

PRC

Administration Guide

Version 6

November, 2006



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 © 1992–2006 by SJ+ System Associates, Inc.

Version: November, 2006

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
More PRC Documentation	xi
Telnet Programs, Character Mode & GUI Mode	xii
Menu Navigation	xii
Standard Function Keys	xiii
Screen & Field Navigation	xiv
Screen	xiv
Field	xiv
Text Conventions	xv
Chapter 1: Maintenance Files	17
PRC Code Files	.21
Project Classes & Class Types	.21
Class Codes	.22
Type Codes	.24
Priority Codes	.28
Status Codes	.29
Drill Down	.34
Triggers	.35
Roles Codes	.36
Activities (Time)	.38
Resolution Codes	.40
Knit Commands	.41
Keywords	.43
PRC Master Files	.45
Project Master	.45
Drill Down	.50
Test Plan	.51
Source	.51
Stat	.52
Request Master	.53
Drill Down	.57
Related Requests	.57
Projects	.58

Table of Contents

Attachments	59
Version Master	60
Drill Down	62
Schedule	62
Sort	63
Pre-Scan	63
Roll Out	64
Customer Master	65
Work Order	68
Contract	69
Technical Information	70
Work in Process	70
Department Master	71
Chapter 2: File Sets & Test Plans	73
File Sets	73
Test Plans	77
Iterations	78
Sources and Pulling	79
Test Steps	80
Nested Test Plans	80
Chapter 3: User Profiles & Timesheets	83
User Profiles	83
Assigning Projects to the User	85
Defining User Preferences	87
Setting Up System Clearances	93
Defining User Restrictions	94
Setting Up Status Clearances	95
Timesheets	97
Chapter 4: Action Menu	99
Global Trace	99
Suitcase Maintenance	101
Rank Projects	103
Drill Down	104
Auto Rank	105
Clear Rank	106
Move Rank	106
E-mail & Notifications	108
Mail Preferences	108
Standard E-Mail Events	110
Standard E-Mail Text	111
Standard E-Mail Recipients	112
Manually Send an E-Mail	115
Ravel File Information	116

Chapter 5: Utilities	117
Create "Compare Accounts" Project	117
Create "Whole Account" Project	119
Compare or Combine Ravel Files	121
Clear PRC Data	121
Remove Realm and Pointers	123
Purge & Separate Archive Files	124
Clean Up PRC Orphans	125
Project Utilities	126
Review or Change States	127
Chapter 6: PRC Configuration	131
Realm Master	131
Spawning	139
Explanation	139
Example	139
Drill Down	140
Work Accounts	141
Suites	142
Additional Details	142
Archives	143
Account Master	144
System Master	146
Menus	148
Files	149
XConfig	150
Platform Master	150
Suite Master	152
Configuration Components	153
Station Master	154
Chapter 7: Setup Menu	157
Install Tracking	157
Name PRC files/items	158
Install Tracking	159
UN-install Tracking	161
PRC Preferences	162
Environment	167
Ravels and Extractions	170
Source and Object Code	173
Remark Line Stamp	176
Request Options	178
QA Options	179
APIs	181
File Options	182
Process Slots	184

Table of Contents

Audits187
Infrequently Used Options189
Standards191
Chapter 8: Glossary 193



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 *Administration Guide*, explains the various concepts and features of **Administration** 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

This manual provides menu-by-menu detailed explanation of the various master file maintenance screens, code file maintenance screens, configuration settings, and utilities that are involved in administrating PRC and managing your software development life-cycle.

This manual is focused on being a complete and thorough reference. There is a companion manual—the *Installation Guide*—that describes PRC’s initial settings, explains the setup, and refers to this manual for the more detailed explanations.

Numerous tip sheets are available that explain common, “occasional” administrative maintenance functions; for example, how to add users, how to add or delete a realm, or how to add an account. The tip sheets are brief summaries and, like the *Installation Guide*, refer to the more detailed information found in this manual.

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.

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 assurance (QA) or 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/Multivalue 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 1), 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

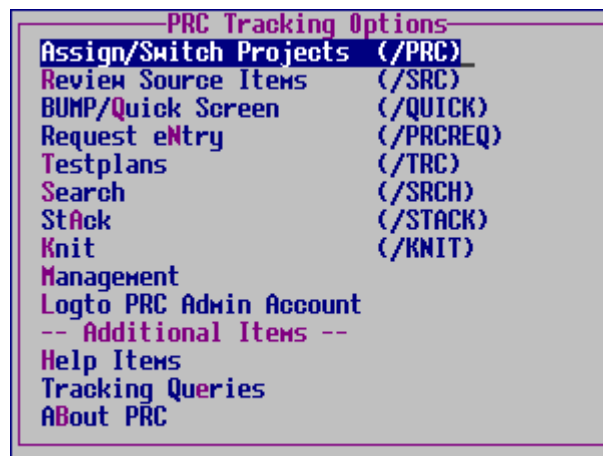


Figure 1: Tracking Options menu in character mode

In SB+ graphical user interface (GUI) mode (Figure 2), 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

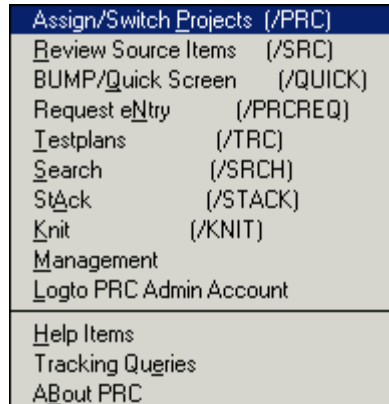


Figure 2: Tracking Options menu in GUI mode

- 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)

In PRC, **F10** actions almost always contain standard options to “copy” or “print,” but can include other functions, as well.

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
<code>Enter</code> , <code>Tab</code> , <code>Esc</code>	A key on the keyboard
<code>Ctrl+S</code>	A key combination where you press the keys simultaneously
<code>Alt P</code>	A key combination where you press the keys one at a time in the order presented

Maintenance Files

Code Files & Master Files Used Throughout PRC

PRC is a multi-value database application, just like yours. It has a number of files that it uses to store information. Within PRC are three primary kinds of files:

- Master files
- Code files
- Tracking files

Master files contain relatively static information that defines something important. The projects are stored in the Project Master file, the user profiles are stored in the User Master file, and so on.

When you select **Files**, the system displays the Files menu. It includes the options for the master files, as shown in [Figure 1-1](#):

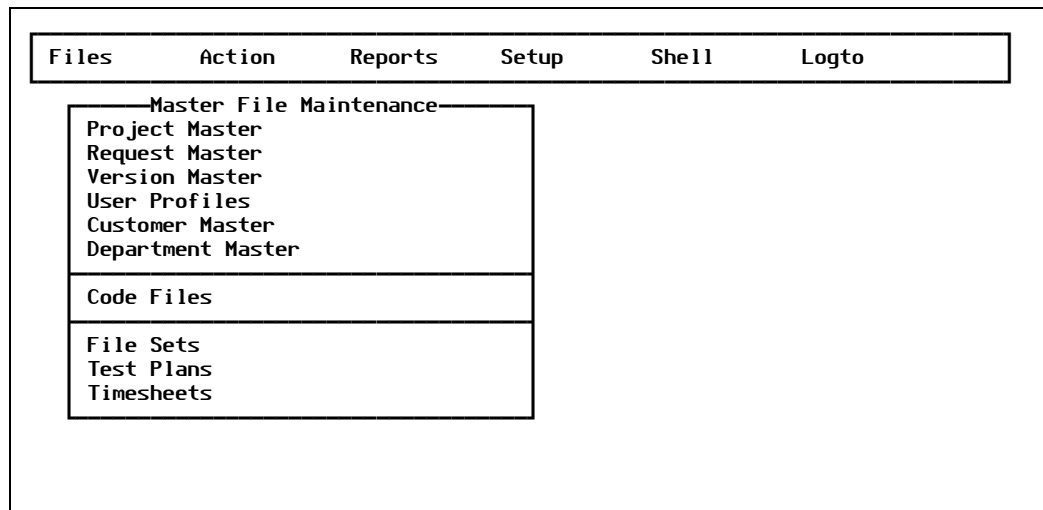


Figure 1-1: PRC Account, Files menu

Code files are very similar to master files in that they are relatively static and they store definitions; however, code files tend to store the definitions that customize your life-cycle: the status codes, the type codes, the priority codes. Another way to view this delineation is that code files support fields that are on the master files.

When you select **Files > Code Files**, the system displays the Code Files Maintenance submenu, as shown in [Figure 1-2](#):

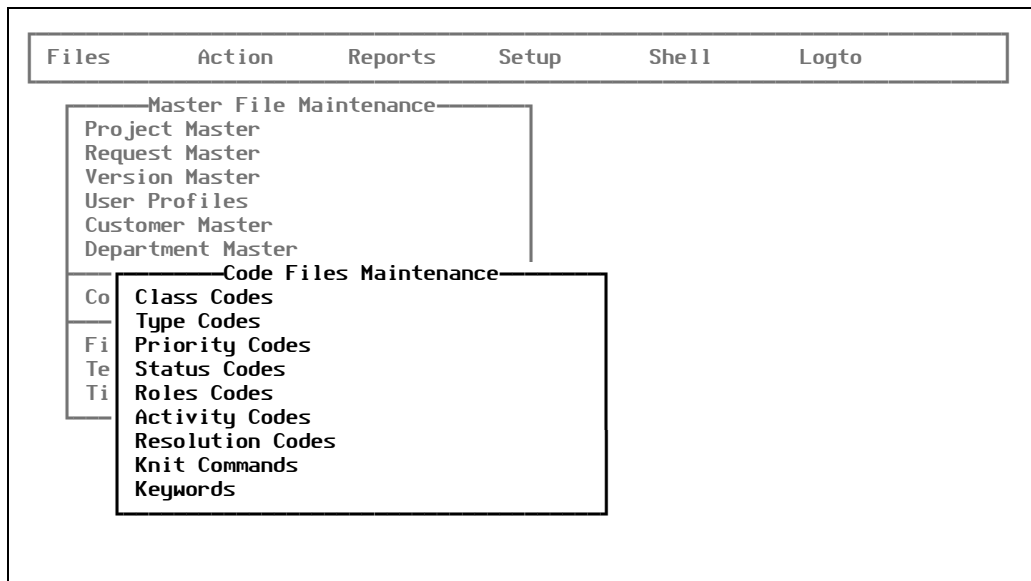


Figure 1-2: Code Files Maintenance submenu

The third type of file is what you might consider to be transaction files in any other type of application. These are the files that store the ongoing activity: what items are checked out to what project, backup copies of items, notes that programmers enter, the “stack” of what each user does each day.

Each of these files are described in [Table 1-1 on page 18](#) and explained in the subsequent sections.

Table 1-1: Codes Details and Special Functions (1 of 4)

Code or Special Function	Detailed Description
Account	<p>The name of the Pick/Universe/UniData “grouping.” Somewhat aligns with a directory in standard Unix and NT lexicon, but is a special-purpose directory in these environments.</p> <p>IMPORTANT: If you create a single, unified name for each real account of your software, then name your actual directories with the realm as a prefix or a suffix. PRC can build the name for you, which means a little less data entry up front and a tidier format overall.</p> <p>For example, your main account is ACME. You can have actual accounts on the system called DEV.ACME, TEST.ACME and ACME. In this case, you only have to tell PRC about ACME. If you decide to name the ACME development account CHEESEDODDLE, you can; but, you must be sure to enter an ACCOUNT entry in PRC for each and every distinct physical account.</p>

Table 1-1: Codes Details and Special Functions (2 of 4)

Code or Special Function	Detailed Description
Activity	<p>This code file supports codes that are used in two ways. There is a <i>Type</i> field on the Activity Code File screen that lets you designate whether you are entering “action” codes or “job” codes.</p> <ul style="list-style-type: none"> • Job codes are for posting time against projects. Examples of job codes might be <i>SPEC</i>, <i>CODE</i>, <i>DOC</i>, and <i>TEST</i>. • Action codes are used in the “next scheduled action” section of the Request System. These codes are sort of “events” that occur against a call or report from the customer. Examples are <i>CALL-IN</i>, <i>CALL-BACK</i>, <i>WAIT.4CUSTOMER</i>, <i>WAIT.4PROGRAMMING</i>, <i>WAIT.4APPROVAL</i>, <i>OPEN.PROJECT</i>, and <i>CALL.ENGINEERING</i>. They represent things that you might do—or plan to do—on a call. Some companies use these to get metrics on how long it takes to resolve a call, and how much of that time was active effort as opposed to elapsed time.
Class	<p>In an SB+ environment, there must be at least one class for each SB+ SYSID, named the same as the SB+ SYSID.</p> <p>For non-SB+ users on PRC Release 3 and 4, one SB+ SYSID was created for you and it must be listed as a class.</p> <p>Note: A SYSID can be further broken up into as many classes as are desired.</p>
Customers	<p>Customer codes exist so you know for whom you are doing the work on the project or from whom the request came.</p> <p>If this is not interesting to your organization, simply add Customer 1 for the company name and let all projects and/or requests be for that customer.</p> <p>If you are not interested in keeping track of the individual, you might enter departments as your customers. However you prefer to track for whom the work is being accomplished—or who needs to get an answer.</p>
Departments	<p>If you enter department codes, you can be more specific when you enter customers by indicating which department needs the work.</p> <p>This field is used in slightly different ways at some sites. Be creative.</p>
Priority	<p>Establish priority codes for projects and requests. Some organizations use 1, 2, 3 while others use High, Medium, Low. Some companies have 10 or more. You should use whatever works for you.</p> <p>One priority code is special: Priority Code 00 (zero zero), which is programmed to have special behavior. In the event of a conflict, this priority causes a “priority swap” of an item’s branch copy. (If this does not make sense to you now, come back to it later.) The best way to explain the problem it solves is through an example:</p> <p>Sam is making intense changes to a major program in the system and he has broken it up into several pieces. Meanwhile, Sally needs to make a very important—yet very minor—fix to the program and must get the change to LIVE right away. She does not have the time to wait for Sam to finish; she uses priority code 00.</p>

Table 1-1: Codes Details and Special Functions (3 of 4)

Code or Special Function	Detailed Description
Realm	<p>Realms are the designated “areas” of your environment. The most common realms (very common) are: DEV, TEST, and LIVE; or DEV, QA, and PROD.</p> <p>You can have as many as you want and you can name them whatever you want. Multiple modules (SB+ SYSIDs and accounts)—even of varied software—can reside under one realm, but the rules of behavior will be the same across all software defined under a given realm.</p> <p>The “rules of behavior” primarily dictate:</p> <ul style="list-style-type: none"> • Who has what privileges in an area • From where other software can be delivered to this area • To where software from this area can be delivered <p>You can get a little more esoteric and have synonym realms to cause different sets of rules on the same physical area.</p>
Resolution	<p>These codes are used by requests, not projects.</p> <p>Once you open a project, it is “resolved” by going through the life-cycle states (see Status Codes, below) and eventually completed and moved to LIVE (or cancelled). However, requests do not always become projects and might be resolved in other ways; for example, “explained to the user,” “sacrificed the user,” “sent manual,” or “declined.”</p>
Role	<p>Roles define certain additional options. Primarily, you use these to define “testers” who can access a couple of screens that programmers do not. Other reasons may come up that require identifying roles and classifying individuals as authorized in certain roles (programmer, manager, tester). At this time, the only role with special behavior inherent in the software is tester. You can use whatever word you choose to identify the name of the “testing role” in your organization.</p>
Station	<p>For environments in which development is accomplished on two separate machines with two separate copies of PRC, this identifier is used to help realign the software. You give each station an ID: 1, 2, etc., or some cute theme (characters played by Mike Myers).</p> <p>The primary function of this code is to make the PROJECT KEYS unique. The station, when present, will be included in the project key. As such, project 1 on “Wayne” will not be confused with—or overwritten by—project 1 on “Austin.”</p>
Status	<p>These codes are central to the DLM process. They control what can be rolled out, what can be checked out, who can do what and when. The first four status codes are hardcoded, although the first two are frequently skipped in some shops:</p> <ul style="list-style-type: none"> • 1 — requested • 2 — reviewed / approved • 3 — actively working / assigned to programmer • 4 — development complete (hasn’t rolled anywhere or been tested by anyone else, but the programmer says it is finished) <p>After this the status codes get a lot more flexible. A common configuration in a simple environment is:</p> <ul style="list-style-type: none"> • 5 — moved to test • 6 — tested • 7 — moved to LIVE
Suite	<p>Suites are used for variations in the software, most commonly for spoken language translations. They enable PRC to understand that under the DEV realm, there might be FRENCH.ACME, SPANISH.ACME, and so on or the variations could be platform variations or flavor differences in your software. Usually, it is used for either multi-lingual support or not at all.</p>

Table 1-1: Codes Details and Special Functions (4 of 4)

Code or Special Function	Detailed Description
System	These codes are SB+ system identifiers (SYSIDs). Even if you are not developing in SB+, one overall SB+ SYSID will have been created for your environment and must be nominated here.
Type	<p>These codes have the most varied use amongst the codes explained here. The most common use for a “type” code is to separate bugs from enhancements (for example, SJ+ has two codes: BUG, ENH). Some companies use them for billable and non-billable. Other companies use them to keep track of SCREEN changes, UPDATE changes, and REPORT changes. One company uses this for HARD and EASY. Yet another company has codes that indicate the expected number of hours of effort (<10, 10-20, 20-40, 40-80, and 80+) then use the “class” code to indicate the number of hours per month that would be saved by this project being done and then a ratio of these two numbers is used to generate the priority. (Good idea, huh?)</p> <p>Type codes can be used to indicate whether sub-projects conflict and other various behaviors throughout PRC can be type-specific.</p>
Users	<p>Each user in PRC must be a unique SB+ user ID. Many profile preference settings exist to fine-tune a user ID, but mainly user IDs are set up according to the user’s security clearance in each realm.</p> <p>You want to think about who are all of the users PRC must recognize. Then, for each realm, whether they can log in, make changes in general, and can make changes to items already checked out.</p> <p>You can enter a profile called DEFAULT and/or a default profile for each SB+ security group called DEFAULT*GROUPNAME. For each person who logs in to a PRC-enabled account and does not already have a profile, the system uses the default profile created for him/her.</p> <p>It is important to note that changes to the default profile would only affect <i>new</i> users, so if you want to use this feature to make a mass profile change, you must first clear the PRC .USR file of all profiles (except the defaults).</p>
Version	<p>PRC requires a beginning version: pick one. Afterwards, you can perform incremental version rollouts that follow a numbered scheme. PRC will keep the numbers in sequence; that is, if you begin with 8.0 and roll it out, PRC will open 8.1 next.</p> <p>If you never roll out a numbered version, that first version will exist for all time. The versions, incidentally, can be used to roll out any “collection” of projects, with version names of any kind.</p>

PRC Code Files

Project Classes & Class Types

Projects can fall into several categories. Those categories are called “classes,” which are represented by class codes.

Class codes help you:

- Categorize projects
- Relate projects to the location of the source components—realm, account, and system

Classes are subdivided into class types. All class types in PRC are pre-defined:

- **TSB** — Tracked
All projects to which you expect to make software changes.
- **M** — Maintenance
Some companies open projects for functions that have nothing to do with software changes. These projects (like network issues or even copier maintenance) would never otherwise have items tracked against them.
- **P** — Perpetual (conflict)
This is the one thing you must always remember: this is a type of project that remains open—it never closes. Items checked out to a **P** project do not conflict with other projects.
- **PNC** — Perpetual (non-conflict)
Items checked out to a **PNC** project will conflict with other projects if those projects attempt to attach these items.
- **R** — Resources required
Projects that will require certain resources; for example, a certain individual, a touch screen, a particular computer or server, or anything along those lines.
- **A** — Analysis
Projects that will not have anything but time charged against them.

Class Codes

If a certain project class covers software creation or modification—as opposed to general areas that MIS supports but does not develop—the class code file must contain realm, account, and system information.

They are often used to break up modules. The same class of project can be used for changes in the one account and system, but across multiple realms. This is why you can enter more than one valid realm on the Class Code Maintenance screen.

The class code is more than just a way to break the software into modules; it also identifies the physical location of the software that belongs to that class. For example, all project classes that are **TSB**, meaning they will have software tracked against them, must have the realm, account, and system ID specified.

To use class codes this way, the realm prefix/suffix option must be used (see [“System Master” on page 146](#)) so that PRC can make certain assumptions. Classes of projects may exist that do not require software change, thus would not require account and system locations. The optional “types” of class codes are explained in [“Type Codes” on page 24](#).

The system uses classifications for allocating resources and reporting. You define class codes in the classification file `PRC.CLASS`.

There must be a class code named for each SB+ system ID.

To work with class codes, perform the following steps:

1. Select **Files > Code Files > Class Codes**.

The system displays the Class Code Maintenance screen, similar to the one shown in Figure 1-3:

Class Code Maintenance	
Class Code	Description
AP	Accounts Payable
Sub-class of	
Resource Responsible	JOHN
Tracking Type	TSB Tracked (SB+ Sys)
Used for projects in: REALM(s)	Location, this class of software:
DEV	Account Must match? Yes
	PRC6 PRC6
	System
	PRC

F2-Save F4-Del F10-Action

Figure 1-3: Class Code Maintenance screen

Explanations of the fields are as follows:

- Class Code** The code or ID that represents this project classification.

Press **[F3]** to select from the list of valid entries.
- Description** A concise but meaningful description of this project class.
- Sub-class of** If this project class is a sub-classification of another class, indicate the parent classification in this field.
- Realm** The default realm for work performed on this class of projects. You can override this value on the specific project. Items are described by the realm/account/system combination in the SB world.

Press **[F3]** to select from the list of valid entries.
- Account** The Pick/Universe/UniData account to which projects of this class default. Items are described by the realm/account/system combination in the SB world.

PRC will display the account name alone—without the realm prefix or suffix—if you choose that option on the System screen. For more information, see [“Realm Master” on page 131](#).

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

Must match?	Indicates whether the account in which the project is being worked must match the value of the <i>Account</i> field. Valid entries are Y (Yes) and N (No). Press F3 to select from the list of valid entries.
System	The name of the SB+ system, if applicable. The value of this field may be—and often is—the same as the <i>Account</i> field. Items are described by the realm/account/system combination in the SB world. Press F3 to select from the list of valid entries.
Resource	The computer, printer, or individual required to see or address this class of project. Usually pertains to hardware. Indicating a resource associated with this class of project enables you to produce reports (for example, backlog reports) by that particular resource. This feature is useful for evaluating the need for a new resource or for cost analysis of an existing resource.
Responsible	The user ID of the individual who is responsible for this class of project or request. Press F3 to select from the list of valid entries.
Tracking Type	Indicates the type of project or request classification.

2. Fill in each field with the appropriate information.
3. Press **F2** (Save) to save this class code.

Type Codes

PRC breaks down projects by type code for reporting, searching, and security purposes. Types are a second, less-structured opportunity to categorize projects.

Type codes, in addition to class codes, provide you with two ways of differentiating projects:

- *Classes* determine whether software changes will be made and where they will be made, and is frequently a module code.

For example, one class of project could be **AR** (Accounts Receivable) and another class could be **IM** (Inventory Management).

- *Types* can be associated with what life-cycle states the project goes through and who can perform tasks at each state.

For example, one class type could be **BUG** (bug or issue) and another type could be **ENH** (enhancement).

You can create finer detail for an individual's authorization by using project types. For example, you can define a junior programmer with the authorization to work on bugs, but not new development.

Therefore, projects can differ by classification and type. An AR BUG project is different from an IM BUG project. At the same time, an AR BUG project is different from an AR ENH project.

You can set up the Type file with codes and descriptions only. Many reports select or sort by project type. Some examples of type codes are delivered with PRC and may be used, added to, or replaced. Use **F3** to review these suggested type codes such as "Bug," "Development," "Enhance," "Billable."

You must have at least one type code on file. Even if your organization wishes to delete type codes, the system will not operate properly without at least one type code defined.

To work with type codes, perform the following:

1. Select **Files > Code Files > Type Codes**.

The system displays the PRC Type Code Maintenance screen, similar to the one shown in [Figure 1-4](#):

PRC Type Code Maintenance

Type Code NEW	Used on doc Type(s)
	PRJ Project
	REQ Request
	VER Version

Description
New Development

For "DATA" type only:
 Insert Remark lines? Yes
 Close Status _____

Additional Information

F2-Save F4-Del F5-Stat/Role F10-Action

Figure 1-4: PRC Type Code Maintenance screen

Explanations of the fields are as follows:

Type Code

The code or ID that represents this classification of project or request. Some type codes that are delivered with PRC as suggestions include:

- **BUG** — Bug
- **DATA** — Data edit
- **ENH** — Enhancement
- **MAINT** — Maintenance, non-software

Type Code (cont)

- **NEW** — New development

- **C/S** — PSI Customer Service forms
- **RB** — RedBack

Press **F3** to select from the list of valid entries.

Used on Doc Types Indicates the kinds of documents that can be classified with this type code. You can choose one or more of the following:

- **CUST** — Customer
- **MASTER** — Master
- **PRJ** — Project
- **REQ** — Request
- **TEST** — Test plan
- **VER** — Version

Press **F3** to select from the list of valid entries.

Description A concise but meaningful description of this type code.

Insert Remark lines? Indicates whether a remark is inserted into programs. Valid entries are **Y** (Yes) and **N** (No). This field works in conjunction with the *Close Status (DATA)* field.

For example, you choose to have a particular project type *not* put remarks into an otherwise “remarked” file when the type is for emergency *DATA* projects.

Press **F3** to select from the list of valid entries.

Close Status (DATA) The status to which the *DATA* project will bump. Items are checked in and the project is closed when this type of project is bumped.

Press **F3** to select from the list of valid entries.

Additional Information Additional information that must be captured with projects of this type. This field is not displayed anywhere or do anything within *PRC*.

Some companies create similar code types, such as *BUG* and *CRITICAL . BUG*, that need further explanation. Using the example, a company uses this field to elaborate about how something is only a critical bug when certain criteria have been met.

2. Fill in each field with the appropriate information.
3. Press **F5** (Stat/Role) to define states and roles.

The system displays the State/Roles for Type sub-screen, similar to the one shown in Figure 1-4:

```

PRC Type Code Maintenance
Used on doc Type(s)
Type Code NEW          PRJ Project
                       REQ Request
                       VER Version

Description
New Development

Insert Remark lines? Yes
Close Status (DATA)

Additional Information

States/Roles for Type
State      Role
3 Active & Assign
_____

F2-Save F4-Del F10-Action

```

Figure 1-5: States/Roles for Type sub-screen

Explanations of the fields are as follows:

State

The pre-defined states through which this type of project will go. Also, you determine the order of the states.

Press **F3** to select from the list of valid entries.

Role

The role that is authorized at this state. Users can attach or assign themselves to a project only if the status is set up for a role to which they are authorized. For more information, see the *Software Quality and Testing Guide*.

Some suggested entries are:

- **PROGRAMMER** — Developer/programmer
- **ASKED** — Asked for it
- **CLASS** — Individual responsible for this type's class
- **DEV** — Programmer/developer
- **QA** — Quality Assurance tester
- **QC** — Quality Control tester
- **~** — Reserved: Tilde followed by a specific user ID
- **TEST** — Testing

Press **F3** to select from the list of valid entries.

Role (cont)

The reserved entries are:

- **CUSTOMER** — Reserved: Customer on the project will be notified
- **ASSIGNED** — Reserved: Programmer assigned (status **3**) will be notified

4. Press **F2** (Save).

The system returns you to the PRC Type Code Maintenance screen (Figure 1-4 on page 25).

5. Press **F2** (Save) to save this type code.

Priority Codes

Priority codes establish a hierarchy of importance between projects. In some cases, this priority is useful just for reporting, but occasionally priorities determine other characteristics. There are two opportunities to use priority:

- One for the user's priority
- The other to identify MIS' priority in the grander scheme of things

To work with Priority codes, perform the following:

1. Select **Files > Code Files > Priority Codes**.

The system displays the Priority Code Maintenance screen, similar to the one shown in Figure 1-6:

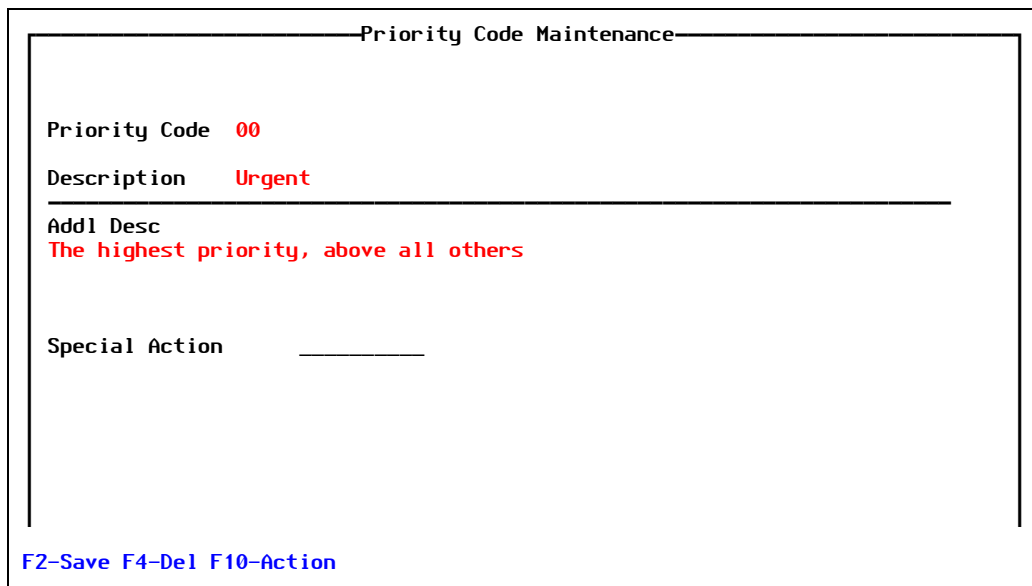


Figure 1-6: Priority Code Maintenance screen

Explanations of the fields are as follows:

Priority Code	The code or ID that represents this project priority. The only pre-defined entry is 00 (Urgent). It creates a lock on all software components checked out to it. No one can change or create a conflict with anything checked out to a priority 00 . Press F3 to select from the list of valid entries.
Description	A concise but meaningful description of this priority code.
Addl Desc	[<i>Optional</i>] Additional details about this priority code.
Special Action	Identifies a process that runs or a manual operation that is triggered (such as e-mail) when a project with this priority is opened or a project is changed to this priority. For example, if you enter <code>E.MAIL,mailto:person</code> in this field, that person or role will receive an e-mail automatically whenever a project is created with or changed to this priority.

2. Fill in each field with the appropriate information.
3. Press **F2** (Save) to save this priority code.

Status Codes

Status codes are the most important of all of the codes in PRC. They define the development life-cycle.

PRC expects projects to move through numbered life-cycle states. The first four states are absolute in PRC. They are:

- **1** — Requested
- **2** — Reviewed/Approved
- **3** — Active/Assigned
- **4** — Development Complete

Beyond these four, you can define any number of states. A typical and simple implementation includes: **5** (Moved to `TEST`), and **6** (Moved to `LIVE`). You can add other numeric life-cycle states plus codes for things like cancelling, closing, placing on hold.

You can also use Alpha codes; however, no hierarchy or life-cycle is defined by Alpha codes. You would use them for closing out projects, cancelling projects, or perhaps placing projects on hold.

Two Alpha states are predefined with special behavior.

Special codes defined in PRC are:

- **c** — (begin with “C”) will ask, then check-in software still checked out to the project
- **x** — (begin with “X”) will ask, then not only check-in the software but optionally replace it to its original state

To work with status codes, perform the following:

1. Select **Files > Code Files > Status Codes**.

The system displays the Status Code Maintenance screen, similar to the one shown in [Figure 1-7](#):

Status Code Maintenance			
Status Code	4	Document Type	Amend? Bump?
Description	Dev Complete	PRJ Project	Yes Yes
		REQ Request	Yes No
—Change to this state—		Roles (that may assign to this state)	
From Status	Authorized		
3	ROOT		
Chg Req Stat to:		Escalate in days to status	
—Email Notification—			
Type	Send Mail To	Mail Text	Mail Attach
ALL	ADMIN		
Live	Event triggers for		
Hist Confl	Change status		
Checked In	Bump Status		
F2-Save F4-Del F5-*Drill F8-Triggers F10-Action			

Figure 1-7: Status Code Maintenance screen

Explanations of the fields are as follows:

Status Code

A code or ID that represents meaningful “states” or “life-cycles” for your software versions, projects, and requests. Default, pre-defined values are:

- **1** — Requested
- **2** — Approved
- **3** — Active & Assigned
- **4** — Development Complete
- **5** — Moved to QC (Quality Control)
- **6** — Passed QC

Other available values include:

- **7** — Moved to LIVE
- **A** — Accepted (User)
- **c** — Closed/Checked-In

Status Code (cont)	Special codes that begin with C will ask whether you wish to check in (release) the software, then check in software still checked out to the project.
	Other available values also include:
	<ul style="list-style-type: none"> • O — Open (Request) • R — Resolved • T — Tested • V — Converted (Request to Project) • X — Cancelled
	Special codes that begin with X will ask whether you wish to check in (release) the software, then not only check in the software but optionally replace it to its original state.
	You can identify life-cycle states by project type.
	Press F3 to select from the list of valid entries.
Description	A concise but meaningful description of this status code.
Document Type	The kind of item that uses this status. Valid entries are:
	<ul style="list-style-type: none"> • PRJ — Project • REQ — Request • VER — Version • Any combination of the above values
	For example, if a status code is to be used on all three types of documents, put all three in this field column.
	This “document type” field, common to many of the PRC support code files, indicates the document type or types on which this status code may be used.
	Press F3 to select from the list of valid entries.
Amend?	Indicates whether documents of this type can be amended when they are in this state. Valid entries are Y (Yes) and N (No).
	Press F3 to select from the list of valid entries.
Bump?	Indicates whether documents of this type can be “bumped” when they are in this state. Valid entries are Y (Yes) and N (No).
	Press F3 to select from the list of valid entries.
From Status	Indicates the previous status of the project, request, or version before it became this particular status.
Authorized	Establishes <i>who can change</i> a project to this state.

For example, if you are defining status **4**, you would typically enter here who can change it from **3** and who can change it from **5**.

Roles	<p>Identifies the user roles that may work on a project in this state.</p> <p>Press F3 to select from the list of valid entries.</p>
Chg Req Stat to	<p>Identifies the status code to which a request or event should be set when the PRC project reaches this status.</p> <p>Press F3 to select from the list of valid entries.</p>
Escalate in days to status	<p>The number of days a project can remain in this status.</p> <p>The status to which the project must “escalate” or close.</p>
Type	<p>The project type code, which will send e-mail to different recipients based on the project type.</p> <p>Use ALL if the e-mail recipient will be the same regardless of project type.</p> <p>Press F3 to select from the list of valid entries.</p>
Send Mail To	<p>Indicates who should receive an e-mail notification when a project arrives at this status. You can enter multiple recipients, separated by semicolons. Valid entries are:</p> <ul style="list-style-type: none"> • ~<user.id> — Routes an e-mail to a specific individual when the project reaches this state. <p>Use a specific user ID prefixed with the tilde symbol (~), just as it is defined in the SB+ security file.</p> • CLASS — Routes an e-mail to the individual specified on the class code in the <i>Responsible</i> field when the project reaches this state. • ASSIGNED — Routes an e-mail to the individual assigned at this status on this particular project. For example, when the project changes to status 5, the person on the project screen in the status 5 slot will receive the e-mail. <p>Be sure you pre-assign the project at each state if you wish to use this option.</p> • ROLE — Indicates any valid role associated with a state. (See “Roles Codes” on page 36.) <p>This can be a little confusing. The e-mail will go to the person associated to the state that is associated to the <i>role</i>, which is not necessarily the state to which the project is changing.</p> <p>For example, when the project is promoted to LIVE (status 7, for example), e-mail the PROGRAMMER role, which is defined as associated to status 3 on the project.</p>

For more information, see the *Software Quality and Testing Guide*.

Press **F3** to select from the list of valid entries.

Mail Text

You can enter specific text or select the name of a text item in a directory file named `PRC.MAILTEXT`.

If no specific text is specified here, PRC inserts the project ID in the Subject line. It also displays, in the body of the message: the project ID (again), the brief description from the project, and the statement that "The referenced project has moved from state x to state y."

Mail Attach

The item to be attached.

Press **F3** to select from the list of valid entries.

Live

Indicates whether this status is the main "live" status. Valid entries are **Y** (Yes) and **N** (No).

Hist Confl

Indicates whether projects are included in an historical conflict scan. Valid entries are **Y** (Yes, the default) and **N** (No).

This field is mainly for Epicor Avante users and works in conjunction with status code **C**.

Background: Projects are in active conflict while they are "open" and the items connected to them are "checked out." Once a project is "checked in," items do not show up in a regular "active conflict scan," although the users are interested in seeing these as conflicts during an "historical scan;" for example, when bringing in a release upgrade or work done off site.

Here on the status definition, this field should normally be set to **Y** (Yes). It is set to **N** (No) so that a project can be marked (typically closed) to this status when it is "checked in forever" and should not even be reviewed during an historical conflict scan; for example, a change is made on site, then a better change comes in with a critical release, such that the original "quick fix" was superseded and is simply not relevant to future upgrades.

Press **F3** to select from the list of valid entries.

Checked In

Answers the question, "Is this a state that would normally be associated with software being checked in?"

Note: Do not change to this state manually. Manually changing status or manually checking in is dangerous.

Only two methods of checking in or releasing software that as been checked out to a project exist in PRC:

Checked In (cont)

- Good: Roll out the project to the realm that has been designated to check in
— or —
- Bad: Manually change the status of a project to C (or just press **F4** (Delete) or **X** to cancel the project)

For example, when you set up the rollout to LIVE and indicate on the Realm screen that the project should go to status **7** and be checked in, you may come to expect a project in status **7** to be checked in; however, manually changing the status to **7** will *NOT* check in the software.

Because many organizations have given in to this temptation in the past, it became necessary to change the logic within PRC. As you are defining the rollouts that do check-ins, PRC writes a marker on the status code.

This logic allows you set up the roll out from DEV to LIVE, to change the project to status **7**, and to check it in; therefore, PRC marks status **7** as a “check in” status. So if you do manually change a project’s status to **7**, the system will pop up a box that reads something like:

```
You are changing this projects status to a status that
would normally be checked in, changing the status
manually here is NOT GOING TO CHECK IT IN.
```

Change status

The process or B: (BASIC) subroutine that will be called when a project is manually changed to this status in the Project Master screen before the project is written.

Bump status

The process or B: (BASIC) subroutine that will run *before* the bump is performed and can stop the bump by sending back `RTN.FLAG = 1`.

The bump from status **3** to **4** already performs special functions. You can customize it further via the *At Bump* field on the Processes in Slots sub-screen. For more information, see [“Process Slots” on page 184](#).

2. Fill in each field with the appropriate information.
 - If you press **F5** (*Drill), see [“Drill Down”](#) below.
 - If you press **F8** (Triggers), see [“Triggers” on page 35](#).
3. Press **F2** (Save) to save this status code.

Drill Down

From any field on the Status Code Maintenance screen ([Figure 1-7 on page 30](#)):

1. Press **F5** (*Drill).

The system displays the sub-screen in [Figure 1-8](#):

Status Code Maintenance			
Status Code	4	Document Type	Amend? Bump?
Description		PRJ Project	Yes Yes
Dev Complete		REQ Request	Yes No
—Change to this state—		Roles (that may assign to this state)	
From Status	Authorized		
3			
Chg Req Sta		te in days to status	
Type	Send	on	
ALL	ADMI	Mail Attach	
Live		Event triggers for	
Hist Conf1		Change status	
Checked In		Bump Status	

F2-Save F4-Del

Figure 1-8: Status Code Maintenance drill-down sub-screen

Use this sub-screen to indicate that more than one person (or group or role) is to be authorized for this particular status change. When more than one entry is listed, the main page will display the asterisk (to remind that a drill-down is available) and a count.

The example in [Figure 1-8](#) shows **ROOT** on the main page is because that is the only authorized “entity” to change a project's status to **3**.

2. Press **F2** (Save) to save the values on this sub-screen.

The system returns you to the Status Code Maintenance screen ([Figure 1-7 on page 30](#)).

Triggers

From any field on the Status Code Maintenance screen ([Figure 1-7 on page 30](#)):

1. Press **F8** (Triggers).

The system displays the Triggers from Status Change sub-screen, as shown in [Figure 1-9](#):

The system displays the Role Code Maintenance screen, similar to the one shown in Figure 1-10:

Role Code Maintenance

Role Code PROGRAMMER	Associated States
	3
	4

Description
Programmer

Additional Information
Not for college interns, who should be assigned the INTERN role.

F2-Save F4-Del F10-Action

Figure 1-10: Role Code Maintenance screen

Explanations of the fields are as follows:

Role Code

The code or ID that represents activities an individual will perform in conjunction with different types of projects and project states.

A role is similar to a job description or job title, in that how an individual relates to a project is implied. The role of an individual prescribes their duties associated to a particular life-cycle state.

Press **F3** to select from the list of valid entries.

States

The project state associated with this role. For example, the role PROGRAMMER is associated with the project state **Active & Assigned**.

Press **F3** to select from the list of valid entries.

Description

A concise but meaningful description of this role code.

Additional Information

Further descriptive information others should know about this role.

2. Fill in each field with the appropriate information.
3. Press **F2** (Save) to save this role code.

Activities (Time)

Activities (job codes) are used to post time against projects and requests. You can use activity codes to also indicate billing information—but this information is optional.

Activities have two primary purposes:

- **J** (Job service) activity codes are used to identify how time was spent. In some cases this will be used for billing purposes.

For example, the activities will be things that people do related to a project such as “write code” or “write spec” or “unit test.”

- **A** (Action) activity codes are used to identify the next action or “who has the ball” for a customer request. (See **F10** (Action) from Request Master screen.)

For example, the activities might include “awaiting information from user” or “spec due” or something similar.

At least one activity code is required to use timesheets.

To work with activity codes, perform the following steps:

1. Select **Files > Code Files > Activities (Time)**.

The system displays the Activity Code Maintenance screen, similar to the one shown in [Figure 1-11](#):

Activity Code Maintenance

Activity Code	Title		
WC	Write Code _____		
Type Code	Unit	Accum	Status
J Job/Service	HR Hou	Yes	3 Active & Assigned
Class	Resource		
Bill Rate	Cost	Bill As	
40.00		Develop	
Complete Description (optional)			
Writing code is a portion of the "Develop" process, which also includes testing, rewriting, and documenting.			

F2-Save F4-Del F10-Action

Figure 1-11: Activity Code Maintenance screen

Explanations of the fields are as follows:

Activity Code The code or ID that represents a type of billable or non-billable work.

Press **F3** to select from the list of valid entries. Applies only after you have defined at least one entry.

Title	Brief title of the activity.
Type	<p>The category in which the activity falls. Valid entries are:</p> <ul style="list-style-type: none"> • J — Job/Service codes Used to identify how time was spent, which can be used for billing purposes. • A — Action codes Used to identify the next action or who is responsible for a customer request. • I — Inventory Used in special situations. If interested in this category, request more information from SJ+. <p>Press F3 to select from the list of valid entries.</p>
Unit	<p>The measure in which this activity is accomplished or billed. Valid entries are:</p> <ul style="list-style-type: none"> • HR — Hour • MO — Month • DY — Day • QR — Quarter • YR — Year • EA — Each • F — Flat • I — Item <p>Press F3 to select from the list of valid entries.</p>
Accum	<p>Indicates whether time spent on this activity should be added into the accumulated total of time spent on the request. Valid entries are Y (Yes) and N (No).</p> <p>Press F3 to select from the list of valid entries.</p>
Status	The project status against which this activity accumulates time. Allows you to report time by status in project reports.
Class	Project class to which this activity is associated or restricted. If you leave this field is blank, this activity can be applied to all classes.

Resource	A limited resource on which this billing code or labor rate is dependent. Can represent a particular computer, individual, or other entity.
Bill Rate	The monetary rate at which this activity will be billed, if applicable.
Cost	The cost associated with this billing code.
Bill As	Indicates a “parent” activity code to which this activity will be billed.
Complete Description (optional)	A fuller description or more detailed explanation of the activity.

2. Fill in each field with the appropriate information.
3. Press **F2** (Save) to save this activity code.

Resolution Codes

Resolution codes identify reasons or actions related to closing a request—in other words, problem reports. These codes are particularly useful for reporting.

Resolutions are not used on projects. A simple way to remember the difference is:

- Requests are resolved
- Projects are completed

The resolution of a request can:

- Require no action on the part of development
- Involve the request turning into a project
- Other scenarios

For more information about requests, see [“Request Master” on page 53](#).

To work with resolution codes, perform the following:

1. Select **Files > Code Files > Resolution Codes**.

The system displays the Resolution Code Maintenance screen, similar to the one shown in [Figure 1-12](#):

Resolution Code Maintenance

Resolution Code **PRJ**

Description
Became a project

Change
REQ Status to **V** **Converted (Req to Pr.j)**

F2-Save F4-Del F10-Action

Figure 1-12: Resolution Code Maintenance screen

Explanations of the fields are as follows:

Resolution Code	The code or ID that identifies how a user problem report is resolved. Press F3 to select from the list of valid entries. Applies only after you have defined at least one entry.
Description	A concise but meaningful description of the resolution.
REQ Status	The status code to which the request should be changed when this resolution code is used. Press F3 to select from the list of valid entries.

2. Fill in each field with the appropriate information.
3. Press **F2** (Save) to save this resolution code.

Knit Commands

KNIT is a split-screen editing and merging utility that is part of PRC. It is called throughout PRC to display differences between an item and a:

- Backup copy of that item
— or —
- Branch copy of that item

The KNIT screen can also be called directly by typing **/KNIT** from any prompt. When you use the slash command to call up KNIT, the system displays a setup or selection screen that allows you to enter the items to be compared on the left and right side.

You can customize these functions in terms of the display prompt and input to be used. In other words, you can make this merge/compare utility more like another tool with which you are familiar. You can make KNIT more intuitive to you, personally.

WARNING

Do not change or modify the **Standard** KNIT set that is shipped with PRC. Create and modify your custom sets.

To work with KNIT commands, perform the following:

1. Select **Files > Code Files > Knit Commands**.

The system displays the Knit Command Maintenance screen, similar to the one shown in [Figure 1-13](#):

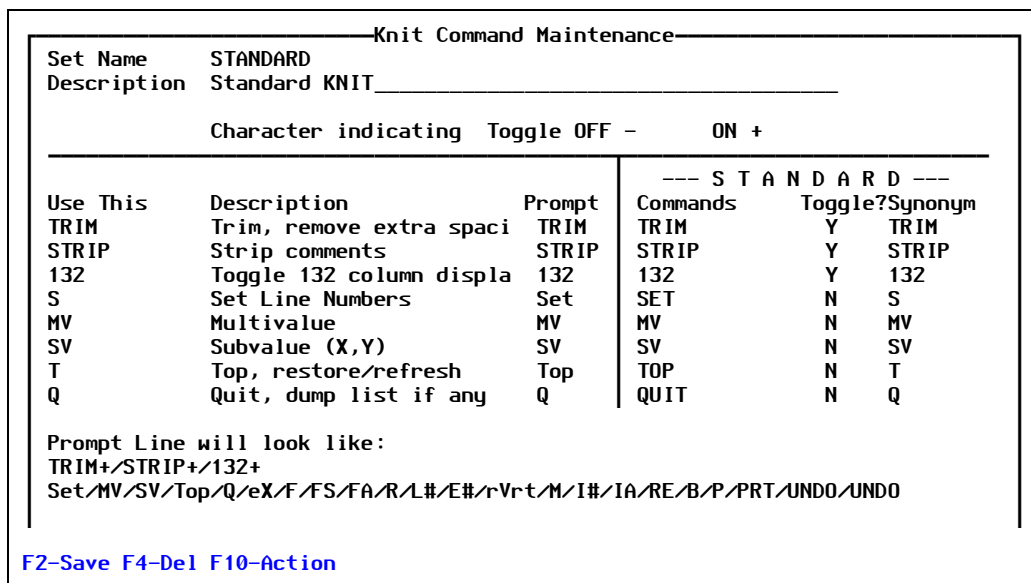


Figure 1-13: Knit Command Maintenance screen

Explanations of the fields are as follows:

Set Name The code or ID that represents a command set. You can modify existing command sets.

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

Description A concise but meaningful description of the command set, such as "Just like COMPARE.TOOLS."

Toggle OFF	A character or small set of characters to indicate that the state of a toggle type command is “on”—such as (on) or + or 1). Keep it as short as you can!
ON	Similar to the field above, a character or small set of characters to indicate an “off” state of a toggle type command—such as (off) or - or 0); for example, <code>Trim(off)</code> uses (off) .
Use This	The actual keystrokes needed to invoke this action or command; for example, if you are modify the command <code>PRINT</code> , you could use P or even OUT . Each value in this window must be unique.
Description	Any description that makes sense to you. Be aware that you may be modifying a command set that others use! Press [F3] to select from the list of valid entries.
Prompt, Commands	Any additional prompts or commands you wish to define, thereby customizing KNIT for your organization. The most common use of these fields is to disable certain functions.

2. Fill in each field with the appropriate information.
3. Press **[F2]** (Save) to save this knit command set.

Keywords

If the Type, Class, Priority, and all the other codes simply are not enough for you to classify your projects, you can use *keywords*.

Keywords that you define are added to a library so you can use **[F3]** to look them up and maintain consistent spelling; otherwise, the system cannot perform validation.

Keywords simply provide you with a way to break your projects down further for searching. For example, some organizations enter “SCREEN” if it is a screen, and so on, or use some private code word.

To work with keywords, perform the following:

1. Select **Files > Code Files > Keywords**.

The system displays the Keyword Maintenance screen, similar to the one shown in [Figure 1-14](#):

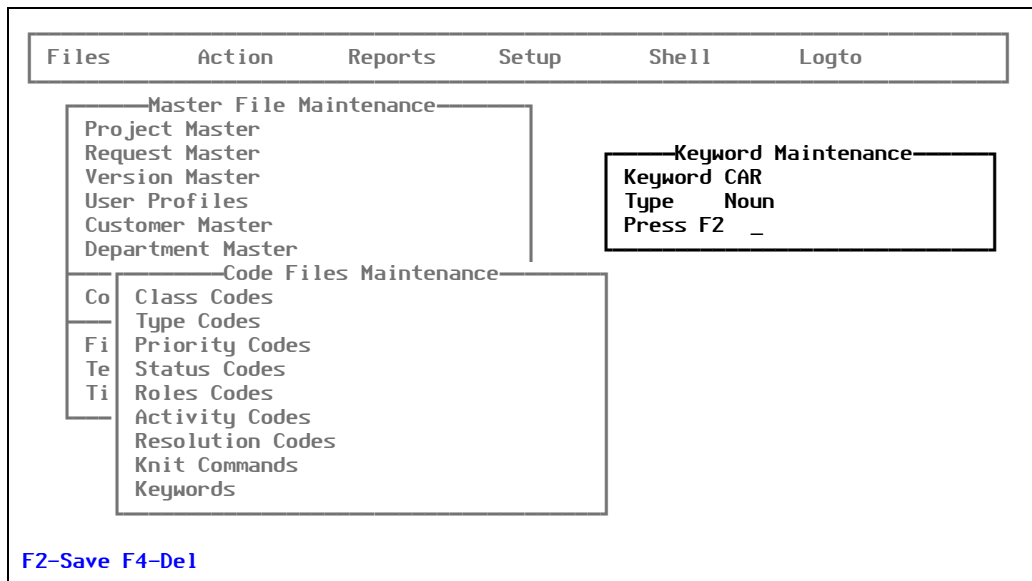


Figure 1-14: Keyword Maintenance screen

Explanations of the fields are as follows:

Keyword A meaningful keyword for flagging RFS by which you can sort later. Make the keyword a single word.

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

Type The kind of word being defined and/or how it will be used. It is not necessary at this time to define verbs, areas, or adjectives. Valid entries are:

- **V** — Verb
- **A** — Area
- **N** — Noun (Recommended)
- **J** — Adjective

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

Press F2 Frequently, fields such as this are placed on the screen to remind the user what to do next. It also makes a screen with only one prompt clearer to understand.

2. Fill in each field with the appropriate information.
3. Press **[F2]** (Save) to save this keyword.

PRC Master Files

Project Master

Projects are the core component of PRC; therefore, being able to work with the Project Master is essential. You use it to create and manage projects for all work that is planned or in development.

The Project Master is the central point of activity for the PRC system. In it, you:

- Create new projects (also called “software change documents”)
- Maintain or modify existing projects
- Bump the status of a project up or down
- Add technical notes and release notes
- Add a “cross-talk” diary

You can determine whether the requesting user can access the diary or not.

- Include estimates and target dates

If you do, actual time accumulated from the daily timesheets and actual dates in terms of status movement is reflected here.

- Work with source items included on the project:
 - Review
 - Add (check-out)
 - Remove (check-in or “flick” to other projects)
 - Scan for conflicts and other errors
 - Roll out from this screen

To work with the Project Master, perform the following steps:

1. Select **Files > Project Master**.

The system displays the Project Master Maintenance screen, similar to the one shown in [Figure 1-15](#):

Project Master Maintenance					
Project	Status 3	Active & Assigned		Prio 1	Rank
4					Keyword
Class CH	Car Hire		Realm DEV		
Customer 1	Default Custome	Type ENH	Enhancement		
Title/Brief Description					
A project to enhance Car Hire.					
Full Description					
*Ver	*Subs	*Target	Requested	09/01/2006	JOHN
1.0			Approved	09/01/2006	JOHN
*Technote	*Synopsis	*Last Talk	*Active & Assig	09/01/2006	JOHN 1
			Dev Complete		
			Moved to QC		
*Requests	0	*Items	Passed QC		
Reference		0	Moved to LIVE		

F2-Save F4-Del F5-*Drill F7-Testplan F8-Source F9-Stat F10-ACTION

Figure 1-15: Project Master Maintenance screen

Explanations of the fields are as follows:

Project The number or ID that represents the project to which you are charging any changes made to and/or time spent on the project.

Press **[F3]** for a list of existing projects.

Status The number that represents the current status of the project, according to your organization’s project life-cycle definitions.

The project status is automatically updated by entry of activity dates or by functions that change the project’s status, such as bumping, rolling out, or rolling back.

The status of a project controls many features throughout the project’s life-cycle. Some states—the numeric ones—are expected to progress and some behavior is specifically tied to the state, such as **3** for “Active and Assigned.” No changes can be made against a project unless it is in status **3**.

For more information on defining status codes, see “[Status Codes](#)” on page 29.

Press **[F3]** for a list of valid statuses.

Prio The number that represents the priority assigned to the project by the requesting department. Valid entries range from **1** (highest priority) through **9** (lowest priority).

For more information on defining priority codes, see “[Priority Codes](#)” on page 28.

Press **F3** for a list of valid entries.

Realm

The designation that represents 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.

Typically a development realm or an “import” realm.

Press **F3** for a list of valid entries.

Class

The code or ID that represents this project’s classification. You can define a classification to mean that work is accomplished on a particular computer, on a particular account/system, by a particular programmer, or any number of things.

For more information on defining class codes, see [“Class Codes” on page 22](#).

Press **F3** to select from the list of valid entries.

Suite

The code or ID that represents the suite within the application to which this project belongs.

Press **F3** to select from the list of valid entries.

Customer

The code or ID that represents the client or department requesting the project.

Press **F3** to select from the list of valid entries.

Type

The code or ID that represents the “type” of project. Breaking down projects into types is useful for reporting and managing. You can also use the project type to establish the process model (what cycle/roles are involved).

For more information on defining type codes, see [“Type Codes” on page 24](#).

Press **F3** to select from the list of valid entries.

Keyword

Special codes or words that identify and group this project in the future. You can attach any number of keywords to a project so you can retrieve and sort projects by keywords later on.

The keywords you add and define are stored in the `KEYWORD` file, which acts as a dictionary.

Press **F3** to select from the list of valid entries.

Title/Brief

Description	A succinct, one-line description of the project. This will be frequently displayed whenever someone sees this project on a list of projects.
Reference	The code or ID that represents a reference to VI#, MOD#, service request number, ticket number, Customer Service Form, or whatever is relevant to your organization.
Full Description	A full textual description of the project as requested by the user.
Orig	<p>The original date the project request was made. If the Request Management system is in use, this field represents the day the project was originally approved.</p> <p>Press F3 to select a date from the pop-up calendar.</p>
Req By	<p>The name of the person who requested the project.</p> <p>Press F3 to select from the list of valid entries.</p>
Review	<p>The date the project was reviewed by the project leader. It indicates the date the individual took responsibility for the project.</p> <p>Press F3 to select a date from the pop-up calendar.</p>
Reviewed	<p>The name of the project leader assuming responsibility for the project.</p> <p>Press F3 to select from the list of valid entries.</p>
Active	<p>The date this project was assigned to a programmer or technician. It indicates the date the project became “activate” (the clock is running, so to speak).</p> <p>Press F3 to select a date from the pop-up calendar.</p>
Assn	<p>The primary programmer or technician assigned to the project.</p> <p>Press F3 to select from the list of valid entries.</p>
Dev Cmpl	<p>The date the project was completed—from a development standpoint—and is authorized to move into the TEST realm.</p> <p>Press F3 to select a date from the pop-up calendar.</p>
Compl By	<p>The individual responsible for completing the project.</p> <p>Press F3 to select from the list of valid entries.</p>
Test	<p>The date that the completed project was moved into the TEST realm. If it is moved more than once, PRC displays the last date.</p>

	Press F3 to select a date from the pop-up calendar.
Tested	The programmer who moved the software for this project into the TEST realm. Press F3 to select from the list of valid entries.
Test Cmplt	The date that project testing was completed. Press F3 to select a date from the pop-up calendar.
Live	The date the project moved into production. Press F3 to select a date from the pop-up calendar.
Live By	The individual who moved the software on the project into the PRODUCTION realm. Press F3 to select from the list of valid entries.
Ver	The version on which this project is targeted to be delivered. In the context of PRC, a "version" is a collection of projects that will be delivered together. Press F3 to select from the list of valid entries.
Subs	<i>[Informational only]</i> Indicates the number of sub-projects that have been opened against this project.
Target	The date this project is required, if it is essential. Press F3 to select a date from the pop-up calendar.
Tech Note	Any internal notes or comments about this project. Use the F5 (Drill) key to pull open a full text window for entering any detailed technical notes for internal use only. The technical notes are not printed in release notes.
Synopsis	An unlimited text window for entering information at the completion of the project. The synopsis is useful for release notes.
Last Talk	The cross-talk window allows one-line diary entries in a moving forward (time and date stamped) format. It is useful for dialog between team members and users. Press F3 to select a date from the pop-up calendar.
Requests	A count of the number of user requests associated with this project. Use the F5 (Drill) key to view, add, or change these associated requests.

Items [Informational only] Indicates a count of the total number of source items on this project.

Use the **F5** (Drill) key to display and maintain these items.

2. Fill in each field with the appropriate information.
 - If you press **F5** (*Drill), see “Drill Down” below.
 - If you press **F7** (Testplan), see “Test Plan” on page 51.
 - If you press **F8** (Source), see “Source” on page 51.
 - If you press **F9** (Stat), see “Stat” on page 52.
3. Press **F2** (Save).

Drill Down

From the Project Master Maintenance screen (Figure 1-15 on page 46), if you press **F5** (*Drill) with the cursor in any field with an asterisk (*), the system “drills down” to find the background information for that piece of data. The resulting screen depends on the field from which **F5** was pressed.

For example, pressing **F5** from the:

- *Ver* (version) field displays the Version Master Maintenance screen (see Figure 1-24 on page 61)
- *Target* field displays the Project Target Info sub-screen similar to Figure 1-16:

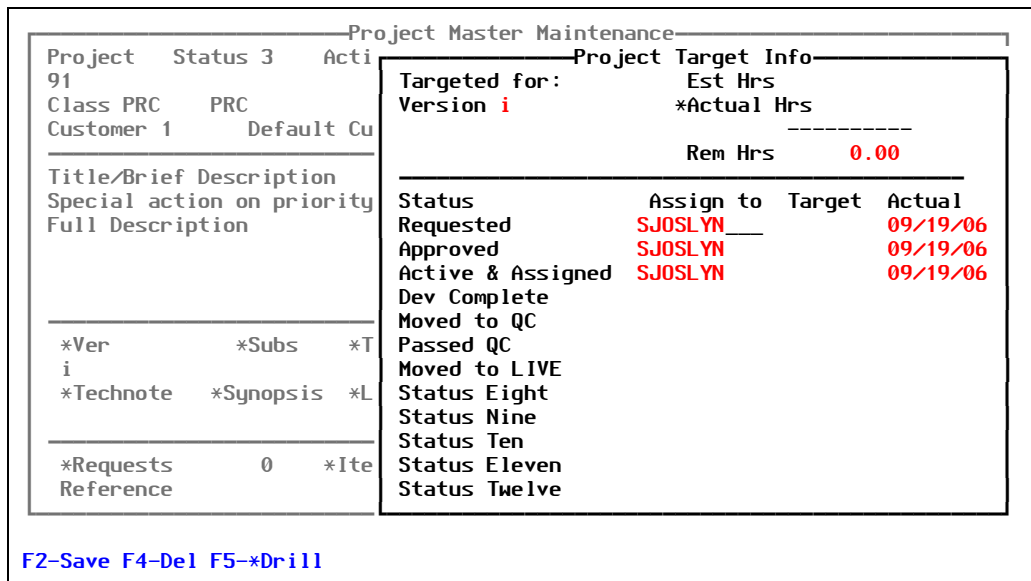


Figure 1-16: Project Target Info sub-screen

This sub-screen displays whatever numeric status codes (up to 12) have been set up in the Status Code Maintenance screen (Figure 1-7 on page 30).

Test Plan

From the Project Master Maintenance screen (Figure 1-15 on page 46), if you press **F7** (Testplan), the system displays the Test Plan (project) sub-screen. If a test plan has been defined for this project, it will appear in this sub-screen:

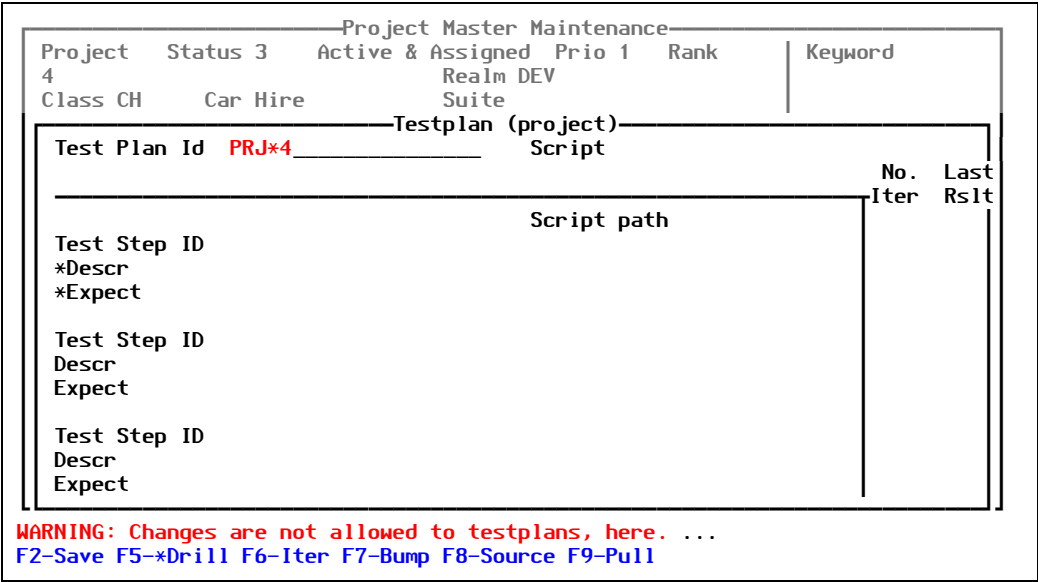


Figure 1-17: Test Plan (project) sub-screen

You are not allowed to make changes to test plans from this screen. To work with test plans, see "Test Plans" on page 77.

Source

From the Project Master Maintenance screen (Figure 1-15 on page 46), if you press **F8** (Source), the system displays the Source screen, similar to the one shown in Figure 1-18:

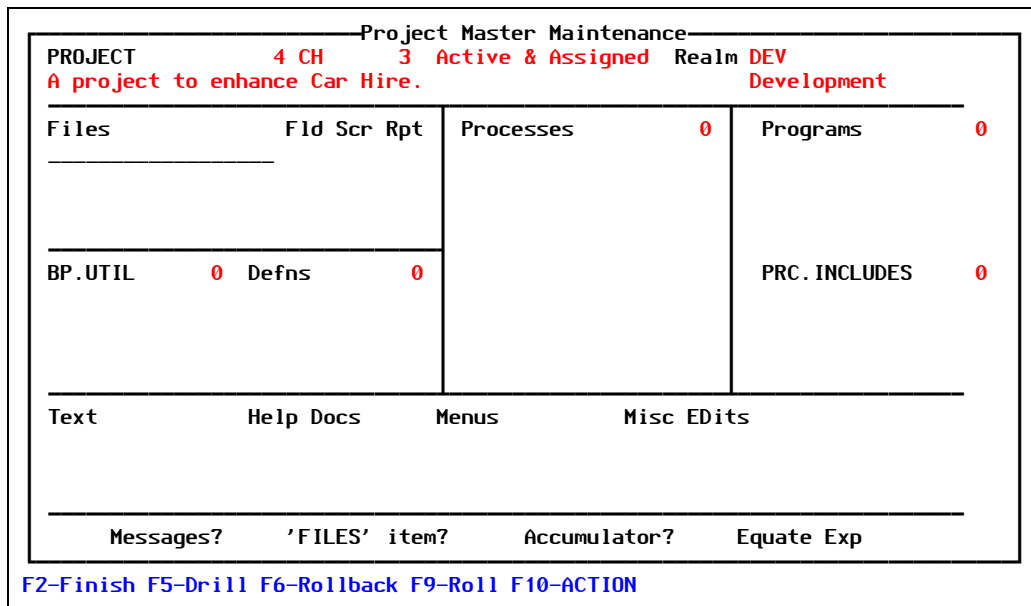


Figure 1-18: Source screen

This is the review source screen, just like the one that can be viewed from the working account; however, the perspective is different.

When you review source elements on this screen (from the Project Master Maintenance screen in the PRC account), you are in the “administrative role.” You are reviewing the items on the project, but your purpose for being here is to roll out (via **F9**) or roll back (via **F6**) the project.

When you are in working mode on an active project and review its source elements, the tools that are available from this screen allow you to:

- Review the changes
- Revert the changes
- Flick items to other projects
- Edit items
- Trace history
- Review notes

Stat

From the Project Master Maintenance screen (Figure 1-15 on page 46), if you press **F9** (Stat), the system displays the Project States sub-screen, similar to the one shown in Figure 1-19:

Project Master Maintenance																															
Project	Status 3	Active & Assigned	Prio 1	Rank																											
4			Realm DEV	Keyword																											
Class CH	Car Hire		Suite																												
Customer 1	Default	Custom	Type ENH	Enhancement																											
Title/Brief Description																															
A project to enhance Car Hire.																															
Full Description																															
<table border="1"> <thead> <tr> <th colspan="3">Project States</th> </tr> </thead> <tbody> <tr> <td>Requested</td> <td>09/01/2006</td> <td>JOHN</td> </tr> <tr> <td>Approved</td> <td>09/01/2006</td> <td>JOHN</td> </tr> <tr> <td>*Active & Assigned</td> <td>09/01/2006</td> <td>JOHN</td> </tr> <tr> <td>Dev Complete</td> <td></td> <td></td> </tr> <tr> <td>Moved to QC</td> <td></td> <td></td> </tr> <tr> <td>Passed QC</td> <td></td> <td></td> </tr> <tr> <td>Moved to LIVE</td> <td></td> <td></td> </tr> <tr> <td>testing</td> <td></td> <td></td> </tr> </tbody> </table>					Project States			Requested	09/01/2006	JOHN	Approved	09/01/2006	JOHN	*Active & Assigned	09/01/2006	JOHN	Dev Complete			Moved to QC			Passed QC			Moved to LIVE			testing		
Project States																															
Requested	09/01/2006	JOHN																													
Approved	09/01/2006	JOHN																													
*Active & Assigned	09/01/2006	JOHN																													
Dev Complete																															
Moved to QC																															
Passed QC																															
Moved to LIVE																															
testing																															
*Ver	*Subs	*T																													
1.0																															
*Technote	*Synopsis	*L																													
*Requests	0	*Ite																													
Reference																															

F2-Accept

Figure 1-19: Project States sub-screen

This sub-screen displays whatever project states have been set up in the Status Code Maintenance screen (Figure 1-7 on page 30).

Request Master

Requests are used for “problem reporting” or other customer feedback. Each request may or may not relate to one or more projects.

To work with the Request Master, perform the following:

1. Select **Files > Request Master**.

The system displays the Request Master Maintenance screen, similar to the one shown in Figure 1-20:

Request Master Maintenance				
Request ID	Req User	Rep	Keyword(s)	
NEW	Date	Time	Status	
Class	Type	Prio		
Title / brief description				
Full Description				
*Work Around	*Last Talk	*Target Info		
*Resolved	R E L A T E D		*Action	Due Court
Attachments	*Requests	*Projects		

F2-Save F4-Del F5-*Drill F6-Related F7-Projects F8-Attach F10-Action

Figure 1-20: Request Master Maintenance screen

Fields with an asterisk (*) next to the field name are fields in which you can drill down for more information using the **F5** (Drill) key.

Explanations of the fields are as follows:

- Request ID** The code or ID that represents this particular request.

Press **F3** to select from the list of valid entries.
- Req User** The client ID of the company that made this request or inquiry. If multiple customers have made this particular request, they may be referenced in the RELATED section in the bottom portion of the screen.

Press **F3** to select from the list of valid entries.
- Rep** The representative or contact at the requesting organization. May be the department, the individual user, or an account representative.
- Date** The date on which the request was originally made.

Press **F3** to select a date from the pop-up calendar.
- Time** The time at which the request was originally made.
- Status** The code or ID that represents the status of the request. The value of this field can be modified automatically by the progression of its associated project.

Press **F3** to select from the list of valid entries.

Class

The code or ID that represents this project's classification. You can define a classification to mean that work is accomplished on a particular computer, on a particular account/system, by a particular programmer, or any number of things.

The project uses its classification to determine software location and other information. The class code often is the module of the software in which the problem is being reported. Various relationships may exist:

- A one-to-one relationship between the "software system" (defined by the SB+ SYSID) and the Class code.
- A many-to-one relationship between more than one logical modules or products in one system.
- A one-to-many relationship between one module in one system with multiple classes of requests for information and reporting.
- There may also be classes of requests or issues that are not related to a particular module.

In the case where the problem or enhancement spans multiple modules, pick the one module that is most directly related to the request and assign the request to that module. PRC will create "systems" automatically.

Class (cont'd)

The module in which the problem resides may be unknown at time of reporting. Some organizations find it helpful to define a class code for "unknown."

Sometimes, issues are reported that are not related to a particular module, but rather to a terminal, a printer, the network, the phone system, or a request for documentation. You can define these situations as classes, also.

Reminder: If a request requires a project to be opened and work performed against that same project, you must assign a class to that project that indicates which software system the changes will be made in (or the parent, in the case of systems).

For more information on defining class codes, see ["Class Codes" on page 22](#).

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

Type

The code or ID that represents the "type" of project. Breaking down requests and projects into types is useful for reporting and managing.

For more information on defining type codes, see ["Type Codes" on page 24](#).

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

Prio	<p>The number that represents the requested priority—the priority of the request from the perspective of the individual making the request. Valid entries range from 1 (highest priority) through 9 (lowest priority).</p> <p>For more information on defining priority codes, see “Priority Codes” on page 28.</p> <p>Press F3 for a list of valid entries.</p>
Keyword(s)	<p>Special codes or words that identify and group this request in the future. You can attach any number of keywords to a request so you can retrieve and sort requests and projects by keywords later on.</p> <p>The keywords you add and define are stored in the <code>KEYWORD</code> file, which acts as a dictionary.</p> <p>Press F3 to select from the list of valid entries.</p>
Title / brief description	<p>A one-line summary of the request—a quick explanation.</p>
Full Description	<p>A more verbose or detailed explanation of the request, if necessary. Elaborates on the one-line summary in the <i>Title/Brief Description</i> field.</p>
Work Around	<p>Enter the Work Around.</p>
Last Talk	<p>Enter the Last Talk.</p> <p>Press F3 to select a date from the pop-up calendar.</p>
Target Info	<p>A target information sub-screen that provides fields for the promised or expected delivery date and version.</p>
Resolved	<p>The date on which the issue was finally resolved or delivered to the customer.</p> <p>Press F3 to select a date from the pop-up calendar.</p>
Action	<p>The activity currently being performed or due, including the party responsible for the activity and deadline information. Works in conjunction with the <i>Due Court</i> field. In other words, the value of these two fields answers the questions, “What is being done? The ball is in who’s court?”</p> <p>When a new activity is entered, use the “bump” procedure to push the stack of historical activity down, rather than simply replacing the data in the field. What was the current activity will push down and should be added to review historical activity log. The next activity may be entered, as well, and will pop into the current activity the next time the bump feature is used.</p>

Due	<p>Represents when an activity is due and who is supposed to be doing it. Works in conjunction with the <i>Action</i> field. In other words, the value of these two fields answers the questions, “What is being done? The ball is in who’s court?”</p> <p>When a new activity is entered, use the “bump” procedure to push the “who’s court” value down, to stay in association with the current activity.</p>
Attachments	<p>Indicates whether attachments—hardcopy samples or error print-outs or items of that nature—were delivered with this request. Press F5 (Drill) to indicate what they are and where they are located.</p>

2. Fill in each field with the appropriate information.
 - If you press **F5** (*Drill), see “Drill Down” below.
 - If you press **F6** (Related), see “Related Requests” on page 57.
 - If you press **F7** (Projects), see “Projects” on page 58.
 - If you press **F8** (Attach), see “Attachments” on page 59.
3. Press **F2** (Save).

Drill Down

From the Request Master Maintenance screen (Figure 1-20 on page 54), if you press **F5** (*Drill) with the cursor in any field with an asterisk (*), the system “drills down” to find the background information for that piece of data. The resulting screen depends on the field from which **F5** was pressed.

Related Requests

From the Request Master Maintenance screen (Figure 1-20 on page 54), if you press **F6** (Related), the system displays the Related Requests sub-screen, as shown in Figure 1-21:

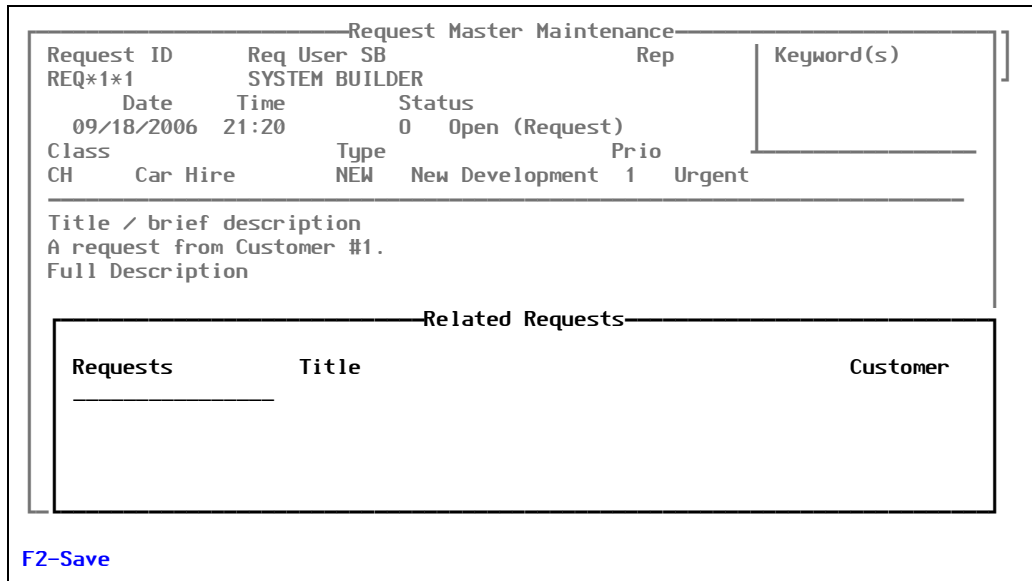


Figure 1-21: Related Requests sub-screen

This sub-screen shows other requests related to this request. You can use it to select other requests to make them related to this one.

This sub-screen is simply a way to give more weight to a particular request, if several requests are actually related to it. For example, if three customers asked for the same thing, it gives the overall request a higher priority.

But, this sub-screen also allows you to perform the more obvious task of showing that this request is closely related to another request that has already been submitted—and that they can be resolved together.

Projects

From the Request Master Maintenance screen (Figure 1-20 on page 54), if you press **F7** (Projects), the system displays the Projects sub-screen, as shown in Figure 1-22:

Request Master Maintenance

Request ID	Req User	SB	Rep	Keyword(s)
REQ*1*1		SYSTEM BUILDER		
Date	Time	Status		
09/18/2006	21:20	0 Open (Request)		
Class	Type	Prio		
CH Car Hire	NEW New Development	1 Urgent		

Title / brief description
A request from Customer #1.
Full Description

Projects				
PRJ No.	St	Title	Assn to Ver	Ravel Name

F2-Save

Figure 1-22: Projects sub-screen

Use this sub-screen to make an association between requests and projects. With this association, the request can automatically be updated and the user notified as the project goes through its life-cycle.

Note

The relationship between requests and projects is flexible. It can be:

- Many-to-one (many requests are associated to one project)
- One-to-many (a request can be broken up into many projects)
- One-to-one (a single request to a corresponding project)

Attachments

From the Request Master Maintenance screen (Figure 1-20 on page 54), if you press **F8** (Attach), the system displays the Attachments sub-screen, as shown in Figure 1-23:

Request Master Maintenance						
Request ID	Req User	SB	Rep	Keyword(s)		
REQ*1*1	SYSTEM	BUILDER				
	Date	Time	Status			
	09/18/2006	21:20	0 Open (Request)			
Class	Type	Prio				
CH	Car Hire	NEW	New Development	1	Urgent	
Title / brief description						
A request from Customer #1.						
Attachments						
Attach File	Attach Type					
Attach Desc						

F2-Save F4-Del

Figure 1-23: Attachments sub-screen

Use this sub-screen to name attachments that go along with this request (or list their actual file locations).

Version Master

Versions are used to manage quantities of projects gathered together to “baseline” a version or even as a “master project” to combine projects logically.

To work with versions, perform the following:

1. Select **Files > Version Master**.

The system displays the Version Master Maintenance screen, similar to the one shown in [Figure 1-24](#):

Version Master Maintenance			
Version	System	Keyword	
Description			
Source Realm Status	No. Sub-Releases		
*Project	Title	Assigned to	Status

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

Figure 1-24: Version Master Maintenance screen

Explanations of the fields are as follows:

Version The revision/version/release number that identifies this stage of the software for a module.

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

System The name of the SB+ system (software module) for which this is a version definition. This field is alphanumeric with a maximum length of 12.

Press **[F3]** to select from the list of valid entries. You can tag multiple items from this lookup window.

Description A concise but meaningful description of the release itself and its purpose.

Source Realm The realm from which the source will be coming. Valid entries include:

- **D** — Development realm (typically used)
- **R** — Imported revisions

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

No. Sub-Releases *[Informational only]* The number of sub-releases issued against this release/revision/version.

Status Status of Project/Version.

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

Project	Projects that are assigned as part of this revision. Press F3 to select from the list of valid entries.
Keyword	Special codes or words that identify and group this version. You can attach any number of keywords to a version so you can retrieve and sort them by keywords later on. Press F3 to select from the list of valid entries. You can add keywords to the dictionary “on the fly.”

2. Fill in each field with the appropriate information.
 - If you press **F5** (Drill), see “[Drill Down](#)” below.
 - If you press **F6** (Sched), see “[Schedule](#)” on page 62.
 - If you press **F7** (Sort), see “[Sort](#)” on page 63.
 - If you press **F8** (Pre-Scan), see “[Pre-Scan](#)” on page 63.
 - If you press **F9** (Roll Out), see “[Roll Out](#)” on page 64.
3. Press **F2** (Save).

At least one Version must be opened for each system, even in a “perpetual” system that has no distinct versions. (If that is the case, create one default version.)

Note

Only a version ID, a text description, and an **A** (Active) status is required to get started.

Drill Down

From the Version Master Maintenance screen ([Figure 1-24 on page 61](#)), if you press **F5** (Drill) with the cursor in the *Project* field, the system “drills down” to find the background information for that piece of data. The resulting screen is the Project Master Maintenance screen (see [Figure 1-15 on page 46](#)).

Schedule

From the Version Master Maintenance screen ([Figure 1-24 on page 61](#)), if you press **F6** (Sched), the system displays the Schedule/Target Info sub-screen, as shown in [Figure 1-25](#):

Version Master Maintenance		Keyword
Version 1.0	System CH	
Description Demo		
Source Realm DEV	No. Sub-Releases 0	
Schedule/Target Info		
Targeted/Scheduled Dates		Actual Dates
Initiate/Open	_____	Initiate/Open
Freeze		Freeze
Complete		Complete
Test Release		Test Release
Live Release		Live Release

F2-Save F4-Del F5-*Drill F8-Scan F9-Roll!

Figure 1-25: Schedule/Target Info sub-screen

Sort

From the Version Master Maintenance screen (Figure 1-24 on page 61), if you press **F7** (Sort), the system sorts by project.

Pre-Scan

From the Version Master Maintenance screen (Figure 1-24 on page 61), if you press **F8** (Pre-Scan), the system displays the Pre-Scan sub-screen, as shown in Figure 1-26:

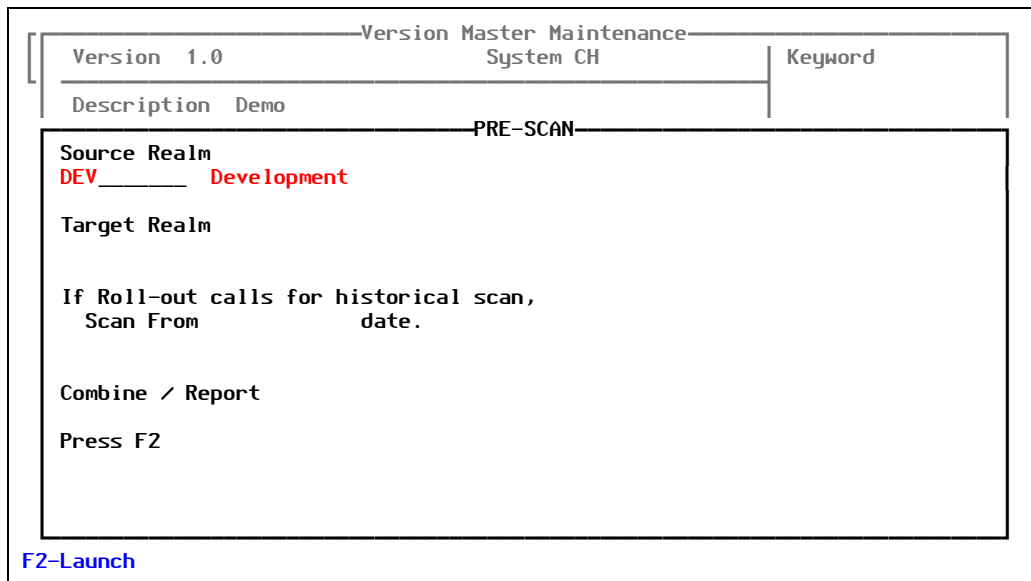


Figure 1-26: Pre-Scan sub-screen

Use this sub-screen to launch a pre-rollout scan on a version. It is similar to the scan performed on an individual project in that it has:

- A “from” and “to” realm
- If historical rollouts are indicated, it will ask for the beginning date

A version that is rolled out is a collection of multiple projects. The offer to combine/report the conflicts or other failures is useful: rather than responding to errors one-by-one, the entire version is scanned for problems. They are presented in one report.

Roll Out

From the Version Master Maintenance screen (Figure 1-24 on page 61), if you press **F9** (Roll Out), the system displays the Roll-Out sub-screen, as shown in Figure 1-27:

Version Master Maintenance			
Version	System CH	Keyword	
1.0		ROLL-OUT	
Source Realm		Next Ver	1.1
DEV	Development		
Target Realm		Suite	
Scan From		(Historical)	
Ravel Name			
Press F2	(to begin rollout)		
MOVEMENT HISTORY			
Rolled out to:	Realms	on Date	Ravel Files

F2-Launch

Figure 1-27: Roll-Out sub-screen

Use this screen to roll out a version.

Customer Master

A customer can be a company, an individual, a department—whatever is meaningful in terms of who has requested a project and who should be notified about questions or completion.

“Customers” are used primarily for validating the “requested by” entry on both requests and projects. Many organizations choose to have those fields validate against an existing employee or a customer database. Some organizations populate these files from those existing databases, to avoid changing the fields on the screens.

To work with the Customer Master, perform the following:

1. Select **Files > Customer Master**.

The system displays the Customer Master Maintenance screen, similar to the one shown in [Figure 1-28](#):

Customer Master Maintenance						
Customer	1	Station		Status	3__ Active & A	Dept/Repr
Name	Default Customer			Main Phone	(617) 555-1111	
Addr	123 Main Street			Fax		
City	Boston	St	MA	Zip	02111	
Contact	Connor Tact	Title/Loc	HR	Extension	321	Contact Email
						ctact@company.com
Software Information				Expirations		WO WARRANTY
Modules	Ver	Inst Dt	By	Warranty	Support	*WO Cnt
CH	1.0	09/07/06	SJ	09/07/07	03/07/07	

F2-Save F4-Del F5-W0 F6-Contract F7-Tech F8-WIP

Figure 1-28: Customer Master Maintenance screen

Explanations of the fields are as follows:

- Customer ID** The code or ID that represents this client’s account.
 Press **[F3]** to select from the list of valid entries.

- Station** [Informational only] The customer type or station to which the customer is related. (Stations are a future concept).

- Status** The state of the account, which will often be used to inform technical support on what basis this customer is to receive support. Can be updated by the billing process.
 Press **[F3]** to select from the list of valid entries.

- Dept/Repr** The individual (account representative, department, division, or individual) that is primarily responsible for this customer.

- Name** The name of the client company.

- Address** The street address of the client company.

- Address2** The second line of the client company address, if necessary.

- City** The city for the company’s address.

- St** The two-character postal code for the client company’s state.

- Zip** The ZIP code for the company address.

Main Phone	The main telephone number to reach this company. Other phone numbers may be listed below with contact or location.
Fax	The main FAX number to reach this company.
Modem Ph	The main modem number used for dialing in, as well as any alternate modem numbers. For other numbers that require explanation, use the contact fields below.
Contact	The name of the contact at the customer site.
Title/Loc	The contact's title or their responsibility in regard to you.
Extension	The telephone extension, car phone, home phone, pager number or other information to assist in reaching this individual.
Contact Email	The contact's e-mail address.
Modules	The names of software modules owned by this client.
Ver	<p>Indicates the "latest and greatest" version number for this module that has been installed at the customer site.</p> <p>You can approach this field two ways for "current version" in relation to more recent patches, bug fixes, and/or customizations:</p> <ul style="list-style-type: none"> • The company may issue new sub-versions whenever something is issued to a client. In this scenario, the value of this field truly represents the "latest and greatest" information. • The patches are issued by project and are listed in the next (drill-down) field for individual projects delivered. Information about the "latest and greatest" is derived by reviewing their last version upgrade and the projects delivered since that upgrade. If all issues are given a version number, it may still be useful to track the individual projects that have an associated "warranty period" on the project drill-down on this screen.
Inst Dt	<p>The date on which this module was installed—primarily for warranty and support contract purposes.</p> <p>Press [F3] to select a date from the pop-up calendar.</p>
By	Indicates who installed this software/version/upgrade was installed, or how it was installed.
Warranty	<p>The date on which the warranty on this module (perhaps individual work orders in the module) expires.</p> <p>Press [F3] to select a date from the pop-up calendar.</p>

Support The date on which support for this module expires or through which support is paid. Details about allocated/used support hours are on the subscreen.

Press **F3** to select a date from the pop-up calendar.

WO Cnt The field from which you press **F5** (WO) or **F6** (Contract) to drill down for that specific information.

Press **F3** to select from the list of valid entries.

2. Fill in each field with the appropriate information.
 - If you press **F5** (WO), see “[Work Order](#)” below.
 - If you press **F6** (Contract), see “[Contract](#)” on page 69.
 - If you press **F7** (Tech), see “[Technical Information](#)” on page 70.
 - If you press **F8** (WIP), see “[Work in Process](#)” on page 70.
3. Press **F2** (Save).

At least one Version must be opened for each system, even in a “perpetual” system that has no distinct versions. (If that is the case, create one default version.)

Note

Only a version ID, a text description, and an **A** (Active) status is required to get started.

Work Order

From the Version Master Maintenance screen ([Figure 1-24 on page 61](#)), if you press **F5** (WO) with the cursor in the *WO Cnt* field, the system displays the Work Order Warranties screen, as shown in [Figure 1-29](#):

Customer Master Maintenance			
Customer 1	Station	Status 3	Active & A Dept/Repr
Name Default Customer		Main Phone (617) 555-1111	
Addr 123 Main Street		Fax	
		Modem Ph	
City Boston	St MA	Zip 02111	
Contact	Title/Loc	Extension	Contact Email
Connor Tact	HR	321	ctact@company.com

Softw		Work Order Warranties	
Modules	Ver	Module CH	Deliv Wrnty Exp
CH	1.0	Work Orders	

F2-Save F4-Del

Figure 1-29: Work Order Warranties sub-screen

Contract

From the Version Master Maintenance screen (Figure 1-24 on page 61), if you press **F6** (Contract), the system displays the Rate & Support Allocation sub-screen, as shown in Figure 1-30:

Customer Master Maintenance			
Customer 1	Station	Status 3	Active & A Dept/Repr
Name Default Customer		Main Phone (617) 555-1111	
Addr 123 Main Street		Fax	
		Modem Ph	
City Boston	St MA	Zip 02111	
Contact	Title/Loc	Extension	Contact Email
Connor Tact	HR	321	ctact@company.com

Rate & Support Allocation						
Activity Cd	Unit	Rate	Alloc Used	Period Clear	Alloc	Exp

F2-Save F4-Del

Figure 1-30: Rate & Support Allocation sub-screen

Technical Information

From the Version Master Maintenance screen (Figure 1-24 on page 61), if you press **F7** (Tech), the system displays the Technical Information sub-screen, as shown in Figure 1-31:

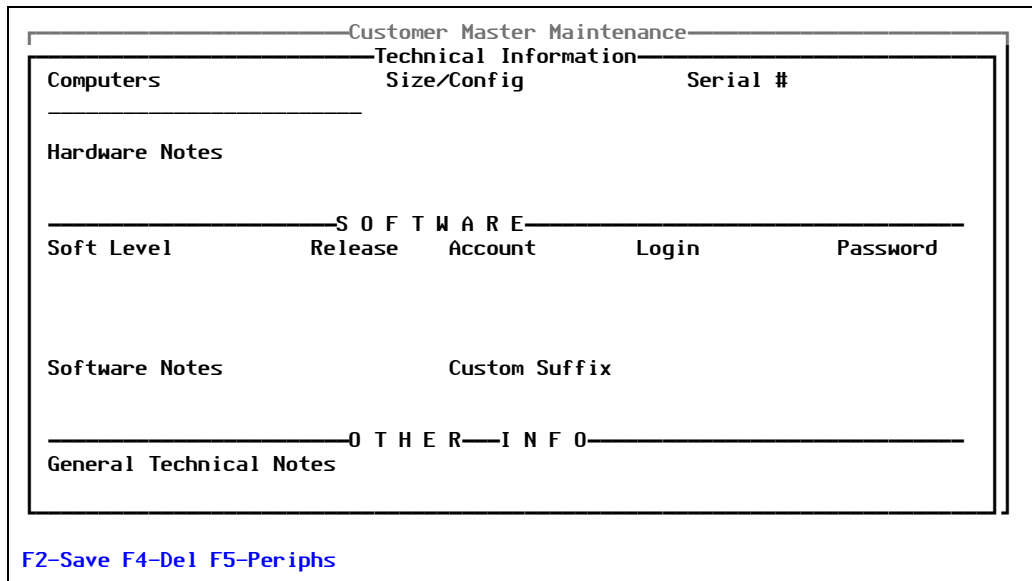


Figure 1-31: Technical Information sub-screen

Work in Process

From the Version Master Maintenance screen (Figure 1-24 on page 61), if you press **F8** (WIP), the system displays the Work in Process sub-screen, as shown in Figure 1-32:

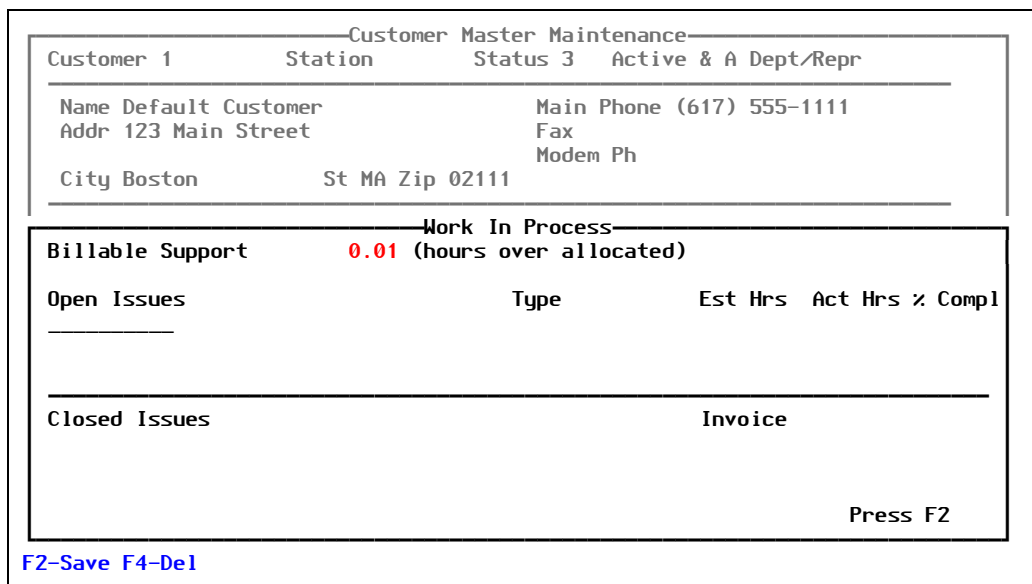


Figure 1-32: Work in Process sub-screen

Department Master

A customer can be a company, an individual, a department—whatever is meaningful in terms of who has requested a project and who should be notified about questions or completion.

“Department” can refer to either your own organization or the customer organization. You can also use it to subdivide customers by account representatives or other criteria.

To work with the Department Master, perform the following:

1. Select **Files > Department Master**.

The system displays the Department Master Maintenance screen, similar to the one shown in [Figure 1-33](#):

Department Master Maintenance

Department/ Cost Center/ Representative Identifier _____

Description or Title

Dept. Head or "Report to"ence

RESERVED FOR FUTURE USE

F2-Save F4-Del

Figure 1-33: Department Master Maintenance screen

Explanations of the fields are as follows:

Identifier The department code or representative code for use on requests and projects throughout PRC.

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

Description or Title A full description or name for the department.

Dept. Head or "Report to" The name of the department head or, at least, the MIS-related “the buck stops here” person.

2. Fill in each field with the appropriate information.
3. Press **[F2]** (Save).

File Sets & Test Plans

Groups of Files & Quality Tests

File Sets

A file set is simply a list of files. There are a few “special purpose” file sets in PRC, but you can add any number of file sets and name them according to any scheme.

Once you create a file set, it is used mainly to restrict a string search (the **/SRCH** screen) to a particular set of files.

For more information about the file sets for special purposes, see the “File Sets” tip sheet.

To work with file sets, perform the following steps:

1. Select **Files > File Sets**.

The system displays the System File Sets screen, similar to the one shown in [Figure 2-1](#):

System File Sets								
Set Name	SOFTWARE							
Files	Data/	Rem	Synonym	Prefix	De- lim	*Over/ Part	Compiled *Y/N	*Incl
	Dict							
HELP	Dict	No		SYSID		Yes	No	0
MENUS	Dict	No		SYSID		Yes	No	0
DEFN	Dict	No		SYSID		Partial	No	0
CONTROL	Dict	No		SYSID		Yes	No	0
PROGS	Data	Yes		SYSID		Yes	Yes	0
PROCESS	Data	No		SYSID		Yes	No	0
USERTEXT	Data	No		SYSID		Yes	No	0
DRIVERS	Data	No		SYSID		Yes	No	0

F2-Save F4-Del F8-Alphabetize F9-*Add'l F10-Action

Figure 2-1: System File Sets screen

Explanations of the fields are as follows:

Set Name	<p>The code or ID that represents this file sets. Standard file sets provided with PRC are:</p> <ul style="list-style-type: none">• SOFTWARE — primary tools files for SB+ accounts• SOFTWARE2 — secondary software files, SB+ or non-SB+ SOFTWARE and SOFTWARE2 are always searched, compared, owned, and so on, in addition to files named in other sets.• OWN*XX — Files owned by a system (if not in FILES)• TRACK*XX — Additional tracked files (SB+ or non)• NOTRACK*XX — Files not to be tracked by EDITOR• SEARCH*XX — Default search set for system• COMPARE*XX — Default compare set for systsem SEARCH and COMPARE default to OWN. <p>You can create and specify additional file sets for searches and comparisons.</p> <p>Press F3 to select from the list of valid entries.</p>
Files	<p>The names of files that are to be associated with this set name.</p> <p>The names may span realms, accounts, and systems; therefore, no F3 help is available.</p>
Data	<p>Identifies what parts of this file belong to this file set. Applies to searches and comparisons, primarily. Valid entries are:</p> <ul style="list-style-type: none">• DICT — Dictionary• DATA — Data• BOTH — Both dictionary and data <p>Press F3 to select from the list of valid entries.</p>
Rem	<p>Indicates whether this is a file in which the items should be stamped with a remark. Usually applies to program files only. Valid entries are Y (Yes) and N (No).</p> <p>Remarks are established on the Preference sub-screen.</p> <p>Press F3 to select from the list of valid entries.</p>
Synonym	<p>The synonym or abbreviation for this file name, if applicable. The system “evaluates” this value to the real file name during rollouts and other operations.</p>
Prefix	<p>Indicates how this file is prefixed in the working accounts. Valid entries are:</p> <ul style="list-style-type: none">• SYSID — System ID

For example, SB+ tools files use the naming convention *XXPROCESS*, where **XX** is the SYSID.

- **ACNT.NAME** — Account name

Press **F3** to select from the list of valid entries.

File Delim

The delimiter for files. Leave this field blank if the prefix is attached directly with no delimiter; for example, *PRCPROCESS*. Valid entries are:

- Dash character (-)
- Underscore character (_)
- Period (.)
- Dollar sign (\$)

File Over

Indicates whether an item can be overwritten during rollout. Valid entries are:

- **Y** — Yes, the item is delivered, overwriting
- **N** — No, the item will never be delivered if its already there

Only use if the item is new.

- **P** — Partial (see drill-down)

Press **F3** to select from the list of valid entries.

Y/N

Indicates whether this file is compiled file. Valid entries are:

- **Y** — Yes, this is a compiled file
- **N** — No, it is not a compiled file

Compiled files on UniData have a *DIR*, or have a *.O* extension on UniVerse, and so on.

Press **F3** to select from the list of valid entries.

Incl

Indicates included item. Use **F5** (Drill-down) to enter the (program) file name(s) that might **INCLUDE** items from this file, and details about how to handle this.

Press **F3** to select from the list of valid entries.

2. Fill in each field with the appropriate information.
3. If you press **F8** (Alphabetize), the system redisplay the System File Sets screen, but listed in alphabetical order as shown in [Figure 2-2](#):

System File Sets								
Set Name SOFTWARE								
Files	Data/	Rem	Synonym	Prefix	De- lim	*Over/ Part	Compiled *Y/N	*Incl
	Dict							
CONTROL	Dict	No		SYSID		Yes	No	0
DEFN	Dict	No		SYSID		Partial	No	0
DRIVERS	Data	No		SYSID		Yes	No	0
HELP	Dict	No		SYSID		Yes	No	0
MENUS	Dict	No		SYSID		Yes	No	0
PROCESS	Data	No		SYSID		Yes	No	0
PROGS	Data	Yes		SYSID		Yes	Yes	0
USERTEXT	Data	No		SYSID		Yes	No	0

F2-Save F4-Del F8-Alphabetize F9-*Add'l F10-Action

Figure 2-2: System File Sets screen

- If you press **F9** (Add'l) while the cursor is in a field that has a star next to it (*Over/Part*, *Compiled Y/N*, *Incl*), the system displays a sub-screen that corresponds to that field. The example in [Figure 2-3](#) is for the *Compiled Y/N* field:

System File Sets								
Set Name SOFTWARE								
Files	Compiled File Details					Catalog		
	Y/N	Verb	Compile Opts	Obj file	UV	Y/N/S	Type	Add'l word
CONTROL	N					No	N/A	
DEFN	No					No	N/A	
DRIVERS	No					No	N/A	
HELP	No					No	N/A	
MENUS	No					No	N/A	
PROCESS	No					No	N/A	
PROGS	Yes	BASIC				Yes	DIRECT FORCE	
USERTEXT	No					No	N/A	

F2-Save F4-Del F5-*Drill

Figure 2-3: Compiled File Details sub-screen

- Press **F2** (Save).

Test Plans

Quality Management (QM) begins at the beginning of any software project, and persists throughout the software development life-cycle. PRC aspires to provide cohesion and vision to QM throughout the life-cycle and to provide the tools to automate and support good QM processes.

There are two main aspects to the test management:

- The ability to set up test plans that can be general or that can be for specific projects. These can be very simple or very elaborate. The information on how to set these up is included in this section.
- The other part is the ability to use the "QA Bump" by attaching these test plans to projects and marking the steps as passed or failed.

For more information, see the *Software Quality & Testing Guide*.

Essentially, this menu option in the PRC account is administrative portion of test plans. Use this screen as a place to store named test plans. Basically, this is a kind of master file in which you create named test plans.

However, the *Software Quality & Testing Guide* explains more about *using* test plans and about creating them "on the fly" for a project.

To work with test plans, perform the following:

1. Select **Files > Test Plans**.

The system displays the Test Plans screen, similar to the one shown in [Figure 2-4](#):

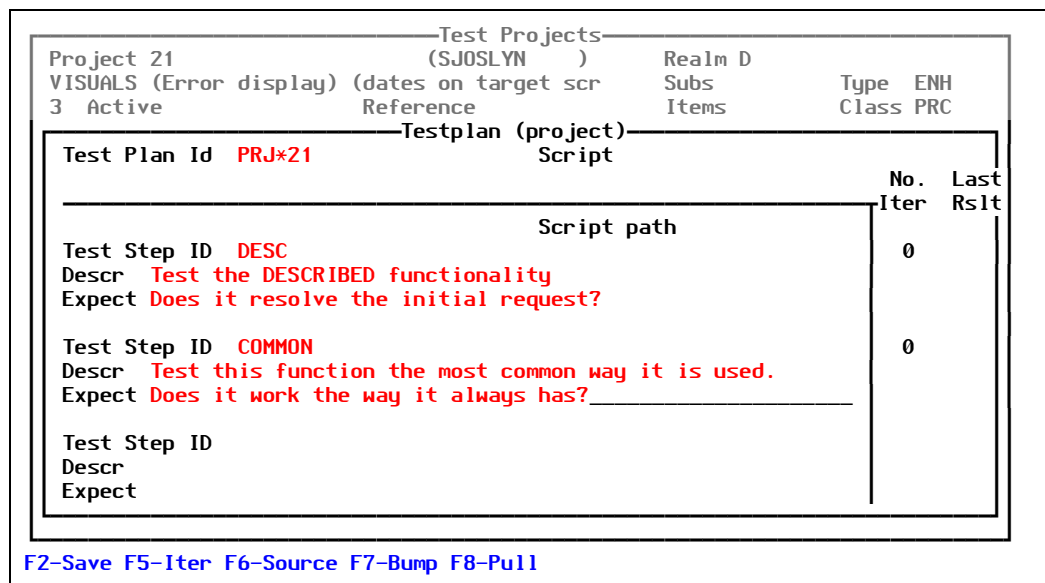


Figure 2-4: Test Plan (project) screen

Explanations of the fields are as follows:

Testplan ID	The code or ID of this test plan. A test plan can contain another test plan as one of its steps. For more in-depth information, as well as information on how to use test plans, please see the <i>Software Quality & Testing Guide</i> . Press F3 to select from the list of valid entries.
Descr	A concise but meaningful description of the entire, overall test plan.
Test Step ID	The code or ID that represents an individual test step, which are explained in “Test Steps” and “Nested Test Plans” on page 80.
Step Script	A file name or path location of a stored script. “Script” in this sense includes scripts created using test script tools made by Rational [®] , Segue [®] , Compuware [®] , and so on; programs or scripts written in a powerful script language; or just a recorded script made using SBClient.
Test Step Desc	A concise but meaningful description of the step.
Test Step Expect	The expected result of this test step once it is performed. For example, if the test step is “add an employee,” the value of this field is, “The employee is added to the master file and can be recalled by their employee ID.”

2. Fill in each field with the appropriate information.
 - If you press **F5** (Iter), see [“Iterations”](#) (below).
 - If you press **F6** (Source), see [“Sources and Pulling”](#) on page 79.
 - If you press **F7** (Bump), the system will do one of the following:
 - Promote the project to the next status (as defined in the QA Setup menu from Preferences) if all the steps are “passed”
 - Offer to open a sub-project and pull in the notes about whatever failedFor more on bumping, refer to the *Project Tracking Guide*.
 - If you press **F8** (Pull), see [“Sources and Pulling”](#) on page 79.
3. Press **F2** (Save) when you have completed creating or editing this test plan.

Iterations

From the Test Plan (project) screen ([Figure 2-4 on page 77](#)):

1. Place the cursor on a particular test step and press **F5** (Iter).

The system drills down into the specific iterations of testing against that step. It displays the Test Plan (project) Iterations sub-screen, similar to the one shown in [Figure 2-5](#):

Test Projects			
Project 21	(SJOSLYN)	Realm D	
VISUALS (Error display)	(dates on target scr	Subs	Type ENH
3 Active	Reference	Items	Class PRC

Testplan (project) iterations	
Test Step: DESC	
Test the described functionality.	
Does it resolve the initial request?	
Date & Time	Pass/Fail Notes
03/24/05 08:16PM	F
Nope, this is not what I want. I have completely changed my mind about it.	

F2-Save

Figure 2-5: Test Plan (project) Iterations sub-screen

The system displays the step name, description, and expected results at the top of the sub-screen.

An area below the top information contains pairs of lines that represent three fields: *Date & Time*, *Pass/Fail*, and *Notes*.

2. In the *Date & Time* field, enter the date/time stamp.
3. In the *Pass/Fail* field, enter whether the step passed or failed (type **P** or **F**).
4. On the next line (*Notes*), enter your comments about the test.

If the value of *Pass/Fail* is **F**, the text you enter in the *Notes* line will become the default description of sub-projects created by bumping when a step fails.

Sources and Pulling

Existing, stored test plans can be pulled onto a new test plan automatically, based on the source items that are on the test plan and to which test plans those source items have been attached previously.

From the Test Plan (project) screen ([Figure 2-4 on page 77](#)):

1. Press **[F6]** (Source).
PRC launches the Attach Source function to indicate the source items that should be attached to this test plan.
2. Press **[F8]** (Pull) to gather all previous test plans that have previously had each of those source items attached.

PRC pulls each test plan on as a step of this test plan.

Test Steps

A test plan is a set of “steps” (things to test, things to do).

For example, you are making a test plan to test the Employee Master screen. You determine someone should perform the steps in [Figure 2-6](#):

```

TESTPLAN = EMPLOYEE.MASTER
STEP 1 - Add a new employee
STEP 2 - Delete an employee
STEP 3 - Change an employee's name
STEP 4 - Change an employee's sex (it does happen)
    
```

Figure 2-6: Test Plan Example #1

In PRC, the steps for your text plan would look like [Figure 2-7](#):

```

-----Test Plans-----
Testplan Id  EMPLOYEE.MASTER  Script
*Descr      To test the Employee Master screen.
Attach Cnt  0
-----
Cnt      Test Step      Step Script
1.      1              Add a new employee
*Desc   Add a new employee
*Expect
2.      2              Delete an employee
*Desc   Delete an employee
*Expect
3.      3              Change an employee's name
*Desc   Change an employee's name
*Expect
4.      4              Change an employee's sex
*Desc   Change an employee's sex
*Expect
F2-Save F4-Del F5-*Drill F7-Source
    
```

Figure 2-7: Test Plans screen for Example #1

Nested Test Plans

An example of a test plan that contains other test plans as part of its overall steps is when you make a test plan for validating pay amounts.

The test plan is called **VALIDATE . PAYCHECK** and it contains the steps in [Figure 2-8](#):


```

TESTPLAN = VALIDATE.PAYCHECK
STEP 1 - Ensure that the FICA is correct
STEP 2 - Ensure that the insurance deduction is correct
STEP 3 - Ensure that the net payment amount is correct

```

Figure 2-8: Test Plan VALIDATE.PAYCHECK

So now the test plan above called **EMPLOYEE.MASTER** might include **VALIDATE.PAYCHECK** as one of its steps.

Now you have the steps in [Figure 2-9](#):

```

TESTPLAN = EMPLOYEE.MASTER
STEP 1 - Add a new employee
STEP 2 - Delete an employee
STEP 3 - Change an employee's name
STEP 4 - Change an employee's sex
STEP 5 - VALIDATE.PAYCHECK

```

Figure 2-9: Test Plan EMPLOYEE.MASTER

When a QA technician calls up the test plan **EMPLOYEE.MASTER**, they see the “exploded” test plan—the version that has all steps—that actually looks like [Figure 2-10](#):

```

TESTPLAN = EMPLOYEE.MASTER
STEP 1 - Add a new employee
STEP 2 - Delete an employee
STEP 3 - Change an employee's name
STEP 4 - Change an employee's sex
STEP 5 - Ensure that the FICA is correct
STEP 6 - Ensure that the insurance deduction is correct
STEP 7 - Ensure that the net payment amount is correct

```

Figure 2-10: Exploded Test Plan EMPLOYEE.MASTER

The example of the exploded plan does not occur when you select **Files > Test Plans**. It only happens when you go to *use* that test plan by:

- Calling up a project in **/SRC**
- Pressing **F10** (QA)
- Entering the test plan, such as **EMPLOYEE.MASTER**

Then the system displays a question similar to the one in [Figure 2-11](#):

Bump	Print	Notes	TOC	QA	Other			
Project 98		Review Source Items (SJOSLYN)		Realm D				
Problem deleting subprojects (Keystone)		Reference		*Subs	Type ENH			
3	Active & Assigned	Testplan (project)		Items	3 Class PRC			
Test Plan Id	EMPLOYEE.MASTER	Script		No.	Last			
				Iter	Rslt			
Test Step ID	<div style="border: 1px solid black; padding: 5px;"> <p>The plan you've selected has sub-plans. Would you like these additional plans exploded into the steps of this project's plan? (Up to two levels deep.)</p> <table border="1" style="width: 100%; text-align: center;"> <tr> <td>Continue</td> <td>No, 1 level</td> </tr> </table> </div>				Continue	No, 1 level		0
Continue					No, 1 level			
*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 2-11: Explode question

User Profiles & Timesheets

Files Concerning Users

User Profiles

Each PRC user must have a defined profile that outlines where they may go and what they may do there in terms of the configuration and change.

To work with profile information for PRC users, perform the following steps:

1. Select **Files > User Profiles**.

The system displays the User Profile screen, similar to the one shown in [Figure 3-1](#):

User Profile							
User ID	JOHN	JOHN DOE	Group	ROOT			
Email	john@company.com			Project	4		
Telnet	SB Client						
Roles				Auth Class			
PROGRAMMER				CH Car Hire			
QA				PRC PRC			
Realms		Clearance		Default			
D	PRC Internal	Devel 3	No Restrictions	1	ENH	@SYSID	
DEV	Development	2.3	Restricted (No Con	1	ENH	@SYSID	
Use F6 to review and modify user preferences.							
F2-Save F4-Del F5-Assign F6-Prefs F7-System F8-Restr F9-Status F10-Action							

Figure 3-1: User Profile screen

Explanations of the fields are as follows:

User ID

For convenience, it is recommended that the user ID should match that same user's login ID.

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

Name	The user name will be pulled from the SB+ profile that must be previously set up.
Roles	<p>The code or ID that represents programmer/analyst/manager classifications and their duties in relation to project revision control.</p> <p>For more information on defining roles codes, see “Roles Codes” on page 36.</p> <p>Press [F3] to select from the list of valid entries. You can tag multiple items from this lookup window.</p>
Auth Class	<p>Modules or categories of projects on which this particular user is allowed to work.</p> <p>Press [F3] to select from the list of valid entries.</p>
Realms	<p>The realm or realms into which this user may log with the appropriate security. Enter the common working realm <i>first</i>, as it becomes the default realm.</p> <p>Press [F3] to select from the list of valid entries.</p>
Clearance	<p>This user’s security clearance level for each realm on the system. Indicates whether the user may: log into the realm, make changes, under what conditions they may make changes, and so on.</p> <p>If you have not had training or you are in doubt as to what the setting should be, the most prudent and common setting is 2.3. This value allows changes to be made to any software that is not checked out to someone else.</p> <p>Press [F3] to select from the list of valid entries.</p>
Prio	<p>The default priority for projects that this user opens.</p> <p>For more information on defining priority codes, see “Priority Codes” on page 28.</p> <p>Press [F3] to select from the list of valid entries.</p>
Type	<p>The default project type for projects this user opens on the “quick” project screen. It is recommended that you enter the most common choice here.</p> <p>Press [F3] to select from the list of valid entries.</p>
Class	The default class for projects created by this user. You can always modify this value later. You can also enter @SYSID or @ACNT.NAME .

Press **F3** to select from the list of valid entries.

2. Fill in each field with the appropriate information.
If you press **F5** (Assign), see [“Assigning Projects to the User”](#) on page 85.
If you press **F6** (Prefs), see [“Defining User Preferences”](#) on page 87.
If you press **F7** (System), see [“Setting Up System Clearances”](#) on page 93.
If you press **F8** (Restr), see [“Defining User Restrictions”](#) on page 94.
If you press **F9** (Status), see [“Setting Up Status Clearances”](#) on page 95.
3. Press **F2** (Save).

Some of the preferences that are elected by individual user may require a general consensus from the development group. Defaults will apply if no preferences are selected.

You may set up a user named “default” and if specific users are not set up differently, these settings will be used. These can also be defined by group (ROOT*DEFAULT, and so on).

Note

Each user must have an individual SB+ User ID—even if default settings are used in PRC.

Assigning Projects to the User

Perform the following:

1. Press **F5** (Assign) from the User Profile screen ([Figure 3-1 on page 83](#)).
The system displays the User Project Assignments screen, similar to the one shown in [Figure 3-2](#):

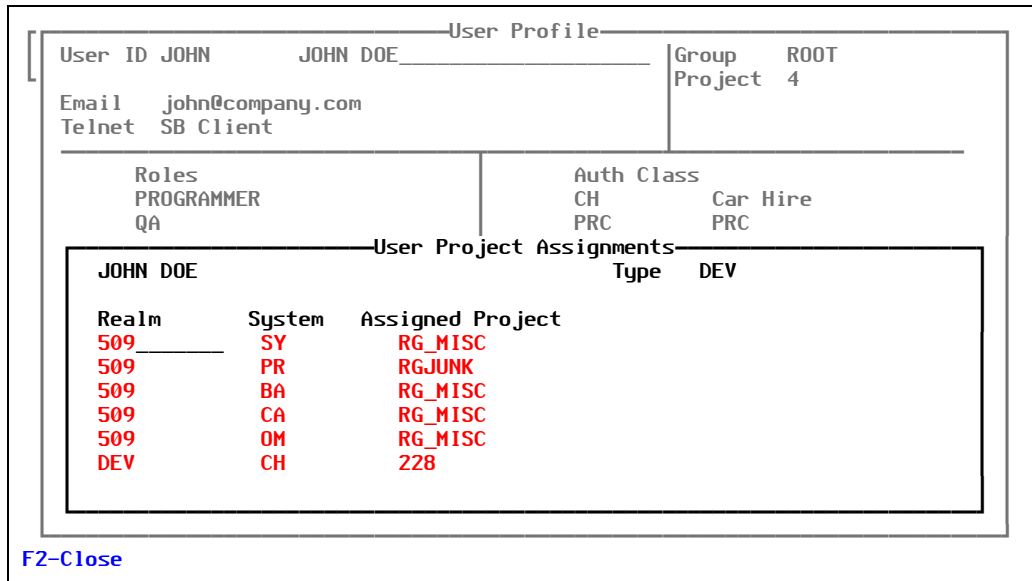


Figure 3-2: User Project Assignments screen

Explanations of the fields are as follows:

Realm The realm or realms into which this user may log with the appropriate security. Works in conjunction with the *System* field to allow you to assign a specific project to this user.

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

System The system or systems into which this user may log with the appropriate security. Works in conjunction with the *Realm* field to allow you to assign a specific project to this user.

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

Assigned Project A project tied to this user in the *Realm* and *System* specified.

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

Type A type of user profile. Valid entries are:

- **DEV** — Development
- **EP** — Extended Privileges
- **LITE** — No PRC Privileges

2. Fill in each field with the appropriate information.
3. Press **[F2]** (Close).

The system returns you to the User Profile screen ([Figure 3-1 on page 83](#)).

Defining User Preferences

Perform the following:

1. Press **F6** (Prefs) from the User Profile screen (Figure 3-1 on page 83).

The system displays the User Preferences screen, similar to the one shown in Figure 3-3:

```

User Profile
-----
User ID JOHN      JOHN DOE      |Group  ROOT
-----
Pref HOST Editor SE_____

Desktop editors/merge utility:
Editor cmd path
DIFF cmd path
Work Dir (full path)
Work Dir (filename)
Documenting on?  No          Edit backups/day      3
Daily, Cont, Prj (D/C/P) D  No. of stacks (days) 3
-----
Project related settings-----
Create NEW Projects?  Yes          PRJ by Port?         No
SUB projects?        Yes          Clear (forget) PRJ  No
-----
Knit Default Settings-----
Key Management        No          Identical items?    No
Side (primary)       Left        Knit Workspace      No
Command Set          STANDARD    Lookahead            50
-----
F2-Close F6-Infreq

```

Figure 3-3: User Preferences screen

Explanations of the fields are as follows:

Preferred Editor

The editor to be launched when PRC calls an editor automatically under a few circumstances. Other editors are still available—this is just the default. Valid entries are:

- **ED** — Pick Line Editor
- **SE** — SB+ Screen Editor
- **VI** — Unix (or UV, Unix) VI
- **AE** — Unidata Editor
- **EDIT-LIST** — Edit list
- **EDU** — Universe Editor
- **FROG** — Frog
- **PEGASYS** — Pegasys (LaBonte)
- **JED** — Getronics
- **WIN.EDIT** — WIN.EDIT
- **EDIT.PLUS** — EDIT.PLUS
- **TED** — TED

- **BE** — BE
- **EDIT** — EDIT

Press **F3** to select from the list of valid entries. This list is limited by the editors defined in the PRC Preferences screen.

Email Address

This user's e-mail address, in either the `<username>` or `<username@full.address>` format.

If you leave this field blank, the SystemBuilder/PRC USER.ID is used, @SERVERNAME (for more information, see preferences API screen).

Edit backups/day

Indicates the number of backed up copies of each item that should be made each day. Valid entries are:

- **0** through **99** — Integers only determine the number of times a backup is to be made
- **+1** — The first daily backup only

Kept only during the active life of the project.

No. of stacks (days)

A stack of changed items for each day by each user (regardless of assigned project) in reverse chronological order. The number in this field determines how many days back a stack is retained.

Documenting on?

Indicates whether documentation by the programmer is required whenever the programmer files an item by pressing **F2**. It allows the programmer to enter a description of the change while the knowledge is still "fresh." Valid entries are **Y** (Yes) and **N** (No).

Press **F3** to select from the list of valid entries.

Daily, Cont. or by PRJ?

Defines how often you will be prompted for a document line (works in conjunction with the *Documenting on?* field above). Indicates how often the system should prompt the programmer for documenting the item.

If documenting is off (the value of the *Documenting on?* field is **N**), you are not prompted at all. If documenting is on, you can have it prompt once per day (per thing you change) or once per project (per thing you change) or every single time you change a thing. Valid entries are:

- **D** — Once per day, for each item you change
- **P** — Once per project, for each item you change
- **C** — Continuous, every time you make a change

For example, you open project “7” and edit the program “apple.” When you file it, PRC will ask you for a document line. If you edit “apple” again on the same day, the system will not ask for a document line again if you have it set to **D** or **P**. If you edit “apple” the next day and have this field set to **D**, PRC will ask again; however, if you have it set to **P**, PRC will not ask again, as it already has a document entry for this project.

Press **F3** to select from the list of valid entries.

Flick/Remove?

Indicates whether this user is allowed to remove items from a project—*unaudited*—by flicking the item to the same project. Valid entries are **Y** (Yes) and **N** (No).

Press **F3** to select from the list of valid entries.

Conflict Default

Indicates the standard resolution of any item that comes in conflict. You can change the method of conflict resolution—this is simply the default. Valid entries are:

- **T** — Thinking
Used throughout the software to prevent “accidental choices.” It is a way to make sure you do not hit **Enter** and make a choice that you regret. It is not actually a choice; instead, it forces you to choose one of the other options by entering its hot key or by arrowing over to it.
- **C** — Copy (branch)
- **O** — Overwrite

Press **F3** to select from the list of valid entries.

Check from Realm

The name of the realm from which the user can check out (or compare) the item.

Create new Projects?

Indicates whether this user is allowed to create new projects “on the fly” from within the working or tracked account. Valid entries are **Y** (Yes) and **N** (No).

Press **F3** to select from the list of valid entries.

PRJ by Port?

Indicates whether to track individual sessions against a project. Valid entries are:

- **Y** — Yes, track individual sessions against a project
This option will track your project by what port you are logged in on.
For example, you log into the “RollerBladesRUs” account on project “8” today via port 15. You come in the next day and log into that same account—but you happen to be on port 17

this time—it is *not* going to put you on project “8.” Instead, it is going to put you on the project to which you were last assigned when you were in on port 17.

- **N** — No, track the whole user-realm session
- **A** — By work account

If you work on projects across multiple accounts (that are in the same realm), this option will track what project you are on in each account.

For example, you have a central software account and then dozens of accounts that point to that software for each of the customers that are on that version. This option allows you to be in the “ACME Furniture” account on project “7” and in the “RollerBladesRUs” account on project “8.” You can go home, come back, log in, and get a different port number—but when you log into “RollerBladesRUs,” PRC remembers project “8.”

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

Clear project at Menu?

Indicates whether to clear the currently-assigned project number each time the user logs in or changes systems. In other words, do you want the system to forget the last project the user was in when the user logs back on?

Valid entries are **Y** (Yes) and **N** (No).

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

Pull Help?

Do you want all help items that are attached to any item on your project to automatically pull in to the project? Valid entries are **Y** (Yes) and **N** (No).

Answering “yes” can have ramifications.

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

Automatic timer?

Indicates whether to enable an automatic timer for each project whenever the user is assigned to one. Valid entries are **Y** (Yes) and **N** (No).

If you answer “yes,” the system records OFF when the user logs off or is assigned to another project.

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

Checkout Mode?

Indicates whether the system requires software to be checked out in advance of any changes being filed. Valid entries are **Y** (Yes, enable Checkout Mode) and **N** (No).

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

Suppress Mult Reminders?

Indicates whether the user will *not* be warned of a *particular* conflict a second time until rollout—in other words, the user is warned only once. Valid entries are:

- **Y** — Yes, suppress further warnings
- **N** — No, warn the user more than once

Press **F3** to select from the list of valid entries.

Compare before Tracking?

Indicates whether the system will compare this item when it is filed (the programmer presses **F2**) and only track it if it changed, or track the item regardless of any changes.

Note: Due to SB+ limitations, Screen, Report, and File Update definitions are currently tracking item whether or not they have changed.

Under normal circumstances, you would probably wish to track only those items that have changed. It is sometimes desirable to be able to “thumb through” some source items, pressing **F2** to “tag” them onto the project. Thumbing through items is particularly useful for projects that were already underway when PRC was installed.

Valid entries are **Y** (Yes) and **N** (No).

- **Y** — Yes, compare the item and track it if changes are made
- **N** — No, simply track the item, regardless of changes

You may want to select **N** at first, then change the value to **Y** once all of your “old projects” have been entered and all source items marked.

Press **F3** to select from the list of valid entries.

Lib/Sandbox

Indicates whether a special mode is enabled. The “sandbox” mode redirects all program edits into a personal copy or file, depending on settings on the PRC Preference screen (see [“PRC Preferences” on page 162](#)). Valid entries are:

- **Y** — Yes, enable the sandbox
- **N** — No, leave sandboxing disabled
- **C** — Conflicts only

Many programmers have worked outside the SB+ environment (and similar environments) and are comfortable with taking copies of programs to make changes. However, you may find this feature to be a cumbersome way in which to work. Before answering “yes,” please discuss this subject with your configuration manager or with SJ+ Systems Associates.

Press **F3** to select from the list of valid entries.

Ask To Track?

Indicates whether the system will ask “do you wish to track this item” each time the user changes an item. Valid entries are:

- **Y** — Yes, ask about tracking changes first
- **N** — No, let the system automatically track changes

Press **F3** to select from the list of valid entries.

Key Management

Indicates whether you wish to store multiple named sessions or the one from the last Knit session. Valid entries are:

- **Y** — Yes, store multiple named sessions
- **N** — No, store the last KNIT session under your user ID/port

Press **F3** to select from the list of valid entries.

Side (primary)

The side on which the system will place the *real* item—the one into which you will edit—when KNIT is called automatically. Valid entries are:

- **L** — Left side
- **R** — Right side

Press **F3** to select from the list of valid entries.

Identical items?

Indicates whether KNIT should display two items, even if it thinks they are identical? Valid entries are **Y** (Yes) and **N** (No).

Press **F3** to select from the list of valid entries.

Knit Workspace

Indicates whether KNIT should write the items into a workspace until you indicate they should be written back out to the real files (it will ask). Valid entries are **Y** (Yes) and **N** (No).

Press **F3** to select from the list of valid entries.

Lookahead

The number of lines to “look ahead” when **Insertall** is used in KNIT.

Command Set

The table of personalized commands to be used by the KNIT routine. You can define a set as explained in “[Knit Commands](#)” on page 41. You may reference a table here, your own or that of your group or company.

Press **F3** to select from the list of valid entries.

2. Fill in each field with the appropriate information.
3. Press **F2** (Close).

The system returns you to the User Profile screen (Figure 3-1 on page 83).

Setting Up System Clearances

Perform the following:

1. Press **[F7]** (System) from the User Profile screen (Figure 3-1 on page 83).
The system displays the System Clearances screen, similar to the one shown in Figure 3-4:

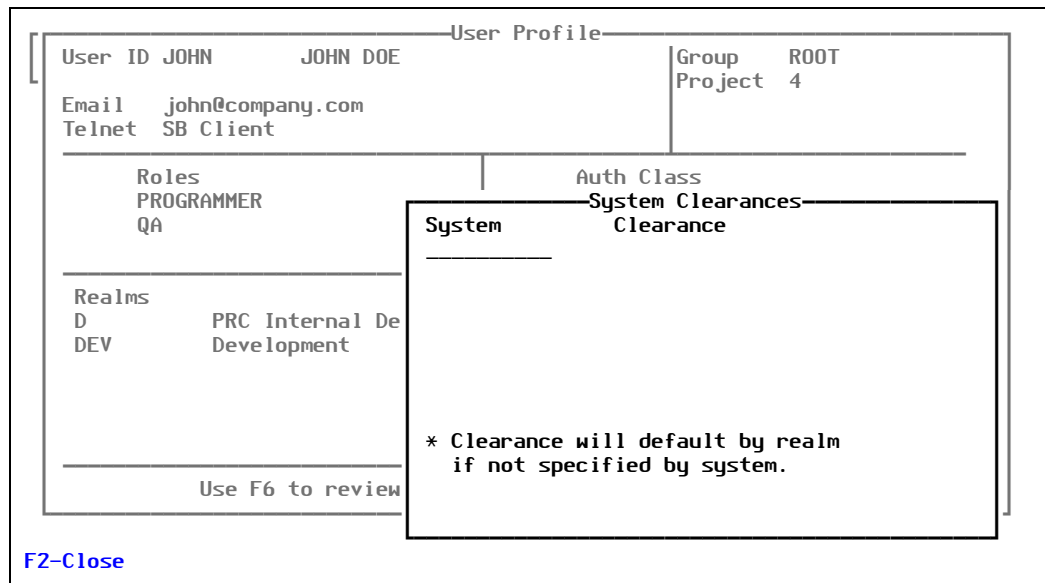


Figure 3-4: System Clearances screen

Explanations of the fields are as follows:

System The system identifier, or SYSID.

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

Clearance This is the same clearance information as on the front page of the user profile, but is a way to refine the clearance restrictions.

For example, you could define this person as having a level **3** clearance in **DEV**—except on **GL** projects, for which they have level **2.3**.

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

2. Fill in each field with the appropriate information.
3. Press **[F2]** (Close).
The system returns you to the User Profile screen (Figure 3-1 on page 83).

Defining User Restrictions

Restrictions are associated by realm. You must use call this feature from a realm line.

PRC comes with certain inherent restrictions. This screen allows you to either remove restrictions (un-restrict or un-track) or increase restrictions (super-restrict or disallow) certain functions or executable commands.

For those who are unfamiliar with SB+, these functions are typed with the slash; for example, when you want to modify a field, you type **/FD**. Therefore, to indicate that you want to restrict or un-tracked the “modify field” command for this user, you would type **/FD** in one of the columns on this sub-screen of the user profile.

Perform the following:

1. Press **[F8]** (Restr) from the User Profile screen (Figure 3-1 on page 83).
The system displays the User Restrictions screen, similar to the one shown in Figure 3-5:

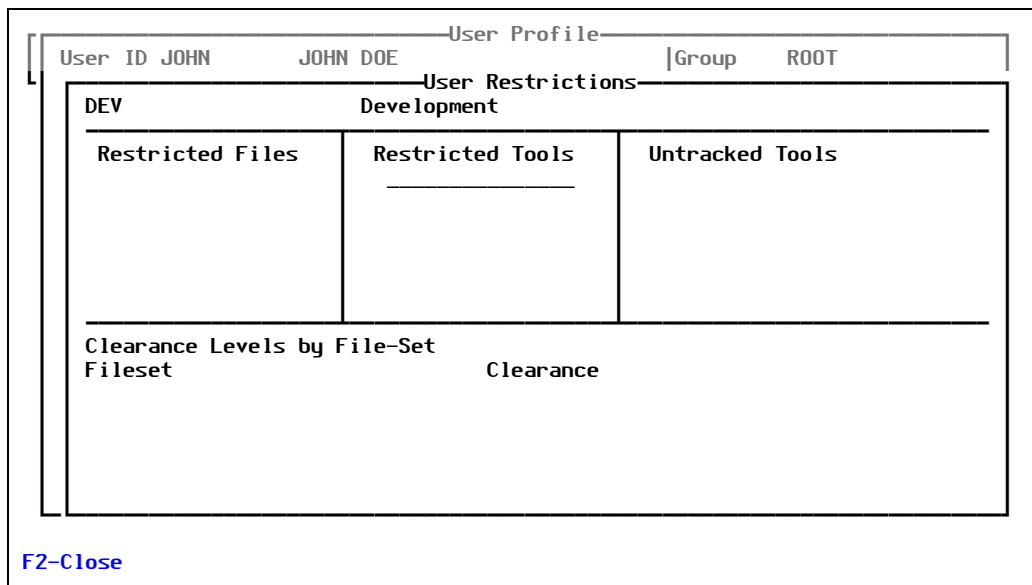


Figure 3-5: User Restrictions screen

Explanations of the fields are as follows:

Realm The realm in which this user is restricted from the following files, tools, and so on.

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

Restricted Files Files in this realm that this user is not allowed to use or modify.

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

Restricted Tools	<p>The executable file that launches tools or utilities that this user is not allowed to use in this realm.</p> <p>Press F3 to select from the list of valid entries.</p>
Untracked Tools	<p>Functions or tools that will be ignored by PRC, and for which changes will be neither prevented nor tracked.</p> <p>Note to SB+ Users Only: Instead of naming an executable command, you can use line 1 names for DEFN items. SB stores a bunch of different types of things all in the XXDEFN file. It puts a name on line <1> of those items; for example, D for dialog box and ARD for access report definition. To un-track access reports (a very common item that organizations “un-track”), enter ARD in this column.</p> <p>Press F3 to select from the list of valid entries.</p>
Fileset	<p>A fileset to be defined for a particular clearance level.</p> <p>SystemBuilder tools are in the <code>SOFTWARE</code> fileset.</p> <p>Press F3 to select from the list of valid entries.</p>
Clearance	<p>The code or ID that represents the clearance level associated with only those files in specified in the <i>Fileset</i> field for this user in this realm.</p> <p>Press F3 to select from the list of valid entries.</p>

2. Fill in each field with the appropriate information.
3. Press **F2** (Close).
The system returns you to the User Profile screen ([Figure 3-1 on page 83](#)).

Setting Up Status Clearances

Perform the following:

1. Press **F9** (Status) from the User Profile screen ([Figure 3-1 on page 83](#)).
The system displays the User Status Clearances screen, similar to the one shown in [Figure 3-6](#):

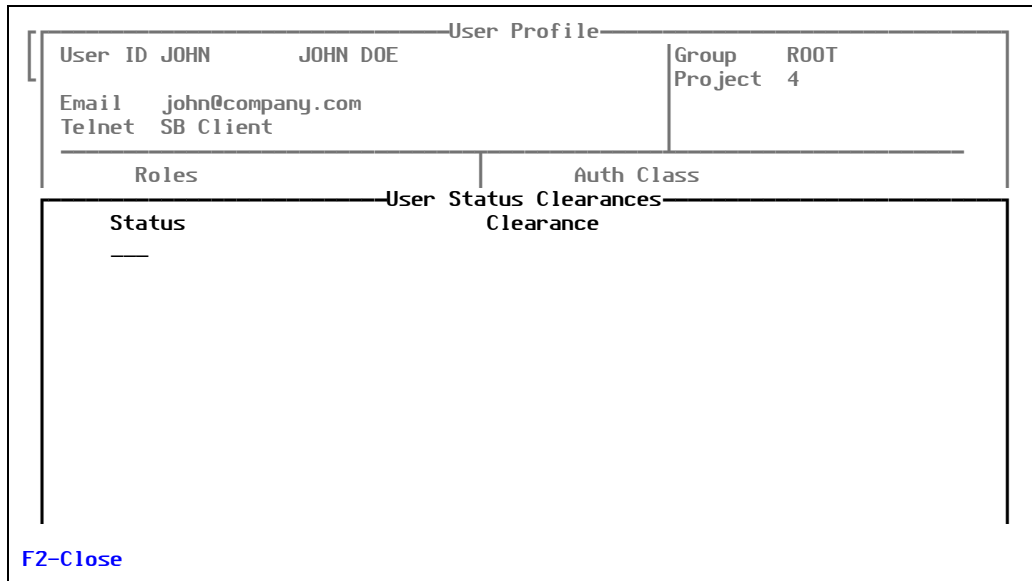


Figure 3-6: User Status Clearances screen

Explanations of the fields are as follows:

Status The code or ID of the status of the *other* conflicting project.

Press **F3** to select from the list of valid entries.

Clearance The clearance that this user will have when confronted with a conflict with another project in this clearance. Basically, whether this user can overwrite or not. Overwriting refers to when you have a conflict.

For example, you try to edit the program “apple” and PRC informs you: “Sally Jo has apple checked out on project 19.” The user’s clearance determines what happens next. If you are allowed to overwrite, you can go ahead and make changes to “apple,” right along with Sally Jo.

Press **F3** to select from the list of valid entries.

2. Fill in each field with the appropriate information.
3. Press **F2** (Close).

The system returns you to the User Profile screen ([Figure 3-1 on page 83](#)).

Timesheets

Developers use timesheets, filling them in by date, to post time being spent on individual projects. The time logged here is reflected in the *Actual time* column on the project.

To work with timesheets, perform the following:

1. Select **Files > Timesheet**.

The system displays the Daily Timesheet screen, similar to the one shown in [Figure 3-7](#):

Daily Timesheet			
Date	11/12/2005		
User			
Project Diary/Details	Activity	Time (Begin End)	Elapsed
			Total Elapsed

F2-Save F4-Del

Figure 3-7: Daily Timesheet screen

Explanations of the fields are as follows:

- Date** The date of the time sheet being created or reviewed.
Press **[F3]** to select a date from the pop-up calendar.
- User** The user ID of the individual entering or reviewing this timesheet.
- Project** The project number or issue number against which you are going to post time spent.
Press **[F3]** to select from the list of valid entries. It lists all issues and/or projects that have been worked on “today”:
- Issues — The issue was created today
 - Projects — The individual assigned herself or himself to work on this project (via **/PRC**) today

Activity	<p>The code or ID that represents the activity to which this time is charged. The activity must:</p> <ul style="list-style-type: none">• Match the document type For example, development cannot be charged to a phone call.• Be authorized to the client, if applicable <p>The Client Master has a subscreen for authorizing and allocating activity codes. Currently, you are unable to restrict an activity code to a type of charging reference document (call, issue, project).</p> <p>Press F3 to select from the list of valid entries.</p>
Diary/Details	<p>A <i>brief</i> description of work performed on the project on this date.</p>
Begin, End	<p>[<i>Informational only</i>] The time at which work on this project began and ended for this day.</p>
Elapsed	<p>Elapsed time spent on this project for this time slot on this day.</p>

2. Fill in each field with the appropriate information.
3. Press **F2** (Save).

Action Menu

PRC clumps several features and utilities that perform particular actions you may need to perform. These actions include global searches, project ranking, and working with ravel files.

When you select **Action**, the system displays the Action menu, as shown in [Figure 4-1](#):

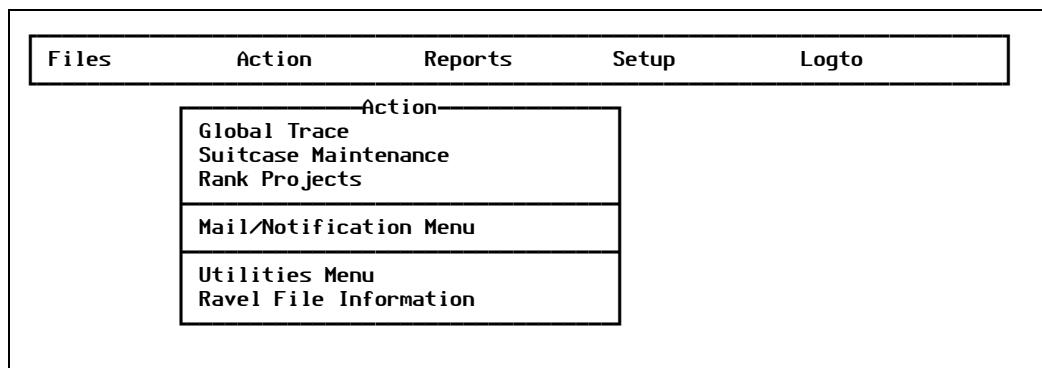


Figure 4-1: Action menu

For explanations of the **Utilities Menu** options, see [Chapter 5, "Utilities."](#)

Global Trace

When you do not know who made a change, when the change was made, or even *where* the change was made, this menu option provides the answer. This option is located in the PRC account to perform a global trace because it transcends individual realms or modules. The feature searches for any portion of the name of a changed item.

For example, you want to find out when the APPLE program in the BP file changed:

- If you want the results to be specific, you trace for **BP!APPLE**.
- If you want to find anything called APPLE in the file or item name, you simply trace for **APPLE**.
- If you know the item you want to find is in BP and starts with the letter A, you search for **BP!A**.

To work with Global Trace, perform the following steps:

1. Select **Action > Global Trace**.

The system displays the Global Trace screen, similar to the one shown in [Figure 4-2](#):

Global Trace			
(partial) file or item name: Search String menu			
Source items found on projects	Checked out on		
	*Cur Prj	*Past	PRJ
MD!CHMENU!BATCH.REG.RPT.MENU	1		5
MD!CHMENU!PA.AST.DOWNLOAD.MENU	1		1
MD!CHMENU!BATCH.RPT.MENU	1		1
ED!RG.BP!FIND.ITEM.IN.MENUS	1		1
MD!CHMENU!ORDER.PROCESSMENU	1		1
PD!CHMENU!OR.GREATCLIPS.MENU	1		0
PD!CHMENU!REBUILD.MENU.DEFNS	1		0
ED!QA.BP!UTILITY.REPAIR.FILE.MENU	1		0
MD!SYMENUS!VAR.UTILS.MENU	1		0
MD!CHMENU!AP.REG.RPT.MENU	1		5
MD!CHMENU!CA.REPORT.MENU	0		3
MD!CHMENU!EMP.INQUIRY.MENU	0		2
MD!CHMENU!CUST.RPT.MENU	0		2
vMD!CHMENU!SUM.SALES.RPT.MENU	0		2

F2-Leave F5-*Drill

Figure 4-2: Global Trace screen

Explanations of the fields are as follows:

Search String The string for which you are searching.

Cur Prj The number of projects that *currently* have the item on this row checked out. Works in conjunction with the rows of the *Source items* column.

To see details about any current project, place the cursor on a particular project and press **F5** (Drill) to “drill down” for more information.

Past Prj The number of projects that have historically had this item (on this row) checked out. Works in conjunction with the rows of the *Source items* column.

To see details about any past project, place the cursor on a particular project and press **F5** (Drill) to “drill down” for more information.

Source items The results of your search are listed in this column. Each one is a source item contains or uses the value in the *Search String* field.

2. Fill in each field with the appropriate information.

- If you press **F5** (*Drill) from either of the *Curr Prj* column or *Past Prj* column, the system “drills down” into that item’s history to present either the current project to which this item is checked out or the projects in the past to which it was checked out, respectively. The system displays the information in the format shown in the example in [Figure 4-3](#):

Global Trace

(partial) file or item name:
Search String menu

HISTORICAL Projects For MD!CHMENU!BATCH.REG.RPT.MENU

Project	Realm	Status	Last touched Date	Programmer
PRJ*54662	DEV	8	12/02/05	JOHN
PRJ*33992.1	DEV	8	01/30/06	SUSAN
PRJ*220.CU	DEV	8	05/04/06	JOHN
PRJ*1079.CU	DEV	8	10/27/06	JOHN
PRJ*101224	QA	7	08/01/06	SUSAN

F2-Leave F5-Drill PRJ F6-Drill Item

Figure 4-3: Drill down sub-screen from Global Trace

- You can drill down further for each project or item using **F5** (Drill PRJ) or **F6** (Drill Item), respectively, or press **F2** (Leave) to return to the Global Trace screen.
- Press **F2** (Leave) when you are done with Global Trace.

Suitcase Maintenance

A “suitcase” is that collection of items/components you want to deliver that are connected by dependencies. The Suitcase Maintenance feature provides you a way to gather those components.

The concept is similar to that of a Version Master (see [page 60](#)), which is a roster/index/table of contents of all projects that must be delivered together in order to deliver the desired (sought) item or project.

To set up project suitcases, perform the following:

- Select **Action > Suitcase Maintenance**.

The system displays the Project Suitcase screen, similar to the one shown in [Figure 4-4](#):

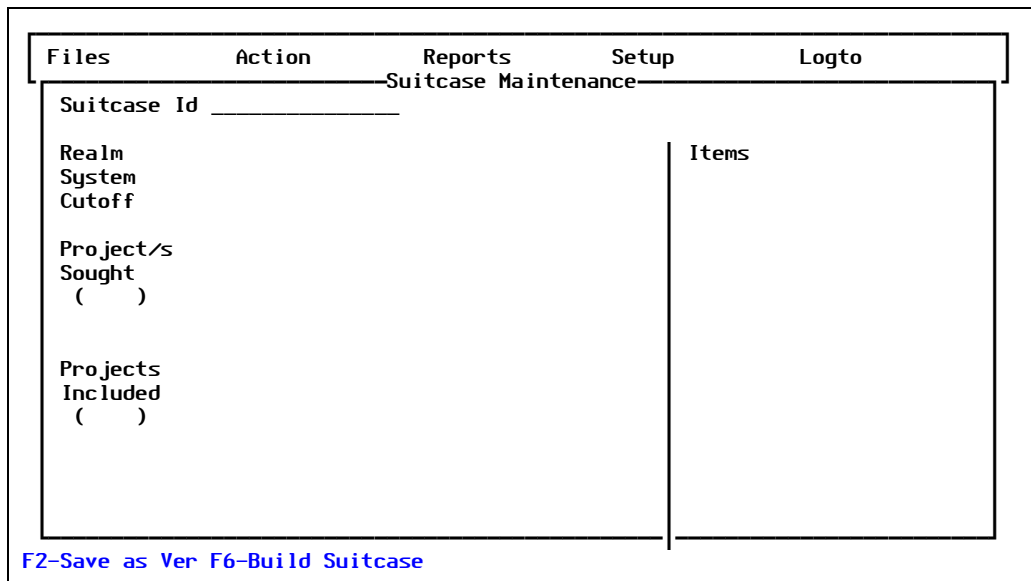


Figure 4-4: Suitcase Maintenance screen

Explanations of the fields are as follows:

- Suitcase ID** The code or ID that represents a project “suitcase” or collection of sub-projects.
- Realm** The realm in which work for this version is being performed.

Press **[F3]** for a list of valid realms.
- System** The name of the SB+ system (software module) for which this is a version definition.

Press **[F3]** for a list of valid system IDs.
- Cutoff** The date on which this suitcase is no longer available for adding projects. Once the cutoff date is reached, you are not able to modify its contents.

Press **[F3]** for a list of valid cutoffs.
- Project/s Sought** The project or projects you wish to deliver. PRC will find any others that need to go with this suitcase based on dependencies.

Press **[F3]** for a list of valid project IDs.
- Projects Included** The code or ID of the project or projects that are already included in this suitcase and set aside for delivery.
- Items** The items included in a project.

2. Fill in each field with the appropriate information.
3. If you press **F6** (Build Suitcase), the system launches the build automatically when you add a project. You can also use **F6** to rebuild if you are removed or changed or want to, for whatever reason. This function will be documented with a tip sheet in the near future.
4. Press **F2** (Save as Ver).

Rank Projects

Project ranking in PRC is a way to gather all projects of a given priority so you can sort them into a particular order. This, in turn, allows you to create listings ranked by importance.

You can recall them to the ranking screen any time in the future and re-order them. All projects that are ranked must be of the same priority and have the same status.

To work with ranking projects, perform the following:

1. Select **Action > Rank Projects**.

The system displays the Rank Projects in Priority screen, similar to the one shown in [Figure 4-5](#):

Rank Projects in Priority						
Priority	1	Urgent	Status	3	Assigned to	JDOE
*Prj Nbr	Last Touched	No. Items	Assn	Rank		
PRJ*101 WEB FILE DOWNLOAD	08/31/05	1	JDOE	-1		
PRJ*102 ACT Process	08/19/05	1	JDOE	-1		
PRJ*104 PAL REPORT	09/26/05	0	JDOE	-1		
PRJ*112 PRIOR-YEAR REPORT	10/26/06	4	JDOE	-1		
PRJ*120 FILE AUDITING	09/28/06	5	JDOE	-1		
PRJ*126 FIELD TESTING	08/15/06	0	JDOE	-1		

F2-Save F4-Del F5-*Drill F6-Auto Rank F7-Clear Rank F8-Move Rank

Figure 4-5: Rank Projects in Priority screen

Explanations of the fields are as follows:

Priority

The priority of projects you are ranking. The list of projects will be restricted to projects that match the priority you specify.

Press **[F3]** for a list of valid priority codes.

Status

The state or states of the projects you are ranking. The list of projects will be restricted to projects that match the status you specify.

Press **[F3]** for a list of valid status codes. You can select more than one.

Assigned to

The user ID to which this ranking is restricted within projects assigned to that person.

If you leave this field blank, the ranking applies to all projects for the status and priority.

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

Prj Nbr

The projects that match the priority and status. The projects to be ranked.

To see details about any project in this column, place the cursor on a particular project and press **[F5]** (Drill) to “drill down” for more information.

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

Last Touched

[Informational only] The date on which this item was last “touched” or modified.

2. Fill in each field with the appropriate information.
 - If you press **[F5]** (*Drill), see “[Drill Down](#)” on page 104.
 - If you press **[F6]** (Auto Rank), see “[Auto Rank](#)” on page 105.
 - If you press **[F7]** (Clear Rank), see “[Clear Rank](#)” on page 106.
 - If you press **[F8]** (Move Rank), see “[Move Rank](#)” on page 106.
3. Press **[F2]** (Save).

The *No. Items*, *Assn*, and *Rank* fields are informational only.

Drill Down

From the Rank Projects in Priority screen ([Figure 4-5 on page 103](#)), if you press **[F5]** (*Drill) with the cursor in the *Prj Nbr* field, the system “drills down” to find the background information for that project.

The system displays the Project Master Maintenance screen (see [Figure 1-15 on page 46](#)). After you are done reviewing the project information, press the **[Esc]** key and select **No** when the system asks if you want to keep any changes.

For more information on the Projec Master, see "Project Master" on page 45.

Auto Rank

From the Rank Projects in Priority screen (Figure 4-5 on page 103), if you press **F6** (Auto Rank), the system displays the question shown in Figure 4-6:

Rank Projects in Priority					
Priority	1	Urgent	Status	3	Assigned to JD0E
*Prj Nbr		Last Touched	No. Items	Assn	Rank
PRJ*101		08/31/05	1	JD0E	-1
WEB FILE DOWNLOAD					
PRJ*101688		<div style="border: 1px solid black; padding: 5px;"> <p>[PRC.RANK.AUTO.D] PRC will now automatically rank the projects listed in the order which they are now displayed.</p> <p>Yes, Please No, Thank you</p> </div>			-1
ACH Build file					-1
PRJ*101695					-1
PAY REGISTER R					
PRJ*112		10/26/06	4	JD0E	-1
PRIOR-YEAR REPORT					
PRJ*120		09/28/06	5	JD0E	-1
FILE AUDITING					
PRJ*126		08/15/06	0	JD0E	-1
FIELD TESTING					
F2-Save F4-Del F5-*Drill F6-Auto Rank F7-Clear Rank F8-Move Rank					

Figure 4-6: Automatically rank projects question

If you select the **No, Thank you** option, the system returns you to the Rank Projects in Priority screen without sorting anything.

However, if you select **Yes, Please**, the system generates a ranking for each project, according to the order in which they appear on the screen. PRC automatically changes the values of the *Rank* column, as shown in the example in Figure 4-6:

Rank Projects in Priority						
Priority	1	Urgent	Status	3	Assigned to	JDOE
*Prj Nbr	Last Touched	No. Items	Assn	Rank		
PRJ*101 WEB FILE DOWNLOAD	08/31/05	1	JDOE	1		
PRJ*102 ACT Process	08/19/05	1	JDOE	2		
PRJ*104 PAL REPORT	09/26/05	0	JDOE	3		
PRJ*112 PRIOR-YEAR REPORT	10/26/06	4	JDOE	4		
PRJ*120 FILE AUDITING	09/28/06	5	JDOE	5		
PRJ*126 FIELD TESTING	08/15/06	0	JDOE	6		

F2-Save F4-Del F5-*Drill F6-Auto Rank F7-Clear Rank F8-Move Rank

Figure 4-7: Ranked projects

Clear Rank

From the Rank Projects in Priority screen (Figure 4-5 on page 103), if you press **F7** (Clear Rank), the system changes all of the values in the *Rank* column to **-1**. This change effects all of the projects in the list.

Move Rank

From the Rank Projects in Priority screen (Figure 4-5 on page 103), if you press **F8** (Move Rank), the system allows you to move one item at a time to a different spot in the order.

For example, your cursor is in the project at the bottom of the screen, currently ranked #6. When you press **F8** (Move Rank), the system displays a *Move to (position)* field at the bottom of the screen, as shown in Figure 4-8:

Rank Projects in Priority						
Priority	1	Urgent	Status	3	Assigned to	JDOE
*Prj Nbr	Last Touched	No. Items	Assn	Rank		
PRJ*101 WEB FILE DOWNLOAD	08/31/05	1	JDOE	1		
PRJ*102 ACT Process	08/19/05	1	JDOE	2		
PRJ*104 PAL REPORT	09/26/05	0	JDOE	3		
PRJ*112 PRIOR-YEAR REPORT	10/26/06	4	JDOE	4		
PRJ*120 FILE AUDITING	09/28/06	5	JDOE	5		
PRJ*126_____	08/15/06	0	JDOE	6		

Move to (position) 1_
 F2-Save F4-Del F5-*Drill F6-Auto Rank F7-Clear Rank F8-Move Rank

Figure 4-8: Move to position field

Type the number of the position to which the project should be ranked and press **Enter**. The system ranks the projects accordingly and displays them in the new order, as shown in Figure 4-9:

Rank Projects in Priority						
Priority	1	Urgent	Status	3	Assigned to	JDOE
*Prj Nbr	Last Touched	No. Items	Assn	Rank		
PRJ*126_____	08/15/06	0	JDOE	1		
PRJ*101 WEB FILE DOWNLOAD	08/31/05	1	JDOE	2		
PRJ*102 ACT Process	08/19/05	1	JDOE	3		
PRJ*104 PAL REPORT	09/26/05	0	JDOE	4		
PRJ*112 PRIOR-YEAR REPORT	10/26/06	4	JDOE	5		
PRJ*120 FILE AUDITING	09/28/06	5	JDOE	6		

F2-Save F4-Del F5-*Drill F6-Auto Rank F7-Clear Rank F8-Move Rank

Figure 4-9: Moved project to new position

E-mail & Notifications

When you select **Action > Mail/Notification Menu**, the system displays the following:

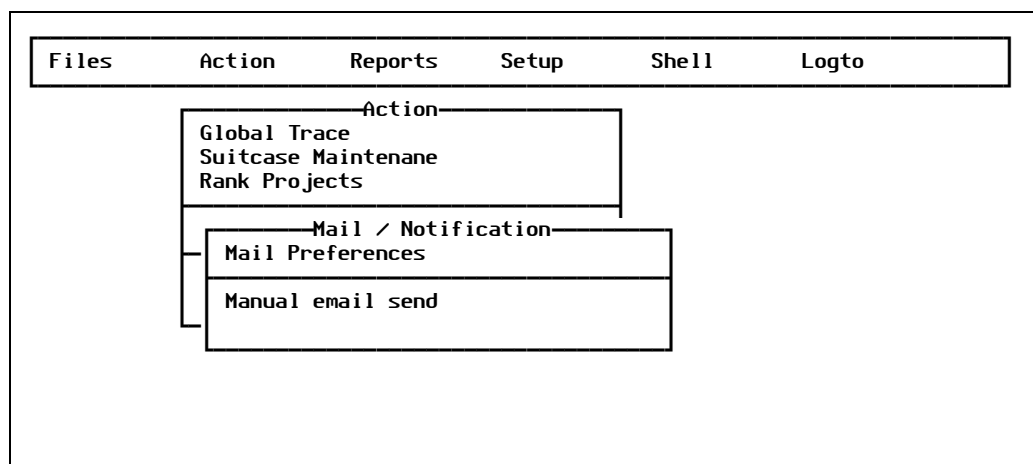


Figure 4-10: Mail/Notification menu

This section explains these two menu options, which make up most of the settings and options for automatic e-mail notification. The other screen that identifies recipients and the mail text to be sent is the Status Codes Maintenance screen, in which you define the following fields:

- *Type*
- *Send Mail To*
- *Mail Text*
- *Mail Attachments*

For more information, see [“Status Codes” on page 29](#).

Mail Preferences

To set up Internet e-mail capabilities, perform the following steps:

1. Select **Action > Mail/Notification Menu > Mail Preferences**.

The system displays the PRC Notification screen, similar to the one shown in [Figure 4-11](#):

PRC Notification	
Email Program	BLAT
Full Command Path	C:\windows\system32\blat.exe
Server	company.com
Sender	PRCAdmin@company.com
Path To PRC.MAILTEXT	PRC.MAILTEXT
Delimiter	; (for multiple recipients)
<hr/>	
F2-Save F4-Del F5-Events F6-Text F7-Recipients	

Figure 4-11: PRC Notification screen

Explanations of the fields are as follows:

Email Program	The mail service you wish to use. Valid entries are: <ul style="list-style-type: none"> • SB+ — the default • BLAT — for Windows • SENDMAIL — for Unix
Full Command Path	The entire path for the command to be executed for the e-mail service defined in the <i>Email Program</i> field.
Server	The name of the mail server.
Sender	The default e-mail address to use as the “from user” for the notifications. On some systems, the “from” e-mail address must be a real address; on other systems, you can use a dummy address.
Path to	The path to the text template to use. The text templates for the e-mail content are stored, by default, in a directory type file in the PRC account called PRC.MAILTEXT . If you wish to use a different location, enter the path to the template here.
Delimiter	The character the system should recognize as the separator between e-mail addresses for multiple recipients.
Open area	The text part of a mail message can be laid out here.

- Fill in each field with the appropriate information.
 - If you press **F5** (Events), see “Standard E-Mail Events” below.

- If you press **F6** (Text), see “Standard E-Mail Text” on page 111.
 - If you press **F7** (Recipients), see “Standard E-Mail Recipients” on page 112.
3. When you have finished, press **F2** (Save) to save your e-mail preferences.

Standard E-Mail Events

You can review, define, and modify the standard events already defined in the PRC system. Each event triggers a particular type of e-mail to particular recipients.

To edit or create standard events that trigger e-mail, perform the following:

1. From the PRC Notification screen (Figure 4-11), press **F5** (Events).
The system displays the PRC Notification sub-screen, similar to the one shown in Figure 4-12:

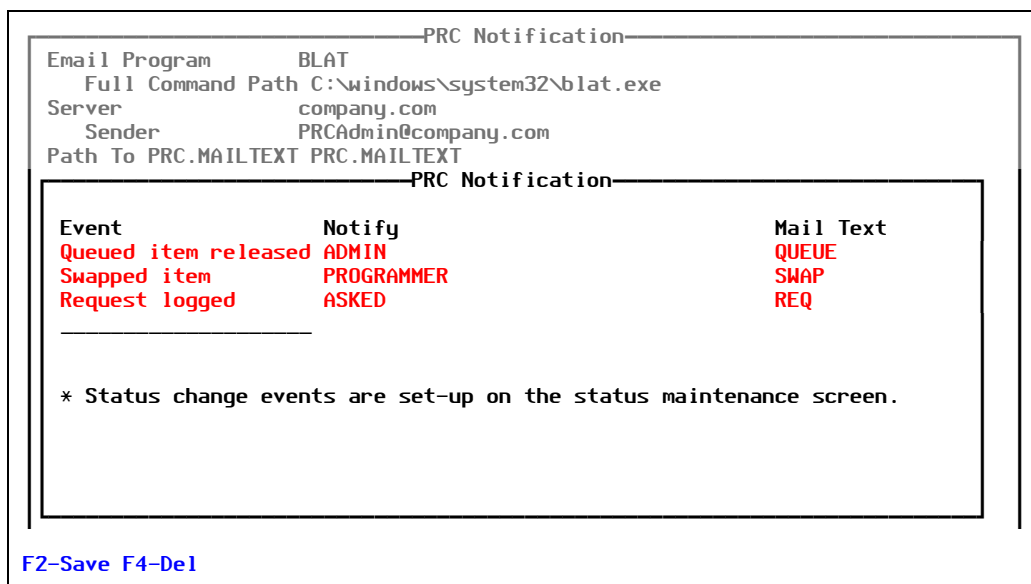


Figure 4-12: PRC Notification sub-screen

The fields in each line represent a standard mail event:

Event The event upon which an item’s status has changed. You define these events in “Status Codes” on page 29. Valid entries include:

- **QUEUE** — Queued items released
When developers have clearance level 2 . 4, the system offers them an option to “queue” for an item in conflict that they want to edit. When the item is released by the conflicting project, the system can send an e-mail to the person in the queue (and optionally others) to notify them that the item has been added to their project.
For more information, see the *Conflict Queuing* tip sheet.

- **REQ** — Request logged
When someone opens a new request in PRC's Request and Problem Reporting System, individuals can be notified via e-mail.
Note: This event covers a request, which is a separate entity from a project. If you wish to have notification that a *project* has been opened, use **STAT** (see below).
- **STAT** — Status changed
If you wish to have notification that a project has been opened, simply use the status change event for the status that a new project would be (probably **1** or **3**). Likewise, use this to send off an e-mail whenever a project's status changes to any other state.
- **SWAP** — Swapped item
When projects have priority **00** and an item in conflict is edited, the system presents the developer user with the option of swapping with the project that has the item checked out. Rather than taking a branch copy, this priority **00** project takes the original item and forces the other project to use the branch copy. The programmer who just had their item yanked off of their project is notified of this event.

Notify

The title, group, role, or individual that should receive the e-mail notification. Multiple recipients can be separated by the character defined in the *Delimiter* field (see [page 109](#)).

Mail Text

The standard text to accompany this e-mail notification. You define the standard body of the mail text (see the next section).

2. Fill in or edit each line according to the needs of your organization.
3. When you have finished, press **F2** (Save) to save your e-mail preferences.

Standard E-Mail Text

The text you edit or create in this sub-screen is contained in the **PRC.MAILTEXT** file.

To edit or create standard text for e-mail, perform the following:

1. From the PRC Notification screen ([Figure 4-11](#)), press **F6** (Text).
The system displays the Mail Text sub-screen, similar to the one shown in [Figure 4-13](#):

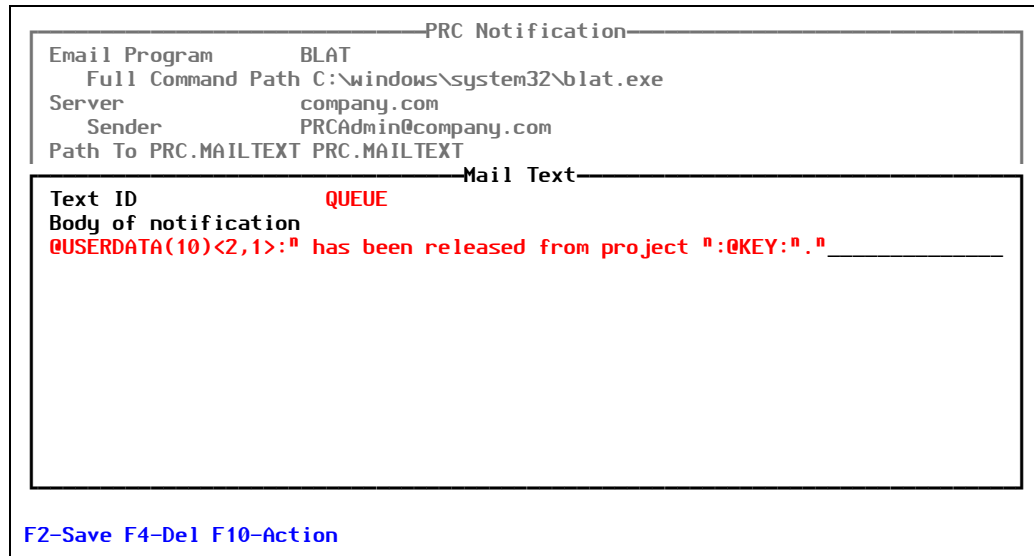


Figure 4-13: Mail Text sub-screen

The standard mail text items you define here will be contained in the `PRC.MAILTEXT` file. You can modify any text items, but four exist that you must not rename. The four are:

- **QUEUE**
- **REQ**
- **STATUS**
- **SWAP**

2. Fill in each field with the appropriate information.

You can modify the text and use any functioning SB+ expression or common variable.

For example, in [Figure 4-13](#), you are editing the text for the `QUEUE` notification. The item name is in common variable `USERDATA(10)<2,1>` and the project ID from which the item was released is the `@KEY` variable at the moment this notification is called.

If you wish to include information on an e-mail and are not sure where that information might be available in memory at that event moment, call support for assistance.

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

Standard E-Mail Recipients

The individual(s) who will receive the e-mail notification can be identified in a number of ways. Some are hardwired into the system to have a specific meaning, while others can be created for the purpose of notification. To review the standardized options and add your own, perform the following:

1. From the PRC Notification screen ([Figure 4-11](#)), press **F7** (Recipients).

The system displays the Recipients sub-screen, similar to the one shown in [Figure 4-12](#):

PRC Notification			
Email Program		BLAT	
Full Command Path C:\windows\system32\blat.exe			
Recipients			
Job Title	Email	Dist Groups	Email
PRC Administrat	~SJOSLYN	Developers	
Librarian	~JCATLIN	Managers	
Code		Sends e.mail to	
~USER.ID	THIS specific user (email address on PRC profile.)		
ROLE	The user associated to the status associated to this role.		
Keyword (put this actual word)			
"CLASS"	The person responsible for the class of project.		
"ASKED"	The original requester on the request/P.R.		
"CUSTOMER"	The user in the client/customer field on project		

F2-Save F4-Del

Figure 4-14: Recipients sub-screen

The fields in each line represent a standard mail event:

Job Title

The job title to which you want to send automatic notifications; for example, PRC administrator, librarian, or testing manager.

The type of recipients by their job. Explanations of the various entries are shown in the bottom portion of the screen.

A job title can be created on this mail preferences screen and attached to a user ID.

To send the notification to a particular individual, type a tilde (~) and their user ID.

To send the notification to everyone assigned a particular role, enter the code for that role in the *Job Title* field.

Sending notifications to a role is the most powerful option—but a bit more complex to set up:

- First, create a role by selecting **Files > Code Files > Role Codes**. The system displays the Role Maintenance screen. A role is associated to particular states. For example, the role of **QC** may be associated with state **5** (Testing). The roles and their associations will, of course, depend on your development life-cycle and settings.
- Once the role is set up and associated to a state, you can enter that role as a recipient for a certain notification. What will happen is the person associated to that state on the particular project will be notified.

So, for example, when a programmer bumps their project to status **4** (Dev complete), you want the tester to be notified. The tester is assigned the **QC** role, which is associated with status **5**. You enter **QC** in the *Job Title* field. Now when a project hits status **4**, an e-mail goes out to whoever is assigned and/or targeted to be the tester on that project. If no one is slated on a project at the particular state for the role, the target sub-screen of the Project Master will pop up so you can make those assignments.

Press **[F3]** for a list of valid job titles.

Email

The title, group, role, or individual that should receive the e-mail notification. Multiple recipients can be separated by the character defined in the *Delimiter* field (see [page 109](#)). Valid entries include:

- Role
A role to which one or more employees may be assigned. Role codes are defined in “[Roles Codes](#)” on [page 36](#).
- Specific user
When someone opens a new request in PRC’s Request and Problem Reporting System, individuals can be notified via e-mail.
The type of project that will direct status change e-mail notifications, if project types are in use. If not, type the keyword **ALL** in this field.

Distribution Groups

The groups that will receive e-mail notifications; also known as a distribution list.

A distribution group can be named and associated to an e-mail group address.

Mail text options – Specific text may be entered into this field, or the NAME of an ITEM in a MAILTEXT file can be entered here, and the text in that item will be used. This feature has been retired because it is complex and no one used it but it can be un-retired if anyone is interested in using it.

If no specific text is specified here the project ID will be in the regarding line and the body of the message will display the project ID again, the brief description from the project and the statement that “The referenced project has moved from state x to state y.”

Press **[F3]** for a list of valid distribution groups.

Email

The title, group, role, or individual that should receive the e-mail notification. Multiple recipients can be separated by the character defined in the *Delimiter* field (see [page 109](#)).

The type of project that will direct status change e-mail notifications, if project types are in use. If not, type the keyword **ALL** in this field.

2. Fill in or edit each line according to the recipients you want for each notification.
3. When you have finished, press **F2** (Save) to save your settings for recipients.

Manually Send an E-Mail

To send an e-mail manually, perform the following steps:

1. Select **Action > Mail/Notification Menu > Manual e-mail send**.

The system displays the Mail things screen, similar to the one shown in [Figure 4-15](#):

Mail things

Mail Body _____
Subject

Recip List
Cc List
Bcc List

Sender

Attachments

F2-Save F4-Del

Figure 4-15: Mail things screen

Explanations of the fields are as follows:

Mail Body

The name of an item in the **PRC.MAILQUEUE** that will form the body of the e-mail message. All bodies have to be pre-defined, which is explained in [“Mail Preferences” on page 108](#).

PRC includes a handful of standard mail bodies pre-defined.

Subject

The text for the Subject line.

Recip List

The addresses of the recipients, separated by commas.

Cc List	The addresses of the recipients who are on the “carbon copy” list (being copied about this e-mail, to keep them informed), separated by commas.
Bcc List	The addresses of the recipients who are on the “blind carbon copy” list (being secretly copied about this e-mail), separated by commas.
Sender	The address of the sender. You can leave this blank, if desired, to use the default sender, which is set up in the mail preferences
Attachments	The full path of any attachments, including its file name and extension.

2. Fill in each field with the appropriate information.
3. Press **[F2]** (Save).

Once you save this screen, the e-mail is distributed. Works with many e-mail systems, primarily SENDMAIL with Unix and BLAT with Windows.

Ravel File Information

This menu only serves as online help to describe the different ways to accept delivery of a ravel file. Each selection displays a screen full of explanation and then launches a particular program so you can look at the ravel screen. You will never actually run any ravel options in the PRC account because they all require a specific setup on the receiving machine.

You discuss your company’s plans for ravelling during the PRC installation meetings.

Menu options include:

- **About Ravel Files**, which is extra help text about the other three menu options
- **Simple Extract**
- **Simple Extract (SB Reqd)**
- **PRC Customer Side**

Utilities

Useful Actions for Managing PRC

When you select **Action > Utilities**, the system displays the Action Utilities submenu, as shown in [Figure 5-1](#):

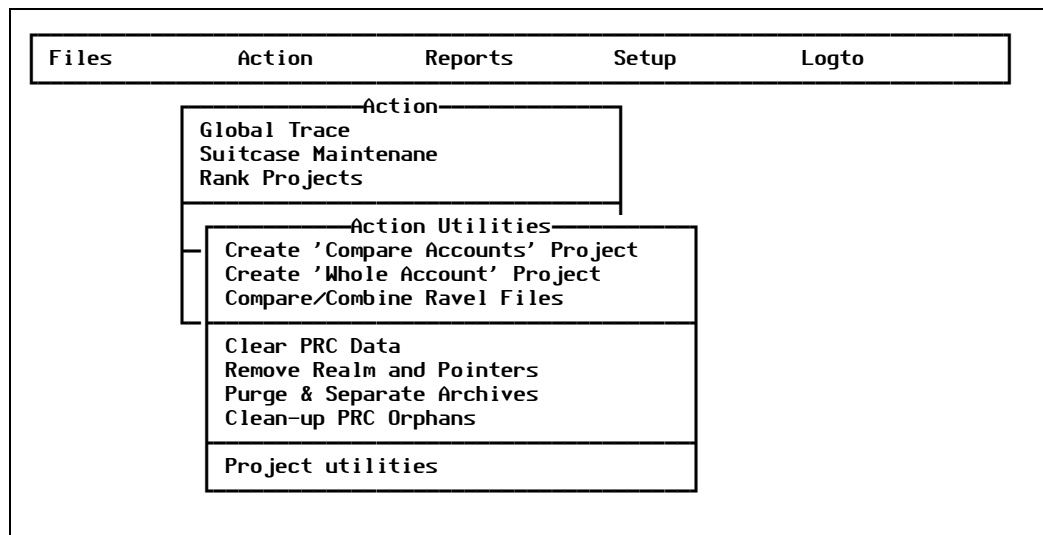


Figure 5-1: Action Utilities menu

For explanations of the other **Action** menu options, see [Chapter 4, "Action Menu."](#)

These utilities are special purpose functions.

Create "Compare Accounts" Project

This option allows you to run a utility that is rarely used.

To compare two projects, perform the following steps:

1. Select **Action > Utilities > Create "Compare Accounts" Project**.

The system displays the Create "Compare Accounts" Project screen, similar to the one shown in [Figure 5-2](#):

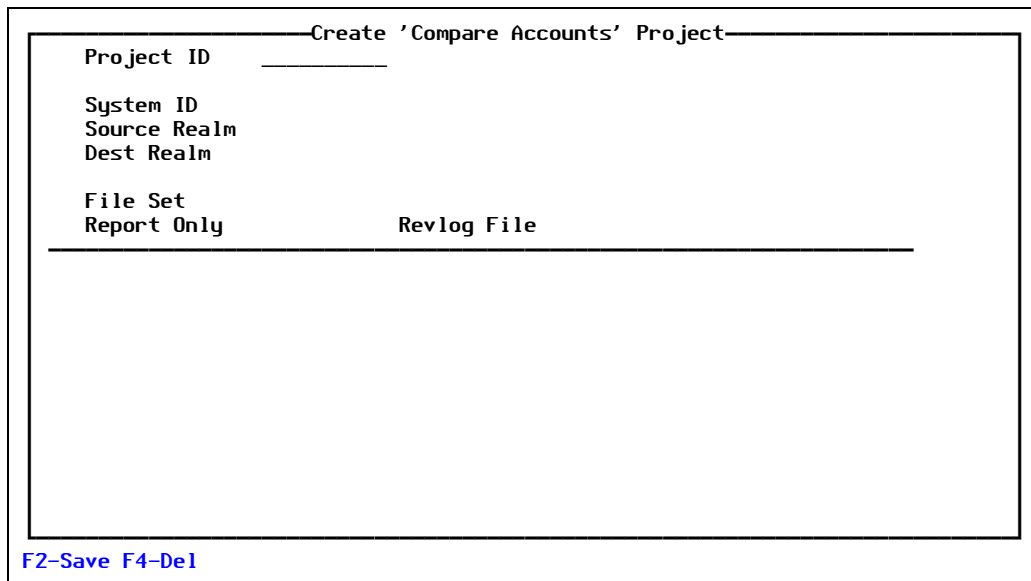


Figure 5-2: Create “Compare Accounts” Project screen

Explanations of the fields are as follows:

Project ID The code or ID that represents the project to be “built up” by the results of this comparison. The project must already be created (with proper class and realm) before using this option.

Source Realm *[Informational only]* The source realm on a “destination system” may or may not be the same as it was when the revision was created on the source machine. The PRC system gets a lot of its instruction from the Realm Master files. There are behavior instructions both when a release is created on the source machine and when it is unravelled on the target machine. These instructions may or may not be different.

Certainly, the paths are different: the target realm path will be REMOTE on the source machine, but will need to be the true location on the destination machine. Whether a compare is done, whether conflicts are analyzed against active projects only or against all history, and whether conflicting items may move or not are checked during both sides of the release function. Therefore, what you enter here should be the name of the REALM which has the behavior and structure that you want on this destination system. In all likelihood the name will be the same as the development REALM is named on the development machine, but that is not what is important.

The source realm on the destination system is like a staging area: the software does not need to exist there, but the tracking revision logs will. The behavior is set up in the definition.

Dest Realm	The name of the realm into which this software should be delivered. This is where the REALM ID of where the software should actually be installed is entered. Remember that the REALM definitions on the setup menu must be set up properly for this destination machine so that all will work right. Press [F3] for a list of valid realms.
File Set	The name of a file set to use for this function. The default is the OWN set for the system, if present, or the XXCONTROL FILES item, if OWN NOF.
Report Only	Indicates whether to only create a report about the results of this comparison, rather than actually create a project from the results of this comparison. Valid entries are: <ul style="list-style-type: none"> • Y — Yes, display/print a statistics report only • N — No, create an actual project containing the “different” items in the source realm Press [F3] for a list of valid entries.
Revlog File	If you do not want log items into the default revision log file REALM.SYSTEM.REVLOG, enter the name of the file into which you want the system to log items; otherwise, leave this field blank.

2. Fill in each field with the appropriate information.
3. Press **[F2]** (Save).
The system builds a project.

Create “Whole Account” Project

If you want to create a project from a file set outside of project tracking, you can use this menu option. This option will also create a ravel file, if you desire. Bundling a project creates a history of the items as if they *were* tracked.

Note

Do not use this option except in rare circumstances. This feature was originally created for a unique situation. It is neither a supported nor recommended function.

To bundle a project, perform the following steps:

1. Select **Action > Utilities > Create "Whole Account" Project**.

The system displays the Create "Whole Account" Project screen, similar to the one shown in Figure 5-10:

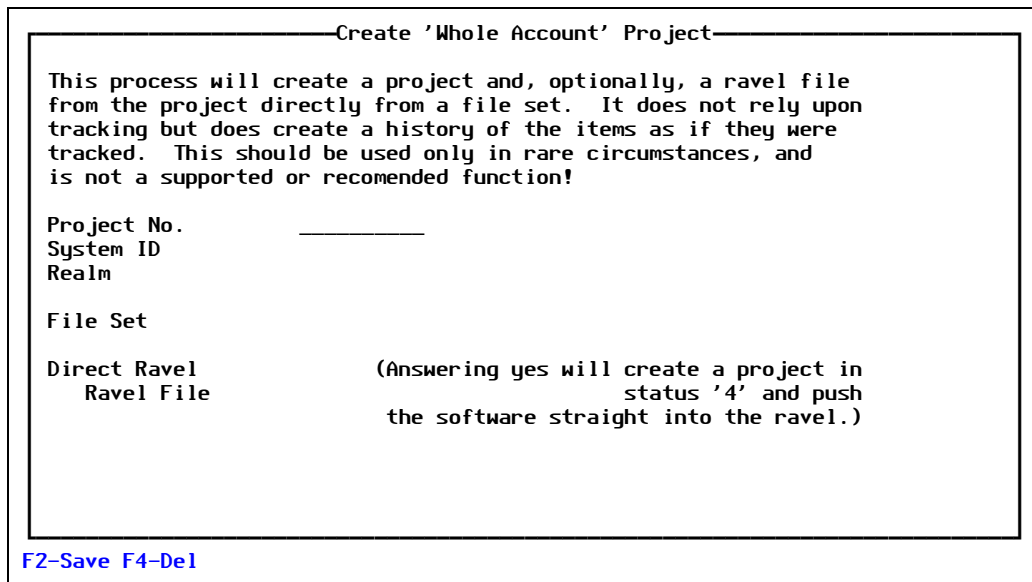


Figure 5-3: Create "Whole Account" Project screen

Explanations of the fields are as follows:

- | | |
|---------------------|--|
| Project No. | The code or ID that represents a project to be built from all of the current contents of a particular system ID. |
| System ID | The code or ID that represents the SB+ system from which this project will be bundled.

Press [F3] for a list of valid system IDs. |
| Realm | The code or ID that represents the realm from which the project will be bundled. |
| File Set | The name of a file set to use for this function. The default is the OWN set for the system, if present, or the XXCONTROL FILES item, if OWN NOF. |
| Direct Ravel | Indicates whether a ravel file should be made directly from the bundled project. Valid entries are Y (Yes, create a ravel file) and N (No, just create the project).

Press [F3] for a list of valid entries. |
| Ravel File | The name of the ravel file that contains items you would like to merge, if the value of the <i>Direct Ravel</i> field is Y . |

2. Fill in each field with the appropriate information.
3. Press **F2** (Save).

Compare or Combine Ravel Files

This is a rarely-used utility.

To compare or combine ravel files, perform the following:

1. Select **Action > Utilities > Compare/Combine Ravel Files**.
The system displays the message shown in [Figure 5-4](#):

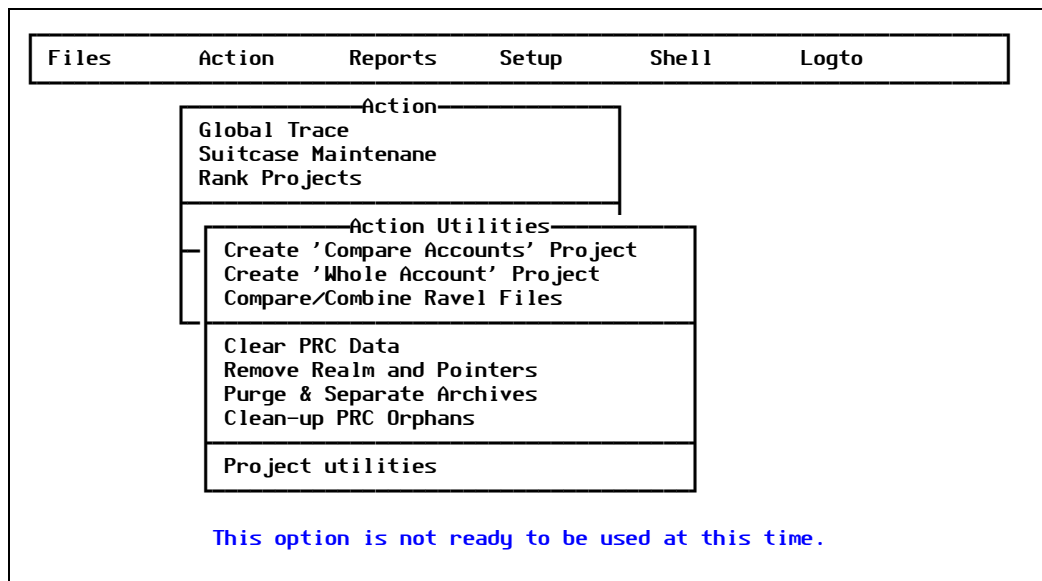


Figure 5-4: Compare/Combine Ravel Files message

Clear PRC Data

Use this option to clear all transaction history for the specified realms and systems. The system prompts you for the master and code files first.

WARNING

Use this option with extreme caution. You would most likely use this feature only after an initial trial/training period. If you like your settings, be careful what you clear.

To clear partial files, either some or all, perform the following:

1. Select **Action > Utilities > Clear PRC Data**.

The system displays the Cleanup PRC Data screen, similar to the one shown in [Figure 5-5](#):

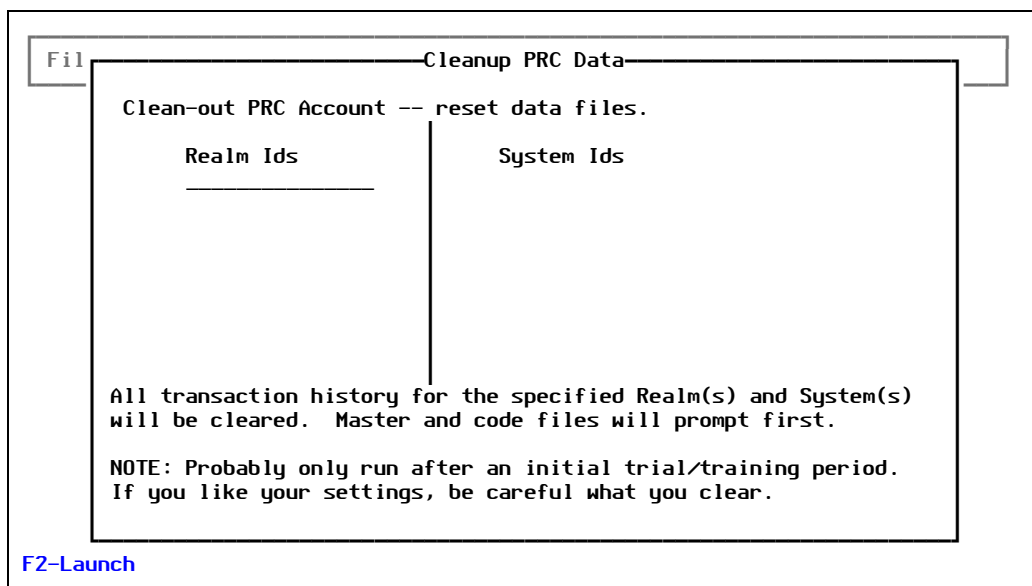


Figure 5-5: Cleanup PRC Data screen

Explanations of the fields are as follows:

- Realm IDs** The code or ID that represents one or more realms you would like to remove.

Press **F3** for a list of valid realms.
- System IDs** The ID that represents one or more SB+ systems you would like to remove for the realms listed above.
- Press F2** Frequently, fields such as this are placed on the screen to remind the user what to do next. It also makes a screen with only one prompt clearer to understand.

2. Fill in each field with the appropriate information.
3. Press **F2** (Save).

Remove Realm and Pointers

Use this feature to delete all associated data files within a realm, essentially erasing the realm itself from the system.

An alternative is retiring the realm, which you can do if you wish to preserve the history in the transaction files. To retire a realm, simply mark it as “inactive.” The decision whether to erase the history from that realm is a separate issue.

To remove or erase a realm from the system, perform the following steps:

1. Select **Action > Utilities > Remove Realm and Pointers**.

The system displays the Remove Realm Files & Pointers screen, similar to the one shown in [Figure 5-6](#):

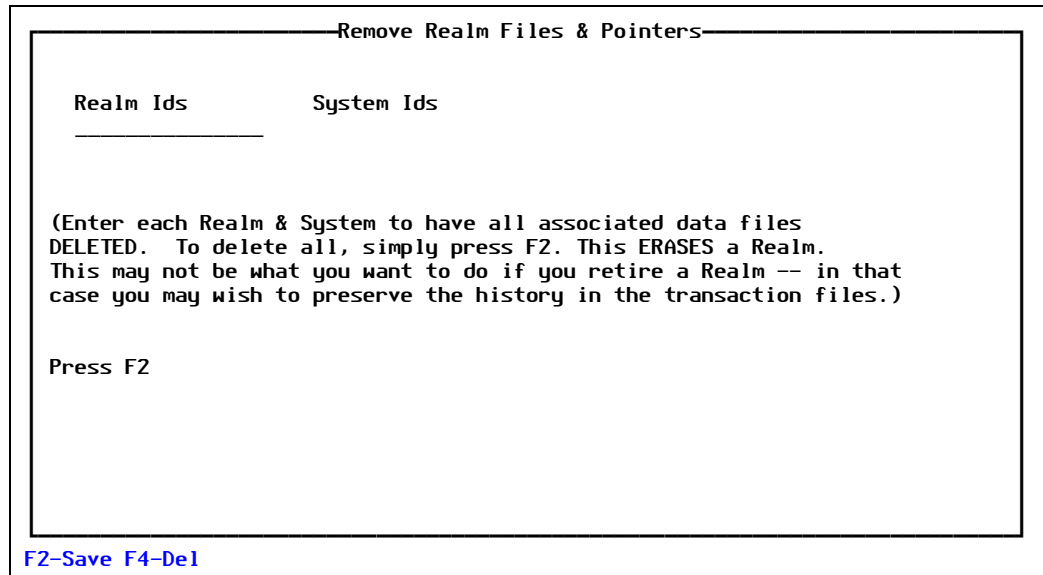


Figure 5-6: Remove Realm Files & Pointers screen

Explanations of the fields are as follows:

Realm IDs	The code or ID of one or more realms you would like to erase. Press F3 for a list of valid realms.
System IDs	The ID that represents one or more SB+ systems you would like to erase for the realms listed above.
Press F2	Frequently, fields such as this are placed on the screen to remind the user what to do next. It also makes a screen with only one prompt clearer to understand.

2. Fill in the *Realm Id* and *System Id* fields with the appropriate information. If you are erasing more than one realm, enter as many lines as necessary.
3. Press **F2** (Save).
PRC erases the realm or realms you specified.

Purge & Separate Archive Files

Note

PRC separates items archived before the date you specify in this option into another file. You can store those files offline or simply delete them from the system.

Each realm and system has its own data section. The files that make up a data section include:

- RELEASE
- JRNL
- ARCHIVE.BU

Data sections are archived to history files with a similar name.

For example, assume the realm is DEV and the SYSID is AP. A data section named XXX.RELEASE, DEV.AP is archived to HST.RELEASEDEV.AP.

To purge archives, perform the following:

1. Select **Action > Utilities > Purge & Separate Archives**.

The system displays the Archive Rollout Data screen, similar to the one shown in [Figure 5-7](#):

Archive rollout data
Purge Archives to separate file

Archive Date _____ Arc Modulo
(Date displayed is last archive date, if applicable).

Items archived before the entered date will be separated into another file. These files can be stored off-line or simply deleted from the system.

Each realm and system has its own data section. Files are RELEASE, JRNL and ARCHIVE.BU. Thus data sections XXX.RELEASE,DEV.AP, for example, is archived to HST.RELEASE,DEV.AP. (DEV is the Realm and AP is the SYSID.)

Press F2 to begin.

F2-Save F4-Del

Figure 5-7: Archive Rollout Data screen

2. Enter a date in the *Archive Date* field. The system moves all rollout archives prior to that date to the history file for archive and deletion from the system.
Press **[F3]** to select a date from the pop-up calendar.
3. Enter the preferred modulo for the archived data files in the *Arc Modulo* field. The system creates all archive files using this modulo.
For more information about modulos and archives, please refer to the database administration guide for your database.
4. Press **[F2]** (Save) to begin the archiving process.

Clean Up PRC Orphans

This is a special-purpose utility. Essentially, this is a way to gather any items that are checked out—but no longer checked out to a valid project. The only known way for this to occur is if someone deletes a project from TCL or some other unusual circumstance occurs.

To clean up “orphans,” select **Action > Utilities > Clean Up PRC Orphans**.

The system displays the PRC REVLOG Integrity message, as shown in [Figure 5-8](#):

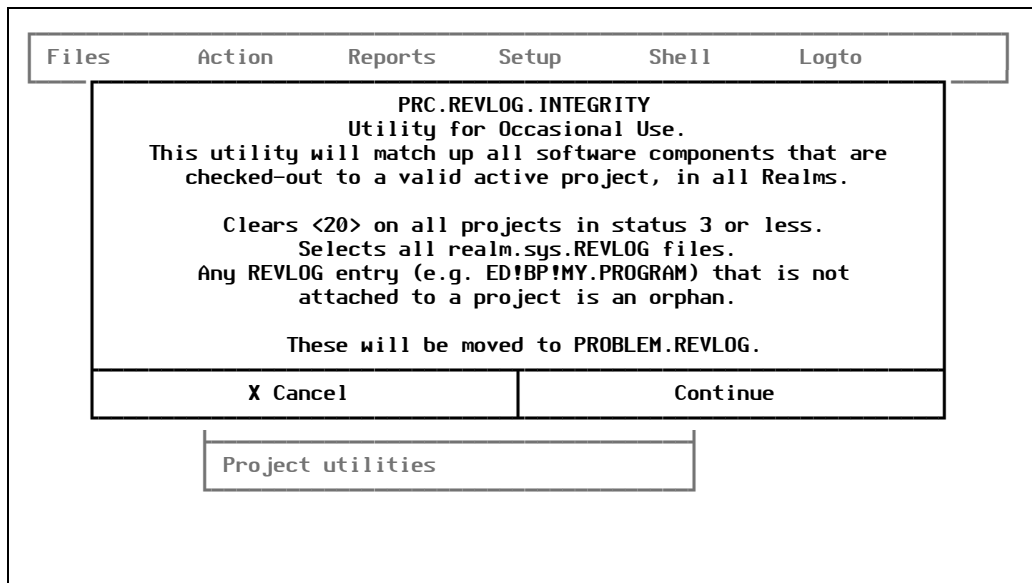


Figure 5-8: PRC REVLOG Integrity message

Project Utilities

When you select **Action > Utilities > Utilities Menu > Project Utilities**, the system displays the following:

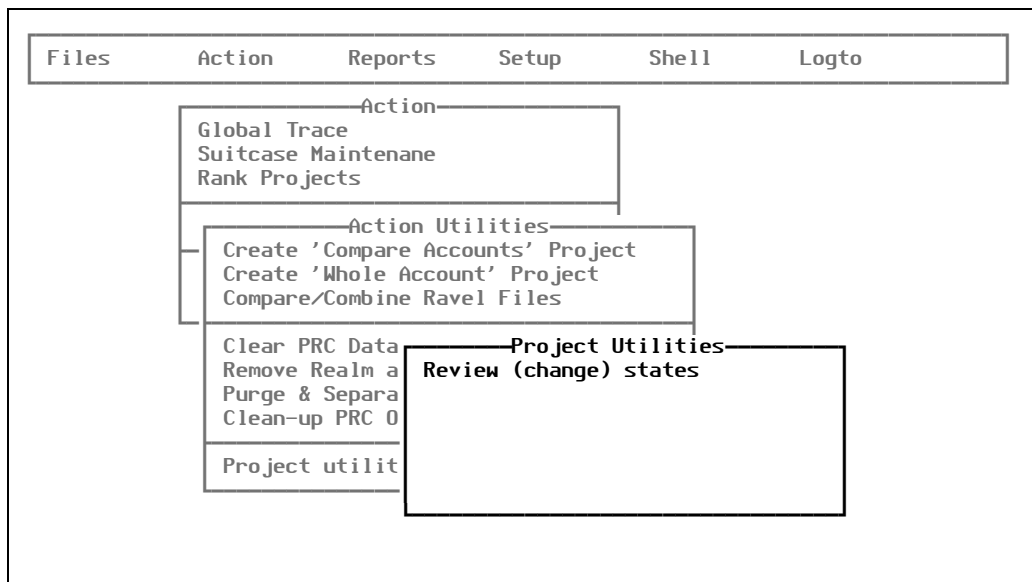


Figure 5-9: Project Utilities sub-menu

This section explains this menu option.

Review or Change States

To compare two projects, perform the following steps:

1. Select **Action > Utilities > Utilities Menu > Project Utilities > Review (change) states**.
The system displays the following warning in [Figure 5-10](#):

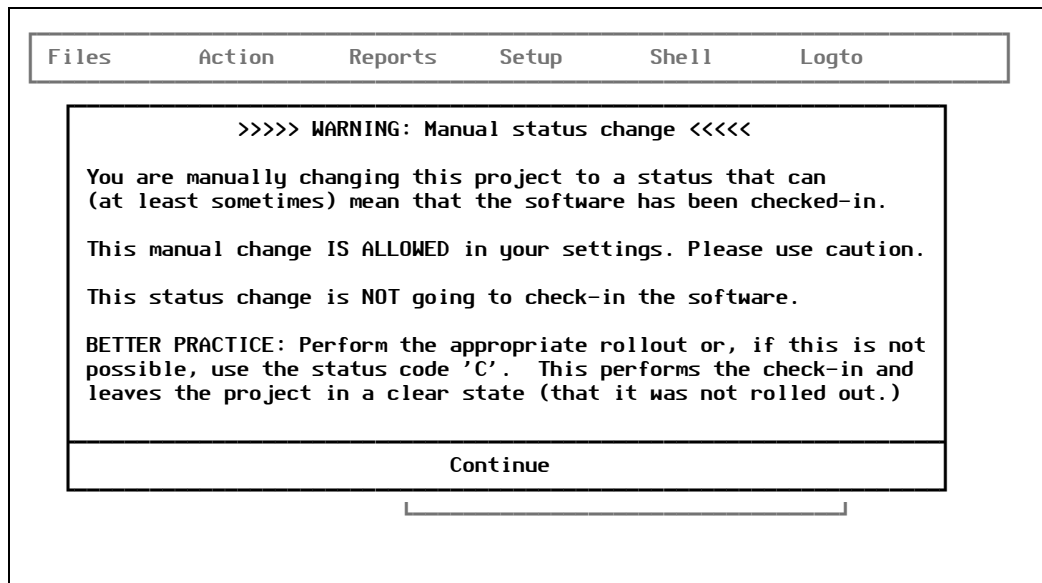


Figure 5-10: Warning screen

2. Once you have read the warning, select *Continue*.
The system displays the Review (Change) Project States screen, similar to the one shown in [Figure 5-11](#):

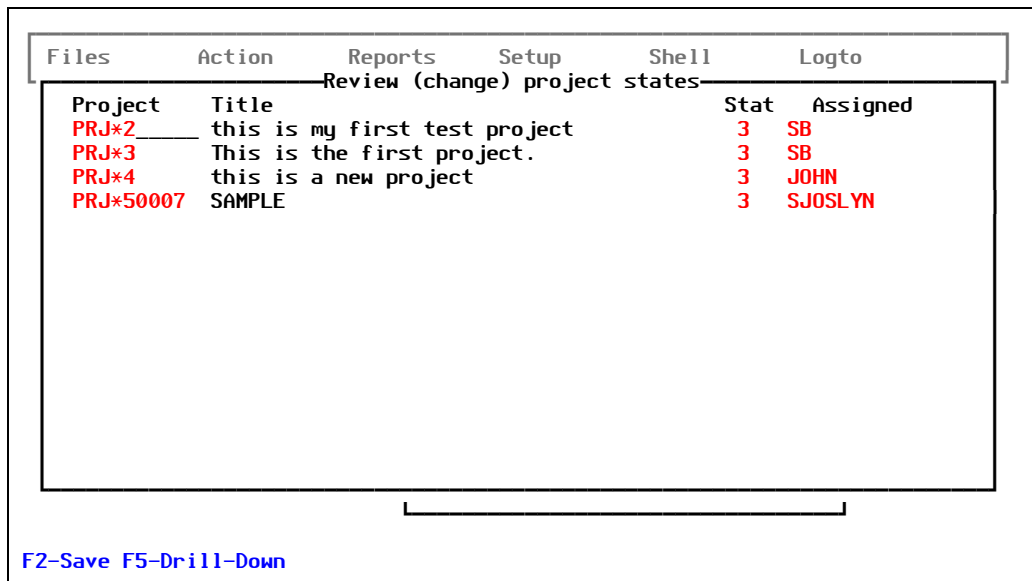


Figure 5-11: Review (Change) Project States screen

Explanations of the fields are as follows:

- Project** The code or ID that represents the projects available to manually change.

- Title** [*Informational only*] The description of this project.

- Stat** The current state of the associated project.

- Assigned** The user to which the associated project is currently assigned.

- If you press **F5** (Drill-Down) while the cursor is in any *Project* field, the system displays the following:

Review (change) project states			
PROJECT	3 CH	3 Active & Assigned	Realm DEV
This is the first project.			Development
Files	Fld Scr Rpt	Processes 0	Programs 0
BP.UTIL 0	Defns 0		PRC. INCLUDES 0
Text	Help Docs	Menus	Misc EDits
Messages?	'FILES' item?	Accumulator?	Equate Exp

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

Figure 5-12: Review (Change) Project States screen

If you decide to return to the previous screen, press **F2** (Finish).

3. To change the status code or assignment of a particular project, press **Enter** while the cursor is on the project you choose to change to advance to the *Stat* and *Assigned* fields.
4. Change the value of the *Stat* and *Assigned* fields as desired.
5. Once you are done, press **F2** (Save).

PRC Configuration

Realm, Account, System, Platform, Suite, Station

Certain master files (Project, Request, Version, Customer, and Department) are explained in [Chapter 1, “Maintenance Files”](#) and pertain to information and tasks that change or grow on a daily basis.

The master files explained in this chapter pertain to overall, system-wide information upon which the daily information is based.

For explanations of the other options in the **Setup** menu, see [Chapter 7, “Setup Menu.”](#)

Realm Master

A realm in PRC is a defined region (directory, account, or accounts) that is governed by a particular set of rules. Those rules define the users that can make changes, the conditions under which those changes can be made, and where software can be delivered or received. A realm can sometimes be one or more than one Pick or U2 account.

To work with the Realm Master, perform the following:

1. Select **Setup > Configuration > Realm Master**.

The system displays the Realm Master screen, similar to the one shown in [Figure 6-1](#):

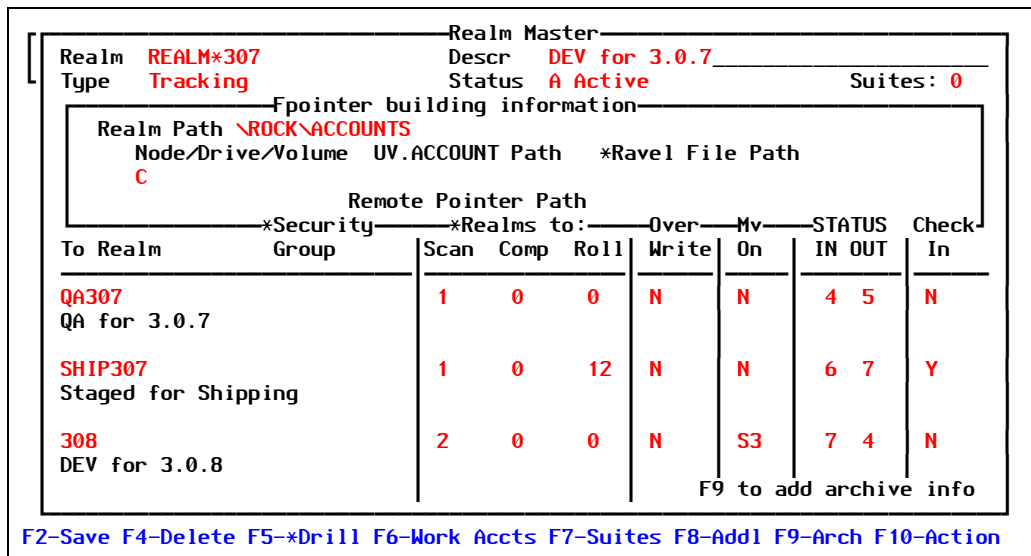


Figure 6-1: Realm Master screen

Explanations of the fields are as follows:

Realm

The code or ID that represents the entity, area, or “world” defined by this path. A realm, as the name implies, is like a world that is a geographical location on disk; also, depending on the name of the realm, it can refer to the purpose of the area.

Typically, at least one development and one production realm exist. Each may include a number of different accounts or systems, and each should be an exact replica of the other.

Defining a realm on UNIX-based systems can be simpler than in other implementations. Because a different directory can be set up for the realm, all of the account names and systems names can be identical.

On non-UNIX or non-directory-underneath implementations, the realms will likely have to be identified as part of the ID of the account or system. Therefore, there would be an **AP** account (for the LIVE realm) and **D.AP** (for the DEV (development) realm). In these configurations, there is no true geographical or physical separation between realms.

As part of the definition of a realm, we include its identification code (one letter should be enough), a path name for those implementations where it is applicable, and some rules of behavior for the realms.

For example, one realm behavior would dictate the realms in which software could be moved from one realm to another. It could also define in what status must a project be before it can be moved into this realm.

Press **F3** to select from the list of valid entries.

Type

Enter one of the specialized Realm Types. Valid entries are:

- **T** — Tracking
Tracking realms are where changes are allowed and tracked. Tracking software must be installed on each system within the tracking realm, after designating the realm type here.
- **TP** — Tracking (Primary)
- **TS** — Tracking / Support
- **TI** — Tracking Import
- **TSTD** — Tracking Standard
- **R** — Remote
Remote realms are located at a client site or another computer. A rollout must create a ravel file for transport. The second half of the rollout is performed on the destination system (the “unravel” process).
- **O** — Online
Online realms are either on the same computer or are at least accessible across a network so that software items are moved directly from one realm to another.
- **I** — Tracking Tiers
- **RT** — Remote Tiers
- **TN** — Tracking (No Unraveling)
- **C** — Custom
Customization realms are actually archive areas. Software is not really installed there, but is ravelled up and archived along with the tracking items. A realm can be reconfigured by rolling all projects in a certain realm back.

Press **F3** for a list of valid types.

Status

Indicates whether projects can be accessed. Valid entries are:

- **A** — Active
Normal
- **F** — Frozen
No projects can be opened nor changes saved, even by users with the appropriate clearances.

Projects move through statuses as they progress. They begin with being in a review mode, then are activated to work, test and live with cancelling and placing on hold as options along the way. This field allows restricting what status of software may be moved *in* to this realm; for example, the LIVE realm should be restricted to only status **5** (Test Complete). The development realm may allow all statuses to move in, or all except cancelled or held. Whatever statuses are entered here will be allowed to move *in*.

Press **F3** for a list of valid statuses.

Realm Path

The full path to the realm, if it is necessary to build F-pointers. Use a preceding backslash character (\) to enter absolute UNIX paths, which must begin with a slash character (/).

The full path name is necessary for DOS and UNIX environments to build file pointers for moving and validating software. Users on an environment that support Q-pointers and who selected Q as their need not enter a path.

If the realm exists on another computer system and releases are to be ravelled into a file for transport, the value of the *Type* field is **R** for remote and no path is required because the path would not be on this computer. In this case, the realm is defined on the target computer (if it is PRC-enabled) and will have the proper path there, if applicable.

Node/Drive/Volume

The node name (or disk name) in a networked or multi-disk (prime) environment.

UV.ACCOUNT Path

[*UniVerse only*] The literal (not *relative*) path for the UV.ACCOUNT file for this realm.

Important: Include the node name, if it is relevant.

Ravel File Path

The path to a separate directory where ravel files should be created and kept. It is the path PRC will be used to create (UD) or move (UV) the file.

Remote Pointer Path

The path that goes in the F-pointer in the VOC, if it is different from the one specified in the *Ravel File Path* field, used to create ravel files on another (networked) machine.

To Realm

The other realm or realms into which software in *this* realm may be moved.

For example, you give the TEST realm permission to move all items to LIVE or other parallel realms, because software coming out of TEST is supposed to be tested and ready for live or production realms.

The **R** realm, which contains imported revisions (work that was done off-site), should most likely be restricted to moving out to only the **D** (development) realm.

Press **F3** for a list of valid realms.

Security Group

The groups and/or users that are allowed to perform this rollout. It may also be the groups or users that are specifically restricted from performing this rollout.

The security permissions and restrictions can be a group, an individual user, or a “role.”

To exclude a group, user ID, or role, use the hash/pound/NOT sign (#). For example, **#PROGRAMMER** will preclude the individual(s) in the role of PROGRAMMER on a particular project from rolling out that project.

Any place where PRC wants you to enter an individual or group, it runs the same algorithm. If you press **F1** help on them, the system provides a complete explanation.

To add users and groups, place the cursor in this field and press **F5** (Drill). The system displays the Groups/Users Authorized subscreen. Press **F3** for a list of users and groups you can add.

Realms to: Scan

Displays the number of “other” realms that are scanned for conflicts when you roll out this realm to the destination realm in the *To Realm* column. Press **F5** (Drill) to work with other realms that are scanned during conflict analysis.

When you press **F5** with the cursor in this field, the system displays the Scanned Realms subscreen. It lists the realms in the *Realm* column and includes an *Active/Hist* column, which indicates whether each realm is one of the following:

Realms to: Scan (cont)

- **A** — Active
The source realm is listed first by default. You can change it or add other realms. Active scans search for these realms for conflicts with the source realm and its active projects.
- **H** — Historical
Historical scans search open *and* closed projects from a beginning date, prompted at the rollout.

Why would you want to have PRC scan against closed projects? One example is if baseline development is ongoing in a DEV realm, but bug-fixes are being made in a SUPPORT or “field release” realm. After the bug fix is complete in SUPPORT, it is a good idea to have it scan against what is going on in baseline development before delivering it to a customer.

Press **F5** (Drill) to add other realms to scan during conflict analysis.

Realms to: Comp

Displays the “other” realms that are compared to this realm. If you list realms in this column, projects will be compared to these other realms to look for conflicts.

If you want to compare realms, you must enter information in this field. Otherwise, if it blank, the default value is “do not compare to a project in any other realm.”

Note: Comparing realms does not involve a source or destination. It is a way to tell PRC to compare against *yet another* realm that’s neither the source nor the destination.

Why would you want to compare two realms? One common scenario is the case of a DEV realm and a TEST realm. When development reached a certain point, the DEV realm was rolled into the TEST realm. Several bugs were found in the TEST realm; but because each one was a small fix, the changes were actually made in the TEST realm. So when it is finally time to roll out the DEV realm to the LIVE realm, you would want to compare DEV to the TEST realm to expose any minor fixes that were not incorporated into the DEV realm.

When the cursor in this field and you press **F5**, the system displays the Compare to Realms subscreen. The subscreen lists the realms in the *Realm* column to which the source realm will be compared when you roll this project out to the destination realm (which is listed in the *To Realm* field).

Realms to: Comp (cont)

A comparison is a non-fatal notification during the rollout. It is used generally as a “just in case” option. Items that fail the comparison are displayed during the rollout and stored in the comparison log. You can view the comparison log from the Project Master screen by pressing **F8** (Source), then pressing **F10** (Action) for the Action bar.

Press **F5** (Drill) to add other realms to scan during conflict analysis.

Realms to: Roll

Displays the number of other realms that are designated as “parallel rollouts.” These other realms would be updated automatically during this primary rollout.

When you press **F5** with the cursor in this field, the system displays the Parallel Realms subscreen. It lists the realms in the *Realm* column to which this realm will be rolled out, secondary to the destination realm in the *To Realm* field.

Press **F5** (Drill) to add other realms to scan during conflict analysis.

Override

Indicates whether overriding an option is allowed when software is moved from this primary to this associated realm. Valid entries are:

- **Y** — Yes, the conflict scan will warn about conflicts, then proceed by overriding
In other words, after PRC decides that you should *not* perform this rollout, the system will allow you to override its recommendation and roll out anyway.
- **N** — No, overriding is not acceptable
- **P** — Password required to override

Note: Under normal circumstances, you do *not* want a project to roll out if conflicts exist amongst its source items.

If rollouts are a problem (you are always getting conflicts messages and trying to roll out a project appears to be “too much work”), do not set this field to **Y**. Re-consider some of the your development procedures. Some suggestions are listed in the SCM Plan Worksheet.

Mv On

Indicates whether the software can “move on” from this destination realm—in other words, the target realm becomes the source realm and the next rollout goes from there.

The term “move on” refers to moving the source (origin) of a project to the realm to which it is currently being rolled. Valid entries are:

- **N** — No, software will always rollout from the original realm (probably DEV)
For example, roll it from DEV to TEST, then later when you roll it again it will roll the software from DEV to LIVE.
- **Y** — Yes, software will roll to this destination realm and the project itself will change to that realm so that the *next* rollout will actually pick the software up from this destination realm
For example, roll it from DEV to TEST, then later when you roll it again it will roll the software from TEST to LIVE.
- **Sn** --- Spawn, where *n* is the status of the newly-spawned sub-project (typically **3** because you would want to open the sub-project)

For further explanation, see [“Spawning” on page 139](#).

Typically, software is developed in a DEV realm, moved to a TEST realm, then—when the time comes for delivery to a PROD (production) realm—the software is still picked up from the original development area. In a “move on” rollout, the source realm for the project is moved on so that the roll to PROD will actually pick up the software from the TEST realm.

Note: The examples above are exactly that—examples. Your realms may be named differently and you may have more stages between original development and production release. The “move on” capability is useful if software is to move to a parallel environment for testing and you want to release the conflicts in the development realm as soon as the project moves to test. You will probably need to do this if there is a bottleneck at that phase and having items remain checked out in DEV is cumbersome.

Recommendation: Use this feature in conjunction with the Compare feature, which allows you to manage the most common problem: software evolving in a TEST realm.

Status: In

The code or ID that represents the project’s state in this realm before it can move on to the realm indicated in the *To Realm* column.

The project must already be in this state before in can be rolled into the realm designated on this line.

Press **F3** for a list of valid status codes.

Status: Out

The code or ID that represents the project’s state once it is has “moved on” to the realm indicated in the *To Realm* column.

The project must changed to this state after it has rolled into the realm designated on this line. In other words, if the move from the source realm to the destination realm indicates a change in status, define the status to which it will change in this column.

If the movement does not indicate a change in status, leave this field blank—the status of the project will be unaffected by this rollout.

Press **F3** for a list of valid status codes.

Check In

Indicates whether ++ Release (let go/move to history) tracking on items when moved from this primary to this associated realm? Valid entries are:

- **N** — No, software will always rollout from the original realm (probably DEV)
- **Y** — Yes, software will roll to this destination realm and the project itself will change to that realm so that the *next* rollout will actually pick the software up from this destination realm

In other words, PRC lets go and moves it into history.

When tracking is “let go” in this manner, the project will no longer cause a conflict during work in the source realm. It will also not flag conflicts during roll-outs from the source realm when a regular “Active Scan” rollout is used. The project would still flag conflicts if a roll-out were done with an “historical” scan.

2. Fill in each field with the appropriate information.
 - If you press **F5** (*Drill), see “[Drill Down](#)” on page 140.
 - If you press **F6** (Work Accts), see “[Work Accounts](#)” on page 141.
 - If you press **F7** (Suites), see “[Suites](#)” on page 142.
 - If you press **F8** (Addl), see “[Additional Details](#)” on page 142.
 - If you press **F9** (Arch), see “[Archives](#)” on page 143.
3. Press **F2** (Save).

Spawning

Explanation

A sub-project is “spawned” when a conflict arises.

The ability to “move on” becomes complicated if rollbacks are common and you have many realms. It is possible to “move on while not moving on”—otherwise known as “spawning sub-projects.”

You allow spawning if you develop to more than one version and need to get bug fixes from one version to another.

The original project will remain in the original realm with its original project ID, and its status changed according to the *Status Out* setting during the rollout. Because a conflict was detected, PRC will create one or two spawned sub-projects:

- They will both be identical to the “parent” project being rolled out in terms of type, description, class, and so on.
- Each will have a suffix of a realm ID.
 - One spawned sub-project will have a suffix of the source realm and will contain all of the items that were able to roll into the destination realm without any conflict.
 - The other spawned sub-project with the destination realm suffix will contain the items that were in conflict on the destination realm. This is the “branch copy.” (For more information, see the *Project Tracking Guide*).

Example

Here is an example, broken out by bullet points to help you follow the logic:

- You have project PRJ*1000, which includes five programs.
- The project was developed and completed in DEV3, which is the development account for Version 3 of your software.
- It is a good bug fix and you want it in DEV4, the account in which the new release is being developed.

- In the Realm Master, you define **REALM*DEV3** with the following:
 - **DEV4** in the *To Realm* field
 - You drill down in the *Scan* field, list **DEV4**, and set it to **A** (Active)
 - **S3** in the *Mv On* field
 - **4** in the *Status In* field, and **5** in the *Status Out* field
- You roll PRJ*1000 out to DEV4.
- One of the five programs on PRJ*1000 is in conflict on DEV4.
- When the rollout is finished, you end up with three projects compared to starting with just one:
 - PRJ*1000, which is still anchored in DEV3 and now has a status of **5**. It still contains its original five items.
 - PRJ*1000.DEV3, which contains four programs—the ones *not* in conflict on DEV4. This project is anchored in DEV4 now, meaning that its “source realm” is DEV4.
 - PRJ*1000.DEV4, which contains one program—the program in conflict. It is a branch copy, which means it is an exact copy of the program exactly as it was in DEV3, and it has been renamed `PROGNAME&PRJ*1000.DEV4`. It is listed in the program file in DEV4.

Drill Down

From the Realm Master screen ([Figure 6-1 on page 132](#)), if you press **F5** (*Drill) with the cursor in any field with an asterisk (*), the system “drills down” to find the background information for that piece of data. The resulting screen depends on the field from which **F5** was pressed.

For example, pressing **F5** from the:

- *Security Group* field displays a screen to allow the entry of multiple users, groups or roles that are permitted or excluded from this rollout (see above).
- *Scan*, *Comp* (Compare), or *Roll* fields displays the Scanned Realms sub-screen similar to [Figure 6-2](#):

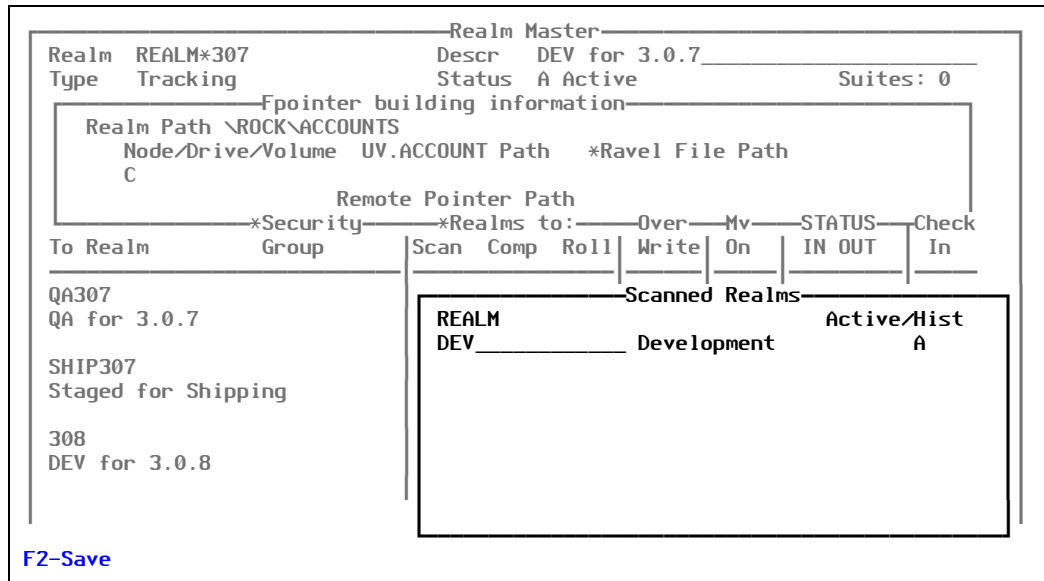


Figure 6-2: Scanned Realms sub-screen

This sub-screen displays whatever realm codes have been set up in the system thus far (Figure 6-1 on page 132).

Work Accounts

From the Realm Master screen (Figure 6-1 on page 132), if you press **F6** (Work Accts), the system displays the Work (Satellite) Accounts sub-screen.

If a working or satellite account has been defined for this project, it will appear in the sub-screen similar to Figure 6-3:

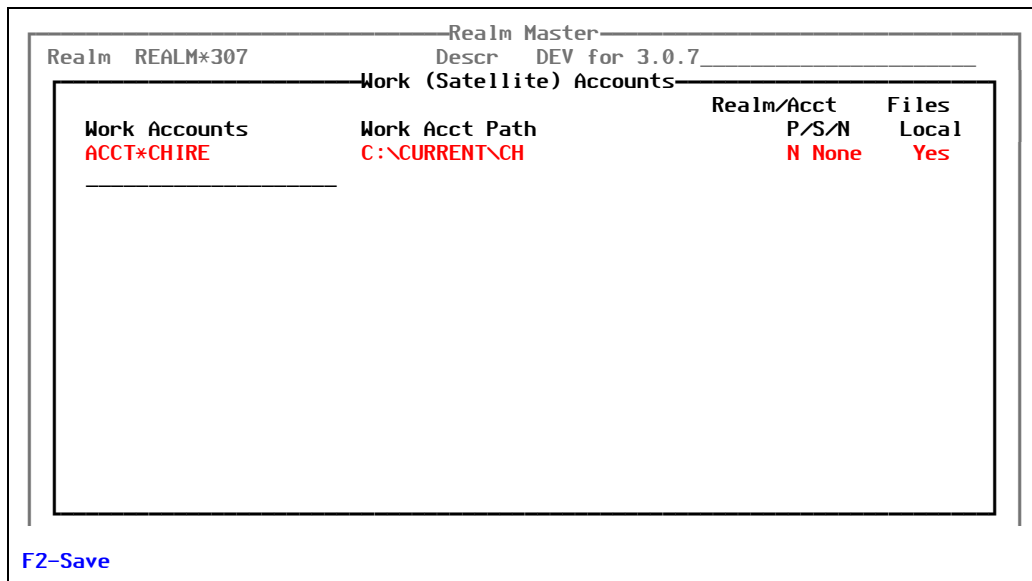


Figure 6-3: Work (Satellite) Accounts sub-screen

Work accounts are partially separate accounts, in that they share some or all of their software files with the main account for the realm.

The primary reason for listing a work account on this screen is to associate the account to a particular realm and to indicate to PRC that VOC changes (such as file pointers and program catalogs) need to be made in this account.

If there is a **WORK** or **WORK* <realm>** file set, PRC understands that those files are local to the work account and delivers into them separately.

Suites

From the Realm Master screen (Figure 6-1 on page 132), you press **F7** (Suites), but this is rarely used. For more information on suites, see "Suite Master" on page 152.

Additional Details

From the Realm Master screen (Figure 6-1 on page 132), if you press **F8** (Addl), the system displays the Realm Additional Details sub-screen, similar to the one shown in Figure 6-4:

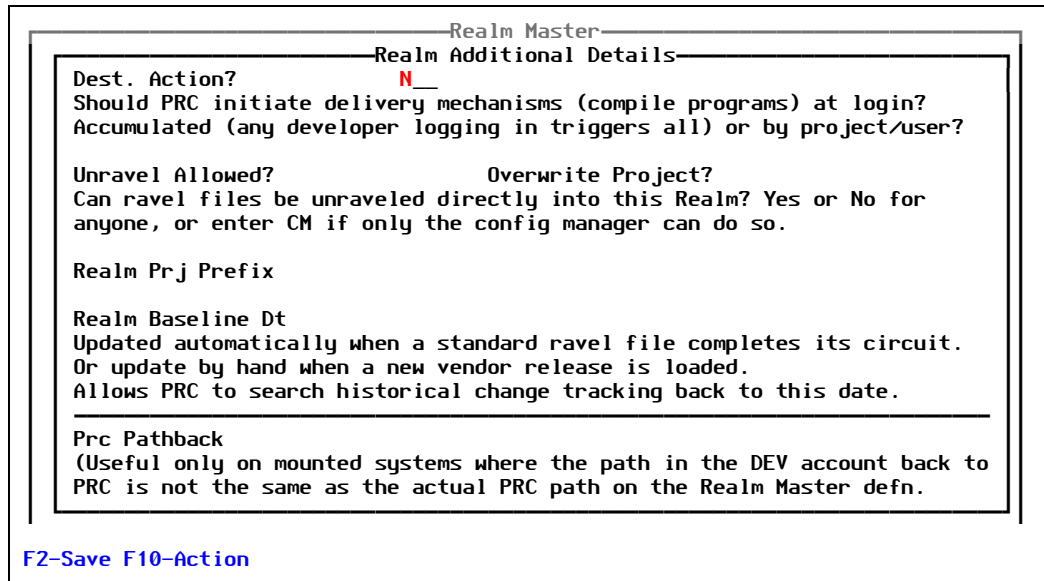


Figure 6-4: Realm Additional Details sub-screen

The *Dest Action* field indicates that this realm is a “receiving realm”—software is delivered into this realm—and the finishing portion of those deliveries should be performed in this realm. What those finishing actions will be is determined by other settings (see tip sheet), but kicking off the command to catalog the programs in that account is the most common. This field pertains to rollouts made directly into this realm, not unravels.

The *Unravel Allowed* field indicates whether projects are allowed to be unraveled in this account.

The *Overwrite Project* field indicates whether a project in a ravel file can be overwritten into the project file on that system.

The *Realm Prj Prefix* field indicates whether the realm ID will be included in projects opened in the realm.

The *Realm Baseline Dt* field represents a “birthdate” for the realm and is used during rollouts and unravels to determine how far back into closed projects conflicts should be scanned.

The *PRC Pathback* field is rarely necessary, but used for certain types of networking.

Archives

From the Realm Master screen (Figure 6-1 on page 132), if you press **F9** (Arch), the system displays the Archives sub-screen, similar to the one shown in Figure 6-5:

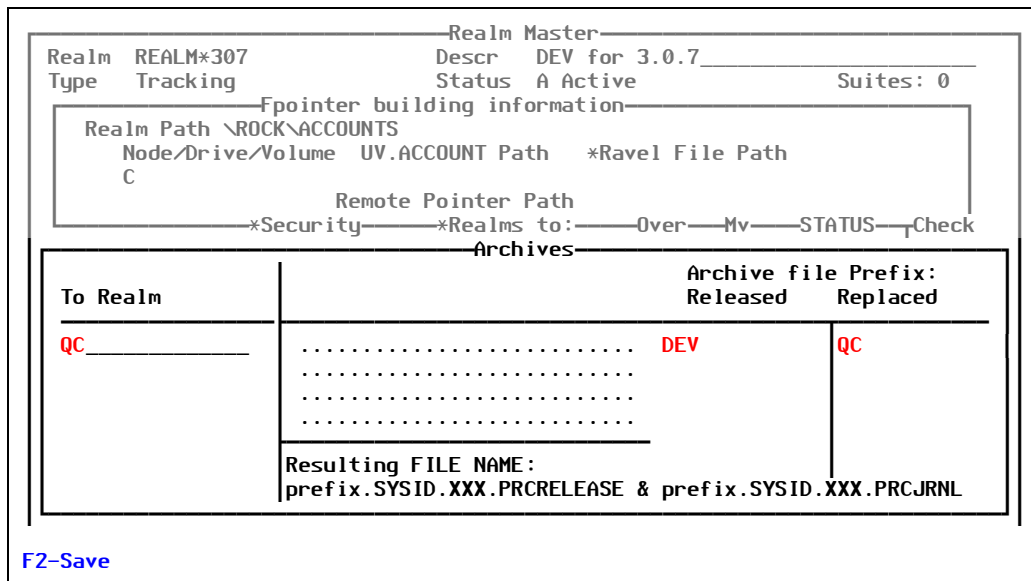


Figure 6-5: Archives sub-screen

You must enter values into these fields to enable archiving.

These fields were originally designed as yes/no prompts. Because of numerous discussions among different PRC users over what made sense, they eventually became a slot to specify the realm prefix for the file. Typically (in the author’s opinion, anyway), it makes sense for:

- The software being released to be archived with a prefix of the realm *from* which it is released
- The software being replaced to be archived with a prefix of the realm *into* which it is being replaced

This means that on a rollout from DEV to QC, the *Released* prefix would be **DEV** and the *Replaced* prefix would be **QC**.

Account Master

An account is a UniData, Universe, or Pick entity. PRC only cares about accounts for creating pointers. An account must simply be added to this screen with a description, so that it can be validated.

To work with the Account Master, perform the following:

1. Select **Setup > Configuration > Account Master**.

The system displays the Account Master screen, similar to the one shown in [Figure 6-6](#):

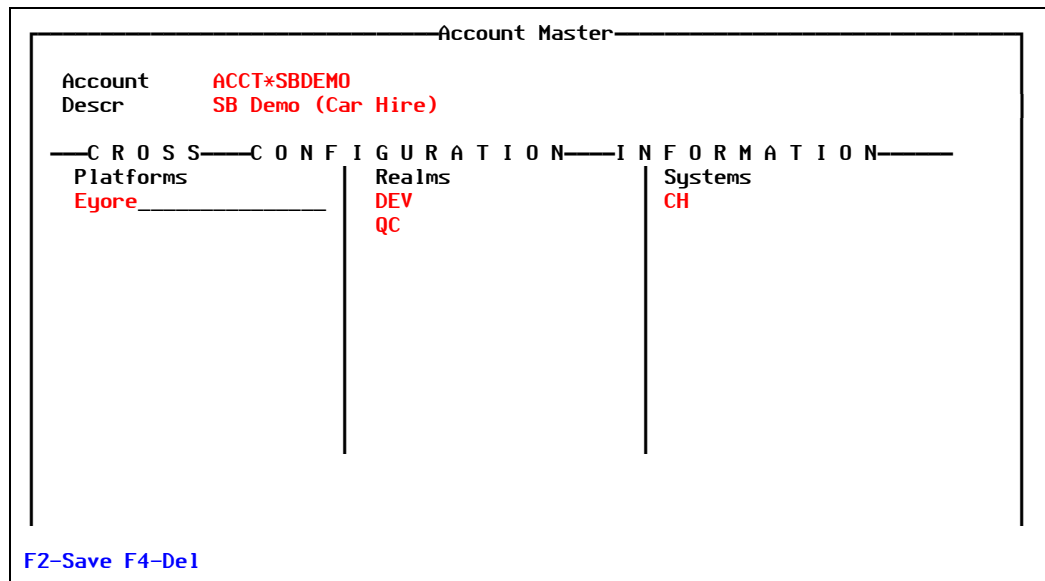


Figure 6-6: Account Master screen

Explanations of the fields are as follows:

Account For PRC purposes, this account information is only loaded so that validations can be performed throughout the system. (Think along the line of a spell checker.)

If you will be naming the actual accounts with a prefix or suffix that designates the realm (for example, *DEV.account* or *account.DEV*), it is only necessary to enter this "master account" record once for the base name of the account, without the prefix or suffix. If there is already a way to validate the account (known only for UniVerse platforms), this Account Master file is not necessary.

Press **F3** for a list of valid accounts.

Platforms The names of the platforms on which this account resides.

Press **F3** for a list of valid platforms.

Realms Each realm in which this system resides, along with its associated account name.

Press **F3** for a list of valid realms.

Systems *[Informational only]* The SYSIDs for all systems that are located within this account.

Press **F3** for a list of valid system IDs.

2. Fill in each field with the appropriate information.
3. Press **[F2]** (Save).

System Master

System is the SB+ system identifier (SYSID) for SB+ applications. For companies that do not use SB+ to develop their software, one dummy SB+ SYSID is created and that system is entered into the system master screen in PRC.

A “system,” in this context, is most like a module. Once a system has been added, projects can be opened for that specific system, files can be “owned” by a specific system, and user security clearances and authorizations can be specified by system.

To work with the System Master, perform the following:

1. Select **Setup > Configuration > System Master**.

The system displays the System Definitions screen, similar to the one shown in [Figure 6-7](#):

System Master						
System ID	SYSTEM*CH					
Descr	Car Hire					
				Initiate at Repository		
Realm	Account	P/S/N	Menu/Common	SYSID	Rev	Nbr Installed
DEV	SBDEMO	P Pre	Menu	MAS	1.0	09/01/06
QC	SBDEMO	P Pre	Menu		1.0	09/01/06

F2-Save F4-Del F5-Menus F6-Files F8-XConfig F10-Action

Figure 6-7: System Definitions screen

Explanations of the fields are as follows:

System ID The name of the SB+ system.

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

Realm	<p>Each realm in which this system resides, along with its associated account name.</p> <p>Press [F3] for a list of valid realms.</p>
Account	<p>The name of the account in which the system resides (in all realms). This is not a full path name, but is a regular account name in PICK, and so on.</p> <p>Press [F3] for a list of valid accounts.</p>
P/S/M	<p>Indicates how the realm ID is appended to the account or product name for actual account names. Valid entries are:</p> <ul style="list-style-type: none"> • P — Prefix • S — Suffix • N — Neither or “no convention” <p>Each real account name must be entered individually on the Account Master screen.</p> <p>On the Account Master screen, you may enter each account base name (or product name) one time; for example, AP or WALDO. These accounts may actually exist on disk with a prefix or suffix of the realm ID; for example, DEV . AP or WALDO . DEV.</p> <p>Press [F3] for a list of valid entries.</p>
Menu/Common	<p>Indicates whether you are using the SB+ Common to initialize PRC or a menu to invoke it. Valid entries are:</p> <ul style="list-style-type: none"> • COMMON — Common If PRC is to activate at the set common. • Menu — Menu Use the [F5] (Drill-down) to indicate what menus to invoke PRC (after display). <p>Press [F3] for a list of valid entries.</p>
SYSID	<p>The name of the global SYSID for a central repository. Central repository tracking refers to the situation where your organization is working in multiple accounts/SYSIDs within a realm—but you want item checked-out status to be centralized or global.</p> <p>If you want changes within this realm and SYSID to track in a central or other SYSID’s repository, enter that SYSID in this field. Useful if work will be done across accounts and systems.</p> <p>If you leave this field blank, a central repository will not exist. Most organizations leave this field blank.</p>

Rev Nbr The revision, version, or release number to identify this stage of the software for a module.

Installed The date that the tracking software was last installed on the system.

2. Fill in each field with the appropriate information.
 - If you press **F5** (Menus), see “Menus” on page 148.
 - If you press **F6** (Files), see “Files” on page 149.
 - If you press **F8** (XConfig), see “XConfig” on page 150.
3. Press **F2** (Save).

Menus

From the System Definitions screen (Figure 6-7 on page 146), if you press **F7** (Menus), the system displays the System Menus sub-screen, similar to the one shown in Figure 6-8:

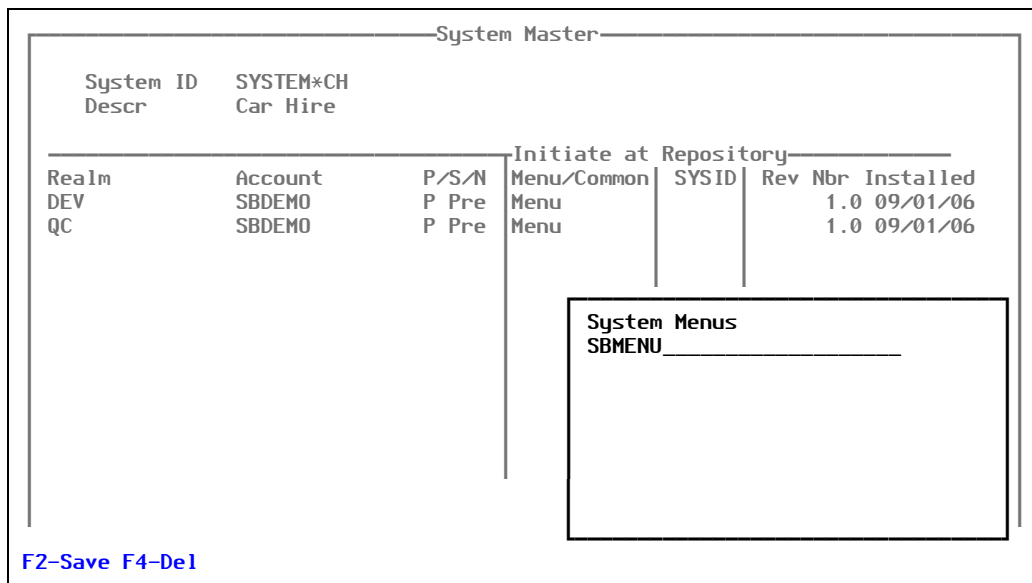


Figure 6-8: System Menus sub-screen

If you have elected to have PRC called from `MENUS`, this drill-down is where you indicate the name of the (SB+) menus from which PRC should be launched.

Files

From the System Definitions screen (Figure 6-7 on page 146), if you press **F6** (Files), the system displays the Which File Set sub-screen, similar to the one shown in Figure 6-9:

System Master

System ID SYSTEM*CH
 Descr Car Hire

Realm	Account	P/S/N	Initiate at Repository			
			Menu/Common	SYSID	Rev Nbr	Installed
D						06
Q	Which File Set?					06
	X <Cancel>	Owned	Tracked	Not-tracked	Not Rolled	

F2-Save F4-Del F5-Menus F6-Files F8-XConfig F10-Action

Figure 6-9: Realm Additional Details sub-screen

Make the appropriate selection to view the kind of files you wish to see. After you make the selection, the system displays the System File Sets sub-screen, similar to the one shown in Figure 6-10:

System Master

System File Sets

Set Name **FILES*OWN*CH**

Files	Data/ Rem Dict	Synonym Prefix	De- lim	*Over/ Part	Compiled *Y/N *Incl

F2-Save F4-Del F8-Alphabetize F9-*Add' l F10-Action

Figure 6-10: System File Sets sub-screen

For more information, see “File Sets” on page 73.

XConfig

From the System Definitions screen (Figure 6-7 on page 146), if you press **F8** (XConfig), the system displays the Systems’ Platforms sub-screen, similar to the one shown in Figure 6-4:

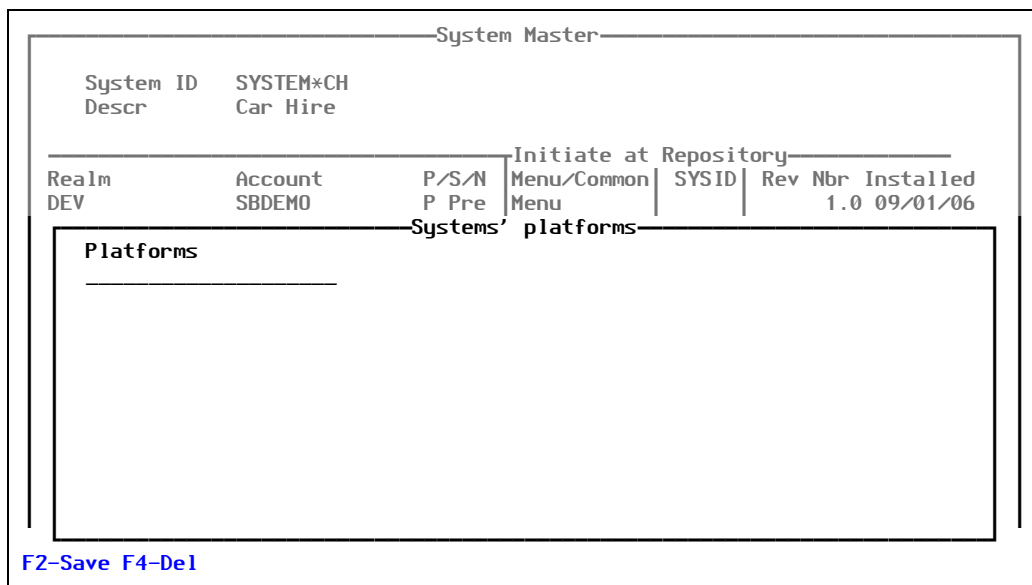


Figure 6-11: Systems’ Platforms sub-screen

This screen displays an internally updated cross-configuration of what realms and systems are on what platform.

Platform Master

A platform has no real function in PRC. However, the Platform Master is a way for you to enter the names of your hardware platforms, which allows you have a complete configuration management plan.

To work with the Platform Master, perform the following steps:

1. Select **Setup > Configuration > Platform Master**.

The system displays the Configuration screen, similar to the one shown in Figure 6-12:

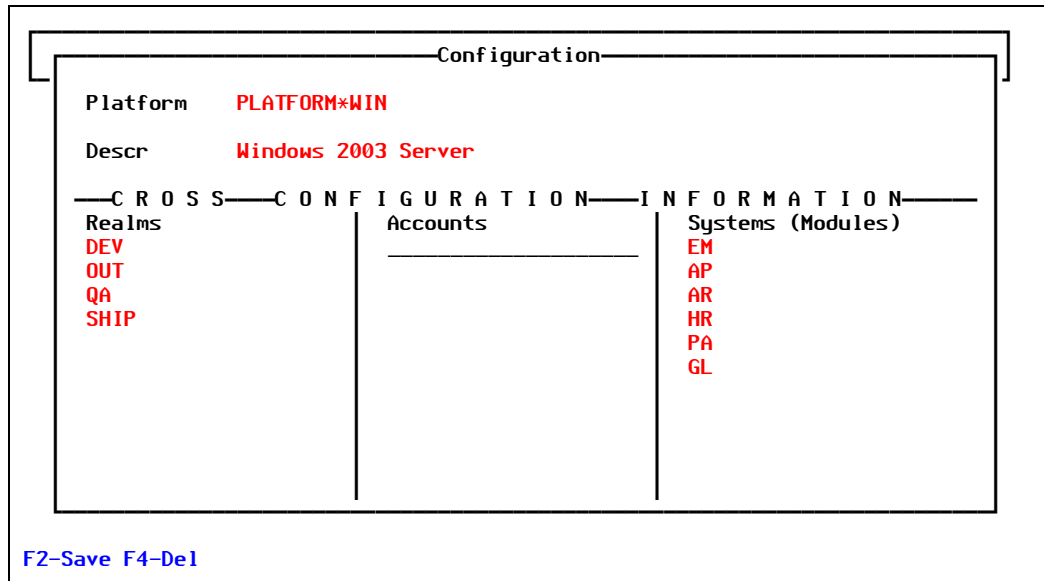


Figure 6-12: Configuration screen

Explanations of the fields are as follows:

Platform

The name of the computer or operating system platform, or its nickname, node name, or manufacturing name.

“Platform” refers to the actual hardware platforms that are included in the overall configuration. Use this screen to begin from a “top down” viewpoint. If only one computer platform involved, you do not have to come to that screen—but it may still be useful as a place to describe all of the realms, accounts, and systems (modules) on one screen.

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

Realms

Name each realm in which this system resides

Accounts

Name each account associated with the realms listed in the *Realms* column that reside on this platform.

Systems (Modules)

[Informational only] The SYSIDs for all systems that are located within this account.

2. Fill in each field with the appropriate information.
3. Press **[F2]** (Save).

Suite Master

Suites are rarely used, but can be useful in specific applications. You can use suites to create “flavors” of accounts within realms—to make a realm multi-dimensional, you might say. One way they have been used is to create another language version; for example, you have the **DEV** realm, then there a Spanish suite and an English suite.

To work with the Suite Master, perform the following:

1. Select **Setup > Configuration > Suite Master**.

The system displays the Suites screen, similar to the one shown in [Figure 6-13](#):

The screenshot shows a terminal window titled "Suite Master" with the following content:

```

Suite Id  SUITE*ES
Descr    Espanol
-----
Language  Espanol
Platform  Unidata
Version   5.1.3.2a
  
```

At the bottom left of the terminal window, the text "F2-Save F4-Del" is displayed.

Figure 6-13: Suites screen

Explanations of the fields are as follows:

Suite ID The code or ID that represents a suite (possibly the country code).

Press **[F3]** for a list of valid suites.

Language The language this suite is in (leave blank if no language issues).

Platform The name of the platform if this suite is platform-specific; for example, **UD** for UniData, **UV** for UniVerse, and so on).

Press **[F3]** for a list of valid platforms.

Version The version number if this suite is version-specific, in case, for example, rollouts from higher versions are not allowed.

Press **[F3]** for a list of valid versions.

2. Fill in each field with the appropriate information.
3. Press **F2** (Save).

Configuration Components

Use this screen to define the detailed items/entities that are tracked using PRC. The definitions here provide the tool used to modify, the location of the items, specific details about the configuration of the items, and a few rules of conduct about the item. A standard or typical U2 application is pre-configured and SystemBuilder's (SB+) specific tools and components are pre-defined.

To work with configuration components, perform the following steps:

1. Select **Setup > Configuration > Configuration Components**.

The system displays the following warning shown in [Figure 6-14](#):

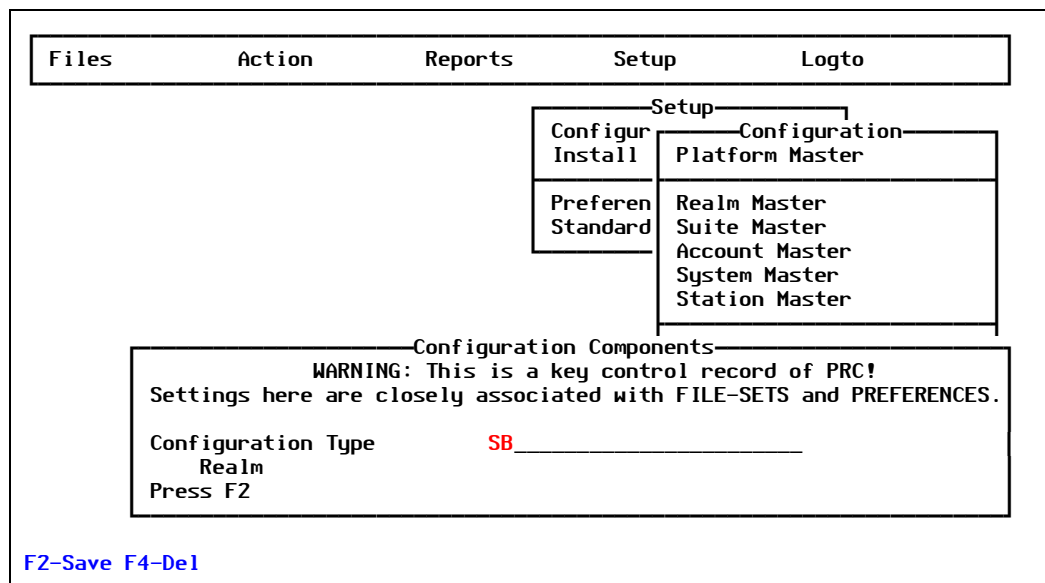


Figure 6-14: Warning

Explanations of the fields are as follows:

Configuration Type Primarily differentiates an SB+ development environment from "all others." Can be Realm-specific. Valid entries are:

- **SB** — SB+
- **SBR** — SB+, realm-specific
- **NON** — Non-SB+

- **NR** — Non-SB+, realm-specific

Press **F3** for a list of valid entries.

Realm

The specific realm for which you are defining the files and tools, if applicable.

Press F2

Frequently, fields such as this are placed on the screen to remind the user what to do next. It also makes a screen with only one prompt clearer to understand.

2. Fill in each field with the appropriate information.
3. Press **F2** (Save).

Station Master

A “station” is a physical instance of PRC. The concept of a station allows PRC to:

- Track two or more places where projects are created
- Identify on which station the project originated

The identification is accomplished by including it in the project key.

For example, if you have station 1 and station 2:

- Project 1 on station 1 would be **PRJ*1*1**
- Project 1 on station 2 would be **PRJ*2*1**

To work with configuration components, perform the following steps:

1. Select **Setup > Configuration > Station Master**.

The system displays the Station Master screen shown in [Figure 6-15](#):

Station Master

Station ID _____

Description

Development?

F2-Save F4-Del

Figure 6-15: Station Master screen

2. Fill in the information for each field, as necessary.
To select a previously-defined station, press **F3** in the *Station ID* field.
3. When you are done, press **F2** (Save).

Setup Menu

This chapter explains the remaining menu options in the Setup menu, which you use to further define your PRC system.

When you select **Setup**, the system displays the Setup menu, as shown in [Figure 7-1](#):

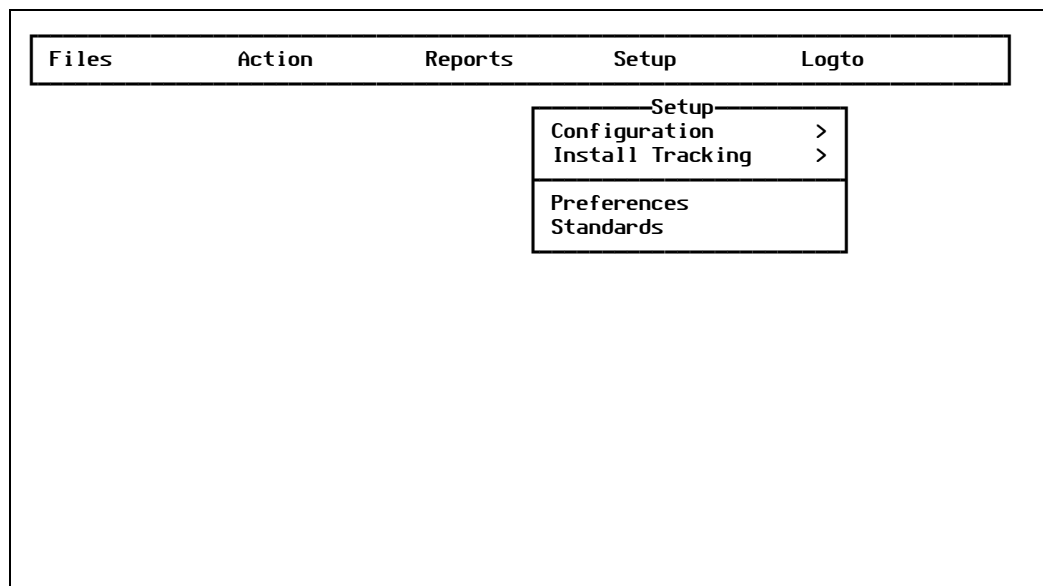


Figure 7-1: PRC Administration Setup menu

For explanations of the **Configuration** menu options, see [Chapter 6, “PRC Configuration.”](#)

Install Tracking

Once PRC has been configured with realms and their path locations, systems and their account names, you must install the tracking feature to deliver the components of PRC that are necessary for routine tracking into those accounts.

- The first item on the install tracking menu, Name PRC files/items, lists all the bits and pieces that get moved into the other accounts.
- Install Tracking moves those pieces over and catalogs the programs.

- Un-install Tracking un-delivers the components and de-catalogs the programs.

Name PRC files/items

This menu option lists all of the items that will be installed on the nominated account during the Install Tracking process.

To name files and items in PRC, perform the following steps:

1. Select **Setup > Install Tracking > Name PRC files/items**.

The system displays the Tracking Items screen, similar to the one shown in [Figure 7-2](#):

PRC Tracking Items				
Pointed to	Moved over			
Prc Files	Prc Process	Prc Defn	Help Items	Prc Menus
PRC.ACTIVITY	BUMP	PRC.ABOUT	PRC.COMPARE.ITEM	PRC.CUSTOM
PRC.ATPROGS	ECHO.OFF.GUI	PRC.ACCOUNTX	PRC.CONFLICT.SCA	PRC.PRCMEN
PRC.CIA	ED	PRC.ACCOUNTX0	PRC.FLICK.ITEMS	PRCCUST
PRC.CLASS	FD.LOG	PRC.ACTIVATE	PRC.KNIT.HELP	PRCLITE
PRC.CLIENT	FSRCH	PRC.ADD.PRJ	PRC.PROJECT.B.FI	PRCMENU
PRC.CODE	GLOBAL.LOG	PRC.ADD.SUB	PRC.PROJECT.B.SO	PRCMENU.HE
PRC.CODE.AUD	I*DMSHELL*COMPARE.	PRC.AND.OR	PRC.TRACE.HISTOR	PRCMENU.MG
PRC.DEPT	I*PRC.CODE*PASSWOR	PRC.ASK	PRCMENU.B.4	PRCMENU2
PRC.IA	I*PRC.CODE*PASSWOR	PRC.ASSIGNX0	PRCMENU.B.5	
PRC.KNIT.COD	I*PRC.KNIT.CODE*MA	PRC.AUTO.SISTER	PRCMENU.B.6	Voc/Md
PRC.MAILTEXT	I*PRC.NOTES*DOCLIN	PRC.AUTO.SUB	PRCMENU.B.7	F1
PRC.MVPROGS	I*PRC.PROJECT*MAIN	PRC.BACKUP.D	PRCMENU.REM	FNAME
PRC.PRCWORK	I*PRC.PROJECT*MAIN	PRC.BADWORD.WARN	PRCMENU2.REM	FPATH
PRC.PRIORITY	I*PRC.PROJECT*MAIN	PRC.BASEX		ID
PRC.PROJECT	I*PRC.PROJECT*PRC.	PRC.BASEX0		KEY
PRC.PROJECT.	I*PRC.PROJECT*PRC.	PRC.BLOWUP.OPT		PRCMENU
PRC.RBPROGS	I*PRC.PROJECT*TEST	PRC.BRANCH.DEL		TITLE
PRC.REQUEST	I*PRC.PROJECT*TEST	PRC.BRANCH.KNIT.		

F2-Save F4-Del

Figure 7-2: Tracking Items screen

Explanations of the fields are as follows:

- Prc Files** The names of all of the PRC files for which a new system must have Q-pointers. PRC provides you with a complete list by default.
- Prc Process** The names of all of the PRC processes that must be copied into a new account/system to install tracking.
- Prc Defn** The name of all of the DEFN items (dialog boxes, code tables) that must be installed in a new account/system for tracking purposes.
- Help Items** PRC tracking help items that must be moved into the tracking system's help file.
- Prc Menus** The PRC menus.

Voc/Md

The VOC/Master Dictionary.

2. Fill in each field with the appropriate information.
3. Press **[F2]** (Save).

Install Tracking

“Installing tracking” refers to PRC enabling a working account—it places some processes, definitions, menus and help items where they are accessible from the account(s) where work will be done. You can designate the PRC system files as the global files so that these components need not be moved into the working account itself.

The first item on the Install Tracking menu is a screen that displays all components that will be installed on the destination “tracked” account. You may add to or delete from these lists to have additional components of your own installed or to disable PRC features.

Except where “short named” for convenience, all components begin with **PRC**. The installation program checks for items in the destination account that are named the same as PRC’s components. This pre-check is run automatically if tracking has never been installed on a system before. Pre-checking may be run before the first installation by pressing **[F6]** (Pre) from the Install Tracking screen.

Note

The items that are most frequently in conflict with the item you are using are the **.LOG** processes. These processes, when present, are called automatically at file time by SB+. To review any **.LOG** processes already in use, see the SB+ Revision Control Menu, Modify Version Number.

The only **.LOG** processes that are allowed to exist are:

- **PD.LOG**
- **SD.LOG**
- **RD.LOG**
- **FD.LOG**
- **GLOBAL.LOG**

Any other process that ends in **.LOG** must be deleted from the **XXPROCESS** file and the **DMPROCESS** file.

If you have processes running in these now, they may be renamed and called from PRC. Please call for details.

To install tracking for a system, perform the following:

1. Select **Setup > Install Tracking > Install Tracking**.

The system displays the Install Tracking screen, similar to the one shown in [Figure 7-3](#):

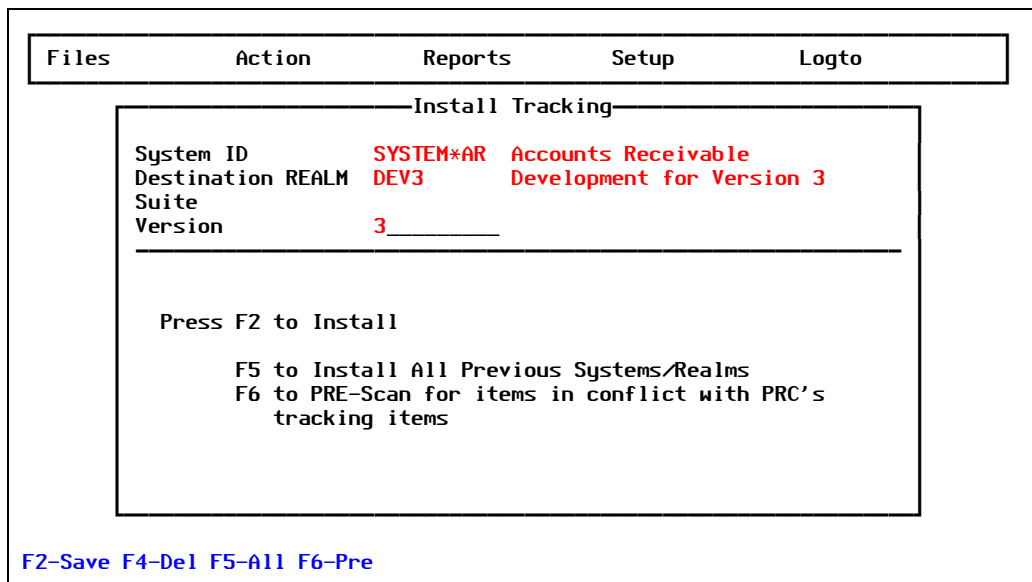


Figure 7-3: Install Tracking screen

Explanation of the fields are as follows:

- System** The ID of the system where this software resides.
- Destination Realm** The realm ID where the tracking is to be installed. Must be designated by a **T** realm type on the Realm Master.
- Suite** The suite against which tracking will begin, if applicable.
- Version** The version against which tracking will begin. *This version* must exist for *this system* in the Version Master file.
- Installing on** The account on which you are installing, where all the configuration and master file information ties together. The location where the tracking components will reside.

The system will display a message for confirmation before it begins to install tracking.

2. Fill in each field with the appropriate information.

The first time you install tracking, you must use this screen for each and every system in every realm where software changes are to be tracked.

- If you press **F5** (All), you can reinstall tracking on all systems and realms. You can use this feature for every subsequent installation (after you have already installed a system once).

This function changes the version number (only if version rollouts, which update the version number automatically, are not used) to upgrade PRC.

- If you press **[F6]** (Pre), the system will pre-check everything before installing. You can use this feature for every subsequent installation (after you have already installed a system once).
3. Press **[F2]** (Save) to install.

UN-install Tracking

This menu option un-delivers components and de-catalogs programs that were involved when you installed tracking.

To uninstall tracking capabilities, perform the following:

1. Select **Setup > Install Tracking > Un-install Tracking**.

The system displays the Strip Tracking screen, similar to the one shown in [Figure 7-4](#):

```

Files      Action      Reports      Setup      Logto
Strip Tracking

System ID  _____
Destination REALM
Suite
Version
-----
Press F2

REMEMBER TO FIRST:
** Check for any processes being called from PRC's
.LOG processes (e.g. PRC.PRELOG) which must be
renamed (to .LOGS) if they should still be run.
** Check for any processes that were run from PRC.LOAD
that must still be run from the first menu.

F2-Save F4-Del

```

Figure 7-4: Strip Tracking screen

Explanations of the fields are as follows:

System ID The SB+ system ID in which you originally installed tracking.

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

Destination Realm The name of the destination realm into which software will be rolled during this procedure.

Press **[F3]** for a list of valid realms.

- Suite** The suite in which you originally installed tracking.
- Version** The version number to be installed. This version number will get stamped in the housekeeping of that system, but must first exist on the Version Master here.

Press **F3** for a list of valid versions.

2. Fill in each field with the appropriate information.
3. Press **F2** (Save).

PRC Preferences

The preferences screen contains a number of sub-screens accessible from the **F6** (Addl Opts) key. This set of screens provide settings for all kinds of details about how PRC will perform, including what TCL commands are tracked and what kinds of files are created and many other items.

To set up system-wide preferences, perform the following:

1. Select **Setup > Preferences**.

The system displays the PRC Preferences screen, similar to the one shown in [Figure 7-5](#):

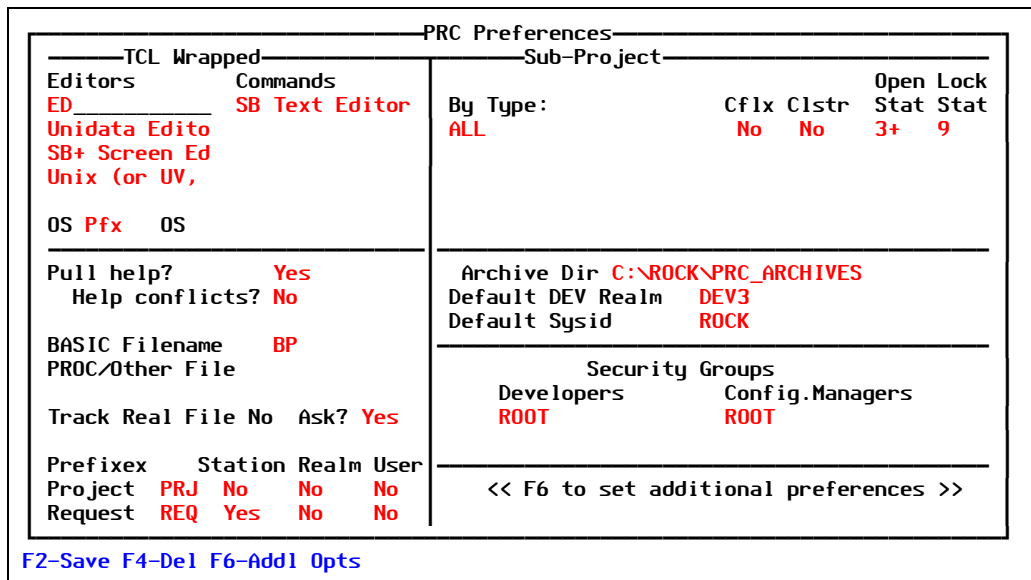


Figure 7-5: PRC Preferences screen

Explanations of the fields are as follows:

Editors

Indicates the editors to wrap. Wrapping will cause PRC to be invoke before and after each edit using each named editor. Certain behaviors differ on certain platforms. Valid entries are:

- **ED** — PICK line editor
- **SE** — SB+ screen editor
- **VI** — UNIX or UniVerse UNIX
- **AE** — UniData editor
- **EDIT-LIST** — Edit list
- **EDU** — UniVerse editor
- **FROG** — Frog
- **PEGASYS** — Pegasys (LaBonte)
- **JED** — Getronics
- **WIN.EDIT** — WIN.EDIT
- **EDIT.PLUS** — EDIT.PLUS
- **TED** — TED
- **BE** — BE
- **EDIT** — EDIT

Press **[F3]** for a list of valid entries.

Commands

Indicates the other OS commands to wrap. Valid entries are:

- **COPY** — Copy
- **DELETE** — Delete
- **CREATE.FILE** — Create file
- **DELETE.FILE** — Delete file
- **CLEAR.FILE** — Clear file
- **CREATE-FILE** — Create file
- **DELETE-FILE** — Delete file
- **CLEAR-FILE** — Clear file
- **SB.TEXT.EDITOR** — SB Text Editor (for **/MP** command)
- **CREATE.INDEX** — Create index
- **BASIC** — BASIC

Press **[F3]** for a list of valid entries.

OS Pfx

The prefix to be used on all OS-level verb wrappers.

Pref Subtype

The project type codes that restrict how sub-projects behave to within types.

Cflx

Indicates whether conflicts will create warnings. Valid entries are:

- **Y** — Yes
All projects will create conflict warnings.
- **N** — No
Sub-projects will not warn of conflicts with each other.

If multiple projects are desired, yet they have items in common it may be desirable to set this flag to **N** (No) so that sub-projects and sister projects can be worked on simultaneously without constant conflict warnings; however, sister projects must roll out together.

For example, project 1000 may have sub-projects 1000.1 and 1000.2. If the conflict flag is **N** (No), all three projects can be open in the same system simultaneously and edits to common source items will be allowed without a warning.

Clstr Indicates whether sub-projects must be forced onto the same rollout. Valid entries are **Y** (Yes) or **N** (No). Answer **Y** to ensure that projects and sub-projects stay together.

Stat The project status that claims priority over other conflicts, and against which sub-projects can be opened.

Include a plus sign (+) after the status code to indicate that status and above or a minus sign (-) to indicate that status and below.

The typical value of this field is **4** (Development Completed). For example, status **4** is the project priority that disallows conflicts. When a project is marked **4**, it will be “locked.” Whether it is seconds or days before you actually roll it out, no one can change any item associated with the project until after it is rolled out and conflicts are released.

To disable this feature, enter **6** in this field. That value ensures only live or closed projects have priority (but they are released anyway, so it will never happen).

If someone changes a program while it is included in a “locked” project, the person will be forced to make a copy with their changes. You may want to consider locking only project with status **5** (Tested) instead.

Another way you can lock projects is by assigning them priority double-zero (**00**).

Default DEV Realm The default **DEV** realm.

Archive Dir The directory path where archive files will be created.

Default SYSID The primary or global system ID. PRC uses it as the default value on screen prompts.

Pull help?	Indicates whether the system, when it “explodes” a project, will automatically pull all help items associated with items checked out to the project. Valid entries are Y (Yes) or N (No).
Help conflicts?	Indicates whether help items cause conflicts or just “roll along with it,” regardless of conflicts. Valid entries are Y (Yes) or N (No).
BASIC Filename	<p>The name of any secondary program file (besides <i>XXPROGS</i>) that should be tracked independently.</p> <p>The primary purpose of identifying a file name is to dedicate a column of the Review Source Items screen to items in this file. (Whether or not an editor or items in a particular file are tracked is handled elsewhere.) In other words, this column nominates a particular file as having its own column on /SRC.</p>
PROC/Other File	<p>The preferred <i>PROC</i> library file name (or other miscellaneous software file not associated with <i>SB+</i>). Will translate to <i>xxx.PROC-LIB (system.your filename)</i>.</p> <p>Many of the software systems that are re-engineered into <i>SB+</i> came from a legacy system that supported <i>PROCS</i>. Entering a file here will track edits against items in that file (per system). This can be <i>any</i> file that should be specifically tracked, not just <i>PROCS</i>; otherwise, edits generally end up in the <i>Misc Edits</i> column.</p>
Track Real File	Indicates whether you want to track the real file. Valid entries are Y (Yes, track the real file) or N (No, track the f-pointer).
Ask?	Indicates whether <i>PRC</i> should stop and ask before tracking as the real file name. Valid entries are Y (Yes) or N (No).
Project	The project identifier—a 3-character prefix, such as PRJ for “project.”
Pref Prj Realm	Indicates whether you prefer to have the realm included in the project ID. Valid entries are Y (Yes) or N (No).
Prj Station	Indicates whether the station ID should be included in the project ID. Valid entries are Y (Yes) or N (No).
Pref Prj Usr	Indicates whether the project ID will include the user ID. Valid entries are Y (Yes) or N (No).
Request	The 3-character prefix for requests or feedbacks or bug fixes (request for service, program change request, and so on).
Developers	The name of the security group to which programmers (or developers, whichever you call them) belong.

Typically, programmers or developers have user IDs that belong to a security group with very high security. If you place all developers into a particular group, you can capitalize on that fact by having certain processes behave differently when the user is a programmer.

For example, programs that are issued and waiting to be compiled can be launched automatically when the programmer logs in—which, normally, you would not want to disturb the end-users with that capability.

Config Managers

An SB+ group ID to include a whole group as configuration managers, or individual user IDs prefixed according to SB+ convention with the tilde (for example, ~SJOSLYN).

Configuration managers are able to perform tasks to other people’s projects that regular users are not able to perform (bump to, flick to, flick from).

Rolling out security is defined on the realms and is not controlled here.

- 2. Fill in each field with the appropriate information.

Press **F6** (Addl Opts).

The system displays the More Preferences screen, similar to the one shown in [Figure 7-6](#):

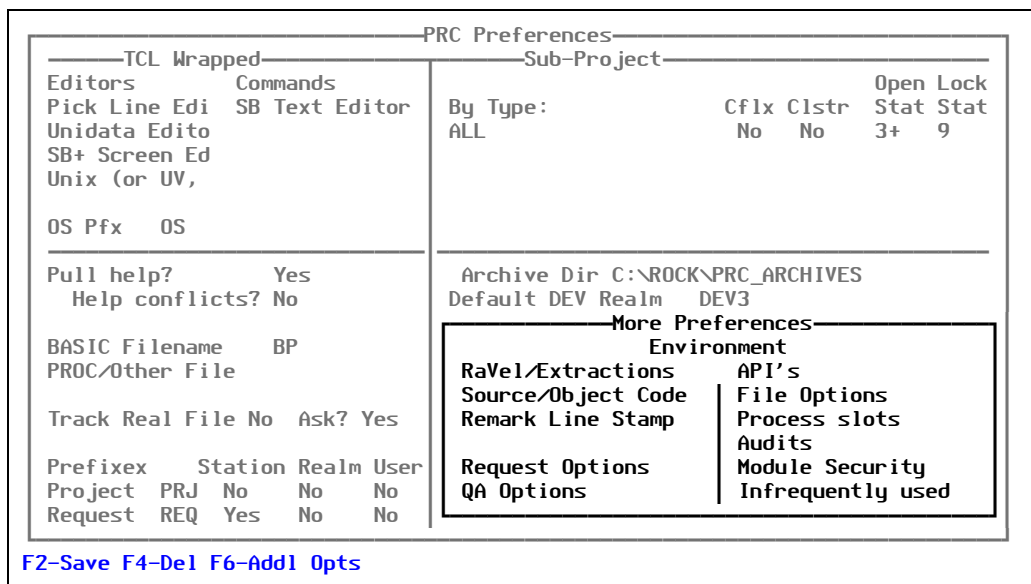


Figure 7-6: More Preferences screen

If you select Environment, see “Environment” on page 167.

If you select Ravel/Extractions, see “Ravels and Extractions” on page 170.

If you select Source/Object Code, see “Source and Object Code” on page 173.

If you select Remark Line Stamp, see “Remark Line Stamp” on page 176.

If you select Request Options, see “Request Options” on page 178.

If you select QA Options, see “QA Options” on page 179.

If you select APIs, see “APIs” on page 181.

If you select File Options, see “File Options” on page 182.

If you select Process Slots, see “Process Slots” on page 184.

If you select Audits, see “Audits” on page 187.

If you select Infrequently Used, see “Infrequently Used Options” on page 189.

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

Environment

To set up your environment preferences from the More Preferences screen (Figure 7-6 on page 166), select **Environment**.

The system displays the Environment subscreen, similar to the one shown in Figure 7-7:

```

PRC Preferences
-----TCL Wrapped-----Sub-Project
Editor Pick L Environment Lock Stat
Pick L 9
Unidat  Operating System/Platform UDNT
SB+ Sc  Pointer Type F
Unix (  Dmprocess? No
OS Pfx  Global? Yes
Pull h  (Development Environment)
Help    Configuration Table System Builder
BASIC  Subtype FWD
PROC/O Where Used (Off/Which) Off
Track

Prefix Station Realm User Request Options Audits
Project PRJ No No No QA Options Module Security
Request REQ Yes No No Infrequently used

F2-Save F4-Del

```

Figure 7-7: Environment subscreen

Operating System/ Platform

The code or description that represents the operating system or platform upon which the software system is being developed. Valid entries are:

- **UVNT** — UniVerse on Windows NT (2000, XP, 2003)

- **UDNT** — UniData on Windows NT (2000, XP, 2003)
- **R83** — Pick Systems Release R83
- **UV** — Vmark UniVerse
- **UD** — UniData
- **SB** — SBOPEN
- **PI** — PI/OPEN
- **AP** — Advanced Pick

Press **F3** to select from the list of valid entries.

Pointer Type

The preferred type of file pointer, if the operating system offers a choice. Valid entries are:

- **F** — F-pointer (literal paths)
- **Q** — Q-pointer (relative, PICK-style)

Press **F3** to select from the list of valid entries.

Dmprocess?

Indicates whether processes will be moved into **DMPROCESS** for global access. Valid entries are:

- **Y** — Yes, move processes into **DMPROCESS**
- **N** — No, move them into the **XXPROCESS** file of each tracked system, where **XX** represents the system

Press **F3** to select from the list of valid entries.

Global?

Indicates whether global processes, definitions, and menu files can be used from PRC. Valid entries are:

- **Y** — Yes, global processes, definitions, and menu files can be used from PRC
- **N** — No, they cannot be used from PRC

IMPORTANT

PRC will look at the global file names *in the PRC system*. If these are PRC, then each “installed system” will be referenced to PRC as the global file location. If you are already using a global area, be sure to indicate these file names in the PRC CONTROL file (in this case, the installation program will place the required processes, definitions, menus, and help files *into* those global files. If this preference is **Y** (Yes)—but no global files are nominated—PRC files are assumed as the global set. If this preference is **N** (No), PRC installs into each local definition, help file, menu file, and either the local or the **DMPROCESS** file (depending on the answer to that preference).

Press **F3** to select from the list of valid entries.

Configuration Table The way in which your system is configured for development. Valid entries are:

- **SB** — SystemBuilder (SB+)

Use this entry if all of your development work is in SB or through an editor.
- **NON** — NON-SB+ development

Use this entry if SB+ is not present.
- **BOTH** — SB+ & NON-SB+ development

This entry indicates an additional tool or environment.

For example, if you are an SB+ shop and you do not want to create a secondary **/SRC** screen with additional columns for *more* types of software components, select **SB** for this field.

Do not worry: your BASIC programs, processes, and anything that you create with an editor will be fine. Anyone using a special version of PRC for environments in which SB+ is not present—or where SB+ is present along with another tool—may want to consider configuring their own **/SRC** screen. If you do not know what this all means, give SJ+ a call to discuss it.

Press **F3** to select from the list of valid entries.

Subtype Enter the Subtype. Valid entries are:

- **N** — None
- **O** — Odyssey
- **A** — Epicor/Avante
- **P** — Prelude/ADS
- **F** — FWD
- **M** — Masterpack
- **I** — Information (Prime)
- **AAON** — Aaon
- **J** — Jenkon/Summit
- **C** — Companion/Megawest 2000

Press **F3** to select from the list of valid entries.

Where Used Indicates whether PRC should refer to the `WHERE USED` tables. If you are not sure what this means, enter **0** (zero, off) and ask SJ+ about it. Valid entries are:

- **0** — Off
- **A** — Advanced/Avante
- **B** — BASIC or generic

Press **F3** to select from the list of valid entries.

When you have finished with the settings, press **F2** (Save). The system returns you to the More Preferences screen (Figure 7-6 on page 166).

Ravels and Extractions

To set up your preferences for ravels and extractions from the More Preferences screen (Figure 7-6 on page 166), select **Ravel/Extractions**.

The system displays the Extraction Options subscreen, similar to the one shown in Figure 7-8:

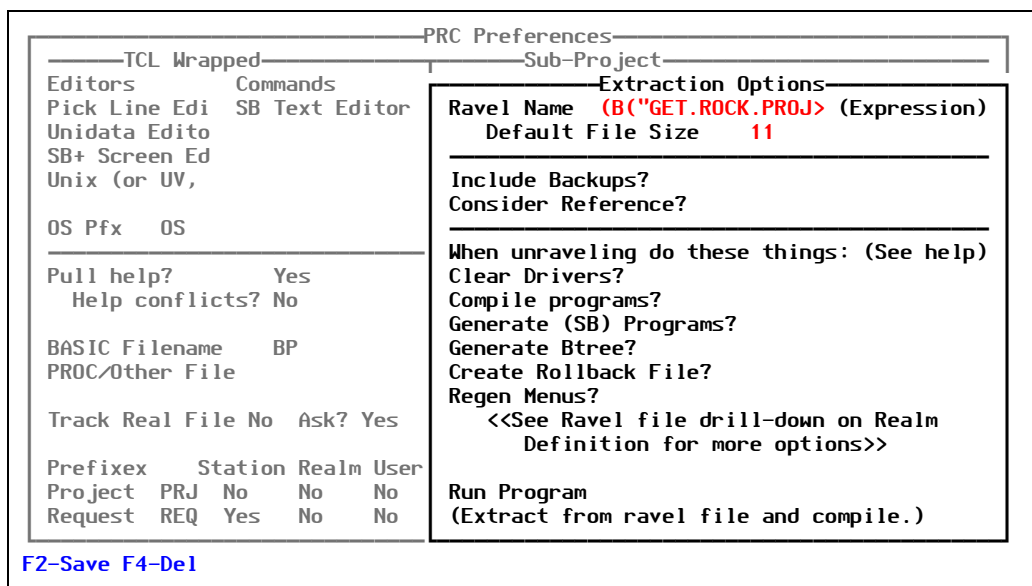


Figure 7-8: Extraction Options subscreen

Ravel Name

A valid SB+ expression. During a rollout, the ravel file will default from here, but may be overridden at that time.

An “expression” is something in SB+ language that can be resolved; for example, a literal string concatenated to a variable, such as one of the common variables. You can enter the following as an expression:

("PRC_" :@KEY)

What you would get at runtime would be the literal string **PRC** (because it is in quotes) plus whatever the value of **KEY** was at the time it ran. The syntax for this is all part of SB+ that is beyond the scope of this document.

What you are doing here specifically is setting up what PRC will use as a ravel file name. So whatever you enter as an expression in this field will dictate the ravel file's name whenever someone rolls out a project.

Continuing with the example above, when someone rolls out project 19 to a "remote realm" (to create a ravel file), the ravel file will come up with **PRC_19** as the *suggested* file name. However, the user can override the suggestion and change it to **BANANA** if she or he chooses—and that becomes the ravel file's name.

An example is shown in [Figure 7-9](#) below:

```

Project Master Maintenance
PROJECT      98 PRC      4 Dev Complete      Realm D
Problem deleting subprojects (Keystone)      PRC Internal Develop
-----
Project Rollout
Src Rlm      (rolling from)
D            PRC Internal Develop

Target Realm Suite      Scan From Ravel Name
OUT          PRJ98
OUT

Press F2 to launch rollout or scroll down through movement history
-----MOVEMENT HISTORY-----
Rolled out to      Rolled Back from
Realms      on      at      Ravel File      Realms      on      at
  
```

F2-Launch

Figure 7-9: Example of an automatic ravel file name

The ravel file is in red and came up automatically due to the setting of this field. If the person conducting the rollout pressed **F2** at this point, **PRJ98** would be the ravel file name; otherwise, the person can enter any name and *then* press **F2** and get that as the ravel file name.

Default File Size

Define the default ravel file size.

Include Backups?

Indicates whether the original backup (before work for this project was done) of each item will be included in the ravel file. Valid entries are:

- **Y** — Yes, include the original backup
- **N** — No, do not include it

Press **F3** to select from the list of valid entries.

Consider Reference? Indicates whether conflicts will be ignored if the two different projects address the same reference. Valid entries are:

- **Y** — Yes, pay attention to conflicts to the same reference
- **N** — No, ignore conflicts

Press **F3** to select from the list of valid entries.

Clear Drivers?

Specific drivers that are necessary are normally cleared during an online rollout. This field indicates whether you want the whole driver file cleared on a remote (extract) rollout. Valid entries are:

- **Y** — Yes
- **N** — No
- **X** — Ask

The definitions on this screen are stored in a control item that is delivered with each ravel file. The control item defines a self-extracting set of procedures. In other words, these options build a control item that goes in the ravel file that then controls what happens when the ravel file is unraveled on the other machine.

The control item and its procedures will be ignored if the ravel file is unraveled on a system with PRC Site[®] installed, using the Unravel program on the **PRCSITE** menu. The control item defined on that site will be used instead.

Press **F3** to select from the list of valid entries.

Compile programs? Indicates whether the programs that are moved be compiled on the destination system. **Remember:** Whether or not they are compiled, they still need to be cataloged. Valid entries are:

- **Y** — Yes
- **N** — No
- **X** — Ask

Press **F3** to select from the list of valid entries.

Generate (SB) Programs?

Indicates whether the system will regenerate code for all processes marked GC after extraction on the destination system. Valid entries are:

- **A** — Regenerate *all*
- **D** — Regenerate *delivered* (generated)
- **N** — No, do not regenerate
- **X** — Ask

Press **F3** to select from the list of valid entries.

- Generate Btree?** If you are using SB+ Btrees and the definition record has changed (\$CROSS.REF), use this field to indicate whether the Btree will be regenerated to the new definition. Valid entries are:
- **Y** — Yes, regenerate the Btree
 - **N** — No, do not bother
 - **X** — Ask
- Press **F3** to select from the list of valid entries.
- Create Rollback File?** Indicates whether the unravel program will create a “rollback ravel file” for un-unraveling or comparing. Valid entries are:
- **Y** — Yes, have the unravel program create a rollback ravel file
 - **N** — No, do not bother
- Press **F3** to select from the list of valid entries.
- Regen Menus?** Indicates whether SB+ (GUI, type 5) menus should regenerate upon delivery. Valid entries are:
- **Y** — Yes, regenerate GUI menus upon delivery
 - **N** — No, do not bother
- Press **F3** to select from the list of valid entries.
- Run Before Unravel** The name of the program (FILE! PROGRAM) to be pulled from the ravel file, compiled, and run *before* an unravel is performed. If the system does not find the name in the ravel file, it will attempt to run the file from the destination system. If the system cannot find it, the unravel proceeds and does not run anything.
- Run After Unravel** The name of the program (FILE! PROGRAM) to be pulled from the ravel file, compiled, and run *after* an unravel is performed. If the system does not find the name in the ravel file, it will attempt to run the file from the destination system. If the system cannot find it, the unravel proceeds and does not run anything.

When you have finished with the settings, press **F2** (Save). The system returns you to the More Preferences screen ([Figure 7-6 on page 166](#)).

Source and Object Code

To set up your source code and object code preferences from the More Preferences screen ([Figure 7-6 on page 166](#)), select **Source/Object Code**.

The system displays the Generated Code Options subscreen, similar to the one shown in Figure 7-10:

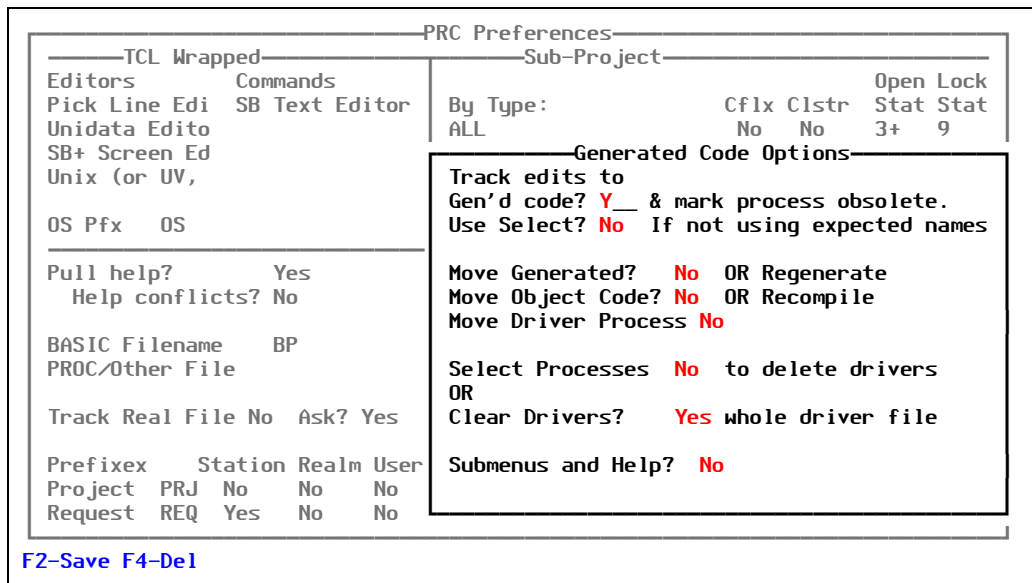


Figure 7-10: Generated Code Options subscreen

Gen'd code? Indicates whether BASIC code will generated and tracked. Valid entries are:

- **Y** — Yes
- **N** — No

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

Use Select? Indicates whether the system will select all processes to check for generated code. Valid entries are:

- **Y** — Yes, SELECT all processes; look for generated code
- **N** — No, every process must be named to match the generated program exactly

If you indicate here that you will only use SB+ standard naming convention (by entering **N** in this field), then PRC knows how to build the name of the drivers it needs to delete. That choice is much faster than having it select the whole file for any drivers that may call the particular process.

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

Move Generated? Indicates whether generated code will be moved; otherwise, the processes are moved and code must be regenerated at the destination. Valid entries are:

- **Y** — Yes, move generated code to the destination site
- **N** — No, move processes and regenerate at the site

Press **F3** to select from the list of valid entries.

Move Object Code? Indicates whether the object code and catalog entries will be moved; otherwise, programs must be recompiled and cataloged when they are moved. Valid entries are:

- **Y** — Yes, move object code and catalog entries
- **N** — No, recompile and recatalog programs at the site

Press **F3** to select from the list of valid entries.

Move Driver Process Indicates whether the I* and R* processes associated with any changed screen or report writer report will be pulled onto the project. Valid entries are:

- **Y** — Yes
- **N** — No

Press **F3** to select from the list of valid entries.

Select Processes Indicates whether processes associated to screen definitions on the project will be pulled onto the project as well, primarily to delete the specific drivers. Valid entries are:

- **Y** — Yes (default value)
- **N** — No

Press **F3** to select from the list of valid entries.

Clear Drivers? Indicates whether the whole driver file will be cleared each time a rollout is performed, as an alternative to selecting the processes. Considered “overkill,” but probably faster and harmless. Valid entries are:

- **Y** — Yes, clear entire driver file at each rollout
- **N** — No, do not bother

Press **F3** to select from the list of valid entries.

Submenus and Help?

Indicates whether PRC will gather all submenus and related help any time a type 5 (GUI) menu is delivered. Valid entries are:

- **Y** — Yes
- **N** — No

Press **F3** to select from the list of valid entries.

When you have finished with the settings, press **F2** (Save). The system returns you to the More Preferences screen (Figure 7-6 on page 166).

Remark Line Stamp

To set up your remark line stamp parameters from the More Preferences screen (Figure 7-6 on page 166), select **Remark Line Stamp**.

The system displays the Editor Options subscreen, similar to the one shown in Figure 7-11:

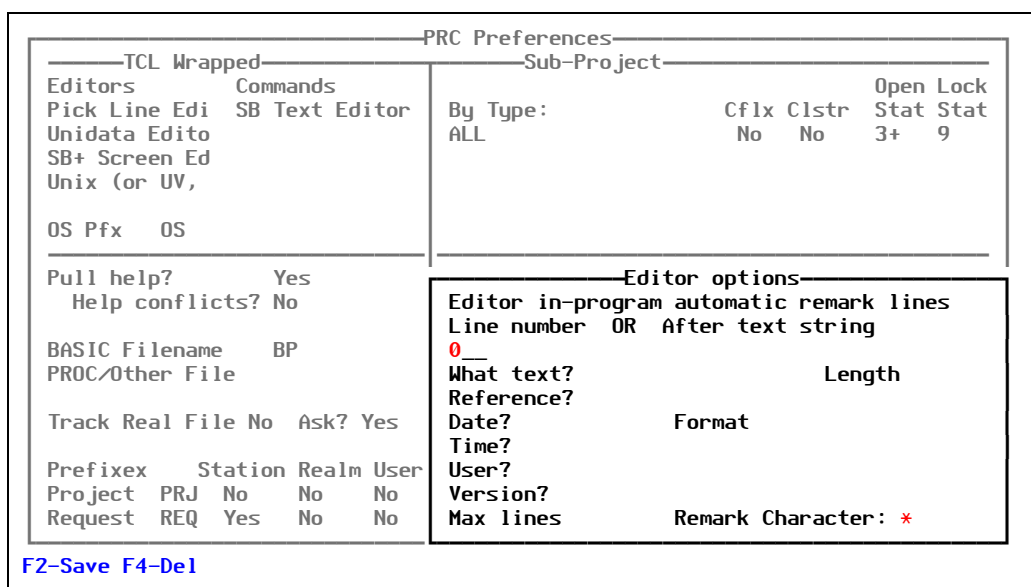


Figure 7-11: Editor Options subscreen

Line number

The line number on which PRC should place a remark line in programs. Valid entries range from **-1** (the bottom of the program) to **0** (to turn off or to use “locate text option”) through **50**.

After text string

The string entered here will be located in the program, and remarks inserted on the first line *after* this string is found.

What text?

Indicates which text to include on the edited item: the notes or the project title. Valid entries are:

- **NOTE** — Detail documenting
- **TITLE** — Project title text

Press **F3** to select from the list of valid entries.

Length	<p>The number of characters on a line; usually, 40 characters fit on a line with the other stuff.</p> <p>If you want to use the full length of the line and allow the text to wrap to the next line, leave this field blank.</p>
Reference?	<p>Indicates whether the system will include the reference information on the remark line. Valid entries are:</p> <ul style="list-style-type: none">• Y — Yes, include the reference• N — No <p>The reference is often used for outside document numbers; for example, the help desk.</p> <p>Press F3 to select from the list of valid entries.</p>
Date?	<p>Indicates whether the system will include the date in the text inserted on the edited item. Valid entries are:</p> <ul style="list-style-type: none">• Y — Yes, include the date• N — No <p>Press F3 to select from the list of valid entries.</p>
Format	<p>Defines the date format; for example, D2/, D4Y, and so on.</p>
Time?	<p>Indicates whether the system will include the time in the text inserted on the edited item. Valid entries are:</p> <ul style="list-style-type: none">• Y — Yes• N — No <p>Press F3 to select from the list of valid entries.</p>
User?	<p>Indicates whether the system will include the USER in the text inserted on the edited item. Valid entries are:</p> <ul style="list-style-type: none">• Y — Yes• N — No <p>Press F3 to select from the list of valid entries.</p>
Version?	<p>Indicates whether the system will include the current version number. Valid entries are:</p> <ul style="list-style-type: none">• Y — Yes• N — No <p>Press F3 to select from the list of valid entries.</p>

Max lines Define the maximum number of remark lines for the remark section. If you enter a number, additional (older) remarks will be rolled to the bottom.

When you have finished with the settings, press **F2** (Save). The system returns you to the More Preferences screen (Figure 7-6 on page 166).

Request Options

To set up your options concerning requests from the More Preferences screen (Figure 7-6 on page 166), select **Request Options**.

The system displays the Request Preferences subscreen, similar to the one shown in Figure 7-12:

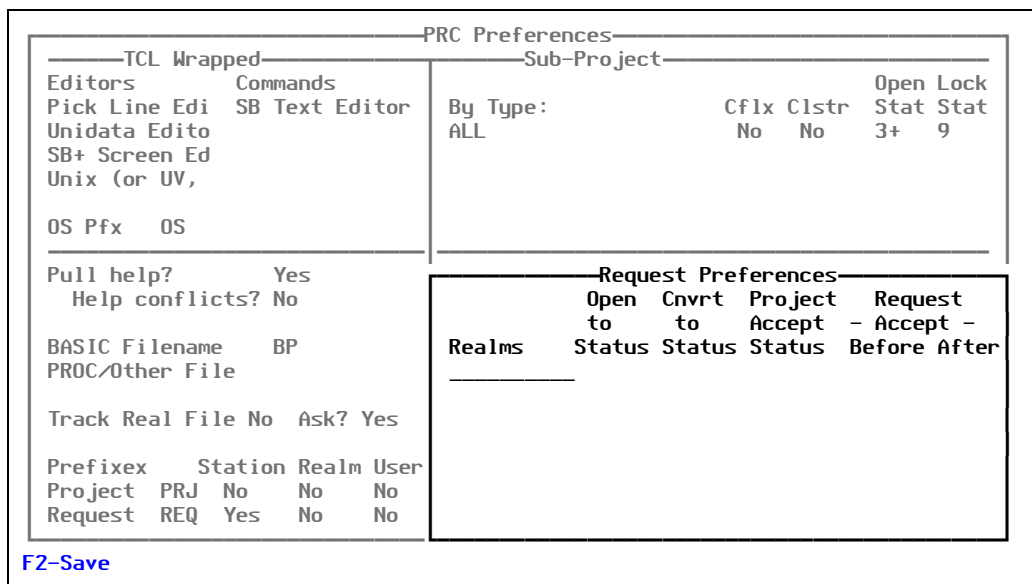


Figure 7-12: Request Preferences subscreen

Realms The realms in which requests will be opened and accepted. Typically, the selection is a live or production realm for the convenience of the business users.

Press **F3** to select from the list of valid entries.

Open to Status The status code that represents the initial state in which a request should originate. For more information, see “Status Codes” on page 29.

Press **F3** to select from the list of valid entries.

Cnvert to Status

The status code that represents the state a request becomes when it has been converted into a project. For more information, see [“Status Codes” on page 29](#).

Press **F3** to select from the list of valid entries.

Project Accept Status

The status code that represents the state a project must be in order to be accepted.

Press **F3** to select from the list of valid entries.

Request Accept: Before

The status code that represents the state a request must be *before* it can be accepted.

Press **F3** to select from the list of valid entries.

Request Accept: After

The status code that represents the state of a request *after* it has been accepted. In other words, once a request has been accept, it becomes this state.

Press **F3** to select from the list of valid entries.

When you have finished with the settings, press **F2** (Save). The system returns you to the More Preferences screen ([Figure 7-6 on page 166](#)).

QA Options

To set up your options concerning quality assurance and testing from the More Preferences screen ([Figure 7-6 on page 166](#)), select **QA Options**.

The system displays the QA Preferences subscreen, similar to the one shown in [Figure 7-13](#):

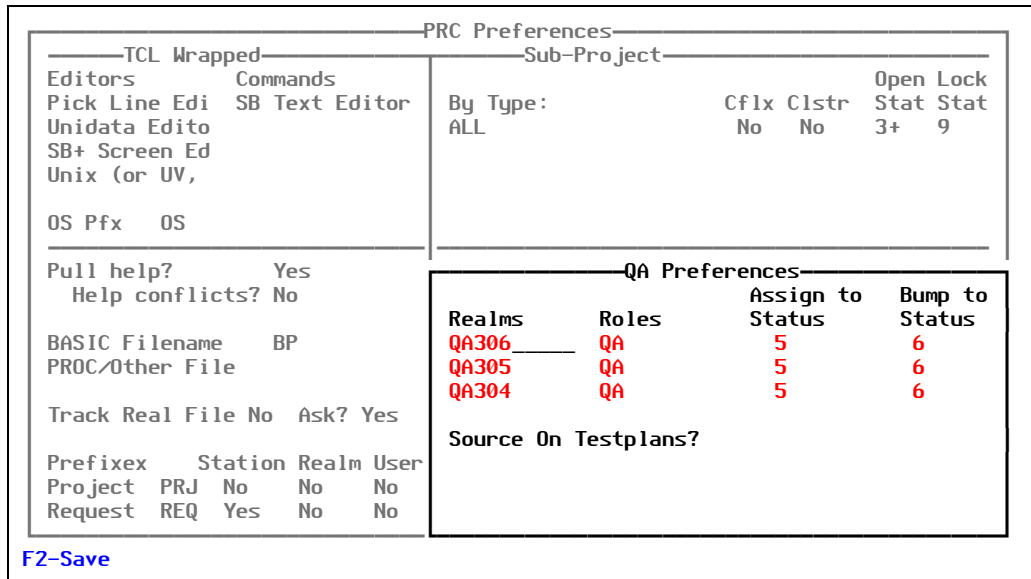


Figure 7-13: QA Preferences subscreen

Realms The realms in which quality assurance (QA) will be performed.

Press **F3** to select from the list of valid entries.

Roles The role code that represents the QA activity in this realm. For more information, see [“Roles Codes” on page 36](#).

Press **F3** to select from the list of valid entries.

Assign to Status The status code that represents the state in which a project must be when it is run through QA in this realm. For more information, see [“Status Codes” on page 29](#).

Press **F3** to select from the list of valid entries.

Bump to Status The status code that represents the state to which project will be bumped when QA is completed in this realm.

Press **F3** to select from the list of valid entries.

Source On Testplans? Yes turns on the feature where source items are stored against tests. Valid entries are:

- **Y** — Yes
- **N** — No

For additional information about this feature, see the *Software Quality & Testing Guide*.

Press **F3** to select from the list of valid entries.

When you have finished with the settings, press **F2** (Save). The system returns you to the More Preferences screen (Figure 7-6 on page 166).

APIs

To set up your API definitions from the More Preferences screen (Figure 7-6 on page 166), select **APIs**.

The system displays the API Definitions subscreen, similar to the one shown in Figure 7-14:

```

PRC Preferences
-----
TCL Wrapped
Editors      Commands
Pick Line Editor  SB Text Editor (
SB+ Screen Editor Delete
Unix Editor
Ultra Edit

Sub-Project
By Type:      Cflx Clstr Stat Stat
ALL           No  No  3+  Z

API Definitions
Desktop

Telnet App      SB Client

Problem Rptng
Change Control
Testing Tool

Downld Methd    SBClient
Directory       C:\DUMPS

OS Pfx  OS
-----
BASIC Filename  PRC.INCLUD
PROC/Other File BP.UTIL

Track Real File Yes Ask? No

Prefixes  Station Realm User
Project  PRJ  No  No  No
Request  REQ  Yes No  No

F2-Save F4-Del

```

Figure 7-14: API Definitions subscreen

Telnet App

The telnet application that interfaces with PRC. Valid entries are:

- **SBC** — SBClient
- **W** — Wintegrate
- **A** — Accuterm

Press **F3** to select from the list of valid entries.

Problem Rptng

The name of the desktop problem reporting tool.

Change Control

The name of the desktop change control tool.

Testing Tool

The name of the desktop testing tool that runs scripts.

Downld Methd The type of automatic download to optionally launch during a ravel process. Valid entries are:

- **SBC** — SBClient
- **W** — Wintegrate
- **FTP** — File Transfer Protocol (FTP)

Press **F3** to select from the list of valid entries.

Directory The directory (on the local desktop) in which ravel should be dumped.

When you have finished with the settings, press **F2** (Save). The system returns you to the More Preferences screen (Figure 7-6 on page 166).

File Options

To set up your options concerning files from the More Preferences screen (Figure 7-6 on page 166), select **File Options**.

The system displays the File Options sub-screen, similar to the one shown in Figure 7-15:

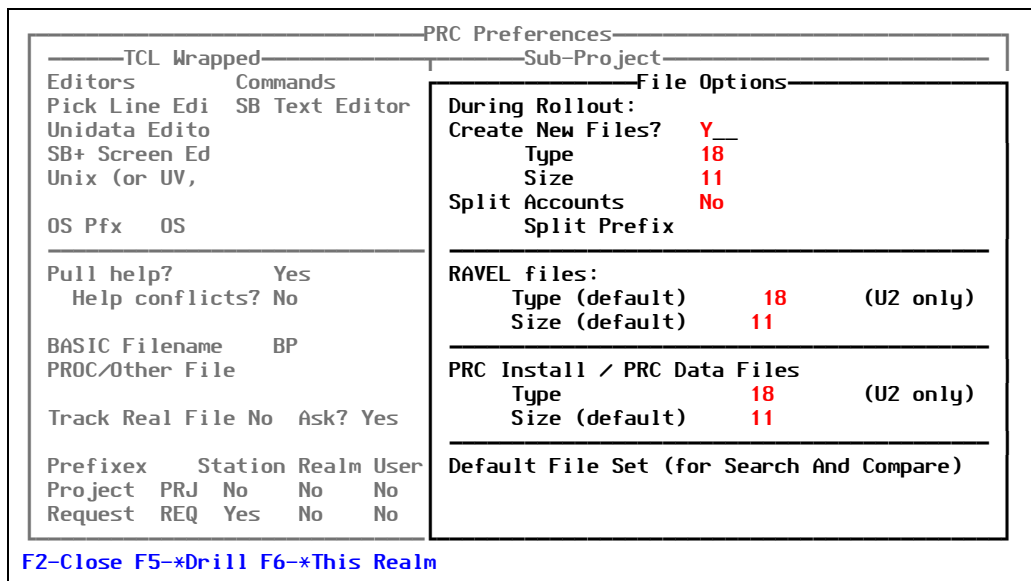


Figure 7-15: File Options sub-screen

The following parameters are referenced during a rollout:

Create New Files? Indicates whether PRC will automatically create files for you during a rollout. Valid entries are:

- **Y** — Yes
- **N** — No
- **X** — Ask
- **C** — Custom

The Custom setting means that there is a program on the “process slots” that is supposed to create files. When PRC reaches a spot where it needs a new file, this setting causes PRC to find that program and run it, rather than using the standard UniData/UniVerse `CREATE . FILE` command.

Press **F3** to select from the list of valid entries.

Type The default or most common file type to use when a file is created.

Size The default file size to use when hash-type files are created.

Split Accounts Indicates whether the dictionary section goes in the base realm location and the data sections of the created file go in each associated data account. Valid entries are:

- **Y** — Yes
- **N** — No

Press **F3** to select from the list of valid entries.

Split Prefix The code that becomes the split prefix if software is structured in a “split account.”

SB+ allows you to create a split account environment. Such an environment means all software, including those items that reside in dictionaries, are in one account.

When SB+ creates this split for you, it requests a prefix that it will use for all pointers to that system. If you are developing in a split account environment, defining the split prefix in this field ensures that PRC always references the file name without the split prefix, so that items are not tracked two different ways.

The following parameters pertain to ravel files:

Type (default) The default file type, according to platform.

Size (default) The default size for a ravel file.

The following parameters pertain to PRC installation and data files:

Type The file type to be used when PRC automatically creates tracking files.

Size (default)	The default data file size, expressed as a modulo. Use a prime number.
Default File Set (for Search And Compare)	The file set or sets that should be used by default for searches and comparisons.

When you have finished with the settings, press **F2** (Save). The system returns you to the More Preferences screen (Figure 7-6 on page 166).

Process Slots

To define processes that should be run around certain events from the More Preferences screen (Figure 7-6 on page 166), select **Process Slots**.

The system displays the Processes in Slots subscreen, similar to the one shown in Figure 7-16:

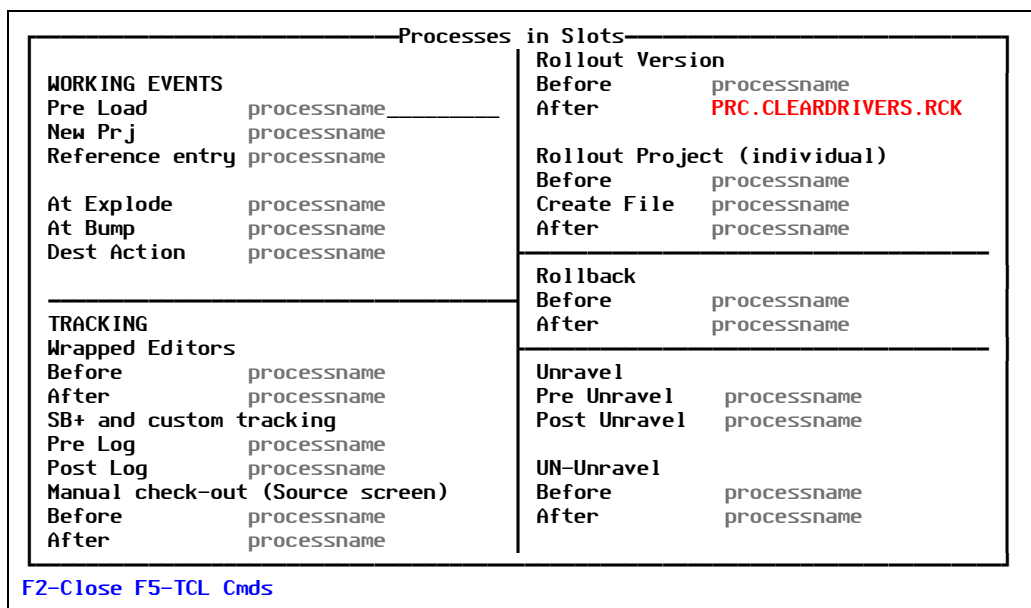


Figure 7-16: Processes in Slots subscreen

The following processes are associated with working events:

Pre Load	Run the specified process before loading PRC. The process can be used to <i>not</i> to load PRC, if desired under certain circumstances.
New Prj	Run the specified process when a new project is created from the /PRC command in the working or development account.

Sends back the key if you built the KEY, and sets RTN.FLAG to **X** to abort.

Reference entry

Reference is a field on the project that is often a help desk or Review Source Item (**/SCR**) number.

See PRCPROGS PRC.REFERENCE.V for a coding sample on validating the reference field. Placing the call here to PRC.REFERENCE.V without customizing will make the field mandatory but will not otherwise validate what is entered.

At Explode

Use this one to “pull” things on to a project.

At Bump

The name of your custom program that should run during a “bump” and that may cause the bump to fail. To fail the bump, set RTN.FLAG to **1**.

Dest Action

PRC will run the process you specify before the standard “destination action.” Typically, the specified process compiles programs in a target account after a rollout.

The following processes are associated with tracking:

Wrapped Editors:**Before**

This call will have two arguments.

After

This call will have two arguments.

SB+ and custom tracking:**Pre Log**

Enter the Pre Log.

Post Log

Enter the Post Log.

Manual check-out (Source screen):**Before**

PRC will run the process you specify *before* allowing an item on the Source screen to be checked out manually. To disallow the check-out, set RTN.FLAG to **1**.

After

PRC will run the process you specify *after* an item is checked out manually on the Source screen.

The following processes are associated with rollouts:

Rollout Version:**Before**

PRC will run the process you specify *before* a version rollout has completed.

After

PRC will run the process you specify *after* a version rollout.

Rollout Project (individual):

Before	PRC will run the process you specify <i>before</i> a rollout. PRC runs the process as a project is prepared for rollout for any customized actions that must be taken at that time.
Create File	Enter the Create File.
After	Enter the After.

The following processes are associated with rollbacks:

Rollback:

Before	Enter the Before.
After	Enter the After.

The following processes are associated with unravels:

Unravel:

Pre Unravel	The process you specify is run at a site, before the unravel program.
Post Unravel	PRC will run the process you specify <i>after</i> the unravel programs have finished on a site.

UN-Unravel:

Before	Enter the Before.
After	Enter the After.

When you have finished with the settings, press **F2** (Save). The system returns you to the More Preferences screen ([Figure 7-6 on page 166](#)).

If you press **F5** (TCL Cmds), the system displays the TCL Commands subscreen, similar to the one shown in [Figure 7-17](#):

Processes in Slots	
WORKING EVENTS	
Pre Load	processname
New Prj	processname
Reference entry	processname
At Explode	processname
At Bump	processname
Dest Action	processname
<hr/>	
TRACKING	
Wrapped Editors	
Before	processname
After	processname
SB+ and custom tracking	
Pre Log	processname
Post Log	processname
Manual check-out (Source screen)	
Before	processname
After	processname
<hr/>	
TCL Commands	
Rollout Version	
Before	processname
After	processname
Rollout Project (individual)	
Before	processname
Pre Copy	procname
Post Copy	procname
Pre Delete	procname
Post Delete	procname
Pre Create File	procname
Post Create	procname
Pre Delete File	procname
Post Delete	procname
Pre Clear File	procname
Post Clear File	procname_____

F2-Close

Figure 7-17: TCL Commands subscreen

The settings are self-explanatory.

When you have finished with the settings, press **F2** (Save). The system returns you to the Processes in Slots subscreen ([Figure 7-16 on page 184](#)).

Audits

To set up your environment preferences from the More Preferences screen ([Figure 7-6 on page 166](#)), select **Audits**.

The system displays the More Preferences subscreen, similar to the one shown in [Figure 7-18](#):

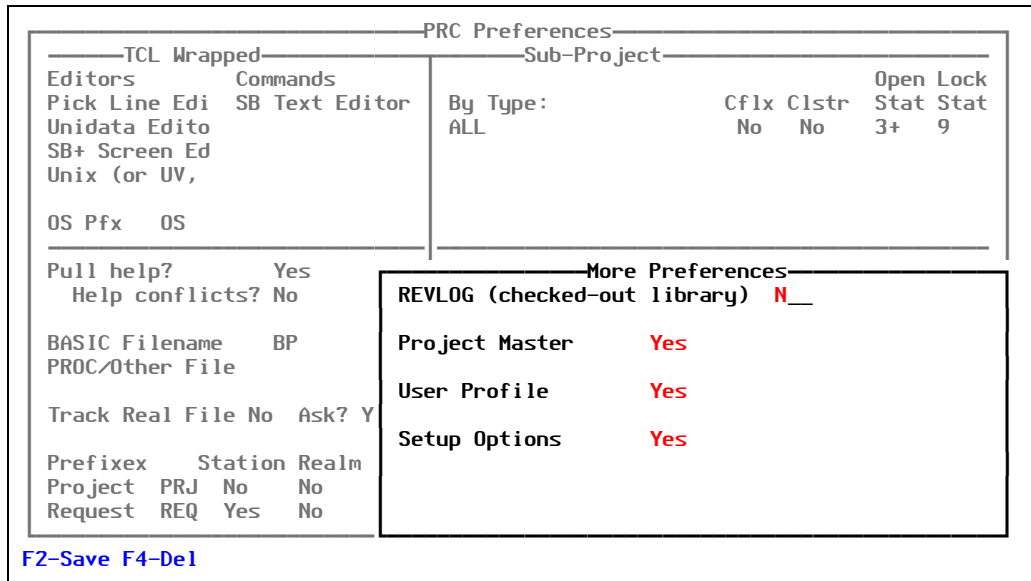


Figure 7-18: More Preferences sub-screen

REVLOG (checked-out library)

Indicates whether PRC will audit programmatic writes and deletes to the “check-out repository” revision log (REVLOG). Valid entries are:

- **Y** — Yes
- **N** — No

Press **F3** to select from the list of valid entries.

Project Master

Indicates whether PRC will turn auditing on for all changes to Project Master information. Valid entries are:

- **Y** — Yes
- **N** — No

Press **F3** to select from the list of valid entries.

User Profile

Indicates whether PRC will audit all change to user profiles. Valid entries are:

- **Y** — Yes
- **N** — No

Press **F3** to select from the list of valid entries.

Setup Options

Indicates whether PRC will audit all changes to your PRC preferences and/or setups (realms, and so on). Valid entries are:

- **Y** — Yes
- **N** — No

Press **F3** to select from the list of valid entries.

When you have finished with the settings, press **F2** (Save). The system returns you to the More Preferences screen (Figure 7-6 on page 166).

Infrequently Used Options

To set up your environment preferences from the More Preferences screen (Figure 7-6 on page 166), select **Infrequently Used**.

The system displays the Infrequently Used Opts subscreen, similar to the one shown in Figure 7-7:

TCL Wrapped					PRC Preferences				
Editors					Sub-Project				
Pick Line Edi	SB	Text	Editor		By Type:	Cflx	Clstr	Open	Lock
Unidata Edito					ALL	No	No	3+	9
SB+ Screen Ed									
Unix (or UV,									
OS Pfx	OS								
Pull help?	Yes				Archive Dir	C:\ROCK\PRC_ARCHIVES			
Help conflicts?	No				Infrequently used opts				
BASIC Filename	BP				User Id From	SB_			
PROC/Other File					Case For User Id	A			
Track Real File	No	Ask?	Yes		Multiple Users	Yes			
Prefix	Station	Realm	User		Custom Suffix				
Project	PRJ	No	No	No	File/Item	I			
Request	REQ	Yes	No	No					

F2-Save F4-Del

Figure 7-19: Infrequently Used Opts subscreen

User ID From

Determines which user ID to use as the PRC user ID for infrequent options. Valid entries are:

- **SB** — SystemBuilder
The “SB” login itself for SB+.
- **O** — Operating systems
The user login associated with the operating system.

Press **F3** to select from the list of valid entries.

- Case For User ID** Determines the kind of character conversion that should take place on the user ID in order to log in. Valid entries are:
- **U** — Convert all user ID characters to upper-case
 - **L** — Convert all user ID characters to lower-case
 - **E** — Do not convert any characters; leave them as entered
- Press **F3** to select from the list of valid entries.
- Multiple Users** Indicates whether PRC will track multiple users against a project. Valid entries are:
- **Y** — Yes
 - **N** — No
- Press **F3** to select from the list of valid entries.
- Custom Suffix** A literal string or an expression that becomes a suffix to the main program files if changes are to be copied into a custom or workspace area. Works in conjunction with the *File/Item* field.
- For example, you type **@USER.ID** in this field. Whenever a program in *XXPROGS* is edited, PRC will copy it into the user's individual *XXPROGS.USERID* file. Likewise, if the value of the *File/Item* field is **I** (Item), the user ID suffix is added onto the item.
- If you make customizations to purchased software, use these fields to automatically rename files or items. Be aware, however, that this approach leads to some difficulties and may not be necessary with PRC in place.
- From the software reseller's point of view, the situation of custom changes for different clients could be handled with "custom realms."
- File/Item** Works in conjunction with the *Custom Suffix* field. When a custom suffix is added, to what object is the suffix attached? Valid entries are:
- **F** — File
 - **I** — Item
- Press **F3** to select from the list of valid entries.
- Display Dest Action?** Indicates whether PRC will display the destination action.
- Destination actions are tasks that PRC has to do on the destination account to finish the delivery; for example, cataloging programs. This "display it or not" field lets you set up whether PRC does it silently or whether it presents the compile messages all over the screen when PRC is performing the task.

Valid entries are:

- **Y** — Yes
- **N** — No

Press **F3** to select from the list of valid entries.

When you have finished with the settings, press **F2** (Save). The system returns you to the More Preferences screen (Figure 7-6 on page 166).

Standards

Certain programming standards can be enforced by PRC. Currently, the standards include key lengths, certain VOC entries that should not be rolled out, and certain strings that should not be included in an item that is being delivered (the most common—almost universal—string is DEBUG).

To define system-wide standards, perform the following:

1. Select **Setup > Standards**.

The system displays the Standards screen, similar to the one shown in Figure 7-20:

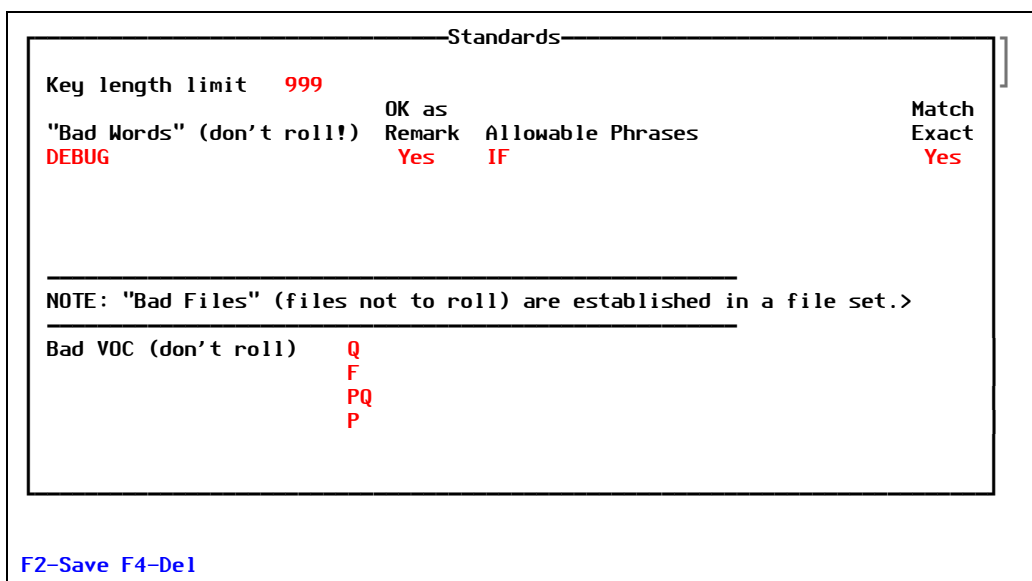


Figure 7-20: Standards screen

Explanations of the fields are as follows:

Key Length Limit	The maximum length that any key to any field can be. Limited by the operating system.
Bad Words	Search items for strings, disallow rollout if strings are found; for example, DEBUG.
Remark	Indicates whether this string is allowed in a program as a remark line. Valid entries are Y (Yes) or N (No).
Allowable Phrases	Another string that this word <i>is</i> allowed if this other string is present. Indicates they are allowed as a phrase, not single words.
Exact	Indicates whether the word must be found as an <i>exact discrete match</i> in order to be classified as “bad.” Valid entries are Y (Yes) or N (No). For example, if you enter Y (Yes) in this field and DEBUG is classified as a bad word, DEBUGMEBABY would pass and DEBUG would fail.
Bad VOC	The type of VOC items that may not roll out. Prevention is based on the value of attribute 1 .

-
2. Fill in each field with the appropriate information.
3. Press **F2** (Save).

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

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 **REXHIST** 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.

	<p>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.</p>
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	<p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p>
/KNIT	<p>The SB+ command that displays the Knit split-screen utility.</p> <p>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 <i>Software Knitting Guide</i>.</p>
/PRC	<p>The SB+ command that displays the Assign/Switch Projects screen.</p>
project	<p>The standard, central unit of work in PRC.</p> <p>Sometimes known as a task or a software change request (SCR) in other software configuration management applications.</p>
project-centric	<p>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.</p> <p>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.</p>
/QUICK	<p>The SB+ command that displays the Quick Project Update screen.</p>
ravel file	<p>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.</p>

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	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.
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	The automatic feature that can be set to track time against projects.
tool file	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.
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	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.
unravel	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.
versions	In the context of PRC, a collection of projects that will be delivered together.

Symbols

.mak ix
/BUMP 193
/KNIT 196
/PRC 196
/QUICK 196
/SRC 197
/STACK 198

A

account
 tracking 198
Accuterm xii
activity
 code 38
 defined 38
archives ix
archiving
 deltas x
 methodologies explained x
assign 193
audit changes 193

B

branch
 copy 194
bump
 defined 193

C

change
 tracking viii
check-in 194
check-out x, 194
 mode x
class
 project 21
clearance
 levels x
compliance ix
 reporting viii
components
 checking out x
conflict
 defined 194
 scan 195
customer request ix

D

deltas x
DoD ix
doom 195
drill down 195

E

environment
 database development x
 mainframe x
 realm ix
 U2/Multivalue ix
explode 195
extension
 .mak ix

F

feature
 source control ix
field
 navigating a xiv
 special keys for xiv
fields
 Destination Realm 160
 Installing on 160
 Suite 160
 System 160
 Version 160
file
 item ix
flick
 defined 195
function keys
 common xiii

I

IEEE ix
ISO 9003 ix
item
 defined ix
 tracking changes ix

J

junk project 196

K

key

- adding lines xiv
- Alt+Tab xiv
- backslash xiv
- down-arrow xiv
- Escape xiv
- F2 xiv
- slash xiv, 197
- Tab xiv
- up-arrow xiv
- knowledge
 - required vii
- M**
- MAK file ix
- menus
 - Tracking Options xii
- O**
- object
 - item ix
- P**
- peripheral management viii
- PRC
 - 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
- PRC Preferences screen x
- Preferences screen x
- project
 - changes viii
 - child viii
 - collection of ix
 - defined viii, 196
 - parent viii
 - sub-project defined viii
 - tracking viii
- project-based ix
- project-centric ix, 196
- R**
- ravel
 - file 196
- realm
 - defined ix, 197
 - DEV ix
 - LIVE ix
 - remote 197
 - TEST ix
- region
 - realm ix
- request
 - customer ix
 - defined viii
- requests 197
- rollback
 - mechanism viii
- rollout
 - defined 197
- rule
 - realm ix
 - set of ix
- S**
- sandbox x
- SBClient xii
- SCM 197
- screen
 - navigating a xiv
 - special keys on xiv
- software change request (SCR) ix
- software configuration management 197
- 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
- sub-project ix
 - changes viii
 - defined viii
- sub-version

defined ix
super-project ix
system ID 160

T

task ix
telnet xii
This Item 198
timer 198
tool file 198
trace
 defined 198
tracking
 account 198
type 25

U

unit

of work viii
unravel
 defined 198
user profile
 check-out mode x

V

Version 62, 68
version ix
 archived ix
 defined ix, 198
 master ix
Visual Basic ix

W

Wintegrate xii

