

PRC

Project Tracking Guide

Version 6

November, 2005



Notice

The information contained in this document, as well as the software it describes, is subject to change without notice. SJ+ Systems Associates, Inc. makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. SJ+ Systems Associates, Inc. shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

The software described in this document is furnished under a license agreement or nondisclosure agreement. The software may be used or copied only in accordance with the terms of the agreement.

This document contains proprietary information, which is protected by copyright. All rights are reserved. No part of this document may be copied, reproduced, transmitted, or translated into another language without the prior written consent of SJ + Systems Associates, Inc. The information contained in this document is subject to change without prior notice.

UniVerse, Unidata, SB+, and SBClient are trademarks of IBM Corporation in the United States, other countries, or both. Windows is a registered trademark of Microsoft, Inc. All other brand and product names are trademarks or registered trademarks of their respective owners.

Corporate Headquarters:

Phone: (954) 796-9868
FAX: (954) 796-9868 – call first
<http://www.sjplus.com>

Copyright © 1993–2005 by SJ+ System Associates, Inc.

Version: November, 2005

Table of Contents

Introduction	vii
Who Should Read This Book?	vii
Focus and Audience	vii
General Knowledge Required	vii
What is PRC?	viii
Terminology	viii
PRC & Other Source Control Systems	ix
Basing Work on Projects	ix
Checking Out Components	x
Methodologies For Archiving	x
Where This Book Fits In	xi
More PRC Documentation	xiii
Telnet Programs, Character Mode & GUI Mode	xiii
Menu Navigation	xiv
Standard Function Keys	xv
Screen & Field Navigation	xvi
Screen	xvi
Field	xvi
Text Conventions	xvii
Chapter 1: Projects, Versions & the PRC Menu	19
The Project States Defined in PRC	19
A Project's Development Life Cycle	19
The Programmer's Critical Path	20
The PRC Menu for Project Tracking	21
Super-Projects (Versions) & Sub-Projects	22
Versions	22
Sub-Projects	23
Sub-Projects & Syster Sub-Projects	25
What Are They?	25
Why Should I Use Them?	26
How Do I Use Them?	27
Chapter 2: The Programmer's Critical Path	29
The Critical Path	30
Step 1: Assign	30
Step 2: Work	31
Step 3: Bump	31
Step 4: Roll Out	32
Additional Menu Options Outside the Critical Path	32
Chapter 3: Switching Between & Assigning Projects	35
Switching to Another Project	36
Creating a New Project	39

Table of Contents

Chapter 4: Working on Projects	43
Editing Items vs. Your Security Settings	44
Reviewing & Managing Items in Your Project	45
Overview	45
Reviewing Items Tied to Your Project	45
Manually Adding Items to Your Project	45
Manually Removing Items From Your Project	46
How to Use Review Source Items (/SRC)	46
Adding Items to Projects (Checking Out)	51
Removing Items from Projects (Flicking)	52
Drilling Down	53
Exploring This Item	54
Bumping the Project	56
Scanning for Conflicts	57
Chapter 5: Bumping Projects	59
Completing Project Development	59
How to Bump Projects	61
Chapter 6: Rolling Out Software	67
Considerations Before Rolling Out	67
What to Roll Out	68
How & Where to Roll Out	68
How to Roll Out From the PRC Account	70
Summary	70
Detailed Steps for Project Rollouts	71
Log In to PRC Account	71
Select Individual Project or Version Rollout	71
Select the Destination Realm	74
Launch the Rollout	76
How to Roll In (Extract)	77
How to Roll Back	78
Rollback Steps	78
Conflict Considerations	79
Chapter 7: Using Tracking Management	81
User Profiles	81
Timesheets	83
Queues	85
Chapter 8: Using Non-Critical Tools	89
Scanning for Conflicts	89
How to Scan for Conflicts	89
Resolving Conflicts	90
Setting Up the Project for QA	91
Tracing an Item's History	92
Comparing Items	94

Creating a Compare Project	94
Using the Knit Utility	94
Using the Review Source Items Screen Action Bar	96
Searching for Strings	99
Reviewing the Changed Items Stack	102
Managing Backups	103
Where Are the Backups Kept?	105
Project Development	105
Project Closeout	105
Project Rollout	105
Reviewing Backups	106
On Open Status 3 Projects in Development Realms	106
On Open Projects That Are Not Checked In	106
On Closed (Checked In) Projects	107
On "Live" Systems With PRC Installed	107
On "Live" Systems Without PRC Installed	107
Outside Editors or Utilities	108
Chapter 9: Tracking Queries	109
Overview	109
Menu Selections	109
Revision Log (by PRC)	110
Revision Log (by USER)	111
Revision Log (by ITEM)	112
Review detail docs	113
Documenting By Project	115
Review Change Metrics (Current)	116
Review Change Metrics (Historical)	117
Chapter A: Glossary	119
Chapter B: About PRC	125
<i>Index</i>	<i>127</i>

Table of Contents



Introduction

PRC[®] is a software development life cycle management application. In other words, it helps you manage or administer the development of your software from start to finish. This publication, the *Project Tracking Guide*, explains the various concepts and features of **Project Tracking** within PRC.

This guide assumes you already have knowledge concerning programming. If you need information or help concerning that subject, please refer to other sources of information.

Who Should Read This Book?

Focus and Audience

The focus of this manual is on the day-to-day mechanics of using PRC. The target audience is the *programmers*, *quality assurance technicians*, and *project managers* who will use PRC routinely to:

- Open projects
- Track changes to the software being developed
- Review changes, compare versions, and manage project content
- Close projects—mark them complete in preparation for rollout
- Perform rollouts—which may or may not involve programmers, depending on your organization's protocol

General Knowledge Required

This guide covers specific portions of PRC's overall functionality. This book occasionally refers to subjects covered in other PRC manuals. In some cases, base knowledge is assumed in the discussion of the subject. It is assumed that the reader has knowledge of: database administration, generally accepted programming conventions, terminology, software quality management, and some software configuration management.

If background information is needed on the subjects listed above, please see your system administrator.

What is PRC?

At its center, PRC is a *source control system*: its primary purpose is to keep track of changes to the source (programs and other components) of a software application. Beyond that, PRC encompasses all of the necessary peripheral management that supports its core purpose. PRC accomplishes all of the following:

- **Request Problem Reporting:** Front-line problem reporting, or the “help desk,” is a system of logging requests made by the users or management of an organization. Requests can be cancelled or resolved without ever going further through the software development life-cycle, which means you can use the Request/Problem Reporting system to track other non-software requests with your organization.
- **Project Management:** The Project Management system supports traditional functions such as estimating hours, assigning priorities and resources, tracking time, following percentage of work completed, and providing automatic feedback to the requesting user as the project moves through its life-cycle. Projects can be generated from requests or created independently.
- **Source Control:** The system keeps track of the software, protects it from unauthorized changes, logs changes, and provides clear methodologies for deployment, archives, and audits.
- **Test Management:** The system provides a framework for establishing test plans, implementation plans, back-out plans, and the ability to grow multiple plans into full regression and integration test plans.
- **Deployment and Rollback Control:** An automated mechanism for delivering all components of a project from one realm to another—even across machines. It insures that everything attached to a project is archived and can be rolled back at any stage.
- **Security Management:** The system nominates files and realms that may be changed or may not be changed, the conditions they may be changed, and by whom.
- **Auditing & Reporting:** The system provides complete auditing for compliance reporting and research.

Terminology

A list of common terms used in PRC are:

- A *request* is a potential or upcoming project.
- A *project* is a “unit of work.” A PRC project can be further broken down into *sub-projects*.

Many organizations manage large projects and track work broken down into several tasks. In PRC, however, the smallest unit of work is a project. A sub-project is simply a “child” project to some other “parent” project.

The basic tenet is that when a programmer is making a change, that change is being tracked against one single project or sub-project.

-
- A *version* is a collection of projects. A version can be further broken down into *sub-versions*, which are still simply a collection of projects.
 - A *realm* is an environment—a specific region (that is, directory)—that is governed by a particular set of rules regarding what change can be initiated there and by whom; for example, DEV, TEST, and LIVE are realms.
 - An *item* is a file or object of whatever is considered to be your source. PRC tracks changes made to source items.

PRC is a project-oriented (or “project-centric”) source control system. Everything is performed against projects. Many PRC entities or documents may be in use simultaneously, including: sub-projects, super-projects, versions, master versions, and customer requests. All of these other entities are related to a project. In PRC, the project is king.

There are dozens of different words that mean the same thing, depending on your organization; for example, military implementations use SCRs (Software Change Requests) to represent **tasks**. It is possible to “translate” PRC to match your corporate culture, whether it is the FDA, the DoD, or a small software company.

For further definitions of terms used within this document, please refer to the glossary at the back of this manual.

PRC & Other Source Control Systems

If you have ever used a source control system before, some portions or areas of PRC will be familiar, while other features or areas may be different and harder to grasp (at first).

Basing Work on Projects

The term “project-based” means something different in the PRC environment than it does in a Visual Basic environment. In Visual Basic, a “project” is designated—and controlled—by a file with the **.mak** extension. This **MAK** file is the controlling element and references all other components: the program, the screen form and other objects, methods, and so on. Each of those components is “versioned” independently, with the **MAK** file keeping track of which version is current and in use.

PRC is not structured in that fashion. A PRC project emulates a Visual Basic project in some ways: it is a controlling element that keeps track of what components (programs, processes, fields, screens, dialog boxes, error messages, and so forth) are included in this project.

PRC versions each component independently in archives. Program files contain all of the current, in-use programs, called by their real/original names; but, any program that has ever been modified by a PRC project is archived in its various past versions. Taking that approach can recreate any version or point in time (necessary for ISO 9003, IEEE, and DoD compliance) and *still* operate in the U2/Multivalued environment.

Checking Out Components

Years back, checking out components in a mainframe environment meant walking down the hall and requesting the tape. While you had the tape, it and all the components on that tape were completely removed from the library. No one else could edit your program because you had it physically in your hand.

Our database development environment makes that kind of total ownership unrealistic (if not impossible). Even knowing in advance what components you need to modify on behalf of a project is not relevant in today's programming environment.

If you feel your organization absolutely must have the ability to pre-check out components, PRC can support that in various ways: turning on Check-Out Mode in the user profile, creating "sandboxing" conventions and indicating them on the Preferences screen, or a semi-pre-checkout approach where each component that is edited is pulled from another account (for example, from the production library) at the time of edit.

What is more convenient in our programming environment is a first-come, first-served approach to checking out components. In other words, if you edit a component and no other project currently has it checked out, it is "belongs" to you:

- If you do not change or edit the component, it is released when you exit.
- If you *do* make a change, the component is checked out to your project. Others may still be able to edit the item, depending on their clearance level and the priorities of your project and theirs.

Checking out components, clearance levels, and priorities are explained in the *Administration Guide* and must be defined by your management team.

Methodologies For Archiving

People who have used a source control system in the past often ask about archiving and the methodologies behind the process.

One archiving method used by popular "freebie" source control systems in UNIX (SCCS or RCS) is to archive the deltas between one version and another only. This method saves on disk space; but, it requires storing and intelligent parsing of a kind of pseudo-code. Archiving small deltas between two versions requires a pre-compiler to dig through old versions and becomes more and more time-consuming.

Other source control systems archive full working copies of all components every time they are changed. These copies devour a great deal of disk space and can become equally time-consuming to pre-compile.

PRC takes a middle-of-the-road approach. While a project is active, PRC stores a copy of a component every time it is changed or modified. The system rotates through a maximum number of copies, which is determined on the user's profile, each day (the default is three). If the same program on the same project is edited 15 times in one day, a full-copy archive of the program exists from the first time it was edited today, as well as the last three times.

When you close a project in PRC, all of the intermediary archives are discarded and the first copy from the first day is archived into a separate archive file. The copy as it is delivered is also archived, and the copy that is replaced during a rollout/unravel is also archived. In the long run, then, there are three full working copies of every component changed on a project, when the project is completed.

In addition, you have the option to archive the software you are delivering—and the software that it is replacing—when you perform a rollout.

PRC includes a split-screen editor that displays the differences between any two versions of any component. It also provides easy maneuverability to merge them or to revert them.

Where This Book Fits In

Projects go through several *states* or stages, from the moment a person thinks something should be added or changed to the moment it is included in the software application for general use. PRC assists you in tracking the projects as they go through their various states along the development path.

PRC assigns a corresponding *status number* to each state for easy tracking. PRC is flexible, allowing you to tailor the meaning of some states according to your needs. The meaning of other states, however, are “fixed” and cannot be changed.

In general, projects in PRC go through the following states:

- 1. The project is formally requested in order to fill a need or correct an issue.**
- 2. The project is reviewed by the appropriate person and approved for development.**
- 3. The project is started by a developer and becomes “active.”**
- 4. Development of the project is completed and it is waiting to be tested.**
- Testing the project is started by a member of Quality Control or Quality Assurance.
- The project passes testing and is waiting to be incorporated into the overall product.
- The project is delivered into the software system.

After the first four states, your states or stages may vary widely from this example. You can also skip the first two states, so that you can open projects quickly in a pre-activated state of **3**. The meaning of states **1–4** is hardwired in PRC.

[Figure 1](#) illustrates the project states and state-changing processes within PRC:

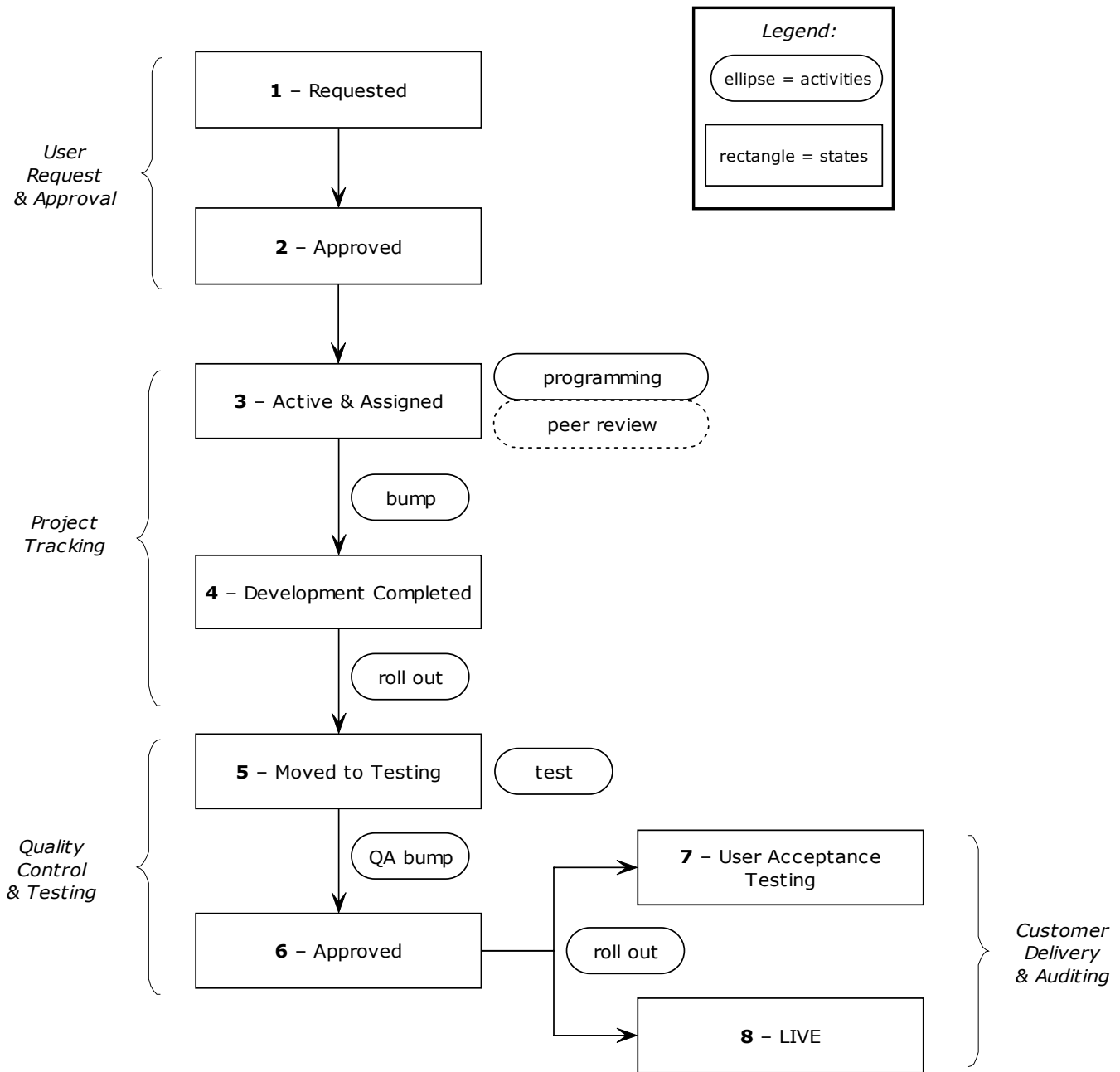


Figure 1: An basic, simplified example of PRC project states and the books that cover them

More PRC Documentation

Information about PRC is available in the following publications:

- *Installation Guide*
For installing all of PRC and configuring enough of PRC to get you going.
- *Administration Guide*
The definitive guide to PRC's configuration and security settings.
- *Project Tracking Guide*
The options for tracking changes during the development process, as well as rolling out projects.
- *Software Quality & Testing Guide*
Explains options for automating and managing the quality testing process.
- *Software Knitting Guide*
The definitive guide to merging software using the split screen view.
- *Software Request & Acceptance Guide*
Explains how a non-development site uses PRC to make requests and accept the projects derived from those requests.
- *Software Receiving & Auditing Guide*
Explains how a non-development site uses PRC to review and accept delivered software changes.
- PRC Tip Sheets (Tech Bulletins)
Each one addresses particular functions and FAQs.
- *PRC Reports*
Descriptions of available reports, as well as information for creating custom reports.

Telnet Programs, Character Mode & GUI Mode

The most common telnet applications used by U2/Multivalve developers (in 2005) seem to be Wintegrate, Accuterm, and SBClient (for those using SB+).

Other telnet applications can certainly be used and in almost all cases some initial effort must be put into creating an interface between PRC and the mode/emulation that your particular telnet application is using.

In the case of SBClient, you can use SBClient in either the traditional or "old school" character-based mode in which you navigate strictly using keys and function keys or the graphical user interface (GUI) mode using your mouse as well as the keys on your keyboard.

PRC was written in SB+ in character mode, so navigation is sleek and consistent throughout. All examples and screen captures are shown in character mode.

If you are using Wintegrate or Accuterm certain components can be presented in GUI mode (as of this writing) and there is a large development focus on this and other client-based GUI front-ends. Check with SJ+ System Associates about what you are hoping to use for a front-end to your development environment.

Menu Navigation

In character mode (Figure 2), you can select a menu option by performing one of the following:

- Use the arrow keys to move to the desired selection, then press **Enter** or the spacebar
- Press a key that corresponds to one of the underlined letters, such as **R** for Review Source Items, **Q** for BUMP/Quick Screen, and so on

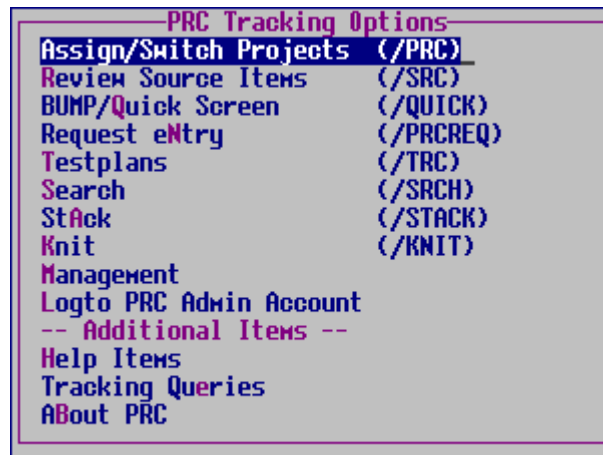


Figure 2: Tracking Options menu in character mode

In SB+ graphical user interface (GUI) mode (Figure 3), you can select a menu option by performing one of the following:

- Use your mouse to point to an option and left-click
- Press the **Alt** key, release it, and then press a key that corresponds to one of the underlined letters; for example, **Alt Q** for BUMP/Quick Screen, and so on

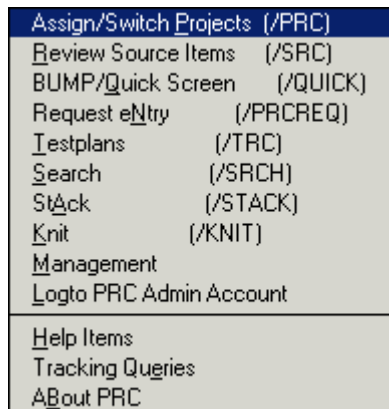


Figure 3: Tracking Options menu in GUI mode

- Just as in character mode, you can use the arrow keys in GUI mode to move to the desired selection, then press **Enter** or the spacebar

To go back to a previous menu level, use the **Esc** key.

If you are in GUI mode, some buttons that are labeled **sF5** (or something similar). That is shorthand for you to press, in this example, **Shift+F5** to display the requested information.

Standard Function Keys

From any screen in PRC, you access various features and subscreens by using function keys. The functions keys that pertain to a screen are listed along the bottom of that screen. Several functions are reserved and always perform the same purpose:

- **F1** — Help, which displays a message describing the current field
- **F2** — Save or Accept, which saves the values on the displayed record and updates the system
- **F3** — Intuitive Help, which lets you select a value for a field from a list of existing values
- **F4** — Delete, which removes the displayed record and does not save its values
- **F5** — Drill-down, which PRC uses to let you view and modify further information about an item
- **F10** — Action Bar, which displays an additional menu bar at the top of the screen (only available if it is listed at the bottom of the screen)

Screen & Field Navigation

Screen

To exit a screen without saving any input values you may have entered, use the **[Esc]** key.

Press the **[F2]** key to both accept the current data in a screen and exit the screen. You can press **[F2]** at any time, but the system will allow you to exit only if you have entered values in all mandatory fields.

If the system displays **INS** in the lower right corner, the screen is in full-text mode. Whenever it is in full-text mode, you can press **[Enter]** to add lines. You will have to press an additional **[F2]** to close that particular field and move to the next one on the screen.

Field

To navigate through the fields of a screen, use **[↓]** or **[Tab]** keys to move forward from field to field, and **[↑]** and **[Alt+Tab]** keys to move backward to previous fields.

To clear a field and make its value null, use the backslash (\) character as the first character in the field. You can press the **[Delete]** key repeatedly to delete one character at a time.

To invoke other features and processes, use the slash (/) character. When you press the slash character, the system displays command line in which you type a command for the feature or process you wish to invoke. In other words, it is a two-step procedure. In this book, the text will not explain the command line every time and will abbreviate the two commands; for example: to call up TCL, type **/TCL**.

Text Conventions

The text formatting in this document uses the following conventions:

Table 1: Typographical Conventions

Font	Description
Bold San serif	The name of a PRC module or folder
Bold	The name of a menu, menu item, toolbar button, or icon
Regular	The name of a screen, dialog box, property sheet, or tab
<i>Italic</i>	The name of a control (field, button, or drop-down list) within a screen or sub-screen
Courier	A path name, system value, or in general, text displayed by the computer
Bold Courier	File names, file extensions, or in general, text to be entered by the user or system administrator
Enter, Tab, Esc	A key on the keyboard
Ctrl+S	A key combination where you press the keys simultaneously
Alt P	A key combination where you press the keys one at a time in the order presented



Projects, Versions & the PRC Menu

Projects go through several *states* or stages, from the moment a person thinks something should be added or changed to the moment it is included in the software application for general use. PRC assists you in tracking these various states along the development path by assigning a corresponding *status code* to each state.

An introduction to the PRC menu for Project Tracking is included, explaining which options you use and how they relate to project states.

Projects can be grouped or associated with *versions* or super-projects, as well as *sub-projects* and *system sub-projects*.

The Project States Defined in PRC

One of PRC's strengths is its flexibility: you can define PRC to perform in the manner that is best for your organization. You can follow the default project states or tailor them according to your organization's methodology and practices.

A Project's Development Life Cycle

The following list presents the default states defined in PRC:

1. The project is formally requested in order to fill a need or correct an issue.
2. The project is reviewed by the appropriate person and approved for development.
3. The project is started by a developer and becomes "active."
4. Development of the project is completed and it is waiting to be tested.
5. Testing the project is started by a Quality Control or Assurance (QC/QA) staff member.
6. The project passes testing and is waiting to be incorporated into the overall product.
7. The project is delivered into the software system.

After the first four states, your states may vary widely from this example. You can skip the first two states, so that you can open projects quickly in a pre-activated state of **3**. The meaning of status codes **1-4** is hardwired in PRC.

The Programmer's Critical Path

Figure 1-1 illustrates the two project states within PRC that programmers work with most, as well as the two state-changing processes (known as the “critical path” for development):

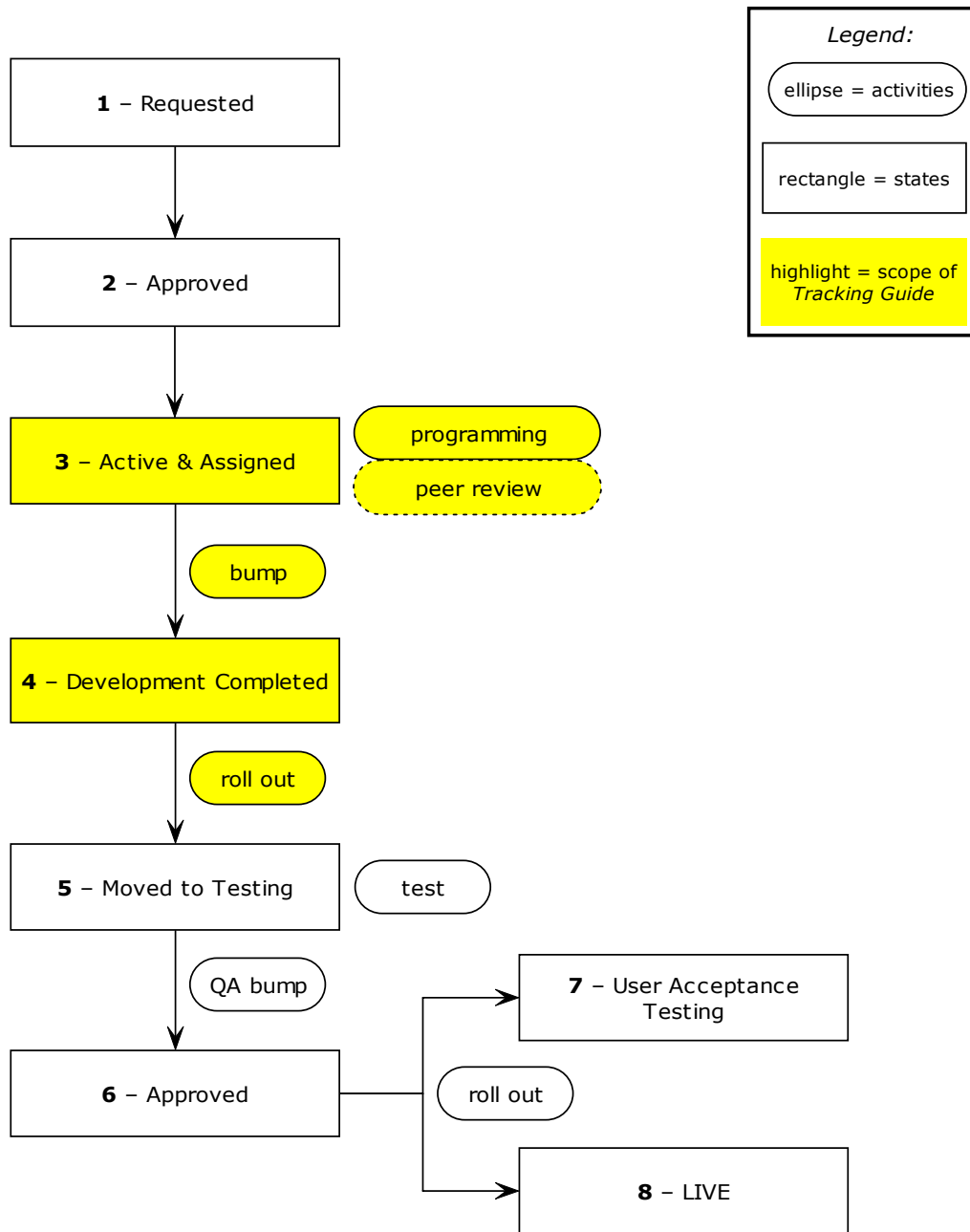


Figure 1-1: PRC project states and functions that programmers work with most

A more detailed explanation is contained in [Chapter 2, “The Programmer’s Critical Path.”](#)

The PRC Menu for Project Tracking

After PRC has been installed to the appropriate accounts, you can access the PRC menu from TCL by typing **PRCMENU** (or the keyword that your company decided upon during the installation/training visit). From any other PRC menu or prompt, you can type **/PRCMENU**, as shown in [Figure 1-2](#):

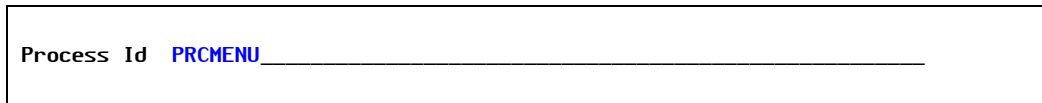


Figure 1-2: The PRCMENU command

After you press the **Enter** key, the system displays the Tracking Options menu (PRC menu), as shown in [Figure 1-3](#):

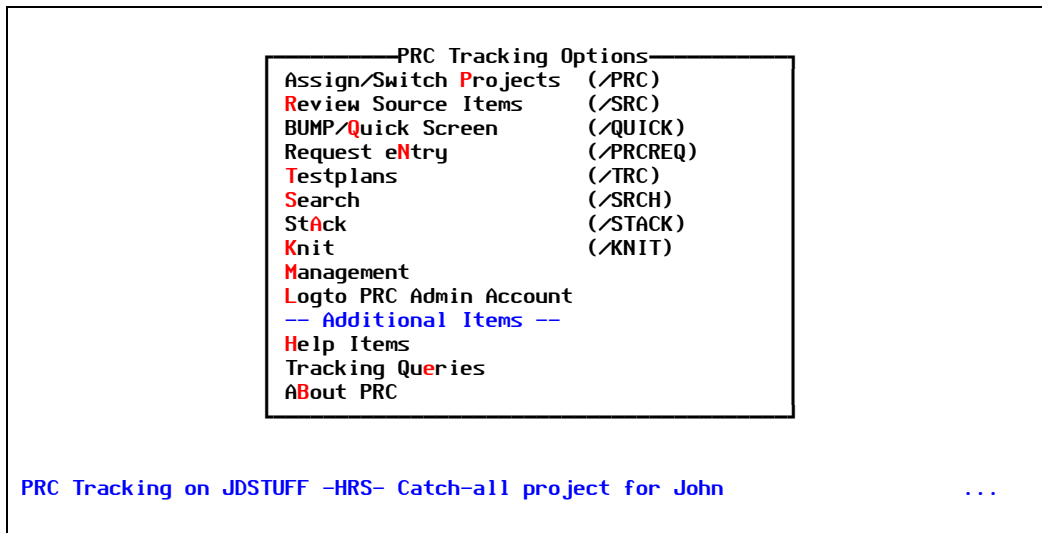


Figure 1-3: The PRC menu

You can access each function one of two ways, by:

1. Typing the slash character followed by a PRC command (as noted on the right)
2. Using keys that correspond to the menu:
 - Typing a “hot” character (a letter that is colored differently than the rest)
 - Using the arrow keys

For example, if you want to assign or switch projects, you could either:

- Type **/PRC** from anywhere in the system
— or —
- Type **P** from the menu

From any account which has had tracking installed, PRC users may access a variety of functions to support their daily work effort. All of these functions may be found through the PRC menu (**/PRCMENU**), either as a direct option or as a sub-option. Every feature a programmer must use is documented in this guide and every feature documented here is accessible from this starting point.

The advantage of using the shortcuts (slash commands) is that you do not have to go to the PRC menu. If you remember the commands, you can be working anywhere in the system and simply type the command to see the PRC screen. From there, you can either perform a function or simply review some information, then return to your work.

Tip

*If a user wants to memorize just one command, it is **/PRCMENU**. From there, a user can access every PRC feature and function.*

After you select an option from the PRC menu and processing is complete, the system redisplay the menu and you can select another option.

As a programmer, your critical path will have you work mostly with the **/PRC** and **/SRC** options. The vast majority of your work, however, will be in the **/SRC** screen.

Super-Projects (Versions) & Sub-Projects

Versions

A *version* is, in simple terms, a rather “loose” collection of projects. A version can be created at any time and any collection of projects added to it. Projects can be added to a version at any time, such as:

- When they are first created
- While they are in progress
- After they are completed

Once projects are collected together, they can be rolled out together by rolling out the version instead of the individual projects. Versions are a convenient way to roll out a group of projects, but you can still roll out projects individually.

The same project can appear on multiple versions.

Reasons to use versions include:

- To put together projects that are in conflict with one another. As long as you put them together on a version and deliver them together, and nothing on any project in the version conflicts with a project not on the version, the conflicts will not stop the delivery.
- To create a mini-release or master project to keep a deliverable for a subject matter.

Sub-Projects

Sub-projects are a tighter relationship. A sub-project must be created *as* a sub-project right from the start. You cannot go into a project and “make it into a sub-project later.”

The most common use for sub-projects is to manage iterations of testing; in other words, after a project has been marked as status **4** (Development Complete) and perhaps rolled to a test realm, if re-work is necessary, it may be preferable to open a sub-project rather than re-open the whole project. Why? Two main reasons:

1. So that an accurate history is kept of how many iterations to QA occurred and exactly what was re-worked.
2. So that just the item(s) undergoing re-work have to be re-delivered.

PRC can be set to understand that sub-projects don’t conflict among themselves. While a project is still checked out in DEV and under TEST, the same items that are on that project can be reworked against a sub-project and rolled out.

To create sub-projects that are simply numeric iterations, you can use the PRC assignment window, enter the parent project ID, and a period/dot (.) character. PRC will add on the next available sub-project number.

Some other types of sub-projects are available for special purposes:

- A sub-project that was created during spawning will sometimes have **Sn** (where **n** is the next number) or a realm ID as the suffix after the dot (period). For more information, see the tip sheet on spawning.
- A sub-project that was created for the purpose of modifying software in a separate SB+SYSID from the parent project will have the SYSID as the suffix after the dot (period).
- Sub-projects that are created to split a project between two developer users might have the developer user as the suffix after the dot.
- How the sub-projects relate to each other in terms of whether they conflict with each other and whether they may be rolled out independently can be set up on the Preferences screen (in the PRC account) according to project type.

Options associated with sub-projects by Type code. The value **ALL** can be used to indicate all types, as shown in [Figure 1-4](#):

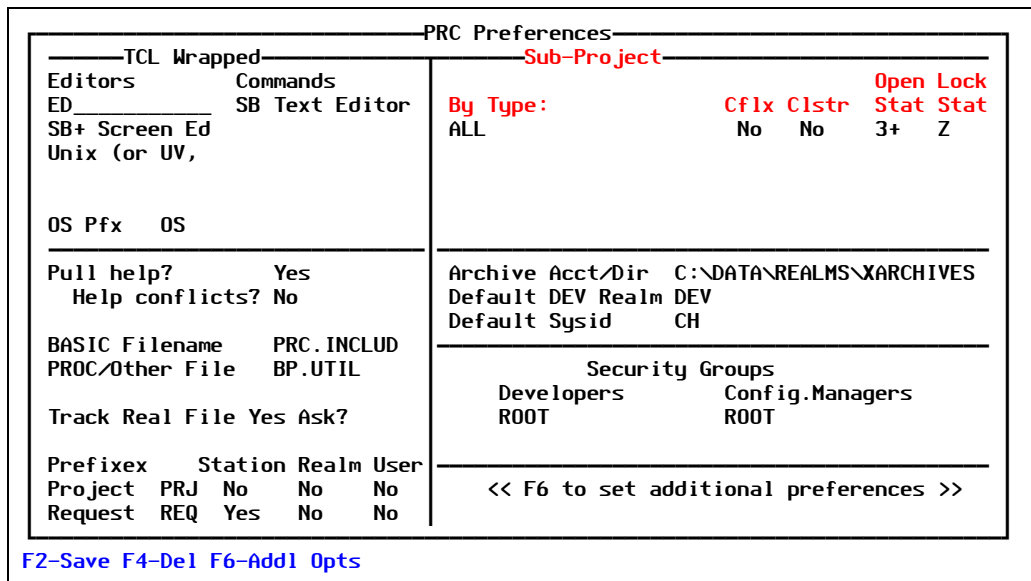


Figure 1-4: The Sub-Project area of the PRC Preferences screen

Explanations of the fields in the *Sub-Project* area of the Preferences screen are as follows:

- Cflx** Short for “Conflicts,” this field indicates whether sub-projects of this type should create conflicts among themselves and their own parent during development. Valid entries are:
- **Yes** — create conflicts
 - **No** — ignore conflicts
- Clstr** Short for “Cluster,” this field indicates whether a sub-project may be rolled out independently or together with its sibling and parent projects. Valid entries are:
- **Yes** — must be rolled out together, as a cluster
 - **No** — can be rolled out independently of other projects and sub-projects
- Open Stat** Short for “Open Status,” this field controls a parent project’s status when it has sub-projects opened against it. You can decide that sub-projects should only be opened:
- While the parent is still in development
 - Against parent projects that are in test status (if you are using the sub-projects for iterations of testing)
 - Against projects that are completely closed out and delivered to LIVE

The use of the + or – symbols next to the status number indicates that sub-projects may be opened against parent projects in that state or any state above (+) or below (-) that state.

Lock Stat

Short for “Lock Status,” this field indicates the status at which projects (any projects, not just sub-projects) can be conflicted against. This only applies if you have users whose security clearance allows them to create any conflicts (typically not).

You can open sub-projects in other realms, accounts and SYSIDs from the parent. You can password protect them, also. In each case, an extra prompt will warn you about the situation.

Sub-projects can be opened by one developer user against another developer user’s parent project with a special password, if applicable.

Note

Many of the configurable settings are tuned using the PRC Preferences screen and its sub-screens. For more information on the other sub-screens and settings, see the *Administration Guide*.

Sub-Projects & Syster Sub-Projects

What Are They?

Sub-Projects

A sub-project is a “child” project that is related to another project. It is named after the “parent” project with a dot-number suffix. For example, Project 12 might have sub-projects 12 . 1, 12 . 2, and so on.

Sub-projects can be non-conflicting among themselves (unlike siblings of the human variety). Sub-projects can also be forced to roll out together; for example, 12 . 1 would not be rolled out alone, but would have to be combined on a version with 12 and 12 . 2.

System Sub-Projects

A system sub-project is a sub-project that is opened for the purpose of associating software components from multiple systems with one project.

System sub-projects also use a dot suffix, but rather than using a number, they are given the SYSID as a suffix. For example, suppose Project 12 was associated with changes in AP. A few things needed to be moved in PU. Therefore, the system created Project 12 . PU.

Why Should I Use Them?

Sub-Projects

There can be a vast number of reasons to “relate” multiple projects together. A couple of the most common ones are:

- Track QA iterations

Say Project 12 is complete and rolled to the QA realm. During testing in the QA realm some items require re-work. A new sub-project (12 . 1) can be opened. Since the components on 12 have already rolled to QA, only the reworked components need to be re-delivered. Work against 12 . 1 does not need to conflict with work on 12 (see the Preferences screen, [Figure 1-4 on page 24](#)).

Sub-project 12 . 1 can be rolled out either collectively with 12 or independently.

- Delineate without conflicts

In some instances, it is desirable to have two or more people working on closely-related projects, but still have them be separate projects. If the sub-projects are set to not allow conflicts, one programmer can work on project 12 and someone else on project 12 . 1— even if they edit some of the same programs, the programmers will not be alerted about conflicts.

In this case, it would be important to enter **Yes** in the *Clstr* (Cluster) field, which is the setting that forces projects and sub-projects of this particular type to be rolled out together. Using that setting means you would never inadvertently deliver one person’s part of the overall project without the other people’s.

System Sub-Projects

SystemBuilder (SB+) is geared around the system ID (SYSID). For each system, there is a separate program file, process file, DEFN file, help file, menu file, and drivers file.

PRC was designed to work with SB+, which means that projects are aligned with a SYSID. Therefore, tracking can be automatic, the **/SRC** screen makes sense (one process column, one program column, and so on), and rollouts are straightforward.

It is recommended that a smart way to use SB+ is to not delineate the modules into separate system IDs. Once that structure has been used, PRC must follow through with it.

Syster sub-projects provide the capability to do work in more than one system charged against the same parent project ID.

How Do I Use Them?

Sub-Projects

To open a sub-project, use the Project Assignment screen (**/PRC**). Enter the parent project ID followed by a dot (or period). PRC automatically increments the sub-project number; for example, if it is the first one, the system makes it sub-project 1. If sub-project 8 already exists, PRC creates sub-project 9. The system imports basic information from the parent, such as title, description, class, and so on.

System Sub-Projects

Opening syster projects—projects that are intended to support the spanning of multiple System IDs—is more automatic. If you are assigned to a project for one module, then you switch over to another module, PRC will either:

- Remember the last project you were assigned to in the new module and reassign you to it
— or —
- Clear away the project you are assigned to and ask for a new project assignment (see [“User Profiles” on page 81](#))

Whenever you first log into an account or switch system IDs, PRC verifies that you have a valid project for the combination of this account and system. According to a setting on your user profile, PRC may automatically switch you to a project for the current account/system. If it does, you have to remember to switch yourself to a sub-project manually.

For example, you are in the AP system, working on project 3. You log onto the AR system and work for awhile on project 4. When you log back to the AP system, one of two scenarios can occur:

- PRC may (according to your user profile setting) automatically switch you back to project 3 because you were working on project 3 the last time you were in AP. In this circumstance, if what you want to do is work on an AP portion of the project in AR (project 4), you have to stop, go to **/PRC**, and create 4.AP as a new project.
— or —
- If PRC does *not* automatically switch you to some other project (either because you do not happen to have one in AP or because your setting on the user profile is not to do so), you will be prompted to create a 3.AP sub-project automatically as soon as you attempt to save a change in the AP system—while still assigned to the AR project.

For more information on PRC user profiles, see the *User Profiles* tips sheet.

The Programmer's Critical Path

As shown in the previous chapter, projects go through several *states* or stages. This chapter explains the steps programmers and development teams will follow during the development life cycle within PRC.

Figure 2-1 illustrates the four steps within PRC that programmers follow as their critical path:

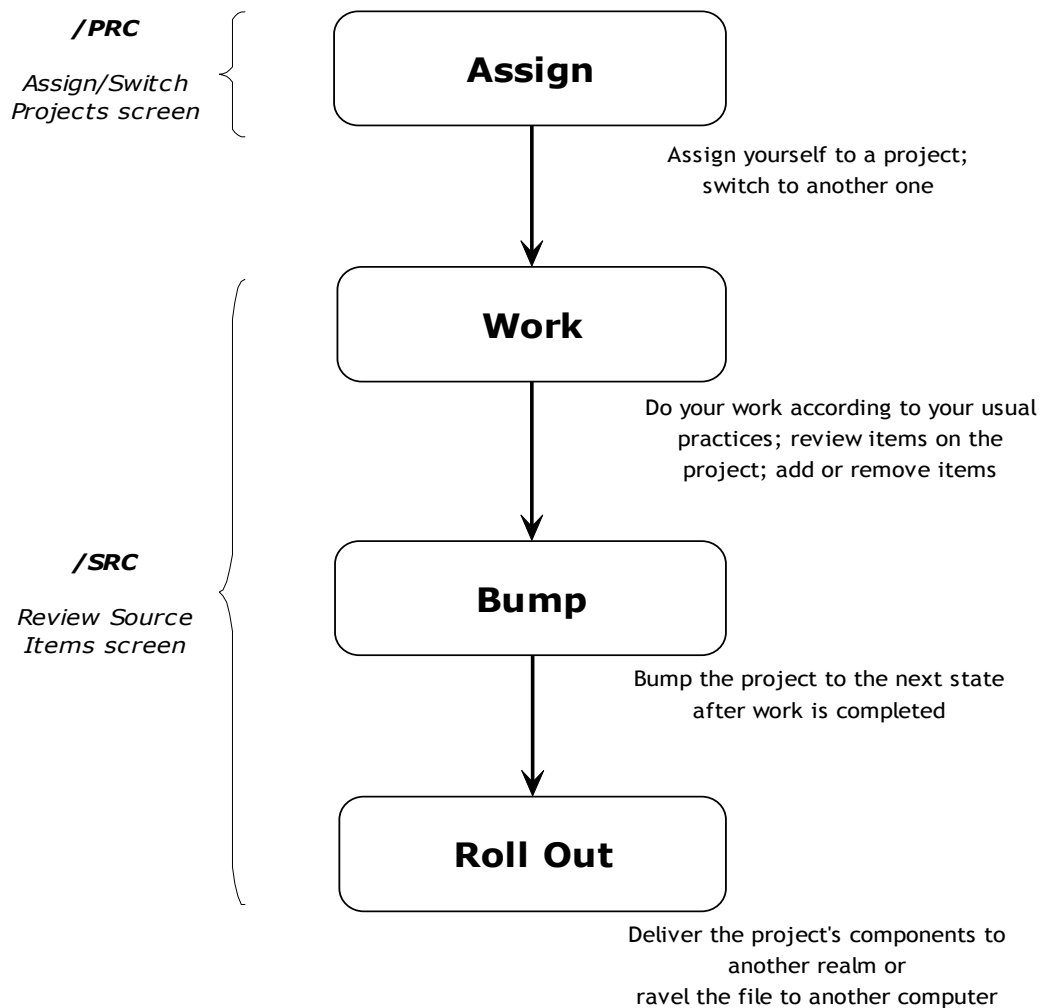


Figure 2-1: The "critical path" programmers follow within PRC

Figure 2-1 illustrates what is known as your “critical path.” As programmers, you do not want to be bothered with too many steps outside of developing your applications. So, consider working in state **3**, bumping the project to state **4**, and rolling out the project to state **5** to be your critical path—the most essential things you will use in PRC.

The Critical Path

As you develop your application, your interaction with PRC is summed up by the following steps:

1. Before you make changes, assign yourself to a status code **3** project.
2. You can review the items that are associated with your project, add items, and remove items from the project.
3. When you have finished the project, bump it up to status code **4** so the system knows you are done and can make sure the associated items do not have any conflicts.
4. Roll one or more projects out to another realm so someone can officially test the completed project.

Step 1: Assign

A project can be assigned to a programmer two ways:

- By a manager, after a user request has been approved and has become a project
- By the programmer, when the programmer has reached a stopping point with one project and wants to work on another one

In the context of this book, “assigning a project” refers to what a programmer does to switch from work on one project to a different project or creating a new project.

From the PRC menu, press **P** for **Assign/Switch Projects** or you can type the **/PRC** command. This option allows you to:

- Change the project against which the system is tracking your work
- Create new projects “on the fly,” usually for internal reasons or projects that are transparent to the end users

For more detail on assigning projects, see [Chapter 3, “Switching Between & Assigning Projects”](#) on page 35.

Step 2: Work

Once you open or switch to a project, you edit source code and open items as you normally would. The difference now is that PRC:

- Tracks every item you change and associates them with the current project, realm, and account
- Checks out those items so others cannot modify them
- Records any new items you add to the project
- Records all items you delete from the project
- Tracks all items you move to other projects

From the PRC menu, press **R** for **Review Source Items** or you can type the **/SRC** command. *This option is the central feature of PRC.*

When you enter a project ID in the Review Source Items screen, the system populates the formatted screen with all software components that are checked out or have been modified against that particular project. You use this screen to:

- *See* what is on the project
- *Add* items to the project manually
- *Remove* items from the project, either by flicking the item:
 - To another project, thus keeping the changes made to the item
 - Back to the current project, thus removing the changes

You can launch several helpful utilities from this screen, also.

For more detail on using these features, see [Chapter 4, “Working on Projects”](#) on page 43.

Step 3: Bump

Once you have finished working on a project, the project must be marked complete so that the system knows it is ready to roll out (be moved to another realm for testing). Marking a project as finished, from a development perspective, means changing the project state from **3** (Active/Assigned to a programmer) to **4** (Development Complete).

The term “bump” is used to describe when a project changes state, but the software is not delivered. The process of changing from status **3** to **4** is known as *bumping*. A rollout also changes a project’s state—but the software is moved from one place to another.

The bump from **3** to **4** examines and “scrubs” the project and may require some decisions be made by the programmer. However, everything remains in place in the development realm and the project is marked status **4**.

Bumping can be accessed or launched from three different places within the development account:

- **/BUMP**
- **/QUICK** (or access Quick from the menu) and choose **F7**
- **/SRC** (or access Review Source from the menu) and choose **F7**

For an explanation on using this feature, see [Chapter 5, “Bumping Projects”](#) on page 59.

Step 4: Roll Out

Delivering a project's software components from one realm to another, or into a ravel file for delivery to another (computer) system.

For information on rollouts, see [Chapter 6, “Rolling Out Software”](#) on page 67.

Additional Menu Options Outside the Critical Path

Some handy functions and tools are available that are not necessary to the critical path (completion and delivery of a project). These tools are very helpful and you find them on the PRC menu.

- Press **S** for **Search** or type the **/SRCH** command
This option searches for one or more specific strings within the entire application or within designated “file sets.”
- Press **A** for **Stack** or type the **/STACK** command
Similar to a TCL stacker, this option presents a view of all items that have been modified by that user in reverse chronological order. From this screen, you can eliminate redundant listings, edit any item in the stack, view a history trace of any item, or view all items that are backed up for a particular software component.
For more detail on the two features mentioned above, see [Chapter 8, “Using Non-Critical Tools”](#) on page 89.
- Press **H** for **Help Items**
This option provides you with more detailed explanations that go beyond using the **F1** help.
- Press **E** for **Tracking Queries**
This option leads to a menu of inquiries and reports regarding the status of work currently in development.
For information on the **Tracking Queries** menu, see [Chapter 9, “Tracking Queries”](#) on page 109.

- Press **[K]** for **Knit** or type the **/KNIT** command

Knit is PRC's split-screen merge utility.

For more information and detailed explanations, please see the *Software Knitting Guide*.

Switching Between & Assigning Projects

Critical Path Step 1

A project can be assigned to a programmer two ways:

- By a manager, after a user request has been approved and has become a project
- By the programmer, when the programmer has reached a stopping point with one project and wants to work on another one

Project assignments by a manager are covered in the *Software Request & Acceptance Guide*.

In the context of this book, “assigning a project” refers to what a programmer does to switch from work on one project to a different project.

Using the PRC option **Assign/Switch Projects (/PRC)** allows you to:

- Change the project against which the system is tracking your work
- Create new projects “on the fly,” if the user’s particular security clearance allows for it

Switching to Another Project

IMPORTANT

In most situations, programmers can engage projects that are status **3** (Active & Assigned) only. It must be assigned to the particular realm, account, and system in which you are currently logged and to which you have been assigned.

In some cases, a programmer can also assign himself to a project that is in status **2**, which means that a request (requested project) has been approved. This ability is appropriate in IT shops where the manager is not concerned with who works on which project. Instead, the manager is just interested in getting them approved—then programmers can assign themselves from a pool of approved (status **2**) projects.

A project may have been opened as a status **1**, then reviewed and approved through some process and marked **2**, or—if the PRC Request sub-system is in operation—the user may have logged a request and the act of approving that request will have created a project in status **2**.

When a programmer attempts to assign herself to a status **2** project, she will be prompted to activate and assign this project to herself. This action bumps the project to status **3** and assigns it to her.

You can be assigned multiple open, active projects. However, PRC actively tracks your work against just one project at any one time per login session.

For example, you have been working on a project, but you cannot continue because you are waiting on someone else. You want to work on a different project that has been assigned to you. Therefore, you “switch” from the first project to the second using the **/PRC** screen.

Note

Although it is not recommended, there are ways to assign multiple programmers to a particular project—or better yet, ways to create associated sub-projects for different programmers working on the same overall task.

For more information, see the “Multiple Programmers on a Project” tip sheet.

PRC has an automatic timing feature that allows it to track actual start/stop/elapsed time that each programmer spends assigned to each project. To turn this feature on, go to the user profile. For more information, see [“Timesheets” on page 83](#).

The main use for this feature is seen when you are building timesheets and then press **F3** to look up a timesheet. The system will pull the details from the automatic log, if it has been turned on.

If the timing feature is on, the timer is set on or off by project by using the **/PRC** window to assign and switch projects.

To switch from one project to another, follow these steps:

- 1. Press **[P]** for **Assign/Switch Projects** from the PRC menu (Figure 1-3 on page 21) or type the **/PRC** command from anywhere in the system.

The system displays the Assign/Switch Projects pop-up screen, as shown in Figure 3-1:

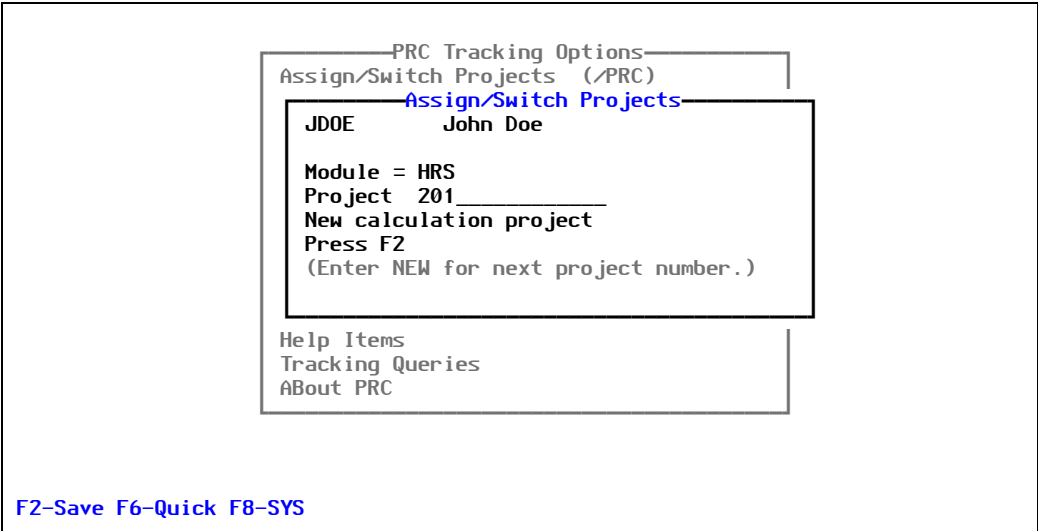


Figure 3-1: Assign/Switch Projects pop-up screen

The system displays your user ID and user name, according to your user profile, and the module and project in which you are currently working.

- 2. Enter the appropriate project ID directly in the *Project* field or press **[F3]** to select from a list of existing projects.
 - a. If you press **[F3]**, the system requests which type of lookup function you want to use, as shown in Figure 3-2:

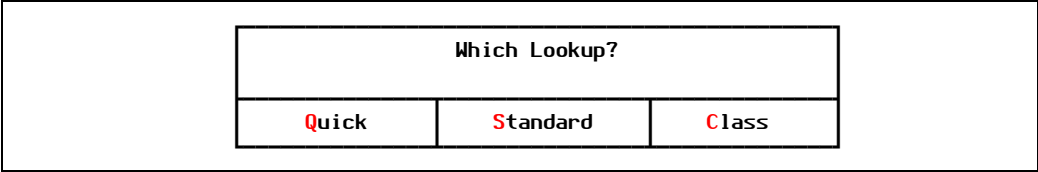


Figure 3-2: Which Lookup screen

If you select *Standard*, the system prompts for several field values, as shown in Figure 3-3:

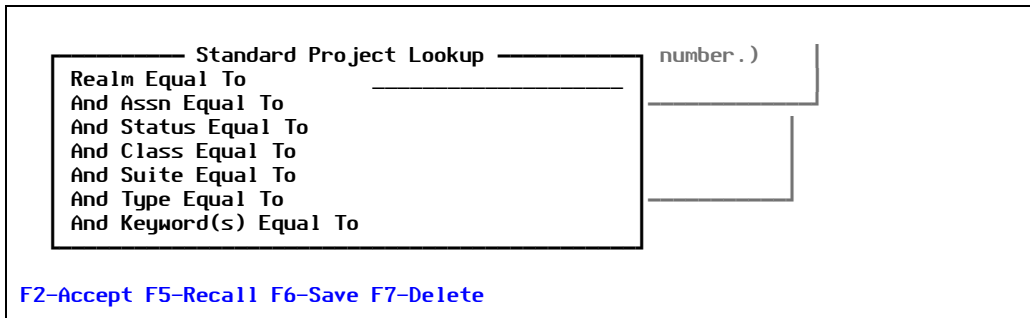


Figure 3-3: Standard Lookup screen

If you select *Quick*, however, the system allows you to select from a list, as shown in Figure 3-4:

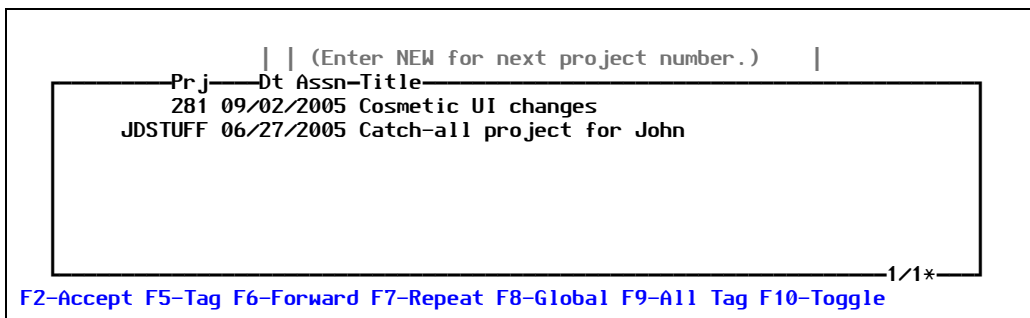


Figure 3-4: Quick Lookup screen

The Quick lookup actually produces the same list as the Standard lookup—it just uses the current realm, current user, and *Active* status for default values. The Quick lookup displays only those projects to which you can assign yourself at this time.

- b. Select the project to which you want to switch.

After you select another project, the system display that project in the Assign/Switch Projects pop-up screen.

Note

You can have PRC call custom programs to validate or format project IDs.

For more information, see the “Process Slots” tip sheet.

3. Press **[F6]** (Quick) to modify the title and description of the project.
You can also use **[F6]** to “bump” the status up or down. For more information on bumping, see [Chapter 5, “Bumping Projects”](#) on page 59.
4. Press **[F2]** (Save).
The system is now tracking you against the project you selected.

Creating a New Project

A setting in your user profile dictates whether you can create a new project.

If you are allowed to do so and are in the working/development account, you can create a new project by following these steps:

1. From the PRC menu (Figure 1-3 on page 21), press **[P]** for **Assign/Switch Projects** or type the **/PRC** command.

The system displays the Assign/Switch Projects pop-up screen, as shown in Figure 3-5:

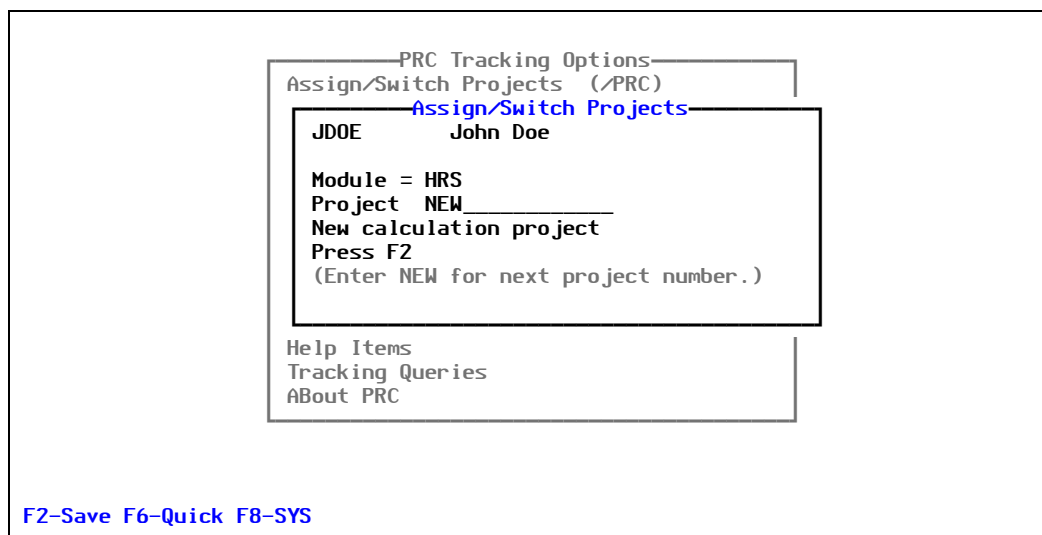


Figure 3-5: Assign/Switch Projects pop-up screen

The system displays your user ID and user name, according to your user profile, and the module and project in which you are currently working.

2. Enter a project ID in the *Project* field one of two ways:
 - By entering a non-existing alphanumeric project ID , which will ask if you intend to create a project with that exact ID
 - By typing the word **NEW** in the *Project* field, which will create a project using the next sequential numeric project ID available in the system (see the *Administration Guide*)

When anyone creates a new project “on the fly” like this from the working/development account, PRC makes a few assumptions based on the settings of the user’s profile, then displays the Quick screen described below is presented so that limited information may be entered about the project (title and description).

3. Press **[F2]** (Save).

The system initiates the new project using the ID you specified or a temporary ID if you typed **NEW**, but it only contains minimal information. Therefore, the system displays the Quick Project Update screen for you to fill in more detail about this new project.

An example of the Quick Project Update screen is shown in Figure 3-6:

Assign/Switch Projects
 JDOE John Doe

Quick Project Update

Project	Prio	1__	Status	3	Active & Assigned
PRJ*QUICK19	Class				Version 2.1.0
Customer 1	Type	BUG	Bugs		Rank
	US				Reference

Title/Brief Description

Full Description

F2-Save F7-Bump

Figure 3-6: Quick Project Update for a new project

The *Prio*(rity), *Rank*, *Reference*, *Status*, *Version*, and *Class* fields are informational only. Their values default from the project’s definition in the Project Master.

Explanations of the fields you can edit are as follows:

- Project** The ID or number that represents the project.

Press F3 to select from a list of available projects.
- Type** Indicates the type of project, as defined by your organization. Useful for reporting and managing, you can also use it to establish the process model (what cycle/roles are involved).

Press F3 to select from a list of valid types.
- Version** The version on which this project is targeted to be delivered. (Your organization may use these for versions or projects.)

Press F3 to select from a list of available versions.
- Title/Brief Description** This field should be a meaningful one-liner that describes the project for screens and reports that cannot display a full description.
- Full Description** A full textual description of the project as requested by the user.

4. Enter a value in the *Class* field.
5. Enter information in the *Title/Brief Description* and *Full Description* fields.
6. Press F2 (Save).

The system creates the new project.

If you typed **NEW** in the Project field (Figure 3-5 on page 39), the system displays a temporary ID (as shown in Figure 3-6) until the project is created by pressing **F2**.

As soon as the project is created by saving it in the Quick Project Update screen, control returns to the project assignment window (Figure 3-5 on page 39) and your newly created project ID is displayed in the *Project* field.

Take note that when you went to the Quick Project Update screen, you created a project but you have *not assigned yourself* to that project—not until you press **F2** at the Assign/Switch Projects screen.

For example, the user just added a project called “Cosmestic UI changes.” In Figure 3-6, the system assigned it the temporary ID PRJ*QUICK19. After the user filled in the information and saved the project, the system assigned it a legitimate ID—the next sequential number or other project ID as per a custom program that your company has in place.

However, until you close that window—which creates the new project—and you return to the original assignment window and press **F2** again, you have not switched your current tracking project to the new project.

Working on Projects

Critical Path Step 2

Once you open or switch to a project, you edit source code and open items as you normally would. The difference now is that PRC:

- Tracks every item you change and associates them with the current project, realm, and account
- Checks out those items so others cannot modify them
- Records any new items you add to the project
- Records all items you delete from the project
- Tracks all items you move to other projects

PRC is designed to help you manage your project, not just track changes. PRC provides you with several tools to assist you in managing project development:

- **/SRC**, the feature that lets you review and manage the items with which you're working. *This option is the central feature of PRC.* It is the feature you will use the most and perhaps find the most helpful.

When you enter a project ID in the Review Source Items screen, the system populates the formatted screen with all software components that are checked out or have been modified against that particular project. You use this screen to:

- *See* what is on the project
- *Add* items to the project manually
- *Remove* items from the project, either by flicking the item:
 - To another project, thus keeping the changes made to the item
 - Back to the current project, thus removing the changes

You can launch several helpful utilities from this screen, also.

- **F5** (Drill), the function key you can press when the cursor is on a certain file, that lets you view that file's related components.
- **F6** (This Item), the function key you can press when the cursor is on a certain item, that lets you choose one of several courses of action for that item:
 - If the project is active, you can edit the item, trace its history, compare it to another item, or make notes about it
 - If the item is marked as deleted, you can remove it from the project, doom it for deletion, or ignore it

This chapter explains how to use these features.

Editing Items vs. Your Security Settings

In this book, the term “edit” refers to making changes. You can make changes through the use of common editors; for example, **AE**, **SE**, **vi**, and **ED**. It also refers to the use of SB+ development tools, such as **/PD**, **/FD**, **/SD**, or other editors or development tools your organization uses.

When PRC is installed at your organization, your usual practices for software development are unchanged—as long as you are editing items that are not in conflict.

Your ability to edit items depends on settings that management and/or the site administrator have determined for you. (For more information, see the *Administration Guide*.) If you try to edit an item that has been checked out to another project, the options the system displays depend on those settings.

Note

Whether or not you are allowed to modify an item, the system allows you to open the item in the tool or editor. However, if your profile settings dictate you are not allowed to make changes, PRC will display a warning and then discard any changes.

After you open an item in an editor, the system allows you to edit the code and make changes. You may think the changes are being implemented—but it is not necessarily true. When you exit from the editor, PRC examines the item, the item’s checked-out status, and your user clearance. If it determines:

- You did not make any changes, nothing happens.
- The item was already checked out and you modified the item (but your clearance does not allow you to modify it), PRC informs you that it is discarding your changes.
- Your clearance does *not* allow you make changes (to either that type of item or within this realm), PRC informs you that it is discarding your changes.
- Your clearance *does* allow you make changes (to either that type of item or within this realm), PRC may either:
 - Allow you to overwrite your changes onto the checked out copy of the item
 - or —
 - Require you to make a copy of the item for your own project with only the changes you just made

You will be advised and prompted with a dialog box to choose your course of action. Once PRC makes a copy (referred to in mainstream systems as a “branch”), your project can not be closed until the project that owns the original of the item has been closed. Once it is closed, PRC will take you through the steps to merge your changes into the previous project's changes. For more information about merging changes, please refer to [Chapter 5, “Bumping Projects”](#).

Reviewing & Managing Items in Your Project

Overview

You can use the `/SRC` command, which displays the Review Source Item screen, for the simple task of quickly displaying an overview of the software components that are checked out to a particular project. It is a quick and easy way to survey the current situation.

The Review Source Items screen has several complex, integrated, and powerful features. The functions make it an invaluable tool to manage the project.

Along the critical path, you use the screen to:

- Review what is on your project
- Add elements to the project
- Remove elements from the project, by:
 - Placing them on an alternative project, or
 - Undoing the changes

In addition, you can use several utilities to assist you in developing your product. You use these utilities by pressing the `F6` (This Item) key while the cursor is on a particular item.

Reviewing Items Tied to Your Project

You can use the Review Source Items screen for a formatted view of all changes tracked against (or checked out to) a particular project. The currently assigned project is the default; however, you can view a different project's details by entering another project ID.

If the project is not in `Active` status, the system displays a reminder that it must present the project in `View Only` mode. If you are not the owner of the project, you are not allowed to make changes from this screen.

Manually Adding Items to Your Project

As you make changes to source items, PRC adds them to the project automatically. In addition, you can add items to the project manually by simply typing them into the appropriate column on the screen.

As it does with automatically-added items, PRC examines any item you add. The system looks for conflicts and, if it is clear, will check it out to the project.

IMPORTANT

The “bump” process ([Chapter 5, “Bumping Projects”](#)) will offer to discard elements:

- That have been checked out but did not actually change
This means you can add items to a project if you expect to work on them—and be assured that you can clear them off if you do not end up changing them.
- That you changed but undid your own changes, which ultimately means that the item is unchanged

Manually Removing Items From Your Project

Removing items from a project can be trickier than adding them, depending on the situation. PRC will not allow an item to be removed from a project when the item has been changed.

If an item has been modified and you try to remove it, PRC prompts you to move the item to another project, instead. To what project you move the item requires some thinking and decisions. Moving items off one project and onto another (known as *flicking*) will end up meaning one of two things:

- You undo the change and place the item back in the available software pool
— or —
- You actually move the item from this project to another

This is also handy for those occasions when a change is made to a program and then removed, such as during debugging. If the end-result of the item is the same as it was when you checked it out, the bump will offer to remove it from the project for you.

How to Use Review Source Items (/SRC)

To review and work with the source items associated with a project, type the **/SRC** command from anywhere in the system.

For SB+ development environments, the system displays the Review Source Items screen that is similar to the one shown in [Figure 4-1](#):

Bump	Print	Notes	TOC	QA	Other
Review Source Items					
Project 20		(SJOSLYN)		Realm DEV2	
Testing with PSI upload		Reference		*Subs	Type ENH
3	Active			Items	Class PRC
Files		*Fld*Scr*Rpt	Processes	5	Programs
PRC.PROJECT		1 2 0	I*PRC.PROJECT*MAINT.		
PRC.REVISION		0 1 0	I*PRC.PROJECT*MAINT.		
			I*PRC.PROJECT*MAINT.		
			I*PRC.REVISION*MAINT		
			LOAD.OUT		
BP.UTIL		0 Defns			PRC. INCLUDES
					0
Text		Help Docs	Menus	0	Misc Edits
					SAVEDLISTS!L2000
					SAVEDLISTS!TEST000
					2

F2-Save F5-Drill F6-This Item F7-Bump F8-Scan F9-QA F10-Action

Figure 4-1: Review Source Items screen

This screen is specifically configured for SB+ applications. For other development environments, please see alternate configuration below.

For non-SB+ development environments, the system displays the version that is similar to the one shown in Figure 4-2:

Review Source Items					
Project 1		(SJOSLYN)		Realm D	
Cosmetic - start with USER (F6).				Items	18
3	Active & Assigned			Type	ENH
Files		Dict	NONS	0	Accuterm
					0
Universe	0	Unidata	0	Codes	0
				Includes	0
Pick	0	Other (X8)	0	Other (X9)	0
				Misc Edits	0

F2-Save F5-Drill F6-This Item F7-Bump F8-Scan

Figure 4-2: Non-SB+ Review Source Items screen

According to whether you are using PRC in an SB+ or non-SB+ environment:

- In an SB+ development environment, all but two columns are pre-configured for you: PROCESSES, PROGRAMS, DEFNS, TEXT, HELP DOCS, and MENUS. Two columns are not pre-configured (which, in the example in [Figure 4-1 on page 47](#), are set to BP . UTIL and PRC . INCLUDES).
- In a non-SB+ environment, all of the columns are configurable like these last two and the following field descriptions do not apply - they will be specific to your own development environment.

In all cases, the `Misc Edits` column appears last. It is a catch-all for any edits that are meant to be tracked but do not have their own column. Items in the `Misc Edits` column include the file name and the item name; for example, `OTHERPROGRAMS!PROGRAM.ONE`.

To review and work with the source items associated with a project, follow these steps:

1. Either press `[R]` for **Review Source Items** from the PRC menu or type the `/SRC` command from anywhere in the system.

Each time you invoke this screen on an active project, PRC gathers all new changes and displays them—a process known as “exploding” the project. While the system explodes the project, it displays a box on the lower right side of the screen that scrolls through the items on the project.

This screen is the central feature of PRC. When you enter a project ID in the Review Source Items screen, the system populates the formatted screen with all software components that are checked out or have been modified against that particular project. You can launch several helpful utilities from this screen.

When you select this screen, the system defaults to your current project. However, you can enter other project IDs to review other projects.

Each of the columns or sections on the Review Source screen are dedicated to a particular part of your application software.

Explanations of the fields that are common to the various environments are as follows:

Project

The code or ID that represents the project to which your work will be “charged” or against which your work will be recorded. The default is the project to which you are currently assigned. You can:

- Press `[Enter]` to review the items tracked against the default
- Enter another project (for the current realm and system) to review *that* project’s source changes.

Files This column lists files that have been assigned to this project; either you assigned them manually or PRC added them automatically because you modified them.

In an SB+ environment, each file has associated fields, screens, or reports that are listed in the **Fld*, **Scr*, and **Rpt* fields in the same row. Any of these items that change are associated with a particular file and are, in fact, stored in the dictionary of the file that they define or are defined by.

In a non-SB+ environment, dictionary items managed in a similar fashion, but in a single column.

To present this multi-level relationship, PRC displays a count of the fields, screens, or reports that have changed for each file in the same row or line as the file's name.

Press **[F3]** to select from a list of valid components to check them out (add them manually).

Press the backslash (\) key to check in components (either delete items or "flick" them to another project).

Misc EDits This column lists the "miscellaneous edit" items that do not fall under other categories..

Explanations of fields in an SB+ environment are as follows:

***Fld** Works in conjunction with the *Files* field.

Displays a count of the number of field definition items for the file listed in the *File* column that are included on this project.

The three fields **Fld*, **Scr*, and **Rpt* are unique on this screen in that you do not enter any data for them. Each column displays a numeric count of the number of dictionary records that have been modified for the associated file.

To see the names of these items that have changed, as well as to add or delete screens, fields, or reports from the project's revision control source list, place the cursor on the appropriate item count and use the **[F5]** (Drill) key. Once a sub-screen opens, you can "flick", add, and explore (**[F6]**) features work on these columns exactly as they do on the other columns of the source screen that do not have a drill-down capability.

To view, add to, or delete from the actual names of these dictionary records, press **[F5]** (Drill). For further explanation, see "[Drilling Down](#)" on page 53.

***Scr** Works in conjunction with the *Files* field.

Displays a count of the number of screen definition items for the file listed in the *File* column that are included on this project.

This field works in similar fashion to the **Fld* field.

***Rpt** Works in conjunction with the *Files* field.

Displays a count of the number of report definition items for the file listed in the *File* column that are included on this project.

This field works in similar fashion to the **Fld* field.

**Processes,
Programs,
Procedures,
Defns, Text** These columns list software components that have been assigned to this project; either you assigned them manually or PRC added them automatically because you modified them. The columns and items may vary by organization if you are on a non-SB+ environment; but if you are on SB+, they are listed in the order of the columns:

- Processes
- Programs in XXPROG files
- Procedures in PROCLIB files
- Definitions in DEFN records
- Text in XXUSERTEXT files

Press **F3** to select from a list of processes.

2. Work within the fields, as necessary:
 - If you want to add items, see [“Adding Items to Projects \(Checking Out\)”](#) on page 51.
 - If you want to remove items and either move them to another project or revert back to a version prior to any changes, see [“Removing Items from Projects \(Flicking\)”](#) on page 52.
3. Other options in addition to the standard **F1** through **F4** can be found by pressing any of the following function keys:
 - Press **F5** (Drill) on fields that have an asterisk to drill down to see component names for an item. See [“Drilling Down”](#) on page 53.
 - Press **F6** (This Item) to explore more information about an item. See [“Exploring This Item”](#) on page 54.
 - Press **F7** (Bump) to change the status of the project from **3** (Active & Assigned) to **4** (Development Complete). See [“Bumping the Project”](#) on page 56.
 - Press **F8** (Scan) to explore more information about an item. See [“Scanning for Conflicts”](#) on page 57.
 - Press **F9** (QA) to explore more information about an item. See [“Setting Up the Project for QA”](#) on page 91.

- Press **F10** (Action) to explore more information about an item. See [“Using the Review Source Items Screen Action Bar” on page 96.](#)
- 4. When you are done making any changes or reviewing the information, press **F2** (Save) to save the project.

Adding Items to Projects (Checking Out)

As stated earlier, you can add items to this screen and the project in one of two ways:

- Automatically, by simply working on the item in the usual fashion while the project is your current project
— or —
- Manually, by entering the item name into the appropriate column or by using **F3** to look up and select (see below)

Automatically

To add an item to a project automatically, follow these steps:

1. Press **P** for **Assign/Switch Projects** from the PRC menu ([Figure 1-3 on page 21](#)) or type the **/PRC** command from anywhere in the system.
2. Switch to the project in which you want to work or verify that you are already assigned to it.
3. Press **F2** to save the project assignment.
4. Open up an item you need to modify in an editor and make the appropriate changes.
Once you save the modified item, it is checked out to your current project.

Any item you change that is not already checked out to another project will automatically be checked out to the current project as soon as you make a change to it. After you make a modification, the item appears on this screen. It is that simple.

Manually

To add an item to a project automatically, follow these steps:

1. Either press **R** for **Review Source Items** from the PRC menu ([Figure 1-3 on page 21](#)) or type the **/SRC** command from anywhere in the system.
2. Place the cursor in the appropriate column (*Files, Processes, Programs, BP, Menus, or Misc Edits*).
3. Press **F3** to look up and select a component, or type the full component name yourself.
The system validates the column entries. If you want to check out an item that you have not created yet, PRC will warn you that the item does not exist and ask if you wish to check it out anyway.

Once validation is complete, the item is checked out to your project.

4. Press **F2** to save the project.

You cannot add items that are checked out on other projects this way.

Removing Items from Projects (Flicking)

To delete an item from the source screen (and, therefore, the project), follow these steps:

1. Place the cursor on the first character of a line that contains the item you want to remove.
2. Type a backslash or backslash character (\) over an item and press **Enter**.

This is the standard method in SB+ to delete the contents of a field.

3. When the system prompts you for a project ID (a “flick to” project) at the bottom of the screen, enter one of the following:
 - a. Enter another valid project ID to flick the item to that project.

In order to prevent changes that are not audited, the changed component must be moved or “flicked” to another project. If the change actually belongs on another project, or you want to put it there and the other project belongs to you (or you have “configuration manager” authority) enter that project as the “flick to” project. It is sometimes convenient to keep a “junk” project open for each user, so that items can be flicked there when it is not immediately clear what should actually be done with the changed item—but you know you want it off of this project.

If you flick the item to another project that is assigned to you, PRC removes the item from the current project and adds it to the one you specify—whether or not the item has been changed.

This action always requires some thought. Before flicking to another project, ask yourself: “Can my project function in the new realm without this piece? Will this piece operate in a standalone fashion with the other project?” It could, for example, call a new subroutine that is on this project and when the piece is rolled out to another account with the other project, it would cause a failure.

- b. Enter the same (current) project ID if you do not want the item on another project.

If your user profile is set up appropriately, you can enter the same project ID as the flick to project. For example, if you are cleaning up project 7, you flick the item back to 7. Flicking an item back to its current project is a trick that informs PRC that you really do not want the item anywhere else.
4. What happens next depends on changes within the item:
 - a. If the item has not changed, the system removes it from the project—whether you indicated flicking it to another project or the same project.
 - b. If the item *has* changed, the system automatically brings up the Knit split screen editor. It presents you with the current version containing your changes and the previous version before you made any changes on this project. Concerning the item, you can:
 - Replace it with the original backup version, and flick the item off of the project.

You can do this by typing **v** (Revert) in the Knit screen. For more information, refer to the *Software Knitting Guide*.
 - Flick it to an all-purpose “catch-all” project.

When the focus is on getting one project to rollout and you know that the project will survive without a particular component—but you are not sure where to flick the component—it is helpful to have a handy “junk” or general purpose project that you keep open at all times for this purpose.

You can go—frequently—into the Review Source Items screen for your catch-all project periodically and clear it out by moving items to their appropriate “real” project.

- If you have the password and authority, delete the project that actually has items checked out to it and tell PRC “just let go.”

This is described in the *Administration Guide* and should be used *with extreme caution*.

Drilling Down

If you press **F5** (Drill) from the Review Source Items screen (Figure 4-1 on page 47), the system “drills down” into a file. The system opens up a detail screen of the component names.

You can use **F5** on this screen only from the fields (**Fld*), screens (**Scr*), or reports (**Rpt*) counter columns.

To drill down, perform the following steps:

1. Place the cursor on the appropriate column and row and press **F5**.

For example, you place the cursor in the **Fld* column for a particular row. The number displayed there is the count of how many fields are tracked in this project for the file on the left.

The system displays the Dictionary Items box on the right side, similar to the one shown in Figure 4-3:

Review Source Items			
Project 50039	(JDOE)	Realm DEV2	
BILLING REPORTS		*Subs	Type BUG
3 Active	Reference 1	Items 20	Class BA
Files		Dictionary items	
	*Fld*Scr*	File	File Items
D.EMPLOYEE.PAYROLL	0 0	D.REPORT.WORK	10
D.REPORT.WORK	10 1		BBD.BILL.AMT
			BBD.DEDN.ARREARS.AMT
			BBD.EE.DEDUCT
			BBD.NET.BILLED
			BBD.PAY.DATE
			BBD.PLAN.ID
			BBD.SOC.SEC
			RPT.EMPNAME
			W.B.PLAN.ID
			W3.EE.ID
(NOT USED)	0 Defns		
Text	Help Docs		

F2-Finish F6-This Item

Figure 4-3: Results of drilling in the **Fld* column

When the system displays one of these sub-screens, the function keys listed at the bottom of the screen change.

2. Make any changes, as necessary.
Just as you can do on the other columns of the source screen that do not have a drill-down capability, you can “flick”, add, and press **F6** (This Item) to explore.
3. Press **F2** (Finish).
4. PRC records any changes made there, closes the sub-screen, and redisplay the Review Source Items screen with a new count.

To bypass these columns without displaying the pop-up sub-screen, simply press **Enter** or the down-arrow key (slightly faster).

Exploring This Item

When the cursor is resting on an item in the Review Source Items screen, you can press **F6** (This Item) to open several avenues of exploration on that component.

Options for an Active Item

When you press **F6** and the item is a regular software component, the system displays the options shown in [Figure 4-4](#):

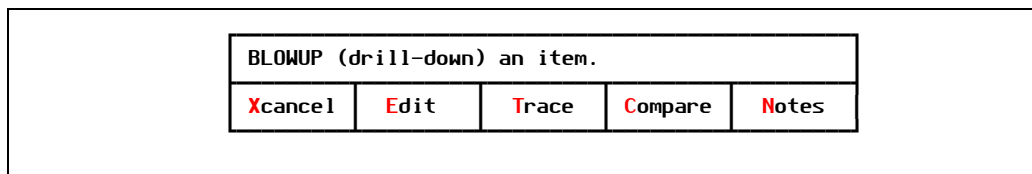


Figure 4-4: Exploring an item (drill-down dialog)

- **Edit**
Edit the actual item, invoking the editor designated on your user profile or choosing another appropriate tool.
- **Trace**
Trace the history of the item, in terms of all PRC projects on which it is currently checked out (there will be more than one project if the item is in conflict). In addition, the system displays the item’s history for all PRC projects to which it has ever been assigned.
For more information, see [“Tracing an Item’s History” on page 92](#).
- **Compare**
Compare the item against another item in another file or against this item’s own backups, which are kept automatically by PRC.
If you compare this item to a different item, the system uses the Knit utility (see the *Software Knitting Guide*).
If you want more information on backed up items, see [“Managing Backups” on page 103](#).

- **Notes**
Review notes that were entered during development. The system prompts you for whether you want notes on this entire project or just this item.

Options for a Deleted Item

During the course of development against a project, you may delete an item using an editor or some other tool (or if the DELETE verb itself is being tracked).

PRC marks such items on the project with a *DEL suffix, which you can see on the Review Source Items screen. The *DEL suffix is an ambiguous marker and such items prevent the project from being delivered. Deleted items can be addressed one of two ways:

- Collectively by using the BUMP function
For more information, see [Chapter 5, “Bumping Projects.”](#)
- Individually by pressing **F6** (This Item)

When you press **F6** (This Item) and the item has been deleted, the system displays the options shown in [Figure 4-5](#):

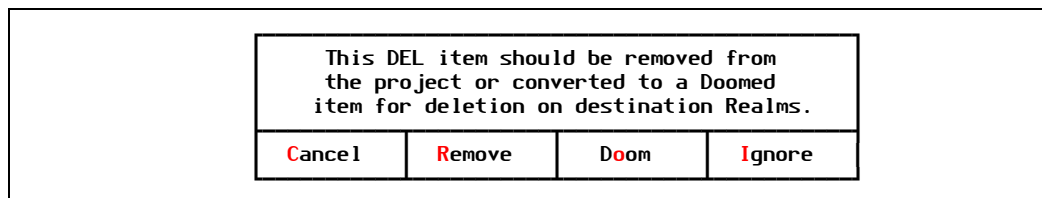


Figure 4-5: Deleted item dialog

- **Remove**
Remove the *DEL reference from the item, take no further action against the item, and take it off the project.
- **Doom**
Change the *DEL designation to *DOM (doomed) and delete the item from the destination realms during a rollout or unravel.
- **Ignore**
Temporarily ignore the deleted item. PRC will ask you what to do with this item again when you bump the project.

Options for a Branched Item

During the course of development against a project, you may have made a branch copy of a particular item (if your clearances are set to allow it). A branch is spawned when:

- a. You attempt to edit something that is already checked out to another project.
- b. If your clearance is set accordingly, PRC asks you if you would like to make your own branch copy.

- c. Again, depending on your clearance, PRC also asks you whether you want your copy to be pulled from the project before the conflicting project makes any changes against it.

This is typically what you would want. In some clearance settings, you will be asked if you want a branch copy, but then the system assumes (and does not ask if) you want the original from the archive before the other changes are made.

Once you make your own branch copy, PRC will always redirect you automatically into the copy when you subsequently edit the item.

Branch items are named `<item_name>&PRJ<nnnn>` and show up on the Review Source Item screen. Branched items can be addressed one of two ways, depending on whether the parent item is in conflict:

- If the item is no longer in conflict, you can knit the merged item back into the real item using the Knit split screen merge editor. Optionally, the system then deletes the branched copy.
- If the parent item is still in conflict, the system does not allow the merge (or the rollout).

When you put your cursor over a branch item that has a parent still checked out on another project, **F6** (This Item) will not offer any options. Instead, it display a dialog box that says this item is a branch copy and its parent item is still checked out on a particular project.

When you press **F6** (This Item) and the item is a branch copy, the system displays the options shown in [Figure 4-6](#):

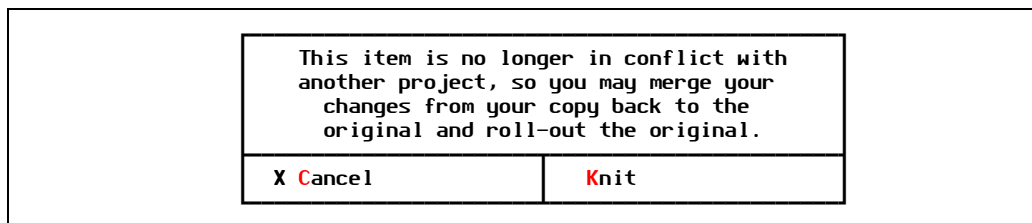


Figure 4-6: Branched item dialog

- **Knit**
If the parent item is clear, your version of the item and the parent item will be displayed and the Knit utility may be used to bring your changes back into the mainstream. If the parent item is still checked out to another project, this dialog box ([Figure 4-6](#)) will not appear and knitting into the parent will not be allowed.

For more information on the Knit function, see the *Software Knitting Guide*.

Bumping the Project

If you press **F7** (Bump) from the Review Source Items screen, the system changes or “bumps up” the status of the project from **3** (Active & Assigned) to **4** (Development Complete).

For more information, see [Chapter 5, “Bumping Projects.”](#)

Scanning for Conflicts

When a project is finished, it is time to take a hard look at the project. You must make sure it is ready to be rolled out to the customer site, test area, or wherever else it should be delivered.

During the course of your work, you had the opportunity to review the individual source items using the **/SRC** screen. Now the Review Source Items screen is useful for reviewing the project as a whole. Use this screen to ensure that everything that belongs on the project is there and anything that does not belong is removed. You can remove unwanted items by using the “flick” option (see [“Removing Items from Projects \(Flicking\)”](#) on page 52).

A conflict occurs when two or more projects have “checked out” or made changes to an item without making a branch copy of the item.

To discover if you have any conflicts, run the **[F8]** (Scan) function from the **/SRC** screen.

- If no conflicts exist, the system displays the dialog box shown in [Figure 4-7](#):

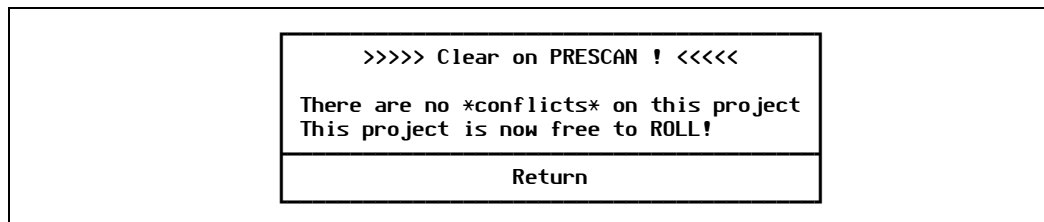


Figure 4-7: Result of pressing **[F8]** (Scan)

- If you are in conflict with other projects, you have a variety of options to resolve the conflicts. There is no single answer about what to do—it all depend on why you are in conflict.

Note

The system performs this same conflict analysis when a project is:

- Closed using the bump feature
- Rolled out from the PRC account

Using the **[F8]** option while you are still working on a project is not required, but is an opportunity to check for conflicts without leaving the account or closing the project.

A conflict occurs when an item is checked out to more than one project, or if the version that is checked out is a “branch copy.” For more information on conflicts and how to resolve them, see [“Resolving Conflicts”](#) on page 90.

Bumping Projects

Critical Path Step 3

The process of changing a project's status without moving the software is known as *bumping*. A "bump" can occur when:

- A developer finishes work on a project
- A tester completes testing on a project

A rollout also changes a project's state—but the software is moved from one place to another. The term "bump" was coined to indicate those situations when the project changes state, but no software is moved.

Because this document is written for programmers, bumping will refer to the process a programmer performs after work on a project is done.

You can bump a project from two screens:

- Quick Project Update screen
- Review Source Items screen

Most developers bump from this screen because it affords them an opportunity to review the items that are on the project—very convenient.

Completing Project Development

Bumping is the activity that marks a project from a status code of **3** (Active) to **4** (Development Complete). A project can not be rolled to any other realm until it is upgraded to status **4**.

When you tell PRC to bump a project, it performs a great deal of investigative work before closing the project. The system examines and "scrubs" the project and may require some decisions be made by the programmer. If the bump succeeds, everything remains in place in the development realm and the project is marked status **4**.

When your project is marked status **4**, you know that it is free of the following ambiguous situations: conflicts, deleted items, and branch copies of items.

- **Conflicts**

Conflicts will always be fatal to the bump process. If there are conflicts on your project you, must go back and "manage" the project (see "[Reviewing & Managing Items in Your Project](#)" on page 45 and "[Scanning for Conflicts](#)" on page 89).

- **Deleted items**

You can resolve deleted items during the bump. PRC will inform you whether the item is really deleted or has been recreated. You can choose to discard the reference to the deleted item or to actually “doom” the item during the rollout. “Dooming” an item means that wherever this project is rolled, including ravel files, the item in question will be deleted from the target account.

- **Branches**

Branch copies can be fatal to the bump if the original item is still checked out on an active project. However, if the project that had the original has been completed and closed out, PRC immediately helps you merge the item as part of the bump. The system presents the original item and your branch copy in the side-by-side editor called Knit, which has functions to assist you with merging your changes into the original item.

When you finish and close the Knit program, PRC asks if you are ready to discard the branch copy. When you answer “yes,” the branch item itself is deleted and the reference to it on the project is modified to the original item. Now the original item is checked out to *your* project. The bump process then continues on and, if no other problems are found, successfully changes the project status to **4**.

No work may be performed against a project that is in any status other than **3**. You and the configuration manager (project leader, release manager, or whoever is responsible for “the big picture”) can be assured that if a project is marked status **4**, it is truly ready to be rolled out and the rollout itself can be performed by a non-technical person.

A state **4** project can no longer be modified by adding or removing items, neither can software changes be made on its behalf. Usually, you can still make minor text changes via the **/QUICK** screen or the main Project Master screen; however, those changes can be locked down at this state, as well, according to status settings. For more information on the main Project Master screen and status settings, refer to the *Administration Guide*.

IMPORTANT

Allowing projects to be changed to status **4** manually—bypassing the bump function—is not recommended.

The bump process from **3** to **4** has been specifically designed to be:

- Used by the programmer who did the work
- Initiated within the development account
- Executed every time the programmer wants the state to change to **4**

It performs *specific tasks* to ensure that everything check, changed, and handled properly to “tidy up” the project up for delivery. Decisions that must be made are presented to the programmer, who is most familiar with the work that was done.

If programmers are allowed to simply change the state without performing the cleaning process, dangerous and unacceptable results may occur.

Once a project is in status **4**, it may languish a while, waiting delivery and testing—but it is clean and sealed. No one beyond the original programmer may come along to move the project—it is ready to roll out.

How to Bump Projects

The “bump” feature handles a combination of possible situations on your project. It checks for conflicts, deleted items (asking if you meant to “doom” them on the destination system), missing items and will warn you if you have a branch copy that needs to be merged and will disallow the merge if the parent is still in conflict. If you can succeed in bumping your project to status **4** you can be certain that it is indeed ready to be rolled out and will roll-out successfully.

1. Launch the Bump process from one of the following three different places within the development account:

- **/SRC** or access **Review Source** from the menu (Figure 5-1) and choose **F7**

The system displays the Review Source Items screen (Figure 5-1). Press **F7** to start the bump process:

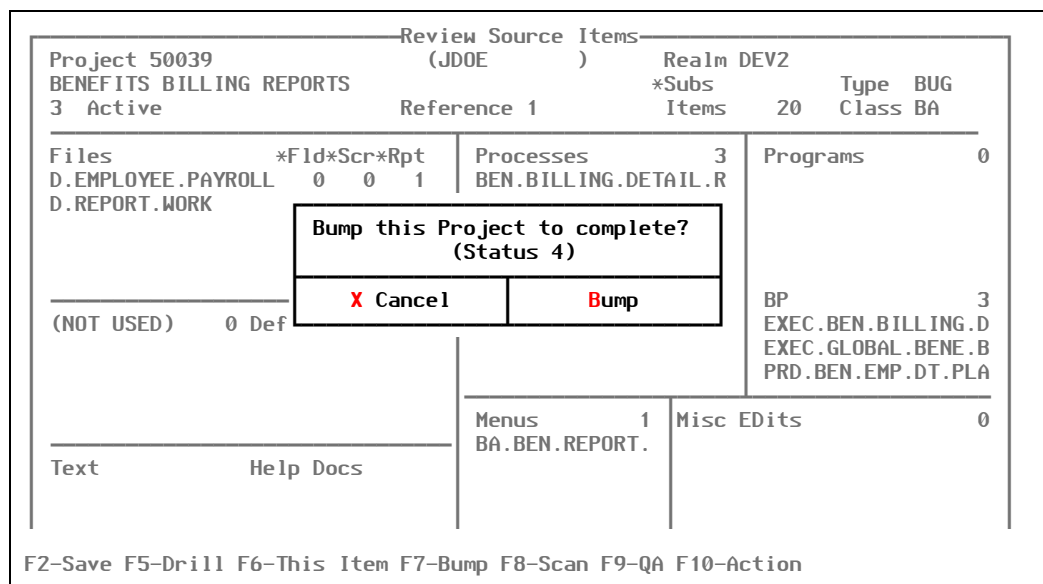


Figure 5-1: Review Source Items screen, showing result of pressing **F7** (Bump)

- **/BUMP**
- **/QUICK** or access **Quick** from the menu

Both **/BUMP** and **/QUICK** displays the Quick Project Update screen (Figure 5-2). Press **F7** to start the bump process:

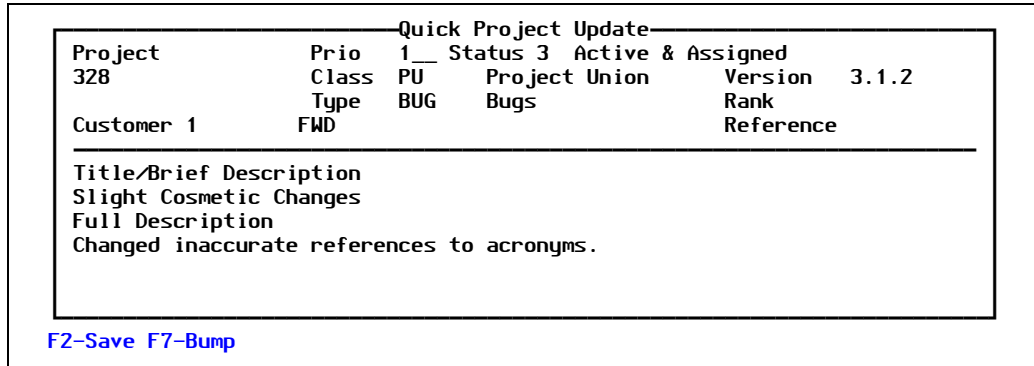


Figure 5-2: Quick Project Update screen, from which you press **F7**

2. The system performs a rigorous diagnostic procedure to ensure that the items on the project are truly complete and ready to roll-out. PRC ensures they are not:

- Missing

If an item is missing, the system displays the message shown in [Figure 5-3](#):

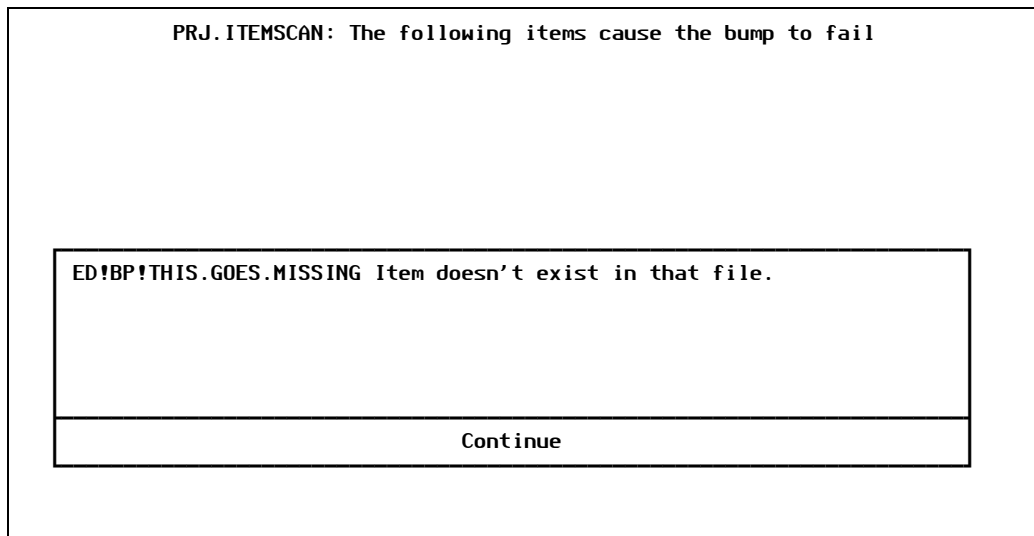


Figure 5-3: Choices for a missing item

- Deleted

If an item is deleted, the system displays the message shown in [Figure 5-4](#):

Review Source Items					
Project 6512	(SJOSLYN)	Realm D			
Sample for bumping.		*Subs	1	Type ENH	
3 Active & Assigned	Reference	Items		Class PRC	
Files	*Fld*Scr*Rpt	Processes	0	Programs	0
BP.UTIL	Item ED!BP!THIS.GOES.MISSING was added to the project, but was not actually changed. (Or it was changed and then changed back.) When compared to ED!BP!THIS.GOES.MISSING\$PRJ*6512\$13798. Remove from project?				0
	Yes, remove	No, carry on	Remove ALL	Keep ALL	1
Text	Help Docs				

F2-Save F5-Drill F6-This Item F7-Bump F8-Scan F9-QA F10-Action

Figure 5-4: Choices for a deleted item

- In conflict

If an item is unchanged, the system displays the message shown in [Figure 5-5](#):

Scanning for items in conflict

Project :PRJ*6504 Scanning against D

11 ED!PRCPROGS!PRC.PRJ.ROLLOUT.NOTES

Item is in DIRECT/ACTIVE conflict with				
Project	Title	Stat	Ver	Assn
PRJ*6508	TEST SWAP	3	6.4w	SJOSLYN

Enter 'N' for No pause, or RETURN to continue and pause at next conflict.

Figure 5-5: Choices for a conflicting item

For more information, see [“Resolving Conflicts”](#) on page 90.

- Unchanged

If an item is unchanged, the system displays the message shown in [Figure 5-6](#):

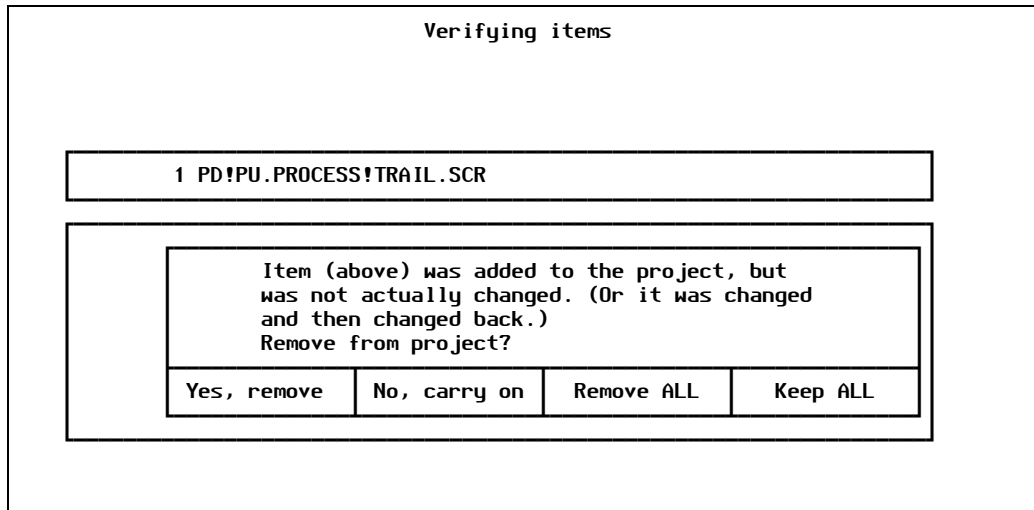


Figure 5-6: Choices for an unchanged item

3. Once the bump process is complete, PRC displays the message in [Figure 5-7](#):

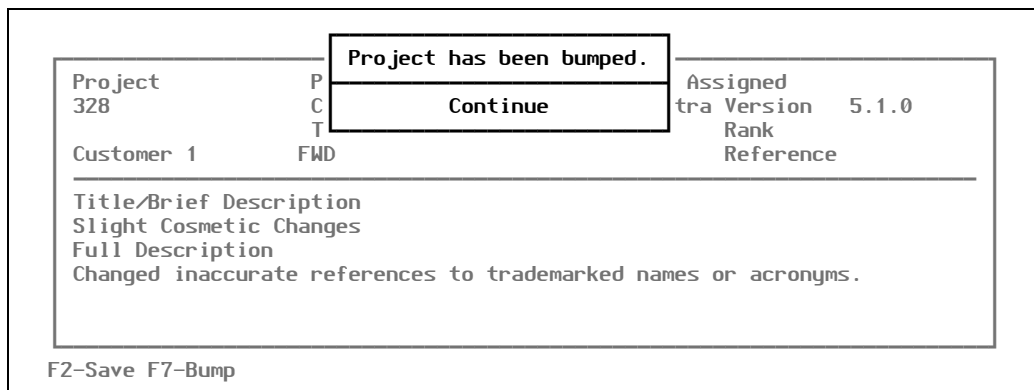


Figure 5-7: Bump completed successfully

Once you have bumped the project, the configuration manager knows that the project is not just marked complete, but is truly in a condition where it can be rolled out.

Bump is normally attached to the status change between **Active** and **Development Complete**, before the rollout to a test area.

Note

After a project has been bumped to status **4**, you can quickly and easily reactivate it simply by using **F7** again and choosing "reactivate."

You can reactivate only from status **4** back to status **3**. If the project progresses past status four there are other considerations, such as whether to rollback, to re-opening/re-activating the project.

Rolling Out Software

Critical Path Step 4

This chapter walks you through the steps for “rolling out” the project with the software changes you have made.

At some PRC installations, programmers are only responsible for getting the project up to status **4** (Development Complete). PRC can be configured to automatically send an e-mail to alert the person responsible for the next status or activity on project that it has reached that status.

When software is ready to be delivered from one place to another the action that is taken is referred to as a rollout. It might be a “promotion” in some environments, and in many cases we are “promoting” a set of software on a project into LIVE. But not all rollouts are really a promotion, so we like to make the distinction.

“Roll out” is a verb. The item that gets rolled out is a project (or in the case of a version, a collection of projects).

- In some cases, that project or version is rolled directly to another account on the same machine.
- In other cases, it is rolled into a *ravel* file, which is then transported (via media, FTP, e-mail, or magic carpet) to another machine and delivered using an *unravel* function.

Either way, no matter what’s going or where it’s going, the act of delivering the software is called “rolling out.”

Considerations Before Rolling Out

You must ask yourself a couple of questions before you perform an actual rollout:

- **What should be rolled out?**

When a project is first opened and assigned to a programmer, it has a description of a discrete solution to be provided, and often a schedule or deadline. This determines what is supposed to be on the project.

When the programmer reaches the end of the active development cycle on the project, the overall project should be re-examined in light of whether:

- It is complete
— and —
- Everything that ended up on it really should be delivered with it

Sometimes this means flicking items off to other projects, or adding items from other projects, or putting the project together with other projects on a version to deliver.

- **How will it be rolled out?**

When you decide *how* to roll out the software, you have to consider *where* the software is to be delivered. Those considerations go hand-in-hand.

What to Roll Out

You must decide on whether the project should be rolled out:

- Individually
— or —
- With other projects, grouped into a set that represents a version or a larger “project”

Combinations of projects are called *versions* in PRC. They can be projects-of-tasks, sub-releases, or any entity in which it makes sense to combine multiple projects. They can be comprised of 2 projects or 200 projects, and can be rolled out at some regular interval—say, Thursdays at 3:00 PM.

How & Where to Roll Out

You must decide on how the project should be rolled out:

- By rolling the software to another account *on the same machine*
(or across the network but addressing the files as if they were local)
— or —
- By bundling everything into a single file for delivery *to a remote site or machine*
(delivered via telecommunication or media)

These decisions are illustrated in [Figure 6-1](#):

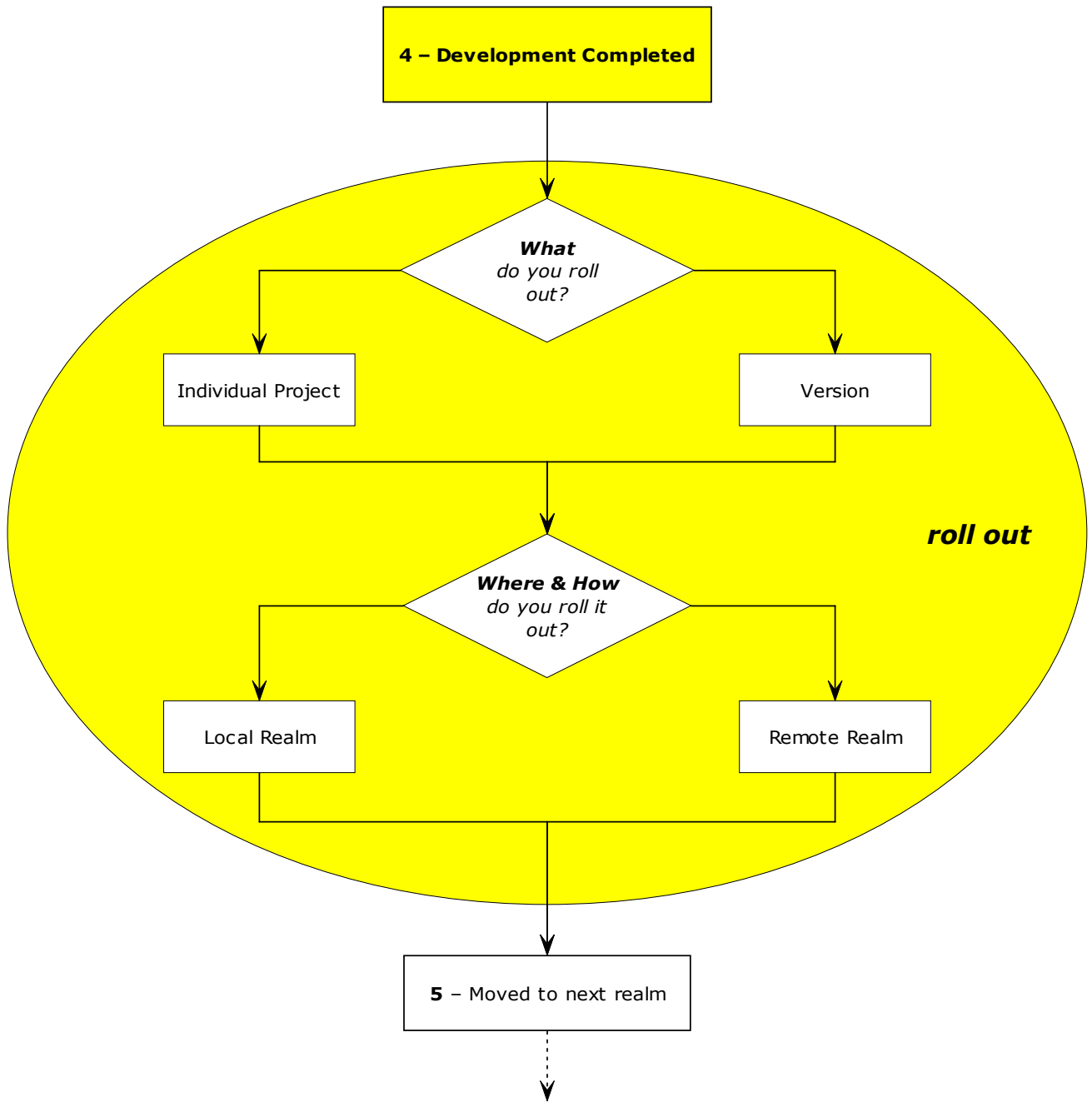


Figure 6-1: Decisions to make for rolling out

How to Roll Out From the PRC Account

Summary

The overall steps for a rollout are:

1. Log in to the **PRC** account.
All rollouts are performed from that account, not the development account.
2. Decide which screen to use to launch the rollout:
 - The Project Master screen (in the case of individual project rollouts)
 - The Version Master Record screen (in the case of version rollouts)
3. Choose which realm to send the rollout:
 - A local realm (in the case of delivering software within the same machine)
 - A remote realm (in the case of creating ravel files for a different machine)
4. Launch the rollout.
Resolve any conflicts, if the system finds some while it prepares the roll out.

There are more fields on the main project screen in the PRC account than on the Quick Project Update screen that you may have used from **/QUICK** to create a new project. Your organization may decide on some additional information that you will add to the project from there.

Reminder

At this point, all of the work to ensure the project is ready to roll should be finished. Nevertheless, at this point, flicking and pre-scanning for conflicts are still available from the **[F8]** version of the Source screen from the main Project Master screen. The items on the source screen that deal directly with development (“This Item” options, tracing, knitting, and so on) are *not* available from this screen. You will notice that the function key descriptions are different.

Flicking—or any adding and removing—is only available while the project is in status **3**.

Detailed Steps for Project Rollouts

To roll out work that has been accomplished, follow the steps laid out in this section.

Log In to PRC Account

1. Type the slash (/) character.
2. Type **PRC** in the command line.

If you successfully log into the PRC account, the system displays the PRC Administration menu, as shown in [Figure 6-2](#):

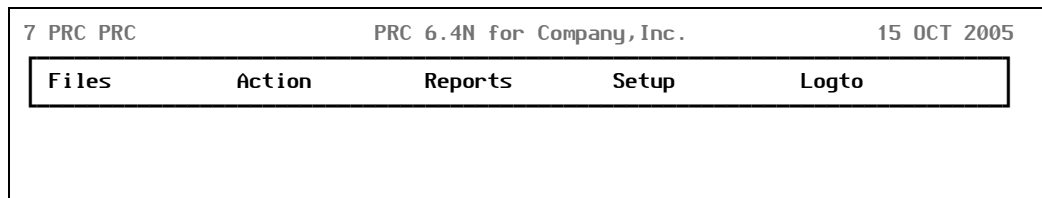


Figure 6-2: PRC Administration menu

Proceed to the next section.

Select Individual Project or Version Rollout

3. Select the appropriate master screen from the **Files** menu.

The menu option you choose depends on what you are rolling out. If you are:

- Rolling out a single project, go to [“Individual Project” on page 71](#).
- Rolling out a collection of projects (version), go to [“Version” on page 73](#).

Individual Project

- a. Select **Files > PRC Project Master**.

The system displays a blank PRC Project Maintenance screen.

- b. Enter the project ID that you wish to roll out in the *Project* field or press **F3** to select from the list of valid projects for this account and realm.

The system redisplay the PRC Project Maintenance screen, similar to the one shown in [Figure 6-3](#):

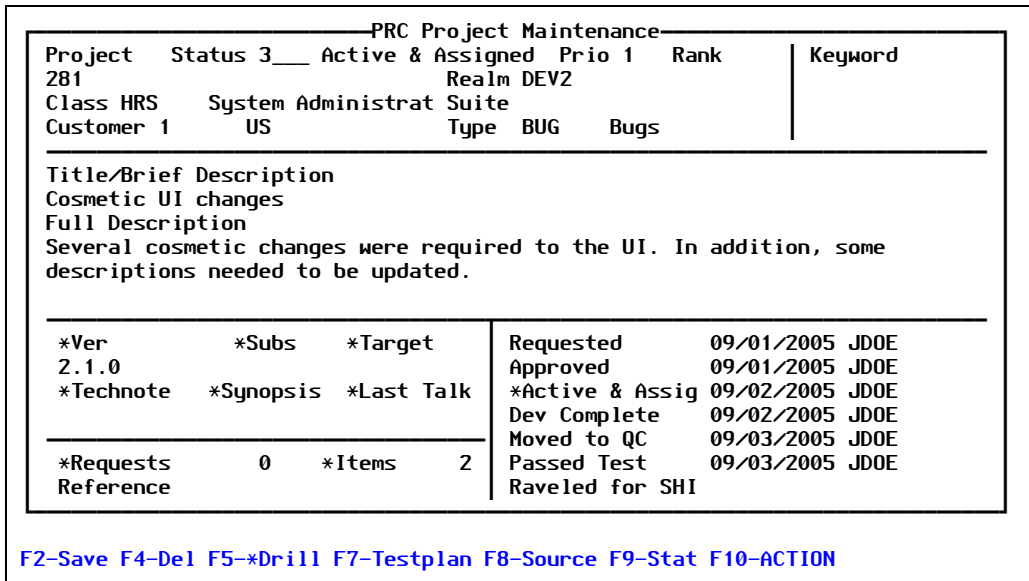


Figure 6-3: Project Maintenance screen, showing a project

- c. Press **F8** (Source) to review the source items that are on the program.
The system rearranges the PRC Project Maintenance screen to show the source items, similar to the one shown in [Figure 6-4](#):

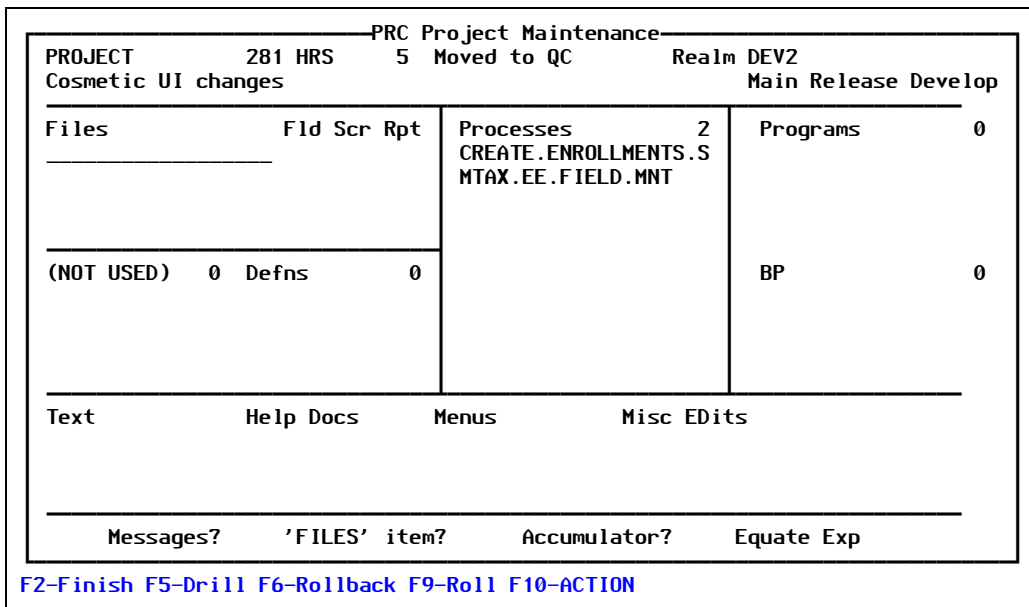


Figure 6-4: Showing the source items

As you can see, in order to roll out a project, you must go through both the main project screen—providing an opportunity to review the description and other primary information—and the **F8** (Source) screen, for a last review of the software

components about to be delivered. Most users agree that it is a good idea to take a look at the project *and* the components that are on it before actually launching those components into another realm.

- d. Press **[F9]** (Roll) to roll the project out.
Skip to “[Select the Destination Realm](#)” on page 74.

Version

A “version” in PRC is really just a collection of projects that are to be kept and/or delivered together.

Note

A project may be contained in more than one version.

- a. Select **Files > Version/Sub-Version**.
The system displays a blank Version Master Record screen.
- b. Enter the version ID that you wish to roll out in the *Version* field or press **[F3]** to select a system, and then select from a list of valid versions for that system.
The system redisplay the Version Master Record screen, similar to the one shown in [Figure 6-5](#):

Version Master Record				
Version	MAR19	System	PRC	
Description	Final testing stages, March 19			
Source Realm	DEV2	No. Sub-Releases	0	
Status	5 Moved to QC			
*Project	3	Title	Assigned to	Status
PRJ*14		Rank, file-set and other miscellany	SJOSLYN	5
PRJ*15		Misc edit items that have an object fi	SJOSLYN	5
PRJ*19		Upload (smarten for non-ravel) (PSI)	SJOSLYN	3

F2-Save F4-Del F5-*Drill F6-Sched F7-Sort F8-Pre-Scan F9-Roll-Out F10-Action

Figure 6-5: Version Master Record screen

- c. Press **[F8]** (Pre-scan).
This feature has a check-box for “report only.” This is useful for large versions which contain many projects and many items, so that all problems and conflicts can be reported in one consolidated report before attempting the rollout itself.

The conflict scanning and actual rolling out are performed on a project-by-project basis. The whole version is rolled out—but each project is rolled one after another. This way, individual projects can still be rolled back.

Note

Projects within the same version will not conflict with each other, even if they do contain conflicting items. For this reason, combining two or more projects into a version and delivering them together can be an option for dealing with conflicts at delivery time.

- d. Press **F9** (Rollout) to roll the version out.

Proceed to the next section.

Select the Destination Realm

You perform the same steps when you roll out software, whether you select a target realm that is “online,” meaning it is on the same machine, or “remote,” meaning it is on a different machine. PRC will automatically handle things differently, depending on the target realm.

Realms are designated as online or remote in the realm definition itself. For more on realm definitions, see the *Administration Guide*.

Note About Security Clearance

Sometimes people ask why it is necessary to log to the PRC account to perform a rollout. The primary reason is security. In order to comply with the regulations of ISO 9003, *et al*, the rollout must be secure.

In addition, if rollouts were performed from the development area, the results would include pointers straight into the live area from the development area. Such an arrangement would fail compliance and is a poor idea if control of the live environment is desired.

No matter *what* is being delivered and *how* it will be delivered, the rollout itself must be performed from the PRC account. So in order to perform a rollout, you must have the clearance settings to log into the PRC account and then the appropriate settings to roll the particular project (or set of projects) to the particular destination.

In the wake of new compliance initiatives such as Sarbanes Oxley, decisions about security clearances are dictated by a combination of factors. In particular, the “segregation of duties” requirement is addressed in PRC: you have the ability to restrict a rollout to a group of individuals; then, you can further restrict it so that the individual responsible for a certain “role” on the project (such as programming, testing, or approving) cannot be the particular individual who rolls out that particular project(s) to a particular realm.

You can abandon the roll-out after the conflict scan, if you would like.

If you are rolling out a single project, the Project Rollout subscreen will look similar to the one shown in Figure 6-6:

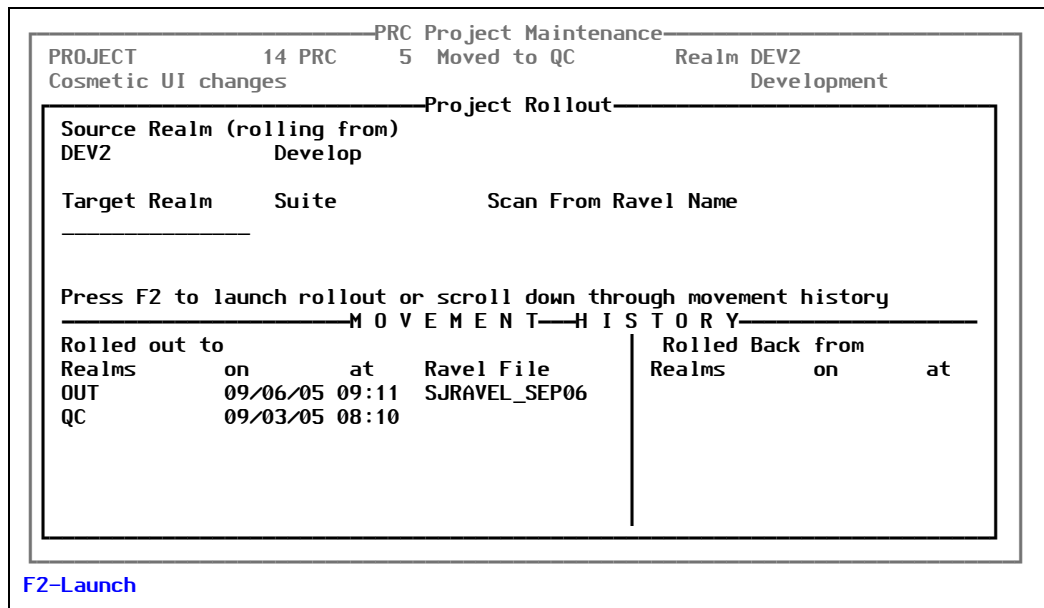


Figure 6-6: Project Rollout screen

4. Select the realm to which you want to roll out the project in the *Target Realm* field. The system normally displays a default realm based on the settings on the Realm screen and according to the current status of the project.

Press **[F3]** to select from the list of valid realms.

If you try to roll out a project that is in the wrong status for this rollout (according to the realm settings), the system will display a warning message at the bottom of the screen. If you receive that message, you will not be able to proceed with this rollout.

If the “scan” for that realm has been set to “Historical” (in a drill-down option on the Realm screen), the system prompts you for an “as of” date.

5. Enter an “as of” date.

This allows scanning for conflicts with projects since that date, whether they are now closed or still active. The default “as of” date will be the birthdate of the realm if one has been entered on the Realm Master maintenance screen. An active scan only checks for conflicts against other active projects.

When would you use an historical scan?

- When one programmer takes a save and does work on a different system. When its time to roll his changes in, it is not enough to know whether his changes are in conflict with other projects open at that moment, but must be checked against everything since the date of his save, even if those projects are closed and moved on to test.
 - When new software changes from a vendor using PRC are delivered to an end-user customer using PRC where site changes have been made.
6. The conflict analysis will be run first.

- If there are conflicts, they will be displayed. They may be optionally printed and the rollout will halt.
- If there are no conflicts, the screen will display “No Conflict Clear to Roll.” Please remember that this means only that there are no conflicts.

There are occasionally other errors that will prevent the roll-out (such as missing files) which are encountered next and displayed.

Note

Conflict and other scanning is all done in a pre-roll-out mode for two reasons:

- No partial rolls
- Schedule

During a scheduled rollout, all of the conflict scanning and other integrity checks are performed when the project is scheduled to roll out, then the rollout itself may be run later.

For more information, see [“Scanning for Conflicts” on page 89](#).

Launch the Rollout

7. Press **F2** (Launch) to launch the rollout to the other realm.

When all integrity checks have been passed, PRC will move the items one-by-one to the destination realm. PRC knows about all of the “internal” SB+ items, like the dictionary section with the dot (.) prefix, the **.TXT** section of screens and other items. There are screens to configure these kinds of things for your development environment during the initial setup if the development environment is not SB+. SB+ is pre-configured and no set-up is required.

While PRC is delivering each component, it displays a count and item names on the screen. When everything has successfully delivered, the system displays a “No Errors” dialog box.

There are some errors that are non-fatal, such as an item that was on the project did not exist in the source (development) realm. These messages may be displayed if they occur, by typing **D** (Display).

8. If such errors are reported, you may do one of three choices:
 - a. Select **F6** (Rollback) to “roll back” a previously rolled out project.
 - b. Type **E** (Exit) and leave it delivered in spite of the errors.
 - c. Type **P** (Print).
9. If you are rolling out to a ravel file, the system prompts you for a name for the ravel file.

How to Roll In (Extract)

There are a number of ways that software can be received from PRC. These functions are covered thoroughly in the *Software Receiving & Auditing Guide*. In general, they are:

- Source realm and Destination realm are on the same machine.
There is no separate receiving step required in this scenario. The rollout installs the software exactly where it needs to be. In some cases, according to settings, it may be necessary for a developer user to log in to the destination account to kick off some activities (compiles, catalogs, indexes, and so on).
- Source realm and Destination realm are on two different machines; both machines have PRC installed.
There is a menu available in the destination account called **PRCCUST** that provides tools to find, view, deliver and undeliver ravel files into that account.
- Source realm and Destination realm are on two different machines; the receiving machine does not have PRC installed.

At the simplest level, there is a brainless extraction program that can be used to simply deliver stuff out of a ravel file into a particular account on another machine.

The “extraction” or “rollin” at the destination site may be performed in four different ways, as shown by the four menu options (**PRC > Action > Ravel Files**) in [Figure 6-7](#):

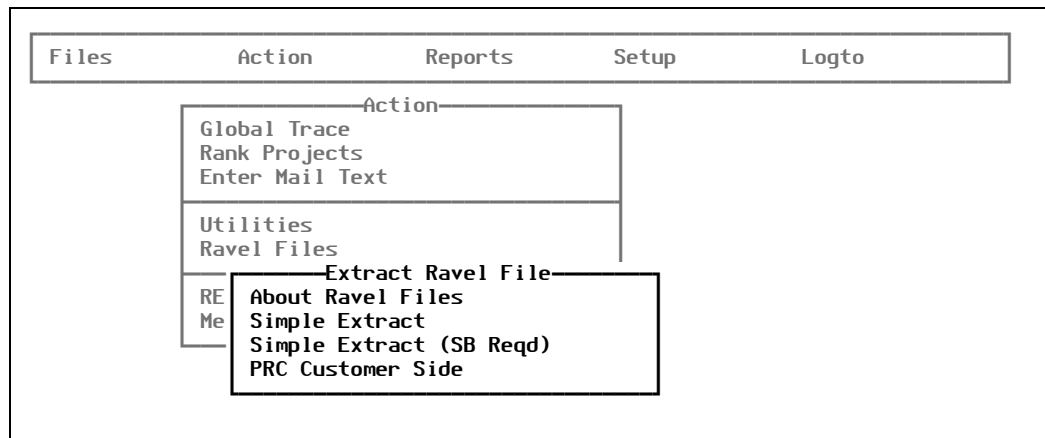


Figure 6-7: Extract Ravel File menu

The first menu option, **About Ravel Files**, is summary help text.

The next two options, **Simple Extract** and **Simple Extract (SB Reqd)**, are the simplest form of extraction. They rely solely on the control item that is placed into the ravel file on the system where the ravel file was created.

- The **Simple Extract** menu option is intended for ravel file delivery to non-PRC sites—remote sites where PRC is not installed.

- The **Simple Extract (SB Req'd)** menu option is intended for ravel file delivery to PRC sites—remote sites on a system where PRC is installed and configured.
There can be other settings that will override some of the business that will occur during the ravel. These settings are also covered in the *Administration Guide*.
- The last menu option, **PRC Customer Side**, creates a ravel file that uses a version of the unravel program that is much more powerful and configurable.

How the extraction works at the remote site is actually programmed during the rollout, based on the field setting in the Extraction Options sub-screen (select **PRC > Setup > Preferences > [F6] (Addl Opts) > Ravel/Extractions**). On that sub-screen, you can define the format for the default ravel file names, whether or not the drivers file should be cleared, and whether programs should be regenerated and/or compiled.

For more detail about ravel files and settings, see the *Administration Guide*.

If your organization is regularly delivering ravel files to a non-PRC environment, your company has probably built its own customer-side delivery mechanism.

How to Roll Back

Rollbacks are a way to recover the realm into which the project was rolled. They are a fast, easy, and reliable way to “un-deliver” a project, version, or ravel file.

No software is delivered from one realm to another during a rollback. The realm that the project was delivered into is set back to its condition before that rollout. This is similar to a “transaction rollback” in data processing.

Rollback Steps

The rollback is also accessed through the **[F8] (Source)** screen from the main project screen, by pressing **[F6] (Rollback)**.

1. Select **Files > PRC Project Master**.
The system displays a blank PRC Project Maintenance screen.
2. Enter the project ID that you wish to roll back in the *Project* field.
The system redisplay the PRC Project Maintenance screen, similar to the one shown in [Figure 6-3 on page 72](#).
3. Press **[F8] (Source)** to review the source items that are on the program.
The system rearranges the PRC Project Maintenance screen to show the source items, similar to the one shown in [Figure 6-4](#):

PRC Project Maintenance									
PROJECT	281 HRS	5	Moved to QC	Realm	DEV2				
Cosmetic UI changes					Main	Release	Develop		
Files	Fld Scr Rpt			Processes	2	Programs	0		
				CREATE.ENROLLMENTS.S					
				MTAX.EE.FIELD.MNT					
(NOT USED)	0	Defns	0			BP	0		
Text	Help Docs		Menus	Misc EDits					
Messages?		'FILES' item?	Accumulator?	Equate Exp					

F2-Finish F5-Drill F6-Rollback F9-Roll F10-ACTION

Figure 6-8: Showing the source items

In similar fashion to rolling out a project, you must go through both the main project screen followed by the **[F8]** (Source) screen.

4. Press **[F6]** (Rollback) to roll back the project.

Conflict Considerations

Whenever you perform an un-unravel process or rollback, the system performs the same conflict scanning as was set to occur during the initial rollout or unravel.

- When you un-unravel in an account, the conflicts will be against other software that may since have been rolled into this account. When conflicts exist, it is likely that you will need to un-unravel each thing that has been unraveled since the item you are attempting to un-unravel—like peeling back the layers of an onion.
- When you roll back in an account, the situation might be simpler. The conflicts for which the systems warns you are in the development area. All the rollback is really doing is setting the destination account back to its previous condition.

Since nothing is actually “rolled into development” (think of it not as a backward rollout, but more of a “transaction rollback,” specifically on the destination realm), there is no harm in carrying on with the rollback despite conflict warnings. The trouble will come when you are ready to re-deliver the items from DEV, because when you rolled back, you re-checked-out the software items on the project being rolled out. Now those items will be in conflict.

Remember that a rollback and an un-unravel are about getting the changes out of the destination account.

Using Tracking Management

A few less-commonly-used features are available from the working development accounts by choosing **Management** from the PRC menu (Figure 1-3 on page 21).

The system displays the Tracking Management sub-menu:

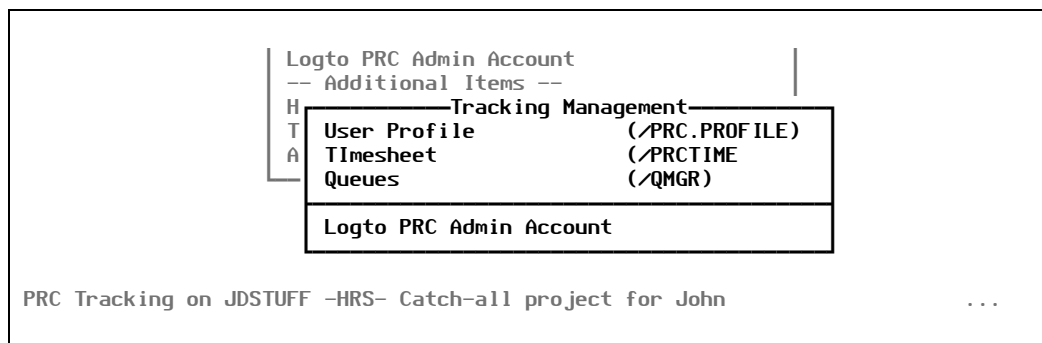


Figure 7-1: Tracking Management menu

This chapter explains the options in the Tracking Management sub-menu.

User Profiles

This option is a miniature version of the PRC user profile defined by the system administrator. For information on the main PRC user profile settings, refer to the *User Profiles* tip sheet (or use **F1** help).

You cannot modify sensitive configuration and security settings from here, but certain personal options *can* be modified from here to suit your own preferences.

To work with user profiles, perform the following steps:

1. Select **User Profile** or type **/PRC.PROFILE**.

The system displays the PRC User Profile screen, similar to the one shown in Figure 7-2:

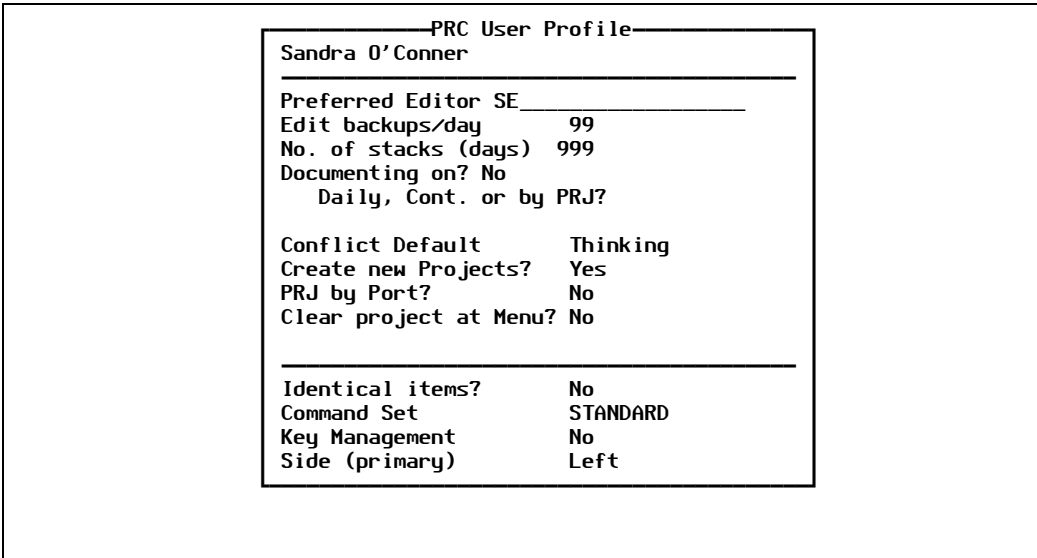


Figure 7-2: PRC User Profile screen

Explanations of the fields are as follows:

- Preferred Editor** The editor called by the system when PRC needs one. Other editors are still available, but this is the default under those special circumstances.
- Edit backups/day** The number of backup copies made per day for each item (the first one of the day is not included). These copies are only kept during the active life of the project.
- No. of stacks (days)** A reverse chronological stack of changed items per day per user, regardless of assigned project. The value of this field determines the number of days back the stack goes.
- Documenting on?** Indicates whether PRC requires comments to be recorded each time an item is filed by using **F2**. Valid entries are **Y** (Yes, comments are required) and **N** (No documentation needed).

- Conflict Default** If the user's security clearance is high enough to allow conflicts to be created, this field determines which will be the most common (default) choice. This default will be the first option in the dialog box whenever it needs to come up. Valid entries are:
- **Overwriting**
Creates a conflict.
 - **Copying**
Creates a branch copy.
 - **Thinking**
Prevents the programmer from accidentally creating a conflict or a copy when they hit return an extra time before realizing what was being prompted. Recommended as the best choice.
The "thinking" option simply returns to the dialog box. The programmer must enter either the hot letter O (Overwrite) or C (Copy).
- PRJ by Port?** Tells PRC to keep each login session attached to its own project. Therefore, the same user ID can be logged in multiple times and tracking against a different project in each session.
- Clear project at Menu?** If the value of this field is **Yes**, PRC will never assume what project you are working on when you first log into the system. It will always ask you for the project assignment.

The remaining fields in the lower section of the screen are the default values for the Knit feature. For more information, see the *Software Knitting Guide*.

Timesheets

You can use timesheets to track time spent on projects. "Activity Codes" are set up in advance (in the PRC account, see the *Administration Guide*) and then time can be posted by activity into the projects. This is not meant to be a full time-tracking or payroll system, but can be a handy way to get a feel for actual hours spent vs. estimates. If the activity codes are designed cleverly, timesheets can provide interesting metrics about timing within the life-cycle.

When you select **Timesheet** or type **/PRCTIME**, the system displays the Daily Timesheet screen, similar to the one shown in [Figure 7-3](#):

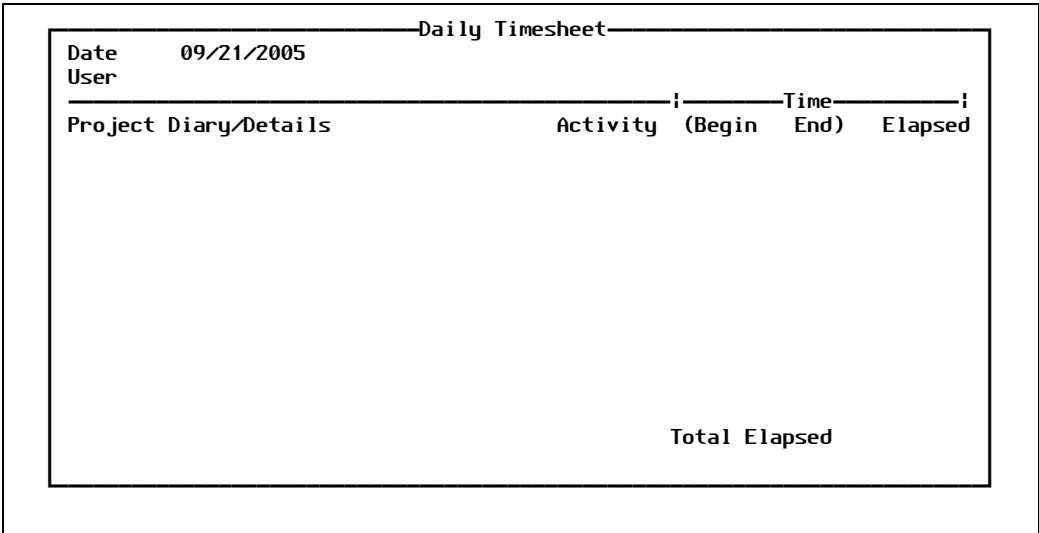


Figure 7-3: Daily Timesheet screen

Explanations of the fields are as follows:

- Date** The date for this timesheet. The system uses the current date as the default.

- User** The PRC developer to whom this timesheet belongs. The system uses the current user ID as the default.

- Project** The ID of a project against which time is posted.

Press **[F3]** for a list of valid project IDs.

- Diary/Details** [*Optional*] The details about work done on this project/activity on this date.

- Activity** The code or ID that represents the activity done for this project on this date. Activity type codes delivered with PRC are:
 - **SPEC** — Specifications
 - **DOC** — Documentation
 - **CODE** — Coding
 - **TEST** — Testing

Activity types are defined in the Code Files menu in the PRC account. You can use the types listed above, add on to them, clear them, and create your own.

Elapsed Time	<p>The total amount of elapsed time that you wish to charge to this activity line on this date by this user.</p> <p>In the default configuration, this field is required.</p>
Begin Time, End Time	<p>[<i>Optional</i>] If these fields are activated, the above <i>Elapsed Time</i> field is not an input field, Instead, it will be the calculated elapsed time between the beginning and ending times entered in these two fields.</p> <p>If the “timer” field is on in the PRC user profile, then as the developer switches between projects, those beginning and ending times are marked. During timesheet entry, you can press F3 to drag those recorded details for a day onto the timesheet.</p> <p>For more information on PRC user profiles, see the <i>User Profiles</i> tips sheet.</p>
Total Elapsed	<p>Calculated total elapsed time for the day, this is a vertical total of the elapsed time field (whether that field is entered or calculated).</p>

Queues

A new feature in PRC Release 6 is the ability to “queue” an item that is already checked out. The feature is available to users who have the appropriate clearance settings. For the sake of simplicity, assume the user does have the appropriate security clearance.

In PRC, a queue is a list of projects that want to check out an item. A queue is created when an item is checked out and a request is made to check it out to a different project. PRC gives the developer who wants the already-checked-out item the choice of whether they want to “wait in line” for the item.

For example, Dick wants to edit an item. Little does he realize, Jane checked it out an hour ago to *her* current project. When Dick tries to open it (which would normally check out the item to his project automatically), the system informs him that it is already checked out and prompts Dick about whether he wants to queue his project against the item or not.

Then you can review items and the projects in queue for them via the Manage Queue Items screen.

When you select **Queues** or type **/QMGR**, the system displays the Manage Queue Items screen, similar to the one shown in [Figure 7-4](#):

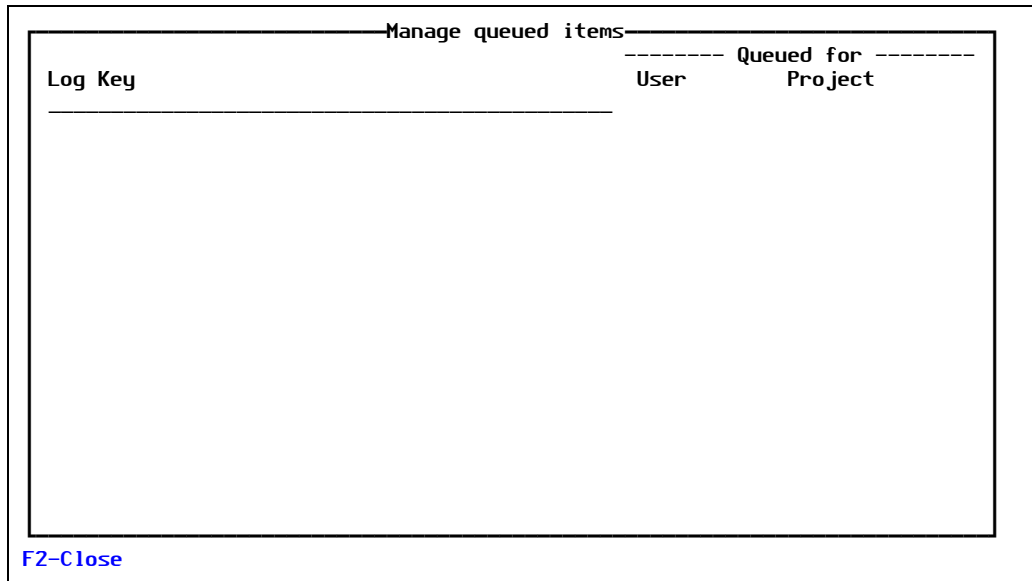


Figure 7-4: Manage Queued Items screen

You can view items and the projects that are in queue to use them in this screen. You can use this screen to add projects to or remove projects from an item’s queue, as well.

Explanations of the fields are as follows:

Log Key The unique identifier that is used to track an item’s checked-out status and history. The format is a type code (often **ED**) followed by the file name and item name, separated by exclamation marks (!), known as “bangs.”

For example, ED!BP!PROGRAM.NAME.

Queued for When the item becomes available—is released by the project and user that originally had it checked out—it will automatically be checked out to the next user/project combination on the queue.

When it becomes available, the user who was queued for it is notified via e-mail, if the PRC is set up to do so (see the *Administration Guide*).

User The user who has queued for this item when it becomes available.

Project The project ID on which the user queued for the item.

2. Enter the appropriate values in the fields, as necessary.
3. Press **F2** (Close).

Using Non-Critical Tools

Several useful utilities that are not always on the critical path for programmers are available via the Review Source Items screen and the PRC menu.

- From the Review Source Items screen, you can press:
 - **F8** (Scan) to scan for conflicts
 - **F9** (QA) to use test plans
- From the Review Source Items screen, press **F6** (This Item) and select:
 - **Trace**
 - **Compare**
- From the Review Source Items screen, press **F10** (Action), which produces the Action bar from which you can select:
 - Print
 - Technical Notes
 - Table of Contents
 - Other
- From the PRC menu:
 - **Search**
 - **Stack**
 - Manage backups
 - **Knit**

Scanning for Conflicts

How to Scan for Conflicts

A “conflict” is created when two or more projects modify the same item (screen, field, process, or program).

To check your project to see if the source items are in conflict with any other project, follow these steps:

1. Either press **[R]** for **Review Source Items** from the PRC menu (Figure 1-3 on page 21) or type the **/SRC** command from anywhere in the system.

The system displays the Review Source Items screen.

2. Press **[F8]** (Scan).

The system performs a conflict analysis similar to the one that is performed as part of the rollout process. This analysis will ask for the destination realm so that it can locate the particular rules for the rollout (where conflicts should be sought, and so on). It will then perform the analysis and inform the user of any conflicts.

If you press **[F8]** and no conflicts exist, the system displays the dialog box shown in Figure 8-1:

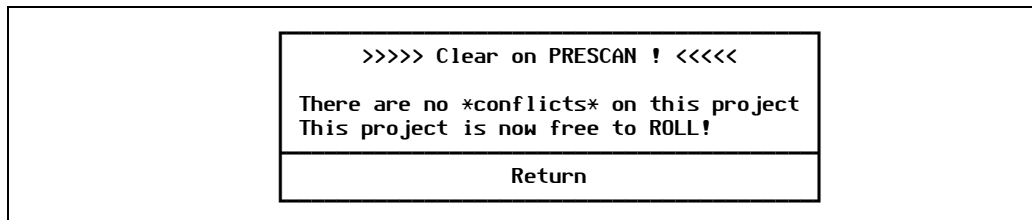


Figure 8-1: Result of pressing **[F8]** (Scan)

Otherwise, the system displays all source items that have a conflict. To determine what other project(s) affect each item, see “Tracing an Item’s History” on page 92.

Resolving Conflicts

PRC may warn you of a conflict: an item that is already checked out to another project. Unfortunately, there are no “pat” or standard answers to resolving conflicts. They are as unique as snowflakes.

What you will do to resolve a conflict will depend on your clearance settings, corporate culture, and the unique situation each time. If you use the Knit utility to resolve conflicts, see the *Software Knitting Guide* for more information.

It is difficult to determine a procedure for resolving a conflict because each one is unique. However, some recommendations may help:

- Whenever possible, keep programmers and projects isolated in areas of the system, minimizing crossover items.
This is the most pro-active, from-the-top, prevention-oriented procedure.
- Allow projects to conflict and then bundle them together on a version when its time to roll out.
Items modified by more than one project that are rolled out together in a version will not conflict with each other.
- Consider using sub-projects when the situation warrants.
Sub-projects can be set (by a preference) to *not* conflict with one another.

Another option can force sub-projects to be “clustered,” meaning they *must* be rolled together. Each sub-project will be named with a suffix. For example, project 100 may have sub-projects 100.1 and 100.2 for specific portions.

- Sometimes it is best to have one programmer remove their changes and the item from their project long enough for the conflicting project to be rolled out.
- When all else fails, the realm can be defined as allowing conflicting items to be rolled out anyway.

WARNING: This is an incredibly bad idea; but—in rare circumstances—it does seem necessary.

Setting Up the Project for QA

First, the use of test plans in PRC is optional. If you do not want to use test plans, you can establish a life-cycle that allows a status for testing and a status for marking a project as “passed testing”—but you make those state-changes manually.

However, if you *are* going to use test plans in PRC, you can create them in advance or “on the fly.” You can build them as a hierarchy, where one line of one test plan represents another entire test plan.

To prepare the project for quality testing, press **[F9]** (QA) from the Review Source Items screen (Figure 4-1 on page 47). The system displays the Testplan subscreen, similar to the one shown in Figure 8-2:

Test Plan Id		Script	No. Iter	Last Rslt
PRJ*281				0
Test Step ID	Script path			
*Descr				
*Expect				
Test Step ID				
Descr				
Expect				
Test Step ID				
Descr				
Expect				

F2-Save F5-*Drill F6-Iter F7-Bump F8-Source F9-Pull

Figure 8-2: The Testplan subscreen, the results of pressing **[F9]** (QA)

The Testplan subscreen, available from the working development and testing accounts, is where you can attach a test plan to a particular project.

A quality tester accesses it for noting iterations of testing on individual steps, marking them as passed or failed. For detailed information about creating and using test plans, testing steps, iterations of test steps, and “QA bumping” a project, see the *Software Quality & Testing Guide*.

Tracing an Item’s History

To “trace the history of an item” means to find out to what PRC projects an item is currently checked out, as well as on what projects it has ever been modified or against which it has been checked out. There are several ways to trace an item’s history.

Each time a source item is checked out (or modified), it is stamped in a central repository called `XXX.REVLOG` (`SYSID.REVLOG`).

Whenever items are checked in, the summary stamp is moved to the `XXX.REVHIST` file and the `REVLOG` stamps are removed. An item is checked in by one of these two methods:

- Defining the project status as **C** (closed and checked in). For more information, see the *Administration Guide*.
- Performing a rollout to a realm that has its *Release Tracking* field (on the Realm Master screen) set to **Y** (Yes). For more information, see the *Administration Guide*.

To trace an item’s revision history the most common way, perform the following steps:

1. From the Review Source Items screen (Figure 4-1 on page 47), place the cursor on an item and press **F6** (This Item).
2. From the dialog the system displays (Figure 4-4 on page 54), select **Trace**.

When you select **Trace**, the system displays the Revision Tracing screen, similar to the one shown in Figure 8-3:

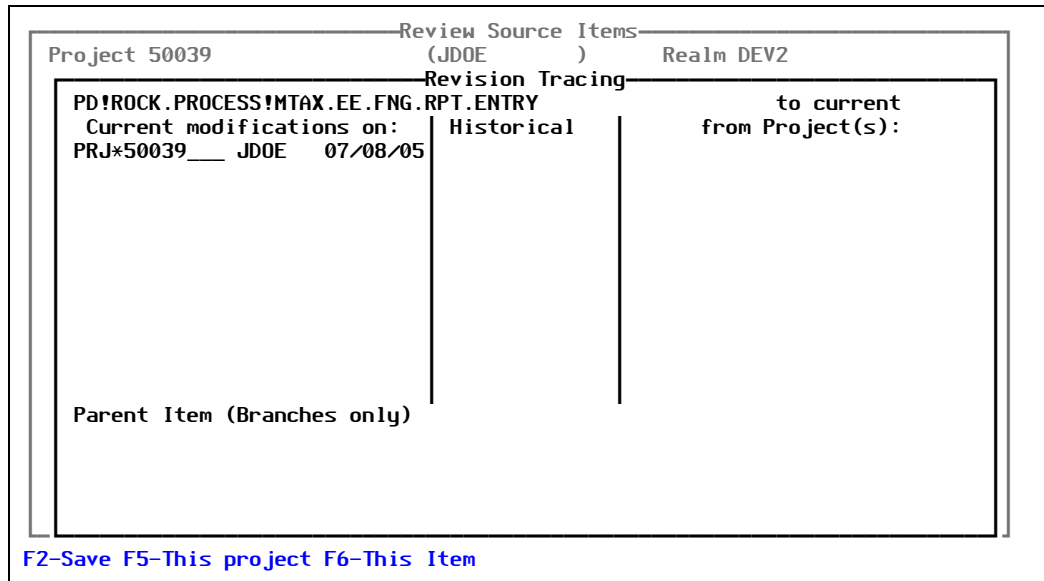


Figure 8-3: Revision Tracing screen

The screen shows all active projects against which the item is currently modified and the history of all projects against which this item has ever been modified.

More than one project in the active column indicates that the item is in conflict.

Explanations of the fields is as follows:

- | | |
|-------------------|---|
| Trace Prj | Here are listed any projects which this item is currently being revised against. More than one project in this column indicates that the item is in "conflict." |
| Trace Dt | Indicates the date of the item version. |
| Trace Hist | Any closed (checked-in) project which this item has ever been revised against will be listed in column. |

3. Press **F2** (Save).

You can also trace an item's history:

- From the PRC Log Stack screen (**/STACK**) screen, place the cursor on any item, press **F6** (This Item), then select the **Trace** option
- From within any tool or item, type **/TRACE** at any time

Comparing Items

There are a variety of ways to perform comparisons.

Creating a Compare Project

Use this feature as one way to compare one realm to another realm, creating projects automatically from the differences between the two realms.

To set up this feature, perform the following steps:

1. Log into the PRC account.
2. Select **Action > Utilities > Create Compare Project**.

The system displays the Compare a System in Two Accounts screen, similar to the one shown in [Figure 8-4](#):

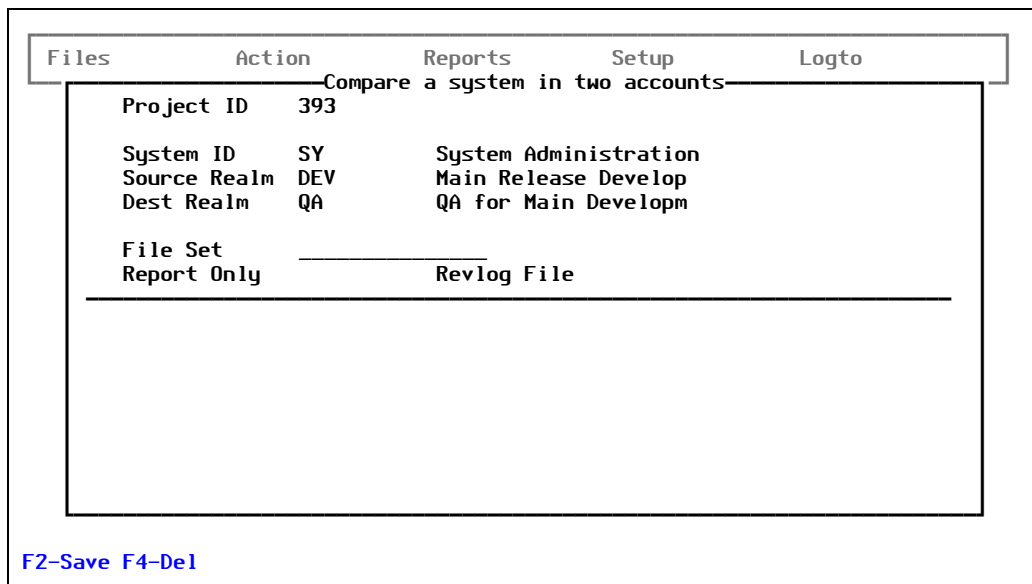


Figure 8-4: Compare a System in Two Accounts screen

Using the Knit Utility

You can also compare any two items at any time by typing **/KNIT** and entering the file name and item name for two distinct items. Use the Knit utility can be a necessary way to examine items and individually compare them, even if a larger-scale routine is used to identify what items are different.

For more information, see the *Software Knitting Guide*.

You can perform the following steps:

1. From the Review Source Items (/SRC) screen, place the cursor on the item of interest.
2. Press **[F6]** (This Item).
3. Select **Compare**.

The system inquires as to whether you want to compare the item to its backup entries or to another file and item, as shown in [Figure 8-5](#):

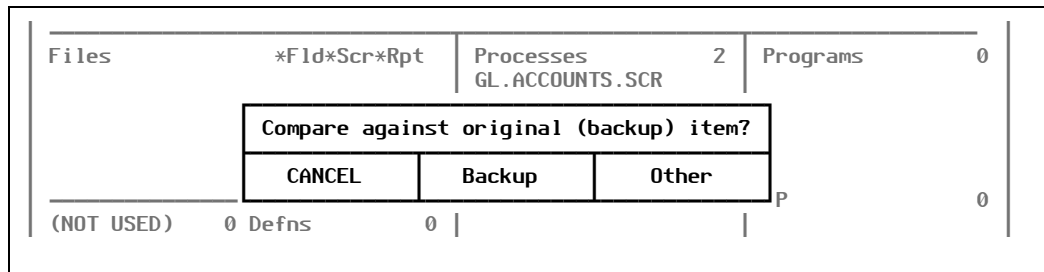


Figure 8-5: Compare to Backup or Other question

4. Select either **Backup** or **Other**:
 - If you choose **Backup**, the system displays a selection box of all backups made of this item during the course of your project, similar to the one shown in [Figure 8-6](#):

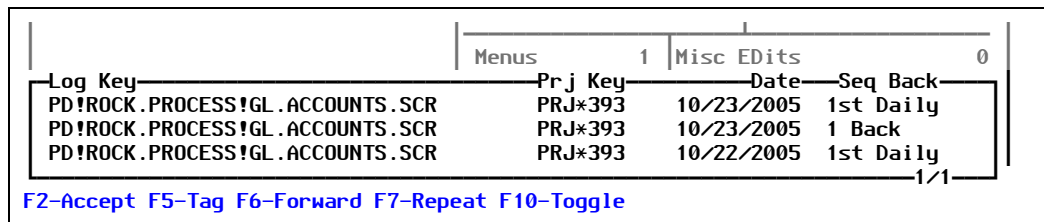


Figure 8-6: List of backups

You can choose from other days, such as a day when you know you went off on a tangent. However, if you wish to check and see what *all* of the changes are that you actually made to this item since beginning the project, simply choose the first one in the selection box, which will be the oldest backup—made the first time you modified this item for this project.

After you select the backup to which you want to compare, the system displays the Knit utility. For more information, see [“Managing Backups” on page 103](#).

- If you choose **Other**, the system goes straight to the Knit utility with the item on which you pressed **[F6]** as the first or primary item. The system requests which item should be compared to the primary item.

An example is shown in [Figure 8-7](#):

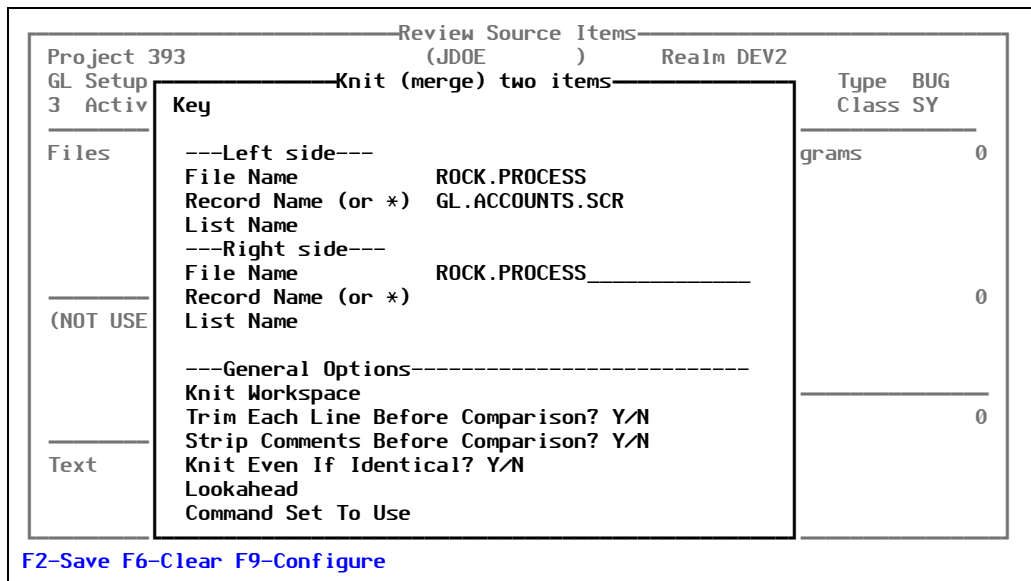


Figure 8-7: Knit utility to compare to another other item

Using the Review Source Items Screen Action Bar

If you press **[F10]** (Action) from the Review Source Items screen (Figure 4-1 on page 47), the system opens an Action bar or menu across the top of the screen that contains several other options.

Activating the action bar allows access to the following functions:

- **[B]** (Bump): Another opportunity to bump (close) the project.
- **[P]** (Print): Allows you to print either a formatted project form or the source definition for the item on which the cursor is resting when you called up the Action bar.

When you press **[P]**, the system displays the text box shown in Figure 8-8:

Bump	Print	Notes	TOC	QA	Other
Project JDSTUFF		Review Source Items (JDOE)		Realm DEV2	
Catch-all project for John		Reference		*Subs	Type BUG
3	Active & Assigned			Items	3 Class SY
Files	*Fld*Scr*Rpt	Processes	2	Programs	0
		STATE.MNT			
Print Project Details					
X Cancel		Defn item	PRJ Form		
(NOT USED)	0 Defns	0		BP HELLO	1
			Menus	0 Misc Edits	0
Text	Help Docs				

F2-Save F5-Drill F6-This Item F7-Bump F8-Scan F9-QA F10-Action

Figure 8-8: Print Project Details

- **[N]** (Notes): Allows you to enter technical notes about the project. It is a text field using the text editor. This makes the window itself behave like a full-screen editor or mini-word processor. Fields like this one that are Text Editor fields must be closed by pressing **[F2]** (which will also close this window).

When you press **[N]**, the system displays the text box shown in Figure 8-9:

Bump	Print	Notes	TOC	QA	Other
Project 50039		Review Source Items (JDOE)		Realm DEV2	
Technical Notes					
B					
3					
-					
F					
D					
D					
-					
(
-					
T					

F2-Accept Field F5-Block F6-Find F7-Go F9-Process F10-Action

Figure 8-9: Technical Notes

- T (TOC): Create a table of contents for producing an SystemBuilder-generated manual. It organizes all of the items checked out to the project by a guess-work process of starting at whatever menu it finds and placing the screens and other items afterward.

When you press T, the system displays the text box shown in [Figure 8-10](#):

Bump	Print	Notes	TOC	QA	Other
Review Source Items					
Project JDSTUFF		(JD0E)		Realm DEV2	
Catch-all project for John		Reference		*Subs	Type BUG
3	Active & Assigned			Items	3 Class SY
Files		*Fld*Scr*Rpt	Processes	2	Programs 0
<div style="border: 1px solid black; padding: 5px; margin: 10px auto; width: 80%;"> Table of contents created for PRJ*JDSTUFF. You will want to review, sequence (F7), and possibly add to this created item. </div>					
(NOT USED)		Return		Edit TOC	
				LO 1	
		Menus		0	Misc EDits 0
Text		Help Docs			

F2-Save F5-Drill F6-This Item F7-Bump F8-Scan F9-QA F10-Action

Figure 8-10: Table of Contents

If you select *Edit TOC*, the system displays the screen shown in [Figure 8-11](#):

Bump	Print	Notes	TOC	QA	Other
Review Source Items					
Build Contents List					
Manual Id	PRJ*JDSTUFF	Manual Type		U_	Help Level 3
Heading		Footing			
Type Id	Description		Txt Options	Chap	Section
C	Chapter - Misc Proce		N		N

F2-Save F4-Del F5-User Doc F6-Assoc.Pr F7-Reseq F8-Gen F9-Prod F10-Action

Figure 8-11: Editing the Table of Contents

- **[Q]** (QA): Allows you to define test steps of the project. Each step may then have multiple iterations of testing with a pass or fail status related. Reports and checks can be constructed to insure that all steps have been passed.

For more information, see [“Setting Up the Project for QA” on page 91](#).

- **[O]** (Other): This is another version of the `/SRC` screen which is configurable for other environments (non-SystemBuilder). This additional screen may be used with an SB+ application or in place of the main `/SRC` screen.

Searching for Strings

Use the `/SRCH` feature to search the entire software configuration (or designated portions) for one or more strings using Boolean (AND, OR) connectives. The results of the search are displayed on a screen resembling—and with similar functionality to—the Review Source Items screen. You can save the resulting set of components as a new PRC project or add them to an existing PRC project.

Examples of use:

- Search for the string “DEBUG” anywhere in the application before freezing a release.
- Search out all uses of a particular variable.

You can enter more than one string. For example, you want to find every use of the `@WORK` variable, position 1. You search for `WORK<1>` and `W1`, to be sure you catch all of them.

This option searches for one or more specific strings within the entire application or within designated “file sets.”

The file set `SOFTWARE` comes pre-defined and may be searched (the tools files without all the dicts) by specifying `SOFTWARE` in the *File Set* field. For information about creating and using file sets, please see the **FILES** menu help.

- In an SB+ development environment, if you do not specify that the search is within a file set, the system will search all tools files (`XXPROCESS`, `XXPROGS`, `XXDEFN`, `XXMENUS`, and so on), as well as the dictionary sections of all files listed in the main system **FILES** item. To see that file list, type `/TM` to bring up the SB+ Tools menu, then select **Other Tools/Utilities > Housekeeping > Data File Characteristics**.
- In a non-SB+ development environment, the `SOFTWARE2` and `WORK` file sets will be used if no others are specified.

To search for strings, perform the following steps:

1. Press **[S]** for **Search** from the PRC menu ([Figure 1-3 on page 21](#)) or type the `/SRCH` command from anywhere in your system.

The system displays the Search Utility screen, similar to the one shown in [Figure 8-12](#):

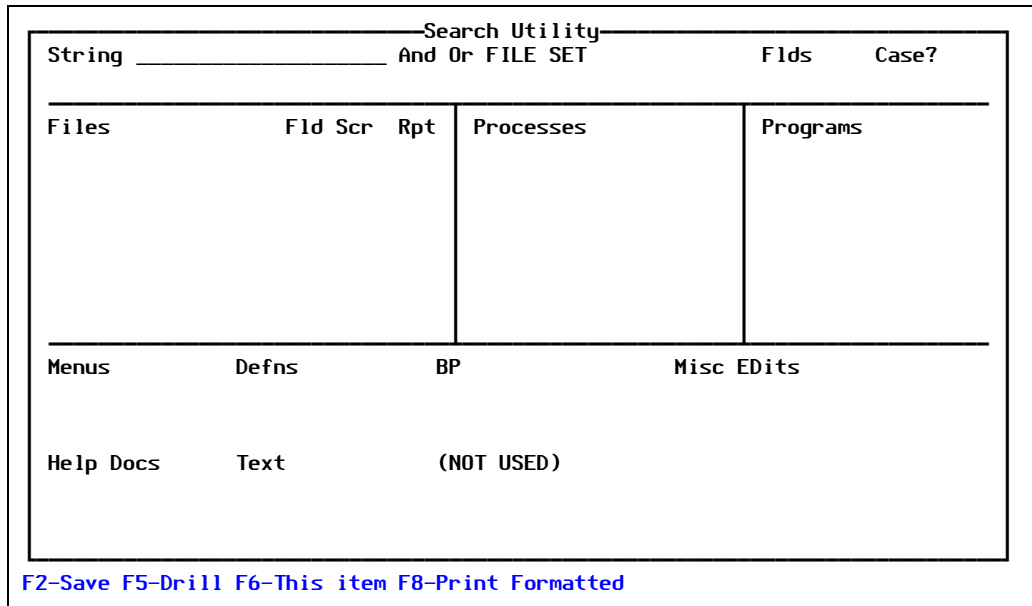


Figure 8-12: Search Utility screen

Explanations of the fields are as follows:

String The keyword or phrase for which you want to search throughout the system.

And Or The boolean connective you want to use to narrow your search. A value in this field is not necessary if you are searching for just one string.

If you want to search for more than one string, use:

- **AND** if you want only items that contain all of the strings
- **OR** if you want any item that contains any of the strings

File Set A subset of files to be searched. The system default is all software tools files plus the DICT of all files in the system’s FILES control record. A file set reduces the number of places P will search, if you specify a set, thereby reducing the amount of time to find the keyword or string.

For more information on file sets, see the *Administration Guide*.

- Flds** If you are interested in locating a particular string (or multiple strings) that only occur in a certain field position within the searched file items, enter the field (or fields) here.
- For example, you are searching for all monetary conversions. You search for the string MR (in capital letters because it is case-sensitive). You can also specify that you only want items that have an occurrence of that string in field 7 (that will isolate the conversions.) Otherwise, the string MR might be common and produce a lot of results that are not relevant.
- Case?** Indicates whether you want the search to be case-sensitive. Valid entries are:
- **Y** — Yes, search for the keyword or string exactly as it appears in the *String* field
 - **N** — No, search for the string in both upper- and lower-case.

2. Fill in the fields you need to search for the string or strings you want.

Enter multiple strings and indicate whether they are to be “and” or “or” connected; for example, find all programs and processes that use *both* of these strings, or *either* of these strings.

3. Press **F2** (Save).

After the first fields, which concern the search itself, this screen is virtually identical to the Review Source Items screen. All field functionality is the same, as well as the **F5** (Drill) and the **F6** (This Item) options. Please refer to that section of the manual for a review of the options and features available from these screens.

Note

Items appearing on this screen are not checked out to this screen, as it is not a project. Therefore, you can remove unwanted items by simply using the standard backslash (\) to delete field contents.

However, if you press **F2** at this screen, PRC will create a project and check out the items on this screen to the newly-created project.

If you do not press **F2**, the search result is *not* saved as a project; however, while using this screen and the **F6** (This Item) features on it, the system tracks items that have changes made to them against whatever project the user was currently assigned.

Reviewing the Changed Items Stack

Similar to a TCL stacker, the **/STACK** feature presents a view of all items that have been modified by that user in reverse chronological order. From the PRC Log Stack screen, you can eliminate redundant listings, edit any item in the stack, view a history trace of any item, or view all items that are backed up for a particular software component.

Background: Each time you press **F2**, PRC keeps a “stack” of the items that are being logged, similar to a TCL stack.

To review that items that were stacked against a particular user on a particular day, perform the following steps:

1. Press **A** for **Stack** from the PRC menu (Figure 1-3 on page 21) or type the **/STACK** command from anywhere in your system.

The system displays the PRC Log Stack screen, similar to the one shown in Figure 8-13:

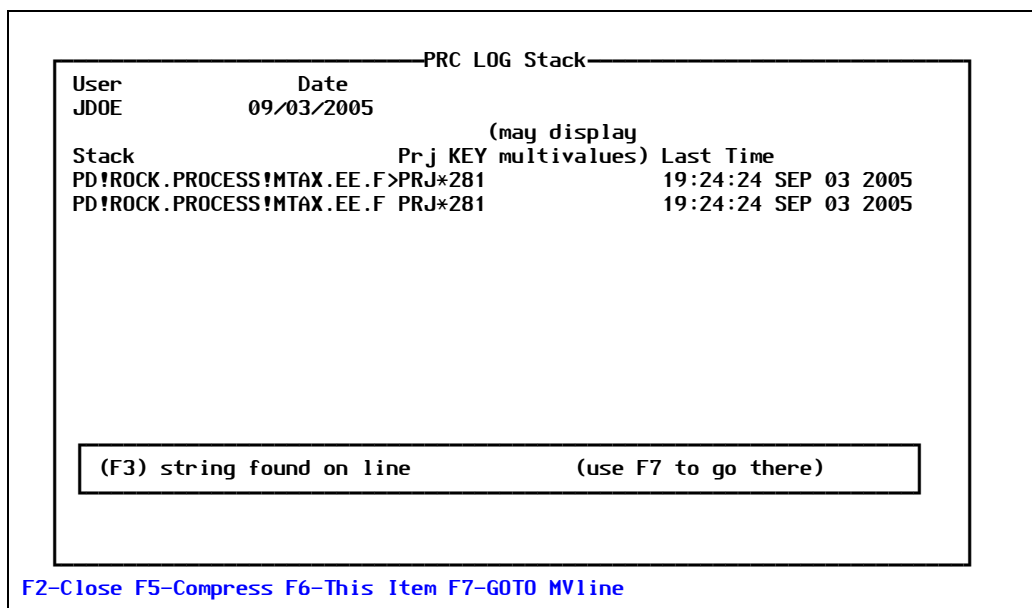


Figure 8-13: PRC Log Stack screen

The system displays a list of all items changed and the time/date stamp when they were changed, in reverse chronological order (the item most recently changed is on top). An item may be listed multiple times.

Explanations of the fields are as follows:

User The user’s name. The default is the current @USER.ID value.

Date	The appropriate date. The default is the current (today's) date.
Stack	The format of the item in the list is similar to a revision log item: type!filename!itemname. Examples in the screen above include, the field in the CUST file called NAME , the screen in the CUST file called MAINSCREEN , a process, a program edit.

2. You can use the following function keys:
 - a. If you press **F5** (Compress), the system displays the list in alphabetical order by the type of item and multiple/redundant entries are eliminated.
 - b. If you press **F6** (Edit), the system returns to editing a given item from the stack.
 - c. If you press **F7** (Trace), the system reviews the change history of any item in the stack. This displays the same Trace History screen available from **/SRC** and by using **/TRACE** from within the SB+ tools.
 - d. If you press **F8** (Backups), the system searches out all backed-up copies of the specific item, presents them in order—see the next section, “Managing Backups”.

IMPORTANT

The backup screen is available from the **F6** (This item) function on the **/SRC** screen also, but with a fundamental difference: While the **/SRC** is “project-centric,” meaning that the backup copies listed will be only those versions during the work on that particular project, the **/STACK** screen **BACKUP** option will review all “back up” versions of the component, regardless of what project the work was done against.

3. If you are finished using this screen, press **F2** (Save).

Managing Backups

A backup is created each time an item is modified, up to the daily number specified on the user profile (the default is three). In addition to the daily number, the system retains a “first daily.” Thus, if a program is edited ten times in one day, there are four backups: one before the first edit today, then the most recent three.

Only those changes made against the current project being are included. When a project is cancelled, these backups are used to replace all components to their original state.

When a project is “checked in” (usually by rolling out), all backups are cleared away except for the first edit to each item, so that actual changes for a project may be reviewed. (Incidentally, in a separate function but related to this—the software as it is delivered is also preserved, and in an on-line roll-out the software replaced on the destination may also be preserved.)

To work with backups, perform the following steps:

1. From the Log Stack screen (Figure 8-13 on page 102), press **F6** (This Item)
2. Select **Backups**.

The system displays the Backup Management screen, similar to the one shown in Figure 8-14:

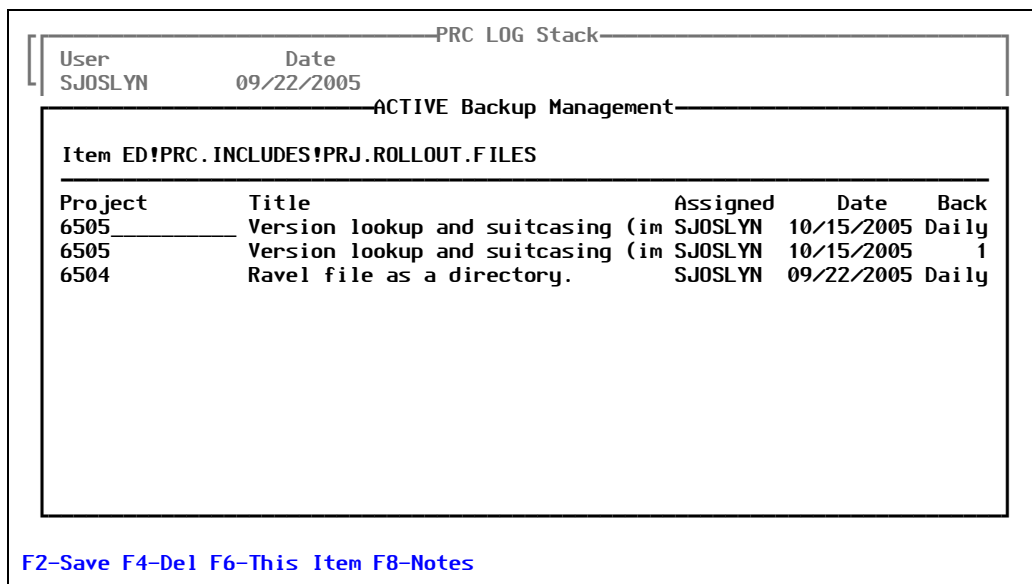


Figure 8-14: Backup Management screen

Explanations of the fields are as follows:

- Full Item** The complete revision log key; for example, **FD!CUST!NAME**.
- Edit Prj** The project key against which the item was edited and tracked.
- Date** The date on which the change was made.
- Back** The sequence. Say it out loud with the word “back,” as in “one back,” “two back,” and so on. The most recent backup is today’s date, your project, and the sequence **1**.

3. You can use the following function keys:

- a. If you press **[F5]** (Notes), the system displays any detailed documentation that was entered against this item, which may describe the purpose of the changes.
 - b. If you press **[F6]** (Edit), the system executes an appropriate tool to edit an individual backup item or offers to run the Knit utility, which will display the backup item in comparison with the current version of the item.
For more information on Knit, see the *Software Knitting Guide*.
 - c. If you press **[F7]** (Copy IN), the system reverts back to a particular backup version of the item. This option backs up your current version into the “1 back” slot, so that a revert could be undone by reverting again.
4. Press **[F2]** (Save).

Where Are the Backups Kept?

Project Development

During the active life of a project, backup copies are kept in a file called:

`REALM.SYSTEM.BU`

There is at least one copy for each day that an item was changed. More copies may exist, up to the number set on the specific programmer’s user profile.

The key to this file includes the “log key” (for example, `ED!filename!itemname`), as well as the project and the date and a sequence number. If you wanted to find the most recent backup copy of a particular item, you could select:

- The file for the item name
- The date
- The **[&1]** ending, which indicates “one back” or “one copy ago”

There are fields set up in the dictionary of this file that may be helpful.

Project Closeout

When a project is closed out, the intermediate copies are all destroyed but the oldest copy of each item on the project is moved to `REALM.SYSTEM.ARCHIVE.BU`.

Project Rollout

When a project is rolled out, the copy of what is being delivered goes into `REALM.SYSTEM.PRCRELEASE`. When the copy of what is *replaced* on the destination account or machine is placed into `REALM.SYSTEM.PRCJRNL`.

Note

If the rollout is from one account to another on the same machine, this all happens at once and the files are all on the same machine.

However, in the case of a rollout to a ravel file:

- The **ARCHIVE.BU** and **PRCRELEASE** files are populated when the rollout to the ravel file is performed
- The **PRCJRNL** copies are made on the destination machine during the unravel

Reviewing Backups

On Open Status 3 Projects in Development Realms

The easiest way to review backup copies for an active (status **3**) project is to:

1. Go to the **/SRC** screen, place the cursor on an item, then press **[F6]** (This Item).
2. Select **Compare**.
3. Select **Backup**.
4. Scroll through the date and sequence numbers to find the one you want to see in more detail and press **[Enter]**.

The system brings up the Knit function with the current version of the program on the left side and the backup copy that you selected on the right side.

On Open Projects That Are Not Checked In

There are two states of projects that we tend to think of and refer to as “open”:

- One is an active project—in status **3**.
- Another is as long as the items remain checked out in the development realm.

The multiple/transient backups as described previously are still available until the project is checked in. Remember that checking in occurs in one of two ways:

- The project is rolled out to a realm that has the control setting on the realm screen to do the check-in at rollout time
— or —
- The project is changed manually to a status of **C** or **Cx** (where **x** is any letters or numbers)

So, while the project is not checked in, the **[F6]** (This Item) > **Compare** > **Backup** route can still be taken; however, you will not be able to edit the item or revert to the backup copy because the project is “locked.”

On Closed (Checked In) Projects

When checking in occurs, intermediate BU items are destroyed and the oldest copies are archived into **ARCHIVE.BU**. You can still review and compare them, but you have to use **TRACE** to do so.

TRACE can be accessed in one of several ways:

- Log into the **PRC** account, then select **Action > Global Trace** to:
 - Search the whole system, across all realms
 - Use the function keys to drill down into found copies
- Go to the **/SRC** screen on the project, press **F6** (This Item), choose **Trace**, then put the cursor on the project in the *Historical* column and use the function keys to drill in.
- Go to the **/STACK** screen, and use the function keys to go to **Trace**, then drill in further.

On “Live” Systems With PRC Installed

When PRC is installed on the destination machine, the unravel does *everything* to create the same history on that machine—as if the work *had* been done on that machine. You can use the same steps described in “On Closed (Checked In) Projects” above.

On “Live” Systems Without PRC Installed

If set up properly, PRC can make backup copies during the simple extraction and unravel programs—the ones you use at sites that do not have PRC. These programs will copy into this backup file, but there is no PRC “intelligence” on the site to do anything with them. Nevertheless, they are there as a precaution and knowledgeable personnel can get to them.

In addition, settings on the development machine can put the backup copies right into the unravel file. To set this up, perform the following steps:

1. Log into the **PRC** account.
2. Select **Setup > Preferences**.
The system displays the PRC Preferences screen.
3. Press **F6** (Addl Ops).
4. Select **Ravel/Extractions**.

The system displays the Extraction Options screen, similar to the one shown in [Figure 8-15](#):

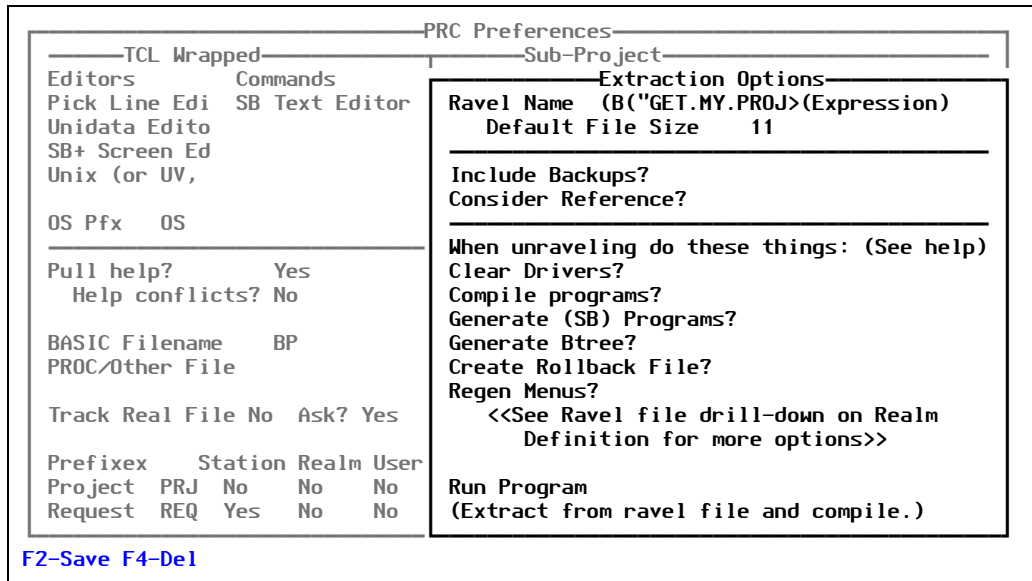


Figure 8-15: Extraction Options screen

5. Enter **Yes** in the *Include Backups?* and *Create Rollback File?* fields.
6. Press **[F2]** (Save).

You will have to manually search and revert to these copies on the destination machine—PRC is not there to do the work!

Outside Editors or Utilities

Several vendors of outside editors have added logic to their application to support PRC. If you are want to use **Wintegrate**, **Accuterm**, **Hyperterm** or **Pegasys** editors, please call for specific instructions.

For other implementations where you can get to the read and the write events, put these same subroutines in place—but be sure to set the variable **P.TYPE** to something other than ED, FD, SD, or MD. You will need more logic when checking the value of **GONO** and whatever is in **MSGBOX** to interact with the user about what to do.

Note

Beginning with PRC Version 6, much of this has been automated for certain outside editors.

Tracking Queries

From the PRC menu (Figure 1-3 on page 21), select **Tracking Queries**.

The system displays the Tracking Queries sub-menu:

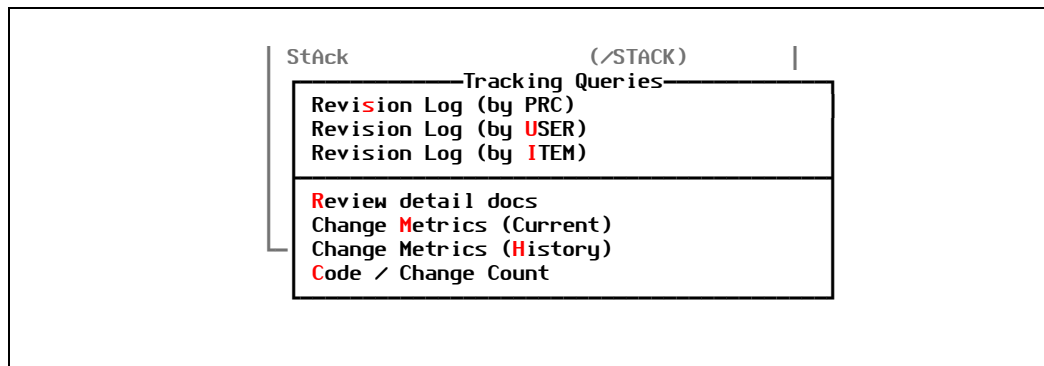


Figure 9-1: Tracking Queries menu

This chapter explains the options in the Tracking Queries menu.

Overview

“Tracking Queries” are a small number of reports that have to do with what’s being worked on by whom—from a current, active, this-realm, right-now perspective.

There are many more reports in the PRC account which provide useful information from a broader perspective.

Menu Selections

To select a menu option, use the arrow keys to move around the menu, then press **Enter**, when the selection is highlighted. An option may also be chosen by entering one of the following keys:

- [S] for Revision Log (by PRC). Access query which lists revision logs by input PRC task ID (SCR).
- [U] for Revision Log (by USER). Access query of active revision logs by input user ID.
- [I] for Revision Log (by ITEM). Access query of active revision logs by software item name (or SCR).
- [R] for Review detail docs. Review the detail documentation entered during development in association with and input SCR or software component name.
- [C] for Change Metrics (Current). "Metrics" are ways of measuring change or progress in the software development life-cycle. These reports simply attempt to quantify the changes.
- [M] for Change Metrics (History). Quantifies what has changed, historically, during a range of dates. (More reports of a metric or quantifying nature are in progress.)

After an option is selected and processing is complete, the menu redisplay and another option can be selected.

Revision Log (by PRC)

The "revision log" (or REVLOG file) is where all active changes are tracked. This report, available from the tracking options menu from the within the tracked account will list, from the default (current system, current realm) REVLOG file, all active tracking records.

This version of the report is by PRC project ID. The system prompts for the project before the report is processed.

Active Revision Log by Project			Page	1	22 Oct 1996
TP	File Name	Item Name	Prj	USR.....	DATE.....
D	PRCDEFN	PRC.BRANCH.DEL	1	SUSAN	10/18/96
D	PRCDEFN	PRC.BRANCH.KNIT.OK	1	SUSAN	10/18/96
D	PRCDEFN	PRC.CHG.METRIX	1	SUSAN	10/22/96
D	PRCDEFN	PRC.KNIT.NOT	1	SUSAN	10/18/96
D	PRCDEFN	PRC.REVLOG.REVIEW	1	SUSAN	10/22/96
D	PRCDEFN	PRC.SETUP.FINISHED	1	SUSAN	10/15/96
ED	PRCPROGS	PRC.TESTPROGRAM	1	SUSAN	10/18/96
PRC					

Revision Log (by USER)

A second-approach to the active tracking records in the default (current system, current realm) revision logging file, this one prompts for the user ID and will list all active tracking items for that user, regardless of project.

Active Revision Log by User		Page	1	22 Oct 1996
Prj	TP	Item Name	USR.....	DATE.....
1	D	PRC.BRANCH.DEL	SUSAN	10/18/96
1	D	PRC.BRANCH.KNIT.OK	SUSAN	10/18/96
1	D	PRC.CHG.METRIX	SUSAN	10/22/96
1	ED	PRC.REVHIST	SUSAN	10/21/96
1	ED	TRACKING.ITEMS	SUSAN	10/16/96
1	ED	PRC.SE.SUB	SUSAN	10/18/96
1	ED	TEST	SUSAN	10/18/96
1	ED	TEST&PRJ*1	SUSAN	10/18/96
1	ED	TEST*DEL	SUSAN	10/18/96
1	FD	DFLT.RLM	SUSAN	10/15/96
PRC				

Revision Log (by ITEM)

Still a third way of looking at the items which are actively tracked, this report looks at the file from the perspective of the item itself. It allows entry of a string which may be any part of the item name (program name, field name, etc.) and produces a similar report to the 'Revision Log' by PRC Project and by User, this time by the items tracked.

Item (string) entered was **BRANCH**, two items were found, whose IDs contained this string:

Active Revision Log by Item		Page	1	22 Oct 1996
Project	Ptype	Item Name	USR.....	DATE.....
PRJ*1	D	PRC.BRANCH.DEL	SUSAN	10/18/96
PRJ*1	D	PRC.BRANCH.KNIT.OK	SUSAN	10/18/96
PRC				

Review detail docs

A dialog presents the option to report detail documenting by item or by project, as shown in Figure 9-2:

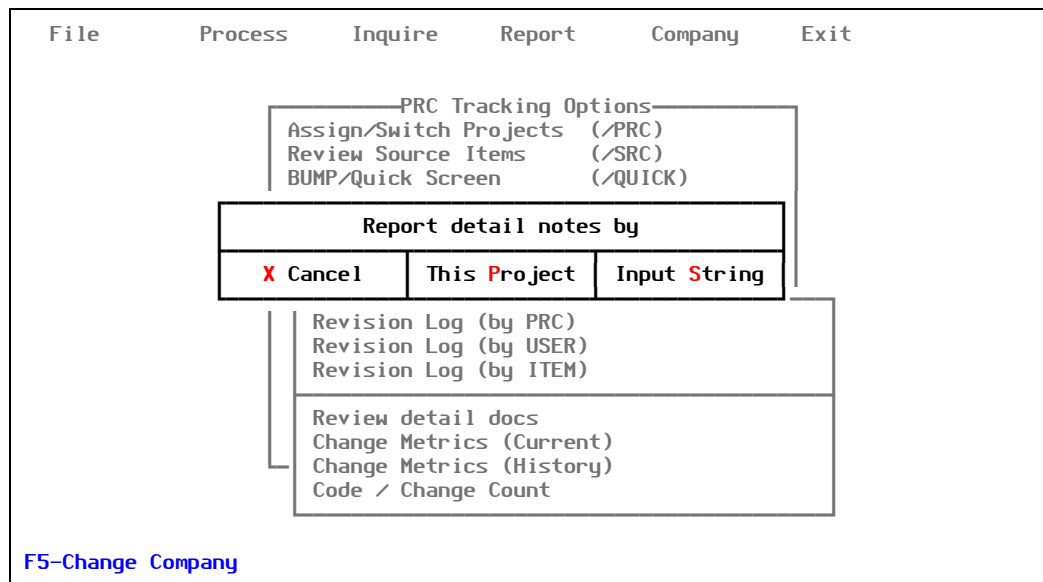


Figure 9-2: Report Detail Notes dialog

When the detail documenting reports option is chosen and the “by Item” window selected, the prompt will be for item name (or partial item name). Similar to the previously mentioned report, this one lists all of the detail documenting entered against an item (or items that match the partial string entered), regardless of the project against which they are tracked.

The report generated will be similar to the one shown on the next page:

Documentation Detail by Item		Page	1	22 Oct 1996
Tp	Item Name	Date	Prj	Note
ED	PRCPROGS!PRC.COMPARE	12/14/95	370	BASIC PRCPROG ctrl H, ctrl H oops
ED	PRCPROGS!PRC.STAMP.PLUS	12/06/95	347	Several PRJs PRC.STAMP.PLUS for REL 2.9
ED	PRCPROGS!TEST	12/09/95	369	Put a line of documentation, another day.
		12/02/95	344	This is a line of documentation
PD	PRCPROCESS!PRC.LOAD	12/06/95	347	Oh please don't make me write any more!
[405] 5 items listed out of 5 items.				
PRC				

Documenting By Project

When the detail documenting reports option is chosen and the “by project” window selected, the project ID will be prompted for and then all detail documenting notes (the ones added on the one-line prompt at the bottom of the screen whenever a change was filed if the user profile option was set for it to do so) for that project will be presented, including the item name that the note was made against.

The report generated will be similar to:

Documentation Detail by Project			Page	1	22 Oct 1996
Tp	Item Name	Date Prj	Note		
ED	PRCPROGS!TEST	12/02/95 344	MM		
ED	PRCPROGS!PRC.STAMP.PL	12/06/95 347	Several PRJs affected PRC.STAMP.PLUS		
PD	PRCPROCESS!PRC.LOAD	12/06/95 347	Oh please don't make me write any more.		
PD	PRCPROCESS!PRJ.DEFKEY	12/06/95 347	I'M NOT FINISHED HERE		
ED	PRCPROGS!TEST	12/09/95 369	put a line of documentation		
ED	PRCPROGS!PRC.CREATE.C	12/14/95 370	BASIC PRCPROG OMPARE		
[405] 6 items listed out of 5 items.					
PRC					

Review Change Metrics (Current)

Many new reports will be added to continue down the path of identifying metrics—or quantifying the changes in your software development in terms of who, what, when, what kind of items, what kind of changes.

This report is intended for “lite sites” and reports changes made on site as well as those delivered for a total count of changes in a given time frame.

Change Metrics (Current)	Page	1	22 Oct 1996
Component Type		Cnt	

Definition Record		14	

Edited Item		149	

Field Definition		33	

Menu Definition		5	

Process Definition		30	

Screen Definition		21	
=====			
** Grand Total		266	
[405] 266 items listed out of 266 items.			
PRC			

Review Change Metrics (Historical)

Similar to the Site Metrics report that is based on what is in the current/active logging, this particular report runs from the development system itself and polls the history files for the same sort of quantifying information: what kind of changes, how many, when and by whom?

Change Metrics (Historical)		Page	1	22 Oct 1996
Ptype Name	Ptype Cnt			
Definition Record	14			
Edited Item	149			
Field Definition	33			
Menu Definition	5			
Process Definition	30			
Screen Definition	21			
** Grand Total	266			
[405] 266 items listed out of 266 items.				
PRC				

Glossary

assign The verb that describes the association of a project to a developer for tracking purposes. Multiple PRC projects can be assigned to one developer and the word means different things in each case: PRC tracks the changes against one project for each assigned user at a time. And according to user profile settings, PRC tracks the changes against one project for each session.

There may be any number of projects open for each user, and each of these projects is assigned to that user. The *Assigned to* field on the project contains that user ID.

To review the projects that are *assigned to you*, type **/PRC** and press **F3** (Lookup). When you choose a project from that lookup list, the system:

- Changes the project to which you are assigned to the one you choose
- Tracks the changes you make against that project

audit (changes) The process that PRC performs in the background while it is enabled in an account. PRC tracks, or audits, all authorized changes to:

- The software being developed
- The PRC configuration
- Any PRC settings

bump

- 1.) A popular dance from the 1970s.
- 2.) Any process that changes a project from one status to another, typically to the next one up—but *does not move the software*.

The most common bump is the development bump, which changes the state from **3** (Active and Assigned) to **4** (Development Complete). This particular bump also involves a variety of diagnostics that ensures the project is truly ready for the next step. Another example is the testing bump, when the QC tester declares the project passed the testing phase.

A rollout is not considered a bump because, while it changes the project state, it also moves the software.

/BUMP The SB+ command that bumps a project from status **3** to **4**. PRC performs a variety of diagnostics that ensures the project is truly ready for status **4**.

branch copy

A copy of an item that is already checked out to another project.

When a developer attempts to modify an item that has been previously checked out to another project, the system offers to make a “branch copy” of the item (if the developer’s security clearance is set appropriately). This item is copied and named with the project as a suffix; for example, `<itemname>&PRJ<nnn>`.

For some security clearances, the branch copy will not include the changes made on behalf of the conflicting project; thus, the two versions of the program are branched from the same starting point.

Once a branch copy is made for a particular item on a particular project, all future edits—while assigned to that project—are diverted to the branch copy.

Branch copies may not be rolled out. Tools are provided to merge the changes back into the primary item, once the primary item has been checked in by the first user and/or project.

check in

The returning an item or component to the pool of items that are available for modification.

When work against an item is complete (or cancelled) the item is “checked in” or “checked back in” to the system. Typically occurs during a rollout. Also occurs through manual intervention by using the appropriate **Cn** status code, or when a project is canceled using an **Xn** status code, or when an item is deleted using **F4** (Delete).

The technical act of checking in an item/component is removing that item’s tag in the **REVLOG** file. In the case of rolling to **LIVE** or “closing” the project manually, the entry is moved into the **REVHIST** file. For more information, see the tip sheet on PRC transaction files.

check out

The tagging an item to a particular project, effectively removing it from the pool of available items and preventing it from being modified by other projects.

In a conventional SCM environment, each software item is checked out to a particular project in advance of work being done against it. It is possible to use this form of “pre-checkout” in PRC using the Review Source Items screen; however, it is not necessary as items will be checked for availability and then checked out to a project “on the fly” as the work is performed.

conflict

The condition when two or more projects check out or attempt to check out the same item.

Only occurs when one developer checks out an item, then a second developer who has the proper security clearance also checks out that same item. Most PRC installations have very few, if any, developers with the clearance level to allow this situation.

conflict scan	<p>The process that reviews all items on a particular project to insure that none of them are in conflict with any other projects.</p> <p>Runs automatically before a rollout. Can be run independently via [F8] from the /SRC screen. Setting on the Realm Master determine: whether a conflict scan is run, what realm(s) it runs against, and whether it checks for opened since a certain date or open and closed since a certain date.</p>
doom	<ol style="list-style-type: none">1.) The feeling of dread and despair that washes over a development team when they realize, after their release is totally messed up and will take days to recover, they should have purchased PRC.2.) The state of a deleted item that indicates it is to be deleted on any realm to which is is rolled out or unraveled.
drill down	<p>To display more detailed information about a particular field on a screen.</p> <p>Produced from the /SRC screen by placing the cursor on the item of interest, then pressing [F5] (Drill). Items that have drill-down capability are indicated by an asterisk on the screen (fields, screens, and reports).</p>
explode	<p>To pull all modified items onto a project's Review Source Items screen.</p> <p>Exploding works when a project is accessed via either the Project Master screen in the PRC account or via the /SRC screen in the tracking account and the project is status 3 (Active). A box appears in the lower right of the screen and the items on the project scroll through it.</p>
flick	<p>To take an item off of a project.</p> <p>An item can be flicked to another project or removed from a project completely (as long as there is no change associated to that item on that project or you are willing to revert the change).</p>
Install Tracking	<p>The process run from the PRC account that enables a pre-configured account for tracking projects in PRC. Can be run any time because it harmless (does not clear any tracking data).</p> <p>Should always be run when anything changes: files are added to the list of tracked files, a new realm is added, or a new account is added. Always a good first step in any troubleshooting situation because it resets all paths and pointers.</p>

- junk project** A special project that is considered as a temporary holding place or catch-all for miscellaneous work. Typically use for items that need a “home” after a developer has flicked them off of a project that is being prepared for rollout.
- Also useful for occasions when a programmer is not sure what work they will be doing, and therefore do not want the changes to be tracked against a project that has a known purpose or specific goal.
- Junk projects are recommended because of their convenience. Many teams use the developer’s initials as the junk project name. Keep junk projects open and active. Can be closed down periodically and a new one opened.
- Items checked out to junk projects are still checked out; therefore, junk projects must be periodically reviewed to figure out where the items really belong.
- /KNIT** The SB+ command that displays the Knit split-screen utility.
- Knit allows you to merge & edit items. Displays two versions of an item side-by-side and highlights the differences between them. Provides many tools for examining and editing the items. For more information, see the *Software Knitting Guide*.
- /PRC** The SB+ command that displays the Assign/Switch Projects screen.
- project** The standard, central unit of work in PRC.
- Sometimes known as a task or a software change request (SCR) in other software configuration management applications.
- project-centric** The concept that PRC organizes itself around the entity called a project. In PRC, the project is king: they have all changes tracked against them, time is charged to them, and rollouts are done according to them.
- Projects may be broken down and combined by the use of sub-projects and versions, but the orientation of the entire system is still centered around the project entity itself.
- /QUICK** The SB+ command that displays the Quick Project Update screen.
- ravel file** A hashed or directory file into which all components of a project or collection of projects are deposited for movement to another system. The contents of the ravel file can be delivered via one of several “unravel” options.

realm	<p>A defined region (directory, account, or accounts) that is governed by a particular set of rules. Those rules define who may make changes, under what conditions those changes can be made, and where software can be delivered or received.</p> <p>Can sometimes be a Pick/U2 account, but may comprised of more than one.</p>
remote realm	<p>A realm that exists in name only, as a reference. Rollouts to a remote realm create ravel files.</p>
requests	<p>1.) Popular songs asked for by name during radio programs, performances, and dance parties.</p> <p>2.) Frontline problem reports presented by the user community. Can be converted into a project. Sometimes attached to an existing project. Often resolved without ever opening a project or making a software change.</p> <p>Requests and projects can have one-to-many or many-to-one relationships.</p>
rollout	<p>The verb that defines the actions taken to deliver software from one realm to another via a project or a version. Refers to the same act, whether the result is a direct delivery into another working account or to a “remote” realm (in which a ravel file is created).</p> <p>The verb clause “rollout to a ravel file” may be shortened to “ravel out,” but the action and the steps required for the action is the same whether the rollout is directly into an account or indirectly into a ravel file.</p>
SCM	<p>The acronym for “Software Configuration Management,” which is the industry-standard expression for the entire set of activities related to what software is running where, since when, changed by whom, for what purpose, and so on.</p>
slash command (/)	<p>In SB+, any process may be invoked at any prompt by preceding it with the forward slash character (<i>/</i>). For your convenience, many of the frequently-used utilities in PRC are found on the PRC menu, and their associated slash command is listed next to them.</p> <p>You can even access the PRC menu via the slash command /PRCMENU.</p>
/SRC	<p>The SB+ command that displays the Review Source Items screen, which is the main screen from which you can access the core of PRC’s tracking account capabilities. You can view all items tracked against a project in a multi-column format. Use this screen to add, remove, research, and edit the items.</p>
/STACK	<p>The SB+ command that displays a list of items in reverse chronological order, showing when the items that were changed, stored by date, and the user that made the change or changes.</p>

This Item	<p>The function used to trigger various processes you can perform on an item, such as editing the item, compared it against its own backups or against other items, tracing its history, and reviewing notations about the item.</p> <p>To use This Item, press F6 on the Review Source Items screen (/SRC) when cursor is on the item for which you want more information.</p>
timer	<p>The automatic feature that can be set to track time against projects.</p>
tool file	<p>The six or eight (depending on how you count them) SB+ files that contain the software elements specific to an SB+ environment. Each of these tools files has its own column on the /SRC screen to display lists of items in that file that have been changed. For example, XXPROCESS, XXPROGS, XXDEFN, XXMENUS, XXUSERTEXT and XXHELP, where XX is the system ID.</p>
trace	<p>The feature that allows you to view an item's history, in terms of the projects on which it is currently being revised or ever has been revised. The actual change made on any historical project can be reviewed using Knit. Tracing is accessible from:</p> <ul style="list-style-type: none">• The /SRC screen and pressing F6 (This Item)• The /STACK screen
tracking account	<p>An account where changes are made and tracked. Has Project Tracking installed, which enables PRC in the account. Designated as a tracking realm on the Realm Master screen.</p>
unravel	<p>The process of delivering items from a ravel file to the appropriate location within a working account. Unraveling includes regenerating, compiling, and cataloging, which put the software changes into affect.</p>
versions	<p>In the context of PRC, a collection of projects that will be delivered together.</p>

About PRC

If you need to check on which version of PRC you are using and your number of licensed users, select **About PRC**. The system displays the About PRC screen, similar to the one shown in [Figure B-1](#):

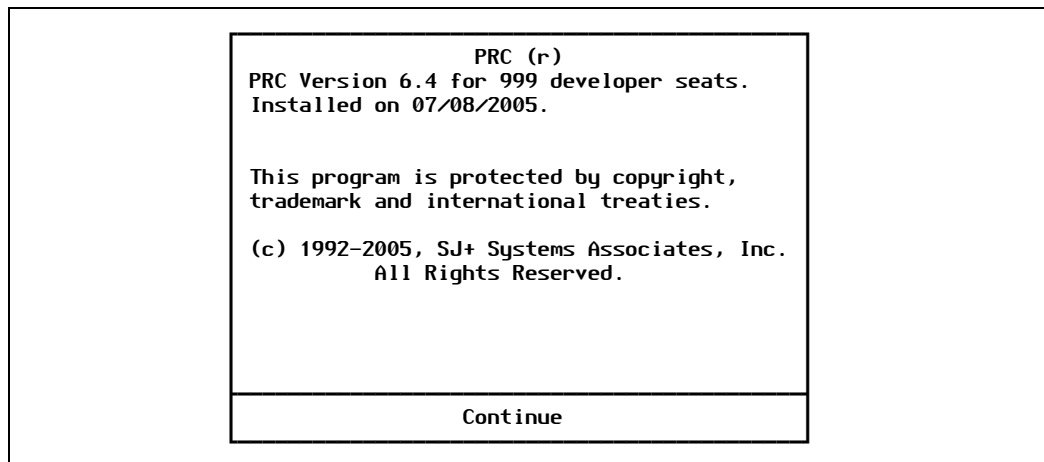


Figure B-1: About PRC screen

This should always contain PRC copyright and version information. Some minor patch releases and local customizations may not be reflected in this screen. Before relying on this information, check with SJ+ Systems Associates, Inc.

Symbols

- .mak ix
- .REVHIST 92
- .REVLOG 92
- /BUMP 61, 119
- /KNIT 33, 94, 122
- /PRC 35, 37, 122
- /PRC.PROFILE 81
- /PRCMENU 21
- /PRCTIME 83
- /QMGR 85
- /QUICK 60, 61, 70, 122
- /SRC 43, 46, 90, 123
- /SRCH 32, 99
- /STACK 32, 93, 102, 123
- /TM 99
- /TRACE 93, 103

A

- account
 - comparing system in two 94
 - PRCCUST 77
 - tracking 124
- Accuterm xiii, xiv, 108
- Action bar 96
- activity codes 83
- archives ix
- archiving
 - deltas x
 - methodologies explained x
- assign 119
- audit changes 119

B

- backup
 - copies 103
 - location 105
 - managing 103
- boolean 99
- branch 44
 - copy 120
- BU file 105
- bump
 - Action bar 96
 - defined 59, 119
 - occurrences 59

- bumping
 - branches 60
 - conflicts 59
 - deleted items 60
 - discarding items 46
 - explained 59
 - QA 92

C

- change
 - allows to overwrite item 44
 - disallow 44
 - merge 44
 - required to make copy of item 44
 - tracking viii
- change metrics 110
- check-in 120
- check-out x, 120
 - display items that are 45
 - mode x
- clearance
 - disallow changes 44
 - levels x
 - queuing 85
 - security 74
 - segregation of duties 74
- codes
 - activity 83
- column
 - configurable 48
 - Misc Edits 48
- compare
 - items 94
 - using Knit 94
- Compare project 94
- compliance ix
 - reporting viii
- components
 - backup versions 103
 - checking out x
 - display overview 45
 - review 45
- conflict
 - allowing 90
 - analysis 90
 - checking for 89
 - defined 120
 - prevention 90

- projects in same version 74
- resolution 90
- rollout 90
- scan 121
- scanning for 89
- connectives 99
- copyright information 125
- critical path
 - illustrated 29
 - Step 1 explained 35
 - Step 2 explained 43
 - Step 3 explained 59
 - Step 4 explained 67
 - use of Review Source Items screen 45
- customer request ix

D

- debugging 46
- deltas x
- destination realm 77
- development life cycle 19
- development tools 44
- DoD ix
- doom 43, 121
- drill down 121
- drilling down 43

E

- edit
 - change 44
- editors 44
 - actions after exiting 44
 - support PRC 108
- environment
 - database development x
 - mainframe x
 - realm ix
 - Review Source Items screen in non-SB 47
 - Review Source Items screen in SB 46
 - U2/Multivalued ix
- explode 121
- extension
 - .BU 105
 - .mak ix
 - .REXHIST 92
 - .REVLOG 92
- extract
 - customer side 78
 - simple 77
- Extract Ravel File menu 77
- extraction
 - how it works 78
 - options 107

F

- feature
 - Knit 94
 - Pre-scan 73
 - Queues 85
 - source control ix
 - Timesheets 83
 - User Profile 81
- field
 - navigating a xvi
 - special keys for xvi
- fields
 - *Fld 49
 - *Rpt 50
 - *Scr 50
 - Activity 84
 - And Or 100
 - Back 104
 - Begin Time 85
 - Brief Description 40
 - Case 101
 - Clear project at Menu 83
 - Conflict Default 83
 - Create Rollback File 108
 - Date 84, 103, 104
 - Defns 50
 - Diary Details 84
 - Documenting on 82
 - Edit backups 82
 - Edit Prj 104
 - Elapsed Time 85
 - End Time 85
 - File Set 100
 - Files 49
 - Flds 101
 - Full Description 40
 - Full Item 104
 - Include Backups 108
 - Log Key 86
 - Misc Edits 49
 - Module 39
 - Number of stacks 82
 - Preferred Editor 82
 - PRJ by Port 83
 - Procedures 50
 - Processes 50
 - Programs 50
 - Project 37, 39, 40, 48, 84
 - Queued for 86
 - Stack 103
 - String 100
 - Target Realm 75
 - Text 50
 - Title 40

- Total Elapsed 85
 - Trace Dt 93
 - Trace Hist 93
 - Trace Prj 93
 - Type 40
 - User 84, 102
 - Version 40
- figures
- critical path 29
 - decisions about rolling out 69
- file
- item ix
 - view related components 43
- flick
- defined 121
- function keys
- common xv
- ### H
- Hyperterm 108
- ### I
- IEEE ix
- ISO 9003 ix, 74
- item
- adding to project 51
 - associated with project 45
 - automatically add to project 45
 - backups 103
 - branch 44
 - checked out 45
 - checking out 51
 - compare 54
 - comparing to another 43, 94
 - copies of 103
 - defined ix
 - deleting 43
 - discarded by bumping 46
 - doom 55
 - dooming for deletions 43
 - edit 54
 - editing 43
 - ignore 55
 - ignore deletion 43
 - making notes about 43
 - manually add to project 45
 - manually remove from project 46
 - move to another project 46
 - notes 55
 - oldest copy 105
 - queuing 85
 - remove 55
 - removing from project 43
 - reviewing 102
 - stack of 102
 - trace 54
 - tracing history 43, 92
 - tracking changes ix
 - work with 48
- ### J
- junk project 122
- ### K
- key
- adding lines xvi
 - Alt+Tab xvi
 - backslash xvi, 101
 - down-arrow xvi
 - Escape xvi
 - F2 xvi
 - F5 (Drill) 43, 101
 - F5 (Notes) 105
 - F6 (Edit) 105
 - F6 (This Item) 43, 101
 - F7 (Copy In) 105
 - slash xvi, 123
 - Tab xvi
 - up-arrow xvi
- knowledge
- required vii
- ### L
- log key 105
- log stack 102
- ### M
- MAK file ix
- menu
- PRC menu 21
 - Tracking Options 21
- menus
- Extract Ravel File 77
 - Tracking Management 81
 - Tracking Options xiv
 - Tracking Queries 109
- metrics
- change 110
- ### N
- notes
- Action bar 97
- ### O
- object

- item ix
- other
 - Action bar 99
- P**
- P.TYPE 108
- Pegasys 108
- peripheral management viii
- PRC
 - About 125
 - auditing viii
 - compared to Visual Basic ix
 - defined vii
 - deployment control viii
 - documents ix
 - entities ix
 - familiar source control features ix
 - peripheral management viii
 - pre-check out x
 - project management viii
 - project-based ix
 - project-centric ix
 - project-oriented ix
 - reporting viii
 - request problem reporting viii
 - rollback control viii
 - security management viii
 - source control viii
 - source control system viii
 - test management viii
 - unfamiliar source control features ix
 - version 125
- PRC Administration menu 71
- PRC Preferences screen x
- PRC Project Maintenance screen 72
- Preferences screen x
- Pre-scan feature 73
- print
 - Action bar 96
- project
 - assigning a 35
 - assigning explained 35
 - changes viii
 - child viii
 - collection of ix, 22
 - compare 94
 - contained in versions 73
 - defined viii, 122
 - development life cycle 19
 - group of 23
 - master 23
 - parent viii
 - removing an item 43
 - stages xi, 19
 - states xi, 19
 - status code 19
 - status number xi
 - sub-project defined viii
 - tracking viii
 - tracking time 83
 - version 22
 - within same version 74
- Project Master screen
 - rolling out 70
- Project Rollout screen 75
- project-based ix
- project-centric ix, 103, 122
- projects
 - what is tracked and managed 43
- Q**
- quality assurance (QA) 19, 23
 - Action bar 99
- quality control (QC) 19
- queries
 - tracking 109
- queuing 85
- R**
- ravel 67, 107
 - file 122
- realm
 - Compare project 94
 - defined ix, 123
 - destination 77
 - DEV ix
 - LIVE ix
 - on different machines 77
 - remote 123
 - source 77
 - TEST ix
- region
 - realm ix
- reports 109
- repository 92
- request
 - customer ix
 - defined viii
- requests 123
- reverse chronological order 102
- Review Source Items screen 95
 - Action bar 96
 - features of 45
 - in non-SB environment 47
 - in SB environment 47
 - layout by environment 47

- revision log 110
- revision tracing 92
- rollback
 - mechanism viii
- rolling back
 - conflicts 79
 - explained 78
- rolling out
 - clustered sub-projects 91
 - considerations before 67
 - explained 67
 - preparation 71
 - screen 75
 - security clearance 74
- rollout
 - defined 123
- rule
 - realm ix
 - set of ix
- S**
- sandbox x
- Sarbanes Oxley 74
- SBClient xiii
- scan for conflicts 89
- SCM 123
- screen
 - navigating a xvi
 - special keys on xvi
- screens
 - About PRC 125
 - Active Backup Management 104
 - Assign/Switch Projects 37, 39
 - Backup Management 104
 - blowup (drill-down) dialog 54
 - branched item dialog 56
 - Build Contents List 98
 - bump completed successfully 64
 - choices for a conflicting item 63
 - choices for a deleted item 63
 - choices for a missing item 62
 - choices for an unchanged item 64
 - Compare a System in Two Accounts 94
 - Compare against original item? 95
 - Daily Timesheet 84
 - deleted item dialog 55
 - Extraction Options 108
 - Knit Two Items 96
 - list of backups 95
 - Log Key list 95
 - Manage Queued Items 86
 - PRC Administration menu 71
 - PRC Log Stack 102
 - PRC Preferences 24, 108
 - PRC Project Maintenance 72
 - PRC Project Maintenance with source items 72, 79
 - PRC User Profile 82
 - Print Project Details dialog 97
 - Project Rollout 75
 - Quick Lookup 38
 - Quick Project Update 40, 62
 - result of pressing F7 (Bump) 61
 - result of pressing F8 (Scan) 57, 90
 - results of drilling down in Fld column 53
 - Review Source Items 47, 61, 91, 93, 96
 - Review Source Items (non-SB) 47
 - Revision Tracing 93
 - Search Utility 100
 - Stack 102
 - Standard Project Lookup 38
 - Table of Contents dialog 98
 - Technical Notes 97
 - Testplan 91
 - Version Master Record 73
 - Which Lookup? 37
- search for conflicts 89
- segregation of duties 74
- software change request (SCR) ix
- software configuration management 123
- source
 - file ix
 - item ix
 - object ix
 - tracking changes ix
- source control
 - archiving x
 - defined viii
 - familiar features ix
 - PRC functionality viii
 - project-centric ix
 - system viii, ix
- source realm 77
- stack 102
- stages xi, 19
- states xi, 19
 - general or common xi
 - list of 19
- status code 19
- status number xi
 - xi
- string
 - search for 99
- sub-project ix, 19
 - changes viii
 - clustered 91
 - creating 23
 - defined viii, 25

- explained 23
- how to use 27
- numbering 23
- on PRC Preferences screen 24
- opening a 27
- reasons to use 26
- special purposes 23
- syster 19
- syster defined 26
- types of 23
- why to use 26
- sub-version
 - defined ix
- super-project ix, 19
- syster sub-project 19
 - defined 26
 - example of 27
 - how to use 27
 - opening a 27
 - reasons to use 26
 - why to use 26

T

- Table of Contents
 - Action bar 98
- task ix
- TCL commands
 - KNIT 33
 - PRC 35
 - PRCMENU 21
 - SRC 43
 - SRCH 32
 - STACK 32
- technical notes
 - Action bar 97
- telnet xiii
- test plans 91
- This Item 124

- timer 124
- timesheets 83
- tool file 124
- tools
 - SB 99
- trace
 - defined 124
- tracing 92
- tracking
 - account 124

U

- unit
 - of work viii
- unravel 67, 107
 - defined 124
- user profile
 - check-out mode x
 - setting to allow changes 44
- utility
 - Knit 94
 - Scan 89

V

- version ix, 19, 68
 - archived ix
 - defined ix, 124
 - explained 22
 - master ix
 - reasons to use 23
 - rolling out 23
- Version Master Record screen 73
 - rolling out 70
- Visual Basic ix

W

- Wintegrate xiii, xiv, 108