CASEVision / Tracker User's Guide

Document Number 007-1663-020

CONTRIBUTORS

Written by Margaret-Anne Halse and John C. Stearns
Production by Gloria Ackley
Engineering contributions by Pete Orelup, Jack Repenning, John Templeton, and Stuart Liroff

CASEVision / Tracker User’s Guide
Document Number 007-1663-020

Contents

Figures

Tables

Introduction

This guide describes Request Tracking System (RTS), a system for tracking bugs and requests for enhancements. RTS was designed using the CASEVision™/Tracker tool from Silicon Graphics, Inc.®, a tool for creating and modifying request tracking systems.

RTS is designed to

- meet the basic request tracking needs of most software organizations with only minor modification
- serve as a functioning starter example of a Tracker-based system for those organizations that want to create their own systems

RTS consists of four applications: *rtsquery*, the main application, which provides full functionality, and three supplementary applications for special purposes: *rtssubmit*, *rtsrespond*, and *rtsapprove*.

As you read this guide, please bear in mind that if your organization has modified the standard version of RTS, you may find minor differences from the one described here. The basic concepts and application structure should be the same.

Chapter 1

The Request Tracking System (RTS): An Overview

This chapter provides an overview of the Request Tracking System (RTS). RTS provides a special graphical user interface (GUI) for accessing and submitting bugs and requests for enhancements, referred to collectively as *requests*.

RTS consists of four applications: *rtsquery*, the main application, which provides full functionality, and three supplementary applications for special purposes: *rtssubmit*, *rtsrespond*, and *rtsapprove*.

These topics are covered:

- *rtsquery* user interface
- RTS tracking process

rtsquery User Interface

To understand *rtsquery*, it's helpful to know a few of the terms that are used in Tracker systems. The term *request* refers to the type of data you are tracking in the system, typically bug reports or requests for enhancement (RFE). A request contains all the information associated with the request, such as submitter, owner, priorities, and dates.

The program that you run to access a request is an *application*. An application provides one or more *views*, which are windows for inspecting or changing requests. All applications have a *main view* that appears when you invoke the application. The main view is usually a general purpose window that contains most of the application's functionality. An application also has *auxiliary views*, which you can use for special data or for users requiring limited information or functionality. In addition to the main application, *supplementary applications* are available. Supplementary applications are similar to auxiliary views (they may even access the same data), but supplementary application users do not use the main view.

The standard version of *rtsquery* has one main view *rtsquery*, one auxiliary view *rtsfiles*, and these three supplementary applications:

- *rtssubmit*, used to submit requests
- *rtsrespond*, used by request owners and screeners to respond to requests
- *rtsapprove*, used by request approvers to approve requests

The *rtsquery* Main View

The RTS application main view windows all have the same parts, although their menus and request form areas may differ. The main application for RTS is *rtsquery*, shown in Figure 1-1 with its main sections labeled.

The *rtsquery* window has four main areas:

menu bar — lets you access the window menus

control bar — lets you display a list of requests and perform operations on individual requests

query results area — displays lists of requests that result from queries

request form area — lets you enter or display detailed information about a single request

Figure 1-1 Tracker Window with Major Areas

Menu Bar

The menu bar lets you access five menus (see Figure 1-2).

Figure 1-2 *rtsquery* Window with Menus

The menus are:

File menu — lets you print a file and exit the window

Edit menu — provides editing commands that you can apply to all fields as a group

Query menu — lets you reuse query selection criteria.
"Save..." stores the current set of criteria.
"Load..." enters a saved set of criteria into the appropriate fields.
"Save as Default" saves your current criteria and reuses it as a default when you start up this RTS application.
"Case Sensitive Matches" toggles case sensitivity on or off when you are searching for a string match in a field.

Views menu — lets you display the auxiliary views associated with *rtsquery*. The standard version of *rtsquery* has one auxiliary view called *rtsfiles*, shown in Figure 1-3. The *rtsfiles* window lets you enter or access problem files in the *Found in* list, fixed

files in the *Resolved in* list, and final files in the *Fixed Releases* list.
Your system may have additional views available.

Help menu lets you get version information, summary information for the window, context information for components of the window, and an index of help topics for the system. The Help menu is standard for CASEVision tools.

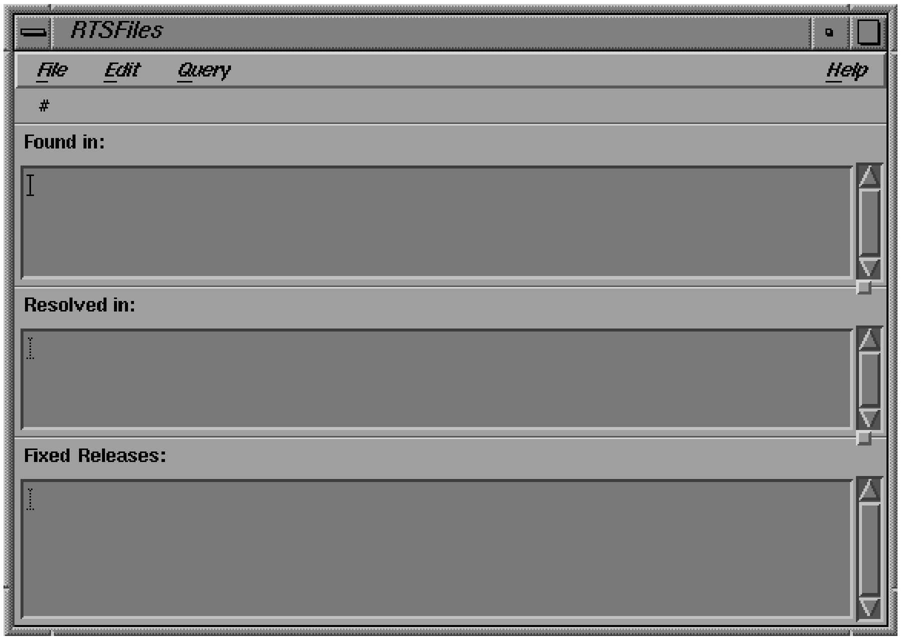


Figure 1-3 *tsfiles* Window

Control Bar

The control bar area of the *rtsquery* window is shown in Figure 1-4, with the *Modes* menu displayed.

The Modes menu displays the commands that you can perform on requests in the database. These commands are enabled or disabled, depending on your currently selected request (if any) and its state. For example, if no information is displayed, the system assumes you are either going to inspect requests or submit new ones; therefore, only the top four menu items are enabled.

When you select a command from the Modes menu, the information displayed in the rest of the window changes, as does the read / write permission on the fields. You can then cancel or apply the currently selected command by using the *Cancel* or <*Apply command*> buttons to the right of the Modes menu. In Figure 1-4, the current command is *Query*.

Commands have two categories: *inspection commands* that review requests and *transition commands* that can change the state of a request. The inspection commands, “Query” and “Display,” let you look up requests, in a list of summaries or individually in detail. The *transition commands* change either the status of the request or the fields in the report. Any field required for a specific transition is highlighted, indicating that an entry is necessary. If no entry is made, then the command is blocked from completion.

The list control buttons *First*, *Next*, *Previous*, and *Last* let you choose a request from the list for display in the request form area.

Figure 1-4 Control Bar Area of *rtsquery* Window with Modes Menu

Query Results Area

The *rtsquery* query results area is shown in Figure 1-5. The number of requests in the current query is displayed at the top of the list area. The list itself displays information that identifies requests in the list. If you use any of the list controls (buttons or menu items) to display a request in the form area, the list will scroll to the displayed request accordingly.

Two small icons that appear in the left margin of the query results area also help to track the status of a request:

- The small rectangle indicates that the request has been displayed in the form area at least once since its submission.
- The check mark indicates that the request has been edited at least once.

The query results area is divided into five columns, each with a title. The text in each column lines up under the title. Text fields are left-justified, and numerical fields are right-justified.

Figure 1-5 Query Results Area

Next to each title is a radio button. When you click on one of these buttons, the button is highlighted and the list of requests will be sorted accordingly. For example, if you click on the ID# button, as in Figure 1-5, the list is sorted in ascending order according to ID#.

Note: You can sort on only one characteristic at a time, in ascending order, from the GUI.

Request Form Area

The request form area in RTS applications contains fields for entering and displaying data. A typical request form area appears in Figure 1-6.

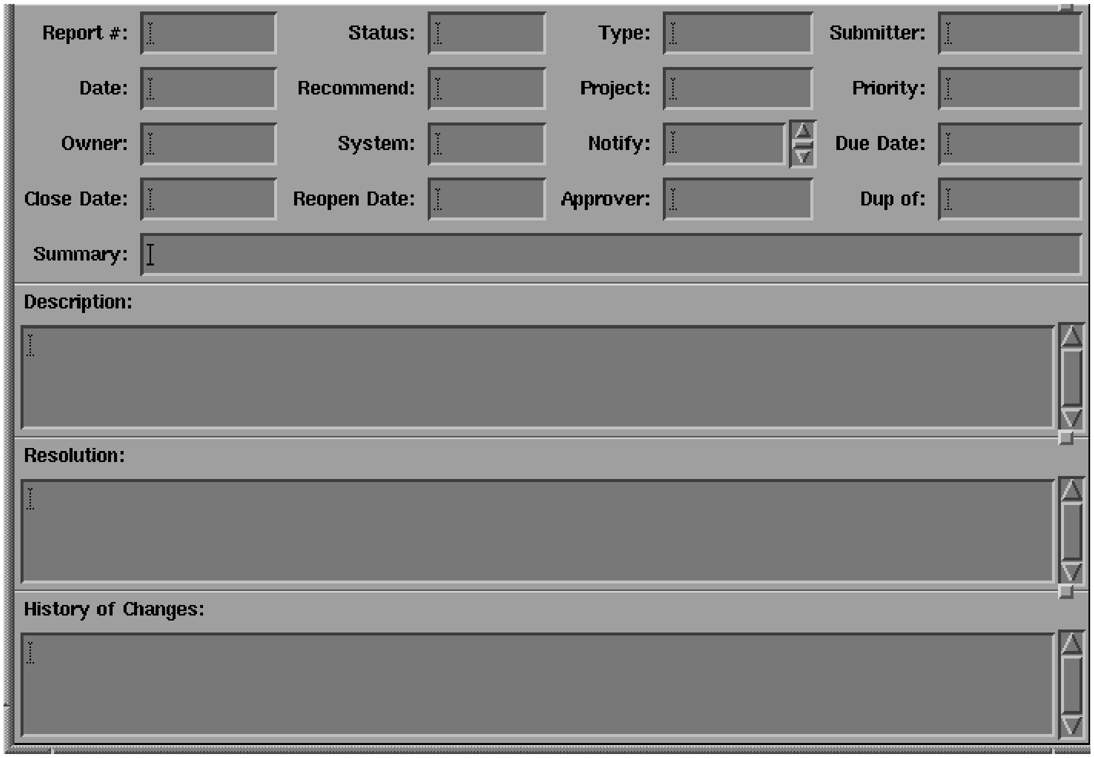


Figure 1-6 Request Form Area

Fields change colors and take on highlighting according to your current operation. If a field is the same color as the window background color, then it is in read-only state; when the field is in a contrasting color, it is writable.

Fields are highlighted if they are required for the current transition command. Fields are also highlighted if they are incorrectly entered. The highlighting color depends on your color scheme. In either case, a highlighted field blocks the transition.

To display a popup menu, hold down the right mouse button while the cursor is over a field. At minimum, the menu contains the items “Reuse” (for using the last value displayed in this field), “Revert” (for returning to the saved value), and “Clear” (for blanking out the field). If predefined values are available for the field, they are displayed on the cascading menu attached to the “Values” item. The cascading menu features a “tear-off” perforation at the top. If you click on the perforation, the menu is displayed in its own window.

If the last item in the Values menu is an ellipsis (. . .), then you can enter a value other than the ones displayed. Figure 1-7 shows the values for the *Status* field.

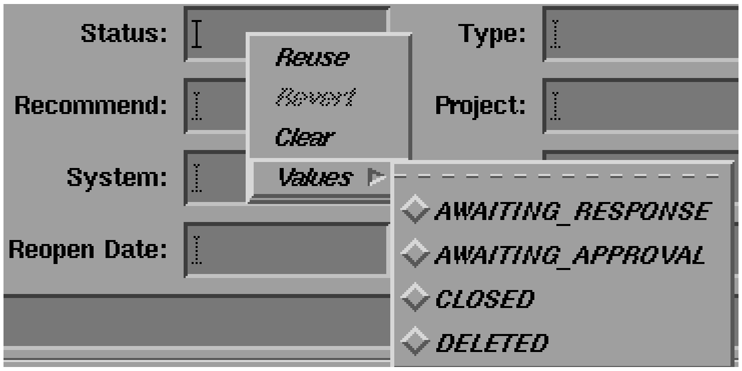
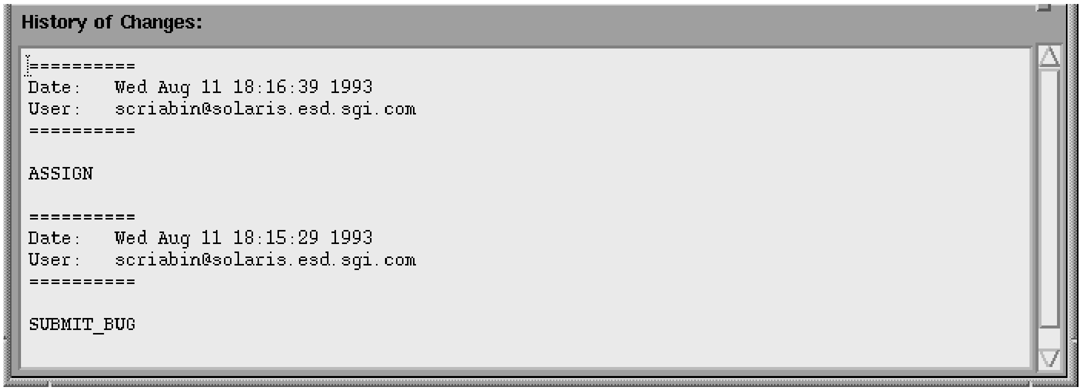


Figure 1-7 Field Menu

The menu for fields that accept long text descriptions such as the *Description* and *Resolution* fields have an item called “Edit...” that lets you enter the text through your editor of choice. The editor default depends on the environment variable setting (see Chapter 3, “RTS Application Reference”) and defaults to *vi* if no variables are set.

The *History of Changes* field is automatically updated by Tracker each time an action is taken on a request. Figure 1-8 shows part of a typical request history.



```
==========
Date:   Wed Aug 11 18:16:39 1993
User:   scriabin@solaris.esd.sgi.com
==========

ASSIGN

==========
Date:   Wed Aug 11 18:15:29 1993
User:   scriabin@solaris.esd.sgi.com
==========

SUBMIT_BUG
```

Figure 1-8 History of a Request

rtsquery Tracking Process

There are probably as many ways of tracking requests as there are software organizations. The CASEVision / Tracker software has been designed with adaptability in mind. The tracking process implemented in the *rtsquery* application represents one particular process; your organization can use this process as is or with slight modifications. This section describes the *rtsquery* process in terms of the life cycle of a request in the system.

The data flow diagram in Figure 1-9 illustrates the request tracking process.

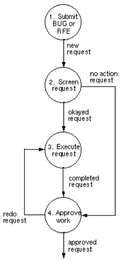


Figure 1-9 Data Flow Diagram of *rtsquery* Tracking Process

A request can go through many different states during its life cycle. A *transition* is an operation performed through a view that changes the request either in terms of status or data. Transitions are accessed from the Modes menu in the view's control bar.

Figure 1-10 shows how the transition commands change the states of a request. (The ovals represent the states of a request; the rectangles are the transition commands used to change the indicated states.)

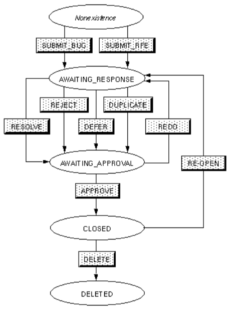


Figure 1-10 State Transitions of a Request

The request tracking process has four major activities:

1. A submitter submits a bug or request for enhancement (RFE) into the request database. At this point the request is in the AWAITING_RESPONSE state.
2. The request is screened before the owner starts to work on it. *rtsquery* permits screening by these three parties:

   - screening committee
   - project manager
   - owner

   Your tracking system administrator can tell you how screening is implemented for *rtsquery* in your organization. If okayed for work, the request is passed on to the owner. If no action is required, that is, it is rejected, deferred, or redundant (DUPLICATE), the request is returned to the database in the AWAITING_APPROVAL state for action by the approving authority.
3. Approved requests are then executed by the owner. If the approving authority calls for a redo, an owner will need to rework old requests. When the owner completes work on a screened request, it enters the AWAITING_APPROVAL state prior to approval.

4. The approving authority must then determine if the request has been adequately satisfied. If the approving authority finds problems, then the request is returned to the owner for correction (REDO); otherwise, the request is closed. If problems arise later, the request can be reopened and returned to the owner.

Chapter 2

Using the Request Tracking System

This chapter takes a task-oriented approach to describing the *rtsquery* request tracking system. The chapter covers the following tasks:

- Getting started
- Submitting a request
- Browsing and displaying requests
- Screening requests
- Executing and reopening requests
- Approving requests
- Printing results

Getting Started

This section shows you how to use RTS applications. The first subsection discusses the differences between the applications. The next subsection presents a general discussion of performing Tracker operations.

RTS Applications

To run RTS applications, type the full pathname of the application and press **`<return>`**. The RTS applications are usually stored in */usr/local/bin*. The standard applications are:

- *rtsquery*, the main application (with full functionality)
- *rtssubmit*, used to submit requests
- *rtsrespond*, used by request owners and screeners to respond to requests
- *rtsapprove*, used by request approvers to approve requests

rtsquery differs from the other applications in that it displays all of the fields in its main view and auxiliary view (see Figure 2-1). All operations can be performed from *rtsquery*, and all information in the database can be viewed from *rtsquery*. The other applications have fewer operations and only limited information for queries. They offer convenience as a trade-off.

Figure 2-1 *rtsquery* Main View with Modes Menu and *rtsfiles*

Procedure for Performing Tracker Operations

For most Tracker operations you'll follow these steps:

1. Type the application name to display the desired window (if the application has not yet been started).

   The window is displayed.

2. Select the desired command from the Modes menu in the control bar.

The window enters the selected mode for performing the command. Note that the command is not executed until you click the <*apply command*> button, which is to the right of the *Cancel* button in the control bar. Depending on the command selected, the window makes these changes:

- Fields become writable or read-only.
- The command name now displays on the Modes menu button and on the <*apply command*> button.
- All fields required for the command are highlighted in red. This includes fields in views (such as *rtsfiles*) that may be hidden from view.

3. Fill in the request form fields with the appropriate information. Remember that holding down the right mouse button displays a menu with these items:

   - "Values" lets you select any valid predefined value.
   - "Clear" clears the field.
   - "Reuse" inserts the value from the previous request (if any).
   - "Revert" restores the original value of the field for that request.

   As the required fields are entered, the highlight disappears. If a field is entered incorrectly, according to the rules for that field, it is highlighted in yellow and must be corrected for the transition to take place. If you don't know the field's requirements, check on-line help.

4. Click the <*apply command*> button to perform the operation.

   Note that the <*apply command*> button is sensitive only when all required fields are filled in. Clicking the button performs the operation and changes the window and database (if affected) accordingly. Any default fields are filled in automatically. The command mode may change as well; for example, after a query, the window switches to display mode.

   In addition, email notifications are made to the owner, project manager, submitter, and any other parties indicated in the *Notify* field. The general format for a “SUBMIT_BUG” notification is:

```
Subject: BUG #ID modified, Transition SUBMIT_BUG

The following report fields have been modified:

Field czar set to: trackeradministrator.
Field owner set to: ownername.
Field ENTITY_ID set to: ID.
Field type set to: BUG.
Field STATE set to: AWAITING_RESPONSE.
Field summary set to: problemsummary.
Field bboard set to: approvingauthority.
Field due_date set to: submitdate plus 30 days.
Field submit_date set to: submitdate.
Field submitter set to: submittername.
Field notify_list changed from: () to: (interestedpartylist,
ownername, submittername).

New value of description:
problemdescription
```

Submitting a Request

You can enter a new request through either *rtsquery* or *rtssubmit*. Both use the transition commands "SUBMIT_BUG" and "SUBMIT_RFE" to submit bug reports and requests for enhancement (RFEs) respectively.

Differences Between *rtsquery* and *rtssubmit*

The major difference between the applications is that *rtsquery* lets you perform the full range of transitions; it has a separate window for entering files. *rtssubmit* is a special purpose application best suited to submitting requests. It has fewer commands in its Modes menu (see Figure 2-2) and only those fields necessary for submitting requests (see Figure 2-3).

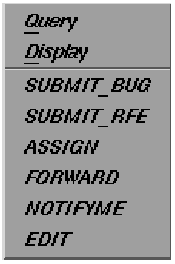


Figure 2-2 *rtssubmit* Modes Menu

rtssubmit conveniently lets you enter problem files in its main window. Since *rtssubmit* has a limited number of fields, it is less suited to displaying details of individual requests.

Request Submission Procedure

To submit a request:

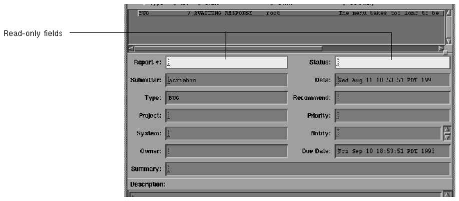
1. From either *rtsquery* or *rtssubmit*, select one of the submit transitions ("SUBMIT_BUG" or "SUBMIT_RFE") in the Modes menu on the control bar.

   The window makes these changes (see Figure 2-3):

- The *Report #* and *Status* fields turn read-only, since these are set automatically by the system.
- The *Description* field is highlighted in red, indicating that it is required.
- The <*apply command*> field changes to the selected submit transition name, although it is not yet enabled.

2. Fill in the *Description* field, which is required, and any other fields needed to provide request information.

   After you make an entry in the *Description* field and move the cursor out of it, the highlighting is removed. At this point, the <*apply command*> and *Cancel* buttons are enabled, since you have fulfilled the *Description* requirement.



BUG 7 AWAITING RESPONSE root The menu takes too long to be

Report #:
Status:
Submitter: scriabin
Date: Wed Aug 11 10:53:51 PDT 199
Type: BUG
Recommend:
Project:
Priority:
System:
Notify:
Owner:
Due Date: Fri Sep 10 18:53:51 PDT 1993
Summary:
Description:

Figure 2-3 Changes to Window after “SUBMIT_BUG” Mode Selected

These two facilities will help you enter data:

- Use the Edit menu to make global edits in the window.
- Hold down the right mouse button and select “Values” to select a standard value. Select “Reuse” to enter the value from the last request.

As a further convenience, these are the default entries:

- *Type*—BUG or RFE
- *Submitter*—current user
- *Date*—current date
- *Owner*—designated request czar (tracking system administrator)
- *Due date*—30 days from submit date
- *Summary*—first line of description

3. To complete the submit transition, click the <*apply command*> field now labeled either “SUBMIT_BUG” or “SUBMIT_RFE.”

   After you click the <*apply command*> button, the following takes place:

   - The default entries are made, unless you have entered different information.
   - The new request is registered in the database.
   - The request enters the AWAITING_RESPONSE state.
   - The window displays all the request information and switches to display mode.
   - Email notifications are made to the owner, submitter, and anyone listed in the *Notif*y field.

   The data flow for the submittal process is summarized in Figure 2-4.

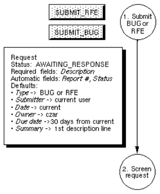

SUBMIT_RFE
SUBMIT_BUG
1. Submit BUG or RFE
Request
Status: AWAITING_RESPONSE
Required fields: *Description*
Automatic fields: *Report #, Status*
Defaults:
• *Type* -> BUG or RFE
• *Submitter* -> current user
• *Date* -> current
• *Owner* -> czar
• *Due date* ->30 days from current
• *Summary* -> 1st description line
2. Screen request

Figure 2-4 Submittal Phase

Browsing and Displaying Requests

You can browse through requests from any of the RTS applications. However, *rtsquery* is best suited to browsing because it is the only application that contains all the available request fields. The other applications use queries to display requests prior to operating on them.

Browsing Procedure

To browse requests in the database, perform the following steps:

1. Select “Query” from the Modes menu if it is not already selected. All fields in the window become writable.
2. Enter the desired selection criteria in the appropriate fields.

   Leaving all fields blank brings all requests in the database into the list, with the first few displayed in the query results portion of the window. Entering one or more fields displays those requests with matching fields (with regard to case-sensitivity). Entering multiple fields has the effect of *and*ing the fields. You can also use the operators in Table 2-1 to refine your queries. When you use these operators, you should enclose all values inside quotation marks (“”).

Table 2-1 Logical Operators

Operator	Description
=	equal to
<>	not equal to
<	less than[a]
<=	less than or equal to
>	greater than
>=	greater than or equal to
match	regular expression match
contains	contains specified value (list fields only)
contains any	contains any of the specified values (list fields only)
contains only	contains only the specified values (list fields only)
= null	unset or non-existent test
<> null	set or existing test
[startrange:endrange]	range entry

[a] The less-than (<) and greater-than (>) operators can be used with dates to mean before and after respectively. For example, >June 25 means after June 25.

3. Click the <*apply command*> button (which displays the command *Query*) to execute the query with your selection criteria.

   All appropriate requests will appear in summary form in the query results area. Figure 2-5 is a typical example of entering selection criteria. It includes all requests in the database owned by Linda, submitted after May 25, 1993.

Figure 2-5 Query Example after Query

4. To display the detailed information for any request in the query results area, you can double-click the request directly, or you can click one of the list control buttons: *First*, *Next*, *Prev*, and *Last*.

   The selected request displays in the request form area of the window. Figure 2-6 shows the *rtsquery* window in display mode.

   When you display a request, a small rectangular icon appears to the left of the request line in the query results area. If you modify the request, a check mark will appear next to the request as well. If you make a subsequent query, the icons will be cleared.

Figure 2-6 Query Example after Display

Query Strategies

All RTS applications let you save different sets of selection criteria and load them when you want to reuse them. For example, you might want to save different sets of criteria with some combination of yourself as owner and one or more of the following fields:

- Project
- Priority
- Due Date

In addition, you can specify a default set of criteria to be loaded when you start up a specific application. For example, if you are doing approval work, you can specify a default of status equal to AWAITING_APPROVAL for running *rtsapprove*

Screening Requests

Only authorized users can screen requests. You begin by browsing and displaying requests, and can then apply different transitions after the selected request displays.

Screening requests follows all the steps mentioned in the section "Browsing and Displaying Requests." The difference is that you deal only with requests in the AWAITING_RESPONSE state.

No Action Commands

The screener has three transition commands for turning down requests:

- "DUPLICATE," if a report on the same topic has been previously submitted.
- "DEFER," if work on the request is to be performed at a later date. This requires an entry in the *Reopen Date* field. Tracker automatically reopens the request on the specified date, switching its status from CLOSED to AWAITING_RESPONSE.
- "REJECT" turns down the request as invalid.

Each command changes the request status from AWAITING_RESPONSE to AWAITING_APPROVAL.

Request Assignment

The "ASSIGN" transition command lets the screener specify the owner who will be executing the request. "ASSIGN" requires that the *Owner* field be changed. If the owner serves as screener, then the "ASSIGN" command is not appropriate.

The screening process is summarized in Figure 2-7.

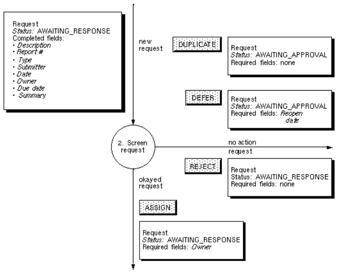


Figure 2-7 Screening Requests

Executing and Reopening Requests

After a request has been screened and okayed, it is turned over to the owner for implementation. (If the owner performed the screening, there is no turnover.) In addition to new requests from the screener, the owner may receive rejected resolutions (redo requests) from the approver.

The request execution process is summarized in Figure 2-8.

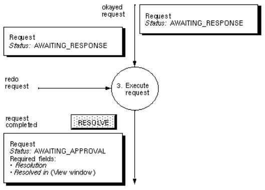


Figure 2-8 Request Execution Process

You can use either *rtsquery* or *rtsrespond* to execute a request. Figure 2-9 shows the *rtsrespond* main view and Modes menu.

Figure 2-9 *rtsrespond* Main View with Modes Menu

To modify an existing request, you first have to bring it up, as discussed in the “Browsing and Displaying Requests” section. When the request is displayed, perform these steps:

1. Select “RESOLVE” from the Modes menu.

   The *Resolution* field in the main view and the *Resolved in* field (in *rtsfiles* if you are using *rtsquery*) are highlighted.

2. Enter the explanation of the fix in the *Resolution* field and any files changed as a result of the request in the *Resolved in* field.

3. Click the <*apply command*> button, which now reads *RESOLVE.*

   The status changes to AWAITING_APPROVAL.

Approving Requests

Approving requests is similar to screening requests in that it requires authority. After the owner has executed the request, the approver must okay it before it can be closed. If the screener declines the request due to rejection, duplication, or deferral, the approver must approve this decision as well.

The first step is to perform a query to create a list of requests in the AWAITING_APPROVAL state. The “APPROVE” command requires entries in the *Approver* and *Close date* fields. The defaults for these fields are the current user and date respectively. When you approve a request, its state changes to CLOSED. If you reject a request, the state becomes AWAITING_RESPONSE.

The approval process is illustrated in Figure 2-10.

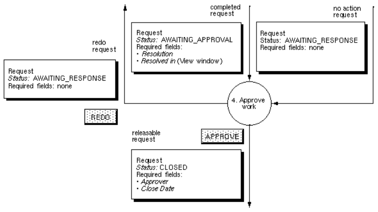


Figure 2-10 Approval Process

You can approve requests using either *rtsquery* or *rtsapprove* (see Figure 2-11).

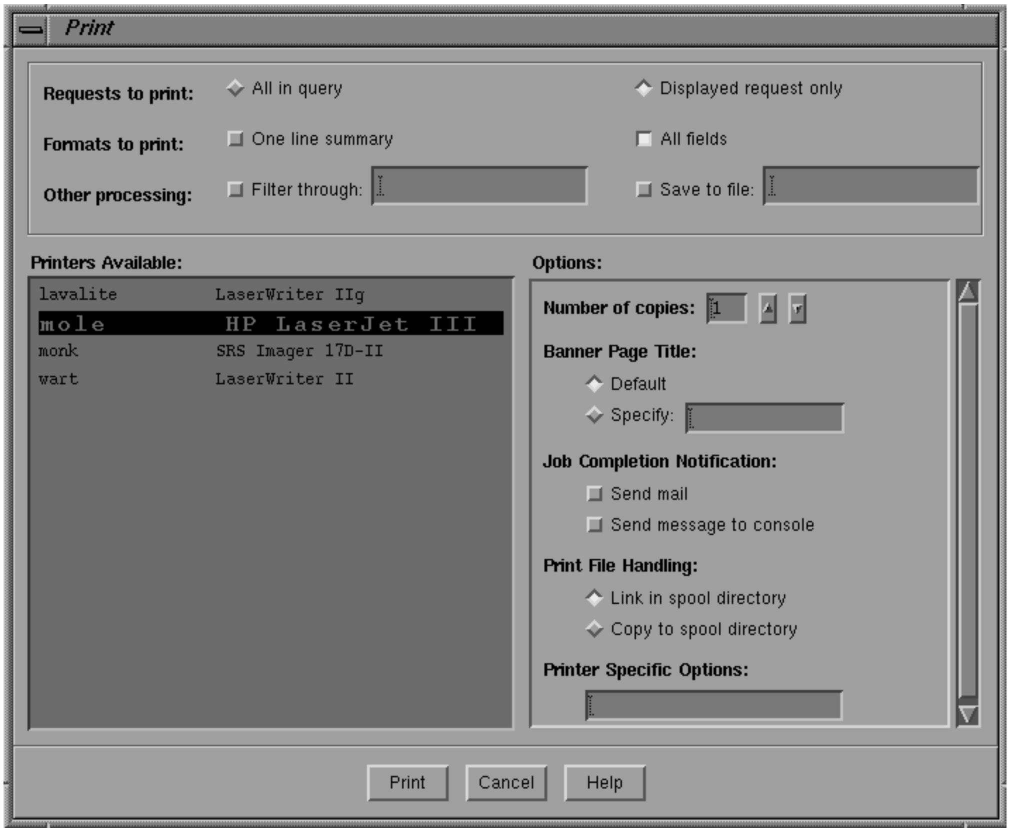
Figure 2-11 *rtsapprove* Main View with Modes Menu

Printing Reports

You can use the Print option on the File menu to print a request or group of requests in a report, or you can save the information in a file. The Print menu is shown in Figure 2-12.

The diamond buttons indicate a “one-of” choice; the square buttons mean you can select any or all of the options offered. When you select the *Save to file* option, the **`Print`** button at the bottom of the menu changes to **`Save`**.

The *Filter through:* option lets you use your own filter to create a suitable report format for the printed requests. Try the sample filter in NAME.



Print

Requests to print: ◇ All in query ◆ Displayed request only

Formats to print: □ One line summary ☑ All fields

Other processing: □ Filter through: □ Save to file:

Printers Available:

```
lavalite      LaserWriter IIg
mole          HP LaserJet III
monk          SRS Imager 17D-II
wart          LaserWriter II
```

Options:

Number of copies: 1

Banner Page Title:

◆ Default

◇ Specify:

Job Completion Notification:

□ Send mail

□ Send message to console

Print File Handling:

◆ Link in spool directory

◇ Copy to spool directory

Printer Specific Options:

Figure 2-12 *Print* Menu Options

Chapter 3

RTS Application Reference

This chapter describes these features of *rtsquery*:

- *rtsquery* main view
- Tracker menus
- Control bar
- Query results area
- Request form area

rtsquery Main View

The RTS application main view windows all have the same parts, although their menus and request form areas may differ.

The main view for *rtsquery* is shown in Figure 3-1, with its menus and parts labeled. It contains:

- a menu bar with five menus: File, Edit, Query, Views, and Help
- a control bar for browsing, displaying, and performing operations on requests
- a query results area containing requests matching the criteria of the most recent query and indicating accessed and modified requests
- a request form containing detailed information

Figure 3-1 *rtsquery* Window with Menus

Tracker Menus

Up to five menus are provided in Tracker application windows: File, Edit, Query, Views, and Help. If the application has no auxiliary views, there is no Views menu.

File Menu

The *File menu* (see Figure 3-2) provides these selections:

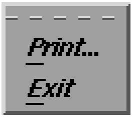


Figure 3-2 File Menu

"Print..." lets you print a request. It displays a dialog box for you to select printing specifications.

"Exit" quits the application

"Hide appears in auxiliary views only, in place of "Exit." It closes the auxiliary view's window.

Edit Menu

The *Edit menu* (see Figure 3-3) provides these selections:

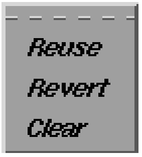


Figure 3-3 Edit Menu

"Reuse" fills all fields with data from the last request

"Revert" restores the values entered previously for the current request

"Clear" blanks out all fields

Query Menu

The *Query menu* (see Figure 3-4) lets you reuse query selection criteria rather than enter it in the request fields each time you want that combination. Its selections are:

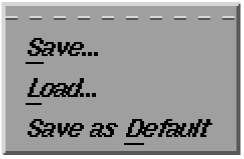


Figure 3-4 Query Menu

"Save..." stores the current set of criteria

"Load..." enters a saved set of criteria into the appropriate fields

"Save as Default"
saves your current criteria and reuses it as a default when you start up this RTS application.

"Case Sensitive Matches"
toggles case sensitivity on or off when you are searching for a string match in a field

Views Menu

The *Views menu* (see Figure 3-5) lets you display auxiliary views. The standard version of *rtsquery* provides one selection "rtsfiles." Your particular configuration may have additional views.

Figure 3-5 Views Menu

"rtsfiles" displays the *rtsfiles* window (see Figure 3-6), which has three fields for entering file names:

Found in: lets you enter or access problem files.

Resolved in: lets you enter changed files.

Fixed Releases: lets you enter final files.

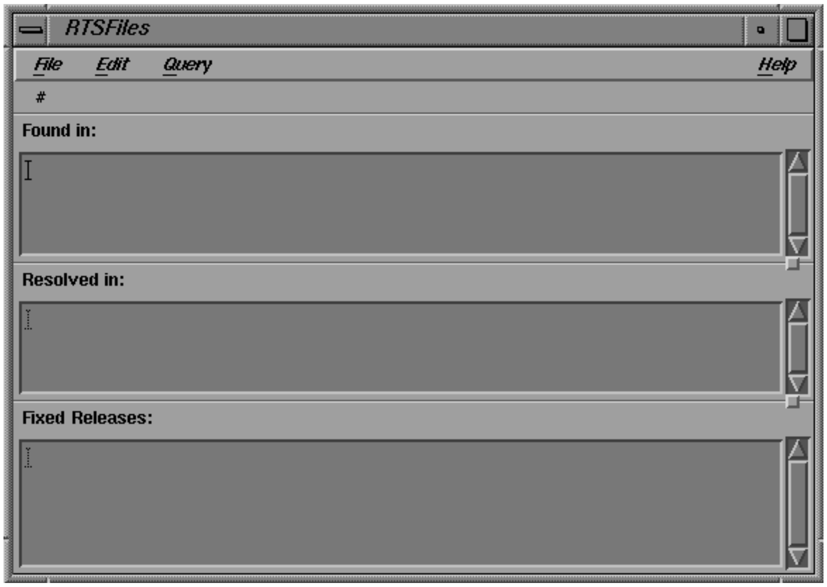


Figure 3-6 *rtsfiles* Window

Help Menu

The *Help menu* (see Figure 3-7) is the same for all CASEVision tools from Silicon Graphics.

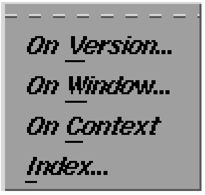


Figure 3-7 Help Menu

"On Version..." displays copyright and version number information

"On Window..." displays the Help Viewer with an overview of the current window's usage

"On Context" lets you select a feature with the cursor and display the Help Viewer describing that feature

"Index..." displays the Help Browser, which lets you browse a list of Help topics and make a request

Control Bar

The control bar lets you perform queries on the request database, access individual requests, and perform operations on them. The control bar for *rtsquery* is shown in Figure 3-8. It contains:

- the Modes menu for resetting the window to accept an operation
- a *Cancel* button for canceling an operation
- an <*apply command*> button for completing an operation. It displays the command for the current operation.
- list controls for displaying the detailed information for requests in the query results area (see the next section, "Query Results Area").

Figure 3-8 Control Bar of *rtsquery* with Modes Menu

Modes Menu

The Modes menu displays the commands that can be performed on requests in the database. These commands are enabled or disabled, according to your currently selected request (if any) and its state. For example, if no detail information is displayed, the system assumes that you are either going to inspect requests or submit new ones so that only the top four menu items ("Query," "Display," "SUBMIT_BUG," and SUBMIT_RFE") are enabled.

When you select a command from the Modes menu, the rest of the window changes according to the information displayed and the read / write permission on the fields. The buttons, *Cancel* and <*apply command*>, let you cancel or apply the currently selected command. They are enabled only when appropriate; for example, <*apply command*> is enabled only after you enter data in all required fields and move the cursor out of the last field.

Performing Operations

In general, operations are performed in three steps:

1. Select the operation from the Modes menu.
2. Select or change the data necessary for the operation.
3. Complete the operation by clicking the <*apply command*> button or stop it by clicking *Cancel*.

The Modes menu provides two types of commands: *inspection commands* and *transition commands*. The inspection commands “Query” and “Display” let you look up requests, in a list of summaries or individually in detail. The *transition commands* change either the status of the request or the fields in the report. Any field required for a specific transition will be highlighted, indicating that an entry is necessary. If no entry is made, then the command will be blocked from completion.

List Control Buttons

The list control buttons are:

First	displays the first request in the list.
Next	displays the next request in the list.
Previous	displays the prior request in the list.
Last	displays the last request in the list.

Modes Menu Selections

The Modes menu selections, shown in Figure 3-9, are:

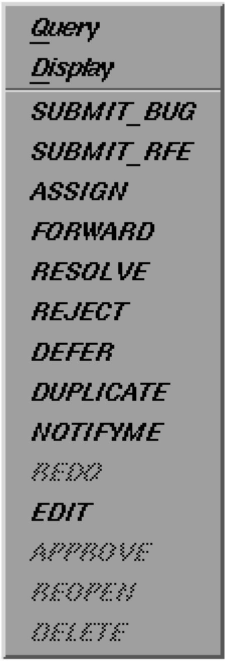


Figure 3-9 Modes Menu

“Query” lets you define a list of requests to display in the query results area by entering field values to be matched. (“Query” is case-sensitive.) For a more detailed description, refer to the section “Browsing and Displaying Requests” in Chapter 2.

“Display” indicates display mode, that is, you can display the detailed information for individual requests in the list. Clicking “Display” while in Query mode displays the first request in the list. While in display mode, you can select requests by clicking them, or you can use the list control buttons: *First*, *Next*, *Prev* (short for previous), and *Last*.

“SUBMIT_BUG”
lets you submit a request classified as a bug. It requires that you specify a description. “SUBMIT_BUG” fills in these fields by default unless you enter different information: *Type* (BUG), *Submitter* (current user), *Date* (current date),

Owner (project leader if known or else approving authority), *Due Date* (30 days from current), *Summary* (first line of description).

"SUBMIT_RFE"
lets you submit a request classified as an RFE (request for enhancement). "SUBMIT_RFE" fills in these fields by default unless you enter different information: *Type* (RFE), *Submitter* (current user), *Date* (current date), *Owner* (project leader if known or else approving authority), *Due Date* (30 days from current), *Summary* (first line of description).

"ASSIGN" lets you assign an owner and due date to a request. These two fields are required.

"FORWARD" lets you change the owner of the request. A different owner name is required.

"RESOLVE" indicates the request has been completed and is ready for approval. It requires an explanation of the fix in the *Resolution* field and the list of files changed in the *Resolved in* field (which is in *rtsfiles* view if you are using *rtsquery*). The *Recommend* field gets set to RESOLUTION.

"REJECT" lets you reject the request as invalid. Its state goes from AWAITING_RESPONSE to AWAITING_APPROVAL. An explanation must be made in the *Resolution* field. The *Recommend* field gets set to REJECTION.

"DEFER" lets you remove a request from AWAITING_RESPONSE state to AWAITING_APPROVAL state. You must provide an explanation in the *Resolution* field and set a new date in the *Reopen Date* field. The *Recommend* field gets set to DEFERRAL.

"DUPLICATE" lets you close out this request, because it is a duplicate of a prior one. You must enter an explanation in the *Resolution* field and enter the report number of the request duplicated by the current request in the *Dup of* field.

"NOTIFYME" lets you add yourself to the list of interested parties to be notified when changes occur to the current request.

“REDO” is for approvers only. It lets you return a request to its owner for further work. The state changes from AWAITING_APPROVAL back to AWAITING_RESPONSE.

“EDIT” lets you change fields in the request (other than *Report #* and *Status*).

“APPROVE” is for approvers only. It closes out the request and sets the value in the *Close Date* field to the current date, unless a different value has been entered.

“REOPEN” takes a request from the CLOSED state to AWAITING_RESPONSE and makes the usual notifications.

“DELETE” takes a request from the CLOSED state to DELETED, which means it will be removed from the request database.

Query Results Area

All query results areas in Tracker applications have the same general characteristics (see Figure 3-10).

Figure 3-10 Query Results Area

The number of requests in the current query is displayed at the top of the list area. The list itself displays information for identifying requests in the list. The vertical scroll bar displays different portions of the list; the horizontal scroll bar lets you display portions of request descriptions that are obscured from view. If you use any of the list controls to display a request in the form area, the list will scroll to the displayed request accordingly.

You can use the sort buttons to sort the list of requests according to one of the characteristics displayed. The list is sorted in ascending order, alphabetically or numerically, according to the chosen category.

Note: You cannot alter the sort order from the GUI. Items are always sorted in ascending order from the sort buttons.

The standard version of *rtsquery* presents requests in the query results area in this format:

```
<request ID> <request state> <owner> <summary>
```

Request Form Area

The fields that appear in the request form area depend on which application you are using and if your configuration has been modified. In total, these fields represent all the potential user-accessible data belonging to a request.

Field Menus

All Tracker applications provide right-button menus to help edit the request fields. Holding down the right mouse button while the cursor is over a field displays a popup menu.

If there are no standard values for the field, the menu contains the items “Reuse” (for using the value from the last displayed request), “Clear” (for blanking out the field), and “Revert” (for returning to the saved value).

If predefined values are available for the field, they can be displayed from the cascading menu attached to the “Values” item. If the last item in the Values menu is an ellipsis (. . .), you can enter values other than the ones displayed. Fields with a large number of value options may display these options in the form of a scrolling list. Figure 3-11 shows a list of projects.

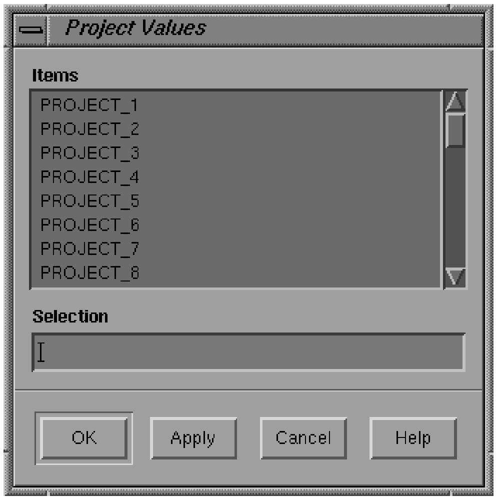


Figure 3-11 Scrolling List of Project Values

If the field allows lists, such as “Notify” in *rtsquery*, a menu like that in Figure 3-12 is displayed. “Format” has two options: “List,” a vertical presentation with a scroll bar, and “Text,” a linear presentation separated by commas. You can make entries only in text mode; the entries are made inside the parentheses, separated by commas. When in list mode, you can remove items from the list by choosing “Delete Selected.”

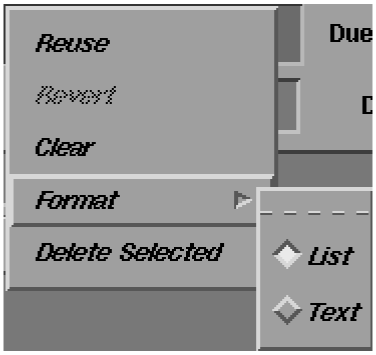


Figure 3-12 Format Menu Selections

If the field allows long text descriptions, a menu displays as shown in Figure 3-13. “Edit...” lets you bring up your editor of choice to edit the text. While the editor window is up, the RTS field is in read-only mode.

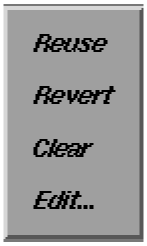


Figure 3-13 ”Edit...” Option

Use one of the following to set your default editor:

- the environment variable *$WINEDITOR* (for editors like *jot* that bring up their own windows)
- the environment variable *$EDITOR* (for editors like *vi* that do not supply their own windows)

- the *editorCommand* setting in *.Xdefaults*

If none of these are set, the editor defaults to *vi*.

Date Entry

Tracker provides a wide variety of date input formats. Dates can be supplied with as little information as the year or month, or specified to the nearest second.

Special formats allow dates to be supplied as a base time point plus or minus a time interval, and also as relative to the current year, month, day, hour or second. Date field values represent a point in time to the nearest second. They do not represent a time interval such as in *six seconds* or *two days*.

Date values are ordered from this starting point, the lowest value, and increasing toward later dates. Therefore, a date that occurs after a specified date will have a higher value. This lets you use comparison operators such as < (before) and > (after).

When you use dates in transitions, you enter a specific point in time. When you are using dates in queries, you can enter a range or condition as well as a specific time.

Date Entry for Transitions

Tracker lets you enter dates in most of the standard formats available in UNIX. Table 3-1 provides examples of date value entries, along with the interpretation by Tracker. These examples assume a current date and time of Tue Jul 28 11:10:14 PDT 1992.

Table 3-1 Date Formats

Date Input	Tracker Interpretation
7/28/92	Tue Jul 28 00:00:00 PDT 1992
July 28, 1992	Tue Jul 28 00:00:00 PDT 1992
July 28 1992	Tue Jul 28 00:00:00 PDT 1992

Table 3-1 (continued) Date Formats

Date Input	Tracker Interpretation
28-July-92	Tue Jul 28 00:00:00 PDT 1992
July 28 10:00 1992	Tue Jul 28 10:00:00 PDT 1992
10:00 July 28 1992	Tue Jul 28 10:00:00 PDT 1992
July 28	Tue Jul 28 00:00:00 PDT 1992
July 28 10PM	Tue Jul 28 22:00:00 PDT 1992
July 28 10PM EDT	Tue Jul 28 19:00:00 PDT 1992
July 1992	Wed Jul 1 00:00:00 PDT 1992
July	Wed Jul 1 00:00:00 PDT 1992
1992	Wed Jan 1 00:00:00 PST 1992
10PM	Tue Jul 28 22:00:00 PDT 1992
10:30:59 PM	Tue Jul 28 22:30:59 PDT 1992

In addition to entering the specific date and time, you can also use these expressions:

this second represents the current date and time, up to the second. The expression now has the same effect

this minute represents the current date and time, up to the minute (stored as the first second in that minute)

this hour represents the current date and time, up to the hour (stored as the first second in that hour)

this day represents the current date with no time specified (stored as the first second in that day). The expression `today` has the same effect.

this month represents the current date and time, (stored as the first second in the first day of the month)

this year represents the current date and time, (stored as the first second in the first day of the year)

Table 3-2 lists some current expressions of the date (assuming the date used in the previous example, Tue Jul 28 11:10:14 PDT 1992).

Table 3-2 Current Date Expressions

Date Input	Tracker Interpretation
today	Tue Jul 28 00:00:00 PDT 1992—Beginning of current day. Equivalent to `this day`.
now	Tue Jul 28 11:10:14 PDT 1992—Current time to the nearest second.
this year	Wed Jan 1 00:00:00 PST 1992—Beginning of current year.
this month	Wed Jul 1 00:00:00 PDT 1992—Beginning of current month.
today + 30:00:00:00	Thu Aug 27 00:00:00 PDT 1992—30 days from today.
July 28 - 72:00:00	Sat Jul 25 00:00:00 PDT 1992—72 hours before July 28th of the current year.

Date Entry for Queries

When you are performing queries involving dates, you often need multiple dates rather than a specific date. These entries are useful in queries:

- a date range in the form [*startdate:enddate*].
- an expression with the operators <, <=, =, >=, >, and <> (> means after the date and < means before).
- the current expressions mentioned above, such as `this year` or `this month`.
- an implied interval such as a day, month, or year. Tracker performs the query according to the specificity of the date unit entered.
- an expression with a base time plus or minus a time interval.

Field Definitions

This is a list of the standard fields in *rtsquery*, which is a superset of the other applications:

Report # is an integer used to identify requests, which is set by the application and cannot be modified. The field is read-only, except when you are performing queries.

Status refers to the state of the request, which can be AWAITING_RESPONSE, AWAITING_APPROVAL, CLOSED, and DELETED. Status cannot be edited directly; it is changed only by transitions.

Type represents the type of request: BUG or RFE. Your configuration may be set up to accept other types of requests. Check the “Values” selection in the right-button popup menu for other values.

Submitter identifies the original submitter of the request.

Date is the date and time when the request was submitted.

Recommend represents a response to a request. The permitted values are DEFERRAL, REJECTION, RESOLUTION, and DUPLICATION.

Note: Only transitions can change the state of a request. For example, using the “EDIT” command to change the *Recommend* field to RESOLUTION does not resolve the request; it only changes the value of this field.

Project identifies the software project. Initially, *rtsquery* has three placeholder project names: PROJECT_1, PROJECT_2, and PROJECT_3. To determine the projects on your system, hold down the right mouse button when the cursor is in this field and select “VALUES.”

Priority is the priority of the request on a scale of LOW, MEDIUM, and HIGH. Since your configuration may be different, hold down the right mouse button when the cursor is in the *Priority* field and select “VALUES” to see the valid priority values.

Owner identifies the person responsible for making the correction. For notification purposes, the name should be the same as that person’s email address.

System identifies the system to which the request is applied. Initially, *rtsquery* has three placeholder project names: SYSTEM_1, SYSTEM_2, and SYSTEM_3.

Notify identifies parties to be informed whenever there is a change to a request. *Notify* is a field of list type.

Due date is the target date for resolution of the request. The standard versions of *rtsquery* and *rtssubmit* set the date to 30 days from the submittal date by default. Your configuration may be different.

Close date is the actual date on which the request is closed by the approving authority.

Reopen date is the new effective submittal date when a request is deferred. This is a required field for the DEFER transition.

Approver establishes the approving authority. Check the “VALUES” item in the right-button menu to see if your system has a special configuration.

Dup of displays the report number of a duplicated request if the current request has been discarded as a duplicate.

Summary is a one-line description of the request. It defaults to the first line from the *Description* field after a “SUBMIT_BUG” or “SUBMIT_RFE” transition if no other information has been entered. You can also edit the *Summary* field using the “EDIT” command.

Description is the detailed description of the request. It is displayed in a scrollable area and thus can take as many lines as necessary for the full description.

Resolution displays a detailed explanation for a transition of the request. It is required for these transitions: RESOLVE, REJECT, DEFER, and DUPLICATE. It is displayed in a scrollable area and thus can take as many lines as necessary.

Found in is a list of file names in which problems related to the request are found.

Resolved in is a list of files that have changed as a result of the request. One or more entries are required in this field for the RESOLVE transition.

Fixed releases is a list of files that represent the software releases containing the fix to the request.

Index