In other words, each time you mark a task complete, the green line will rise by the number of hours that you actually spent on that task.

Interpretation

Comparing the red and blue lines will tell you whether you are ahead or behind schedule. This may seem obvious, because these two lines display the exact same data as the Plan and Actual lines on the [Cumulative Earned Value](#) chart (just with a different vertical scale).

Comparing the blue and green lines will tell you whether tasks are taking more or less time than expected. If the green line is above the blue line, tasks are taking more time than planned. If the green line is below the blue line, tasks are taking less time than planned.

You can compare the red line to the green line to get a rough idea of whether you are devoting as much time to the project as planned. However, this comparison will only be an approximation, because the lines on this chart do not include data from incomplete tasks. To accurately compare planned and actual time spent, view the [Cumulative Direct Time](#) chart.

This chart may be very familiar to individuals that have experience with traditional earned value management systems. The red "Plan Value" line corresponds to BCWS (Budgeted Cost for Work Scheduled). The blue "Actual Value" line corresponds to BCWP (Budgeted Cost for Work Performed). And the green "Actual Cost" line corresponds to ACWP (Actual Cost for Work Performed).

2.7.7. Tracking Task Dependencies

Creating a schedule with the [Task and Schedule](#) tool, and tracking your progress with earned value, is an excellent way to stay on track and deliver your work on time. But when you are working as part of a team, it is also important to coordinate with your team members. This is especially true if your team member needs to use something that you are producing (or vice versa). To facilitate this type of coordination, the dashboard will allow you to define and track task dependencies. *Note: if you are not working as a member of a team, you probably don't need to declare task dependencies.*

Task dependencies are a way of stating, "*this* task cannot start until *those* tasks are finished." You can define dependencies between tasks that appear in earned value schedules. Once defined, the dashboard will help you to track the status of these dependencies, as described below. However, the presence of dependencies does not currently alter the earned value calculations in any way. In particular, **the dashboard does not use these dependencies to perform any sort of critical path analysis**. If you use "Flat View" to specify task order, the dashboard does not prevent you from reordering tasks before their predecessors. These capabilities may be added in the

future. But at the current time, task dependencies are simply a way of showing you the information you need in order to coordinate with your teammates more effectively.

There are two different ways you can define task dependencies:

- When your team is planning a team project, you can define dependencies in the Work Breakdown Structure Editor.
- When you view an earned value schedule in the [Task and Schedule](#) window, you can define dependencies by double-clicking on a cell in the "Dep" column.

Most teams will use the first approach almost exclusively.

Once you've defined a task dependency, the dashboard will display icons to indicate the presence and status of dependent tasks. If a task has a dependency, one of the following icons will be displayed:

?? This depends on at least one other task, but status information for that task is not available.

● This depends on at least one other task that still has not been completed.

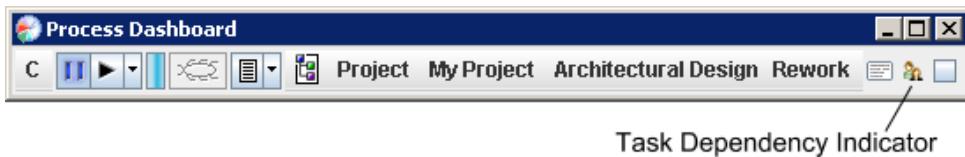
▲ This depends on at least one other task that still has not been completed - and unfortunately, it does not look like the dependent task(s) will be ready as soon as you need them to be.*

✓ All of the dependencies for this task have been completed.

⚡ Other people are waiting for you to complete this task.

! Other people are waiting for you to complete this task - and unfortunately, it does not look like you will complete this task in time to meet their needs.*

These icons are displayed in the "Dep" column of the task list when it appears in the Task and Schedule window, the earned value report, or the earned value weekly view. In addition, the icon will be displayed on the main dashboard window when you select a task that has a related dependency:



You can place your mouse over these icons for detailed status information. (When you see a dependency icon in your web browser, you'll need to click as well.) The detailed status will show you all the dependent tasks (if there is more than one), the completion status of each, the names of the individuals assigned to the task, the projected** completion date, and the task's percentage complete.

2.7.7.1. Misordered Dependencies

As described above, task dependencies are a way of stating, "*this* task cannot start until *those* tasks are finished." Accordingly, when task dependencies are present, it becomes necessary to work on tasks in a particular order.

However, the dashboard does not perform any critical path analysis on your task dependencies, and it does not attempt to compute the optimal order in which tasks should be performed. Thus, you must collaborate with your teammates to determine the best task order.

Once you decide which tasks you will be working on first, second, etc, you should use the [flat view](#) of your task list to record this order in the dashboard. The dashboard will then use this order to compute projected completion dates for each task.

Once all team members have performed this step, the dashboard will compare the projected** task dates to see if any dependent tasks have been placed out of order. For example, imagine the following scenario:

- Person A is working on Task A, and is projected to complete it by June 1.
- Person B is working on Task B, and is projected to complete it by May 1.
- Task B cannot start until Task A is complete

Obviously, the tasks are misordered! In this case, Person A would see a [yellow warning sign](#) next to Task A, with a message "Other individuals need this task earlier than it is projected to finish." Person B would see a [red warning sign](#) next to Task B, with a message "This dependent task is not projected to finish in time."

When you see these warning icons, it is a sign that you need to coordinate with your team members to resolve the problem! It is possible that the other individual just hasn't bothered to record the task order in their [flat view](#) yet. Or, it could be an indication of a serious scheduling problem. Either way, a conversation with your team member is in order. If your team uses task dependencies, it is best for each team member to use flat view faithfully so the dashboard can provide the most useful information for team collaboration.

2.7.7.2. Dependency Dates

By default, the dashboard uses the calculated "Replan" date as the "Projected" date for dependency tracking purposes. This means:

- When you request details for a dependency, the dates displayed are replan dates.
- When two tasks are compared for misordering, the comparison is made between the replan dates for the two tasks.

The "Replan" date is a good choice for dependency tracking purposes because it takes actual data into account, but is not typically as volatile as the forecast date. If you prefer to use "Plan" or "Forecast" dates for dependency tracking purposes instead, specify a value for the `ev.dependencies.compareDates` setting in the [preferences editor](#).

2.7.8. Prediction Ranges in the Task & Schedule Tool

The task and schedule tool has always been capable of calculating metrics that help you determine whether your project is on cost and under budget. The three metrics that are the most useful for this purpose are:

- **Forecast Cost:** Estimates the hours of effort your total project will require.
- **Forecast Completion Date:** Estimates the date you will most likely finish the project, if you continue to make progress in a manner similar to the work you have completed so far.
- **Optimized Forecast Completion Date:** This forecast only appears when you are rolling up schedules from more than one person. It estimates the date you could most likely finish the project, if future progress is similar to past progress, and if you are able to optimally rebalance the overall workload.

In general, these metrics are calculated using standard earned value formulas that can be found in project management textbooks. For more details, see the help topic on [earned value date calculations](#).

But the dashboard also contains a powerful new feature: it can calculate prediction ranges for these forecast values. **This is an innovation developed by the Process Dashboard team.** Since these calculations are new and innovative, the formulas and algorithms are not widely known. Discriminating users such as yourself will want to understand how these values are calculated before trusting/accepting them, so the algorithms are described below.

One important point is definitely worth noting. In the book, *A Discipline for Software Engineering*, Watts Humphrey describes methods for calculating prediction intervals that attempt to be 100% "statistically valid." Unfortunately, the methods described by Watts tend to break down when applied to the data that appears in an earned value schedule, so different statistical approaches must be applied instead.

Some of the algorithms used by the dashboard to calculate earned value ranges require a non-statistical approximation to be made as part of the calculations. This approximation may slightly increase the percentage error of the calculated range. In general, the variance introduced by this approximation is much smaller than the variances that are present in a typical software engineering project, so the approximation is mathematically acceptable. Also, the approximation will cause the dashboard to err on the side of producing more conservative ("wider") forecast ranges. But the bottom line is that you should not treat the calculated ranges as being "statistically valid;" instead, you should view them as being "engineering approximations."

And as always, any forecast or prediction based on historical data should only be trusted if the future work is similar to the completed work. If you have reason to believe that the estimates made for future tasks are fundamentally different than the estimates made for tasks that have been completed, you should use the resulting forecasts with extreme caution, if at all.

Forecast Cost Range

Traditional earned value calculations generate forecast total cost based on the assumption that the cost performance index (CPI) for completed tasks is a predictor of the CPI for the remaining tasks. To calculate the forecast cost range for a schedule, the dashboard simply analyzes how the completed tasks have varied around the current CPI, and uses that information to estimate the potential variability of the CPI for the remaining tasks.

Specifically, the dashboard examines the relationship between "Planned Direct Time" and "Actual Direct Time" for all of the tasks in your task list that have been **completed**. A scatterplot of these points is created, and a line is fit to these points. The sum-squared variance of the points from the line is then used to calculate a prediction interval based upon a students-T distribution with $n-2$ degrees of freedom (where n is the number of completed tasks). Finally, the double-sided interval with probability 70% is used to calculate upper and lower ranges for CPI. This CPI range is applied to the original estimates for the tasks in the schedule that have not yet been completed, and the actual times for completed tasks are added back in to generate the upper and lower boundaries of a range that approximates 70% probability for total cost.

To generate the forecast cost range for a rolled-up/team schedule, the dashboard uses a monte-carlo simulation with several hundred trials, and sums up the forecast costs of each subschedule for each trial.

Forecast Completion Date Range

To calculate accurate range estimates for the forecast completion date, it is necessary to consider variations in both cost **and** schedule. Possible variations in cost are adequately described by the probability density function of the cost range calculated above. To characterize schedule variations, the dashboard analyzes the completed time periods in the schedule. For each period, the planned direct time is divided by the actual direct time to generate "direct time performance index" (DTPI), a metric which is a mathematically related to the "time estimating error" for the period. The dashboard then calculates a weighted log-normal mean and standard deviation of these ratios. A confidence interval for this log-normal distribution is calculated using Angus's parametric bootstrap method, providing a probability density function for DTPI.

With these two probability density functions at hand, a monte-carlo simulation is once again used to characterize the range of the forecast completion date. In each trial, a total cost is randomly generated from the total cost probability density function, and a DTPI is randomly generated from the DTPI probability density function. A hypothetical earned value schedule is created by copying the existing earned value schedule. The planned direct times for each historical period in this trial schedule are set to the actual direct time spent in the period, and the planned direct times for future periods are divided by the random sample for DTPI. Normal earned value extrapolation techniques are then used to estimate when this randomized hypothetical schedule would reach the given random sample for total cost. The resulting date is stored as a single sample for random forecast completion date; two thousand trials are conducted to characterize the density function for forecast completion date, and this density function is used to generate the upper and lower boundaries of a range that approximates 70% probability.

To generate forecast completion date ranges for rolled-up/team schedules, monte-carlo simulations are used once again. In each trial, each subschedule is randomly seeded using the approach described above, yielding a random forecast completion date for each subschedule. In the dashboard, "Forecast Completion Date" is the date the schedule is projected to complete if no rebalancing is performed, so the chronologically latest of these dates in a given trial becomes a single random sample for forecast team completion date. "Optimized Forecast Completion Date" instead takes each of the randomized hypothetical subschedules and rolls them together to create a single team schedule. This randomized hypothetical team schedule is then extrapolated to determine when the random total forecast cost would be reached; the resulting extrapolation becomes a single random sample for optimized forecast team completion date. Several hundred trials are performed to characterize the probability distribution functions for "Forecast Completion Date" and "Optimized Forecast Completion Date," and these density functions are used to generate the upper and lower boundaries of ranges that approximate 70% probability.

2.7.8.1. When are Prediction Ranges Calculated?

By default, the dashboard can only calculate prediction ranges if:

- You have completed at least three tasks in your task list, and
- You have completed at least three periods in your schedule.

When calculating a rolled-up team schedule, each subschedule will need to meet these criteria before prediction ranges can be calculated for the team schedule.

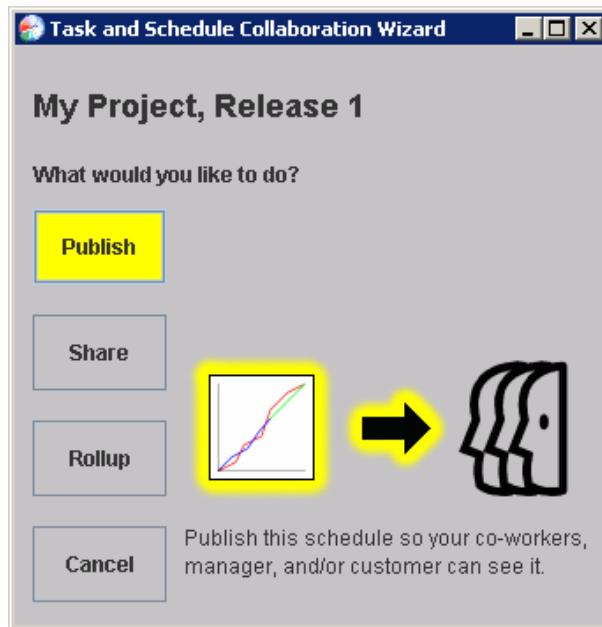
After calculating a prediction range, the dashboard applies several heuristic tests to determine its validity. If any of these tests indicates that the prediction range is not viable, it will be discarded and no range will be displayed on the chart.

Therefore, if ranges are not displayed for your schedule, it does not indicate an error - it just indicates that the data in your schedule can not be used to calculate a meaningful prediction range.

In this case, you can choose to calculate ranges anyway based on historical dashboard data. Choose Tools → Schedule Options, then put a check mark next to the option for historical data. (You can edit this option at the personal level, the team level, or both.) The dashboard will use the historical data you choose to calculate forecast ranges for cost and schedule. These calculations can be performed even before you begin work on the tasks in this schedule! When this schedule contains sufficient data, the calculations will switch to prefer the current data over the historical data.

2.7.9. Task & Schedule Collaboration

The [task and schedule tool](#) contains built-in support for collaboration; just choose Tools → Collaborate. The "Task and Schedule Collaboration Wizard" will appear.



The task & schedule tool supports two different types of collaboration. It allows you to [publish](#) a schedule (so other people can see it), and it allows you to [roll-up](#) individual schedules to create a team schedule.

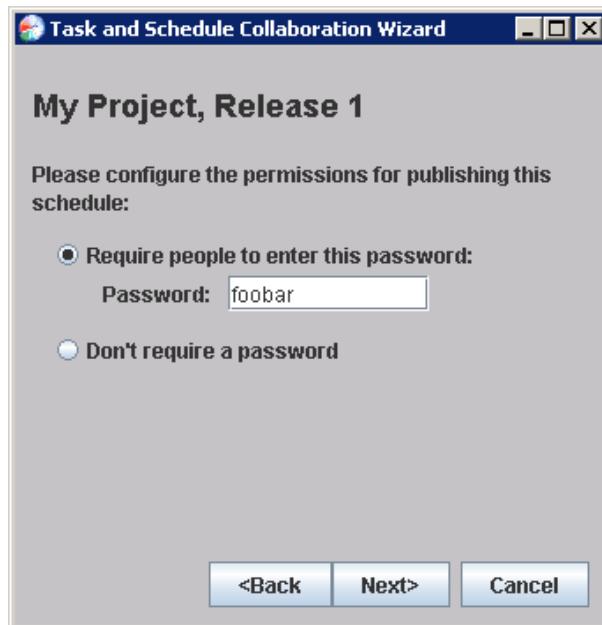
Note: These collaboration features will typically only work within a local area network. In addition, they only work when the dashboard is running. See the [limitations](#) section below for more information.

2.7.9.1. Publishing Schedules

Publishing a schedule opens up permissions so other people can view the HTML earned value report. This is a great way to:

- Let co-workers who depend on your work know when they can expect it to be completed.
- Keep your manager or your customer apprised of your progress.

To publish your schedule, open it and choose Tools → Collaborate. Then, click the Publish button. The dashboard will give you an opportunity to set a password. Setting a password prevents other people from viewing this schedule unless they know the password. If this isn't a concern to you, you can choose not to require a password. Passwords are set on a schedule-by-schedule basis, so you can set different passwords for different schedules, or set the password on one schedule but not another.



The dashboard will save your password settings as soon as you click the Next button. It will then display a final screen containing the URL, username, and password of the published schedule. People will need this information to view the published schedule, so you will need to send this information to them. A hyperlink is provided that will help you to write an email containing this information.

If you forget the URL or the password in the future, you can walk through this process again. Reopen the schedule, choose Tools → Collaborate, click the Publish button, and you'll see the password screen containing the configuration you set earlier. You can edit the password settings if you wish, and your changes will be saved - or you can just click Next without changing the password settings. You will once again be given the final screen with the URL/password, and the hyperlink you can use to email that information.

Note: people will only be able to view the published schedule while you have the dashboard running. For more information, see the [limitations](#) section below.

2.7.9.2. Rolling Up Schedules

When two or more people are working on a project together, it is very helpful to roll up each person's individual schedule to create a team schedule. Creating a team schedule has many benefits:

- It helps you to **track team progress**. You can easily determine whether the project is ahead or behind schedule, and over or under budget.
- It helps you to **forecast** the date when the overall project will most likely complete, as well as the total cost of the finished project.
- It helps to **balance the project workload**. Despite your best planning efforts, you usually find that someone on the team has too many tasks assigned to them. Reassigning tasks so that everyone is forecast to finish at the same time is called "balancing the schedule."

Rolling up schedules is a two-step process: first, each individual must "[share](#)" their schedule. Then, someone must create a master "[roll-up](#)" schedule.

Sharing Schedules

To share your schedule, open it and choose Tools → Collaborate. Then, click the Share button. The dashboard will give you an opportunity to set a sharing password. Setting a password prevents other people from adding this schedule to a master schedule unless they know the password. If this isn't a concern to you, you can choose not to require a password. Passwords are set on a schedule-by-schedule basis, so you can set different passwords for different schedules, or set the password on one schedule but not another.

The dashboard will save your password settings as soon as you click the Next button. It will then display a final screen containing the password and the URL of the shared schedule. You need to send this information to the person who is creating the master "roll-up" schedule. A hyperlink is provided that will help you to email the information to that person.

If you forget the URL or the password in the future, you can walk through this process again. Reopen the schedule, choose Tools → Collaborate, click the Share button, and you'll see the password screen containing the configuration you set earlier. You can edit the password settings if you wish, and your changes will be saved - but realize that changing the password will break any roll-ups that contain this schedule. To avoid that problem, just click Next without changing the password settings. You will once again be given the final screen with the URL/password, and the hyperlink you can use to email that information.

Creating a Roll-up Schedule

One person on the team should create a master schedule which roll-ups all of the individual schedules to the team level. Since that person usually has their own individual schedule as well, the dashboard is designed to create the rollup schedule and add their individual schedule all in one step.

To create the roll-up schedule, open your personal schedule and choose Tools → Collaborate. Then, click the Rollup button. The dashboard will allow you to choose a name for the resulting roll-up schedule. When you click the Next button, the dashboard will create the new roll-up schedule and add your personal schedule to it. When you click the Finish button, the dashboard will open the new roll-up schedule it just created.

Now, you will want to add additional schedules to the roll-up. Just click the Add Schedule button to display the Add Schedule dialog box. This dialog lists all of your personal schedules - you can add one or more of them if you wish. However, you will generally just want to select the "Import a shared schedule..." option. It will prompt you for the URL and password of the shared schedule you wish to add. This is information you need to get from your co-workers - hopefully they emailed it to you as described [above](#). Copy the information from the email into the fields provided, and click OK. At this point, the dashboard will connect to your co-worker's dashboard and retrieve the shared schedule. If it cannot retrieve the shared schedule, you should:

- Contact your co-worker to make certain that they have the dashboard running.
- Double-check that they sent you the right URL and password, and that they haven't changed the password since they sent it to you.
- Try adding the shared schedule again. If these steps don't solve the problem, you may need to take a look at the [limitations](#) section below.

After retrieving the shared schedule, the dashboard will give you an opportunity to designate a "local name" for it. By default, it uses the name of the remote schedule, followed by the name of the person who created the schedule. (This allows you to tell schedules apart, since they might otherwise have very similar names.) The shared schedule will appear in the master schedule by the "local name" you designate.

If you add a schedule by mistake, you can remove it by highlighting it and clicking the Remove Schedule button.

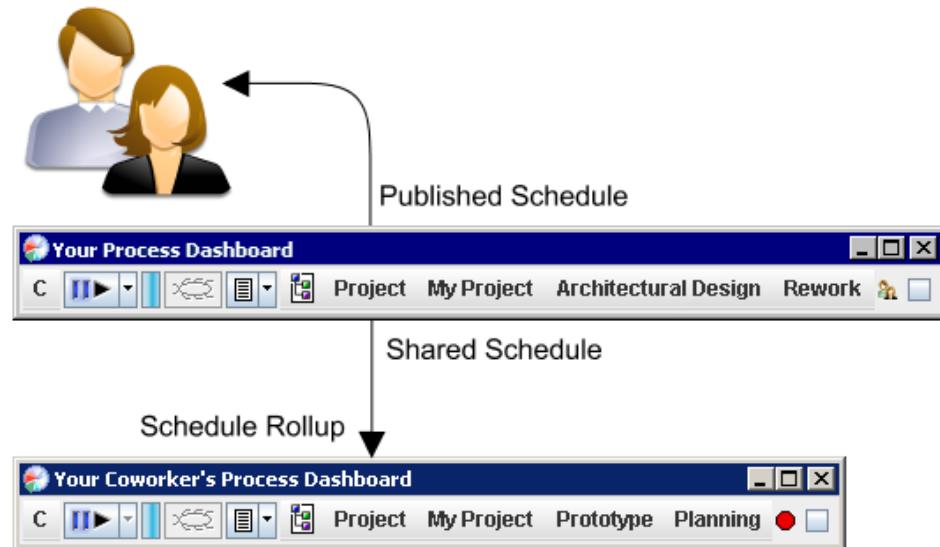
- **Tip:** if someone changes the password on their shared schedule, you can correct the problem by removing that schedule from the master schedule, and adding it back again with the new password.

The Move Schedule Up and Move Schedule Down buttons can be used to rearrange the list of schedules. Note that this will not change any of the roll-up calculations, since these calculations are independent of the order of the list.

Your co-workers will probably want to be able to see the roll-up schedule you just created. Simply open the roll-up schedule, choose Tools → Collaborate, and click Publish. See the section [above](#) on publishing for additional instructions. It is not currently possible to add one rollup schedule to another rollup schedule, so the Share and Rollup buttons will be disabled when you open the "Task and Schedule Collaboration Wizard" from a rollup schedule.

2.7.9.3. Constraints and Limitations

To understand the constraints of these collaboration features, it is helpful to understand how the features work.



When you publish a schedule, the dashboard acts as a web server that your co-workers can connect to with their web browser. Their web browser communicates to your dashboard to see the published schedule.

Similarly, when you share a schedule, the dashboard acts as a server that your co-worker's dashboard can connect to. The dashboard that is computing the roll-up schedule connects to your dashboard and retrieves the latest copy of your schedule in a peer-to-peer fashion.

As a result, your dashboard must be running for the collaboration features to work. If your dashboard is not running when a co-worker tries to look at your published schedule, they will instead see an error page stating "Cannot find server or DNS Error."

Rollup schedules keep a locally cached copy of your schedule, to enable calculating a rollup even when your dashboard isn't running. The rollup schedule will attempt to retrieve a recent version of your schedule. If your dashboard is not running, it will calculate the rollup using the most recent version it has cached.

The collaboration features rely on TCP/IP connectivity and DNS resolution to work. Because of this, you will probably need to be on the same network as your co-worker, manager, and/or customer. If someone is on a different network than you (especially if there is a firewall between your computer and theirs), they may not be able to take advantage of any collaboration features you have enabled.

Most of the time, you interact with the dashboard web server by viewing web pages that begin with "http://localhost". This works fine when the dashboard and the web browser are both running on the same computer. For the collaboration features to work, however, your co-workers need a web address that begins with the DNS name of your computer. Therefore, the dashboard automatically ascertains the name of your computer and uses that to construct collaboration URLs. This process should work 99% of the time. If, however, the dashboard deduces an incorrect name for your computer, then no one will be able to use the collaboration URLs you send them. If this occurs, you should ask your system administrator for the DNS name of your computer, and use it to set the "http.hostname" setting in the [preferences editor](#).

2.8. Data Freezing

By default, the dashboard will "freeze" appropriate pieces of data when the corresponding project phase is marked as completed. The two points where this happens are at the end of the Planning phase and at the end of the project.

In PSP process levels 0.1 and above, when you are planning your project the dashboard will use your historical data to help you create an estimate of how your total time will be distributed across the process phases. Once you click the completion checkbox to complete the planning phase, all your Planned data will be baselined. The estimates you entered for size and time, along with all the calculated computed data elements for estimated data, will be frozen so that your planned data will always stay constant (if you really want to know more about why this is necessary, read the [gory details](#) section).

Similarly, when you mark the project complete, the To Date numbers will be brought up to date and frozen. This will enable you to view this report again in the future without having the To Date numbers changing in the report.

One side effect of the freezing of data is that if you want to change your time or size estimate, you will have to navigate back to the Planning phase and mark it as incomplete by clicking the completion checkbox again. Once it has been marked as incomplete the dashboard will allow you to enter a different estimate.

The same is true of the To Date items on the report. If you navigate back to an old project and start logging time to it, the Actual columns will update as required, but the To Date columns will not. You will need to mark the project as incomplete so that the To Date items will thaw out and recalculate.

The Project Completion checkbox is located on the Project Plan Summary form.

A screenshot of a web-based form titled 'Project Plan Summary'. The form contains several input fields: 'Project Owner' (Sample), 'Start Date' (Jun 25, 2001 09:42:27 AM), 'Completion Date' (Jan 29, 2002 05:59:03 PM), and 'Keywords'. Below these fields is a checkbox labeled 'Completed:'. A callout arrow points from the text 'Project completion checkbox' to the right edge of the 'Completed:' checkbox.

The gory details

The "To Date" elements on the forms are set up to always recalculate themselves based on all the completed data in the dashboard's database. This is fine for most circumstances. However, if you ever want to go back and look at a Project Plan Summary form (or other report or chart) from an older completed project, the dashboard would recalculate those To Date elements all over again. This time the calculation would include data that was not available at the time the project was originally finished. This would preclude anyone from being able to reproduce a previous Project Plan Summary accurately.

Actually, it gets a bit worse. Since the estimates of time spent in each of the process phases is based on the To Date data, all the estimates would be changing on those older forms as well.

In fact, even within a project once you had planned your estimates for time in the various phases, those planned times would immediately change on the Project Plan Summary at the end of the project when you completed it. If the Planned data were still live, the current project's data would be incorporated back into those estimates!

The way we chose to deal with this problem is to introduce the freezing of data elements. Now, once your planning is finished the data will freeze and not be "corrupted" by the finishing of the current project. Also it becomes possible to recreate older Project Plan Summary forms since their formulas won't be changing anymore.

3. Running the dashboard

Both the PSP and the TSP require the collection and analysis of metrics at a *very* fine-grained level. Once data are collected at this level, statistical analyses of the data permit remarkable planning, tracking, prediction, and control of software products and projects.

These metrics collection and analysis processes, however, are not trivial. In any real-world project, tool support for the PSP and TSP become important considerations. Although studies have demonstrated that people can maintain their productivity when using the PSP without tool support, the "frustration factor" inherent with such an approach tries the patience of all but the most disciplined engineers, making PSP behaviors difficult to sustain.

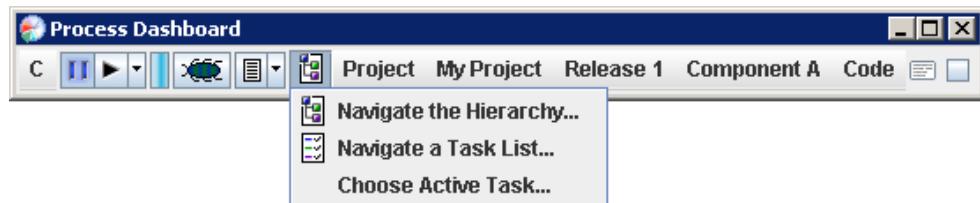
The Process Dashboard was designed to facilitate a solution to this problem. When using the dashboard, the metrics required by the PSP can be measured quickly, easily, and accurately.

3.1. Selecting the active task

As you work, the dashboard maintains a concept of the "currently active task." Items such as the play/pause button, the defect button, the script button and the completion checkbox operate against the active task. In turn, the hierarchy menus allow you to see at a glance what task is currently active, and select a different active task from within your dashboard hierarchy. The image below shows the dashboard ready to work on the Code phase of "Component A," which is part of "Release 1" in "My Project".

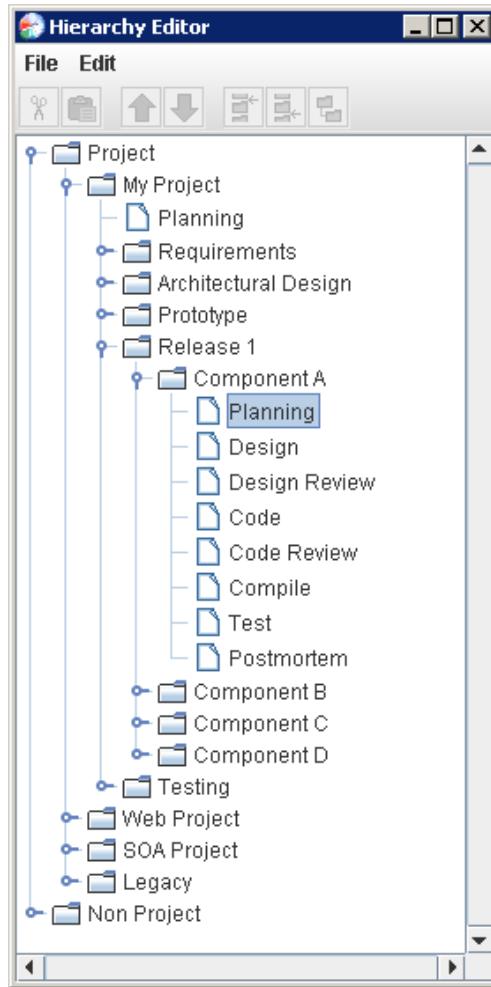


For convenience, there are two different navigation styles you can use to select the currently active task. The navigation style selector allows you to choose between these two styles.



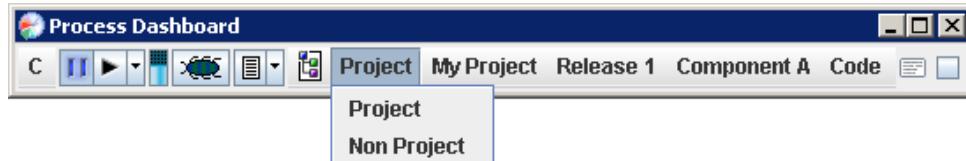
3.1.1. Navigating Tasks Hierarchically

By default, the dashboard uses the hierarchical navigation style. To better illustrate how this hierarchical navigation style works, the following example set of projects will be used. (The project tree is shown in a screenshot from the hierarchy editor. For more information on putting together a project hierarchy and the use of the hierarchy editor see the [Using the hierarchy editor](#) help topic.)



This project hierarchy shows two top level project categories: "Project" and "Non Project". These are the two categories given as default when the dashboard is first run. Under "Project" there are four items defined: "My Project", "Web Project", "SOA Project", and "Legacy". Under "My Project", several items are defined, including "Release 1". "Release 1" has four subtasks, called "Component A" through "Component D". In this screenshot, only "Component A" has been expanded to show its phases. Phases are added into the project hierarchy as children of a project. "Component A" was created from a PSP2.1 template, and so it has all the development phases of a PSP2.1 project.

Back on the dashboard itself, when the first hierarchy menu is selected, it will show all the entries in the project hierarchy that are defined at that level. In this case it shows the two default nodes "Project" and "Non Project".

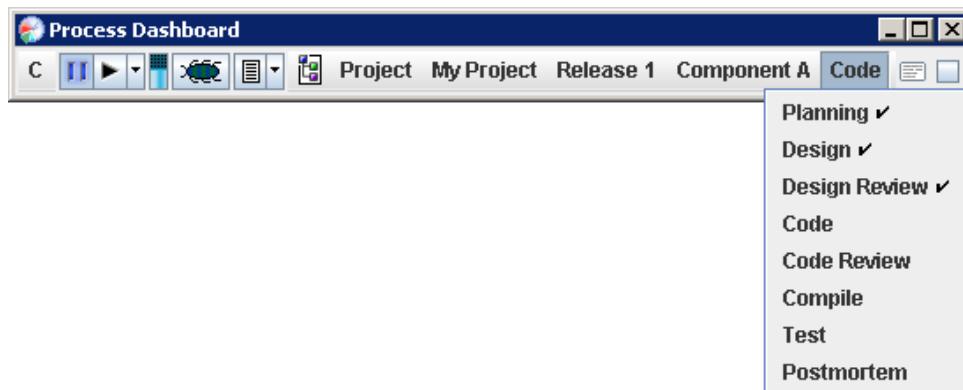


Selecting the second hierarchy menu shows the entries defined at that level under the selection from the first menu. In this case, the user has selected "Project" on the first menu, so the second menu shows the children of "Project" in the project hierarchy.



Selecting any menu in the set of hierarchy menus will show the entries defined as children under the selection on the menu to its left. This

final example shows the PSP 2.1 process phases that are defined for "Component A".



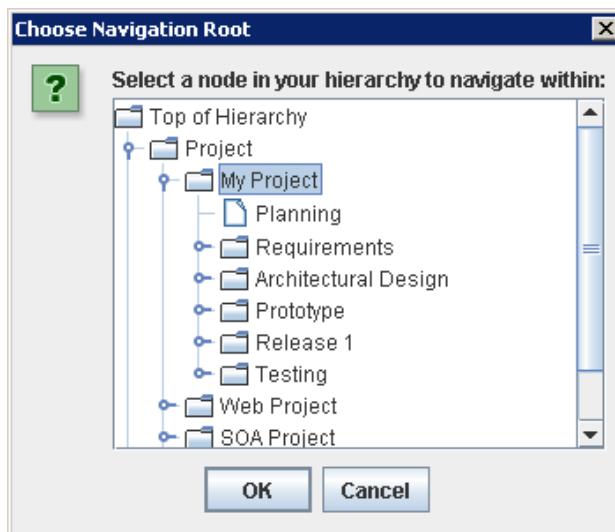
To help you keep track which tasks have marked complete, the hierarchy menus display small checkmarks next to completed items. (You can mark an item complete with the [completion checkbox](#).)

The dashboard window will resize itself as you navigate from task to task, keeping itself as small as possible while still displaying the names of all the nodes for the current task. If you find that the dashboard is still taking up too much horizontal space, you can resize the window to be smaller. The hierarchy menus will display as much of the names as possible. (If you hover over a truncated name, a tool tip will tell you the full name.)



If you've done this by accident, and you want to see the full names again, just resize the dashboard window to be bigger.

Another way to let the dashboard use less space is to navigate within a subset of your project hierarchy. If you click on the task navigation style selector and choose "Navigate the Hierarchy...", the following window will be displayed:



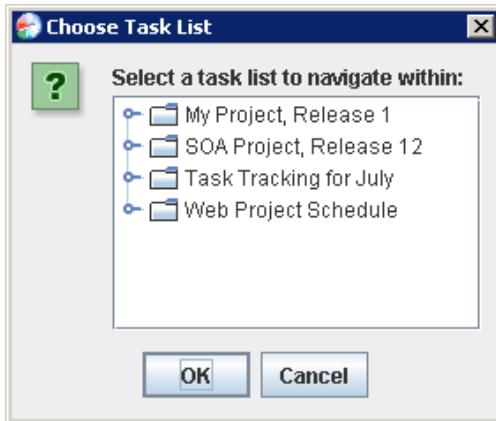
By default, the dashboard will be navigating within your entire project task hierarchy. Choosing "Top of Hierarchy" and clicking OK will request that default behavior. On the other hand, if you know that you will be working within "My Project" for an extended period of time, you could select that hierarchy node and click OK. Then, the dashboard would hide the first two hierarchy menus, displaying only the tasks that fall underneath "My Project".



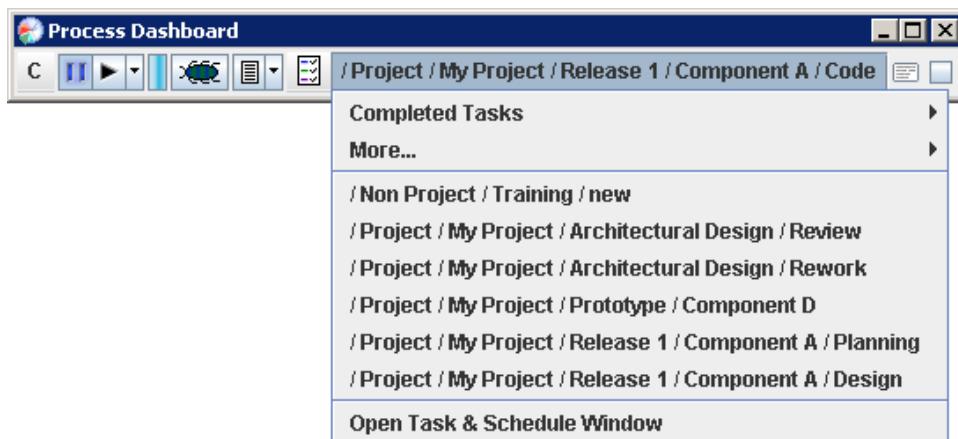
When you have narrowed your view of the hierarchy this way, you can hold your mouse over the task navigation style selector, and it will tell you the name of the hierarchy node you are navigating within.

3.1.2. Navigating Within a Task List

If you are actively working on a project, you may find that you spend most of your time working on tasks that are in a particular earned value task list. A second task navigation style makes this easy. Just click on the task navigation style selector and choose "Navigate a Task List...". A window will ask you for the name of the task list that you are working on.



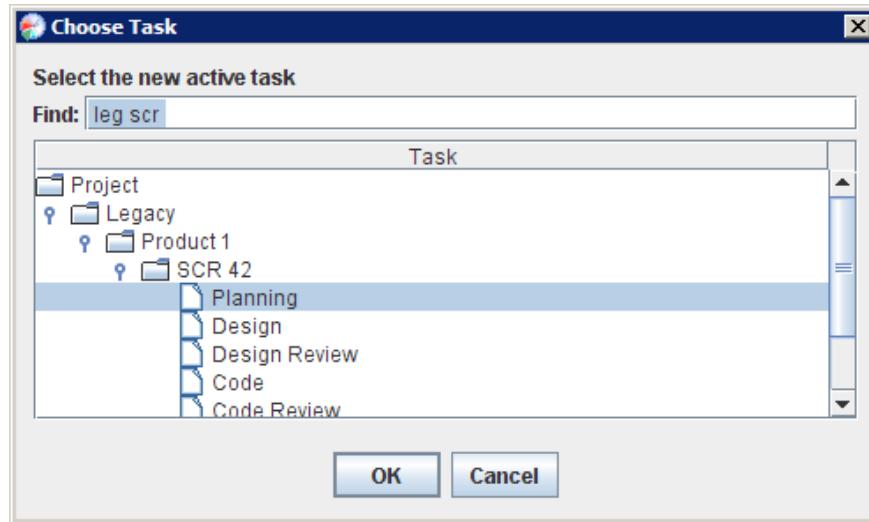
Select the task list that you are working on, and click OK. (If you know you will only be working on a subset of a particular task list, you can drill down into the task list and select the portion you plan to work on.) The dashboard will replace the multiple hierarchy menus with a single menu, which lists the tasks in your task list in chronological order (as defined in the [flat view](#) of the earned value schedule).



For more information on creating earned value schedules, and using the flat view of their task lists, see the [Creating a schedule step by step](#) help topic.

3.1.3. Selecting a Task Directly

The task navigation style selector contains a third menu item which allows you to quickly jump to a specific task. (Note: this feature is only available if you are running version 1.6 or higher of the Java Runtime Environment.) Selecting this option will display a window of tasks, either from the hierarchy or from the current task list (depending on the navigation style you've selected).

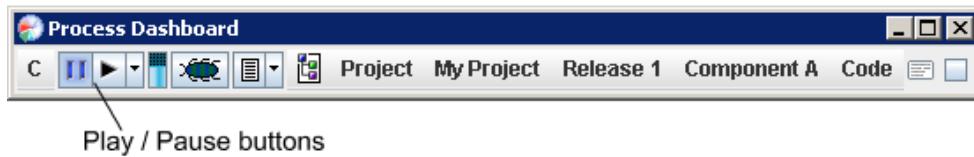


As you type in the "Find" box, the list of selections will automatically be narrowed to show tasks containing all of the letters you've typed. Capitalization is not important, and you can enter multiple filters separated by spaces. In the example above, the user has typed "leg scr". The first filter "leg" has narrowed the tree to show tasks under the "Legacy" portion of the hierarchy. The second filter "scr" has further narrowed the list to show Tasks with the letters "scr" in the name. (The end result is that "leg scr" has narrowed the list to show tasks associated with legacy software change requests.)

You can press the up and down arrow keys to select a specific task, and press the Enter key to change to the highlighted task. In addition, you can click directly on a task to activate it.

3.2. Using the play/pause button

The play/pause button gives access to the timer that is built into the dashboard.



The dashboard timer makes it easy to measure the amount of time spent in the various phases of your development process. When you start the timer, it will automatically make appropriate entries in your [time log](#) for the project/task which you have made active with the [hierarchy menus](#).

When the timer is stopped, the paused icon will have a soft blue glow:



When the timer is running, the play icon will have a soft green glow:



It is **not** necessary to stop the timer before using the hierarchy menus to change the active project/task/phase. You can leave the timer in "play" mode as you move from one phase to another, and it will properly finalize one entry in the time log, and open another.

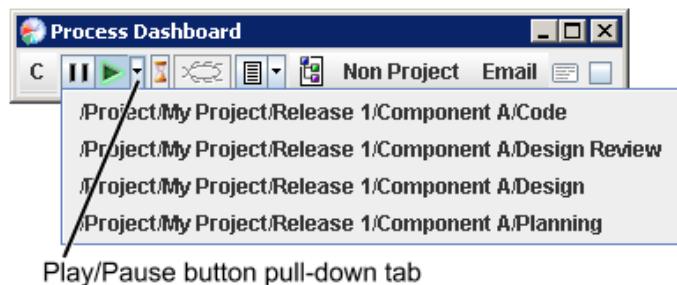
When the timer is running and you first navigate to an activity, you will notice that the dashboard will not create an entry in the time log until you have spent a full minute there; this is by design. Navigating from one task to another may require multiple interactions with the hierarchy menus, which necessarily implies that the "active" project/task will momentarily pass through several intermediate states before finally arriving at the desired task. You would **not** want the dashboard to create extraneous entries in the time log for all those intermediate tasks; therefore, the dashboard waits and creates the time log entry only after you have logged 60 seconds worth of elapsed time to an activity.

When you start the timer, it will make a short "clicking" sound to let you know it is timing you. You can leave the timer running as you move from phase to phase and project to project, and it will play the "timing sound" as you navigate to each new project/phase. Once again, the sound is there to remind you that the dashboard is timing you; the sound will **not** play if the timer is not running. Therefore, as you move from one project/phase to another, if you expect to hear the sound and do not (or vice versa), check the state of the timer.

When you first begin using the dashboard, it may be difficult to remember to start and stop the dashboard timer. A timing reminder feature is available to help; see the [system tray icon](#) help topic for more information.

3.2.1. The play/pause button pull-down control

The play/pause button contains a pull-down control that can be used to quickly jump to a point on the project tree where you have recently recorded time. This provides another way of navigating to a project or project phase in which you were recently working.



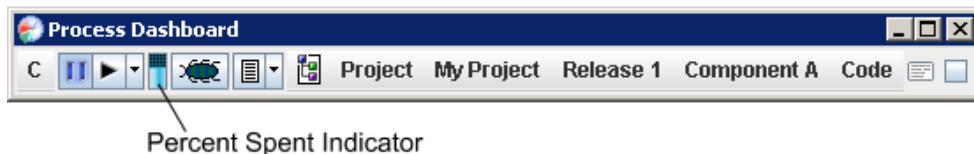
In this example, the individual had recently been working through the phases in a PSP2.1 project, when they were interrupted by an incoming email message. After finishing with the email message, they could select one of the entries in this pull-down list. The dashboard would navigate to the selected task and begin logging time there.

As this example illustrates, the items in the pull-down are not limited to the phases of the current project. Therefore, the pull-down control is very useful when you are juggling several activities. If you keep bouncing back and forth between several unrelated tasks, you can quickly jump to a recent activity and log time there by using the pull-down menu.

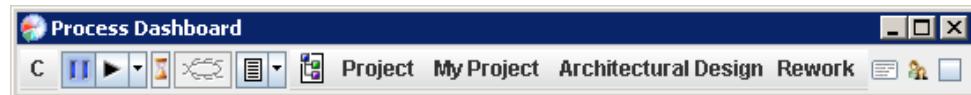
If you accidentally leave the timer running and begin logging time to a task that has been [marked complete](#), a warning message will be displayed. The only purpose of this warning is to give you a heads-up in case you are logging time by accident. If you do not wish to receive these alerts, you can indicate that preference as you dismiss the dialog.

3.3. The Percent Spent Indicator

The percent spent indicator gives you continual feedback about the amount of time you have spent on the current task, and how that compares to the amount of time you planned to spend.



If you have not estimated the amount of time you plan to spend on a task, the indicator will display an hourglass. If you would like to enter an estimate for the current task, click on the hourglass.



Once you have estimated the amount of time you plan to spend on a task, the indicator will begin giving visual feedback. When you first begin working on a task, the indicator will show a full blue bar:



As you begin logging time, the indicator will slowly deplete, like the battery charge indicator on a handheld electronic device. The size of the blue bar shows you what percentage of the budgeted time you have left. In the image below, this individual has used up two-thirds of the time they had planned. One-third of the time is still left:



When a task requires more time than you had planned, it is considered **overspent**. When that occurs, the bar will turn red, and will begin to grow. The more overspent you are, the larger the red bar will become. In the image below, the task has required 50% more time than originally expected:



When a task is 100% overspent (that is, it has taken twice as long as planned), the red bar will completely fill the space, and will not grow any larger.

At any time, you can hold your mouse over the indicator, and see a tooltip showing the planned time for the task, the actual time spent so far, and the percent spent calculation. (Percent Spent = Actual Time ÷ Planned Time.)

3.3.1. Editing the Time Estimate

Sometimes, your original estimate for a task was simply a "best guess." As you work on the task, you often gain additional insights and can produce a more accurate estimate of the time that will be required. In general, you can alter your estimate simply by clicking on the percent spent indicator.

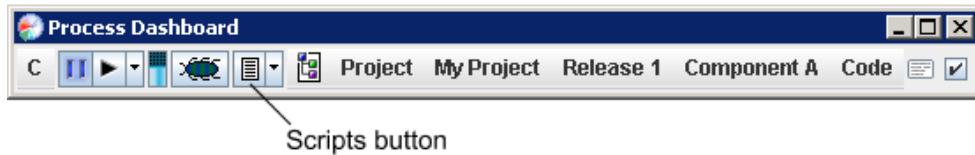
Sometimes, however, the time estimate for the current task is not directly editable. The most common example might be the phases in a PSP project. Currently in the dashboard, you only have to enter the total planned time for the overall PSP project. Then, your historical data is used to spread this time across the various process phases. Since the time for an individual phase is a calculation, it is not directly editable. As a result, if you were to view the Project Plan Summary, the planned time for the phases would be displayed with a "read-only" gray background, and you would not be allowed to edit the values. Similarly, clicking on the percent spent indicator would not allow you to edit the estimated time for the phase.

In such situations, the percent spent indicator can actually play two roles. If the PSP project as a whole is taking much longer than expected, each phase will most likely be overspent as well, and the red bar will bring this to your attention. But even if you are under your estimate on the project as a whole, certain phases may be taking more or less time than expected, and this information can be useful. For example, a low quality product might cause you to find many defects in testing, pushing your test time higher than expected. In this case, the red bar could be an indication that the current phase is not going as planned. On the flip side, if you planned to perform a very thorough code review, a nearly full blue bar may be telling you that you haven't spent enough time reviewing yet.

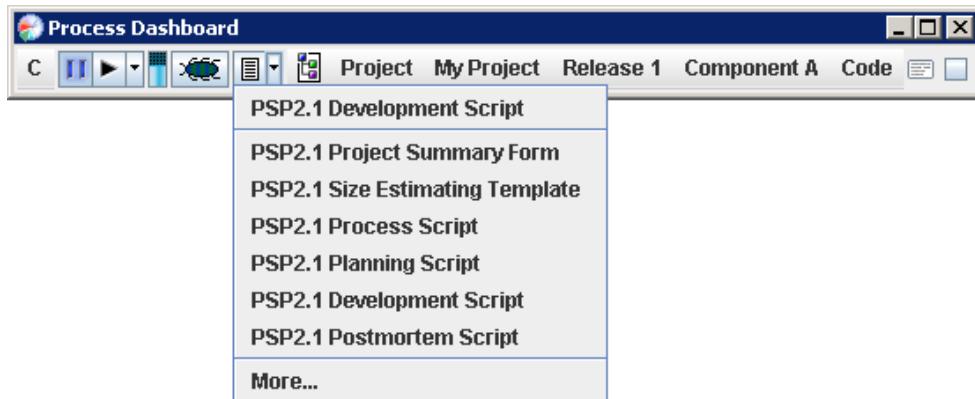
3.4. Accessing the Process Scripts

Most of the scripts necessary to perform your work are included in an add-on module for the Process Dashboard. In particular, the process scripts for all of the PSP levels described in *A Self-Improvement Process for Software Engineers* are included in this optional module. Of course, any dashboard process (including [custom processes](#)) can provide integrated process scripts to guide your work.

To access process scripts, hit the script button on the dashboard.



Clicking the script button will open a menu showing the scripts, forms, reports, and other materials that have been provided by the process for the currently active task. The first item in the list is the default item for the process phase or task you are currently performing.



Choosing an item from this menu will open the named script or form in your web browser.

PSP2.1 Planning Script - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost:2468/Project/My+Project/Release+1/Component+B//psp2.1/planning.htm

PSP2.1 Planning Script

Purpose	To guide the PSP planning process	
Entry Criteria	<ul style="list-style-type: none"> • Problem description • PSP2.1 Project Plan Summary form • Size Estimating, Task Planning, and Schedule Planning templates • Historical size and time data (estimated and actual) • Time Recording log 	
Step	Activities	Description
1	Program Requirements	<ul style="list-style-type: none"> • Produce or obtain a requirements statement for the program. • Ensure that the requirements statement is clear and unambiguous. • Resolve any questions.
2	Size Estimate	<ul style="list-style-type: none"> • Produce a program conceptual design. • Use the PROBE method to estimate the added and modified size of this program. • Complete the Size Estimating template and Project Plan Summary form. • <i>Calculate the 70% size prediction interval (using the PROBE Wizard).</i>
3	Resource Estimate	<ul style="list-style-type: none"> • Use the PROBE method to estimate the time required to develop this program. • <i>Calculate the 70% size prediction interval (using the PROBE Wizard).</i> • Using the To Date % from the most recently developed program as a guide, the dashboard will automatically distribute the development time over the planned project phases.
4	Task and Schedule Planning	<ul style="list-style-type: none"> • For projects lasting several days or more, complete the Task Planning and Schedule Planning templates.
5	Defect Estimate	<ul style="list-style-type: none"> • Based on your to-date data on defects per added and modified size unit, the dashboard will automatically estimate the total defects to be found in this program. • Based on your To Date % data, the dashboard will automatically estimate the number of defects to be injected and removed by phase.
Exit Criteria		<ul style="list-style-type: none"> • Documented requirements statement • Program conceptual design • Completed Size Estimating template • For projects lasting several days or more, completed Task and Schedule Planning templates • Completed Project Plan Summary form with estimated program size, development time, and defect data, <i>and the time and size prediction intervals</i> • Completed Time Recording log

[Next: Development.](#)

[Top](#)

Adapted from "PSP Materials," copyright © 2005 Carnegie Mellon University. Used by permission.

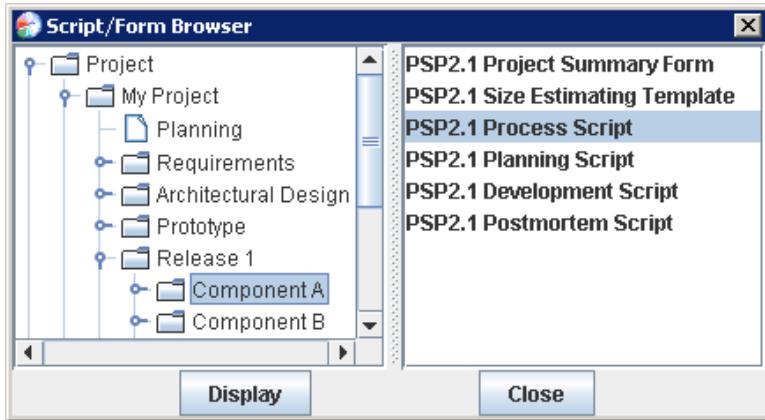
Done

This example shows a project calling up the Planning script for PSP2.1.

Clicking the script button normally opens the menu of scripts and forms. If you prefer, you can reconfigure the script button so that it

automatically opens the default script or form. Just check the appropriate box in the [Preferences Tool](#). After enabling that option, you can use the drop-down button to open the full menu.

You can also select the "More..." option at the bottom of the menu to bring up a script browser. The script browser shows your hierarchy on the left; when you highlight a node in the hierarchy, it displays the available scripts and forms on the right. You can open one of those scripts or forms by double-clicking on it.



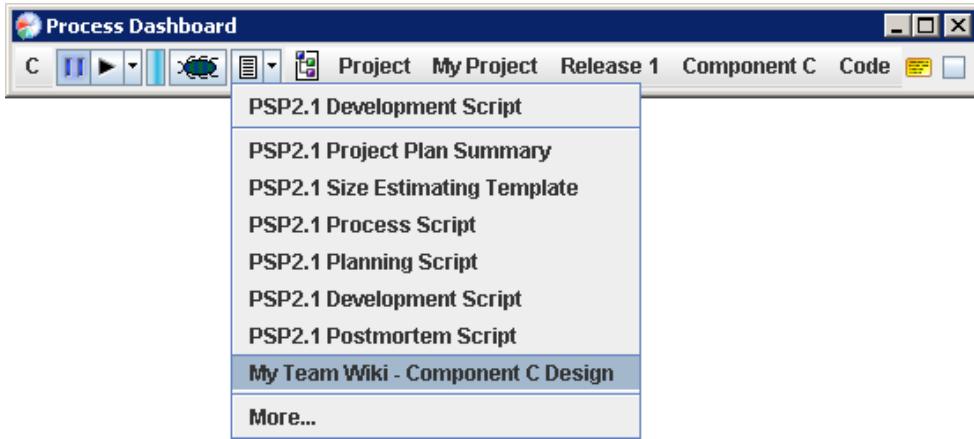
The script browser makes it easy to pull up forms for historical projects without first needing to navigate the hierarchy menus to that project.

3.4.1. Custom Hyperlinks

In addition to the scripts and forms that come with the dashboard, you may have other web pages that are related to the task at hand. For example:

- Your team may have a wiki page, sharepoint site, or other collaboration area for items relating to your project
- The requirements for a component may be accessible at a particular webpage
- Your team may have a wiki page where you are discussing the design of a component
- You may have a link to a page in a defect tracking system describing a set of changes that need to be made

You can easily attach URLs such as these to the projects and tasks in the dashboard. Just [enter a note](#) for a particular task or component, and include the URL in the body of the note. The dashboard will discover these embedded URLs and automatically add them to the script button pull-down menu.



For more information on this feature, see the [note editor](#) help topic.

3.4.2. The timing icons

Many of the built-in process scripts contain little stopwatch icons on them. These icons provide another way to navigate and time a process phase. When you click on one of these icons, the current phase being measured will change to the phase indicated by the icon

and the timer will start measuring time for that phase.

3.4.3. External documents

Creating or obtaining documents is often an integral part of performing a process. The dashboard process scripts sometimes provide hyperlinks for these documents. If you click on a document hyperlink, the dashboard will:

1. Possibly ask you for pertinent information. For example, the dashboard may need to know the name of a directory where it should store documents for this project, and it may display a form requesting that information. Once you provide this information and click "OK," the dashboard generally won't have to ask for it again.
2. Check to see if the file already exists. If not, it may be able to create the file from a template.
3. Open the file for editing.

3.4.4. Entering data into the forms

The various forms necessary to do your work have data entry fields which are to be used to capture the various process metrics. The forms are continually "live", which means they are always ready and waiting for input. There is no "submit" or "send" button on the forms. Just enter the data in the blanks and the dashboard will record it. Some blanks on the forms are greyed out. These data elements are computed by the dashboard and do not require data input. For more information, see the [Entering data](#) help topic.

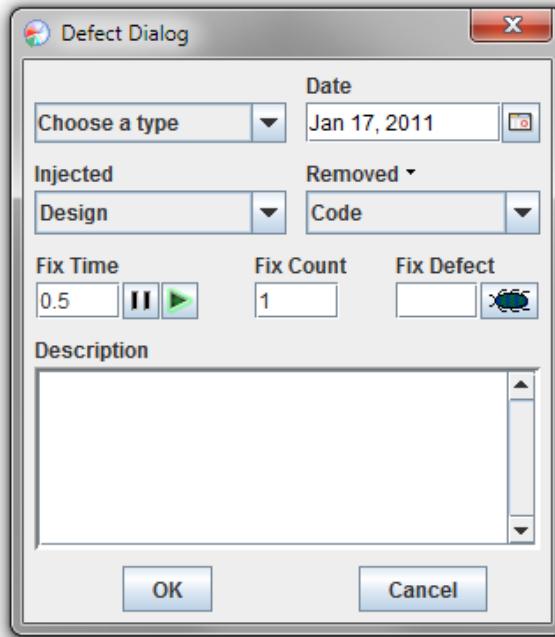
3.5. Entering defects

We all make mistakes. Dealing with those mistakes in a constructive way so that we can learn from them is a lot of what process improvement is all about.

Entering defects in the dashboard is as easy as choosing the defect button.



When the defect button is chosen, the defect dialog pops up so that you can enter information about the defect that you have discovered.



In this dialog box, you enter all the data necessary to describe the defect. Each feature of the dialog box is detailed below.

3.5.1. Defect Type

The defect type pulldown allows you to choose the type of defect. The choices are determined by the defect type standard that is in effect for the currently selected project. You can create new defect type standards, and configure the defect type standards that should be used for various projects; see the [defect editor](#) help topic for more information.

3.5.2. Defect Date

This field captures the date the defect was found and/or removed. It will automatically be filled in with the current date, but you can edit it if the defect was found or removed in the past.

3.5.3. Phase Injected and Phase Removed

The defect injected and defect removed pulldowns allow selection of the process phases in which the defect was injected and removed, respectively. The dashboard will set both of these to what it thinks are appropriate choices given the current development phase, but that default choice can be overridden easily.

Sometimes, you might discover a defect, but might not choose to fix it right away. (For example, you might need to talk with others to determine the best course of action for the fix.) In this case, you can click the "Removed" label, then select the option indicating that the fix

for the defect is still pending. After the defect has been fixed, you can use the [Defect Log Editor](#) to re-edit this defect and change the status to "Removed."

Note: changing this "pending" flag will **not** affect the defect removal counts that appear on the Project Plan Summary form for your project. Those removal counts will still include "pending" defects, even though the defects aren't officially removed. This behavior allows the defect calculations (such as yield, defect density, and defect removal rate) to continue working as expected.

3.5.4. Timing the Defect Fix

The defect dialog has its own built in timer independent of the main dashboard timer. By default, when the dialog first appears, it starts timing the fix of the defect. The fix time display will show the time passing. The intent of this behavior is that most people will discover a defect, push the defect button on the dashboard and immediately start fixing the defect. In this scenario, the dialog will immediately start timing the fix. This behavior can be changed via an entry in the preferences editor. Adding the following advanced setting will cause the defect dialog to appear without starting the defect timer.

```
defectDialog.autostart=false
```

For more information about making this configuration change see the [advanced settings](#) help topic.

The play and pause buttons allow starting and stopping of the defect timer. If you stop fixing the given defect but are not finished with the fix, you can press this button to stop the timer.

The defect fix timer can be started and stopped independently of the main dashboard timer. However, if both timers are running and you pause the main dashboard timer, the defect fix timer will pause too. After being paused in this way, if you restart the main dashboard timer, the defect timer will restart as well.

In any case, you can stop the timer and enter a specific time in the fix time display box.

3.5.5. Fix Count

Sometimes, you may find and fix several related defects. (For example, during the compile phase, the compiler might identify four missing semicolons, and you might fix all of them at once.) To simplify defect entry, you can make one entry in your defect log describing all of the defects, then enter the number of distinct defects in the "Fix Count" field.

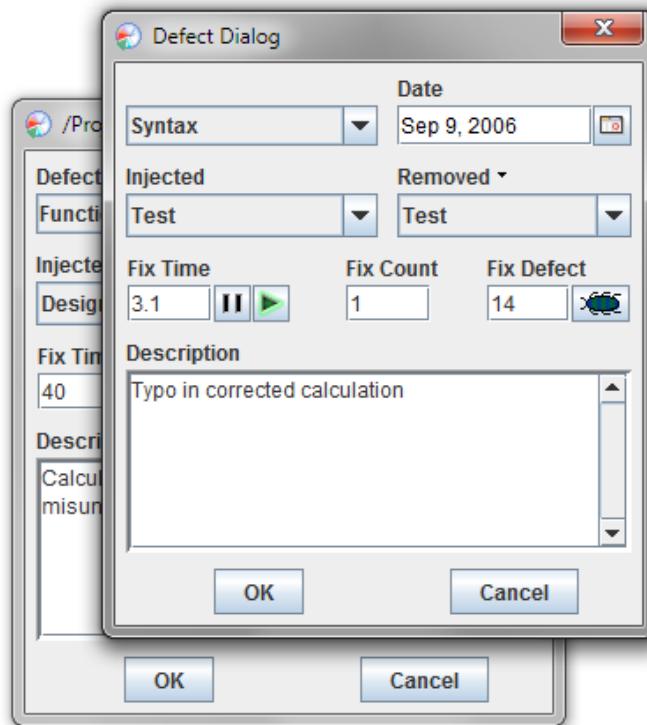
Most of the time, however, you will leave the count at its default value of 1. You should only use the Fix Count field when all of the defects in question have the same type, injection/removal phase, and description. If you use the Fix Count field to record several unrelated defects, it will thwart your ability to analyze the defect data effectively.

As a very special case, you can enter the number "0" in this field to indicate that something isn't really a defect after all, or that it is a defect but should be ignored when calculating quality metrics. (For example, this might be appropriate if you find and fix a defect in legacy code that you did not write.) If you enter a zero in this field, the defect will still appear in defect logs and other reports, but it will not contribute to the "Defects Injected" or "Defects Removed" counts on the plan summary form. As such, it will not affect calculations for yield, defect density, defect removal rates, or other quality metrics.

3.5.6. Fix Defect

The "Fix Defect" entry is where you can put the ID number of another defect if this is a "fix defect". A fix defect is a defect that was injected while fixing another defect. In order to determine the ID number of the other defect, you can pull up the defect editor by choosing "Defect Log" on the "C" menu. One of the columns of information about each defect will show the ID number. For more information about using the Defect Log see the [Using the defect log editor](#) help topic.

The "new fix defect" button to the right of the ID entry provides another way to enter a fix defect. Consider the following relatively common case. You are fixing a bug (so you already have the defect dialog open and it is timing your fix). You edit your code and save the changes. You compile and link to run a test to verify your fix. Immediately you notice that your fix broke something else. At this point, you can click the new fix defect button to get a second defect dialog (to enter this new defect). This second dialog will have the ID number of the previous defect already entered in the ID entry blank. You can now input data relating to this new fix defect. The following screenshot shows two defect dialogs open. The second dialog (the one on top) is being used to enter a fix defect.



This process works with defects that you edit from within the defect log editor as well. See the section below on [fixing multiple defects](#) for more discussion on having two defect dialogs open at once.

3.5.7. Entering a description

The defect description pane is where you record a description of the defect. To facilitate effective process improvement, you should describe the underlying defect itself, and not just the outward symptom of the defect. (For example, an outward symptom might be that the program hangs, while the underlying problem might have been forgetting to increment the index variable in a loop.)

3.5.8. Fixing multiple defects at a time

If you have the need to fix more than one defect at one time, you can do so by simply pushing the defect button on the dashboard again to display another defect dialog. When this happens, the timer on the first dialog will stop (if it was running) and the new defect dialog's timer will start recording time for the new defect. This new defect will become the "current" defect. When this new defect dialog is dismissed, the most previously opened defect dialog will become the current defect. It will pop to the front and its timer will start up again (if it had been running previously).

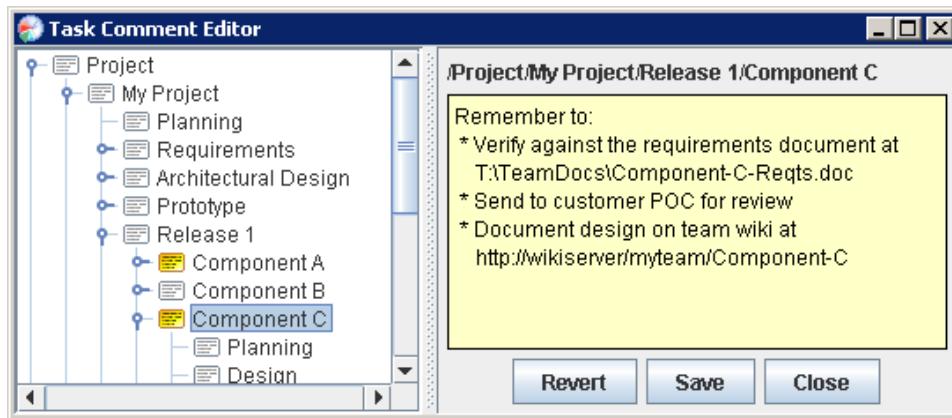
In this way you can keep an arbitrary number of defect dialogs open at once. If this is to be done, we recommend that you enter a description of the defect into the dialog box as soon as it is opened so that when it comes back to the front as the currently active defect, it will be clear what you were working on.

3.6. The Task Note Indicator

Notes and comments can be attached to any project or task in the dashboard. To create, view, or edit these notes, click the note indicator icon near the right edge of the main dashboard toolbar.



The Task Comment Editor window will appear.



This window displays your project/task hierarchy on the left. Click a particular project, component, or task to create, view, or edit the associated note. To delete a note, simply delete all the text in the note, then save the changes.

When a note is associated with the currently active task (or any of its hierarchical parents), the note indicator on the main dashboard toolbar will turn yellow, and the tooltip will provide the text of the note.

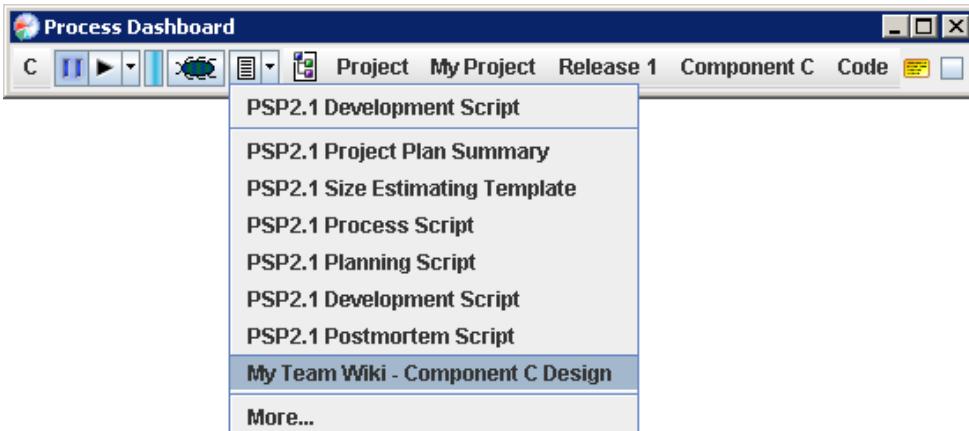


If you attach a note to a component or task within a team project, that note will be shared with other members of the team project when you perform a "Sync to WBS" operation.

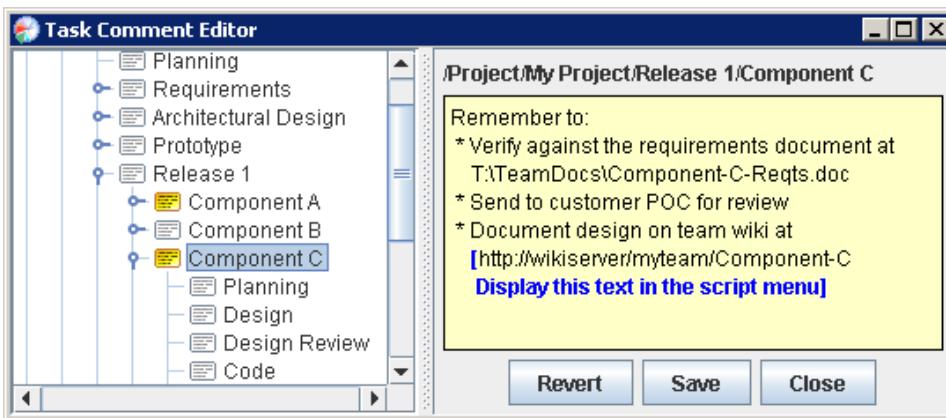
3.6.1. Linking to External Websites

Most people make use of many websites as they work. Your organization may have websites for tracking requirements, defects, and other issues. And your team may have websites for team collaboration.

These sites may include web pages that are relevant to a particular project, component, or task in the dashboard. You can easily link these webpages to your tasks in the dashboard. Just enter a note for a particular dashboard task or component, and include a webpage URL in the body of the note (as shown in the example above). The dashboard will discover these embedded URLs and automatically add them to the script button pull-down menu.



By default, the script button menu item will display the title of the target web page. If you want the menu item to display something else, just include the desired title after the URL, and enclose the item with square brackets []. For example:



3.7. Using the completion checkbox

As you complete the activities for a given process task, you should mark that task as completed via the completion checkbox.



When the completion checkbox is pressed, the dashboard moves the currently active task to the next item in the rightmost hierarchy menu. If the timer is currently running, it will automatically stop recording time for the given phase and start the timing of the next phase. For more information on using the timer, see the [Using the Play/Pause button](#) help topic.

Phases marked as complete will show a little checkmark in the completion checkbox. (If you hover the mouse over the checked box, the tooltip will tell you exactly what date and time the phase was marked complete.)

If you need to go back to a previous phase in a given process, you can still do so even if the completion checkbox has been pressed. The dashboard will still accumulate time in whatever phase you make current. (It will display a warning to make certain you're not logging time to the completed task by mistake, but you can safely dismiss this dialog.)

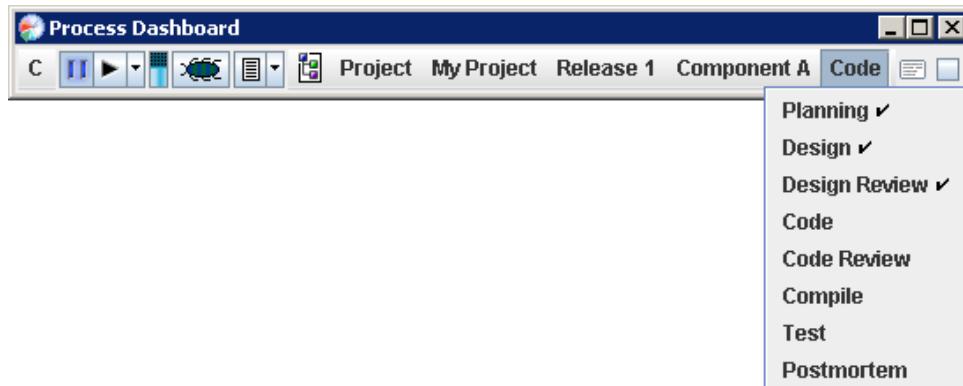
Marking a *phase* complete vs marking the *project* complete

By default, a project will be considered complete when all the phases in the project have been marked complete. However, if you manually edit the project completion date on the plan summary form (either via the input field or the checkbox), then you have manually overridden this behavior. From then on, the project completion status must be edited manually; it will no longer respond to changes in phase completion status. (Actually, you can restore the default behavior by typing **DEFAULT** in the project completion date input field on the plan summary form.)

Why should I bother marking the phases/projects complete?

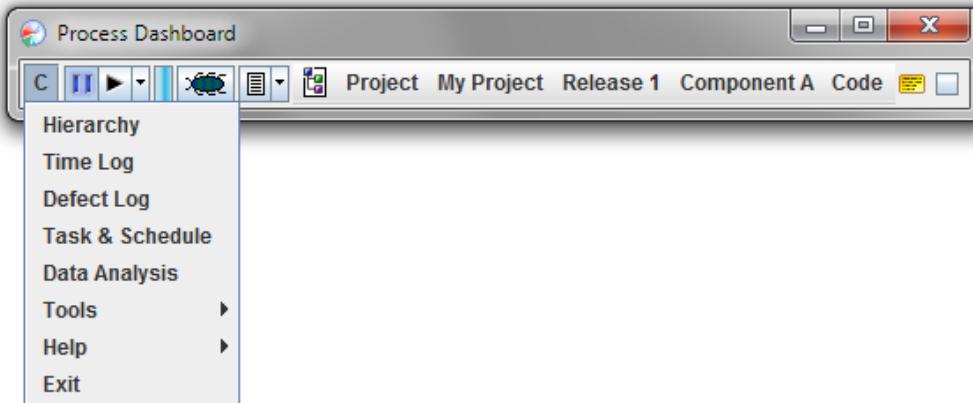
It is worth remembering that the "To Date" data entries, the data analysis reports, and the PROBE tool only use data from projects that are marked complete, so it is good to make a habit of marking your projects complete, either with the completion checkbox or via the Project Plan Summary form.

Also remember that the task and schedule tool tracks your progress by looking to see which tasks have been marked complete, so it is good to make a habit of marking phases complete. To help you keep track of what has and hasn't been marked complete, the hierarchy menus will display a checkmark to the right of completed tasks.



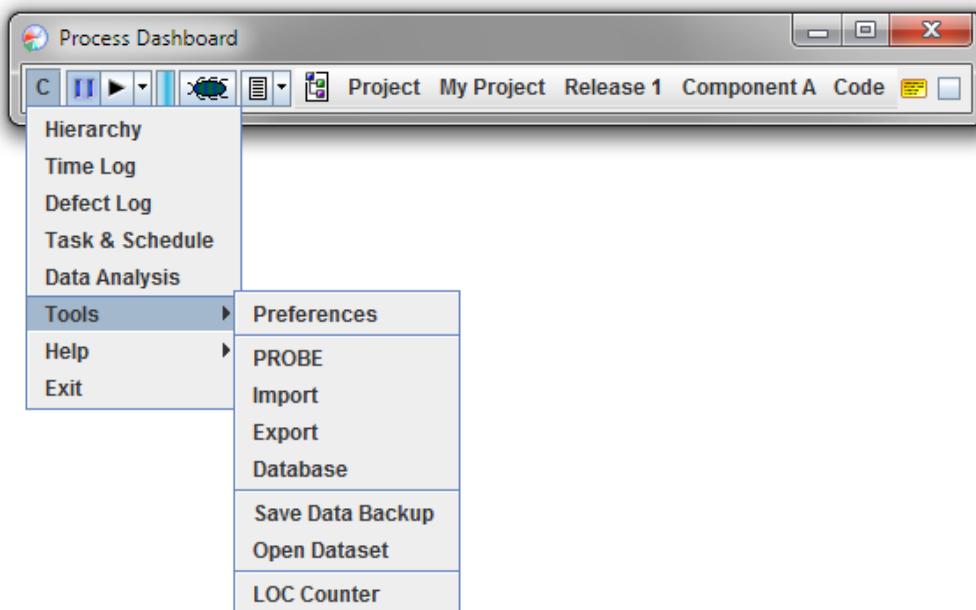
3.8. The "C" (configuration) menu

The dashboard is configured and various actions are taken by choosing the configure menu. It is shown with just the letter "C" in order to keep the dashboard as small as possible.



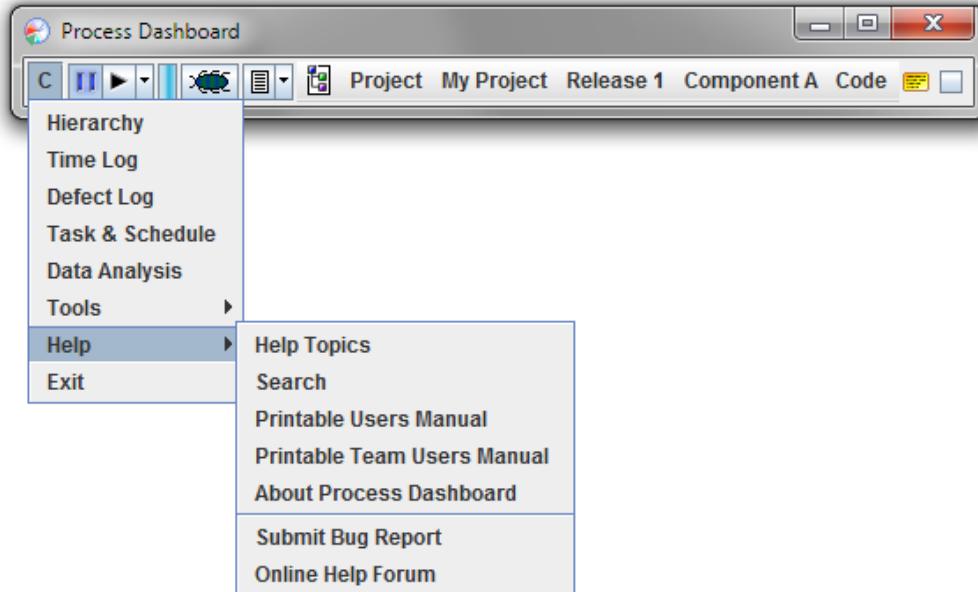
The various entries on this menu allow access to various tools and editors to give you complete control of your PSP data.

- [Hierarchy](#) provides access to and simple editing of the project hierarchy.
- [Time Log](#) provides an editor that allows editing of the time data measured in each of the process phases.
- [Defect Log](#) provides a listing of defects and a method of making changes to the defect data.
- [Task & Schedule](#) provides access to a tool for creating a schedule, and tracking your progress against it using earned value.
- [Data Analysis](#) provides access to many helpful charts, graphs, and reports for postmortem analysis.
- [Tools](#) provides access to a sub menu of less commonly used tools.
 - [Preferences](#) allows you to edit settings that affect the appearance and behavior of the Process Dashboard.
 - [PROBE](#) provides access to a tool to help with the PROxy Based Estimation method.
 - [Import](#) and [Export](#) provide access to tools that allow data to be transferred between dashboards in a compressed structured file format.
 - [Database](#) provides SQL access to the a relational database of team project data
 - [Save Data Backup](#) allows you to save a snapshot of your data for archival or communication purposes.
 - [Open Dataset](#) allows you to view the data in a data backup snapshot that was previously.
 - [LOC Counter](#) provides access to a helpful tool for measuring added, deleted, and modified LOC.



- [Help](#) provides access to a sub menu of help-related items.
 - [Help Topics](#) brings up this help viewer with the Table of Contents pane selected.
 - [Search](#) brings up this help viewer with the Search pane selected.

- [Printable Users Manual](#) displays a printer-friendly version of the Process Dashboard Users Manual.
- [Printable Team Users Manual](#) displays a printer-friendly version of the Team Use Manual.
- [About Process Dashboard](#) displays the "about" screen.
- [Submit bug report](#) will bring up your web browser pointed to our website where you can submit any bug you might find in the Process Dashboard. We want the Dashboard to be the best that it can be, so feel free to submit a bug!
- [Online help forum](#) will bring up your web browser pointed to our website where you can post messages to an online help forum. Your questions can be answered by other users of the Dashboard or by the Dashboard developers who frequent the forum as well.



- [Exit](#) exits the dashboard.

3.9. Using the time log editor

You can use the time log editor to correct mistakes you might make when timing your activities. The time log editor can be opened by selecting the "Time Log" option on the [configure menu](#).

The screenshot shows the 'Time Log Editor' window. On the left, a hierarchical project view displays a tree structure of projects and tasks with their total logged time. In the center, a date filter allows users to specify a range of dates. On the right, a detailed list of time entries shows each activity's start time, duration, and other metadata.

Logged To	Start Time	Delta	Int	Comment
/Project/My Project/Planning	Jan 05, 2006 08:00:00 AM	8:12	0:00	
/Project/My Project/Requirements/Component A	Jan 06, 2006 05:46:55 PM	5:13	0:00	
/Project/My Project/Requirements/Component A	Jan 09, 2006 05:46:55 PM	7:01	0:00	
/Project/My Project/Requirements/Component B	Jan 10, 2006 05:46:55 PM	3:00	0:00	
/Project/My Project/Requirements/Component B	Jan 16, 2006 05:46:55 PM	3:00	0:00	
/Project/My Project/Requirements/Component C	Jan 17, 2006 05:46:55 PM	8:00	0:00	
/Project/My Project/Requirements/Component C	Jan 23, 2006 05:46:55 PM	8:00	0:00	
/Project/My Project/Requirements/Component D	Jan 24, 2006 05:46:55 PM	8:00	0:00	
/Project/My Project/Requirements/Component D	Jan 30, 2006 05:46:55 PM	1:00	0:00	
/Project/My Project/Architectural Design/Draft	Feb 01, 2006 06:18:48 PM	8:12	0:00	
/Project/My Project/Prototype/Component A	Feb 03, 2006 06:21:59 PM	4:12	0:00	
/Project/My Project/Prototype/Component A	Feb 06, 2006 06:21:59 PM	4:41	0:00	
/Project/My Project/Prototype/Component B	Feb 07, 2006 06:21:59 PM	8:41	0:00	
/Project/My Project/Prototype/Component B	Feb 13, 2006 06:21:59 PM	1:11	0:00	
/Project/My Project/Prototype/Component C	Feb 14, 2006 06:21:59 PM	7:13	0:00	

Hierarchical project view Date filter Time entries

3.9.1. Project hierarchy

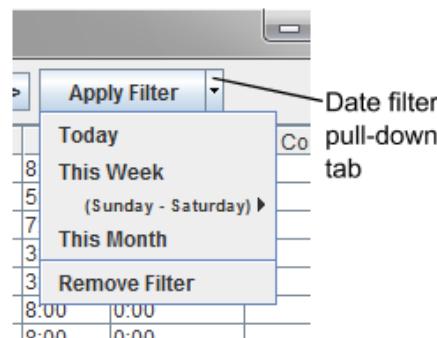
The left pane shows the current project hierarchy. Selecting various branches on the hierarchy will limit the time entries displayed to just those on the branch chosen.

Next to the name of each hierarchy node, the dashboard displays the total amount of time that has been logged against that node and all its children. If you apply a date filter (see below), these times will recalculate based on the filtered view of the time log. You can use the "Time Format" pulldown control to display these times in hours:minutes, decimal hours, or percentages (of either the total time, or the time logged to the parent of the node).

3.9.2. Date filter

Above the split pane is a set of controls that provide date filtering on the time entries display. Once the desired span of dates has been chosen, clicking the "Apply Filter" button will restrict the data shown to the range of dates specified.

The Apply Filter button has a pulldown control that provides for several common date filters.



To the left of the "From" entry box and the right of the "To" entry box are double arrowed buttons. These buttons will jump the date filtering forward and backward by a unit of time equal to the current filtering date range. For example, if you have filtered the data in the Time Log such that only one week of data is showing, pressing these buttons will cause the filtering to move backward or forward by one

week at a time.

In addition, you can enter dates directly in the "From" and "To" boxes to perform a customized date range.

3.9.3. Time entries

To change data in the time entries, just pick on the element that you would like to change. The elements in the time entries pane are editable. The "Logged To" column shows the process phase to which the time was logged. If you edit values in this column, you must type the name of a hierarchy phase exactly; capitalization, spacing, etc. are all significant. If you type an invalid value, the editor will restore the original value. The "Start T" column shows the start time for the activity. The "Delta" column shows the length of time spent in the activity. The "Int" column shows the amount of time that the given activity was interrupted.

3.9.4. Control buttons

The Add and Delete buttons allow the addition and deletion of time entries. When Add is chosen, by default the last entry in the time log will be copied into a new log entry. You can choose which entry will be copied to make the new time entry by picking one of the entries before clicking the Add button, or by highlighting the corresponding leaf of the tree.

The Save and Revert buttons will save the changes or revert the entries to their previous values.

The Summarize button will collapse all time entries that were logged to the same process phase into one entry. The Dashboard will create new entries each time that it starts recording time in a given process phase (i.e. when a process phase is left and then returned to again for more time measurement). This function will allow you to collapse all duplicates entries in the current display.

The Time Card View button will display another view of the time data.

The screenshot shows the 'Time Card' window. At the top, there are controls for 'Month: January' (with a dropdown arrow), '2006', 'Time Format: 1:15' (with a dropdown arrow), and a checked 'Hide empty columns' checkbox. A 'Close' button is also present. On the left, a tree view shows a 'Daily Total' node, followed by a 'Project' node with a 'My Project' node underneath. Under 'My Project', there are 'Planning', 'Requirements', 'Component A', 'Component B', 'Component C', and 'Component D'. The main area is a grid table with columns for days (5, 6, 9, 10, 16, 17, 23, 24, 30) and a 'Total' column. Rows represent time entries for each day, with specific times listed in the grid cells. For example, for Component A, the times are 5:13, 7:01, 3:00, 3:00, 8:00, 8:00, 8:00, 8:00, 1:00, resulting in a total of 12:14.

The Time Card View is handy if you work for an employer that makes you submit time cards or time sheets. You can open this window and refer to the data it contains as you enter data into your employer's time card.

At the top left, a pulldown and a text field allow you to choose the month and year to display. On the left side of the window is an abbreviated version of your hierarchy, containing only the nodes and tasks where you logged time during the selected month. Columns are displayed for the days in the month; tooltips on the column headers display the corresponding day of the week. Cells in the table display the amount of time logged by day and by task.

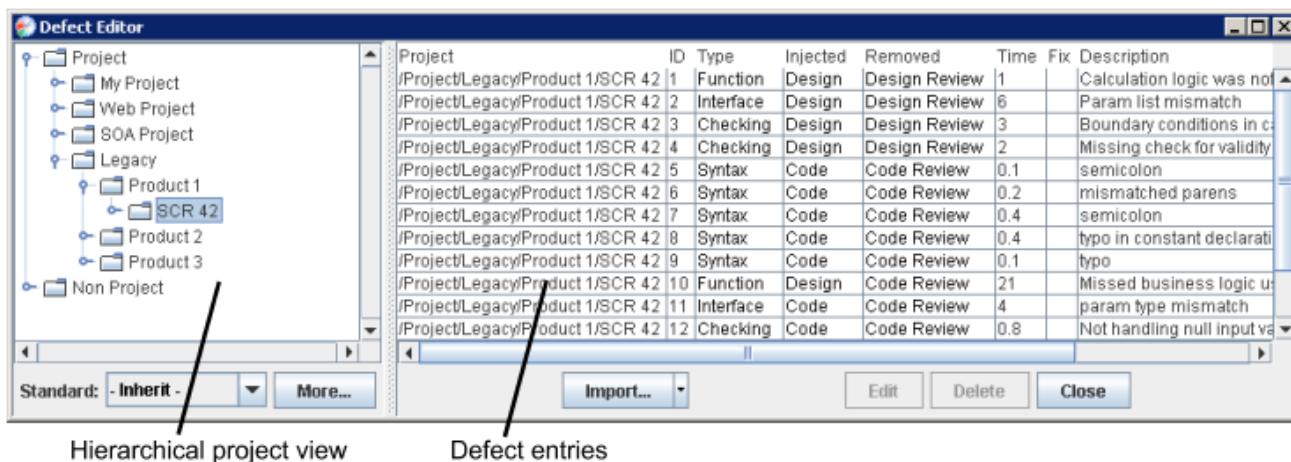
Two controls in the top center of the window allow you to customize the display. You can hide columns for days that contain no time data using the checkbox, and you can change the display time format using the pulldown.

A tooltip is shown over the 'Time Format: 1:15' dropdown menu. The tooltip displays the current format '1:15' and its numerical equivalent '1.25'. Below that, the value '75' is shown, likely representing the raw time value in minutes.

You can choose to display time in hours:minutes, decimal hours, or minutes.

3.10. Using the Defect Log Editor

You can use the defect log editor to correct mistakes you might make when entering defects. The defects log editor can be opened by selecting the "Defect Log" option on the [configure menu](#).



The left pane shows the current project hierarchy. Picking various branches on the hierarchy will limit the defect entries displayed to just those on the branch chosen.

To change data in the defect entries, just pick on the element that you would like to change. When a particular defect entry is picked, the Edit button becomes active. Clicking the Edit button will bring up the same dialog box used to enter defects. All the defect data can be changed using this dialog. For help using the defect entry dialog please see the [Entering defects](#) help topic.

In addition, it is possible to import defect data from external sources by clicking the "Import" button. At the moment, three import sources are built-in:

- A wizard for importing defects that have been copied onto the system clipboard. You can use this to copy and paste defect data from a web page or a spreadsheet.
- A wizard for importing defects from Code Collaborator (<http://smartbear.com/>).
- A wizard for importing defects from Review Board (<http://reviewboard.org/>).

3.10.1. Configuring Defect Type Standards

The dashboard will allow you to configure the "defect type standards" that you would like to use for various projects.

Defect type standards are associated with projects hierarchically. By default, projects just inherit the standard chosen by their parent. In addition, there is a "global default" defect type standard that is inherited when no other settings have been chosen.

As you select various branches in the project hierarchy, a component at the bottom of the left pane will show you the defect type standard setting in effect for the highlighted branch. You can select a different defect type standard from the drop-down list provided. The new value you select will become the default defect type standard for the highlighted branch of your project hierarchy.

Click on the "More..." button at the bottom of the left pane to:

- Edit an existing defect type standard
- Create a new defect type standard
- Change the global default defect type standard

3.11. The System Tray Icon

If your operating system supports it, and if you are running version 1.6 or higher of the Java Runtime Environment, the Process Dashboard will display an icon in the system tray. Double-clicking on the dashboard's tray icon will show the main Process Dashboard window.



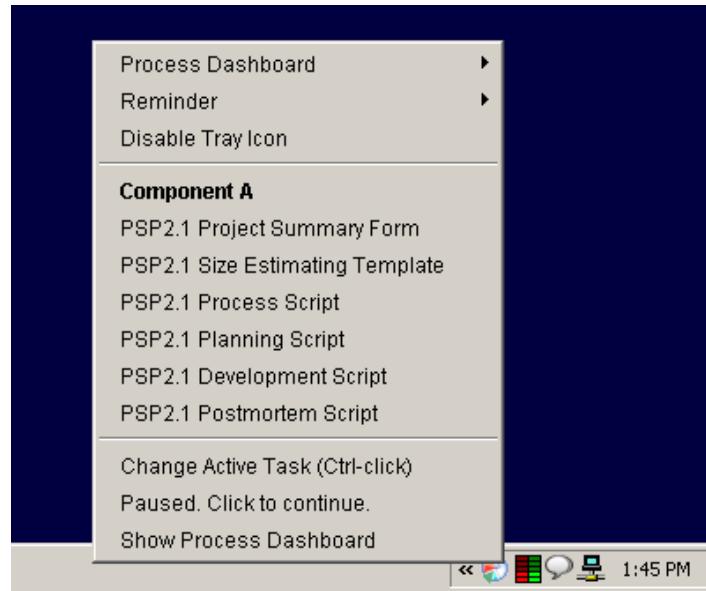
If you place your mouse over this icon, a tooltip will indicate the name of the active task. If you hold down the `Ctrl` key and click on the icon, a window will be displayed allowing you to change the active task.



Clicking once on the icon will start/stop the timer. A small "play" indicator appears on the icon to indicate when the timer is running.



A right-click on the icon will display a popup menu. Among other things, this menu duplicates the contents of the [configuration menu](#) and the [script menu](#).



In addition, the popup contains a "Reminder" submenu. This menu allows you to enable a timing reminder, which will appear at regular intervals. The reminder will ask whether you are still working on a particular task (or whether you are still idle, depending on the state of the play/pause button). If you often forget to start/stop the timer, this reminder can help you establish a consistent habit of metrics collection.

If you do not wish to use the system tray icon, you can disable it using the option provided in the popup menu. In the future, if you decide that you want it back, you can reenable it via the [preferences editor](#).

4. Postmortem analysis

Just as it is necessary to plan a software project, it is also advantageous to learn from the experience of a software project as well. In the PSP course, a lot of postmortem time is spent calculating the various metrics and recording them. The Process Dashboard will take away much of the tedium associated with a Postmortem analysis.

For the most part, the work involved in a postmortem for a project that uses the Process Dashboard is automated. The dashboard knows how to fill out the Project Plan Summary form with all the derived metrics that are a part of the PSP, and it also knows how to create many helpful graphs to let you get a visual feel for how the data look.

The main thing that isn't done automatically is the final LOC counts. You can use the built-in [LOC counter](#), or an external LOC counter of your own choosing, to calculate LOC; then enter the final counts on the Project Plan Summary form (for PSP levels 0 and 0.1) and on the Size Estimating Template (for PSP levels 1 through 3).

4.1. Counting LOC

Measuring size is an important step in the PSP Postmortem phase. If you don't pause in your project postmortem to measure the number of added, deleted, and modified LOC, you may forever lose the opportunity to capture these metrics. It only takes a few moments, and provides you with valuable data for future planning efforts.

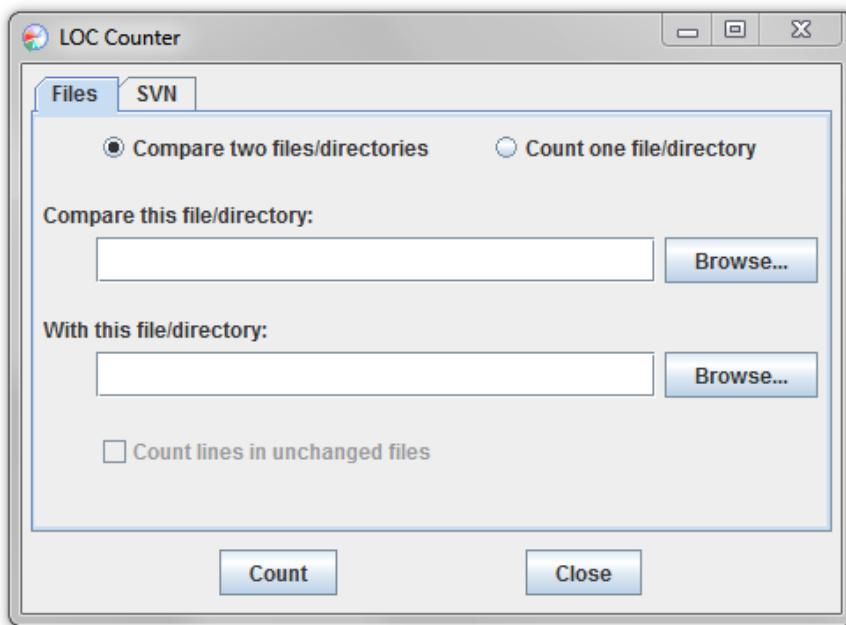
You may already have a LOC counter that you use on a regular basis. (In fact, most people who complete the PSP training course write a LOC counter during the training.) If you do need a LOC counter, one is built into the dashboard.

4.1.1. The LOC Counter Tool

You can access this LOC counter by selecting the appropriate option from the "C → Tools" menu. The LOC Counter Tool can count code in several different ways.

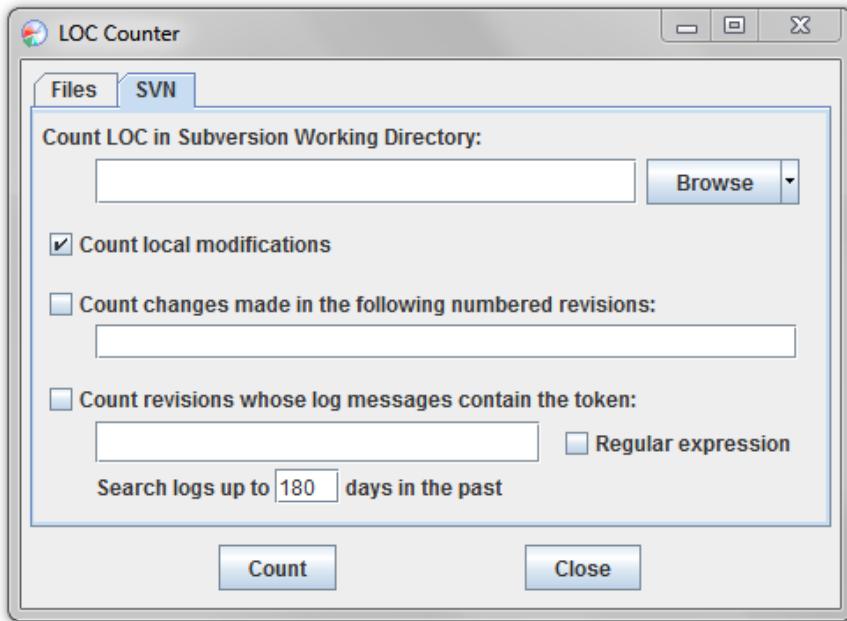
4.1.1.1. Counting LOC in regular files

The "Files" tab of the LOC Counter allows you to count LOC in a single file or in a directory. It also allows you to compare two different files or directories to see how many lines of code were added, modified, and deleted. Just select the appropriate option at the top, then select the files or directories in the fields below. Then press the Count button.



4.1.1.2. Counting LOC changes from Subversion

If you use subversion for version control of your source code, the "SVN" tab makes it easy to count the number of lines of code that were added, deleted, and modified by a particular set of changes. To begin, select the directory on your computer that contains your subversion working copy.



Then, several counting options are provided:

- You can count the changes that have been made in the local working copy, that have not yet been committed to SVN.
- You can enter one or more revisions to count. For example, imagine that you have been working on a particular software change for a period of time. Following an iterative development approach, you have completed portions of the work and committed them at different times as revisions 5, 7, 10, 11, and 12. (Presumably someone else on your team committed unrelated changes as revisions 6, 8, and 9.) You could enter "5, 7, 10-12" in this box, and the LOC counter would show you the changes and LOC counts that were associated with your work.
- Some teams follow a convention where they include a special token in their commit message. For example, while working on Software Change Request #42, you might include the token "SCR42" in the log message each time you commit changes. Once you have finished work on the Software Change Request, you could enter "SCR42" in the third field on this form. (Perl-style regular expressions are also supported if you check the appropriate box.) The dashboard would find all of the revisions whose log message contained that string, and would perform a count just as if those revision numbers had been typed into the previous field.

The options above can be used together: for example, to count the changes in several committed revisions as well as local modifications. After you have selected the desired options, press the Count button.

4.1.2. The Quick LOC Counter

The LOC Counter Tool described above provides powerful options for LOC counting. For quick/simple counting needs, a web-based interface is also provided. You can open this web-based interface by pointing your web browser to <http://localhost:2468/dash/pspdiff.htm>.

In addition, a link to this LOC counter is included in the Postmortem scripts for PSP0.1 and up.

PSP2.1 Postmortem Script - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost:2468/Project/My+Project/Component+B//psp2.1/postmort.htm

PSP2.1 Postmortem Script

Purpose	To guide the PSP postmortem process	
Entry Criteria	<ul style="list-style-type: none"> • Problem description and requirements statement • Project Plan Summary form with program size, development time, and defect data • For projects lasting several days or more, completed Task Planning and Schedule Planning templates • Completed Test Report template • Completed Design templates • Completed Design Review and Code Review checklists • Completed Time and Defect Recording logs • A tested and running program that conforms to the coding and size counting standards 	
Step	Activities	Description
1	Defect Recording	<ul style="list-style-type: none"> • Review the Project Plan Summary to verify that all of the defects found in each phase were recorded. • Using your best recollection, record any omitted defects.
2	Defect Data Consistency	<ul style="list-style-type: none"> • Check that the data on every defect in the Defect Recording log are accurate and complete. • Verify that the numbers of defects injected and removed per phase are reasonable and correct. • Determine the process yield and verify that the value is reasonable and correct. • Using your best recollection, correct any missing or incorrect defect data.
3	Size	<ul style="list-style-type: none"> • Count the size of the completed program. • Determine the size of the base, deleted, modified, base additions, reused, new reusable code, and added parts. (If you do not already have a tool for measuring deleted, modified, and added LOC, you may find this tool useful.) • Enter these data in the Size Estimating template. • Determine the total program size • Enter the size in the Project Plan Summary form.

Link to the LOC Counting Tool

The web interface allows you to count a single file or compare two files to each other. For more sophisticated needs, a hyperlink at the bottom of the page is provided to open the full-featured LOC Counter Tool.

File Comparator

Compare the Files [LOC Counting Options:](#)

Initial File (leave blank to count the LOC in a new file)

Select a file: [Choose File](#) No file chosen
Or copy and paste the original text here:

Modified File (or new file)

Select a file: [Choose File](#) No file chosen
Or copy and paste the modified text here:

[Click here to open the full LOC Counting Tool](#)

The web interface will alternatively allow you to copy and paste code snippets that should be counted or compared. When you do this, keep in mind that the counter can no longer examine your filenames (e.g. "foo.java" or "foo.cpp"), so it may have a difficult time determining which programming language syntax to use for the count. In that case, you may need to enter explicit LOC Counting Options in the field at the top of the form.

4.1.3. The LOC Counter Report

After using any of the tools above, the dashboard will generate a report showing the number of physical LOC that you have added, deleted, and modified. It also displays a "redlined" version of the modified files, which can be very useful for peer reviews.

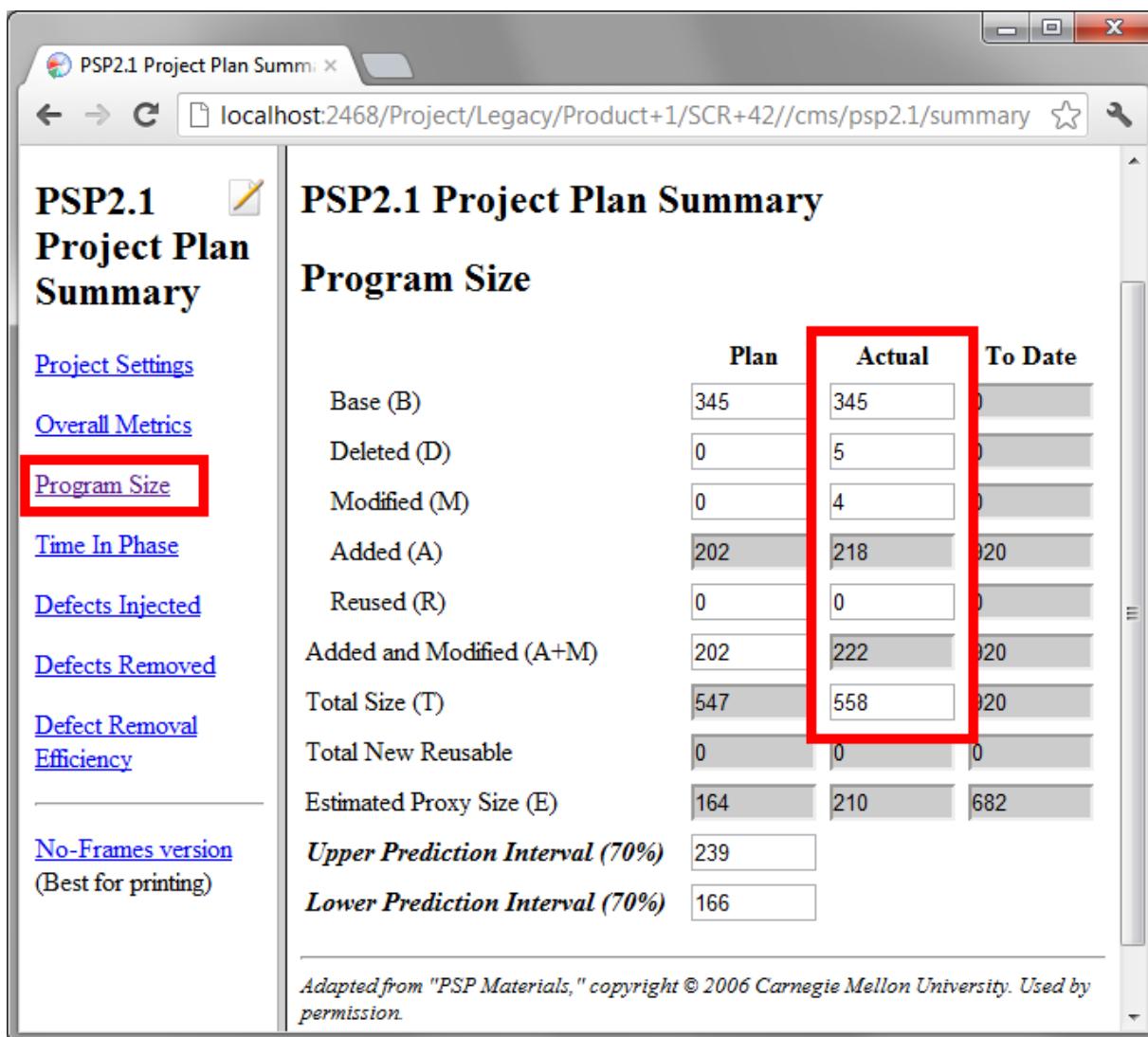
When you use the full-featured LOC Counter Tool, the top of the report will also show summary tables listing the files that were added, modified, or deleted, with individual LOC counts for each file.

4.1.4. Entering PSP Size Data

If you are using the PSP [Size Estimating Template](#), you can drag items from the LOC Report summary tables and drop them onto rows of the Size Estimating Template. The actual LOC for the file in question will be applied to the target row in the Size Estimating Template. (This feature is most useful if your conceptual design lists items at a file-level granularity or above. If you prefer to create separate rows in the Size Estimating Template for each added or modified method, you will need to perform your own manual accounting of the LOC for each Size Estimating row.) You can drag several items from the LOC report onto the same row of the Size Estimating Template, and the actual LOC contributions from each file will be summed. If you drop a file onto the wrong row, you can drag it to a different row or delete it entirely.

Entering actual LOC on the Size Estimating Template is a good way to improve your planning performance. As you apply the actual LOC to the rows of the Size Estimating Template, compare the actuals to your original plan. The differences can help you to understand the types of planning errors you typically make, which can help you to improve your plans in the future.

However, the data on the Size Estimating Template is **not** the official source of information for computing the size of a PSP program. The official source is the Actual Added and Modified field in the "Size" section of the PSP Project Plan Summary form.



The screenshot shows the PSP2.1 Project Plan Summary window. On the left, there's a sidebar with links like Project Settings, Overall Metrics, **Program Size** (which is highlighted with a red box), Time In Phase, Defects Injected, Defects Removed, Defect Removal Efficiency, and a No-Frames version link. The main area is titled 'PSP2.1 Project Plan Summary' and 'Program Size'. It contains a table with columns 'Plan', 'Actual', and 'To Date'. The 'Actual' column is highlighted with a red box. The data in the table is as follows:

	Plan	Actual	To Date
Base (B)	345	345	0
Deleted (D)	0	5	0
Modified (M)	0	4	0
Added (A)	202	218	320
Reused (R)	0	0	0
Added and Modified (A+M)	202	222	320
Total Size (T)	547	558	320
Total New Reusable	0	0	0
Estimated Proxy Size (E)	164	210	682
<i>Upper Prediction Interval (70%)</i>	239		
<i>Lower Prediction Interval (70%)</i>	166		

Adapted from "PSP Materials," copyright © 2006 Carnegie Mellon University. Used by permission.

The actual Added and Modified field is calculated from the values that appear in the actual Base, Deleted, Modified, Reused, and Total Size fields on the Project Plan Summary form. During postmortem, you should open the Project Plan Summary form and enter these values. This task is generally very fast and easy, because these aggregate numbers are readily available in a table near the top of the LOC Counter Report.

To receive good analyses from the PSP, it is **very important** for the "Actual Added and Modified" field to contain the right size. Therefore, you should always open the PSP Project Plan Summary form during the Postmortem phase of each PSP project and enter these values, even if you don't take the time to fill out the Actual columns of the Size Estimating Template.

4.1.5. Advanced Usage

Several configuration settings are provided to control the behavior of the LOC counter. For more information, see the [Advanced Configuration Settings](#) help topic.

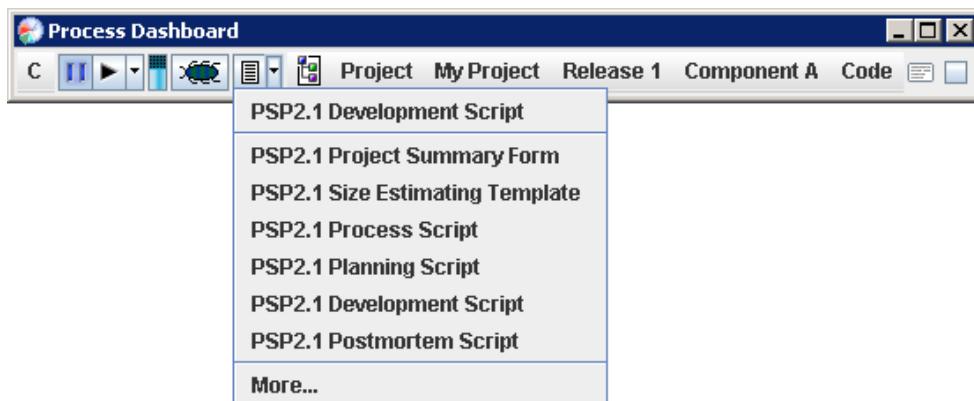
The LOC counter that is built into the Process Dashboard is extendable to allow user defined LOC counting rules. For more information, see the [custom LOC counter](#) help topic.

4.2. Data charts and reports

The Process Dashboard automates many reports and charts that can be quite useful in a project postmortem.

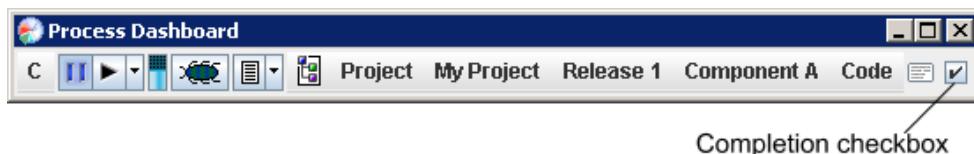
4.2.1. The Project Plan Summary

The main report that contains the most useful postmortem information for your project will probably be the Project Plan Summary form. The Project Plan Summary can generally be produced by navigating to the project for which you would like the report, clicking the script button, and selecting the option for the Project Summary form.



4.2.2. Marking your project complete

It is important to mark your project as complete. This lets the dashboard know that you are finished with the project, and it can use the project data to calculate the various items in the reports and charts. You can edit the project completion date via the appropriate field on the Project Plan Summary. If you haven't edited the project completion date field on the Project Plan Summary, marking all the phases in the project complete will mark the project complete as well.



4.2.3. The Time Log and Defect Log

HTML versions of the Time Log and the Defect Log are also available from the scripts button. If you navigate to the main Process script for any of the PSP processes (usually by following a link marked "top" near the bottom of the various reports and forms), you will find links named "Time Log" and "Defect Log" in the text of the script. These will bring up a read-only version of the two logs.

4.2.4. Graphical charts

The dashboard also makes many graphical charts of various kinds available. Choosing Data Analysis from the C menu will bring up a split view that contains an index of various chart types on the left and thumbnails of the charts available on the right.



Clicking any of the thumbnail charts will bring up a full-sized version of the chart. When viewing a full-sized chart, resizing your browser window will cause the chart to resize as well. Finally, clicking on a full-sized chart will display a table of the data used to draw that chart.

The dashboard provides charts that allow

- Defect analysis - rates of injection and removal of defects in the different process phases, defect injection and removal percentages in the different process phases, and pie charts showing the distribution of defect types.
- Plan analysis - comparisons of planned time and size with actual planned time and size, planning error of both time and size, and breakdowns of the amount of time spent in the different development phases.
- Process analysis - comparisons of productivity rates, defect removal yields, and appraisal to failure cost ratios.
- Quality analysis - various looks at the quality metrics of defect removal yield, failure and appraisal cost of quality, and LOC reviewed per hour rates.
- Defect Densities and Fix Times - tables of data showing where defects were injected and removed in various programs, and the relative costs of removing them.
- Defect Phases/Types - analyses of the various types of defects, and where they are most often removed.
- Pareto - shows defects organized by type sorted by number and by fix time.
- Custom - allows you to produce customized charts and reports based upon your historical data.

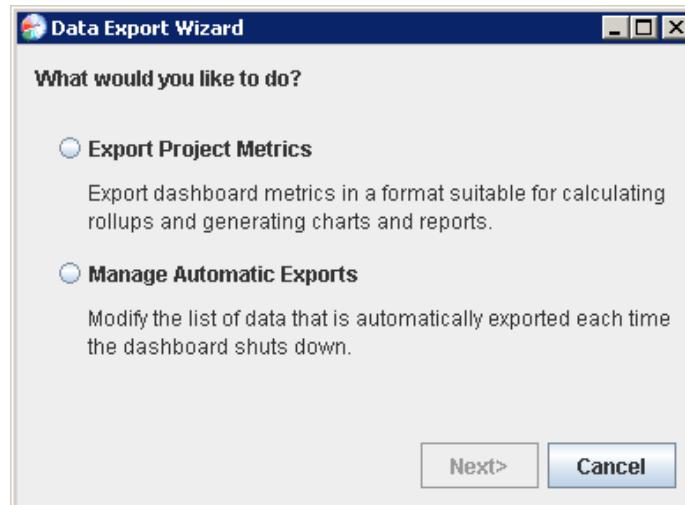
4.3. Exporting your data

4.3.1. Exporting Team Project Data

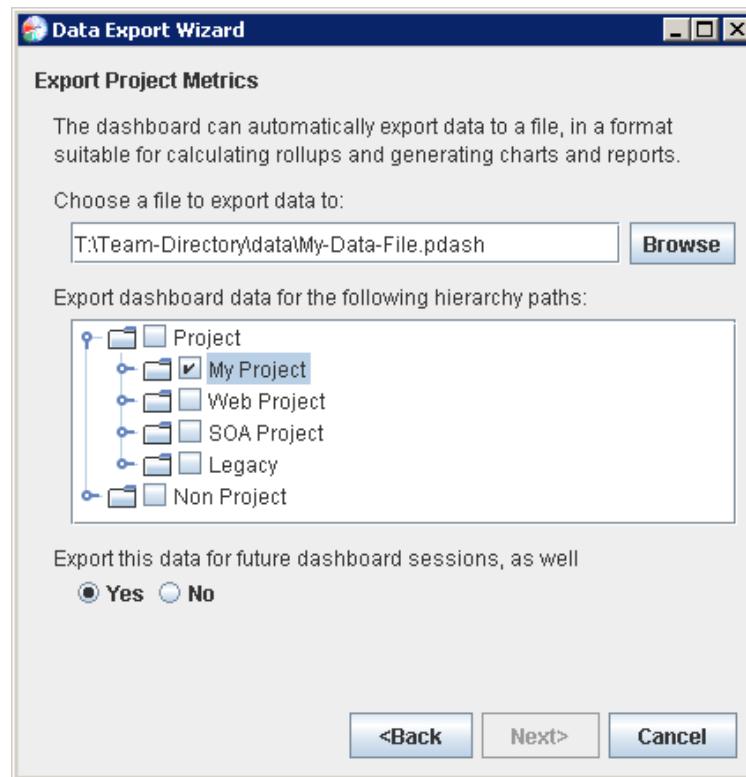
If you are participating on a team project, the dashboard will automatically export your team project data on a periodic basis. However, there may be times that you wish to publish your latest data immediately (for example, if you are on your way to a team meeting). Just click the script menu and choose the "Export My Data Now" option.

4.3.2. Advanced: Exporting Non-Project Data

If you have been collecting data that is not part of a team project, you can export that data if you wish. Just choose "Tools → Export" from the [configuration menu](#).



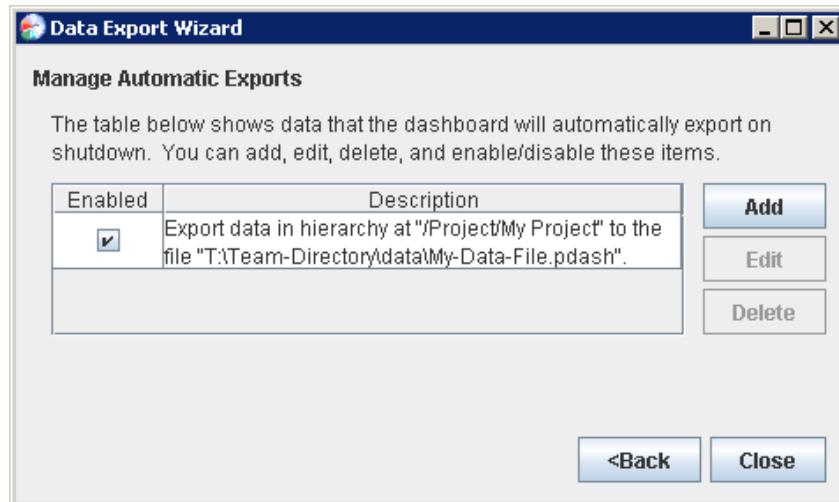
To export data, choose the "Export Project Metrics" option. The following page will be displayed:



This screen allows you to choose the file where data will be exported, and the projects in your hierarchy that should be included in the export. When you click "Next", data for the selected projects will be exported to the designated file. This will include metrics data (e.g.

the data visible on the plan summary form), defect data (e.g. the data visible in the defect log), and earned value data (if any of the selected projects are contained in any earned value task lists).

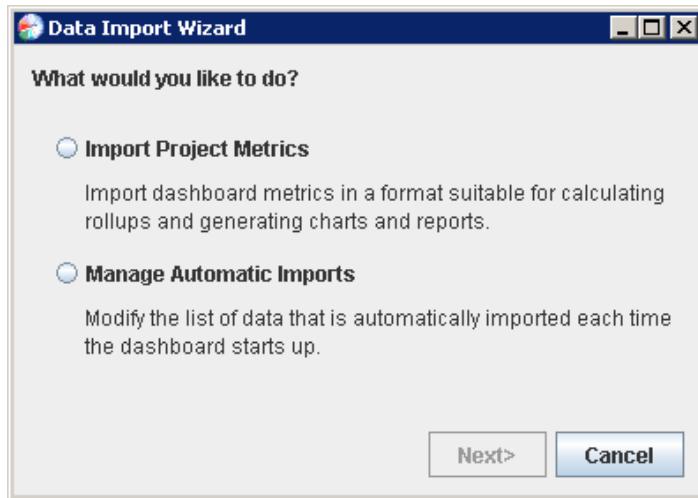
If you would like this data to be exported automatically each time the dashboard shuts down, choose "Yes" on the final checkbox. This will add this export operation to a permanent list of automatic exports. You can view that list, and alter its contents, by choosing the "Manage Automatic Exports" option on the initial page of the Data Export Wizard.



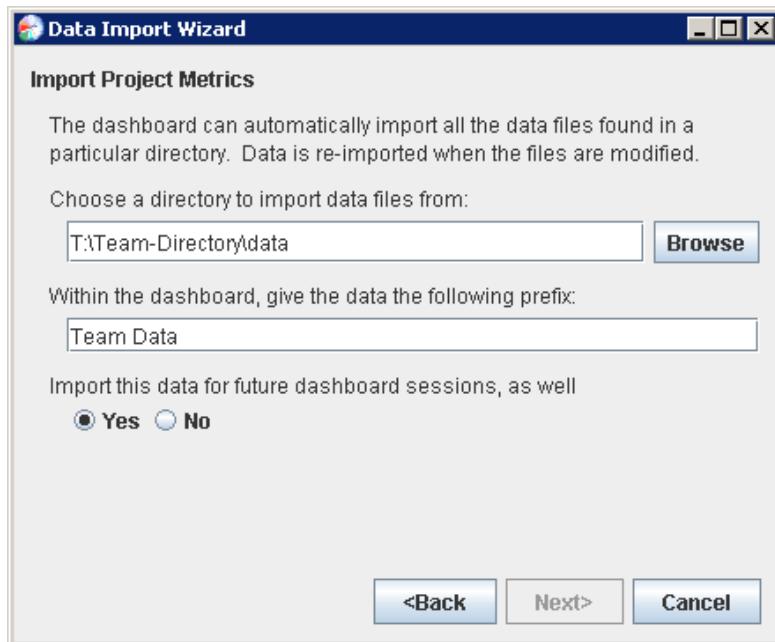
Advanced: Individuals who are interested in viewing/reusing the data in these exported files are free to do so; the export file is a simple ZIP archive with XML files inside. Keep in mind, however, that this file format is subject to change in future versions of the dashboard.

4.4. Importing data

Files that have been exported by the dashboard (as described above) can be imported by the Data Import Wizard. To open the Data Import Wizard, choose "Tools → Import" from the [configuration menu](#).



To import data, choose the "Import Project Metrics" option.



This screen allows you to choose a directory containing files that should be imported. The directory will be scanned for files previously exported by the dashboard (such files should have the ".pdash" suffix). The data in the files will be imported into the dashboard's data repository with the prefix chosen.

Metrics data imported in this way will be visible to tools like [PROBE](#), the [Data Analysis Center](#), and [data rollup templates](#). Earned value data imported in this way can be added to [earned value roll-ups](#). This provides many possibilities for rolling up data to the team level, and for analyzing data trends across projects.

If you would like the data to be imported the next time the dashboard starts, select Yes on the final checkbox. This will add the import operation to a permanent list of automatic imports. You can view that list, and alter its contents, by choosing the "Manage Automatic Imports" option on the initial page of the Data Import Wizard.



4.5. Exporting data to Excel (PC only)

Many of the reporting forms in the dashboard have links that say "Export to Excel...". When this link is clicked, an Excel sheet will come up that contains the data from that form.

By default, the data will be exported **dynamically** as a linked workbook. When the workbook opens, Excel will connect to the dashboard and retrieve the data items. At any time, you can ask Excel to refresh the data by right-clicking on the data and choosing "Refresh." This dynamic refresh behavior can be very powerful. For example, you can create additional sheets in the workbook to analyze and chart the raw data, then save the workbook. In the future, you can make sure the dashboard is running, reopen this workbook, and ask Excel to refresh the data. Excel will reload the raw data and your charts will update.

In some web browsers, that dynamic behavior may not work properly. In that case, you can try switching to a **static** export. The static export will show the values of process variables at the instant that the export was done. For instructions on configuring for static exports, see the [related FAQ](#).

The "Export to Excel..." option will appear on

- [dashboard forms](#) such as the Project Plan Summary form
- tables that are displayed by clicking on a graphical chart (see [Data Charts and Reports](#) for more information)
- the table that is displayed when you click on the "historical data" link from a PSP1.0 or higher process script
- the HTML time log
- the HTML defect log
- the HTML report that is displayed when you click the "Report" button on the [task & schedule tool](#)
- ...and many other reports

4.6. Importing Historical PSP Data

If you have PSP data from historical projects, and you would like to import them into the dashboard so they contribute to your "To Date" data, PROBE calculations, and "Data Analysis" charts/reports, you can use the "Import PSP Data" template. To enter your historical data into the dashboard, follow these steps:

1. Open the [Hierarchy Editor](#).
2. Highlight a node in your hierarchy like "Non Project", then click "Edit → Add Template → Import PSP Data". (*If there is no "Import PSP Data" option in your "Add Template" menu, then you may need to [install the PSP scripts and forms](#).*) A new node will be created in your hierarchy. Give it a descriptive name; you might choose names like "Program 1" if you are entering data from a PSP course.
3. Repeat step 2 for each PSP project whose data you wish to import.
4. Choose "File → Save", then "File → Close" to close the hierarchy editor.
5. Navigate to one of the projects you just created, then click the script button. An abbreviated project plan summary will appear.
The fields on this summary are editable; enter your historical data for the given project.
6. Repeat step 5 for each imported project.

The data entry is a manual process, but not a time-consuming one; it should be possible to enter the data from an entire PSP course in about 15 minutes.

5. Frequently Asked Questions

This is a list of Frequently Asked Questions about the Process Dashboard

5.1. Installation issues

- [My web browser displays an error message when I try to view a script/form/chart/report/etc. What is wrong?](#)
- [My web browser is able to show the forms, but none of the entry blanks have any information. What is wrong?](#)
- [I have several versions of the Java Runtime Environment on my computer. How can I tell the dashboard which one to use?](#)
- [Our Team Dashboard is displaying the "toolbar" user interface designed for personal data collection. How can I fix this?](#)

5.2. Online forms

- [If I have a dashboard script or form open, and I click the script button to open another form, the new form appears in my existing browser window, replacing the old form. Why can't I have two forms open at once?](#)
- [All my data elements say "NO CONNECTION". What is wrong?](#)
- [Some of my data elements say "#VALUE!" or "#DIV/0!". What is wrong?](#)
- [I tried to export my data to Excel with the link on one of the forms, but I just got a small text file. What is wrong?](#)
- [Why is there no "submit" button on the HTML forms?](#)
- [I've started the timer, but none of the numbers on my reports are changing. How can I know that it is working?](#)
- [I made a mistake in one of my numbers on the Project Plan Summary form, but the form won't let me change the numbers! What can I do?](#)
- [What is the significance of the different text colors on the PSP process scripts and forms?](#)
- [What is "Prototypical PSP Data?" I'm seeing it in some of the charts and reports.](#)
- [I click on a document hyperlink in a process script, but the document never opens.](#)
- [I'd like to make a minor change to a process script someone else wrote. Can I do that?](#)

5.3. Files

- [Where is my PSP data stored?](#)
- [How can I backup my psp data and/or move it to another computer?](#)
- [I got a warning box that claims that my data files have been corrupted! Can I get my data back?](#)
- [I'm on a Unix/Linux system, and the dashboard is cluttering my home directory with lots of files! How do I fix this?](#)

5.4. Other questions

- [Can I hide my finished projects from the hierarchy browser \(and other screens\)? They are starting to pile up and are taking up a lot of space.](#)
 - [I have a project in my hierarchy that doesn't have much to do with my current development habits. Can I mark it as an "outlier" so that it doesn't corrupt the Planning and PROBE estimates?](#)
 - [Since the dashboard has a web server in it, do I have to worry about internet worms and viruses attacking my machine?](#)
 - [When the dashboard starts up, it's attempting to make a connection to the internet. Why is it doing this?](#)
 - [What happened to the DCR and SCR processes?](#)
 - [I would like to integrate the dashboard more closely with my IDE. Is it possible to programatically start/stop the timer, and change the active hierarchy path?](#)
 - [Does the Process Dashboard support teams?](#)
 - [Where can I get the latest information about the Process Dashboard?](#)
 - [I think I've found a bug. Whom do I contact?](#)
-

My web browser displays an error message when I try to view a script/form/chart/report/etc. What is wrong?

The dashboard actually contains a very small HTTP server that provides all the web content and thus the content cannot be displayed when the dashboard is not running. If you get an error message like "The page cannot be displayed - cannot find server," check to make certain that the dashboard is running.

If the dashboard **is** running, and you have **never** been able to view a script/form/chart/report, it may be that you need to adjust your proxy settings. Many people work at companies where they do not have a direct connection to the internet, but instead must connect to the web via an HTTP proxy server. If this describes you, you may need to tell your browser to bypass the proxy for the address "localhost". The instructions below may not apply to your browser word-for-word, but should hopefully get you pointed in the right direction:

- For Internet Explorer, choose "Tools → Internet Options". Click on the "Connections" tab, then on the "LAN Settings" button. Make certain the "Bypass proxy server for local addresses" box is **checked**.
-

My web browser is able to show the forms, but none of the entry blanks have any information. What is wrong?

The dashboard uses web browsers in unique and innovative ways. Unfortunately, some browsers aren't up to the task. If you can see the HTML forms but they don't seem to be talking to the Dashboard, then it is likely that your browser (or the Java environment within your browser) is encountering problems.

Your first step should be to ensure that you are using one of the browsers recommended for use with the dashboard. (See the [installation help topic](#) for more information.)

If you are using one of the approved browsers and you still have this problem, please send us an email at processdash-devel@lists.sourceforge.net and we will attempt to help you with the problem.

I have several versions of the Java Runtime Environment on my computer. How can I tell the dashboard which one to use?

(These instructions apply to Windows.)

1. Right-click on the icon you use to run the dashboard, and choose Properties
2. The "Target" field will contain "C:\Program Files\Process Dashboard\pspdash.jar". Change it to read (all on one line):

```
"C:\Program Files\JavaSoft\JRE\1.3.1\bin\java" -jar "C:\Program Files\Process Dashboard\pspdash.jar"
```

(Of course, you may need to modify the path to the "java" program if it is slightly different on your computer.)

Our Team Dashboard is displaying the "toolbar" user interface designed for personal data collection. How can I fix this?

In most cases, when you create a team dashboard or personal dashboard, it will automatically be configured with a flag indicating the dataset type. However, if you have an older Team Dashboard that contains a mixture of team projects and accidentally created personal projects, the dashboard may not be able to infer the correct dataset type, and it may display the "toolbar" user interface that is designed to help individuals collect data. If this occurs, you can fix the problem by choosing "C → Tools → Preferences → Advanced" and setting the "datasetMode" parameter to "team".

If I have a dashboard script or form open, and I click the script button to open another form, the new form appears in my existing browser window, replacing the old form. Why can't I have two forms open at once?

You can have as many scripts and forms open as you like! The behavior you are seeing is a Microsoft "feature" designed to reduce clutter on your screen. Fortunately, you can alter that behavior. The instructions below may not apply to your browser word-for-word, but should hopefully get you pointed in the right direction:

1. In Internet Explorer, choose "Tools → Internet Options".
2. Click the "Advanced" tab.
3. Remove the checkmark next to the option, "Reuse windows for launching shortcuts."

All my data elements say "NO CONNECTION". What is wrong?

The forms always display "live" data - this allows them to auto-recalculate numbers (like a spreadsheet does) when you edit data. It also allows changes made in one browser window to appear immediately in other browser windows. To make all this happen, the browser maintains an open connection to the dashboard, and passes data back and forth across this connection.

If you are viewing a form, and you close the main dashboard window, this open connection is closed as well. When this occurs, the form immediately displays "NO CONNECTION" to let you know that your changes are no longer making it back to the dashboard's data repository. To fix the problem, you must

1. restart the dashboard
2. click the refresh/reload button on your browser.

If you have just opened a page containing a lot of data, the dashboard may need time to perform the calculations. It will display "CALCULATING" while the data is prepared to let you know that the form is not ready yet.

Some of my data elements say "#VALUE!" or "#DIV/0!". What is wrong?

"#VALUE!" indicates that the dashboard cannot calculate a value because the formula references a value that is missing (displayed as "?????") or otherwise invalid (for example, a value like "foo" when a number was expected). When you see "#VALUE!" in a dashboard form, it almost *always* means that there is a "?????" field somewhere that you still need to fill in. If you fill in all the "?????" fields and the "#VALUE!" field does not recalculate, check to ensure that the calculation is not [frozen](#).

"#DIV/0!" indicates that the dashboard encountered a divide-by-zero error when calculating a particular data value. Many times this is due to having a very small number of usable data points recorded by the dashboard. For instance, if you start the dashboard for the first time and try to look at Process Summary forms for a PSP2.1 process, the "% To Date" fields will be filled with "#DIV/0!". This is because the total amount of time recorded in the tool is 0. The percentage calculation attempts to divide by zero and returns the error.

Another problem can contribute to this condition as well. If you are not in the habit of clicking the little "completion" checkbox at the right end of the dashboard, some (or all) of your data will be ignored by the dashboard when creating reports and graphs. The reporting features of the dashboard only report tasks/projects that are marked as complete. See the [Using the Completion Checkbox](#) help topic for more information.

I tried to export my data to Excel with the link on one of the forms, but I just got a small text file. What is wrong?

For the "Export to Excel" functionality to work seamlessly, you must have the ".IQY" file type properly set up on your system. This setup is normally performed by Microsoft Office during the installation process for Excel (although, notably, certain releases of Office 97 do not perform this setup).

If the IQY file type is not properly set up, clicking the "Export to Excel" links may instead cause Notepad to appear with a short, three-line text file. Alternatively, that three line text file might appear in an Excel spreadsheet.

If you encounter such problems, the simplest fix is to add the following setting to the "Advanced" section of the [preferences editor](#):

```
excel.exportMethod=mime
```

And data will be exported to Excel [statically](#).

Why is there no "submit" button on the HTML forms?

The HTML forms used by the dashboard are continually "live". There is no need to press a submit button to send your data to the dashboard. Simply enter your data where necessary, and the dashboard will record it. However, after editing a data field, it is usually necessary to leave that field for your changes to take effect. You can press the TAB button or click somewhere else on the page. If the data you are changing is related to other data items on the page, you should see the other data elements update. For more information, see the [Entering process data](#) help topic.

I've started the timer, but none of the numbers on my reports are changing. How can I know that it is working?

When the timer is running and you first navigate to an activity, the dashboard will not create an entry in the time log until you have spent a full minute there; this is by design. For a more detailed explanation, see the help topic for the [play/pause button](#).

Also, the dashboard will make a quiet "tick" sound whenever the timer starts and when a new project or project phase is chosen by the hierarchy menus when the timer is running. If you do not hear this sound, check the state of the play/pause button. The current state of the timer can always be determined by positioning the mouse pointer over the play/pause button. A tooltip help will pop up containing the current state.

I made a mistake in one of my numbers on the Project Plan Summary form, but the form won't let me change the numbers! What can I do?

The dashboard will "freeze" your planning data once you have completed the planning phase by clicking the completion checkbox. Also it will freeze all the data when the project is marked as complete. You can make these numbers editable by marking either the project or the planning phase incomplete again. For more info on freezing data see the [data freezing](#) and the [completion checkbox](#) help topics.

What is the significance of the different text colors on the PSP process scripts and forms?

In his book *A Self-Improvement Process for Software Engineers* where he outlines the Personal Software Process, Watts Humphrey introduces the various process elements for doing PSP work in a graduated manner. Each PSP process is built upon the previous one. As a way of showing which process elements were added to the previous process definition, the forms in the book highlight the new process elements in ***bold italics***. These highlighted elements return to normal text at the next process level. In our attempt to make our online forms as identical to the PSP forms as possible, we have included that detail.

However, the Process Dashboard makes certain steps on the scripts and forms unnecessary (e.g. instructions to sum up various data items -- the dashboard sums them for you). These unnecessary process steps are displayed as greyed-out text.

Also, there are places where the dashboard *slightly* changes the way certain steps are performed (e.g. instead of performing a PROBE calculation, the PROBE tool should be used). Green text is used to indicate this slight departure from the *true* PSP scripts and forms.

What is "Prototypical PSP Data?" I'm seeing it in some of the charts and reports.

When you first begin using the dashboard, you have no historical data. As a result, previous versions of the dashboard would display "#DIV/0!" in the "Plan" column of your first project if it was based upon anything other than a PSP0 process.

Beginning with version 1.6.2, the dashboard will detect that you have no historical data, and introduce an example dataset based upon industry data and best practices. The dashboard uses this dataset to calculate the plan for your first PSP project.

As soon as you complete at least one PSP project, this example dataset will disappear. In the meantime, it may appear in various dashboard charts and reports.

I click on a document hyperlink in a process script, but the document never opens.

For maximum portability, the document templates provided with the dashboard are rich text files (.rtf). Some browsers don't know what to do with a rich text file, and so they prompt you to save it (not very helpful). To solve this problem, you must:

1. Obtain a program capable of editing rich text files, and install it on your machine. ("Open Office" is one possibility; it is free and runs on Linux. Visit <http://www.openoffice.org/>)
 2. Configure your web browser, and tell it that files ending with ".rtf" and/or files of mime type "application/rtf" should be opened with the program you just installed.
-

I'd like to make a minor change to a process script someone else wrote. Can I do that?

Yes you can! Please see the section on [overwriting process scripts](#) in the help topic that describes how the dashboard finds the script files.

Where is my PSP data stored?

The dashboard stores all of your data in a single directory, in plain text files. This directory is often referred to in this documentation as the "data directory".

You can find out the exact location of your data directory by choosing "Help → About Process Dashboard" from the configuration menu, then clicking on the Configuration tab.

With the current default configuration, data is always stored in the "current working directory" when the dashboard is run.

On Windows systems, the "current working directory" is set by the icon you use to run the dashboard. Right-click on that icon and choose "Properties"

- Typically, the data directory will be shown in the field labelled "Start in:". Altering this value will change the location where the dashboard looks for your data.
- Alternately, the location may be specified as an argument at the end of the "Target" field. In this case, the "Target" field will have an argument of the form "`-location=<some data location>`". Changing this argument will point the shortcut to a different data location.

On Mac OS X systems, the data directory is set by the Application icon you use to run the dashboard. Ctrl-click on that icon, and choose "Show Package Contents." Then find the file "Resources/dataLocation.txt". You can edit this file to change the location where your data is stored.

On Unix/Linux systems, the current working directory is controlled by the shell script or icon you use to launch the dashboard. See [this FAQ question](#) below for more information.

It is possible to have multiple, independent data directories all on the same machine; just make copies of the icon you use to start the dashboard, and configure each icon with a different data location.

How can I backup my psp data and/or move it to another computer?

If you want to backup your psp data, simply back up the contents of your [data directory](#). To move your psp data to a different computer,

- stop the dashboard if it is running
 - copy the contents of your data directory to a directory on the new computer
 - run the dashboard on the new computer, using the chosen directory as the data directory.
-

I got a warning box that claims that my data files have been corrupted! Can I get my data back?

If you did not continue running the dashboard when the dialog warned you that doing so could be quite dangerous, then yes, you probably can.

If you see this message, realize that the dashboard is being overly cautious with your data. As long as you didn't plow on ahead and run the dashboard with the questionable files in place, no data has been lost yet. The best steps to take next are:

1. Examine the corrupt files mentioned in the warning message. If they are of zero length, then there is most likely nothing to worry about. Just delete the zero length files and restart the dashboard.
2. If the files contain something, examine their contents. Removing the "ttt" at the beginning of the filename should indicate what *real* file was being created at the time the problem was detected. See if the temporary files contain any data that is missing from the original file, and copy it over if necessary. Finally, delete the temporary files and restart the dashboard.

Even though the Dashboard has been designed to be safe with your precious process data, it is still a good idea to back up the Dashboard's data directory every so often. See the question about [Backing up your Dashboard data](#) for more information.

I'm on a Unix/Linux system, and the dashboard is cluttering my home directory with lots of files! How do I fix this?

When you start the dashboard (whether on Linux, Windows, or any other operating system), it uses the "current working directory" for data storage.

On Windows, the installer program automatically creates a shortcut for the user which launches the dashboard with the appropriate current working directory, as selected during the installation process.

Unfortunately, the installer is not able to create shortcuts on all Linux/Unix platforms yet, so it may up to you to create an appropriate launcher manually. The dashboard does its best to help you out, by creating a shell script in your chosen data directory called "run-dash.sh". Take a look at the contents of that shell script. Embedded comments in that file include some helpful information; for example, If java and/or firefox are not in your path, you may need to edit a particular line of the script. With those edits, you should be able to use the script to launch the dashboard.

If you have already started using the dashboard and you've already created projects in your hierarchy that you want to keep, you can move them by carefully following the steps below:

- **Shut down the dashboard if it is running. Do not skip this step!**
- Find files in your home directory matching any of the following patterns, and move them into your chosen data directory:

global.dat	time.log	.pspdash
###.dat	timelog.xml	pspdash.ini
###.def	timelog2.xml	state

- Edit the `run-dash.sh` file in your data directory as necessary. Then create a launcher for `run-dash.sh` within your desktop environment (e.g. Gnome / KDE).

Can I hide my finished projects from the hierarchy browser (and other screens)? They are starting to pile up and are taking up a lot of space.

Unfortunately, No. We have this as a feature request.

In the meantime, the Cut and Paste features in the hierarchy browser will allow you to move your completed projects away from the part of the hierarchy where you are doing your active development. You could create a node at the top of the project hierarchy called "Archived Projects" and Cut and Paste your finished projects as children under it. They would still count towards your PROBE Estimates and your Planning data. See the [hierarchy editor](#) help topic for more information.

I have a project in my hierarchy that doesn't have much to do with my current development habits. Can I mark it as an "outlier" so that it doesn't corrupt the Planning and PROBE estimates?

There are basically two concerns when in this kind of situation.

1. Removing the project from PROBE Estimates

2. Removing the project from Planning and To Date calculations

The most important time to identify a project as an outlier is during PROBE-style linear regression. This is easy to do; when using the PROBE tool, you can click on the "Chart" button to visually identify outliers, and the "Filter" button to remove any outliers from the regression calculation. The PROBE Wizard will also let you remove various data pairings from the regression calculation, but it won't show you a graph and thus isn't as intuitive. See the [PROBE Tool](#) and [PROBE Wizard](#) help topics for more information.

To remove a project's effects from Planning and To Date calculations, you can make use of multiple data sets. By defining a customized data set for your current project, you can isolate only the data you know is applicable. See the [multiple To Date data](#) help topic for more information.

A less desirable but simpler way to remove a project's effects would be to mark the project as incomplete. The To Date, Planning, and PROBE calculations only use completed projects for their data sets. The drawback with this approach is that you will lose the accurate completion date for the project that you mark incomplete.

Since the dashboard has a web server in it, do I have to worry about internet worms and viruses attacking my machine?

A good question! The short answer: the web server built into the dashboard is NOT susceptible to any existing worms or viruses, and the default dashboard settings make it impossible for other computers to connect to your dashboard.

The longer answer: the web server in the dashboard is custom code, written from scratch specifically for the dashboard. Because it shares no common code with any other web server, viruses and worms which have been written to attack other web servers (such as Microsoft's IIS web server) will not affect it in any way.

We cannot guarantee that the dashboard web-server code is defect-free, and impervious to attacks by hackers. But those hackers would have to write a virus or worm to specifically target the dashboard. The dashboard currently has less than 10000 users worldwide, so I would imagine it to be a fairly unlikely target.

Fortunately, because the dashboard is written in Java rather than C/C++, it is not susceptible to many common hacking techniques like buffer overruns (a common worm infection strategy). This means that even if a hacker was interested in writing an attack targeted at the dashboard, the worst thing they could probably do is to crash your running dashboard program.

If this information doesn't allay your concerns, you can use the "http.allowRemote" setting to ensure that it is impossible for other computers to connect to the dashboard (this is the default). See the [advanced settings](#) help topic for more information.

When the dashboard starts up, it's attempting to make a connection to the internet. Why is it doing this?

Like many modern software programs, the dashboard provides an "update notification" feature. Once a month, this feature checks to see if an updated version of the dashboard is available. If you have installed add-on process sets, it also checks to see if updated versions of these are available.

If new versions are available, the dashboard will display an alert, listing the items which have been updated. This alert has a checkbox which reads "Don't perform a monthly check for new releases." If you check this box, the dashboard will never perform this check again. You can also disable this check manually by unmarking the appropriate box in the [preferences editor](#).

In the normal case, this "update check" will succeed, and the dashboard will make a note not to check for another 30 days. However, the update check might fail if no network is available (for example, because you are running the dashboard on a laptop that isn't connected to the network). As a result, if the update check fails for any reason, the update checking logic will try again the next time you start the dashboard.

If you see the dashboard attempt to check every time it starts up, this means that the update check is never completing successfully. This could occur if your personal firewall is blocking the connection, or if you are behind a password-protected corporate firewall. In either case, you can turn off this feature entirely as described above.

What happened to the DCR and SCR processes?

The DCR and SCR processes were shipped with version 1.1 of the Process Dashboard. They were later removed from the main

dashboard and distributed as add-on process sets. As of version 1.4.1, they are no longer available.

Explicit written permission from Addison-Wesley is required to modify PSP scripts or distribute modified PSP scripts. Since the DCR/SCR process scripts were derived from the copyrighted PSP process scripts in *A Discipline for Software Engineering*, they fall into this category. We have not attempted to obtain that permission from Addison-Wesley, so these processes are no longer available.

I would like to integrate the dashboard more closely with my IDE. Is it possible to programmatically start/stop the timer, and change the active hierarchy path?

Yes! If you issue an HTTP request of the form:

`http://localhost:2468/control/startTiming.class`

the dashboard timer will start. Similarly, an HTTP request of the form:

`http://localhost:2468/control/stopTiming.class`

will pause the timer. And

`http://localhost:2468/Non+Project/foo//control/setPath.class`

will change the current hierarchy path to "/Non Project/foo". If you want to change the current hierarchy path **and** start the timer, use

`...//control/setPath.class?start`

Note: these commands will have no effect unless the HTTP request originates from the same machine where the dashboard is running.

Does the Process Dashboard support teams?

Yes, the dashboard *does* contain powerful features that make it possible to roll up individual data to the team level. See the [Team Use](#) help topic for more information.

Where can I get the latest information about the Process Dashboard?

The latest developments in the development of the Process Dashboard, including the latest version, can be found at <http://www.processdash.com>

I think I've found a bug. Whom do I contact?

If you like, you can fill out a bug form on our web site at <http://www.processdash.com/bugTracker>. The simplest way to do this is just to pick "Help" → "Submit bug report" from the dashboard's "C" menu. This will open your web browser pointed right to the bug form.

If you would rather send us an email, write to processdash-devel@lists.sourceforge.net

6. Advanced Topics

Want to try your hand at writing your own processes? Want a little more control over how the Dashboard operates? Want to do things that aren't quite obvious? This is the place to look.

This section will tell you how you can change the default behavior for various features of the dashboard, track multiple sets of To Date data, and how to create your own processes for use in the Dashboard.

6.1. Team Use

The Process Dashboard includes extensive functionality which can be used by high-maturity project teams. This functionality will be most useful and intuitive to projects that are applying the Team Software Process^(SM). However, since the dashboard's team functionality is general-purpose, it can be used effectively by non-TSP^(SM) and non-software teams.

For more information and step-by-step instructions, see the [Team Use](#) section of the Users Manual.

6.2. Backing up Dashboard Data

The metrics data you collect is very valuable information. Accordingly, the dashboard provides several mechanisms for backing up this information.

6.2.1. Manual Data Backups

At any time, you can manually save a backup of your data by choosing "C → Tools → Save Data Backup". This will prompt you for the name of a ZIP file, and will save a full backup of your data to the given file.

Optionally, you can choose to save your data as a "Process Dashboard Backup" file. This format contains the exact same data as the ZIP file, but may be useful in organizations that forbid the attachment of ZIP files to email messages.

Finally, you can choose to save your data as a "Redacted Process Dashboard Backup" file. This format allows you to select various categories of information that should be scrambled or removed from the data backup. This format can be useful if you need to protect the privacy of individuals or if you need to obscure proprietary information. However, some types of data cannot be redacted; so if the original data is highly sensitive, you should still use caution and review the resulting backup carefully before releasing it.

6.2.1.1. Viewing the data in a manual data backup

To view the data in a data backup, choose "C → Tools → Open Dataset", and a file chooser window will open. Choose a data backup file in ZIP or PDBK format, and a second Process Dashboard window will open, displaying the data from that file. The data you view in that second window is transient/temporary data extracted from the backup; any changes you make will be discarded when the window is closed.

In addition, the Process Dashboard "Tools for Team Leaders" include a program called the Quick Launcher. You can drag a data backup file onto the Quick Launcher, and it will open a Process Dashboard window displaying the data inside. This functionality works just like the Open Dataset option on the Tools menu, but does not require another Process Dashboard window to be open first.

6.2.1.2. Restoring data from manual data backups

A data backup is just a ZIP file, containing the data in your personal data directory (along with some additional information). If you have lost or corrupted your personal data and you wish to restore your data from a backup, take these steps:

1. Shut down the dashboard. **It is imperative that the dashboard NOT be running** while you perform the steps below.
2. Extract the contents of your data backup file into your personal [data directory](#), overwriting the files that are there.
3. Restart the Process Dashboard.

6.2.2. Automatic, External Backups

The mechanism described above is very useful for interpersonal communication. But for disaster recovery, it is important to have an automated backup strategy. If your computer and hard drive were to crash, you wouldn't want to discover that your most recent backup was several weeks old.

Thus, the dashboard has a mechanism for making backups on a regular basis. To configure an automatic backup, take these steps:

1. Click on the "C" menu and choose "Tools → Preferences".
2. In the User Preferences dialog, select the "Backups" category. A table will be displayed, showing a list of external locations where backups should be saved.
3. Click the "Add" button to add a new backup location. Specify the full name of a backup ZIP file where you would like data to be saved, then click OK.

Each time the dashboard is shut down, it will create a backup ZIP file and save it to the location you've requested. If you are in the habit of leaving the dashboard running overnight, it will also save this backup each evening between midnight and 1 AM. The backup ZIP file will be overwritten each time, so at any point in time the external backup file will contain a copy of your most recent data.

If you want to keep a history of backup files, include the word `%date` in the name of your backup file. (Make certain to type this exactly as written: a percent sign followed by the word "date" in all lower case.) The current date (YYYY-MM-DD) will be included in the name of the file, so backups from dates in the past will not be overwritten. This can be particularly useful for the Team Dashboard, as a way to maintain a record of how the team plan has changed over time. If you use this technique, keep in mind that no old backups will be discarded, so the list of backup files could grow very large over time. You may need to visit the directory periodically and purge old backups that are no longer needed.

The backups created by this process are identical to the backups created by the manual "Save Data Backup" operation. Thus, they can be viewed in the Quick Launcher, and the mechanism for restoring from a backup is the same.

6.2.2.1. Additional Uses

Many teams choose to use backup files for a secondary purpose. Specifically, a process coach, team leader, or PSP instructor may need to look at the data for a particular individual, so they can troubleshoot data entry problems or mentor the individual in their process improvement efforts. The automatic external backups and manual data backups provide a simple way to support this particular need.

Accordingly, your team may decide to agree upon a particular network directory, then have each individual save an automatic backup to that directory. This would provide a disaster recovery plan if a team member's computer were to crash. In addition, it could enable your coach or planning manager to examine data for a particular individual if necessary.

If you choose to follow this approach, consideration should be given to data privacy concerns. For example, you might consider configuring the permissions on the given network directory to prevent read access by arbitrary individuals. On the other hand, if your organization follows high-maturity practices and does not abuse personal data for inappropriate purposes, this may not be a significant concern. Each organization must make these decisions based on their organizational climate. If organizational data privacy concerns outweigh the benefits of information sharing, you may wish to lock down the permissions so that data backups can only be read by the individual who created them.

6.2.3. Automatic, Internal Backups

The mechanisms above can be very useful for disaster recovery, such as the loss of your hard drive. But occasionally you may encounter other problems. For example, you might accidentally delete a large portion of your task hierarchy. Or you might encounter a rare bug in the Process Dashboard that corrupts your data. In these cases, by the time you shut down the Process Dashboard, your external backup would already be overwritten with the problem data.

To address these types of problems, the Process Dashboard automatically maintains a set of internal backups that reflect changes made to your data over the past few weeks. Each time the dashboard starts up or shuts down, and at least once daily, the dashboard will zip up the files containing your data. This ZIP file is stored in a "backup" subdirectory underneath your [data directory](#).

If you look in this "backup" directory, you will typically notice many ZIP files. The most recent file is a **full** backup of your data, similar to a manual data backup. Earlier files are not full backups - they are incremental backups, containing only the files that changed between two points in time.

Incremental backups are not preserved forever. By default, the dashboard will retain backups for three weeks. (This timeframe can be changed via the [preferences editor](#).)

6.2.3.1. Restoring data from automatic, periodic backups

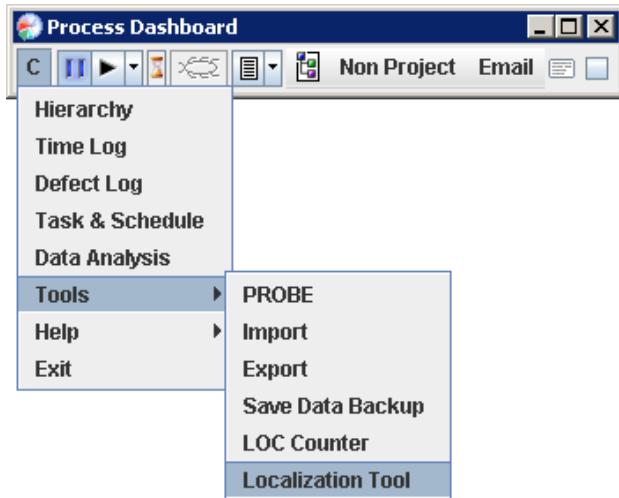
If you decided that your data had been corrupted in some way, and you wanted to restore your data to some earlier point in time, you would follow the steps below. (Make certain you understand these instructions completely before you begin, and **use extreme caution** when following them; incorrect actions could result in permanent data corruption! **Proceed at your own risk.**)

1. Shut down the dashboard. **It is imperative that the dashboard NOT be running** while you perform the steps below.
2. View the contents of your "backup" directory, and sort the files so they appear in chronological order. The names assigned to the files are designed to help you perform this sort correctly.
3. Use a ZIP extraction tool to extract the contents of the most recent ZIP file, placing the files into your data directory. This will overwrite files that are already present in that directory.

4. Next, extract the second-to-last ZIP file, also placing the results into your data directory, and overwriting files. Continue in this manner, working your way backward through the ZIP files, until you reach the file representing the moment in time you wish to restore to.
5. At this point, you may restart the dashboard. It should reflect the newly restored data, but please note the following:
 - Any data that was collected after your selected "restore point" will no longer be present in your dashboard. (After all, you've just intentionally restored the data to that earlier point in time.)
 - The very act of starting the dashboard will cause it to create a new incremental backup file, capturing the changes you made in steps 3-4 above. Thus, if you realize that you did not restore to the correct instant in time, you will need to start again at [Step 1](#), shutting down the dashboard, sorting the new list of backup files, and reextracting all of the incremental backups in order.

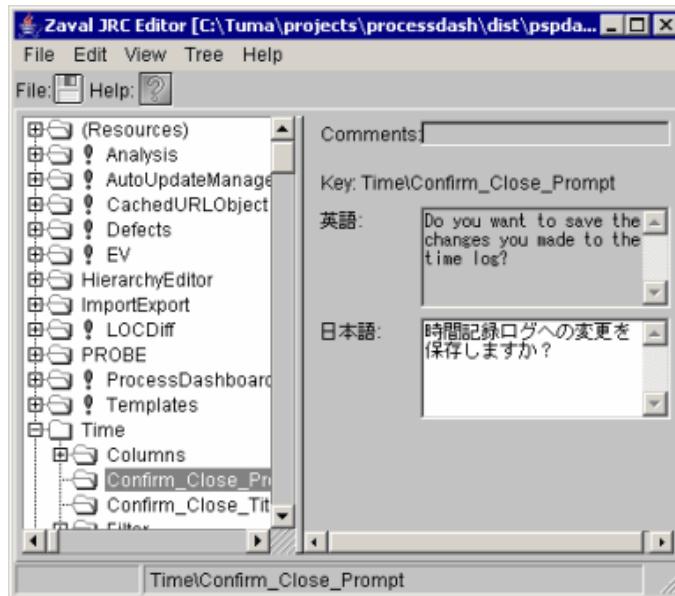
6.3. Localization Tool

The dashboard has many users around the world. To better support all our users, we are currently working to internationalize the dashboard. The Process Dashboard includes a Localization Tool that you can use to translate dashboard messages into your own language. You can open the Localization Tool by selecting "Tools → Localization Tool" from the [configure menu](#).



(Note that the menu option for the localization tool will not normally appear for English-speaking users.) The Localization Tool is based upon the useful Zaval JRC Editor written by Victor Krapivin and others (<http://www.zaval.org/products/jrc-editor/>).

The localization tool arranges translation keys hierarchically and displays the resulting tree on the left. Exclamation points indicate keys that still need to be translated. When you select a key in this tree, the right hand side of the screen displays the English message and a place for translation into your native language.



Some translation keys end with the string "_FMT". The English message associated with these keys will contain expressions like "{0}". These translation items are Java MessageFormat strings; at runtime, the "{0}" will be replaced with a dynamically generated value. To translate these items correctly, you will need to understand the syntax of MessageFormat strings. For more information on MessageFormat syntax, visit <http://java.sun.com/j2se/1.4.1/docs/api/java/text/MessageFormat.html>

Other translations may contain cross-references. A cross-reference is a translation key surrounded by curly braces, with a '\$' sign in front; an example would be \${some.other.key}. When the dashboard sees a cross-reference, it looks up the translation described by the named key, and replaces the reference with that translation.

Cross-references are designed to reduce the burden of translating the dashboard, without sacrificing expressiveness. For example, the

English word "Plan" may be used in two different places inside the dashboard, but it may have different meanings (for example, in English the word "Plan" might be used as a noun, a verb, or an adjective). In another language, different words might be needed to capture these different meanings. Thus, the dashboard might define the following translation keys:

```
Plan=Plan  
Plan_Noun=${Plan}  
Plan_Verb=${Plan}  
Plan_Adjective=${Plan}
```

This approach has two benefits. If your language, like English, reuses the same word in all these situations, then you will only need to translate the "Plan" key, and the other three keys will follow along automatically. If, however, your language requires different words for one or more of these situations, you can edit the other keys, deleting the cross-reference entirely and entering your translation instead.

After you enter translations, click on the Save icon. Your new translations will take effect the next time you restart the process dashboard.

The localizations you enter will be saved automatically into a file with a name like "pspdash_*.jar". You can locate this file by visiting <http://localhost:2468/control/showenv.class>. This file is formatted as a [Process Dashboard Add-on](#), and can be easily shared with other dashboard users. Give them a copy of the file and instruct them to save it into the directory where they installed the Process Dashboard, and they can benefit from your translations as well.

If you choose to use the Localization Tool, please consider sharing your translations with the Process Dashboard development team. This will allow users around the world to benefit from your translation efforts. The Localization Tool can share your translations automatically if you give your consent. (Note that automatic sharing will not work if you have to use an HTTP proxy server to connect to the internet. No error message will be displayed in this case. If you would like to share translations, but you are behind an HTTP proxy server, please send email to processdash-devel@lists.sourceforge.net for further instructions.)

6.4. Editing Preferences

The Preferences Editor allows you to alter configuration settings that affect the appearance and behavior of the Process Dashboard. You can open the Preferences Editor by selecting "Preferences" from the "C → Tools" menu.

Common preferences are displayed in several categories.

6.4.1. User Interface

Window Title: this field allows you to change the title that appears on the main Process Dashboard toolbar. If you regularly work with more than one Process Dashboard dataset, changing the title of each window can make it easier to tell the datasets apart.

Always On Top: If you are running Java 1.6 or higher, this setting will appear, allowing you to request that the Process Dashboard toolbar remain on top of other windows.

System Tray: If you are running Java 1.6 or higher, the dashboard can display an icon in the [system tray](#). A checkbox here allows you to enable or disable this feature. If this feature is enabled, you can also choose how the Process Dashboard should behave when you minimize the main window. If you put a check in the "Minimize to system tray" checkbox, minimizing the dashboard will remove the entry from your application task bar, leaving only the system tray icon. In this mode, double-clicking on the system tray icon will bring the window back.

Translation Support: If your operating system is configured to use a language other than English, and if you (or others) have [entered translations](#) for that language, then the Process Dashboard may be able to display some messages in your language. Here, you can configure the extent to which messages will be translated:

- **Disabled** - US English will be used for all user interface components, messages, reports, charts, and date/number formats.
- **Low** - English will be used for *most* user interface components, messages, reports, and charts. Dates and numbers will be formatted according to the operating system locale.
- **Medium** - User interface components and messages will be displayed in your language whenever translations are available, as will some charts and reports.
- **High** - The dashboard will also attempt to translate the names of process metrics into your language.

Use large fonts: On some systems, it is possible to increase the size of the fonts used in the application. Checking this box will increase the sizes of many fonts, which can make the user interface easier to read on high-resolution displays. (Unfortunately, limitations in the Java platform prevent some font sizes from changing.)

Click on script menu: When you click on the script button, it will normally open the script menu. If you would prefer for it to open the default script/form for the current task instead, you can check this box.

6.4.2. Add-ons

The Process Dashboard can read process templates, scripts, tools, extensions, and other modules from various locations. By default, it will find any modules that are placed in the directory where you installed the dashboard. If you and your team members want to use a [shared network area](#) for Process Dashboard scripts and templates, you can configure that area on this panel. Just ensure that the check box is enabled, and click the "Add" button to add a new template location.

You may have entries in this list that are related to team projects you have joined in the past. If some of the entries are out of date or describe nonexistent network directories, you can remove them from the list.

If your team is not actively using this feature and your organization's network is slow, you may wish to disable this feature entirely. Just uncheck the box on this panel. However, if you receive an error message the next time the dashboard starts, it may be necessary to reenable this feature.

6.4.3. Backups

The dashboard automatically keeps internal backups of data. On this panel, you can configure how many days worth of backup data should be kept. These internal backups are incremental, so they do not take up much space. As a result, the default value (six weeks) is

fairly helpful.

In addition, you can ask the dashboard to make redundant external backups to one or more locations. Just add entries to the table on this preferences panel.

For more information about internal and external backups, see the [data backup](#) help topic.

6.4.4. Miscellaneous

Recommend read-only: Sometimes a Team Dashboard is shared by many people who routinely open it. Most of the time, these people may just need to view rolled up team reports. However, the act of opening the dashboard causes the data to be locked; then, another individual who really needs to make changes is told that the data is in use by someone else. To help with this problem, you can check this box for heavily shared team dashboards. Then, each time an individual opens this dashboard, they will see a prompt asking them if they need read access or if they just need to view reports. This allows the "report-only" individuals to open the dashboard without locking it, leaving the lock available to others who need to make changes.

Check for updated software: The dashboard can perform a monthly check to see if a new version of the software is available. If you do not want it to look for updates, remove the checkmark from this box.

Slow network: If your organization has an unusually slow network, or if you connect to team resources over a VPN, you can check this box. The dashboard will optimize some internal operations differently in an attempt to improve performance.

Numeric IP: The Process Dashboard displays some resources in your web browser. Most of the time, these pages are on the same computer where the dashboard is running. But in a small number of special cases, the dashboard is running on one computer and it displays a URL that you can use from another computer. The most common scenario for this is the Team Project Tools page in the Team Dashboard; it displays URLs that you can share with other people so they can join the project or view the plan summary. By default, it will use the name of the hosting computer to build these "shareable" URLs. But if your organization's network is not configured to register these names in DNS, other individuals may find that the URLs don't work. In that case, you can put a check mark in this box, and the dashboard will use the IP address of the hosting computer (instead of its name) to build the shareable URLs.

6.4.5. Advanced

The categories listed above provide access to the most commonly used dashboard settings. However, the dashboard has an extensive collection of other, infrequently used settings. If you need to change one of those other settings, you can add entries to the table on this panel.

6.4.6. Advanced Configuration Settings

The [preferences editor](#) displays several categories of commonly-used user preferences. Most people will find those settings adequate for their needs.

However, the dashboard supports many other configurable parameters. The list below describes a small set of those parameters that some people may find useful.

If you want to configure one of the settings below, open the [preferences editor](#) and select the "Advanced" section. Click the Add button to add a parameter, then enter the name setting and the value you wish to configure. Then save the changes. In most cases, you will need to restart the dashboard for your changes to take effect.

- **browser.command**

Command used to bring up HTML browser. On Windows and Mac OS X, the dashboard attempts to launch your default browser, so this command is generally not required. On Unix/Linux, the dashboard will launch firefox by default, so you can use this parameter to select a different browser.

Format: a command line.

Default: On Windows and Mac OS X platforms, uses the system default web browser. On Unix/Linux systems, uses "firefox".

Examples: browser.command = /usr/local/bin/opera

- **datasetMode**

Flag indicating whether this is a team or personal dataset. Personal datasets display the user interface with a thin toolbar and allow the logging of time, size, defects, and other metrics. Team datasets show a larger user interface allowing the creation of team projects, which calculate roll-ups of data from multiple individuals.

Format: *team* or *personal*

Example: datasetMode = team

- **dateFormat**

Display format for date fields.

Format: A list of recognized date formats, separated by the | character. The first format in the list will be used when displaying dates. For valid format syntax, see the documentation for java.text.SimpleDateFormat in the Sun Java Developer Kit.

Default: MM/dd/yyyy

Example: dateFormat = dd/MM/yyyy

- **dateTimeFormat**

Display format for date and time fields.

Format: A list of recognized date formats, separated by the | character. The first format in the list will be used when displaying date-times. For valid format syntax, see the documentation for java.text.SimpleDateFormat in the Sun Java Developer Kit.

Default: MMM dd, yyyy hh:mm:ss aaa

Example: dateFormat = MMM dd, yyyy hh:mm

- **defectDialog.autostart**

Controls whether to start the defect timer when opening a new defect dialog.

If this property is set to "true", clicking the defect button on the main dashboard window will cause the defect dialog box to come up with the timer already running. Any other value for this property means the user must manually press the "Start Timing" button to begin the timer for fixing the defect.

Format: *true* or *false*.

Default: *true*

Example: defectDialog.autostart = false

- **defectDialog.phaseMap**

Describes common defect injection/removal patterns. Each pair is interpreted to mean "Often defects are injected in phase A, and removed in phase B." When you remove a defect during phase B of a process, it will then use phase A as the default injection phase.

Format: It should be a list of phase pairs; the pairs should be separated from each other by "|" characters. Each pair should be of the form "phase A=>phase B".

Default: HLD=>Design|Design=>Test

Example: defectDialog.phaseMap = HLD=>Design|Code=>Test

- **defectDialog.restrictSequence**

If this property is set to true, the defect dialog will check injection and removal phases to ensure that defects are NOT removed before they are injected. (You might want to disable this feature if you make use of iterative processes where the real-life task sequence does not strictly follow the order of nodes in the hierarchy.)

Format: *true* or *false*

Default: *true*

Example: defectDialog.restrictSequence = false

- **ev.dependencies.compareDates**

Which calculated date should be used when displaying/comparing projected dates for dependent tasks?

Format: *plan*, *replan*, or *forecast*

Default: `replan`

Example: `ev.dependencies.compareDates = forecast`

- **`ev.sortCompletedTasks`**

As you complete tasks in an earned value schedule, the dashboard will use that information to improve the quality of the planned completion dates for the remaining tasks. It does this by internally reordering the list of tasks so completed tasks appear at the beginning of the task list, in the order they were actually completed. To disable this behavior, and calculate planned completion dates based on explicit task order alone, set this to false.

Format: `true` or `false`

Default: `true`

Example: `ev.sortCompletedTasks = false`

- **`excel.exportMethod`**

The method to use for exporting data into Excel. If this is set to "mime", data will be exported to Excel using mime types; in practical terms, this means that the data you see in Excel will be a snapshot. If this is set to "iqy", tables will be exported using a Microsoft Internet Web Query (.IQY) file; in practical terms, this means that Excel can dynamically requery the dashboard to get updated data.

Format: `mime` or `iqy`

Default: `iqy` if your browser is Internet Explorer, and `mime` otherwise. (This is because other browsers don't correctly handle .IQY files.)

Example: `excel.exportMethod = iqy`

- **`forms.useHoursMinutes`**

Controls the display of time values of dashboard forms. By default, times are displayed in hours:minutes (for example, "1:30" for one hour and thirty minutes). This is a change from earlier versions of the dashboard, which displayed times as a number of minutes (for example, "90"). This setting makes it possible to revert back to the prior formatting.

Format: `true` or `false`

Default: `true`

Example: `forms.useHoursMinutes = false`

- **`http.allowRemote`**

Controls remote access to the dashboard's built-in webserver.

If true, the dashboard will allow connections from other machines.

If false, the dashboard will reject all HTTP connections unless they come from "localhost".

Note that password settings (such as those set on shared/published earned value schedules) establish exceptions that override this setting. Thus, if you have set this to true, setting a password will deny connection attempts that do not correctly identify the password. Conversely, you can leave this set to false, and setting a password will allow connection attempts that correctly supply the password.

If you *never* want other people to be able to connect to your dashboard (whether they know a password or not), set this parameter to "`never`". This will make it impossible for other computers to see your dashboard.

Format: `true, false, or never`.

Default: `never`

Example: `http.allowRemote = true`

- **`http.hostname`**

Most of the time, you interact with the dashboard web server by viewing a web page that begins with "http://localhost". This works fine when the dashboard and the web browser are both running on the same computer. Sometimes, though, it is desirable to construct a URL for other people to use (for example, when publishing or sharing an earned value schedule). The dashboard will ascertain the name of your computer and use that to construct the URL; this process should work 99% of the time. If, however, the dashboard deduces an incorrect name for your computer, you can use this setting to override the built-in name deduction logic.

Format: `hostname`

Default: *none*

Example: http.hostname = mycompany.com

- **http.port**

The port number where the dashboard should listen for incoming http connections.

Format: Number

Default: 2468

Example: http.port = 3000

- **pauseButton.historySize**

The maximum number of entries which should be retained in the play/pause button's history list.

Format: Number

Default: 10

Example: pauseButton.historySize = 20

- **pauseButton.quiet**

If this property is set to "true", the Play/Pause button will not provide any audible feedback. By default, the Play/Pause button will make a clicking sound whenever you start the timer or change tasks with the timer running.

Format: *true* or *false*

Default: *false*

Example: pauseButton.quiet = true

- **pauseButton.compact**

Controls the appearance of the play/pause button on the dashboard.

If true, a single button with both play and pause icons will be displayed. If false (the default), two separate toggle buttons will be displayed.

Format: *true* or *false*.

Default: *false*

Example: pauseButton.compact = true

- **probeDialog.onlyCompleted**

Controls implicit filtering of data by the PROBE tool.

If true, the PROBE tool only correlates data from completed projects.

If false, the PROBE tool will use all available data.

Format: *true* or *false*.

Default: *true*

Example: probeDialog.onlyCompleted = false

- **pspdiff.suffix.c**

The pspdif LOC counter built into the dashboard has support for several common language syntaxes. You can tell it which language to pick based on the suffix of the filename. This rule lets you control which suffixes will cause the pspdif tool to use the C/Java LOC counting rules.

Format: A space-separated list of filename extensions (e.g. ".c .h .java") which signal that a file contains C/Java source code.

Default: .c .cpp .c++ .h .java

Example: pspdif.suffix.c = .c .h .cxx .hxx

- **pspdif.suffix.cobol**

The pspdif LOC counter built into the dashboard has support for several common language syntaxes. You can tell it which language to pick based on the suffix of the filename. This rule lets you control which suffixes will cause the pspdif tool to use the Cobol LOC counting rules.

Format: A space-separated list of filename extensions (e.g. ".cob") which signal that a file contains Pascal source code.

Default: .cob .cbl

Example: pspdiff.suffix.pascal = .p .pas

- **pspdiff.suffix.pascal**

The pspdiff LOC counter built into the dashboard has support for several common language syntaxes. You can tell it which language to pick based on the suffix of the filename. This rule lets you control which suffixes will cause the pspdiff tool to use the Pascal LOC counting rules.

Format: A space-separated list of filename extensions (e.g. ".p") which signal that a file contains Pascal source code.

Default: .pas

Example: pspdiff.suffix.pascal = .p .pas

- **pspdiff.suffix.sh**

The pspdiff LOC counter built into the dashboard has support for several common language syntaxes. You can tell it which language to pick based on the suffix of the filename. This rule lets you control which suffixes will cause the pspdiff tool to use the perl/sh LOC counting rules.

Format: A space-separated list of filename extensions (e.g. ".sh") which signal that a file contains perl or sh source code.

Default: .pl .sh .bash

Example: pspdiff.suffix.sh = .sh .pl .pm

- **pspdiff.suffix.sql**

The pspdiff LOC counter built into the dashboard has support for several common language syntaxes. You can tell it which language to pick based on the suffix of the filename. This rule lets you control which suffixes will cause the pspdiff tool to use the SQL/Ada LOC counting rules.

Format: A space-separated list of filename extensions (e.g. ".sql") which signal that a file contains perl or sh source code.

Default: .sql .ada .adb .ads

Example: pspdiff.suffix.sql = .sql .ada

6.5. Database Support in the Process Dashboard

From its inception, the Process Dashboard has used a specialized data repository to manage project metrics. Beginning in version 2.0, selected metrics can also be accessed in a relational database via SQL. To enable access to this relational database, choose the "Database" option from the "Tools" menu.

For in-depth documentation on this functionality, including the contents of the database schema, please visit:

<http://www.processdash.com/dbhelp>

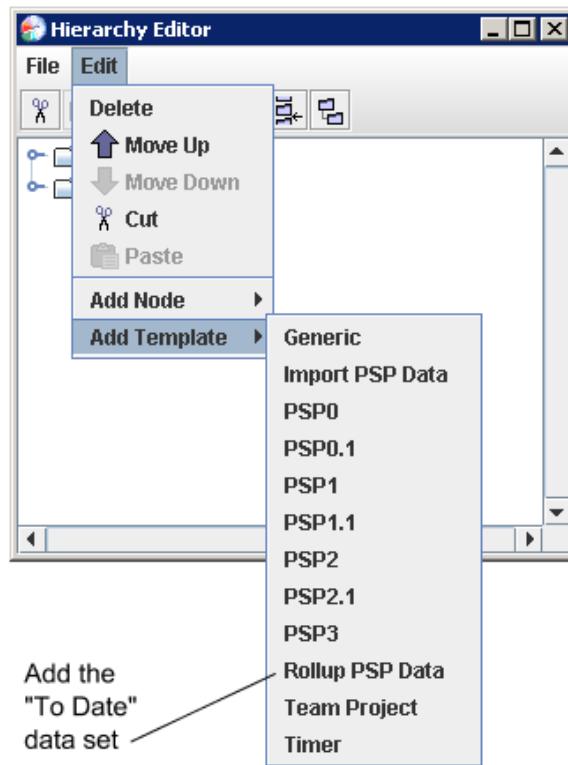
6.6. Using Multiple "To Date" Data Sets

By default, the dashboard maintains one set of To Date data, which rolls up data from all of your completed PSP projects. This may not meet your data needs. For example, if you continually work in several different programming languages, you might want multiple, independent sets of To Date data, one for each programming language. "Data Rollup Templates" address this need.

Data Rollup Templates allow filtering of the data that appears on process forms such as the Project Plan Summary. Once you have created a data rollup in your hierarchy, you can choose to view the forms based on all of your PSP related data or just on the subset that you have defined. The Data Rollup Templates can also be used to rollup data from many small PSP-sized tasks up to the project level. It's just a matter of how you define the data rollup filter expression.

6.6.1. Creating Multiple "To Date" Data Sets

Extra To Date data sets are created in much the same way you would create a new project. You just create the data set in the hierarchy as if you are adding a Project Template.



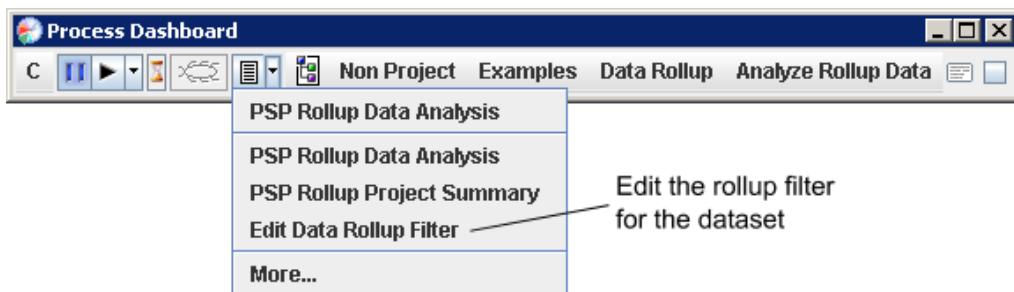
You can rename the data set node just like you can a project. Also, the location of the data set itself in the hierarchy is not important. Just place it wherever it seems to fit best to you. For more information on adding nodes and templates and on using the hierarchy editor in general, see the [hierarchy editor](#) help topic.

Note - there is always a default rollup, defined automatically by the dashboard, which is called "/To Date/PSP/All". This rollup will not appear in your hierarchy, but its data is available for use by projects in your hierarchy.

6.6.2. Customizing "To Date" Data Sets

When you first create a new To Date data set, it will (by default) be rolling up data from all completed PSP projects. Therefore, the first thing you'll probably want to do is customize the filter which is used to generate the list of projects that it will roll up data from. This is done via data entered in an HTML form, much like the standard process forms.

To access the filtering form, use the hierarchy menus to navigate to the To Date data set you have just created. Once there, click the drop-down button on the script button and choose "Edit Data Rollup Filter."



The filter form will appear that will let you edit the expression that is used to filter the project list that is used in the data rollup.

This screenshot shows the 'Edit Data Rollup Filter' dialog box. It has several filter criteria checkboxes:

- Only rollup projects completed after: Jan 01, 1970 05:00:00 AM
- Only rollup projects written with language: [empty input]
- Only rollup projects with keywords matching: [empty input]
- Only rollup projects whose path starts with: [empty input]

Below these is a 'Filter!' button. A note says: 'If you wish, you can edit the filter expression directly.' Below that is a text area containing the expression: '[Completed] > [(Rollup As Of Date)]'. At the bottom, it says 'The resulting list of included projects is:' followed by a list of 10 items: Non Project/Historical Data/Program 1 through Non Project/Historical Data/Program 10. At the very bottom are links for 'Unlock read-only data', 'Export to: HTML Excel', and 'Help...'. A 'Done' button is at the bottom right.

Some common filtering criteria are displayed as choices on the form:

- Only rollup projects completed after a certain date
- Only rollup projects written with a certain language
- Only rollup projects with certain keywords
- Only rollup projects whose hierarchy path starts with a certain string

You can check the appropriate boxes, and edit the details for each criteria. When you're done, click the "Filter!" button. This button will compute the corresponding filter expression, and put it in the box below. This will also cause the "list of included projects" to recalculate, showing you all the projects in your hierarchy that meet the criteria.

If you feel comfortable with the data expression syntax, you can edit the filter expression directly. After you make changes to the expression, you'll have to move the cursor out of the text area (by hitting TAB or by clicking somewhere on the HTML page background) for the included project list to recalculate.

TIP: If the included project list is empty and you don't think it should be, you may have a syntax error in your filter expression. A syntax error in your expression will essentially be treated the same as the expression "false". Note, however, that an empty included project list does not automatically mean your filter expression syntax is bad - it may be a valid expression, and there just might not be any projects that meet the criteria.

6.6.3. Viewing Rolled Up Data

To view rolled up data, just use the hierarchy menus to navigate to a data rollup node, and click the script button. The [Data Analysis](#) center will appear - but it will be displaying only data from the projects that you have included in your rollup! In fact, the main "Data Analysis" center which is launched from the Configure menu is nothing more than a view of the rolled up data for the automatic data rollup called "/To Date/PSP/All".

6.6.4. Using Alternate "To Date" Data Sets

Once you have created an additional data rollup set, you will notice that all of your PSP Project Plan Summaries will contain an extra drop-down list. This drop down list is located in the "Project Settings" section of the plan summary form. The drop-down list is labeled "Base 'To Date' PSP data on:", and the drop down list will contain all the PSP data rollup objects you have defined, plus the extra "default" rollup, which is called "/To Date/PSP/All". Choosing a different rollup from this list will cause this project to use that rollup as the basis for data in the "Plan" and "To Date" columns.



The screenshot shows a form with several input fields and a dropdown menu. The fields include 'Project Owner' (set to 'Sample'), 'Start Date', 'Completion Date', 'Completed' (unchecked checkbox), 'Keywords', 'Language' (set to 'C'), and a dropdown menu labeled 'Base "To Date" PSP data on:' which is currently set to '/To Date/PSP/All'.

The pulldown lets you choose the basis for the report

TIP: remember that data in the plan column freezes when planning is marked complete, and data in the "To Date" columns freezes when the project is marked complete. Therefore, if Planning and/or the entire project is marked complete when you alter the drop-down, the data in the "Plan" and/or "To Date" columns will not recalculate.

6.7. Making a change to a script

Perhaps a pre-existing process suits you well, but you'd like to make some small tweak to one of the HTML scripts. For example, you might like to add company specific instructions to some of the steps (to run a code analyzer perhaps).

When you're viewing the page in question, you notice it has a URL like:

`http://localhost:2468/Project+Path//procID/script.htm`

In this example, "procID" is the ID of the process containing the script, and "script.htm" is the name of an HTML script within that process.

Take the following steps:

1. Choose "Help → About" from the "C" menu, and click the "Configuration" tab. Just above the list of add-ons, a paragraph will tell you where the Process Dashboard is installed. Find that directory on your computer. It should contain a small number of files, including one called "pspdash.jar".
2. Underneath this directory, create a subdirectory called "Templates".
3. Create a subdirectory within "Templates" with the name that appears in the "procID" space in the example above.
4. Save the current version of the HTML file to the subdirectory you just created. This can be done by choosing "File → Save As..." from within your web browser.
 - Give the file the same name that appears in the URL - in this instance, we would save the file as "script.htm". (Note: if the URL contains a "?" or "#", just use the filename that appears before the "?" or "#".)
 - If your web browser has a feature to save "html only", use it.
5. Edit it to your heart's content using the HTML editor of your choice.

Now, every time you visit that page, for any project in your hierarchy that uses this process, the dashboard will show you the page containing your custom changes.

6.8. How the Dashboard finds process files

When it starts, the dashboard searches for process files and add-ins and dynamically loads them. Also, every time you request a web page from <http://localhost:2468/>, the dashboard is using this same search mechanism.

The dashboard searches for files in the following locations, in the following order:

1. Look first in the directories specified by the [add-on search path](#).
2. Next, look in any JAR files contained in that directory.
3. Next, look in any JAR files contained in the parent of that directory.
4. If there is a Templates directory underneath the directory where the Process Dashboard is installed, look there: first for the file, then for JARs containing the file.
5. Next, look within the JAR and ZIP files that are present in the Process Dashboard installation directory.
6. Finally, look inside the Process Dashboard application file. (This file is called "pspdash.jar", and it appears in the Process Dashboard installation directory.)

When looking for a particular file, the dashboard will search in the above locations and stop as soon as it finds a matching file. This means that if you want to customize anything that appears in a process templates directory, you can. Just create your custom version and put it in a location earlier on the search list.

Note: although dashboard add-ons are distributed as ZIP or JAR files, you should refrain from unzipping them. For example, if you were to unzip an add-on in place, it would create a new Templates directory underneath the Process Dashboard installation directory. When you upgrade the dashboard in the future and a new version of the add-on is installed, your unzipped files will be overriding the newly released materials! This will almost certainly cause serious problems, as new and old versions of files will be interacting with each other. It is fine to extract single HTML files from an add-on if you intend to customize that one file. But do not extract files that you don't intend to change, or whose purpose you don't understand.

6.8.1. Sharing changes with team members

If you're working with a team, and everyone on your team would like access to changes you have made to a process script, you can do the following:

1. Create a directory somewhere out on a shared network drive, called "Templates" (for example: "T:\Common\Our Team Folder\Templates" on a Windows network, or "/usr/shared/OurTeam/Templates" on a Unix network)
2. Have each person on your team add this directory to their [add-on search path](#).
3. Put custom versions of files underneath this team templates directory. For example, you could put:
 - HTML files overriding files that are built into the dashboard
 - LanguageFilters defining custom LOC counters
 - [template.xml](#) files defining team processes
 - Data files and HTML scripts/forms supporting those team [processes](#)

and when you make changes to these items, everyone on the team will see them automatically (except as noted below).

6.8.2. Some files are only read at start up

When you change HTML files, everyone will see your changes immediately. The very next time they view the HTML page in question, they will see your changes. Other files, however, the dashboard will only read once. For example, template.xml files, datafiles, and LOC counters will only be read once, when the dashboard starts up. If you make changes to these files, people will have to shut down and restart their dashboard to see your changes.

6.9. Defining a Custom LOC Counter

In the PSP, lines of code (LOC) are often used to measure the size of a software component. To produce a size measurement that is useful for planning and analysis purposes, PSP recommends that you separately account for base, added, modified, and deleted LOC. In addition, it recommends that you exclude comment lines, blank lines, and automatically-generated code from the count.

Although many free and commercial LOC counters are available, they typically only measure total LOC, and they often include comment lines in the counts they produce. For this reason, the dashboard provides a simple [LOC counting tool](#) that developers can use to measure base, added, modified, and deleted LOC in a PSP-recommended way.

To properly exclude comments from the count, this LOC counting tool must have a rudimentary understanding of the programming language syntax. Accordingly, the LOC counting tool is prepopulated with several language filters that recognize the syntax of many popular programming languages. The LOC counter can automatically handle files of the following types:

- C-style languages like C, C++, C#, and Java (*where single-line comments begin with //, and block comments are bounded by /* */*)
- Shell-style languages (*where single-line comments begin with the # character*)
- Pascal and other similar languages (*where single-line comments begin with //, and block comments are bounded by { }*)
- SQL and other similar languages (*where single-line comments begin with --, and block comments are bounded by /* */*)
- Cobol (*which includes line numbers and uses a character in a specific column to flag a line as a comment*)
- A plain-text filter, which recognizes no comment syntax and counts all nonblank lines. (*This filter is used by default if none of the other filters appear appropriate.*)

Of course, many other source code languages exist, and you may find yourself needing to count LOC for a language not handled above. Fortunately, the dashboard allows language filters to be defined and loaded dynamically, so you can create your own custom language filters for use with the dashboard's LOC counter.

To create a custom language filter, follow these steps:

1. [Decide what lines of code should be counted](#)
2. [Create an XML definition for your language filter](#)
3. [Save your XML language filter definition](#)
4. [Restart the dashboard](#)

6.9.1. Step 1: Decide what lines of code should be counted

In the PSP, we seek to identify size measures that correlate well to development effort. While LOC is not perfect, we nevertheless want to craft our LOC counting standard so it supports this goal.

The PSP makes several recommendations to help us get started. For example, it recommends that we do not count comments, blank lines, or autogenerated code.

Following this advice, the dashboard's LOC counter automatically ignores lines that contain only whitespace. In Step 2(c) below, you will tell the LOC counter about the comment syntax for your programming language, so it can exclude comments from the count as well.

Autogenerated code can be trickier. But occasionally you may find that a particular type of autogenerated code follows a simple pattern, and can be excluded. For example, it is common for Java IDEs to automatically manage the `import` statements at the top of the file; if you regularly use that IDE feature, you might choose to exclude `import` statements from the LOC count.

In addition, be wary that autogenerated code may be present in your source code but invisible to you! Some IDEs (for example, Visual Studio) have been known to attach large headers or footers to a source file, but not display those headers/footers when the file is opened in the IDE. Use a plain text editor like Notepad or Emacs to open several typical source files and see if the contents match your expectations.

Finally, the PSP makes a blanket recommendation: if you aren't certain whether a particular line should be counted, try running your metrics both ways, then see which approach gives you a better correlation between size and effort.

As an example, this technique was used by the dashboard development team to determine that curly braces appearing on their own line usually **should not** be included in LOC counts of C-style languages. When you think about this, it becomes clear why. Consider the following three code fragments:

if (test) then-clause; else else-clause;	if (test) { then-clause; } else { else-clause; }	if (test) { then-clause; } else { else-clause; }
---	--	---

For the purposes of this discussion, we aren't concerned about which of these fragments is "superior." We just want to think about how we should count the LOC they contain.

As written, the three fragments are semantically equivalent, differing only in the appearance and placement of the curly braces. The bulk of the programming effort will be invested in writing the test expression and the two conditional clauses; thus, we would expect these three fragments of code to require virtually identical effort to produce from scratch. To produce the best correlation between size and effort, we would ideally like them to have the same size measurement.

If we were to count every nonblank line as a line of code, then the third code fragment would allegedly be twice the size as the first (allegedly requiring twice as much effort to produce). That is counterintuitive and can be rejected straightaway. However, if we simply ignore any curly brace that appears on a line by itself, we will receive identical LOC counts from all three fragments. In this way, ignoring lone curly braces allows us to correct for varying whitespace styles across a large code base, improving the quality of our size measurements.

Once you have decided which lines of code you wish to count, you're ready for the next step.

6.9.2. Step 2: Create an XML definition for your language filter

The dashboard supports a simple declarative XML syntax for the definition of simple language filters. **The discussion below assumes that you are familiar with XML.** If this is not the case, please go read a nice tutorial on XML before proceeding!!

Here is an example of an XML description file for the C-style language filter:

```
<?xml version='1.0'?>

<dashboard-process-template>

<locFilter id="C" fileSuffixes=".c .cpp .c++ .h .java .cs">
    <commentSyntax beginsWith="//> />
    <commentSyntax beginsWith="/*> endsWith="*/> />

    <stringSyntax delimiter="&quot;" escapeChar="\\> />
    <stringSyntax delimiter="&apos;" escapeChar="\\> />
    <stringSyntax beginsWith="@&quot;" endsWith="&quot;" mayInclude="&quot;&quot;" />

    <possibleFirstLine>#include</possibleFirstLine>
    <possibleFirstLine>#define</possibleFirstLine>
    <possibleFirstLine>#if</possibleFirstLine>
    <possibleFirstLine>#pragma</possibleFirstLine>

    <ignoreLine equalTo="{" />
    <ignoreLine equalTo="}" />
</locFilter>

</dashboard-process-template>
```

The file begins and ends with the header and footer (which appear in blue above). Between the header and footer, any number of LOC filters can be defined through the use of the `<locFilter>` element (please note that capitalization is significant). Note that if you are also creating [custom process definitions](#), you can combine custom process `<template>` definitions and custom `<locFilter>` definitions in

the same XML description file, mixing the tags in any order.

Step 2(a): Assign a Unique ID

Each LOC filter must have a unique identifier, which is specified in the `id` attribute of the `locFilter` tag. This identifier will be displayed to end users in the LOC report, so it is best to choose a human-readable string. If you reuse the same ID as a LOC filter built-in to the dashboard (i.e., `C`, `Sh`, `Pascal`, `SQL`, `Cobol`, or `Default`), your definition will replace the built-in filter.

Step 2(b): List Typical Filename Suffixes

When you run a LOC comparison report, the dashboard looks at each file being compared and attempts to determine which language filter is most appropriate - a process called "LOC Filter Selection." This process involves several steps, but the first step is to look at the filename. You can use the `fileSuffixes` attribute to list the filename suffixes (separated by spaces) that are commonly used for source code files in the programming language you are describing.

Step 2(c): Define Comment Syntax

For the purposes of LOC counting, comment syntax is the most important difference between the various language filters. You can declare any number of comment styles for a language filter by creating embedded `<commentSyntax>` elements.

Each `<commentSyntax>` element must have a `beginsWith` attribute, indicating the string of characters that begins a comment. An `endsWith` attribute can also be specified, indicating the sequence of characters that ends the comment. If no `endsWith` attribute is given, the "end-of-line" character will be assumed; thus, single-line comment styles can be specified simply by a `beginsWith` attribute.

If your programming language has blocks of auto-generated code that begin with a distinctive sequence of characters, you may be able to use a `<commentSyntax>` element to describe them. Any matching blocks will be excluded from LOC counts along with the other comments in the file.

Technically, you could choose not to list any `<commentSyntax>` elements. The resulting LOC filter would describe a language without comments, or alternatively, a language where comments are included in LOC counts. The "Default" filter is a preinstalled example of this.

Step 2(d): Define String Literal Syntax

The LOC counter is primarily focused on identifying comments so those can be excluded from the count. However, there are occasions when a comment indicator can appear inside a string literal. Here is an example taken from a C program:

```
printf("Visit http://example.com/ for more information");
```

In this line of code, the `//` that appears inside the string literal could be mistaken as the start of a single-line comment, which would prevent any subsequent changes on the line from being considered toward the "modified" count. To avoid this type of mistake, you can tell the LOC counter about the syntax for string literals in your programming language.

You can declare any number of string literal styles for a language filter by creating embedded `<stringSyntax>` elements. Each `<stringSyntax>` element must have one of the following:

- a `delimiter` attribute, indicating the string of characters that begin and end a string literal
- `beginsWith` and `endsWith` attributes, indicating the strings of characters that begin and end the string literal.

An `escapeChar` attribute can optionally be specified, indicating a character (such as `\`) that is used to escape the following character inside the string.

In addition, a `mayInclude` attribute can optionally be specified to indicate a sequence of characters that are treated specially within the string literal and which should not signal the end of the string. A common example would be doubled-up delimiter characters, which occur in languages like Visual Basic. If your language allows the newline character to appear within a string literal, you can specify "`\n`" for this attribute; otherwise the LOC counter will treat the end-of-line as the end of an improperly formed string constant.

These attributes provide support for the vast majority of string literal constructs in the most popular languages. Some languages (for example, Perl) provide very flexible string quoting mechanisms that would become extremely complex to recognize without implementing a parser. If you use a language whose string rules cannot be expressed with this XML syntax, keep in mind that the LOC counter will not be able to detect comment indicators that appear within the more complex string constructs. Fortunately, this is fairly rare; however, you should be wary of comment indicators that appear inside string literals, and adjust your LOC counts if you know these are present.

Step 2(e): List Possible First Lines (Optional)

During the "LOC Filter Selection" process, if a file is encountered whose filename doesn't match any recognized suffix, the dashboard will look inside the file in an attempt to guess which language it contains.

Since it is very common for source code files to begin with a descriptive comment, the dashboard will first check to see if the first non-whitespace characters in the file appear to match any known comment syntax (specified via a `<commentSyntax>` element as described above).

If a particular file under comparison does not have a recognizable file suffix and does not begin with a recognizable comment, the "LOC Filter Selection" process might still be stumped. In that case, you can optionally list other strings that could legally appear at the beginning of a source code file. This is done by including embedded `<possibleFirstLine>` elements. If the first line of the file begins with one of these strings, this LOC filter may be considered a match.

`<possibleFirstLine>` elements are completely optional; most language filters will not need to use them. In particular, remember that these are only used as a last resort; normally files will be categorized based on their filename suffix.

Step 2(f): List Ignored Lines (Optional)

In addition to comments, there may be other lines of code that you wish to exclude from LOC counts. These can be specified by creating embedded `<ignoreLine>` elements. Each `<ignoreLine>` element must have exactly one of the following attributes:

- `equalTo` – lines of code should be ignored if they are equal to the given value. (Whitespace will be trimmed from the line of code before it is compared to the given value.)
- `beginningWith` – lines of code should be ignored if they begin with the given value. (Whitespace will be trimmed from the line of code before it is compared to the given value.)
- `endingWith` – lines of code should be ignored if they end with the given value. (Whitespace will be trimmed from the line of code before it is compared to the given value.)
- `containing` – lines of code should be ignored if they contain the given value.

The lines of code in the file will be matched against these patterns; if any pattern matches, the line will not be included in LOC counts. Keep in mind that lines are tested one at a time; therefore, multiline patterns are not supported.

6.9.3. Step 3: Save your XML language filter definition

Once your XML language filter definition is complete, it is time to save it somewhere [where the dashboard can find it](#). Follow these steps:

1. Choose "Help → About" from the "C" menu, and click the "Configuration" tab. Just above the list of add-ons, a paragraph will tell you where the Process Dashboard is installed. Find that directory on your computer. It should contain a small number of files, including one called "pspdash.jar".
2. Underneath this directory, create a subdirectory called "Templates".
3. Save your XML language filter definition in that Templates directory. **Important:** you must give your XML language filter definition a filename ending with `"-template.xml"` to signal to the dashboard that the file contains dashboard metadata. If you do not choose a filename ending with `"-template.xml"`, your language filter definition will be ignored.

This is the simplest place to put your XML language filter definition. Of course, if you want to share your LOC counting filter with team members, you may want to put it somewhere else. See the section on [how the dashboard finds process files](#) for more information. Keep in mind that the dashboard will search for `"-template.xml"` files in the "Templates" directories only - subdirectories will not be searched.

6.9.4. Step 4: Restart the dashboard

Once your XML language filter definition is in place, it is necessary to shut down the dashboard and restart it for the new filter to be found. Once found, it will automatically be included in the "LOC Filter Selection" process when a LOC report is run. If you make changes to your LOC filter definition, a dashboard restart will also be required for your changes to take effect.

6.10. Defining your own processes for use in the Process Dashboard

The dashboard has been designed from the ground up to support custom processes. The PSP scripts and forms that you see are not hard-coded into the tool. Instead, they are dynamically loaded from simple HTML and text files. Thus, it is possible to create your own custom process scripts, forms, and data that will be dynamically integrated into the dashboard.

The dashboard process definition framework provides a great deal of flexibility and power for process automation. The more advanced features will be documented in a *Process Authoring Manual*, which may be available for download from <http://www.processdash.com/>. However, creating a simple, custom personal process is not difficult.

Please note: the instructions below will allow you to build a customized process for **personal** use. The resulting process **will not** provide support for team planning or team rollups. If you plan to work as part of a team, and you want to roll data up to view team-level aggregate metrics, you should not follow the steps below. Instead, you should look into the Process Dashboard's [team functionality](#). Most teams can use satisfy their process customization needs by building a [team workflow](#). A small number of teams may require the more sophisticated process customization that comes from defining a [custom metrics collection framework](#).

To create a custom personal process definition, just follow these steps:

1. [Identify your process structure](#)
2. [Create an XML definition for your process](#)
3. [Save your XML process descriptor](#)
4. [Restart the dashboard](#)

6.10.1. Step 1: Identify your process structure

What steps do you want to include in your process? The standard PSP0 process, of course, includes the phases Planning, Design, Code, Compile, Test, and Postmortem. Think through your process and decide upon the list of phases that your process will include.

Think this through carefully. Remember how you had to restart your "To Date" data when you graduated from PSP1.1 to PSP2? The same truth applies here. Once you have defined a custom process and used it in the dashboard, even slight changes to the process definition will invalidate all your historical data. In fact, once you create a process definition and use it to perform a project, you can no longer even change the names of the phases without losing historical data.

If your process definition is still immature and evolving, it is probably better to use the [Generic process](#) for a while. As you begin creating your own custom process definitions, the flexibility of the generic process will allow you to radically change your process from one task to the next in response to process improvement proposals. Once you decide upon and finalize the [list of phases](#) in your process, you are ready to begin using a custom process.

Some guidelines to consider as you identify your process phases:

Don't subdivide things too far. Take a look at the PSP processes, for example. They will define a single phase like Planning, even though it contains several different steps (obtain requirements, make size estimate, make resource estimate, create schedule, etc.). The PSP process definition does not break each of these steps into its own phase! Herein lies the difference between a **phase** and an **activity**. By carefully defining the high level phases in the process, Watts Humphrey was free to incorporate process improvements simply by altering the set of activities in each phase. Remember also that when you enact your process, you will have to collect time and defect data with phase-level granularity. If you break your process up into a zillion tasks that each take only a minute, you will go crazy just trying to log time accurately. Therefore, if your custom process contains more than a dozen phases, it's most likely subdivided too far.

Here's another way to think about this: the dashboard is a useful tool because it seamlessly integrates your **process script** and your **metrics collection framework**. By tying the two things together, you have all the tools you need at your fingertips to enact the process. But as a result, this means that when you create a custom dashboard process, you must understand and think about your custom process from both perspectives. Many people are able to easily think up a script for their process, but you must also take time to think about your metrics collection framework. The phases you choose in your custom process define the "buckets" in your metrics collection framework. If you choose these phases thoughtfully, it will allow room for your process to grow and improve. On the other hand, if you choose phases that correspond one-to-one with activities in your process, you may find that your custom process quickly becomes obsolete.

Choose abbreviations for long phase names. When you write your process script, you may have phases with verbose names like

"High Level Design Review". Realize that if you use this long phase name, the main dashboard window is going to stretch halfway across your screen whenever you navigate to that phase. An abbreviation like "HLD Review" may be more appropriate. You can still refer to the phase by its unabbreviated name when you create your HTML process scripts and forms.

Include Planning and Postmortem phases. In his book, Watts Humphrey encourages you to always include these phases in any process you define. The planning phase serves an obvious purpose of estimating size, cost, schedule, risk, etc. The postmortem phase allows you to ensure accurate collection of data and capture process improvement proposals.

6.10.2. Step 2: Create an XML definition for your process

The dashboard understands a process structure that is described in XML format. **The discussion below assumes that you are familiar with XML.** If this is not the case, please go read a nice tutorial on XML before proceeding!!

Here is an example of an XML description file for a PSP0 process:

```
<?xml version='1.0'?>

<dashboard-process-template>
  <template name="PSP0" defectLog="true">
    <phase name="Planning" type="plan"/>
    <phase name="Design" type="dld"/>
    <phase name="Code" type="code"/>
    <phase name="Compile" type="comp"/>
    <phase name="Test" type="ut"/>
    <phase name="Postmortem" type="pm"/>
  </template>
</dashboard-process-template>
```

The file begins and ends with the header and footer (which appear in blue above). Between the header and footer, any number of process templates can be defined. (The file above only defines one process template.)

A process definition begins with the `<template>` tag and ends with `</template>`. The template tag must have a "name" attribute, and all of the processes you define must have **unique** names. Also, template names cannot contain the "/" character. If you will be collecting defects with your process, you should include the "defectLog" attribute as shown above (and remember that XML is case-sensitive, so "defectLog" will not work).

Between the `<template>` tags, list the phases in order. Each phase is described with a `<phase>` tag. The phase tag must have a "name" attribute, and all of the phase names must be **unique**. As was true of template names, phase names cannot contain the "/" character. The phase type can optionally be specified with the "type" attribute. The following is a list of valid types that can be specified. Their full explanation is not detailed here, but will be in the *Process Authoring Manual* coming soon on our website.

- **Appraisal phase types:** appraisal reqinsp hldr hldrinsp dldr dldinsp cr codeinsp
- **Failure phase types:** failure comp ut it st at pl
- **Overhead phase types:** mgmt strat plan pm
- **Development phase types:** req stp itp td hld dld code doc

Phase types are not required. Even if you specify the phase type for one phase, you can omit it for others. If you do specify phase types, the dashboard will be able to intelligently calculate metrics like Yield, A/FR, and Defect removal efficiency.

6.10.3. Step 3: Save your XML process descriptor

Once your XML process descriptor is complete, it is time to save it somewhere [where the dashboard can find it](#). Follow these steps:

1. Choose "Help → About" from the "C" menu, and click the "Configuration" tab. Just above the list of add-ons, a paragraph will tell you where the Process Dashboard is installed. Find that directory on your computer. It should contain a small number of files, including one called "pspdash.jar".
2. Underneath this directory, create a subdirectory called "Templates".
3. Save your XML process descriptor in that Templates directory. **Important:** you must give your XML process descriptor a filename ending with `"-template.xml"` to signal to the dashboard that this is an XML process template descriptor. If you do not

choose a filename ending with "`-template.xml`", your process descriptor will be ignored.

This is the simplest place to put your XML process descriptor. Of course, if you want to share your process with team members, you may want to put it somewhere else. See the section on [how the dashboard finds process files](#) for more information. Keep in mind that the dashboard will search for "`-template.xml`" files in the "Templates" directories only - subdirectories will not be searched.

6.10.4. Step 4: Restart the dashboard

Once your XML process descriptor is in place, it is necessary to shut down the dashboard and restart it for the new process to be found. Once found, you will automatically have:

- A template on the hierarchy editor's "Add Template" menu for your process
- A template on the hierarchy editor's "Add Template" menu for **rolling up** data from your process
- A custom project plan summary form for your process

6.10.5. Advanced/Optional: Defining HTML scripts

Defining HTML scripts for your process and its phases is easy to do. Here is an example of an XML description file for a PSP0 process, with HTML script definitions added. The text in blue is identical to the example shown in step 2 above; the black text illustrates the additions:

```
<?xml version='1.0'?>

<dashboard-process-template>
  <template name="PSP0" defectLog="true"
    htmlID="top">

    <html ID="sum"
      title="Project Plan Summary"
      href="dash/summary.shtml"/>
    <html ID="top"
      title="PSP0 Process Script"
      href="psp0/script.html"/>
    <html ID="plan"
      title="PSP0 Planning Script"
      href="psp0/planning.html"/>
    <html ID="dev"
      title="PSP0 Development Script"
      href="psp0/develop.html"/>
    <html ID="pm"
      title="PSP0 Postmortem Script"
      href="psp0/postmort.html"/>

    <phase name="Planning" type="plan"
      htmlID="plan"/>
    <phase name="Design" type="dld"
      htmlID="dev"/>
    <phase name="Code" type="code"
      htmlID="dev"/>
    <phase name="Compile" type="comp"
      htmlID="dev"/>
    <phase name="Test" type="ut"
      htmlID="dev"/>
    <phase name="Postmortem" type="pm"
      htmlID="pm"/>
  </template>
</dashboard-process-template>
```

So right after your `<template>` tag, you list all the HTML pages that you want to appear on the script button's drop down menu, in order. (Note that you don't have to list every HTML page used by your process - just the ones that you want to appear on the script button menu.) For each HTML page, you state:

- the title (which should usually match the `<title>` of the HTML page in question); this title will be displayed in the script button's

drop-down menu

- the href to the page, relative to the "Templates" directory
- a unique ID for the HTML page (these only need to be unique within a given process <template>)

Then you (optionally) associate these HTML pages with the phases by adding an `htmlID` attribute which names the ID of the HTML page associated with that phase. You can have HTML pages that aren't associated with any phase, and vice versa. You can even associate an HTML page with the `<template>` tag, as shown above; then any phases which don't specify an `htmlID` of their own will inherit that `htmlID`.

Of course, before the dashboard will be able to open any of these HTML pages, you must actually create them and place them appropriately underneath the "Templates" directory. You can use your favorite HTML editor to create these pages; if you have a program like Microsoft Word, it is capable of saving files in HTML format. Since the hrefs are interpreted relative to the "Templates" directory, for this example we would need to create a directory called "psp0" underneath the "Templates" directory, and put files in that directory called "script.html", "planning.html", "develop.html", and "postmort.html".

The file "dash/summary.shtm" is special. It names an HTML file that is shipped with the dashboard. When you don't name any HTML files for your process, the dashboard assigns it "dash/summary.shtm" by default. (This is why you could create an XML template file, specify no HTML files, and still get a project plan summary form automatically.) However, as soon as you include `<html>` tags in your template definition, the dashboard will no longer automatically assign it "dash/summary.shtm", so you must manually include this entry to ensure that you can view a plan summary form for your process.

6.10.6. Additional Advanced Topics

That should be enough to get you started! For more advanced topics like

- Defining custom forms
- Generating charts and tables of your data
- Defining custom calculations (or altering the existing calculations) for the data in your process
- Integrating external documents into process scripts

Check back on our website and download the *Process Authoring Manual* when it becomes available. In the meantime, if you have a pressing need to understand one of these advanced topics, send email to the development team at `processdash-devel@lists.sourceforge.net`.

6.11. Size Metrics API

Name	sizeMetrics
URI	/api/sizeMetrics
URL Example	http://localhost:2468/api/sizeMetrics
Description	This API makes it possible to store new actual size measurements into the Size Inventory Form for a team project.
Context	Personal dashboard only, within the context of a Team Project
HTTP Methods	<ul style="list-style-type: none"> • POST - adds a new row of actual size data to the Size Inventory Form

6.11.1. HTTP POST

Adds a new row of actual size data to the Size Inventory Form.

This API can be called multiple times in succession to add multiple new rows to the Size Inventory Form. (Note: when multiple rows like these are present for a particular WBS item, they will be summed when the data is rolled up to the team level.)

At the moment, this API only supports size metrics recorded by an individual onto the Size Inventory Form of a team project. Other size metric types, such as the Size Estimating Template of a PSP task, are not currently supported.

By default, invocations of this POST method will be rejected unless they originate from the same computer where the Process Dashboard is running. This policy can be disabled by [setting an advanced preference](#) with the name `api.sizeMetrics.allowRemote` and the value `true`.

6.11.1.1. Request Parameters

Field	Type	Description
hierarchyPath (optional)	String	<p>The full dashboard hierarchy path to the project and component/task where the size metric should be stored. (Example: /Project/My Team Project/Component A/Code).</p> <p>The path must name a component or task within a team project. The root node of the team project is not allowed. If the personal project plan contains "PSP Task" objects, paths to/within those PSP task objects are also not allowed. Any other WBS item path is an acceptable target. (If in doubt, you can view a complete list of valid paths for a given team project by opening the Size Inventory Form and scrolling to the bottom of the page.)</p> <p>This parameter is optional. If omitted, the default is to use the path of the WBS task which is currently selected in the Process Dashboard toolbar. (But note: that currently selected task must also meet the criteria described above for the size measurement to be stored successfully.)</p>
description (optional)	String	A short, human readable description of this size measurement, which will be written into the Description field of the new Size Inventory Form row. (Example: "Perforce Changelist #1234")
sizeUnits (optional)	String	<p>The size units for this size measurement. The value must be a valid size metric as defined in the metrics collection framework used by this Team Project. (For a list of valid values, look at the drop-down list on the "Size Units" field of the Size Inventory Form, and discard the "Inspected **" variants.)</p> <p>This value is optional; if omitted, the API will assume "New & Changed LOC." (The abbreviation</p>

		"LOC" can also be used as an alias for this size metric.)
actSize (required)	Number	The numeric size value which should be stored into the "Actual" column of the Size Inventory Form.
notify (optional)	Boolean	If true, the Process Dashboard window will display a notification to the user, letting them know that size measurements were added to their Size Inventory Form. They will be able to click on the notification to see an abbreviated version of the Size Inventory Form with the new rows highlighted. If this API is called several times in rapid succession to add multiple size measures to a particular team project, the user will only see one notification summarizing all of the rows that were added.

6.11.1.2. Errors

Error Code	Description
unsupported-target-path	The <code>hierarchyPath</code> parameter (or the current task path, if that parameter is missing) does not point to a legal node in a team project. See the requirements described in the "Request Parameters" table above.
parameter-missing	The required <code>actSize</code> parameter was not supplied
parameter-invalid	One of the parameters does not have a valid value: <ul style="list-style-type: none"> The <code>actSize</code> parameter is not a valid number The <code>sizeUnits</code> parameter does not name a valid size metric in the metrics collection framework for this team project

6.11.1.3. Results

application/json - successful completion

```
{
  "stat" : "ok"
}
```

application/json - error example

```
{
  "stat" : "fail"
  "err" : {
    "code" : "parameter-invalid",
    "msg" : "The 'actSize' parameter value 'one' is not a valid number",
    "param" : "actSize"
  }
}
```